Accuracy and Stability of the Coarse Time-Stepper for a Lattice Boltzmann Model

Christophe Vandekerckhove⋆

In collaboration with:
Pieter Van Leemput⋆
Dirk Roose⋆
Yannis Kevrekidis†

⋆ Department of Computer Science
Katholieke Universiteit Leuven, Belgium

† Department of Chemical Engineering and PACM
Princeton University, USA

August 29, 2007

Outline

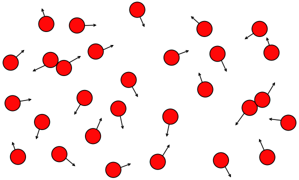
Outline

Outline

Microscopic versus macroscopic modeling

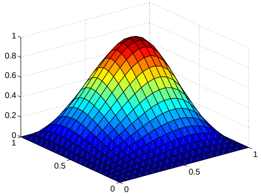
Microscopic modeling

- Example: particle model for the evolution of positions and velocities of particles
- Detailed spatial/temporal behavior
- Computationally expensive $\longrightarrow$ limited to small spatio-temporal domains



Macroscopic modeling

- Example: PDE for density of particles
- Only smooth averaged macroscopic behavior
- Computationally more tractable
- Can be studied using standard numerical tools



Analytical coarse-graining

- Micro-model $\longrightarrow$ macro-model under certain simplifying assumptions

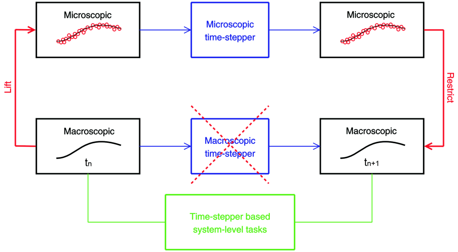
Equation-free computing

Setting

- Microscopic simulator available; we observe smooth macroscopic behavior. . .
- . . . but we fail to derive the macroscopic model (although it exists conceptually)

Equation-free computing (Kevrekidis et al, 2000—)

- Perform macroscopic tasks anyway!
- Main tool: the coarse time-stepper
  - Approximate time integrator for unavailable macroscopic model
  - Each step consists of 3 substeps:
    1) *Lifting*: initialize micro-simulator according to given macro-field
    2) Micro-*simulation* over time Δt
    3) *Restriction*: extract macro-fields
  - Relies on a separation of time-scales
  - To increase efficiency: use as "input" for time-stepper based system-level tasks (time-integration, bifurcation analysis, control,. . .)



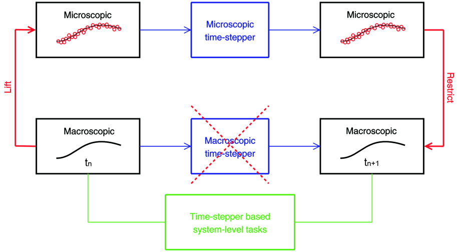
Equation-free computing

Setting

- Microscopic simulator available; we observe smooth macroscopic behavior...
- ...but we fail to derive the macroscopic model (although it exists conceptually)

Equation-free computing (Kevrekidis et al, 2000—)

- Perform macroscopic tasks anyway!
- Main tool: the coarse time-stepper
  - Approximate time integrator for unavailable macroscopic model
  - Each step consists of 3 substeps:
    1) *Lifting*: initialize micro-simulator according to given macro-field
    2) Micro-*simulation* over time Δt
    3) *Restriction*: extract macro-fields



  - Relies on a separation of time-scales
  - To increase efficiency: use as "input" for time-stepper based system-level tasks (time-integration, bifurcation analysis, control,...)

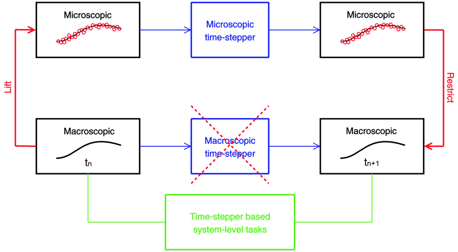
Equation-free computing

Setting

- Microscopic simulator available; we observe smooth macroscopic behavior...
- ...but we fail to derive the macroscopic model (although it exists conceptually)

Equation-free computing (Kevrekidis et al, 2000—)

- Perform macroscopic tasks anyway!
- Main tool: the coarse time-stepper
  - Approximate time integrator for unavailable macroscopic model
  - Each step consists of 3 substeps:
    1) *Lifting*: initialize micro-simulator according to given macro-field
    2) Micro-*simulation* over time Δt
    3) *Restriction*: extract macro-fields
  - Relies on a separation of time-scales
  - To increase efficiency: use as "input" for time-stepper based system-level tasks (time-integration, bifurcation analysis, control,...)



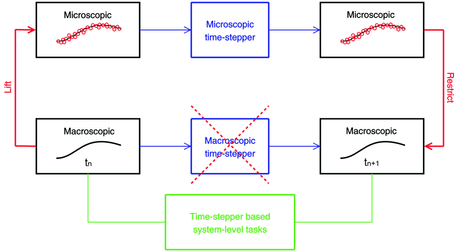
Equation-free computing

Setting

- Microscopic simulator available; we observe smooth macroscopic behavior...
- ...but we fail to derive the macroscopic model (although it exists conceptually)

Equation-free computing (Kevrekidis et al, 2000—)

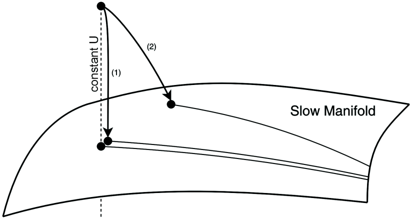
- Perform macroscopic tasks anyway!
- Main tool: the coarse time-stepper
  - Approximate time integrator for unavailable macroscopic model
  - Each step consists of 3 substeps:
    1) *Lifting*: initialize micro-simulator according to given macro-field
    2) Micro-*simulation* over time Δt
    3) *Restriction*: extract macro-fields



  - Relies on a separation of time-scales
  - To increase efficiency: use as “input” for time-stepper based system-level tasks (time-integration, bifurcation analysis, control,...)

Lifting: the hardest part

- Appropriate initialization of the microscopic state (U, V), according to the macroscopic variable U (V:"higher order moments")
- Nontrivial one-to-many mapping: $U \rightarrow (U, V)$
- If a macroscopic equation in terms of only U indeed exists, the higher order moments V quickly become functionals of U: slaving relations V=F(U)
- The slaving relations define a "slow manifold" in microscopic phase space, on which the macroscopic dynamics take place



- Fast attraction towards the manifold does not imply that the CTS computes a correct macroscopic trajectory (U may change)!
- Good lifting (close to the slow manifold) is important!

Goal of the talk

In this talk, we will study the numerical properties of different aspects of equation-free computing when the microscopic simulator is a lattice Boltzmann model

- Good caricature of realistic multiscale problems
- Simple enough to do some mathematical analysis (deterministic, well-known theoretical multiscale expansion)

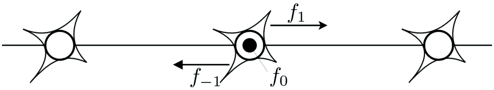
Outline

Model problem: 1D reaction-diffusion LBM

Available “microscopic” model: lattice Boltzmann model

- Simplified kinetic model
- Discrete in space x, time t and crudely discretized in velocity v
- Tracks particle distribution functions $f_{-1}(x_j, t_k)$, $f_0(x_j, t_k)$ and $f_1(x_j, t_k)$
- Macroscopic density of particles: $\rho = \sum_{i=-1}^{1} f_i$
- LBM evolution law:

$$
\begin{aligned}
f_i(x_j + v_i \Delta t, t_k + \Delta t) - f_i(x_j, t_k) = \\
- \omega\left(f_i(x_j, t_k) - \frac{1}{3}\rho(x_j, t_k)\right) + \lambda \frac{\Delta t}{3} \rho(x_j, t_k)(1 - \rho(x_j, t_k))
\end{aligned}
$$



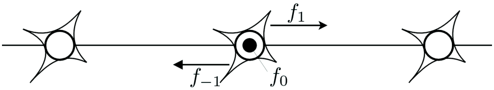
- Diffusive BGK collisions: f_i's relax to local diffusive equilibrium $f_i^{eq} = \rho/3$ with relaxation coefficient $\omega \in (0, 2)$
- Nonlinear reactions: depend on λ and density ρ

Model problem: 1D reaction-diffusion LBM

Available “microscopic” model: lattice Boltzmann model

- Simplified kinetic model
- Discrete in space x, time t and crudely discretized in velocity v
- Tracks particle distribution functions $f_{-1}(x_j, t_k)$, $f_0(x_j, t_k)$ and $f_1(x_j, t_k)$
- Macroscopic density of particles: $\rho = \sum_{i=-1}^{1} f_i$
- LBM evolution law:

$$f_i(x_j + v_i\Delta t, t_k + \Delta t) - f_i(x_j, t_k) = - \omega\left(f_i(x_j, t_k) - \frac{1}{3}\rho(x_j, t_k)\right) + \lambda\frac{\Delta t}{3}\rho(x_j, t_k)(1 - \rho(x_j, t_k))$$



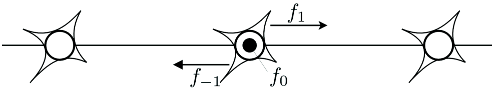
- Diffusive BGK collisions: f_i's relax to local diffusive equilibrium $f_i^{eq} = \rho/3$ with relaxation coefficient $\omega \in (0, 2)$
- Nonlinear reactions: depend on λ and density ρ

Model problem: 1D reaction-diffusion LBM

Available “microscopic” model: lattice Boltzmann model

- Simplified kinetic model
- Discrete in space x, time t and crudely discretized in velocity v
- Tracks particle distribution functions $f_{-1}(x_j, t_k)$, $f_0(x_j, t_k)$ and $f_1(x_j, t_k)$
- Macroscopic density of particles: $\rho = \sum_{i=-1}^{1} f_i$
- LBM evolution law:

$$f_i(x_j + v_i \Delta t, t_k + \Delta t) - f_i(x_j, t_k) = \\ -\omega(f_i(x_j, t_k) - \frac{1}{3}\rho(x_j, t_k)) + \lambda \frac{\Delta t}{3} \rho(x_j, t_k)(1 - \rho(x_j, t_k))$$



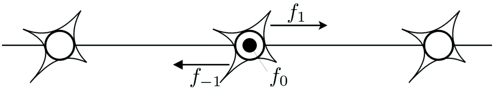
- Diffusive BGK collisions: f_i's relax to local diffusive equilibrium $f_i^{eq} = \rho/3$ with relaxation coefficient $\omega \in (0, 2)$
- Nonlinear reactions: depend on λ and density ρ

Model problem: 1D reaction-diffusion LBM

Available "microscopic" model: lattice Boltzmann model

- Simplified kinetic model
- Discrete in space x, time t and crudely discretized in velocity v
- Tracks particle distribution functions $f_{-1}(x_j, t_k)$, $f_0(x_j, t_k)$ and $f_1(x_j, t_k)$
- Macroscopic density of particles: $\rho = \sum_{i=-1}^{1} f_i$
- LBM evolution law:

$$f_i(x_j + v_i\Delta t, t_k + \Delta t) - f_i(x_j, t_k) = \\ -\omega(f_i(x_j, t_k) - \frac{1}{3}\rho(x_j, t_k)) + \lambda\frac{\Delta t}{3}\rho(x_j, t_k)(1 - \rho(x_j, t_k))$$



- Diffusive BGK collisions: f_i's relax to local diffusive equilibrium $f_i^{eq} = \rho/3$ with relaxation coefficient $\omega \in (0, 2)$
- Nonlinear reactions: depend on λ and density ρ

Lifting and restriction for the LBM

- 1-to-1 correspondence between f_{-1}, f_0, f_1 and velocity moments of the particle distribution functions

  $\rho = \sum_{i=-1}^{1} f_i$ (density)
  $\phi = \sum_{i=-1}^{1} i \cdot f_i$ (momentum)
  $\xi = \frac{1}{2} \sum_{i=-1}^{1} i^2 \cdot f_i$ (energy)

- From Chapman-Enskog multiscale expansion, we can derive
  1) Long-term behavior of LBM: Fisher equation
  $$\frac{\partial \rho}{\partial t} = \left(\frac{2-\omega}{3\omega} \frac{\Delta x^2}{\Delta t} \right) \frac{\partial^2 \rho}{\partial x^2} + \lambda \rho (1-\rho) \quad \Rightarrow \quad U = \rho,\ V = (\phi, \xi)$$
  2) The slaving relations are
  $$\phi = -\frac{2}{3\omega} \frac{\partial \rho}{\partial x} \Delta x + \mathcal{O}(\Delta x^3), \quad \xi = \frac{1}{3}\rho - \frac{\omega - 2}{18\omega^2} \frac{\partial^2 \rho}{\partial x^2} \Delta x^2 + \mathcal{O}(\Delta x^4)$$
- Ideally, we would like to lift with these slaving relations
- In practice: unavailable → numerical alternative: constrained runs
- First however, we study the accuracy and the stability of the coarse time-stepper when lifting with the slaving relations

Lifting and restriction for the LBM

- 1-to-1 correspondence between f_{-1}, f_0, f_1 and velocity moments of the particle distribution functions

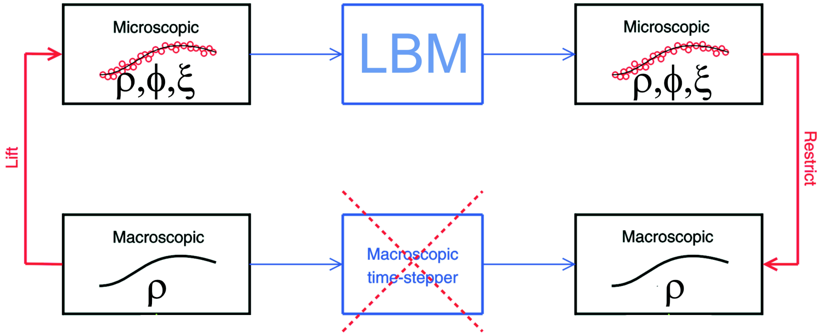
$$\begin{aligned} \rho &= \textstyle\sum_{i=-1}^{1} f_i \text{ (density)} \\ \phi &= \textstyle\sum_{i=-1}^{1} i \cdot f_i \text{ (momentum)} \\ \xi &= \tfrac{1}{2}\textstyle\sum_{i=-1}^{1} i^2 \cdot f_i \text{ (energy)} \end{aligned} \quad \underset{\text{Restrict}}{\overset{\text{Lift}}{\rightleftarrows}} \quad \rho \text{ (macro)}$$

- From Chapman-Enskog multiscale expansion, we can derive
  1) Long-term behavior of LBM: Fisher equation
  $$\frac{\partial \rho}{\partial t} = \left(\frac{2-\omega}{3\omega} \frac{\Delta x^2}{\Delta t} \right) \frac{\partial^2 \rho}{\partial x^2} + \lambda\rho(1-\rho) \quad \Rightarrow \quad U = \rho,\ V = (\phi, \xi)$$
  2) The slaving relations are
  $$\phi = -\frac{2}{3\omega}\frac{\partial \rho}{\partial x}\Delta x + \mathcal{O}(\Delta x^3), \quad \xi = \frac{1}{3}\rho - \frac{\omega - 2}{18\omega^2}\frac{\partial^2 \rho}{\partial x^2}\Delta x^2 + \mathcal{O}(\Delta x^4)$$
- Ideally, we would like to lift with these slaving relations
- In practice: unavailable $\rightarrow$ numerical alternative: constrained runs
- First however, we study the accuracy and the stability of the coarse time-stepper when lifting with the slaving relations

Outline

The coarse time-stepper

Lifting with the slaving relations



- Lifting: appropriate discretization of truncated slaving relations (up to order p)

$$\phi(x,t) = \sum_{p=0}^{\infty} \phi_p(x,t)\Delta x^p = -\frac{2}{3\omega}\frac{\partial \rho(x,t)}{\partial x}\Delta x + \frac{\omega^2-2\omega+2}{9\omega^3}\frac{\partial^3 \rho(x,t)}{\partial x^3}\Delta x^3 + \dots,$$

$$\xi(x,t) = \sum_{p=0}^{\infty} \xi_p(x,t)\Delta x^p = \frac{1}{3}\rho(x,t) - \frac{\omega-2}{18\omega^2}\frac{\partial^2 \rho(x,t)}{\partial x^2}\Delta x^2 + \dots.$$

- Simulation: 1 LBM step
- Restriction: return ρ

Accuracy and stability of the coarse time-stepper

Pure diffusion with $D=1$: $\dfrac{\partial \rho(x,t)}{\partial t} = \dfrac{\partial^2 \rho(x,t)}{\partial x^2}, \quad \rho(0,t)=\rho(1,t)=0.$

- CTS with $p=0$: lifting with equilibrium distributions $f_i = f_i^{eq} = \rho/3$
  - $\rho_{n+1} = A\rho_n$, with $A = [\ldots \quad 0 \quad 1/3 \quad 1/3 \quad 1/3 \quad 0 \quad \ldots]$
  - Truncation error: $\overline{T}(x,t) = \dfrac{\partial \rho}{\partial t} - \dfrac{\Delta x^2}{3\Delta t}\dfrac{\partial^2 \rho}{\partial x^2} = \dfrac{\partial \rho}{\partial t} - \dfrac{\omega}{2-\omega}\dfrac{\partial^2 \rho}{\partial x^2}$
    - Unless if $\omega = 1$, the computed trajectory is the solution of modified equation (diffusion with different D)
  - Stability interval: $\omega \in (0,2)$
- CTS with $p=1$
  - $A = \left[\ldots \quad 0 \quad \dfrac{1-\omega}{6\omega} \quad 1/3 \quad \dfrac{2\omega-1}{3\omega} \quad 1/3 \quad \dfrac{1-\omega}{6\omega} \quad 0 \ldots\right]$
  - Truncation error: $\overline{T}(x,t) = \dfrac{1}{2}\dfrac{\partial^2 \rho}{\partial t^2}\Delta t - \dfrac{1}{3}\dfrac{\partial^4 \rho}{\partial x^4}\Delta x^2 + \dfrac{1}{12}\dfrac{\partial^4 \rho}{\partial x^4}\dfrac{\Delta x^4}{\Delta t}$
    - First-order accurate in time and second-order accurate in space if $\Delta t = \mathcal{O}(\Delta x^2)$ (diffusive scaling)
  - Stability interval: $\omega \in (0.349, 2)$
- CTS with $p=2$: $\quad\ldots$

Comparison to traditional explicit FD scheme for PDE

Traditional explicit FD: $\rho_j^{n+1} = \rho_j^n + D\frac{\Delta t}{\Delta x^2}\left(\rho_{j+1}^n - 2\rho_j^n + \rho_{j-1}^n\right)$

- FD for PDE:
  - Truncation error: $\overline{T}(x,t) = \dfrac{1}{2}\dfrac{\partial^2\rho}{\partial t^2}\Delta t - \dfrac{1}{12}\dfrac{\partial^4\rho}{\partial x^4}\Delta x^2$
  - Stability interval: $\Delta t < 0.5\Delta x^2$
- CTS with $p = 1$:
  - Truncation error: $\overline{T}(x,t) = \dfrac{1}{2}\dfrac{\partial^2\rho}{\partial t^2}\Delta t - \dfrac{1}{3}\dfrac{\partial^4\rho}{\partial x^4}\Delta x^2 + \dfrac{1}{12}\dfrac{\partial^4\rho}{\partial x^4}\dfrac{\Delta x^4}{\Delta t}$
  - Stability interval: $\omega \in (0.349, 2) \Leftrightarrow \Delta t < 1.577\Delta x^2$

The influence of increasing the number of LBM steps M

- Larger M: allows off-manifold initial condition to get attracted to the slow manifold (fast process) $\rightarrow$ improve accuracy
- Density may change: not necessarily the correct trajectory on the SM

CTS with $p = 0$

- Truncation error:

$$\overline{T}(x,t) = \frac{\partial \rho}{\partial t} - \left(1 + \underbrace{\frac{2}{M}}_{\text{slow}} \frac{\omega - 1}{\omega(\omega - 2)}\left(\underbrace{-1 + (1-\omega)^M}_{\text{fast}}\right)\right)\frac{\partial^2 \rho}{\partial x^2}$$

- The accuracy improves when M is increased
- M should be very large to obtain accurate results! Efficiency?!

CTS with $p = 1$

- Stability interval $(\omega_{\min}, 2)$:

	M=1	M=2	M=3	M=4	M=5	M=6	M=8	M=10
$\omega_{\min}$	0.349	0.310	0.311	0.268	0.231	0.227	0.198	0.177

- The stability improves when M is increased

Outline

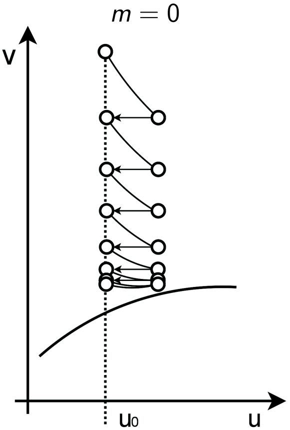
Outline

Constrained runs [Gear & Kevrekidis (2005)]: basic idea

- Goal: find V corresponding to U such that (U, V) is close to the slow manifold, without using the slaving relations
- Class of CR schemes; m-th scheme computes V so that $\dfrac{\mathrm{d}^{m+1}V}{\mathrm{d}t^{m+1}} = 0$
- (U, V) is then m-th order approximation of the desired state on the slow manifold [Gear, Kaper, Kevrekidis, Zagaris (2005)]
- Only microscopic simulator available $\rightarrow$ approximate $\dfrac{\mathrm{d}^{m+1}V}{\mathrm{d}t^{m+1}}$
- Solve the resulting forward difference equation with functional iteration $\rightarrow$ constrained runs functional iteration

Constrained runs functional iteration: interpretation

- When $m = 0$, the scheme repeatedly
  - integrates over a short time interval
  - resets U (in order to "constrain" the macroscopic variable)
- In general: V is updated using an m-th degree interpolant for V

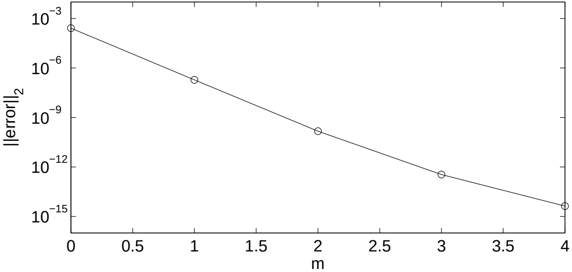


Analysis (accuracy/convergence) of constrained runs FI

- For slow-fast systems: [Gear, Kaper, Kevrekidis, Zagaris (2005)]
- For 1D-RD-LBM ($m = 0$): [Van Leemput, Vanroose, Roose (2005)]
  - Constrained runs FI is stable for all $\omega \in (0, 2)$
  - Converges to a good approximation ($\mathcal{O}(1)$ and $\mathcal{O}(\Delta x)$ terms of the slaving relations are correct)
  - Asymptotic convergence factor $|1 - \omega|$ (again!)

Constrained runs FI for the LBM ($m \geq 0$)

- For the LBM ($\lambda = 0$): accuracy increases as m increases



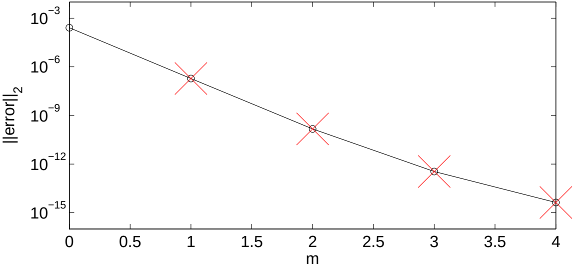
  - Compare to an "exact" slaved state from long LBM simulation
  - $m = 1$: 2 extra terms of the slaving relations correct (up to $\mathcal{O}(\Delta x^3)$)
- If $m > 0$, the fixed point iteration may however be unstable
- For the LBM ($\lambda = 0$): stability interval $(\omega_{\min}, \omega_{\max})$

m	0	1	2	3	4
$\omega_{\min}$	0.000	0.690	0.865	0.929	0.959
$\omega_{\max}$(!)	2.000	1.291	1.133	1.072	1.043

$\rightarrow$ arbitrary slow convergence or divergence

Constrained runs FI for the LBM ($m \geq 0$)

- For the LBM ($\lambda = 0$): accuracy increases as m increases



  - Compare to an "exact" slaved state from long LBM simulation
  - $m = 1$: 2 extra terms of the slaving relations correct (up to $\mathcal{O}(\Delta x^3)$)
- If $m > 0$, the fixed point iteration may however be unstable
- For the LBM ($\lambda = 0$): stability interval $(\omega_{\min}, \omega_{\max})$

m	0	1	2	3	4
$\omega_{\min}$	0.000	0.690	0.865	0.929	0.959
$\omega_{\max}$(!)	2.000	1.291	1.133	1.072	1.043

$\rightarrow$ arbitrary slow convergence or divergence

Outline

Newton-Krylov constrained runs: basic idea

- Replace functional iteration with Newton-Krylov solver
- FI: $V^{k+1} = \mathcal{C}(U; V^k)$ (U parameter, V unknown)
- Solve $g(U; V) := V - \mathcal{C}(U; V) = 0$ with Newton's method:

$$V_{k+1} = V_k + \delta V_k$$
$$\frac{\partial g}{\partial V}(U; V_k) \cdot \delta V_k = \left(I - \frac{\partial \mathcal{C}}{\partial V}(U; V_k)\right) \cdot \delta V_k = -g(U; V_k)$$

- Only microscopic simulator available $\rightarrow$ linearization of g or C not available
- Estimate matrix-vector product

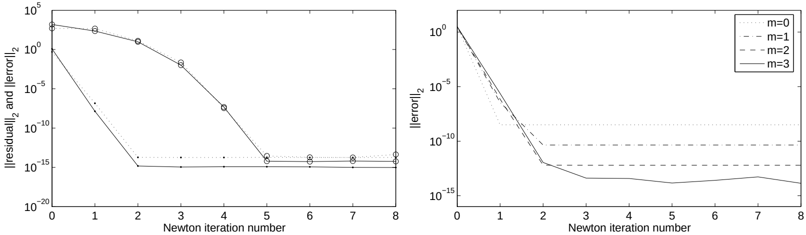
$$\left(I - \frac{\partial \mathcal{C}}{\partial V}(U; V_k)\right) \cdot \delta V_k \approx \delta V_k - \frac{\mathcal{C}(U; V_k + \epsilon \delta V_k) - \mathcal{C}(U; V_k)}{\epsilon}$$

  with

$$\epsilon = \sqrt{\bar{\epsilon}}||V_k||/||\delta V_k|| \quad \text{if } \delta V_k \neq 0, \; V_k \neq 0$$

- Matvec available $\rightarrow$ solve linear subsystems with Krylov method

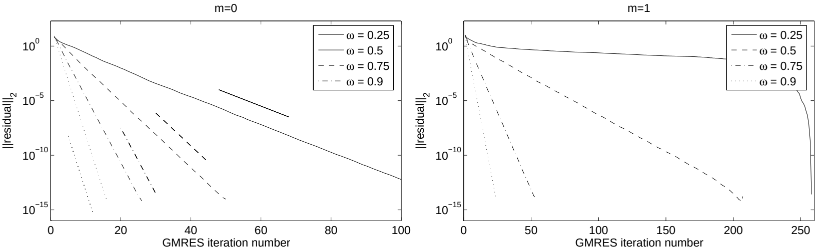
Illustration of the Newton iteration



- **Left**: norm of nonlinear residual / error (again: compare to "exact" slaved state from a long LBM simulation)
- $m = 3, \omega = 1.25$ (FI unstable!), $\lambda = 0$ (linear) or 1000 (nonlinear)
- $\lambda = 0$: 2 Newton steps required (accuracy matvec: 10^{-8})
- $\lambda = 1000$: small number of steps if irregular initial guess (quadratic convergence)
- **Right**: norm of error when $\lambda = 1000$, zero initial guess, various m. Only 2 or 3 steps needed. If $m < 3$: error levels off earlier.

Solving the linear subsystems $Ax = b$ with GMRES

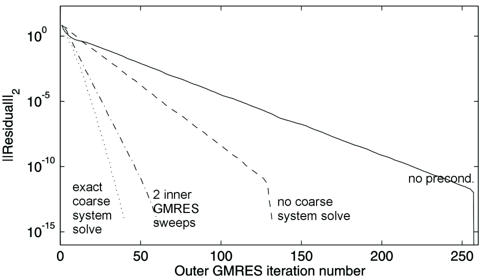
- GMRES: approximates $x^* = A^{-1}b$ by $x_n \in \mathcal{K}_n = \langle b, Ab, \ldots, A^{n-1}b \rangle$ such that $||r_n||_2 = ||b - Ax_n||_2$ is minimized $\rightarrow$ optimal use of expensive matvecs
- LBM, $m = 0$: $||r_n||_2 \leq K|1-\omega|^n$; cf. rate $|1-\omega|$ for FI (again!)
- LBM, $m > 0$: FI unstable if $\omega \not\approx 1$ (eigenvalues $\partial\mathcal{C}/\partial V$ outside unit disk) $\rightarrow$ may also cause slow GMRES convergence ($A = I - \partial\mathcal{C}/\partial V$)
- Using very irregular initial guess (for worst-case behavior):

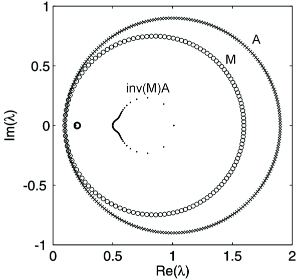


- (Much) faster convergence if zero initial guess

Preconditioning the GMRES iteration (for LBM)

- Additional acceleration possible by incorporating a preconditioner
- We use a coarse grid correction preconditioner [Padiy, Axelsson, Polman (2000)]: $M^{-1} = P^{N}_{N/r} A_c^{-1} R^{N/r}_{N} + \eta I$
  - P and R: traditional prolongation and restriction from multigrid
  - A_c^{-1}: (in)exact coarse system solve using an *inner* GMRES
  - ηI: tuning parameter times the identity matrix
- Inexact *inner* GMRES $\rightarrow$ variable precond. $\rightarrow$ flexible *outer* GMRES
- LBM example: $m = 0$, $\omega = \eta = 0.1$, $A \in \mathbb{R}^{256x256}$, $A_c \in \mathbb{R}^{128x128}$





Outline

Comparison: FI versus NK

ω	FI	NK	Precond. NK
0.1	∞ (∞)	1.9e-02 (518)	1.9e-02 (266)
0.2	∞ (∞)	1.2e-03 (492)	1.2e-03 (120)
0.3	∞ (∞)	1.3e-04 (468)	1.3e-04 (64)
0.4	∞ (∞)	5.9e-05 (308)	5.9e-05 (40)
0.5	∞ (∞)	2.6e-05 (132)	2.6e-05 (30)
0.6	∞ (∞)	1.2e-05 (64)	1.2e-05 (24)
0.7	5.8e-06 (56)	5.8e-06 (36)	5.8e-06 (18)
0.8	3.0e-06 (38)	3.0e-06 (20)	3.0e-06 (16)
0.9	1.6e-06 (24)	1.6e-06 (14)	1.6e-06 (14)
1.0	8.9e-07 (4)	8.9e-07 (8)	8.9e-07 (10)
1.1	5.0e-07 (26)	5.0e-07 (14)	5.0e-07 (16)
1.2	2.8e-07 (50)	2.8e-07 (16)	2.8e-07 (20)
1.3	∞ (∞)	1.6e-07 (16)	3.8e-07 (42)
1.4	∞ (∞)	9.1e-08 (18)	1.2e-07 (72)
1.5	∞ (∞)	1.1e-07 (152)	1.6e-07 (180)
...	...	...	...

- LBM, 128 spatial grid points, ω variable, $\lambda = 100$, $m = 1$, $tol = 10^{-8}$
- Accuracy (compared to $m = 4$ solution) and efficiency (# LBM calls; 2 per GMRES iteration)
- FI: can only be used in a limited range of ω-values
- NK: can always be used
- After convergence, the accuracy is the same (same fixed point)

Comparison: FI versus NK

ω	FI	NK	Precond. NK
0.1	∞ (∞)	1.9e-02 (518)	1.9e-02 (266)
0.2	∞ (∞)	1.2e-03 (492)	1.2e-03 (120)
0.3	∞ (∞)	1.3e-04 (468)	1.3e-04 (64)
0.4	∞ (∞)	5.9e-05 (308)	5.9e-05 (40)
0.5	∞ (∞)	2.6e-05 (132)	2.6e-05 (30)
0.6	∞ (∞)	1.2e-05 (64)	1.2e-05 (24)
0.7	5.8e-06 (56)	5.8e-06 (36)	5.8e-06 (18)
0.8	3.0e-06 (38)	3.0e-06 (20)	3.0e-06 (16)
0.9	1.6e-06 (24)	1.6e-06 (14)	1.6e-06 (14)
1.0	8.9e-07 (4)	8.9e-07 (8)	8.9e-07 (10)
1.1	5.0e-07 (26)	5.0e-07 (14)	5.0e-07 (16)
1.2	2.8e-07 (50)	2.8e-07 (16)	2.8e-07 (20)
1.3	∞ (∞)	1.6e-07 (16)	3.8e-07 (42)
1.4	∞ (∞)	9.1e-08 (18)	1.2e-07 (72)
1.5	∞ (∞)	1.1e-07 (152)	1.6e-07 (180)
...	...	...	...

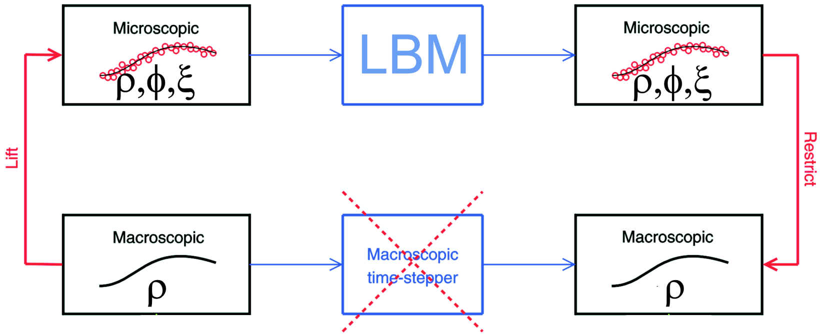
- NK is more efficient, even without preconditioning
- Only near $\omega = 0$, NK becomes very expensive (oversolving!)
- There the preconditioner may keep the cost acceptable

- Much larger preconditioning gain when finer LBM discretization
- Coarse time integration: lifting may become (much) cheaper (ρ smoother $\rightarrow$ residual smoother $\rightarrow$ faster convergence!)

Outline

The coarse time-stepper

Lifting with constrained runs



- Lifting: K steps of the constrained runs functional iteration $(m = 0)$
- Simulation: 1 LBM step
- Restriction: return ρ

Accuracy and stability of the coarse time-stepper

Pure diffusion with $D = 1$: $\dfrac{\partial \rho(x,t)}{\partial t} = \dfrac{\partial^2 \rho(x,t)}{\partial x^2}, \quad \rho(0,t) = \rho(1,t) = 0.$

CTS with K steps of the constrained runs functional iteration $(m = 0)$

- Truncation error: $\overline{T}(x,t) = \dfrac{\partial \rho}{\partial t} - \left(\dfrac{\omega - 2 + 2(1-\omega)^{K+1}}{\omega - 2}\right)\dfrac{\partial^2 \rho}{\partial x^2}$
  - Small K: again solution of diffusion equation with different D
  - As K grows: fast linear convergence $D \to 1$, rate $|1-\omega|$ (again!)
  - Truncation error if $K = \infty$:
$$\overline{T}(x,t) = \frac{1}{2}\frac{\partial^2 \rho}{\partial t^2}\Delta t + \left(\frac{1}{6}\Delta x^2 - \frac{27}{12}\frac{\Delta t^2}{\Delta x^2}\right)\frac{\partial^4 \rho}{\partial x^4}$$
- Stability interval:

	K=0	K=1	K=2	K=4	K=6	K=8	K=25	K=100	K=∞
$\omega_{\min}$	0.000	0.500	0.352	0.305	0.253	0.217	0.107	0.038	0.000
$\omega_{\max}$	2.000	1.250	1.201	1.200	1.200	1.200	1.200	1.200	1.200

  - Now unstable if $\omega > 1.2$ (if Δt is too *small*!)

The influence of increasing the number of LBM steps M

CTS with constrained runs lifting until convergence:

- Accuracy gets better
- Stability interval $(0, \omega_{\max})$:

	M=1	M=2	M=3	M=4	M=5	M=6	M=8	M=10
$\omega_{\max}$	1.200	1.500	1.500	1.500	1.858	1.583	1.708	1.814

  - The stability improves when M is increased (not monotonically)

Outline

Conclusions

We studied the numerical properties of different aspects of equation-free computing when the microscopic simulator is a lattice Boltzmann model

- For time-dependent problems, sufficiently accurate lifting is crucial to obtain a coarse time-stepper that mimics the macroscopic system
- Constrained runs numerically implements such a good lifting
  - if m increases: lifting more accurate but numerics less stable
  - can be stabilized with a (preconditioned) Newton-Krylov solver
- Even if the lifting is sufficiently accurate, the coarse time-stepper may be unstable for sometimes surprising parameter values. Increasing the coarse time step may help