# An approach for both the computation of coarse-scale steady state solutions and initialization on a slow manifold

Christophe Vandekerckhove[1,a], Benjamin Sonday[2,b], Alexei Makeev[c], Dirk Roose[a], Ioannis G. Kevrekidis[b,d]

[a]*Department of Computer Science, K.U. Leuven, B-3001 Heverlee, Belgium*
[b]*Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ 08544*
[c]*Moscow State University, Faculty of Computational Mathematics and Cybernetics (BMK), Moscow, 119899, Russia*
[d]*Department of Chemical Engineering, Princeton University, Princeton, NJ 08544, USA*

## Abstract

We present a simple technique for the computation of coarse-scale steady states of dynamical systems with time scale separation in the form of a "wrapper" around a fine-scale simulator. We discuss how this approach alleviates certain problems encountered by comparable existing approaches, and illustrate its use by computing coarse-scale steady states of a lattice Boltzmann fine scale code. Interestingly, in the same context of multiple time scale problems, the approach can be slightly modified to provide initial conditions *on the slow manifold* with prescribed coarse-scale observables. The approach is based on appropriately designed short bursts of the fine-scale simulator whose results are used to track changes in the coarse variables of interest, a core component of the equation-free framework.

*Keywords:* slow manifold, coarse-graining, time-stepper, initialization

## 1. Introduction

For many problems in science and engineering, the best available model is given at a fine-scale level, while we would like to analyze its behavior at a much coarser-scale "system level". To bridge the gap between the scale of the available model and the scale of interest, one typically attempts to derive a reduced model in terms of an appropriate set of variables (the "coarse observables"). In many cases, the derivation of such a reduced model hinges on the existence of a low-dimensional, attracting, invariant slow manifold, which can be parametrized in terms of these observables. In fine-scale simulations, all initial conditions are quickly attracted towards this slow manifold, on which the reduced dynamics subsequently evolve. In other words, the remaining fine-scale model variables quickly become functionals of (become "slaved to") the observables, and the fine-scale model state can be accurately described in terms of the observables only.

---

[1]Current address: Markt 62, 9800 Deinze, Belgium
[2]Corresponding author: bsonday@math.princeton.edu

Although it should in principle always be possible to derive a reduced dynamical model for a system possessing a slow manifold (see, e.g., Segel and Slemrod (1989) and Ramshaw (1980)), one may fail to do so, for instance because the closures required in its construction are not accurate or even unavailable. For this class of problems, Kevrekidis and coworkers proposed the equation-free framework (Kevrekidis et al. (2003)), which allows one to perform coarse-scale computations based on appropriately initialized short fine-scale simulations. (The term "equation-free" emphasizes that the derivation of reduced equations is circumvented.) In this paper, we propose and study two related equation-free algorithms that are designed to (1) compute stable or unstable coarse-scale steady state solutions, or to (2) systematically initialize a fine-scale model with prescribed values of the coarse observables (see, e.g., prior work in Lam and Goussis (1994); Maas and Pope (1992); Jolly et al. (1990); Gear and Kevrekidis (2005); Gear et al. (2005); Vandekerckhove et al. (2009); Curry et al. (2002); Lorenz (1986) and some further mathematical analysis in Kreiss and Lorenz (1994); Zagaris et al. (2004)). As we will see below, the gain in efficiency obtained with these algorithms stems from the fact that the bulk of the computations are only performed in the space of the observables, which is typically low-dimensional compared to the full fine-scale variables space.

The outline of this paper is as follows: In Section 2, we address the computation of coarse-scale steady state solutions. We succinctly illustrate certain problems encountered by existing, time-stepper based algorithms, and propose a modification that alleviates these problems. Section 3 then discusses a variation of the latter approach that helps initialize the fine-scale simulator *on the slow manifold*. Section 4 uses a lattice Boltzmann fine scale simulator to demonstrate the application of both algorithms; the error characteristics of the algorithm is presented in Section 5 and we conclude with a brief discussion.

## 2. Computing coarse-scale steady state solutions

In this section, we outline two existing methods to compute coarse-scale steady state solutions, as well as the modification leading to the proposed, third method, with the help of a simple model problem. Specifically, we consider the two-dimensional linear time integrator

$$
\begin{bmatrix} x \\ y \end{bmatrix}_{n+1} = \begin{bmatrix} \cos(\alpha) & \cos(\beta) \\ \sin(\alpha) & \sin(\beta) \end{bmatrix} \begin{bmatrix} 0.999 & 0 \\ 0 & 0.1 \end{bmatrix} \begin{bmatrix} \cos(\alpha) & \cos(\beta) \\ \sin(\alpha) & \sin(\beta) \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \end{bmatrix}_n \tag{1}
$$

as our fine-scale simulator with $x$ being the "coarse observable". This arises, for instance, as the explicit Euler integrator for the system

$$
\frac{d}{dt} \begin{bmatrix} x \\ y \end{bmatrix} = M \begin{bmatrix} x \\ y \end{bmatrix} \tag{2}
$$

with time step $\Delta t$, where

$$
\Delta t \, M = \begin{bmatrix} \cos(\alpha) & \cos(\beta) \\ \sin(\alpha) & \sin(\beta) \end{bmatrix} \begin{bmatrix} -0.001 & 0 \\ 0 & -0.9 \end{bmatrix} \begin{bmatrix} \cos(\alpha) & \cos(\beta) \\ \sin(\alpha) & \sin(\beta) \end{bmatrix}^{-1}. \tag{3}
$$

The slow manifold is then $y = \tan(\alpha)x$; note that this manifold can be parametrized by (is the graph of a function over) our chosen observable as long as $\alpha \neq \pm\pi/2$. Any fine-scale initial condition $(x, y)$ is quickly attracted to this slow manifold along a trajectory that (at least far enough away from the slow manifold) approximates a line with slope $\tan(\beta)$.

## 2.1. Description of the methods

The obvious fine-scale steady state solution of (1) is $(x^*, y^*) = (0, 0)$, and this is a stable steady state. In this section, we initially illustrate two existing methods whose purpose is to approximate the *coarse-scale* steady state solution $x^* = 0$. These two methods are based directly on the concept of the *coarse time-stepper* (Theodoropoulos et al. (2000); Kevrekidis et al. (2003)). They approximate the value of $x^*$ by solving

$$\Phi(x, \tau) - x = 0, \tag{4}$$

where $\Phi(x, \tau)$ denotes a coarse time step over time $\tau$ with initial condition $x$. Each such coarse time step (of time $\tau$) consists of the following three substeps:

1. *lifting*, in which an appropriate fine-scale state is constructed according to the value of the observable $x$.
2. *simulation*, in which the fine-scale state is evolved over time $\tau$, with $\tau$ large enough to allow $y$ to get slaved to $x$, but small compared to the coarse time scales.
3. *restriction*, in which the value of the observable is extracted from the resulting fine-scale state.

If the value of $x$ *does not change* substantially during the fast transients towards the slow manifold, it may suffice to choose arbitrary (say, consistently the same) values for the remaining fine-scale variables (in this case, just $y$) in the lifting step. This type of lifting will from now on be called *arbitrary lifting*, and the resulting method for approximating the coarse-scale steady state solution will be called METHOD 1.

If the value of $x$ *does change* substantially during the fast transients towards the slow manifold, a more accurate initialization (i.e., an initialization closer to the slow manifold) should be used. In Gear and Kevrekidis (2005); Gear et al. (2005), it was shown that an accurate initialization can be obtained with the so-called *constrained runs lifting* scheme. In its simplest form, this scheme determines, for a value $x^0$ of the observable, the value of $y$ so that $dy(x^0, y)/dt = 0$. The intuitive reason why this condition (or, more generally, a condition demanding that the $y$-time derivative to be bounded) yields a state $(x^0, y)$ close to the slow manifold is that time differentiation amplifies fast components more than slow components, so that, if the time derivatives are small, the fast components in the remaining fine-scale variables are small. In Gear et al. (2005) it was rigorously shown that, under certain conditions, the resulting state is indeed a good approximation to the desired point on the slow manifold. In practice, it is often convenient to approximate the derivative $dy(x^0, y)/dt$ numerically, e.g. using forward differences. In Gear and Kevrekidis (2005); Gear et al. (2005) it was shown that a functional iteration can then, in many cases, be used to find the zero of the resulting forward difference condition. If the step size of the

functional iteration and the forward difference formula are both equal to the step size of the fine-scale simulator $\Delta t$, the functional iteration takes the following form:

0. Initialize $y$ as well as possible. Then start the iteration 1–3.
1. Evolve the fine-scale simulator over one time step of size $\Delta t$, starting from $(x^0, y)$.
2. Reset the value of $x$ to its original value $x^0$.
3. If the difference between the current and the previous value of $y$ is smaller than a certain tolerance `tol`: end the iteration. Else: go to 1.

The iterative scheme above will further be called the *constrained runs functional iteration* (abbreviated CRFI), and the method for approximating the coarse-scale steady state solution that consists of solving equation (4) with $y$ systematically initialized this way will be called METHOD 2.

In some cases, METHOD 1 may produce very inaccurate results when the value of $x$ changes substantially during the fast transients toward the slow manifold. METHOD 2, on the other hand, may find an accurate approximation to the exact solution $x^*$ (more precisely, the error can be made arbitrarily small by decreasing the value of `tol`). This can easily be seen from the fact that

- $dx/dt$ (or its finite difference approximation) is zero in the coarse-scale steady state solution (this is just equation (4)).

- $dy/dt$ (or its finite difference approximation) is zero at the fixed point of the constrained runs functional iteration (because of the nature of this iteration).

In essence, METHOD 2 computes the fine-scale steady state solution as a "splitting scheme", by solving the system $dy/dt = 0$ within each step of an outer solver for the system $dx/dt = 0$. As a result, the computational complexity of METHOD 2 may be as large as that of directly solving the full fine-scale model (which is exactly what we wanted to avoid). Moreover, METHOD 2 may fail when the constrained runs functional iteration *does not converge* to the correct fine-scale state near the slow manifold (the iteration may actually be unstable, or it may converge to a solution that does not correspond to a state close to the slow manifold). In some cases, these convergence issues may be overcome by using a more advanced computational approach such as the one presented in Vandekerckhove et al. (2009), yet then the issue of overall computational complexity also remains.

To cope with the potential accuracy, convergence or efficiency issues, we now propose a third method, METHOD 3, to compute coarse-scale steady state solutions. Instead of solving (4), this method solves

$$\Phi(x, \tau + \tau') - \Phi(x, \tau) = 0, \tag{5}$$

in which $\Phi(x, \tau)$, $\Phi(x, \tau + \tau')$ denote coarse time-steps over times $\tau$, $\tau + \tau'$ in which we use the arbitrary lifting scheme. As before, the value of $\tau$ should be large enough to allow $y$ to get slaved to $x$, but small compared to the coarse time scales. The variable $\tau'$ represents the time interval of an additional simulation on (or very close to) the slow

manifold. If $\overline{x}$ denotes the solution of (5), we expect $\Phi(\overline{x}, \tau) = \Phi(\overline{x}, \tau + \tau')$ (and *not* $\overline{x}$) to be a very good approximation to the exact coarse-scale steady state $x^*$, as the finite difference approximation of $\mathrm{d}x/\mathrm{d}t$ based on two points on the slow manifold is then zero. The obvious advantages compared to METHOD 2 are that this method is conceptually simpler, that it does not involve the (potentially unstable) constrained runs functional iteration, and that we are no longer solving any systems of the form $\mathrm{d}y(x^0, y)/\mathrm{d}t = 0$ in the space of the remaining (fine scale) variables (which may be an advantage when that space is large compared to the space of the observables $y$).

To summarize: we have outlined three different methods to find coarse-scale steady state solutions. In METHOD 1, we solve (4), in which $\Phi(x, \tau)$ represents a coarse time step based on arbitrary lifting. In METHOD 2, we solve (4), in which $\Phi(x, \tau)$ represents a coarse time step based on the constrained runs functional iteration. In METHOD 3, we solve (5), in which $\Phi(x, \tau)$ represents a coarse time step based on arbitrary lifting.

### 2.2. Numerical illustration

We now illustrate the performance of the three methods described above for the model problem (1) with $\alpha = \pi/6$ and $\beta = -\pi/3$. As the smallest eigenvalue of the Jacobian matrix of the time integrator (1) is $\lambda_2 = 0.1$, it takes about 16 time steps for an $\mathcal{O}(1)$ initial condition to reach the slow manifold $y = x/\sqrt{3}$ up to machine precision.

Some examples of a single iteration with METHOD 1 are given in Figure 1 (top-left). The slow manifold is represented by the thick line; the full model steady state solution $(x^*, y^*) = (0, 0)$ on the manifold is indicated by the filled square. For various initial values of $x$, indicated by the filled circles, we perform a time step with a coarse time-stepper that is based on the arbitrary lifting scheme $x \mapsto (x, 1/2)$ and with $\tau = 50\Delta t$ (remember that $\Delta t$ is the time step of the fine scale time-stepper (1)). The fine-scale trajectories are represented by the fine lines; note that the slope of these trajectories is approximately $\tan(-\pi/3) = -\sqrt{3}$. The end points of the trajectories are indicated by the open circles; these points clearly lie on the slow manifold. The solution of METHOD 1, the $x$-coordinate of the small open circle, is also shown; the initial condition of the corresponding simulation trajectory is encircled as well. For this trajectory, the $x$-values of the initial and end points are equal, as demanded by (4). In this case, however, the value of $x$ found is 0.719, which is clearly different from the exact value $x^* = 0$. As mentioned above, this is due to the fact that the value of $x$ changes substantially during the fast transients towards the slow manifold ($\beta \not\approx \pm\pi/2$); clearly, the solution found with METHOD 1 depends on the value of $\tau$.

An illustration of METHOD 2 is given in Figure 1 (right). Again, the slow manifold is represented by the thick line and the full model steady state solution $(x^*, y^*) = (0, 0)$ is indicated by the filled square. From various initial values of $x$, indicated by the filled circles, we perform the constrained runs functional iteration with $\texttt{tol} = 10^{-15}$, after which we perform an additional simulation over the time interval $\tau = 50\Delta t$ (here, we only chose such a large value of $\tau$ to make Figure 1 more clear; in practice there is little reason to use such a large value of $\tau$). The simulation trajectories and the resetting of the observable in the constrained runs functional iteration are represented by the fine lines and arrows,
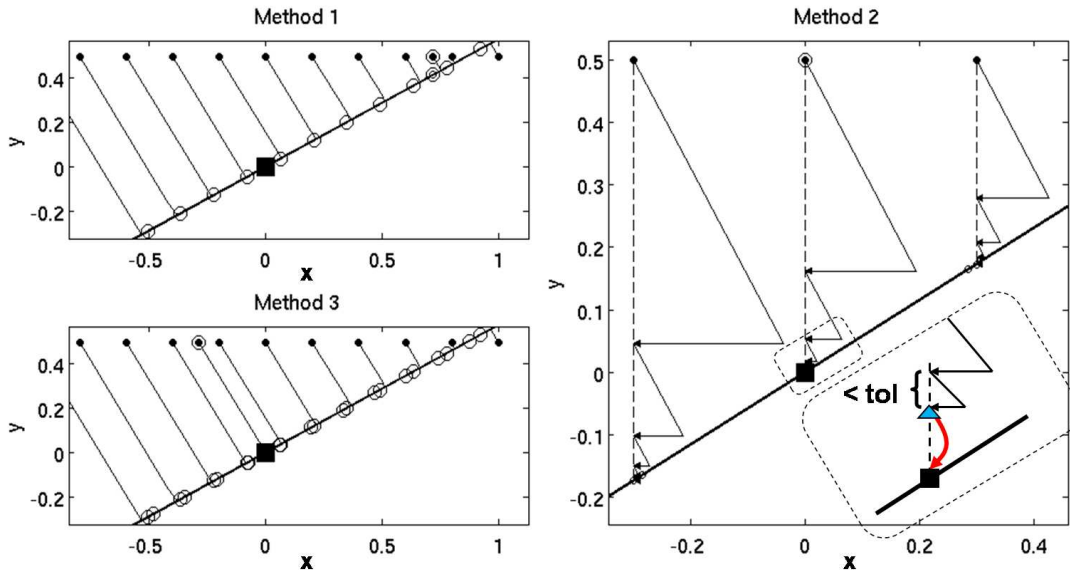
Figure 1: An illustration of the three methods used to compute coarse-scale steady states. Shown are the slow manifold, fast transients, and computed coarse-scale steady states. See text for further explanation.

respectively. The end points of the constrained runs functional iteration and the simulation trajectory afterwards are indicated by the open circles. We observe that, in this example, the constrained runs functional iteration always brings us very close to the slow manifold. The final point reached by METHOD 2 is also shown; the initial condition of the corresponding simulation trajectory is encircled. As in the case of METHOD 1, the initial value of $x$ is the same as the end value of $x$, as demanded by (4). In this case, however, the value of $x$ found is $3.02 \cdot 10^{-15}$, which, as expected, approximates the exact value $x^* = 0$ up to the tolerance `tol`. In the lower right of this figure, one can see the details of METHOD 2: first, the constrained runs functional iteration is performed until the difference in successive $y$ values is less than `tol`, leaving us at the (blue) triangle; next, we perform a coarse time step of length $\tau = 50\Delta t$ using the $y$-coordinates of this triangle for the *lifting*, shown as the curvy (red) arrow; finally, because the $x$-coordinate of the (blue) triangle coincides with the head of the curvy (red) arrow (meaning that $\Phi(x, \tau) - x = 0$), we determine that we are at a coarse-scale steady state.

Due to the fact that the constrained runs lifting brings us very close to the manifold, we may even use a smaller value of $\tau$ than in METHOD 1; even if $\tau = \Delta t$, we obtain $x = 1.05 \cdot 10^{-13}$. It is also worth mentioning that if we had chosen $\beta = \pi/4$, the constrained runs functional iteration would not have converged, as the iteration is then unstable. For our model problem, this can easily be rationalized using geometric arguments (Vandekerckhove (2008)).

An illustration of METHOD 3 is given in Figure 1 (bottom-left). Again, we used the

same line styles and markers as before, but now both the end points of the simulation trajectories over time $\tau = 50\Delta t$ and $\tau + \tau' = 100\Delta t$ are indicated by the open circles (as before, we only chose such a large value of $\tau'$ to make Figure 1 more clear; in practice there is little reason to use such a large value of $\tau'$). The solution of METHOD 3 is also shown; the initial condition and the end point of the corresponding simulation trajectory are encircled. For this solution, the initial value of $x$ *is not the same* as the end value of $x$; yet the $x$-values of the end points of the simulation trajectories after the time interval $\tau$ and $\tau + \tau'$ coincide, as demanded for in (5). The value of $x$ found is now $1.89 \cdot 10^{-18}$, which corresponds to the exact value $x^* = 0$ up to machine precision. Even if $\tau' = \Delta t$, we obtain $x = 3.29 \cdot 10^{-17}$.

## 3. Initializing on a slow manifold

Remarkably, the modification we presented as METHOD 3 above to improve the approximation of coarse-scale *steady state* solutions can form the basis of an algorithm for appropriately initializing our fine-scale simulators given a desired value of the observable(s). In the context of our simple example, this means finding the point $(x^0, y)$ on the slow manifold corresponding to some prescribed value of $x$ we denote $x^0$. We already showed how to use the constrained runs functional iteration (used in METHOD 2) for this purpose. The fixed point of the constrained runs functional iteration lies close to (but, if $x^0$ is not a coarse-scale steady state solution, not exactly on) the slow manifold. More accurate initializations can be obtained by using variants of the constrained runs functional iteration outlined in the literature (Kreiss and Lorenz (1994); Lorenz (1986); Curry et al. (2002); Vandekerckhove et al. (2009); Gear et al. (2005); Gear and Kevrekidis (2005)) that solve $d^{m+1}y(x^0, y)/dt^{m+1} = 0$ for a certain value of $m \in \mathbb{N}$. The larger the value of $m$, the more accurate the procedure will be (Gear et al. (2005)), assuming it converges. For these variants of the scheme, however, the constrained runs functional iteration is not guaranteed to converge to a solution close to the slow manifold, and the computational complexity may be unacceptably large. For these reasons, we now propose an alternative initialization method that has many similarities with METHOD 3, but now, instead of computing coarse-scale steady state solutions, we compute points lying on the slow manifold for a given value of the observable, $x^0$.

Instead of demanding that the finite difference approximation of the time derivative of the observable is zero after a simulation over time $\tau$ as in (5), we now compute $x$ so that

$$\Phi(x, \tau) - x^0 = 0. \tag{6}$$

As in METHOD 3, $\Phi(x, \tau)$ denotes a coarse time-step over time $\tau$ in which we use the simple arbitrary lifting scheme. Again, the value of $\tau$ should be large enough to allow $y$ to get slaved to $x$, but much smaller than the coarse time scales. If $\overline{x}$ is the solution of (6), we expect the fine-scale solution $(x, y)$, obtained after simulation over time $\tau$ starting from the arbitrary lifted state corresponding to $\overline{x}$, to be a good approximation of the desired point on the slow manifold (the simulation step has brought us close to the slow manifold

7

Table 1: Absolute value of the error in the solution of the constrained runs functional iteration and INITMAN.

| | $|\text{error}|$ | | | $|\text{error}|$ |
|---|---|---|---|---|
| CRFI, $m = 0$ | $8.55 \cdot 10^{-04}$ | | CRFI, $m = 3$ | $1.17 \cdot 10^{-12}$ |
| CRFI, $m = 1$ | $9.50 \cdot 10^{-07}$ | | CRFI, $m = 4$ | $3.11 \cdot 10^{-15}$ |
| CRFI, $m = 2$ | $1.06 \cdot 10^{-09}$ | | INITMAN | $2.22 \cdot 10^{-16}$ |

at the desired value $x^0$). The obvious advantages of this algorithm (which we will refer to as INITMAN) compared to the constrained runs functional iteration are that this method is conceptually simpler, that it does not have the same potential convergence issues, and that we are no longer solving a system of equations of the form $\mathrm{d}y(x^0, y)/\mathrm{d}t = 0$ in the space of the "remaining variables" (which may be an advantage if that space is large compared to the space of the observables). We still do, of course, require a solver for equation (6), as would also be required for equation (4) above; this might be something as simple as a Newton or Broyden method, or as sophisticated as a Newton-Krylov GMRES solver (Kelley (1995)).

To illustrate the performance of the (variants of the) constrained runs functional iteration and INITMAN, we again consider the model problem (1) with $\alpha = \pi/6$ and $\beta = -\pi/3$. Using five different values of $m$ for the constrained runs functional iteration, and also using INITMAN, we approximate the value of $y$ so that $(1, y)$ lies as close as possible to the exact point on the slow manifold, $(1, 1/\sqrt{3})$. Table 1 shows, for various values of $m$, the error in the solution found by the constrained runs functional iteration with $\texttt{tol} = 10^{-16}$, and also the error found by INITMAN(solved with a Newton iteration). We clearly observe that, as the value of $m$ increases, the error decreases by a factor of about $(1 - 0.999)/(1 - 0.1) \approx 1.11 \cdot 10^{-3}$, as expected by theory (Vandekerckhove (2008)). The errors in the solution found by both the $m = 4$ constrained runs functional iteration and INITMAN with $\tau = 15\Delta t$ are at the level of machine precision.

## 4. Application to a lattice Boltzmann model

In this section, we will apply the algorithms we have presented to a lattice Boltzmann model (LBM) of a one-dimensional reaction-diffusion system. In Section 4.1, we present the LBM. In Section 4.2 we analytically derive two reduced models in terms of two different coarse observables. In Section 4.3, a detailed description of the numerical results is given.

### 4.1. The lattice Boltzmann model

An LBM (Chopard et al. (2002)) describes the evolution of discrete (particle) distribution functions $f_i(x_j, t_k)$, which depend on space $x_j$, time $t_k$ and velocity $v_i$. For our one-dimensional model problem, only three values are considered for the velocity ($v_i = i\Delta x/\Delta t$, with $i \in \{-1, 0, 1\}$), and each distribution function $f_i$ is discretized in space on the domain $[0, 1]$ using a grid spacing $\Delta x = 1/N$ ($N$ lattice intervals) and in time using a time step

$\Delta t$. The LBM evolution law for the distributions $f_i(x_j, t_k)$ in the interior of the domain is

$$
\begin{aligned}
f_i(x_{j+i}, t_{k+1}) &= f_i(x_j + i\Delta x, t_k + \Delta t) \\
&= f_i(x_j, t_k) - \omega\left(f_i(x_j, t_k) - f_i^{eq}(x_j, t_k)\right) + \frac{\Delta t}{3} F(\rho(x_j, t_k)),
\end{aligned}
\tag{7}
$$

with $i \in \{-1, 0, 1\}$.

Diffusive collisions are modeled by the Bhatnagar-Gross-Krook (BGK) collision term $-\omega(f_i(x_j, t_k) - f_i^{eq}(x_j, t_k))$ as a relaxation to the local diffusive equilibrium (Qian and Orszag (1995))

$$
f_i^{eq}(x_j, t_k) = \frac{1}{3}\rho(x_j, t_k).
\tag{8}
$$

The parameter $\omega \in (0, 2)$ is called the relaxation coefficient and $\rho$ is the (particle) density field, which is defined as the "zeroth" order velocity moment of $f_i(x_j, t_k)$

$$
\rho(x_j, t_k) = \sum_{i=-1}^{1} f_i(x_j, t_k) = f_{-1}(x_j, t_k) + f_0(x_j, t_k) + f_1(x_j, t_k).
$$

It follows directly that the BGK diffusive collisions locally conserve density.

The last term in equation (7) models the reactions, which are assumed to depend only on the density field $\rho$ (Qian and Orszag (1995); Dawson et al. (1993)). In this paper, we will use the specific reaction term

$$
F(\rho(x_j, t_k)) = \lambda\rho(x_j, t_k)\left(1 - \rho(x_j, t_k)\right),
\tag{9}
$$

in which the parameter $\lambda \geq 0$ determines the strength of the reaction "force". Nonlinear reaction terms of the form (9) arise naturally in the fields of heat and mass transfer (Danilov et al. (1995)) or in ecology (Holmes et al. (1994)).

At the boundaries, we impose Dirichlet boundary conditions $\rho(0, t_k) = \rho(1, t_k) = 0$ by assigning the appropriate values to the distribution functions that stream into the domain at $x_0 = 0$ and $x_N = 1$.

Similar to the density $\rho$, we can define the momentum $\phi$ and the energy $\xi$ as (a rescaling of) the first and the second (or in short, the higher) order moments of $f_i$

$$
\phi(x_j, t_k) = \sum_{i=-1}^{1} i f_i(x_j, t_k) = -f_{-1}(x_j, t_k) + f_1(x_j, t_k),
$$

$$
\xi(x_j, t_k) = \frac{1}{2}\sum_{i=-1}^{1} i^2 f_i(x_j, t_k) = \frac{f_{-1}(x_j, t_k) + f_1(x_j, t_k)}{2}.
$$

Later on we will also use the variable $\sigma$, which is defined as

$$
\sigma(x_j, t_k) = \sum_{i=-1}^{1}(2i^2 - 1)f_i(x_j, t_k) = f_{-1}(x_j, t_k) - f_0(x_j, t_k) + f_1(x_j, t_k) = -\rho(x_j, t_k) + 4\xi(x_j, t_k).
$$

## 4.2. Analytical coarse-graining

For the LBM described in the previous section, a Chapman-Enskog multiscale expansion can be used to derive an accurate reduced model for the long-term behavior of the system (Chopard et al. (2002); Succi (2001)). (In practice, the equation-free methods should of course only be used when such a derivation is not possible. Here, however, the analytical derivation provides insight into the problem, which is particularly helpful in understanding the performance of the different methods.) In this section, we will derive two different reduced models: one in terms of the density $\rho$ and another in terms of the variable $\sigma$. For a detailed derivation of the following two equations, we refer to appendix B of Vandekerckhove (2008).

If we use the density $\rho$ as the observable, the reduced model is the partial differential equation

$$\frac{\partial \rho(x,t)}{\partial t} = D\frac{\partial^2 \rho(x,t)}{\partial x^2} + F(\rho) = \left(\frac{2-\omega}{3\omega}\frac{\Delta x^2}{\Delta t}\right)\frac{\partial^2 \rho(x,t)}{\partial x^2} + F(\rho) \tag{10}$$

with Dirichlet boundary conditions $\rho(0,t) = \rho(1,t) = 0$. If we consider the variable $\sigma$ to be the observable, we obtain the partial differential equation

$$\frac{\partial \sigma(x,t)}{\partial t} = D\frac{\partial^2 \sigma(x,t)}{\partial x^2} + \frac{1}{3}F(3\sigma) = \left(\frac{2-\omega}{3\omega}\frac{\Delta x^2}{\Delta t}\right)\frac{\partial^2 \sigma(x,t)}{\partial x^2} + \frac{1}{3}F(3\sigma) \tag{11}$$

with Dirichlet boundary conditions $\sigma(0,t) = \sigma(1,t) = 0$.

As a by-product of the Chapman-Enskog expansion, we find, after dropping the indices $j$ and $k$, and retaining only terms up to second order, that the relation between $\sigma$ and $\rho$ is given by

$$\sigma(x,t) = \frac{\rho(x,t)}{3} + \frac{2(2-\omega)}{9\omega^2}\frac{\partial^2 \rho(x,t)}{\partial x^2}\Delta x^2 + \mathcal{O}(\Delta x^3)$$
$$\rho(x,t) = 3\sigma(x,t) - \frac{2(2-\omega)}{\omega^2}\frac{\partial^2 \sigma(x,t)}{\partial x^2}\Delta x^2 + \mathcal{O}(\Delta x^3). \tag{12}$$

This shows that in a LBM simulation, the value of $\rho$ is approximately three times as large as the value of $\sigma$, at least after a short initial transient. From this point of view, the choice of $\sigma$ as the observable is as natural as the choice of $\rho$.

The fact that the reduced dynamics can be described in terms of $\rho$ or $\sigma$ only implies that the remaining "fine-scale variables" $\phi$ and $\xi$ quickly become functionals of (slaved to) $\rho$ or $\sigma$. These functionals, which we will call slaving relations, are

$$\phi(x,t) = -\frac{2}{3\omega}\frac{\partial \rho(x,t)}{\partial x}\Delta x + \mathcal{O}(\Delta x^3) \tag{13}$$

$$\xi(x,t) = \frac{1}{3}\rho(x,t) - \frac{\omega-2}{18\omega^2}\frac{\partial^2 \rho(x,t)}{\partial x^2}\Delta x^2 + \mathcal{O}(\Delta x^3) \tag{14}$$

in terms of the density $\rho$, or

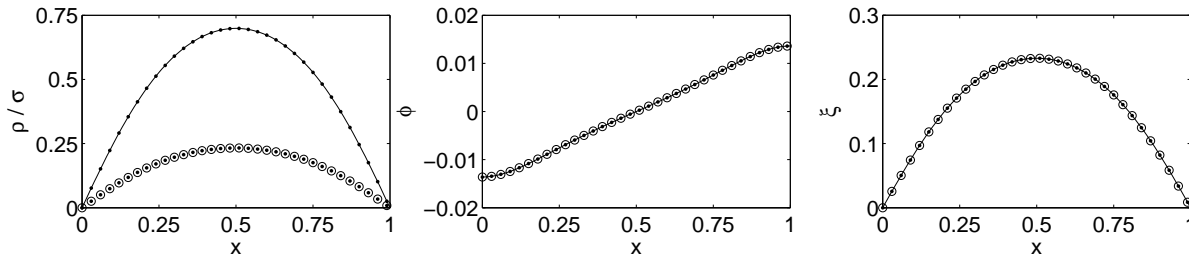$$\phi(x,t) = -\frac{2}{\omega}\frac{\partial \sigma(x,t)}{\partial x}\Delta x + \mathcal{O}(\Delta x^3) \tag{15}$$

10

Figure 2: The full LBM steady state solution, computed in both the $(\rho, \phi, \xi)$- and the $(\sigma, \phi, \xi)$-coordinate systems using solid line and circle markers, respectively. This solution is shown in $\rho/\sigma$ space (left), $\phi$ space (middle), and $\xi$ space (right). Dot markers are also present to demonstrate the validity of (10) and (11) (left), (13) and (15) (middle), and (14) and (16) (right).

$$\xi(x,t) = \sigma(x,t) + \frac{\omega - 2}{2\omega^2}\frac{\partial^2\sigma(x,t)}{\partial x^2}\Delta x^2 + \mathcal{O}(\Delta x^3) \tag{16}$$

in terms of the variable $\sigma$. The slaving relations (13)-(14) or (15)-(16) define a slow manifold in the LBM phase space, on which the reduced dynamics takes place.

The validity of the reduced models (10)-(11) and the slaving relations (12)–(16) is illustrated in Figure 2. Here, we chose $N = 100$, $\lambda = 25$ and $\omega = 1.25$, and computed the full LBM steady state solution in both the $(\rho, \phi, \xi)$- and the $(\sigma, \phi, \xi)$-coordinate systems. These solutions are shown using the solid line and the circle markers, respectively. To simplify the interpretation, we split the solution into the $\rho$-, $\sigma$-, $\phi$- and $\xi$-components, and mapped them onto the spatial domain $[0, 1]$. From these solutions, it is confirmed that the value of $\sigma$ is indeed about one third of the value of $\rho$. Then, we also computed the steady state solutions of (10) and (11), and added them in the form of the dot markers to Figure 2 (left). At the resolution shown here, these reduced solutions are clearly indistinguishable from the $\rho$- and $\sigma$-components of the full LBM solution. Finally, we added to Figure 2 (middle and right), also in the form of the dot markers, the profiles of $\phi$ and $\xi$ according to the slaving relations (13)-(14) (exactly the same results are obtained when using (15)-(16)). Here, we approximated the spatial derivatives of $\rho$ numerically using finite differences. In this case also, the values of $\phi$ and $\xi$ are indistinguishable from the $\phi$- and $\xi$-components of the full LBM solution.

Any LBM initial condition is quickly attracted towards the slow manifold during the transient LBM simulation. For $\omega = 1.25$, for instance, the slow manifold is reached, up to machine precision, in about 25 LBM time steps (the values of $\phi$ and $\xi$ in the simulation converge linearly to the slaving relations with convergence factor $|1 - \omega|$, as can be seen from (7)). The value of the observable may however change during the transient phase towards the slow manifold:

- Suppose that we start from the initial condition $(\rho, \phi, \xi) = (\rho_0, 0, 0)$. In terms of the distribution functions, this corresponds to $(f_{-1}, f_0, f_1) = (0, \rho_0, 0)$. During the initial transient phase, the values of the distribution functions will quickly be redistributed

11

to $(f_{-1}, f_0, f_1) \approx (\rho_0/3, \rho_0/3, \rho_0/3)$, or, in terms of the velocity moments, to $(\rho, \phi, \xi) \approx (\rho_0, 0, \rho_0/3)$ (this follows directly from (7)-(8) and fact that the evolution towards the slow manifold is very fast compared to the diffusive or reactive time scales). Hence, the density $\rho$ *does not change substantially* during the transient phase.

- Suppose that we start from the initial condition $(\sigma, \phi, \xi) = (\sigma_0, 0, 0)$. In terms of the distribution functions, this corresponds to $(f_{-1}, f_0, f_1) = (0, -\sigma_0, 0)$. During the initial transient phase, the values of the distribution functions will quickly be redistributed to $(f_{-1}, f_0, f_1) \approx (-\sigma_0/3, -\sigma_0/3, -\sigma_0/3)$, or, in terms of $\sigma$ and the higher order velocity moments, to $(\sigma, \phi, \xi) \approx (-\sigma_0/3, 0, -\sigma_0/3)$. Hence, the variable $\sigma$ *does change substantially* during the transient phase from $\sigma_0$ to $-\sigma_0/3$.

As we demonstrated before (and will see below), the fact that the observable changes substantially during the transient phase towards the slow manifold may have important consequences for the performance of the different methods.

### 4.3. Numerical results

In all numerical experiments reported below, we use $D = 1$, $\Delta x = 1/N = 1/100$ and $\omega = 1.25$. According to (10)-(11), the value of the LBM time step is then $\Delta t = 2 \cdot 10^{-5}$. For $\lambda$, we will choose $\lambda = 25$ or $\lambda = 5$. For $\lambda = 25$, the LBM (7)–(9) exhibits a nontrivial *stable* steady state solution; for $\lambda = 10$, the nontrivial steady state solution is *unstable* (Vandekerckhove (2008)). These solutions will henceforth be referred to as "the stable steady state solution" and "the unstable steady state solution", respectively.

To solve the nonlinear systems (4), (5) or (6), which we will write below in more abstract form as $g(x) = 0$, we use the most basic implementation of Newton's method in which we estimate the required Jacobian matrices $\partial g/\partial x(x_i)$ in a column-by-column fashion. Note that these Jacobian matrices are typically not very large, as their size is determined by the dimension of the *coarse* subspace; still, the cost of the computational linear algebra should be taken into consideration. Column $l$ of $\partial g/\partial x(x_i)$ is the directional derivative in the direction of the $l$-th unit vector $e_l$, which can be approximated as

$$\frac{\partial g}{\partial x}(x_i) \cdot e_l \approx \frac{g(x_i + \varepsilon e_l) - g(x_i)}{\varepsilon}, \tag{17}$$

with $\varepsilon$ an appropriate small parameter. The resulting linear systems $\partial g/\partial x(x_i)\Delta x = -g(x_i)$ are then solved by Gaussian elimination. For problems for which the coarse subspace is very large, it may be more appropriate to solve the linear systems using, for instance, a Jacobian-free Newton-Krylov method (Kelley (1995)). The use of these advanced methods will not be considered in this article.

#### 4.3.1. Results using $\rho$ as the observable

In this section, we apply our algorithms and the constrained runs functional iteration to the LBM, using the density $\rho$ as the observable. As $\rho$ does not change substantially during the transient phase towards the slow manifold, we expect to obtain good results

Table 2: Two-norm of the error $||e||_2 = ||\rho_i - \rho^*||_2$ in the steady state solution for METHOD 1 $(i = 1)$, METHOD 2 $(i = 2)$, and METHOD 3 $(i = 3)$, using $\lambda = 25$ and $\lambda = 5$. $\rho^*$ is known analytically for each $\lambda$.

|  | METHOD 1 | METHOD 2 | METHOD 3 |
|---|---|---|---|
| $\lambda = 25$ (stable) | $1.01 \cdot 10^{-1}$ | $1.07 \cdot 10^{-13}$ | $2.58 \cdot 10^{-13}$ |
| $\lambda = 5$ (unstable) | $4.79 \cdot 10^{-1}$ | $1.47 \cdot 10^{-12}$ | $1.85 \cdot 10^{-12}$ |

with all methods. Note that the results reported here are in perfect correspondence with the theoretical results obtained in Vandekerckhove et al. (2008), in which it was shown that the coarse time-stepper under consideration is actually a time integrator for a (slightly) modified reaction-diffusion system.

*Coarse-scale steady states: application of* METHOD 1. We approximate the $\rho$-component of the stable or unstable steady state solution with METHOD 1, in which we use the arbitrary lifting scheme $\rho \mapsto (\rho, 0, 0)$ and set $\tau = 25\Delta t$. We choose $\varepsilon = \sqrt{\eta}$, with $\eta \approx 2.22 \cdot 10^{-16}$ the machine precision. As the initial condition for Newton's method, we take $\rho^0 = \sin(\pi x)$ when $\lambda = 25$ and $\rho^0 = -\sin(\pi x)$ when $\lambda = 5$. For both values of $\lambda$, the nonlinear residual reaches the level of machine precision after 5 Newton steps; the resulting solution is called $\rho_1$. Table 2 shows the two-norm of the error $||e||_2 = ||\rho_1 - \rho^*||_2$ for $\lambda = 25$ and $\lambda = 5$ (as before, we use $\rho^*$ to denote the $\rho$-component of the full LBM steady state solution). The error is not large, but it is also not small (the relative error is about 2% when $\lambda = 25$ and about 6% when $\lambda = 5$). This can be explained by the fact that $\rho$ *does change*, although very slightly, during the fast transient phase. Note that changing $\tau$ will also influence the approximation of the steady-state solution.

*Coarse-scale steady states: application of* METHOD 2. We now approximate the $\rho$-component of the stable or unstable steady state solution with METHOD 2, in which we set $\tau = \Delta t$ and use the constrained runs lifting scheme with $m = 0$ and `tol` $= 10^{-14}$, starting from the initial condition $(\phi, \xi) = (0, 0)$. Again, we choose $\varepsilon = \sqrt{\eta}$, and use $\rho^0 = \sin(\pi x)$ or $\rho^0 = -\sin(\pi x)$ as the initial condition for Newton's method. In Van Leemput et al. (2008), it was shown that for $m = 0$ and $\rho$ as the observable, the eigenvalues of the constrained runs iteration matrix lie on a circle with center point 0 and radius $|1 - \omega|$. As a consequence, the iteration is stable for all values of $\omega \in (0, 2)$ and for $\omega = 1.25$ the convergence factor is 0.25 (in other words, about 25 iterations are needed to reach the tolerance `tol`). As in METHOD 1, the nonlinear residual reaches the level of machine precision after 5 Newton steps for both values of $\lambda$; the resulting solution is now called $\rho_2$. Table 2 shows the two-norm of the error $||e||_2 = ||\rho_2 - \rho^*||_2$ for $\lambda = 25$ and $\lambda = 5$. The error is now clearly very small.

*Coarse-scale steady states: application of* METHOD 3. We approximate the $\rho$-component of the stable or unstable steady state solution with METHOD 3, in which we use the arbitrary lifting scheme $\rho \mapsto (\rho, 0, 0)$ and set $\tau = 25\Delta t$ and $\tau' = \Delta t$. To avoid numerical complications due to (nearly) singular Jacobian matrices, we explicitly eliminate the

13

Table 3: Two-norm of the error in the solution of the constrained runs functional iteration and INITMAN, when $\rho$ is the observable.

|  | $\lambda = 25$ | $\lambda = 5$ |
|---|---|---|
| CRFI, $m = 0$ | $9.97 \cdot 10^{-5}$ | $7.00 \cdot 10^{-5}$ |
| CRFI, $m = 1$ | $2.47 \cdot 10^{-7}$ | $2.59 \cdot 10^{-7}$ |
| CRFI, $m \geq 2$ | $\infty$ | $\infty$ |
| INITMAN | $1.03 \cdot 10^{-15}$ | $1.19 \cdot 10^{-15}$ |

boundary conditions $\rho(0) = \rho(1) = 0$ (so the dimension of the Jacobian matrix $\partial g/\partial x(x_i)$ is $(N-1) \times (N-1)$ instead of $(N+1) \times (N+1)$). We also set $\varepsilon = \sqrt[4]{\eta}$ (see also the discussion in Section 5). As the initial condition for Newton's method, we again use $\rho^0 = \sin(\pi x)$ or $\rho = -\sin(\pi x)$. For both values of $\lambda$, the nonlinear residual reaches the level of machine precision after about 6 or 7 Newton steps; the resulting solution is now called $\overline{\rho}_3$. Table 2 shows the two-norm of the error $||e||_2 = ||\rho_3 - \rho^*||_2 = ||\Phi(\overline{\rho}_3, \tau) - \rho^*||_2$ for $\lambda = 25$ and $\lambda = 5$. Again, the error is very small.

*Initialization on the slow manifold: the constrained runs functional iteration and* INITMAN. To test how the constrained runs functional iteration and INITMAN perform when we attempt to initialize on the slow manifold, we set up the following experiment. For both $\lambda = 25$ and $\lambda = 5$, we first perform a LBM simulation of 50 steps, starting from the (arbitrary) initial condition $(\rho^0, \phi^0, \xi^0) = (x(1-x), x, \sin(\pi x))$. This provides us with a LBM state $(\rho^*, \phi^*, \xi^*)$ "on" the slow manifold which is not a coarse scale steady state. Then, we use the constrained runs functional iteration or INITMAN to approximate the values of $\phi^*$ and $\xi^*$ corresponding to $\rho^*$. For the constrained runs functional iteration, we use various values of $m$, set $\mathtt{tol} = 10^{-14}$ and start from the initial condition $(\phi, \xi) = (0, 0)$. For INITMAN, we set $\tau = 25\Delta t$, start from the initial condition $\rho^0 = \rho^*$, and again explicitly eliminate the boundary conditions $\rho(0) = \rho(1) = 0$ to avoid numerical complications due to (nearly) singular Jacobian matrices. We also set $\varepsilon = \sqrt[4]{\eta}$ (see also the discussion in Section 5).

The results are summarized in Table 3. For both the constrained runs functional iteration and INITMAN , and for $\lambda = 25$ and $\lambda = 5$, we tabulate the two-norm of the error $||e||_2 = ||(\phi^\#, \xi^\#) - (\phi^*, \xi^*)||_2$ (we use $(\phi^\#, \xi^\#)$ to denote the solution found by the constrained runs functional iteration or INITMAN). The solution of the constrained runs functional iteration with $m = 1$ is more accurate than the solution obtained when $m = 0$, but for values of $m \geq 2$ the iteration is *unstable* (some of the eigenvalues of the constrained runs iteration matrix are larger than 1 in magnitude). For INITMAN, the nonlinear residual reaches the level of machine precision after about 3 or 4 Newton steps; the resulting solution is called $\overline{\rho}_4$. In this case, $\phi^\#$ and $\xi^\#$ are the values of the higher order moments $\phi$ and $\xi$ obtained after a LBM simulation over time $\tau$ starting from $(\overline{\rho}_4, 0, 0)$. The solution found by INITMAN is clearly very accurate.

Table 4: Two-norm of the error $||e||_2 = ||\sigma_i - \sigma^*||_2$ in the steady state solution for METHOD 1($i = 1$), METHOD 2 ($i = 2$) and METHOD 3($i = 3$), and using $\lambda = 25$ and $\lambda = 5$. $\sigma^*$ is known analytically for each $\lambda$.

|  | METHOD 1 | METHOD 2 | METHOD 3 |
|---|---|---|---|
| $\lambda = 25$ (stable) | 1.70 | $\infty$ | $3.77 \cdot 10^{-13}$ |
| $\lambda = 5$ (unstable) | 2.69 | $\infty$ | $6.16 \cdot 10^{-13}$ |

### 4.3.2. Results using $\sigma$ as the observable

In this section, we apply our algorithms as well as the constrained runs functional iteration to the LBM, this time using the variable $\sigma$ as the observable. As $\sigma$ *does change* substantially during the transient phase towards the slow manifold, we expect to obtain poor results with METHOD 1, METHOD 2, and the constrained runs functional iteration, but good results with METHOD 3 and INITMAN. Tables 4 and 5 are the analogues of Tables 2 and 3; they tabulate the results of the numerical experiments described below. Note that the results reported here are in perfect correspondence with the theoretical results obtained in Vandekerckhove et al. (2008), in which it was shown that the coarse time-stepper under consideration is actually a time integrator for a (slightly) modified reaction-diffusion system.

*Coarse-scale steady states: application of* METHOD 1. We approximate the $\sigma$-component of the stable or unstable steady state solution with METHOD 1, in which we use the arbitrary lifting scheme $\sigma \mapsto (\sigma, 0, 0)$ and set $\tau = 25\Delta t$. As before, we choose $\varepsilon = \sqrt{\eta}$. As the initial condition for Newton's method, we use $\sigma^0 = \sin(\pi x)/3$ when $\lambda = 25$ or $\sigma^0 = -\sin(\pi x)/3$ when $\lambda = 5$ (this choice is motivated by the fact that on the slow manifold, the value of $\sigma$ is about one third of the value of $\rho$; cf. (12)). For both values of $\lambda$, the nonlinear residual reaches the level of machine precision after 3 Newton steps; the resulting solution is called $\sigma_1$. Table 4 shows the two-norm of the error $||e||_2 = ||\sigma_1 - \sigma^*||_2$ for $\lambda = 25$ and $\lambda = 5$ (as before, we use $\sigma^*$ to denote the $\sigma$-component of the full LBM steady state solution). The error is clearly unacceptable (the iteration converges to $\sigma = 0$).

*Coarse-scale steady states: application of* METHOD 2. We approximate the $\sigma$-component of the stable or unstable steady state solution with METHOD 2, in which we set $\tau = \Delta t$ and use the constrained runs lifting scheme with $m = 0$ and $\texttt{tol} = 10^{-14}$, starting from the initial condition $(\phi, \xi) = (0, 0)$. Again, we choose $\varepsilon = \sqrt{\eta}$, and use $\sigma^0 = \sin(\pi x)/3$ or $\sigma^0 = -\sin(\pi x)/3$ as the initial condition for Newton's method. For $m = 0$, $\omega = 1.25$ and $\lambda = 0$ (we use $\lambda = 0$ rather than $\lambda = 25$ or $\lambda = 5$ as the iteration is then linear), and using $\sigma$ as the observable, the eigenvalues of the constrained runs iteration matrix lie in $(-1.416, -0.25) \cup (0.25, 1.416)$. Also for nonzero values of $\lambda$, the eigenvalues of the (varying) iteration matrix fall outside the unit circle. As a consequence, the iteration is *unstable* and METHOD 2 cannot be used; it cannot even be started (see Table 4).

*Coarse-scale steady states: application of* METHOD 3. We approximate the $\sigma$-component of the stable or unstable steady state solution with METHOD 3, in which we use the arbitrary lifting scheme $\sigma \mapsto (\sigma, 0, 0)$ and set $\tau = 25\Delta t$ and $\tau' = \Delta t$. To avoid numerical complications due to (nearly) singular Jacobian matrices, we again explicitly eliminate the boundary conditions $\sigma(0) = \sigma(1) = 0$ and set $\varepsilon = \sqrt[4]{\eta}$ (see also the discussion in Section 5). As the initial condition for Newton's method, we now use $\sigma^0 = -\sin(\pi x)$ or $\sigma^0 = \sin(\pi x)$ (this choice is motivated by the fact that, after the initial transient towards the slow manifold, the value of $\sigma$ is about minus one third of the value of $\sigma^0$, so that we then end up near $\sigma = \sin(\pi x)/3$ and $\sigma = -\sin(\pi x)/3$; cf. the last paragraph in Section 4.2). For both values of $\lambda$, the nonlinear residual reaches the level of machine precision after about 8 ($\lambda = 25$) or 20 ($\lambda = 5$) Newton iteration steps; the resulting solution is now called $\overline{\sigma}_3$. Table 4 shows the two-norm of the error $||e||_2 = ||\sigma_3 - \sigma^*||_2 = ||\Phi(\overline{\sigma}_3, \tau) - \sigma^*||_2$ for $\lambda = 25$ and $\lambda = 5$. Again, the error is very small.

*Initialization on the slow manifold: the constrained runs functional iteration and* INITMAN. We now turn again to the problem of initializing on the slow manifold. To test the constrained runs functional iteration and INITMAN, we set up the following experiment. For both $\lambda = 25$ and $\lambda = 5$, we first perform a LBM simulation of 50 steps, starting from the (arbitrary) initial condition $(\sigma^0, \phi^0, \xi^0) = (x(1-x), x, \sin(\pi x))$. This provides us with a LBM state $(\sigma^*, \phi^*, \xi^*)$ "on" the slow manifold. Then, we use the constrained runs functional iteration or INITMAN to approximate the values $\phi^*$ and $\xi^*$ corresponding to $\sigma^*$. For the constrained runs functional iteration, we use various values of $m$, set $\mathtt{tol} = 10^{-14}$ and start from the initial condition $(\phi, \xi) = (0, 0)$. For INITMAN, we set $\tau = 25\Delta t$, start from the initial condition $\sigma^0 = -3\sigma^*$ (this choice is again motivated by the fact that, after the initial transient towards the slow manifold, the value of $\sigma$ is about minus one third of the value of $\sigma^0$, so that we then end up near $\sigma = \sigma^*$; cf. the last paragraph in Section 4.2) and again explicitly eliminate the boundary conditions $\sigma(0) = \sigma(1) = 0$ and set $\varepsilon = \sqrt[4]{\eta}$ to avoid numerical complications due to (nearly) singular Jacobian matrices (see also the discussion in Section 5).

The results are summarized in Table 5. For both the constrained runs functional iteration and INITMAN, and for $\lambda = 25$ and $\lambda = 5$, we tabulate the two-norm of the error $||e||_2 = ||(\phi^\#, \xi^\#) - (\phi^*, \xi^*)||_2$ (as before, we use $(\phi^\#, \xi^\#)$ to denote the solution found by the constrained runs functional iteration or INITMAN). As already indicated above, the constrained runs functional iteration is always *unstable*. For INITMAN, the nonlinear residual reaches the level of machine precision after about 5 or 6 Newton steps; the resulting solution is called $\overline{\sigma}_4$. In this case, $\phi^\#$ and $\xi^\#$ are the values of the higher order moments $\phi$ and $\xi$ obtained after a LBM simulation over time $\tau$ starting from $(\overline{\sigma}_4, 0, 0)$. As before, the solution found by INITMAN is extremely accurate (one cannot expect to do better, in fact).

## 5. Numerics

In all of the experiments above, we obtained accurate results with METHOD 3 and INITMAN. As we explained, these methods do not suffer from inaccuracies introduced

Table 5: Two-norm of the error in the solution of the constrained runs functional iteration and INITMAN, when $\sigma$ is the observable.

|               | $\lambda = 25$          | $\lambda = 5$           |
| ------------- | ----------------------- | ----------------------- |
| CRFI, $m = 0$ | $\infty$                | $\infty$                |
| CRFI, $m = 1$ | $\infty$                | $\infty$                |
| CRFI, $m \geq 2$ | $\infty$             | $\infty$                |
| INITMAN       | $4.51 \cdot 10^{-14}$   | $1.12 \cdot 10^{-13}$   |

by arbitrary lifting or from instabilities which sometimes accompany the constrained runs functional iteration. In some cases, however, numerical difficulties may also be encountered when applying these methods. Let us illustrate this for METHOD 3 using $\rho$ as the observable. As before, we choose $D = 1$, $\Delta x = 1/N = 1/100$, $\omega = 1.25$ and $\lambda = 25$, use the arbitrary lifting scheme $\rho \mapsto (\rho, 0, 0)$, set $\tau' = \Delta t$, use $\rho^0 = \sin(\pi x)$ as the initial condition for Newton's method, and explicitly eliminate the boundary conditions $\rho(0) = \rho(1) = 0$. The value of $\varepsilon$, however, will now be set equal to $\varepsilon = \sqrt{\eta}$ (with $\eta$ the value of machine precision) instead of to $\varepsilon = \sqrt[4]{\eta}$, and we will vary $\tau$ from 0 to $25\Delta t$.

The results are summarized in Table 6. As expected, the error (compared to the exact solution $\rho^*$) decreases as the value of $\tau/\Delta t$ increases because larger $\tau$ values bring the fine-scale simulator closer to the slow manifold. However, except for very small values of $\tau/\Delta t$, the condition numbers of the Jacobian matrices encountered within Newton's method, $\kappa_i = ||\partial g/\partial x(x_i)|| \cdot ||(\partial g/\partial x(x_i))^{-1}||$, also tend to increase. This can be understood by realizing that some components of the observable $\rho$ are also decaying (at a fast pace, albeit slower than that of the remaining fine-scale variables), so that more and more relative indeterminacy is introduced as the value of $\tau/\Delta t$ increases. In other words, if we perturb the exact solution in the direction of a relatively quickly decaying coarse observable, the nonlinear residual will remain small as this observable will largely be damped out by the time we reach the slow manifold. Since the norm of the Jacobian matrix itself remains nearly constant, it is the norm of the inverse of the Jacobian matrix that increases along with the condition number.

As soon as the condition number and norm of the inverse of the Jacobian reach values larger than $10^8$, we observe that Newton's method no longer converges. This can be explained as follows. If $\Phi(x, \tau)$ and $\Phi(x, \tau + \tau')$ are both of $\mathcal{O}(1)$, the absolute error in $g(x)$ above (remember, we are solving $g(x) = 0$) is of $\mathcal{O}(\eta)$. Due to round-off and truncation error in the finite difference approximation of the Jacobian, the absolute error in the elements of the Jacobian matrices $\partial g/\partial x(x_i)$ is then of $\mathcal{O}(\eta/\varepsilon + \varepsilon) = \mathcal{O}(10^{-8})$. To see the influence of this Jacobian matrix perturbation, we can write

$$(\partial g/\partial x(x_i) + \delta g)(\Delta x + \delta x) = -g(x_i), \tag{18}$$

in which $\delta g$ represents the perturbation matrix (with elements of $\mathcal{O}(10^{-8})$) and $\delta x$ repre-

Table 6: As a function of $\tau/\Delta t$, this table gives the number of Newton iteration steps required to reach the tolerance $\mathtt{tol} = 10^{-14}$, the two-norm of the nonlinear residual $||\Phi(\overline{\rho}_3, \tau + \tau') - \Phi(\overline{\rho}_3, \tau)||_2$, the two-norm of the error $||\Phi(\overline{\rho}_3, \tau) - \rho^*||_2$, the maximal value of the condition number $\kappa_i$, and the maximal norm of the inverse of the Jacobian $||(\partial g/\partial x(x_i))^{-1}||$ encountered during the Newton iteration.

| $\tau/\Delta t$ | # iters. | residual | error | $\max_i \kappa_i$ | $\max_i ||(\partial g/\partial x(x_i))^{-1}||$ |
|---|---|---|---|---|---|
| 1 | 4 | 1.44e-015 | 1.73e+000 | 2.62e+003 | 3.77e+003 |
| 2 | 6 | 7.05e-016 | 5.89e-001 | 5.25e+002 | 3.77e+003 |
| 3 | 5 | 7.16e-016 | 1.94e-001 | 7.34e+002 | 3.24e+003 |
| 4 | 5 | 1.03e-015 | 5.95e-002 | 2.37e+002 | 3.39e+003 |
| 5 | 5 | 1.59e-015 | 1.77e-002 | 1.98e+002 | 3.38e+003 |
| 6 | 5 | 9.36e-016 | 5.12e-003 | 1.70e+002 | 3.36e+003 |
| 7 | 5 | 1.41e-015 | 1.45e-003 | 4.73e+002 | 1.07e+004 |
| 8 | 5 | 1.30e-015 | 4.07e-004 | 1.42e+003 | 3.59e+004 |
| 9 | 5 | 1.37e-015 | 1.12e-004 | 1.53e+002 | 4.26e+003 |
| 10 | 5 | 5.98e-015 | 3.08e-005 | 8.18e+003 | 2.51e+005 |
| 11 | 6 | 6.64e-016 | 8.37e-006 | 1.29e+004 | 4.29e+005 |
| 12 | 5 | 5.25e-015 | 2.26e-006 | 1.88e+003 | 6.78e+004 |
| 13 | 6 | 2.75e-015 | 6.06e-007 | 2.10e+005 | 8.15e+006 |
| 14 | 6 | 1.18e-015 | 1.62e-007 | 1.33e+005 | 5.53e+006 |
| 15 | 6 | 1.16e-015 | 4.30e-008 | 2.47e+004 | 1.09e+006 |
| 16 | 8 | 2.19e-015 | 1.14e-008 | 5.15e+005 | 2.41e+007 |
| 17 | 10 | 4.24e-015 | 3.01e-009 | 1.85e+006 | 9.21e+007 |
| 18 | 10 | 2.10e-015 | 7.89e-010 | 3.82e+005 | 2.00e+007 |
| 19 | 9 | NaN | NaN | 1.51e+008 | 4.12e+009 |
| 20 | 3 | NaN | NaN | 6.12e+008 | 3.54e+010 |
| 21 | 6 | NaN | NaN | 1.19e+008 | 7.20e+009 |
| 22 | 6 | NaN | NaN | 1.22e+009 | 7.69e+010 |
| 23 | 5 | NaN | NaN | 8.19e+009 | 3.80e+010 |
| 24 | 3 | NaN | NaN | 2.11e+008 | 1.45e+010 |
| 25 | 3 | NaN | NaN | 2.00e+009 | 1.43e+011 |

sents the resulting perturbation on $\Delta x$. If we neglect the $\delta g \delta x$ term, we find that

$$\delta x \approx -(\partial g/\partial x(x_i))^{-1} \cdot \delta g \cdot \Delta x. \tag{19}$$

It is well known that our (inexact) Newton method converges if

$$\frac{||\partial g/\partial x(x_i)(\Delta x + \delta x) + g(x_i)||}{||g(x_i)||} < 1, \tag{20}$$

at least if we start sufficiently close to the solution (Dembo et al. (1982)). Using (19), this becomes

$$\frac{||\partial g/\partial x(x_i)(\Delta x - (\partial g/\partial x(x_i))^{-1} \cdot \delta g \cdot \Delta x) + g(x_i)||}{||g(x_i)||} = \frac{||\delta g \cdot \Delta x||}{||g(x_i)||} \lesssim 1. \tag{21}$$

If $|| \cdot ||$ denotes the Euclidean vector norm or its induced matrix norm (i.e., the spectral norm), it holds that

$$\begin{aligned}
\frac{||\delta g \cdot \Delta x||}{||g(x_i)||} &= C \cdot \frac{||\delta g|| \cdot ||\Delta x||}{||g(x_i)||} \\
&= C \cdot D \cdot ||\delta g|| \cdot ||(\partial g/\partial x(x_i))^{-1}|| \\
&= C \cdot D \cdot \kappa_i \cdot \frac{||\delta g||}{||\partial g/\partial x(x_i)||},
\end{aligned} \tag{22}$$

with $C, D \in [0,1]$. (Here, we also used the fact that $\Delta x = -(\partial g/\partial x(x_i))^{-1} \cdot g(x_i) \Rightarrow ||\Delta x|| = D \cdot ||(\partial g/\partial x(x_i))^{-1}|| \cdot ||g(x_i)||$.) Since the values $C$ and $D$ are in practice often approximately equal to 1, it follows that Newton's method converges if

$$||\delta g|| \cdot ||(\partial g/\partial x(x_i))^{-1}|| = \kappa \frac{||\delta g||}{||\partial g/\partial x(x_i)||} \lesssim 1, \tag{23}$$

at least if we start sufficiently close to the solution. As $||\delta g|| = \mathcal{O}(10^{-8})$, this implies that Newton's method is expected to converge if $||(\partial g/\partial x(x_i))^{-1}|| < \mathcal{O}(10^8)$. If $||(\partial g/\partial x(x_i))^{-1}|| > \mathcal{O}(10^8)$, the method is expected to diverge. These theoretical results are clearly confirmed in Table 6.

Note that for our LBM model problem, the error in the solution can be made as small as $\mathcal{O}(10^{-10})$ by using the largest value of $\tau/\Delta t$ for which Newton's method converges. If the desired level of accuracy cannot be reached due to numerical difficulties, one may try the following "trick": increase the value of $\varepsilon$, as we did in Section 4.3 when we set $\varepsilon = \sqrt[4]{\eta}$ instead of $\varepsilon = \sqrt{\eta}$, in an attempt to reduce the error of the finite difference approximation. Remember that the finite difference error consists of round-off error and truncation error so that the total finite difference error is of $\mathcal{O}(\eta/\varepsilon + \varepsilon)$. If the problem is only mildly nonlinear, as in the case of our LBM due to the fact that $\Delta t$ is small, the constant in the $\mathcal{O}(\varepsilon)$ term is small and the finite difference error can be decreased by choosing $\varepsilon > \sqrt{\eta}$.

## 6. Conclusions

In this paper, we have introduced an approach for the computation of coarse-scale steady state solutions as well as an approach for initialization on a slow manifold. These methods were compared favorably to previously suggested ones: METHOD 3 and INITMAN are quick, accurate, and robust, and they bear a striking similarity to each other. We demonstrated the use of each of these methods on a lattice Boltzmann model for a reaction-diffusion system, compared them to previously suggested methods, and verified our error predictions with both numerical results and numerical analysis. These new procedures circumvent the need for long, fine-scale simulations to find coarse-scale steady states, or to appropriately initialize the fine-scale simulator.

Our implementation of the numerical methods in this report has been simple and direct, in order to clearly illustrate the methods and analyze their sources of error. Indeed, in real-world applications, components like the use of higher order derivatives, the reusing of data (for example, for two nearby sets of observables, the corresponding fine-scale initializations are probably similar), or more intelligent Newton steps may clearly be used to improve performance.

## 7. Acknowledgments

## References

Chopard, B., Dupuis, A., Masselot, A., Luthi, P., 2002. Cellular automata and lattice Boltzmann techniques: An approach to model and simulate complex systems. Advances in complex systems 5, 103–246.

Curry, J.H., Haupt, S.E., Limber, M.N., 2002. Low-order models, initialization, and the slow manifold. Tellus A 47, 145–161.

Danilov, V.G., Maslov, V.P., Volosov, K.A., 1995. Mathematical modelling of heat and mass transfer processes. Kluwer Academic Pub.

Dawson, S.P., Chen, S., Doolen, G.D., 1993. Lattice Boltzmann computations for reaction-diffusion equations. The Journal of Chemical Physics 98, 1514–1523.

Dembo, R.S., Eisenstat, S.C., Steihaug, T., 1982. Inexact newton methods. SIAM Journal on Numerical analysis 19, 400–408.

Gear, C.W., Kaper, T.J., Kevrekidis, I.G., Zagaris, A., 2005. Projecting to a slow manifold: Singularly perturbed systems and legacy codes. SIAM Journal on Applied Dynamical Systems 4, 711–732.

Gear, C.W., Kevrekidis, I.G., 2005. Constraint-defined manifolds: a legacy code approach to low-dimensional computation. Journal of Scientific Computing 25, 17–28.

Holmes, E.E., Lewis, M.A., Banks, J.E., Veit, R.R., 1994. Partial differential equations in ecology: spatial interactions and population dynamics. Ecology 75, 17–29.

Jolly, M.S., Kevrekidis, I.G., Titi, E.S., 1990. Approximate inertial manifolds for the Kuramoto-Sivashinsky equation: analysis and computations. Physica D Nonlinear Phenomena 44, 38–60.

Kelley, C.T., 1995. Iterative methods for linear and nonlinear equations. Society for Industrial Mathematics.

Kevrekidis, I.G., Gear, C.W., Hyman, J.M., Kevrekidis, P.G., Runborg, O., Theodoropoulos, C., 2003. Equation-Free Multiscale Computation: enabling microscopic simulators to perform system-level tasks. Communications in Mathematical Sciences 4, 715–762.

Kreiss, H.O., Lorenz, J., 1994. On the existence of slow manifolds for problems with different timescales. Philosophical Transactions: Physical Sciences and Engineering 346, 159–171.

Lam, S.H., Goussis, D.A., 1994. The CSP method for simplifying kinetics. International Journal of Chemical Kinetics 26, 461–486.

Lorenz, E.N., 1986. On the existence of a slow manifold. Journal of the Atmospheric Sciences 43, 1547–1558.

Maas, U., Pope, S.B., 1992. Simplifying chemical kinetics- Intrinsic low-dimensional manifolds in composition space. Combustion and Flame 88, 239–264.

Qian, Y.H., Orszag, S.A., 1995. Scalings in diffusion-driven reaction $A + B \rightarrow C$: Numerical simulations by lattice BGK models. Journal of Statistical Physics 81, 237–253.

Ramshaw, J.D., 1980. Partial chemical equilibrium in fluid dynamics. Physics of Fluids 23, 675.

Segel, L.A., Slemrod, M., 1989. The quasi-steady-state assumption: a case study in perturbation. SIAM Review 31, 446–477.

Succi, S., 2001. The lattice Boltzmann equation for fluid dynamics and beyond. Oxford University Press, USA.

Theodoropoulos, C., Qian, Y.H., Kevrekidis, I.G., 2000. "Coarse" stability and bifurcation analysis using time-steppers: A reaction-diffusion example. Proceedings of the National Academy of Sciences 97, 9840–9843.

Van Leemput, P., Vanroose, W., Roose, D., 2008. Mesoscale analysis of the equation-free constrained runs initialization scheme. Multiscale Modeling and Simulation 6, 1234–1255.

Vandekerckhove, C., 2008. Macroscopic Simulation of Multiscale Systems within the Equation-Free Framework. Phd thesis. Dept. of Computer Science, Faculty of Engineering, K.U. Leuven. ISBN 978-90-5682-947-6.

Vandekerckhove, C., Kevrekidis, I.G., Roose, D., 2009. An efficient Newton-Krylov implementation of the constrained runs scheme for initializing on a slow manifold. Journal of Scientific Computing 39, 167–188.

Vandekerckhove, C., Van Leemput, P., Roose, D., 2008. Accuracy and Stability of the Coarse Time-Stepper for a Lattice Boltzmann Model. Journal of Algorithms and Computational Technology 2, 249–273.

Zagaris, A., Kaper, H.G., Kaper, T.J., 2004. Analysis of the computational singular perturbation reduction method for chemical kinetics. Journal of Nonlinear Science 14, 59–91.