Lifting in hybrid lattice Boltzmann and PDE models

Y. Vanderhoydonc and W. Vanroose

Dept. Mathematics and Computer Science, Universiteit Antwerpen

Abstract

Mathematical models based on kinetic equations are ubiquitous in the modeling of granular media, population dynamics of biological colonies, chemical reactions and many other scientific problems. These individual-based models are computationally very expensive because the evolution takes place in the phase space. Hybrid simulations can bring down this computational cost by replacing locally in the domain — in the regions where it is justified — the kinetic model with a more macroscopic description. This splits the computational domain into subdomains. The question is how to couple these models in a mathematically correct way with a lifting operator that maps the variables of the macroscopic partial differential equation to those of the kinetic model. Indeed, a kinetic model has typically more variables than a model based on a macroscopic partial differential equation and at each interface we need the missing data. In this contribution we report on different lifting operators for a hybrid simulation that combines a lattice Boltzmann model — a special discretization of the Boltzmann equation — with a diffusion partial differential equation. We focus on the numerical comparison of various lifting strategies.

Keywords: Boltzmann equation, Advection-diffusion-reaction equation, Lattice Boltzmann model, Hybrid spatial coupling, Lifting, Slow manifold, Constrained Runs, Smooth initialization

1 Introduction

In the last decades very efficient numerical methods have been developed to simulate and analyze mathematical models described by partial differential equations (PDEs) numerically.

A class of problems that can be efficiently solved is described by the advection-diffusion-reaction equation where an unknown function $u(x,t)$ fits the equation

$$\begin{aligned}\frac{\partial u(x,t)}{\partial t} &= \frac{\partial}{\partial x}\left(a(x,t)\frac{\partial}{\partial x}u(x,t)\right)+b(x,t)\frac{\partial u(x,t)}{\partial x}\\ &\quad +g(u(x,t)), \qquad (1)\end{aligned}$$

with appropriate boundary and initial conditions, $x \in D_x \subset \mathbb{R}$ and time $t > 0$. The coefficients $a(x,t)$ and $b(x,t)$ are, respectively, the diffusion and advection coefficient. The reaction term $g(u(x,t),x,t)$ is often nonlinear and can depend on time. This equation, and its generalizations to more dimensions, are frequently used to model systems in biology, physics, financial mathematics and many other fields of application. In these models $u(x,t)$ plays often the role of a density $\rho(x,t)$. To describe the PDE numerically, the spatial differential operators are approximated by, for example, finite differences. This transforms the PDE into a set of coupled ordinary differential equations (ODEs) with a sparse structure [1]. To arrive at a solution several very efficient methods have been developed such as sparse direct solvers [2], iterative methods based on a Krylov subspace [3] and multigrid methods [4].

However, there are complex systems that can not be accurately modeled mathematically by the advection-diffusion-reaction equation such as (1). Kinetic models, for example, which are frequently used in many areas of science, make use of the Boltzmann equation that describes the evolution of a distribution function $f(\mathbf{x},\mathbf{v},t)$ (function space $C^2_{\mathbb{R}}(D)$) that counts the number of parti-

Ynte Vanderhoydonc, Wim Vanroose
Universiteit Antwerpen, Dept. Mathematics and Computer Science, Middelheimlaan 1, 2020 Antwerpen, Belgium;
E-mail: ynte.vanderhoydonc@ua.ac.be, wim.vanroose@ua.ac.be

cles or individuals in point $\mathbf{x} \in D_{\mathbf{x}} \subset \mathbb{R}^n$, $n \in \mathbb{N}_0$, with a velocity $\mathbf{v} \in D_{\mathbf{v}} \subset \mathbb{R}^n$, at time $t \geq 0$. The equation is

$$\frac{\partial}{\partial t} f(\mathbf{x},\mathbf{v},t) + \mathbf{v}\frac{\partial}{\partial \mathbf{x}} f(\mathbf{x},\mathbf{v},t) + F(\mathbf{x},t)\frac{\partial}{\partial \mathbf{v}} f(\mathbf{x},\mathbf{v},t) = \Omega. \tag{2}$$

This is an evolution law in phase space where $F(\mathbf{x},t)$ is the external force and Ω an integral operator that models the reorganization of the velocity distribution due to collisions or other interactions. This type of equation is used, amongst others, to model traffic flow, where the right hand side models the behavior of drivers and the interaction between vehicles [5]. In biology these equations describe colonies of bacteria or other population dynamics [6]. Another application of the Boltzmann equation is found in plasma physics where the equation is coupled to the electromagnetic field. This coupled equation is known as the Vlasov equation [7]. The particle then moves in the external field while the source for the field is the particle charge density. At the moment it is computationally much more expensive to simulate or analyze numerically a Boltzmann model than an advection-diffusion-reaction model.

The aim of this research is to develop efficient numerical methods for models based on the Boltzmann equation.

One way to increase the efficiency is to use a hybrid model that uses a reduced description in the form of (1) where possible, for example, away from reaction fronts. The spatial domain is then split in heterogeneous pieces each with a different mathematical model, a Boltzmann model where necessary, an advection-diffusion-reaction PDE model where possible. For example, in [8] a hybrid model is presented for combined particle and continuum dynamics where a Lennard-Jones liquid is coupled to the compressible Navier-Stokes equations. Coupled models also play an important role in other applications, especially in simulations of materials. A review of atomistic-to-continuum coupling is found in [9]. To give a more detailed overview of some literature for hybrid models, we refer to [10], [11] and [12]. In [10] adaptive mesh and algorithm refinement is used in parts of the domain where a continuum description is replaced by a particle description. The reason for this is that a continuum description is not sufficient when the grid spacing approaches the fine scale. Dupuis *et. al.* [11] couple molecular dynamics with lattice Boltzmann models (LBMs), special discretizations of the Boltzmann equation, by using a decomposition algorithm based on the Schwarz alternating method. [12] is of particular importance for this work. A one-dimensional domain is considered that consists of a lattice Boltzmann model in one subdomain and a diffusion partial differential equation in an other. In this work the same model problem will be applied.

Another way to increase the efficiency is to use a multilevel algorithm. In [13] such a method is presented to compute steady states of LBMs. A macroscopic to microscopic mapping (e.g. map from density to distribution functions) is used to construct a multigrid solver for the steady states. Traditional geometric multigrid uses a hierarchy of discretizations of the PDE, each with a different grid distance, to construct an iterative solver. Appropriate restrict and lifting operators transfer information between the levels. With this mapping from density to distribution functions, it is possible to construct a lifting operator that brings information from a PDE model to a Boltzmann model. It is then possible to build a hierarchy for the kinetic model where the coarse levels are PDE models rather than the Boltzmann model discretized on a different grid. This leads to an efficient and scalable solver.

This work studies lifting operators for hybrid models that spatially couple a PDE model and a LBM. It extends the work of Van Leemput *et. al.* [12, 14] and Vandekerckhove *et. al.* [15]. In particular, we look at two numerical lifting procedures that work on general problems. One is based on the slow manifold that attracts the dynamics of the LBM, Sect. 3. The other is based on smooth initialization and discussed in Sect. 4. Both express in an implicit way the unknown distribution functions with the help of the density in successive grid points.

It is well known that in phase space the dynamics are quickly attracted to a slow manifold [16] that characterizes the long-term dynamical behavior of the process. On this slow manifold the dynamics can be parametrized by a macroscopic variable such as the density $\rho(x,t)$. This forms the basis of a numerical lifting operator that searches the missing distribution function $f(x,v,t)$ by determining its moments $\phi(x,t) = \int v f(x,v,t)dv$ and $\xi(x,t) = \frac{1}{2}\int v^2 f(x,v,t)dv$ for a given $\rho(x,t) = \int f(x,v,t)dv$ such that $\phi(x,t)$, $\xi(x,t)$ and $\rho(x,t)$ lie on the slow manifold. It can be interpreted as finding the point of intersection between the slow manifold and the plane determined by $\rho(x,t)$.

In [17] it is proven for a diffusion LBM that the point of intersection can be found by a stationary iteration. The algorithm is called *Constrained Runs* (CR) and it resets, after a few steps with the LBM, the density ρ to ρ_0, its initial value. During this iteration the Boltzmann model approaches the slow manifold (Figure 3). It is further proven that the iteration is always stable and the point of intersection is found to first order accuracy compared to the Chapman-Enskog (see Sect. 2.2) relation for a constant extrapolation in time. This corresponds to one step with the LBM before resetting the density.

The higher order versions (higher order extrapolation) of this Constrained Runs algorithm can be unstable. In [15] the instability is circumvented by formulating the point of intersection as a fixed point problem which is found by a Newton-Krylov iteration. This was then used to create a lifting operator for the full state. A serious drawback is that this method requires the construction and solution of large Jacobian systems, which can make it prohibitive for realistic applications. The general properties of the CR-algorithm, independent of the Boltzmann equation, have been analyzed in [18]. The CR-algorithm will be considered for our model problem in Sect. 3.2.

Formulating the lifting operator as a *missing data problem* is an alternative way to arrive at an implicit expression for the unknown distribution function $f(x,v,t)$. A recent technique to fill in missing data is smooth initialization [14]. In a missing data problem, variable $w \in W$ is split into $w=(u,v)$, where $u \in U$ contains the explicitly given data and $v \in V$ the missing data. These variables are considered in normed vector spaces U, V and W, in which W is the direct sum of the normed spaces U and V. In [14] this method is used to create a lifting operator for a lattice Boltzmann scheme with two possible velocity directions in a one-dimensional domain.

In this article we evaluate and compare lifting operators based on higher order Constrained Runs and smooth initialization. In particular, our interest lies in accurate lifting operators for hybrid models that combine a LBM and PDE model in different regions of space. We benchmark the lifting operators to a model problem that is representative for hybrid LBM-PDE models.

The remainder of this work is organized as follows. In Sect. 2 the model problem is defined, which consists of a LBM in one part of the domain and a diffusion PDE in another part. Sect. 2.2 contains some lifting techniques that will be used to couple the different models in the model problem. In particular, a numerical comparison of some existing lifting operators applied in a simple one-dimensional problem is given. An analytically known lifting operator is the Chapman-Enskog expansion. When the analytical expressions are unknown or tedious to derive, the Constrained Runs algorithm (Sect. 3) or a smooth initialization procedure (Sect. 4) can be used. The main drawbacks of these methods are the computational expense or the need to derive analytical expressions. We focus on local updates of these numerical methods. When considering a hybrid model, the lifting operator is only needed at the interface between the different models, which means that the lifting operator can just be used locally at the interface. We conclude and give an outlook in Sect. 5.

2 Model problem

A lattice Boltzmann model [19], [20] is a special discretization of (2). It describes the evolution of one-particle distribution functions $f_i(x,t) = f(x,v_i,t)$ at time $t \in [0,T]$ in point $x \in D \subset \mathbb{R}$ with velocity v_i. The velocities are taken from a discrete set defined by the geometry of the grid. The functions are represented as $f_i : \mathscr{X} \times \mathscr{T} \to \mathbb{R}$ with $\mathscr{X} \times \mathscr{T}$ the space-time grid with space step Δx, time step Δt and $\mathscr{T} = \{0, \Delta t, 2\Delta t, \ldots\} \cap [0,T]$. We consider a one-dimensional domain with only three values for the velocity v_i, $i \in \{-1,0,1\}$

$$v_i = c_i \frac{\Delta x}{\Delta t}, \tag{3}$$

with $c_i = i$, $i \in \{-1,0,1\}$ the dimensionless grid velocities. The lattice Boltzmann equation (LBE) describing the evolution of the distribution functions (with Bhatnagar-Gross-Krook (BGK) approximation for the collision term [21] and no external force in (2)) is

$$f_i(x + c_i\Delta x, t + \Delta t) = (1-\omega) f_i(x,t) + \omega f_i^{eq}(x,t). \tag{4}$$

The equilibrium distributions are given by $f_i^{eq}(x,t) = \frac{1}{3}\rho(x,t)$, $i \in \{-1,0,1\}$. The particle density $\rho(x,t)$ is defined as the zeroth order moment of the distribution functions.

$$\rho(x,t) = \sum_{i\in\{-1,0,1\}} f_i(x,t).$$

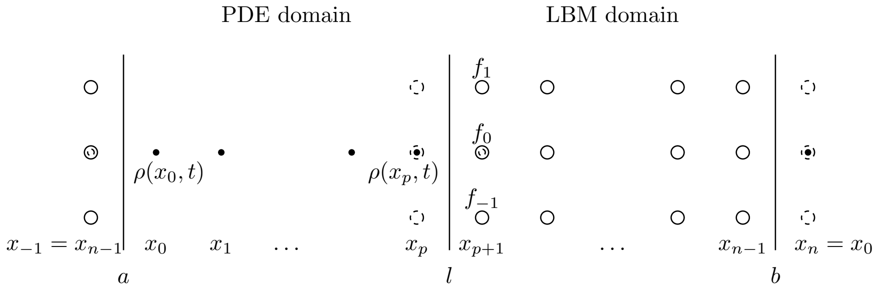

Figure 1: The domain $[a,b]$ in the hybrid model is split into $[a,l[$ on which we solve the PDE model and $[l,b]$ on which we solve the LBM. The solid points (•) represent the grid for the density ρ of the discrete PDE, the circles (○) represent the LBM variables $(f_1, f_0, f_{-1})^T$. The periodic boundary conditions and the coupling are implemented with ghostcells which are drawn by dashed circles. The density in the ghostcells of the PDE domain, in x_{-1} and x_{p+1}, are found by taking $\sum_i f_i$ in x_{n-1} and x_{p+1}, respectively. However, the ghostcells for the LBM domain, in x_p and x_n, require a lifting operator that lifts ρ to $(f_1, f_0, f_{-1})^T$ in these points.

These equilibrium distributions correspond to a local diffusive equilibrium. The BGK approximation represents a relaxation to the equilibrium with associated time scale $\tau = \frac{1}{\omega}$.

Partial differential equations form an alternative for an individual-based description like the Boltzmann equation. They model, at a macroscopic scale, the evolution of the moments of the particle distribution functions like density $\rho(x,t) = \sum_i f_i(x,t)$, momentum $\phi(x,t) = \sum_i v_i f_i(x,t)$ or energy $\xi(x,t) = \frac{1}{2}\sum_i v_i^2 f_i(x,t)$. It is equivalent to determine $\{f_1, f_0, f_{-1}\}$ or $\{\rho, \phi, \xi\}$. The transition between the distribution functions and the moments is straightforward since the matrix M below is invertible.

$$\begin{pmatrix}\rho\\ \phi\\ \xi\end{pmatrix} = \begin{pmatrix}1 & 1 & 1\\ 1 & 0 & -1\\ \frac{1}{2} & 0 & \frac{1}{2}\end{pmatrix}\begin{pmatrix}f_1\\ f_0\\ f_{-1}\end{pmatrix} = M\begin{pmatrix}f_1\\ f_0\\ f_{-1}\end{pmatrix}. \qquad (5)$$

When we look at these functions in a point x at a time t, they can be represented either as $(f_1, f_0, f_{-1})^T \in \mathbb{R}^3$ or as $(\rho, \phi, \xi)^T \in \mathbb{R}^3$. If we focus on the complete discretization in space, with n the number of spatial grid points, the function spaces are $\mathbb{R}^{3\times n}$. It can be shown that the diffusion PDE and the LBM are macroscopically equivalent [22] when

$$\frac{\partial\rho(x,t)}{\partial t} = D\frac{\partial^2\rho(x,t)}{\partial x^2}, \quad D = \frac{2-\omega}{3\omega}\frac{\Delta x^2}{\Delta t}. \qquad (6)$$

This can be checked by using a Chapman-Enskog expansion. Here $f_i(x,t)$, $i \in \{-1,0,1\}$ is written as a series, each term containing higher order derivatives of $\rho(x,t)$ [23].

2.1 Hybrid LBM and PDE model problem

In this paper we study a model problem with periodic boundary conditions. The model problem is shown in Figure 1. A one-dimensional domain $[a,b]$ is considered that couples the diffusion PDE (6) on $[a,l[$ with the LBE (4) on $[l,b]$. Since the focus of the work is on the error caused by the coupling, we use the same grid spacings in space (Δx) and in time (Δt) for both regions. Using a different space-time grid is of particular interest for future work since the grid of the PDE domain can be further coarsened when ρ is smooth. Then some interpolation operator needs to be considered at the coupling point. [24] proposes a method to couple grids with different mesh sizes for classical lattice Boltzmann schemes. Numerical convergence studies are performed there on a simple lattice Boltzmann algorithm solving the advection-diffusion equation.

On the domain $[a,l[$, we discretize the diffusion PDE with cell centered central differences and forward Euler time discretization. The grid points x_j with $j \in \{0,1,\dots,p\}$ cover this domain and for these points it holds that

$$\begin{aligned}\rho(x_j,t+\Delta t) &= \rho(x_j,t) \\ &+\frac{D\Delta t}{\Delta x^2}\left(\rho(x_{j-1},t)-2\rho(x_j,t)+\rho(x_{j+1},t)\right).\end{aligned} \tag{7}$$

On the domain $[l,b]$ we have the LBM for the grid points x_j with $j\in\{p+1,\ldots,n-1\}$

$$f_i(x_j+c_i\Delta x,t+\Delta t)=(1-\omega)f_i(x_j,t)+\omega f_i^{eq}(x_j,t). \tag{8}$$

The periodic boundary conditions lead to the following boundary conditions for the PDE domain

$$\begin{aligned}\forall t:\rho(x_{-1},t)&=\sum_i f_i(x_{n-1},t),\\ \forall t:\rho(x_{p+1},t)&=\sum_i f_i(x_{p+1},t).\end{aligned} \tag{9}$$

The full domain has an initial condition $\forall j\in\{0,\ldots,n-1\}$

$$\rho(x_j,0)=q(x_j). \tag{10}$$

The aim is to construct the boundary conditions of the LBM domain in such a way that $\forall t>0$ and $\forall j\in\{0,\ldots,n-1\}$ the macroscopic density defined as

$$\rho(x_j,t)=\begin{cases}\rho(x_j,t) & \text{if} \quad j\in\{0,\ldots,p\},\\ \sum_i f_i(x_j,t) & \text{if} \quad j\in\{p+1,\ldots,n-1\},\end{cases} \tag{11}$$

behaves as the density of a LBM solved on the full domain.

To formulate these boundary conditions, a lifting operator is required that maps the density $\rho(x,t)$ in the ghost points x_0 and x_p, the unknown of the PDE, to the distribution functions $f_i(x,t)$, $i\in\{-1,0,1\}$ of the LBM.

2.2 Lifting the initial state and the ghost points

The lifting operator is not only needed to construct the distribution functions in the ghost points, but also to construct an appropriate initial condition in the LBM region starting from $\rho(x,t=0)$. Also for the reference solution that solves the LBM over the full domain, we need to construct distribution functions from $\rho(x,t=0)$.

A traditional approach to construct a $\rho(x,t)$ to $f(x,v,t)$ map is based on the Chapman-Enskog expansion. Its use is, however, limited to a few analytically solvable model problems. For the diffusion equation that we are discussing the series is given by [12]

$$\begin{aligned}f_i(x,t) \quad &= \quad \frac{1}{3}\rho(x,t)-\frac{c_i\Delta x}{3\omega}\frac{\partial\rho(x,t)}{\partial x}\\ &\quad -\frac{\Delta x^2}{18\omega^2}(\omega-2)(3c_i^2-2)\frac{\partial^2\rho(x,t)}{\partial x^2}\\ &\quad +\mathscr{O}(\Delta x^3),\end{aligned} \tag{12}$$

where we approximate the derivatives with the help of finite differences. The distribution functions are then expressed, in an explicit way, as a function of the density in three successive grid points.

A comparison figure up to zeroth order in Δx of (12) is given in Figure 2 (top). It plots the absolute difference $|\rho_{\text{hybrid}}(x,t)-\rho_{\text{LBM}}(x,t)|$, which consists of ρ_{hybrid}, the density of the hybrid model with assumption $f_i(x,t)=\frac{1}{3}\rho(x,t)$, $i\in\{-1,0,1\}$ — in both the ghost points and the initial distribution functions of the LBM — and ρ_{LBM}, the density of a full LBM. Full LBM considers a LBM on the whole spatial domain $[a,b]$. The parameters that are used throughout the paper are 200 time steps with $\Delta t=0.001$, $\Delta x=0.05$, $D=1$, $\rho(x,0)=\exp\left(-(x-\frac{L}{2})^2\right)$ with L the length of the spatial domain.

This comparison shows two clear errors: an initial error and an error in the points where the different models are coupled. These errors diffuse over the domain. Both errors occur because of the assumption $f_i(x,t)=\frac{1}{3}\rho(x,t)$, $i\in\{-1,0,1\}$, the first term of the Chapman-Enskog expansion. It is obvious that this lifting operator is not sufficient.

The error decreases when higher order terms in the expansion are used in the lifting. The remaining figures of Figure 2 show the absolute differences with the Chapman-Enskog expansion up to first, second, respectively third order as a lifting operator. However, the Chapman-Enskog expansion is hard to derive for general Boltzmann models, especially if there are velocity dependent collision rates in the model. In the remaining of this paper, it is shown that even the third order relation can be improved by considering numerical techniques to find the lifting operator.

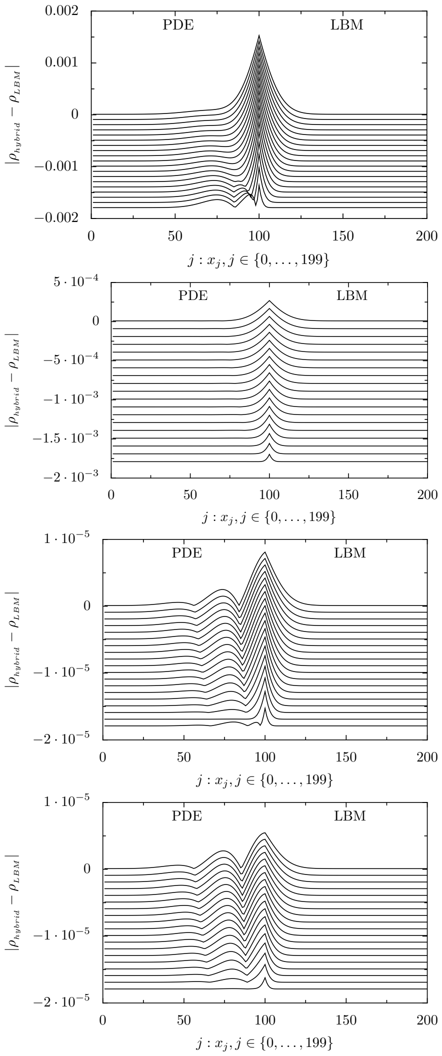

Figure 2: The absolute value of the difference between the density of the hybrid model and the full LBM after 200 time steps. The difference is also shown at earlier time slots, but shifted down for clarity. The lines represent time steps between one and 200. The top line corresponds to time step 191 while the bottom line represents time step 11. The lines in between correspond to jumps with 10 time steps from 11 to 21,...,181 to 191. The domain is the interval [0,10] with 100 cell centered grid points on [0,5[, the PDE domain, and 100 LBM grid points on [5,10]. The lifting operator is $f_i(x,t) = 1/3\rho(x,t)$ (top), first order, second order and third order Chapman-Enskog (bottom) respectively. The lifting operator is used both to find the ghost points of the LBM domain and for the creation of the initial state for the LBM region. The model parameters are $\Delta t = 0.001$, $\Delta x = 0.05$, $D = 1$, $\rho(x,0) = \exp\left(-(x-\frac{L}{2})^2\right)$. These variables define ω in Eq. (4) through Eq. (6).

3 Lifting with Constrained Runs

An evolution of the long-term dynamics is quickly attracted to a slow manifold [16], a lower dimensional part of phase space. On this slow manifold the dynamics can typically be described by macroscopic variables such as the density $\rho(x,t)$. By using the idea of the slow manifold, a numerical lifting operator is constructed that searches the missing distribution functions by determining ϕ and ξ for a given ρ such that these moments lie on the slow manifold. The interpretation is represented in Figure 3. Find, with the help of the underlying Boltzmann model, the point of intersection between the slow manifold and the plane determined by ρ numerically.

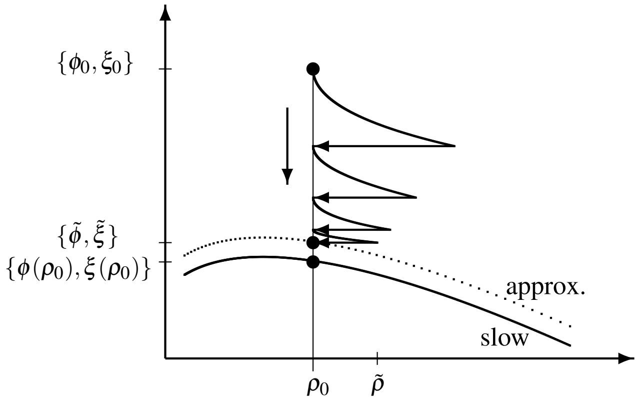
$\{\phi_0, \xi_0\}$
$\{\tilde{\phi}, \tilde{\xi}\}$
$\{\phi(\rho_0), \xi(\rho_0)\}$
approx.
slow
ρ_0
$\tilde{\rho}$

Figure 3: Schematic representation of the Constrained Runs algorithm for the LBM with a constant backward extrapolation in time [12]. This algorithm converges toward the slow manifold. ρ is expressed on the horizontal axis and the missing moments ϕ and ξ on the vertical axis. We start iterating with ρ_0, the known density and initial guesses ϕ_0 and ξ_0 for the missing moments. After each step of the LBM, the density is reset to its initial value. This algorithm gives an approximation for the missing values ϕ and ξ on the slow manifold. This figure is reproduced from [22].

3.1 Constrained Runs algorithm

The goal is to find the point of intersection in Figure 3. This results in the CR-algorithm that finds its origin in systems of ODEs. Given system

$$\begin{cases} \frac{\partial u(t)}{\partial t} = p(u(t), v(t)), \\ \frac{\partial v(t)}{\partial t} = q(u(t), v(t)), \end{cases}$$

where only the initial condition for u, namely $u(0) = u_0$, is given. The aim is to find $v(0) = v_0$ such that the solution of the system evolves close to the slow manifold, which can be seen as the graph of the function $v = v(u)$.

Gear *et. al.* [16] proposed to obtain the v-value from equation

$$\frac{\mathrm{d}^{m+1} v(t=0)}{\mathrm{d}t^{m+1}} = 0, \tag{13}$$

that can be approximated by a forward difference

$$\Delta^{m+1} v(t) \approx \Delta t^{m+1} \frac{\mathrm{d}^{m+1} v(t)}{\mathrm{d}t^{m+1}}.$$

It can be shown that this difference approximation used in the CR-algorithm can be interpreted as a backward extrapolation [25]. It corresponds with a backward extrapolation in time based on a polynomial of degree m that passes through the values v_k $(k = 1, \ldots, m+1)$ while the known variable u is reset to its original initial value u_0.

The general algorithm for a constant extrapolation, $m = 0$, is given by

1. Initialize v_0.
2. Advance the model with one time step Δt. This results in values u_1 and v_1 at time $t = \Delta t$. Since u_0 is given, it is reset to u_0.
3. Difference approximation $\delta = \Delta v_0 = v_1 - v_0$.
4. $\|\delta\| \leq$ tol: end iteration, $\|\delta\| >$ tol: $v_0 \leftarrow v_0 + \delta$ and move to step 2.

3.2 Application of Constrained Runs to the LBM

We now apply the CR-algorithm to find the missing moments in the LBM. We look at Eq. (13) formulated as a fixed point problem [25].

The missing moments are denoted as

$$v = \begin{pmatrix} \phi \\ \xi \end{pmatrix},$$

a long vector $v \in \mathbb{R}^{2n}$, the variable $u = \rho \in \mathbb{R}^n$ denotes the known initial conditions $u(0) = u_0 \in \mathbb{R}^n$. Here n is the number of spatial grid points.

The vector v^k denotes the k-th iterate of the CR-algorithm and the iterations are related by

$$v^{k+1} = \mathscr{C}_m(u_0, v^k),$$

where $\mathscr{C}_m$ denotes one step of the CR-algorithm and m is related to the order of the time derivative that is set to zero in Eq. (13). In general this equation is nonlinear and the fixed point can be found by a Newton iteration. This means solving

$$g_m(u_0, v) := v - \mathscr{C}_m(u_0, v) = 0,$$

for a macroscopic value $u_0 = \rho_0$. Newton's method gives

$$v^{k+1} = v^k - \Big(J(u_0, v^k)\Big)^{-1} g_m(u_0, v^k), \tag{14}$$

where J denotes the Jacobian of g_m. We use the combination of the CR-algorithm with Newton's method as a lifting operator for the hybrid model with different orders of extrapolation. The Jacobian J is approximated by

$$Je_i \approx \frac{g_m(u_0, v + \varepsilon e_i) - g_m(u_0, v)}{\varepsilon},$$

with e_i the unit vector, $i = 1, \ldots, 2n$ and $\varepsilon = 1.0e^{-7}$ [26].

A drawback of using Newton's method is that it requires the construction of the $2n \times 2n$ Jacobian matrix. For the lifting of the initial density this can be quite expensive since the unknown vector v is large. Many evaluations of the underlying Boltzmann model are then required to construct the Jacobian matrix. Similarly, matrix-free methods like GMRES still require many matrix-vector products since the spectrum is unfavorable for fast convergence [25]. Since these methods require many evaluations of the underlying LBM to initialize the missing variables, it will still remain computationally expensive in practice to use these as lifting operators.

In hybrid models, the Jacobian matrices are significantly smaller than $2n \times 2n$ since we only lift in the ghost points of the LBM region. This requires only local updates which results in a smaller Jacobian.

Indeed, the lifting operator only needs to construct the missing distribution functions in x_p and x_n (Figure 1).

Consider ghost point x_p. Suppose that the CR-algorithm with constant extrapolation in time is used to find these missing distribution functions. Since, in each iteration, only one step of the LBM is executed (constant extrapolation), the domain of dependence dictates that we only need a guess for $f_i(x,t)$, $i \in \{-1,0,1\}$ in the points between $x = x_{p-1}$ and $x = x_{p+1}$ (Figure 1). If we know how many iterations k are needed to reach the required tolerance value, the number of points we need to consider at the left and right of x_p can be determined. And if we take one extra point at both sides, we can avoid the influence of the boundaries of the smaller vector. It is thus possible to consider any boundary conditions for the smaller vector.

This means that only $k+1$ points at the left and $k+1$ points at the right of x_p need to be considered. Because we are working with a smaller vector, a smaller Jacobian is required that, in addition, can decrease in size as the iteration progresses.

3.2.1 Constrained Runs with constant extrapolation

We use the Constrained Runs algorithm with constant extrapolation in time. The function we solve with Newton's method is then

$$0 = g_0(u_0, \phi, \xi) = \begin{cases} \phi - \phi_1 \\ \xi - \xi_1, \end{cases} \tag{15}$$

where ϕ_1 and ξ_1 are the moments corresponding to the distribution functions after one step of LBM. The initial condition of the LBM is obtained by extracting the distribution functions from Eq. (5), starting from an initial guess for the moments ϕ and ξ.

This is used to reduce both the initial error and the error in the ghost points of the LBM. The resulting error is shown in Figure 4 (top) which shows $|\rho_{\text{hybrid}}(x,t) - \rho_{\text{LBM}}(x,t)|$. The error is of order 10^{-4} with the same declarations as in Figure 2. The initial distribution functions are based on the Constrained Runs algorithm combined with Newton's method for a constant extrapolation in time.

3.2.2 Constrained Runs with linear extrapolation

When linear extrapolation in time is used the function that we solve with Newton's method is

$$0 = g_1(u_0, \phi, \xi) = \begin{cases} \phi - 2\phi_1 + \phi_2 \\ \xi - 2\xi_1 + \xi_2, \end{cases} \tag{16}$$

where ϕ_1, ξ_1 and ϕ_2, ξ_2 are the moments corresponding to the distribution functions after one and two steps, respectively, of LBM. The initial condition of the LBM is obtained by extracting the distribution functions from Eq. (5), starting from an initial guess for the moments ϕ and ξ.

The results for dealing with the initial error and the error in the ghost points of the LBM by using a linear extrapolation in the CR-algorithm in combination with the method of Newton are given in Figure 4 (second figure). It shows a better absolute difference $|\rho_{\text{hybrid}} - \rho_{\text{LBM}}|$ than the difference based on a constant extrapolation.

3.2.3 Constrained Runs with higher order extrapolation

In the bottom figures of Figure 4, higher order extrapolations are considered. The error that remains in these figures is the modeling error. This error is not caused by the coupling of both models but by the difference in both models. We compare a PDE model on $[a,l[$ with a full LBM. So it is obvious that a certain error is created on the densities.

4 Smooth initialization

An alternative way to arrive at an implicit expression for the unknown distribution functions is to interpret the lifting as a missing data problem. In such a problem a variable $w \in W$ is split into $w = (u,v)$, where $u \in U$ contains the explicitly given data and $v \in V$ the missing data. These variables are considered in normed vector spaces U, V and W, in which W is the direct sum of the normed spaces U and V. This is discussed for a lattice Boltzmann scheme in [14] with two distribution functions f_{-1} and f_1.

We focus on an extension with three distribution functions $\{f_1, f_0, f_{-1}\}$. This will correspond for our LBM to $w = (\rho^T, \phi^T, \xi^T)^T \in W = \mathbb{R}^{3n}$ with $u = \rho \in U = \mathbb{R}^n$ and

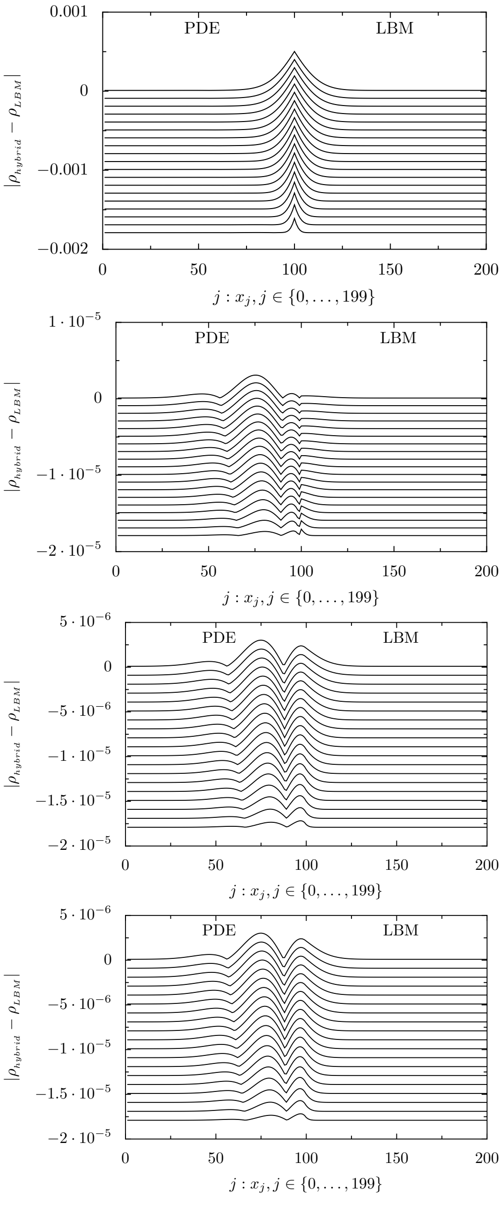

Figure 4: $|\rho_{\text{hybrid}} - \rho_{\text{LBM}}|$ after 200 time steps where the lifting operator based on the CR-algorithm is used in combination with the method of Newton. We show results for constant (top), linear, quadratic and cubic (bottom) backward extrapolation, respectively. The difference is also shown at earlier time slots, but shifted down for clarity. The domain that is considered, is shown in Figure 1. Model parameters are $\Delta t = 0.001$, $\Delta x = 0.05$, $D = 1$ and $\rho(x,0) = \exp\big(-(x - \frac{L}{2})^2\big)$.

$v = (\phi^T, \xi^T)^T \in V = \mathbb{R}^{2n}$. To remain general in notation $E : W \to W$ is considered to be a numerical scheme. This will be the LBM in our case. To define the numerical scheme onto the different components of variable w, some other operators are defined, namely the identity operator $I : W \to W$ and the canonical projection $P : W \to V$ onto the second component (v).

One assumes that the initial values of the missing data should be set so that the evolution scheme, namely the lattice Boltzmann scheme, starts smoothly. Intuitively this corresponds to the assumption that the dynamics are attracted to a slow manifold. If this is not the case, rapid changes could still be possible. This makes the idea of this method quite similar to that of the CR-algorithm. To put the idea of smooth initialization into practice, an m-regularity condition is defined on output w in [14]. The rule to determine the value $v(0)$ is then given in [14] by: *The initial value $w(0)$ of some m-regular output w must coincide with the polynomial backward extrapolation of its values pertaining to the iterations $t_1 = \Delta t, t_2 = 2\Delta t, \ldots, t_{m+1} = (m+1)\Delta t$.* This condition can be written as $P(I - E)w(0) = 0$ for constant extrapolation and $P(I - 2E + E^2)w(0) = 0$ for linear extrapolation.

In the next subsections constant extrapolation will be considered. As can be seen, many analytical expressions need to be written out to use the smooth initialization technique. That is why linear extrapolation has not been considered in this paper.

4.1 Constant extrapolation - full update vector

In this section, we show that for a given density ρ on the full domain the missing moments can be found with the help of assumptions of smooth initialization. We first illustrate this for the full domain and then, in Sect. 4.3, modify this for a local update.

Let us consider a LBM with $f_1(t)$, $f_0(t)$, $f_{-1}(t) \in \mathbb{R}^n$ and, equivalently $\rho(t)$, $\phi(t)$, $\xi(t) \in \mathbb{R}^n$. If a full update — an update in all spatial grid points — of the missing data $(\phi^T, \xi^T)^T$ is done, explicit formulas can be achieved for the technique of smooth initialization. Using the fact that by definition (4) and introducing L and R, left- and right-shift operators, where the left-shift operator moves the elements of the vector to the left over one grid distance and the right-shift operator to the right over one grid distance, the LBM is described by

$$
\begin{aligned}
f_1(t+\Delta t) &= R[(1-\omega)f_1(t)+\omega f_1^{eq}(t)],\\
f_0(t+\Delta t) &= (1-\omega)f_0(t)+\omega f_0^{eq}(t),\\
f_{-1}(t+\Delta t) &= L[(1-\omega)f_{-1}(t)+\omega f_{-1}^{eq}(t)].
\end{aligned}
\tag{17}
$$

We can obtain formulas for $\phi(\Delta t)$ and $\xi(\Delta t)$.

$$
\begin{aligned}
\phi(\Delta t) \quad = \quad & (\alpha_L L+\alpha_R R)\phi(0)+(\beta_L L+\beta_R R)\xi(0)\\
& +(\gamma_L L+\gamma_R R)\rho(0),
\end{aligned}
$$

with

$$
\begin{aligned}
&\alpha_L = \frac{1-\omega}{2}, \quad \alpha_R = \frac{1-\omega}{2},\\
&\beta_L = -(1-\omega), \quad \beta_R = 1-\omega,\\
&\gamma_L = -\frac{\omega}{3}, \quad \gamma_R = \frac{\omega}{3},
\end{aligned}
\tag{18}
$$

for the momentum and

$$
\begin{aligned}
\xi(\Delta t) \quad = \quad & (\alpha\xi_L L+\alpha\xi_R R)\phi(0)+(\beta\xi_L L+\beta\xi_R R)\xi(0)\\
& +(\gamma\xi_L L+\gamma\xi_R R)\rho(0),
\end{aligned}
$$

with

$$
\begin{aligned}
&\alpha\xi_L = -\frac{1-\omega}{4}, \quad \alpha\xi_R = \frac{1-\omega}{4},\\
&\beta\xi_L = \frac{1-\omega}{2}, \quad \beta\xi_R = \frac{1-\omega}{2},\\
&\gamma\xi_L = \frac{\omega}{6}, \quad \gamma\xi_R = \frac{\omega}{6},
\end{aligned}
\tag{19}
$$

for the energy. In matrix form, this becomes

$$
\begin{aligned}
\begin{pmatrix}\phi\\ \xi\end{pmatrix}(\Delta t) \quad &= \quad \begin{pmatrix}\alpha_L L+\alpha_R R & \beta_L L+\beta_R R\\ \alpha\xi_L L+\alpha\xi_R R & \beta\xi_L L+\beta\xi_R R\end{pmatrix}\begin{pmatrix}\phi\\ \xi\end{pmatrix}(0)\\
&\quad + \begin{pmatrix}\gamma_L L+\gamma_R R\\ \gamma\xi_L L+\gamma\xi_R R\end{pmatrix}\rho(0),\\
&= \quad A\begin{pmatrix}\phi\\ \xi\end{pmatrix}(0)+B\rho(0),
\end{aligned}
\tag{20}
$$

where $A\in\mathbb{R}^{2n\times 2n}$ and $B\in\mathbb{R}^{2n\times n}$. Using the assumption of constant extrapolation $(\phi^T,\xi^T)^T(\Delta t) = (\phi^T,\xi^T)^T(0)$, we obtain the missing moments as

$$
\begin{pmatrix}\phi\\ \xi\end{pmatrix}(0) = (I-A)^{-1}B\rho(0),
\tag{21}
$$

if $I-A$ is invertible with $I\in\mathbb{R}^{2n\times 2n}$ the identity matrix. Numerical results show that the spectral radius of A is smaller than one so $0\notin\sigma(I-A) = 1-\sigma(A)$ with $\sigma(A)$ the spectrum of A.

4.2 Link with Constrained Runs algorithm

Consider formula (20). When a direct iteration

$$
\begin{pmatrix}\phi\\ \xi\end{pmatrix}^{k+1}(0) = A\begin{pmatrix}\phi\\ \xi\end{pmatrix}^{k}(0)+B\rho(0),
\tag{22}
$$

is used, the CR-algorithm with constant extrapolation is obtained. A consistency analysis can be completed to show a first order correspondence with the Chapman-Enskog expressions.

If we plot $|\rho_{\text{hybrid}}-\rho_{\text{LBM}}|$, Figure 4 (top) is achieved again. This makes sense since both methods are based on a polynomial backward constant extrapolation.

4.3 Constant extrapolation - local update

In Sect. 4.1 an update of the full vector $(\phi^T,\xi^T)^T$ is used. As with the CR-algorithm in Sect. 3.2, we want to look for a local update formulation. In the ghost points x_0 and x_p a one-to-many map is needed to obtain the distribution functions (Figure 1). The formulas from Sect. 4.1 can be used but these update every spatial grid point, which is not necessary when updating only the ghost points. This leads to unnecessary work, which we want to avoid by focusing on local updates. Consider an infinite domain around x_0 with a LBM domain at the left and a PDE domain at the right. This is what happens at point x_0 in Figure 1 because of the periodic boundary conditions.

The analysis that follows is specific for point x_0 but can easily be adapted for point x_p by substituting the indices of the spatial grid points. The grid points of the PDE domain are denoted by indices $i = 0,1,2,\ldots$. The LBM domain consists of indices $i = \ldots,-2,-1$. The space index 0 is noted as $(.)_0$. To know the missing data in point 0, this data is expressed as a function of the point at the right of index 0 in the PDE domain (index 1) and points of the LBM domain, since we are considering constant extrapolation. This means that point 0 can be expressed in function of point -1 and point 1.

$$\begin{pmatrix}\phi\\ \xi\end{pmatrix}_0 = C\begin{pmatrix}\rho\\ \phi\\ \xi\end{pmatrix}_1 + h(LBM), \tag{23}$$

with $C \in \mathbb{R}^{2\times 3}$ a matrix that expresses the dependence of the backward extrapolation on ρ, ϕ and ξ in point 1. $h(LBM)$ contains the points that only require information of the LBM domain. We do not need to know the function h in detail because there is no missing data in the LBM domain.

The result depends on the known variables in the LBM domain and on ρ, ϕ and ξ in point 1, which poses a new problem in point 1, similarly as in point 0, since ϕ and ξ in point 1 are also missing. This means that we need a choice for the missing data in point 1 that is required to find $(\phi^T, \xi^T)^T$ in point 0.

Equation (23) can be rewritten as

$$\begin{pmatrix}\phi\\ \xi\end{pmatrix}_0 = D\rho_{-1:1} + E\begin{pmatrix}\phi\\ \xi\end{pmatrix}_1 + F\begin{pmatrix}\phi\\ \xi\end{pmatrix}_{-1}, \tag{24}$$

with $D, E, F \in \mathbb{R}^{2\times 2}$, where the dependence of the LBM domain is written out explicitly. $\rho_{-1:1}$ stands for the values of ρ in points -1 and 1. All ϕ and ξ with a negative index are in principle known because they can be derived from the distribution functions f_i, $i \in \{-1, 0, 1\}$ in the LBM domain.

By using the same equation as in (24) the missing data problem can be formulated in point 1.

$$\begin{pmatrix}\phi\\ \xi\end{pmatrix}_1 = D\rho_{0:2} + E\begin{pmatrix}\phi\\ \xi\end{pmatrix}_2 + F\begin{pmatrix}\phi\\ \xi\end{pmatrix}_0. \tag{25}$$

Combining equations (24) and (25) gives

$$\begin{aligned}\begin{pmatrix}\phi\\ \xi\end{pmatrix}_0 &= D\rho_{-1:1} + ED\rho_{0:2} + EE\begin{pmatrix}\phi\\ \xi\end{pmatrix}_2\\ &\quad + EF\begin{pmatrix}\phi\\ \xi\end{pmatrix}_0 + F\begin{pmatrix}\phi\\ \xi\end{pmatrix}_{-1},\end{aligned}$$

which, on its turn, creates a missing data problem in point 2. By reusing (24), this leads to

$$\begin{aligned}\begin{pmatrix}\phi\\ \xi\end{pmatrix}_0 &= D\rho_{-1:1} + ED\rho_{0:2} + E^2D\rho_{1:3}\\ &\quad + E^3\begin{pmatrix}\phi\\ \xi\end{pmatrix}_3 + E^2F\begin{pmatrix}\phi\\ \xi\end{pmatrix}_1\\ &\quad + EF\begin{pmatrix}\phi\\ \xi\end{pmatrix}_0 + F\begin{pmatrix}\phi\\ \xi\end{pmatrix}_{-1},\end{aligned} \tag{26}$$

where the dependence of the unknown moments ϕ and ξ in point 1 re-enters. However, the matrices E^2F and EF are, as we will show below, zero and the missing ϕ and ξ can be expressed as the densities in the coupling region, ϕ and ξ in point 3 and some given data of the LBM domain.

Indeed, Eq. (20) only contains the shift operators L and R with prefactors and can be rewritten for a point as

$$\begin{aligned}\begin{pmatrix}\phi\\ \xi\end{pmatrix}_0 &= \begin{pmatrix}\gamma_L & \alpha_L & \beta_L\\ \gamma\xi_L & \alpha\xi_L & \beta\xi_L\end{pmatrix}\begin{pmatrix}\rho\\ \phi\\ \xi\end{pmatrix}_1\\ &\quad + \begin{pmatrix}\gamma_R & \alpha_R & \beta_R\\ \gamma\xi_R & \alpha\xi_R & \beta\xi_R\end{pmatrix}\begin{pmatrix}\rho\\ \phi\\ \xi\end{pmatrix}_{-1}.\end{aligned}$$

With the notations D, E and F this results in

$$\begin{pmatrix}\phi\\ \xi\end{pmatrix}_0 = D\begin{pmatrix}\rho_1\\ \rho_{-1}\end{pmatrix} + E\begin{pmatrix}\phi\\ \xi\end{pmatrix}_1 + F\begin{pmatrix}\phi\\ \xi\end{pmatrix}_{-1},$$

where

$$\begin{aligned} D &= \begin{pmatrix}\gamma_L & \gamma_R\\ \gamma\xi_L & \gamma\xi_R\end{pmatrix}, \quad E = \begin{pmatrix}\alpha_L & \beta_L\\ \alpha\xi_L & \beta\xi_L\end{pmatrix},\\ F &= \begin{pmatrix}\alpha_R & \beta_R\\ \alpha\xi_R & \beta\xi_R\end{pmatrix},\end{aligned}$$

and because E and F contain the same diagonal elements and opposite off diagonal elements (see Eqs. (18) and (19)), and $\alpha_L = \beta\xi_L$ and $\alpha_L^2 = \beta_L\alpha\xi_L$ this results in

$$EF = \begin{pmatrix}0 & 0\\ 0 & 0\end{pmatrix}, \quad E^2F = \begin{pmatrix}0 & 0\\ 0 & 0\end{pmatrix}, \quad \ldots. \tag{27}$$

Using these properties, the missing data $(\phi^T, \xi^T)_0^T$ in the first PDE point can be expressed as ρ, the variable of the PDE, in the points 0 up to 3, the LBM variables in point -1 and, finally, $(\phi^T, \xi^T)^T$ in point 3. However,

$(\phi^T, \xi^T)^T_3$ is not available from the PDE which also results in a missing variable. Using the same relations repeatedly, we could rewrite this into the densities ρ in the grid points in the ghost region and a missing data problem that moves away from the interface between the LBM and PDE region. However, note that

$$\lim_{k\to\infty} E^k = 0, \tag{28}$$

since E is defined as

$$E = (1-\omega)\begin{pmatrix} \frac{1}{2} & -1 \\ -\frac{1}{4} & \frac{1}{2} \end{pmatrix} = (1-\omega)G,$$

where $G \in \mathbb{R}^{2\times 2}$ has eigenvalues 0 and 1.

If the lifting operator requires a given tolerance of accuracy, the number of iterations k can be determined. If we claim $|1-\omega|^k <$ tol, equation (28) will be fulfilled for k determined by the tolerance value tol.

By using this shortened update in the ghost points of the LBM, again Figure 4 (top) is obtained. The composition of this figure is based on the claim of Eq. (28) with $k = 10$. The value of k is obtained by

$$|1-\omega|^k < \text{tol} = 10^{-10}, \tag{29}$$

with $\omega = 0.9091$ for the model problem, which makes $k > 9.6$ sufficient.

5 Discussion, conclusion and outlook

In this article we have studied a hybrid domain that spatially couples a diffusion partial differential equation (PDE) with a lattice Boltzmann model (LBM). This creates a missing data problem at the interfaces between the models. Indeed, the PDE model has too few variables to provide the LBM with the correct boundary conditions. Different methods are studied in this article to fill in these missing data. In particular, we have focused on a simple LBM and PDE model discretized with equal grid and time steps such that the error created by the coupling can be highlighted.

We have analyzed different solution methods for the missing data problem with various orders of accuracy. First, the Constrained Runs algorithm is considered for various orders of accuracy. We have found that the coupling error decreases with the order of the scheme and that the error is smaller than the modeling error already with the linear extrapolation in time. Higher order versions decrease the error slightly further. This confirms the result of [12] where the error was analyzed for different orders of the analytical Chapman-Enskog expansion. In this paper, we used Newton's method to find the fixed point of higher order Constrained Runs. The errors are estimated by comparing the density of the hybrid model with the density of a full LBM simulation. In a similar way we have analyzed the accuracy when the missing data problem is formulated as a smooth initialization problem. In this work constant backward extrapolation is considered in this context.

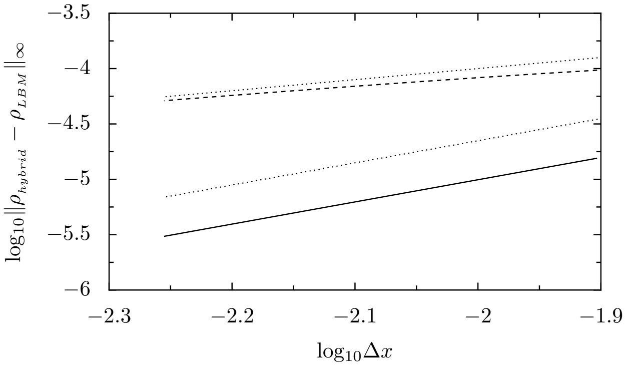

Figure 5: Log-log plot of the error $\|\rho_{\text{hybrid}} - \rho_{\text{LBM}}\|_\infty$ with respect to $\Delta x = L/n$, the grid distance. The dashed line shows the results of constant extrapolation. The linear extrapolation is second order and is shown with a solid line. The black dotted lines are reference lines with slopes -1 and -2. The parameters that are considered: $L = 1$, $n = 80 : 10 : 180$, $\omega = 0.90$, $D = 1$, $\Delta t = 2\Delta x^2(\frac{1}{\omega} - \frac{1}{2})/(3D)$, $t = 0.03$ and $\rho(x,0) = \exp\bigl(-(x - \frac{L}{2})^2\bigr)$.

To compare the accuracy of the different methods in one figure, the log-log plot in Figure 5 shows the results as a function of grid distance Δx. Since ω is considered fixed, Δt changes according to (6). $\|\rho_{\text{hybrid}} - \rho_{\text{LBM}}\|_\infty$ is compared for the different methods — CR-algorithm with constant extrapolation, CR-algorithm with linear extrapolation and smooth initialization with constant extrapolation — at time $t = 0.03$. However, the computational cost of the lifting methods grows significantly with the order of accuracy. Especially the Constrained Runs algorithm needs many LBM steps to find the missing distribution functions in the ghost points of a hybrid model. Since the Constrained Runs algorithm is build around the LBM time evolution it is useful to write the computational cost in terms of how many extra LBM grid points need to be updated during the lifting.

The fixed point of the CR-algorithm is found with Newton's method. In our numerical experiments about three Newton iterations are necessary to converge. This means that three Jacobian matrices need to be constructed with varying sizes depending on the iteration number and the order of the CR-algorithm. If k denotes the number of Newton iterations, the size of the largest Jacobian is $2\times(2\times(k+1)+1)$ by $2\times(2\times(k+1)+1)$ for a constant backward extrapolation. Indeed, $k+1$ points are needed at both sides of the ghost point. Since both ϕ and ξ are unknown, the length of the unknown vector is $2\times(2\times(k+1)+1)$. Note that the following computations and the results in Table 1 are based on this largest Jacobian and not on the possible decrease in size as the iteration progresses.

To construct this Jacobian matrix we need to call $g_0(u_0, v+\varepsilon e_i)$ about $2\times(2\times(k+1)+1)$ times per Jacobian and $g_0(u_0, v)$ once per Jacobian. Each call to g_0 requires one LBM step.

However, depending on the order of g_m the function contains multiple LBM steps and the size of the Jacobian grows. For linear extrapolation, the evaluation of g_1 requires two LBM steps. In addition $2k+1$ grid points are required at both sides of the ghost point to avoid contamination from the boundaries. The largest Jacobian then has a size $2\times(2\times(2k+1)+1)$ by $2\times(2\times(2k+1)+1)$. Similar results can be computed for quadratic and cubic extrapolation in time.

A summary of the computational cost in terms of LBM steps is listed in Table 1. These results are just LBM steps needed to lift the density in one ghost point. This needs to be repeated each time step, although some efficiency can be gained by reusing information from previous time steps.

Bear in mind that to start the algorithm, the initial distribution functions for the LBM domain need to be determined in a similar way. This requires a Newton update of the entire domain (400 grid points (200 points for both ϕ and ξ)). Still, it requires 3 iterations but $g_0(u_0, v+\varepsilon e_i)$ is called 400 times per Jacobian and $g_0(u_0, v)$ one time per Jacobian. This results in $3\times 401 = 1203$ LBM steps (for constant extrapolation) just to initialize the lattice Boltzmann model.

We conclude that the higher order version of Constrained Runs, indeed, finds accurate distribution functions for a given density. However, if the aim is to reduce the computational cost of the LBM domain by replacing it with a PDE model we do not see many advantages. The lifting methods call many LBM steps and this additional computational cost does not outweigh the reduction of the LBM domain. According to the updates of the full domain, the local updates already reduce the computational cost significantly.

We expect similar results in two dimensions where a line of points needs to be lifted at the interface between a PDE and LBM domain. Again the Jacobian matrices will significantly increase in size and many LBM evaluations will be required.

In upcoming work we will analyze a numerical lifting method that combines the ideas of Constrained Runs, smooth initialization and the Chapman-Enskog expansion. Instead of using Constrained Runs to find for each grid point the missing moments ϕ and ξ of $f(x, v, t)$, we use Constrained Runs to find the unknown coefficients of the Chapman-Enskog expansion. This has several advantages. First, it significantly reduces the number of unknowns in the lifting steps: we only need to find the coefficients rather than the full state $f(x, v, t)$. And secondly, it can be done off-line before the calculations. Indeed, once the coefficients are found they can be reused every time step to realize the lifting.

An extension to more general discretizations of (2) is one possibility for future research. We can take a look at higher order methods of the Constrained Runs iterations and different implicit methods to determine the fixed point. Techniques of domain decomposition can be combined with a one-to-many map (to lift $\rho(x, t)$ to $f(x, v, t)$). Another idea is to take a look at mesh refinement. Some numerical analysis of the convergence speed for different realistic situations should give a better view on the used methods. The results of each of these research topics need to be tested and evaluated on realistic problems coming

Table 1: Total number of LBM time steps that are needed throughout a hybrid model at each interface based on local updates performed with the Constrained Runs algorithm in combination with Newton's method. The values for the CR-algorithm are listed per ghost point and per time step. The method based on local updates needs less LBM steps than an update of the entire domain (1203 LBM steps for constant extrapolation in every time step).

Extrapolation CR-algorithm	Number of iterations	Number of calls function g per Jacobian matrix (based on 3 iterations)	Number of LBM steps per call function g	Total number of LBM steps per ghost point per time step	Number of updated grid points in one LBM step
Constant	3 / (2)	19	1	57	18
Linear	3 / (2)	31	2	186	30
Quadratic	3 / (2)	43	3	387	42
Cubic	3 / (2)	55	4	660	54

from plasma physics.

Acknowledgments. This work is supported by research project *Hybrid macroscopic and microscopic modelling of laser evaporation and expansion*, G.017008N, funded by 'Fonds Wetenschappelijk Onderzoek' together with an 'ID/BOF-beurs' 24140 of the University of Antwerp.

References

[1] W. Hundsdorfer, J.G. Verwer, *Numerical solution of time-dependent advection-diffusion-reaction equations.* Springer, Berlin (2003).

[2] T. Davis, *Direct methods for sparse linear systems,* SIAM, Philadelphia, PA (2006).

[3] Y. Saad, *Iterative methods for sparse linear systems.* SIAM, Philadelphia, PA (2003).

[4] U. Trottenberg, C. Oosterlee, A. Schüller, *Multigrid.* Academic Press, New York (2001).

[5] N. Bellomo, M. Delitala, V. Coscia, *On the mathematical theory of vehicular traffic flow, I.* Fluid dynamics and kinetic modelling, Math. Models Methods Appl. Sci. 12, 1801-1843 (2002).

[6] R. Erban, H.G. Othmer, *From individual to collective behavior in bacterial chemotaxis.* SIAM J. Appl. Math. 65, 361 (2004).

[7] J.A. Bittencourt, *Fundamentals of plasma physics.* Springer, New York (2004).

[8] E.G. Flekkoy, G. Wagner, J. Feder, *Hybrid model for combined particle and continuum dynamics*. Europhys. Lett. 52 (3), 271-276 (2000).

[9] M.L. Parks, R.B. Lehoucq, *Atomistic-to-Continuum Coupling*. SIAM NEWS (2006).

[10] A.L. Garcia, J.B. Bell, W.Y. Crutchfield, B.J. Alder, *Adaptive mesh and algorithm refinement using direct simulation Monte Carlo*. J. Comput. Phys. 154, 134-155 (1999).

[11] A. Dupuis, E.M. Kotsalis, P. Koumoutsakos, *Coupling lattice Boltzmann and molecular dynamics models for dense fluids*. Phys. Rev. E 75, 046704 (2007).

[12] P. Van Leemput, C. Vandekerckhove, W. Vanroose, D. Roose, *Accuracy of hybrid lattice Boltzmann/finite difference schemes for reaction-diffusion systems*. Multiscale Model. Sim. 6, 838-857 (2007).

[13] G. Samaey, C. Vandekerckhove, W. Vanroose, *A multilevel algorithm to compute steady states of lattice Boltzmann models.* Coping with Complexity: Model reduction and data analysis. Springer, Berlin, 151-167 (2011).

[14] P. Van Leemput, M. Rheinländer, M. Junk, *Smooth initialization of lattice Boltzmann schemes.* Comput. Math. Appl. 58, 867-882 (2009).

[15] C. Vandekerckhove, I. Kevrekidis, D. Roose, *An efficient Newton-Krylov implementation of the Constrained Runs scheme for initializing on a slow manifold.* J. Sci. Comput. 39, 167-188 (2009).

[16] C.W. Gear, T.J. Kaper, I.G. Kevrekidis, A. Zagaris, *Projecting to a slow manifold: singularly perturbed systems and legacy codes.* SIAM J. Appl. Dyn. Syst. 4, 711-732 (2005).

[17] P. Van Leemput, W. Vanroose, D. Roose, *Mesoscale analysis of the equation-free Constrained Runs initialization scheme.* Multiscale Model Simul. 6, 1234-1255 (2007).

[18] A. Zagaris, C.W. Gear, T.J. Kaper, I.G. Kevrekidis, *Analysis of the accuracy and convergence of equation-free projection to a slow manifold.* ESIAM Math. Model. Numer. Anal. 43, 757-784 (2009).

[19] D.A. Wolf - Gladrow, *Lattice-Gas Cellular Automata and Lattice Boltzmann Models.* Springer, Berlin (2000).

[20] S. Succi, *The Lattice Boltzmann Equation for Fluid Dynamics and Beyond.* Oxford University Press, Oxford (2001).

[21] M. Junk, A. Klar, L. Luo, *Asymptotic analysis of the lattice Boltzmann equation.* J. Comput. Phys. 210, 676-704 (2005).

[22] P. Van Leemput, *Multiscale and Equation-Free Computing for Lattice Boltzmann Models.* PhD thesis, K.U. Leuven (2007).

[23] C. Cercignani, *The Boltzmann Equation and its Applications.* Springer, Berlin (1988).

[24] M. Rheinländer, *A consistent grid coupling method for lattice-Boltzmann schemes.* J. Stat. Phys. 121, 49-74 (2005).

[25] C. Vandekerckhove, *Macroscopic Simulation of Multiscale Systems within the Equation-Free Framework.* PhD thesis, K.U. Leuven (2008).

[26] C.T. Kelley, *Iterative Methods for Linear and Nonlinear Equations.* SIAM, Philadelphia, PA (1995).