Progress in Lattice Boltzmann Schemes: Mesh Adaptation, High-Order Accuracy, Numerical Analysis, and Boundary Conditions

Thomas Bellotti

CNRS – EM2C & Fédération de Mathématiques de CentraleSupélec – CentraleSupélec – Université Paris-Saclay – France

22nd of January, 2026

Advanced Modeling & Simulation (AMS) Seminar Series
NASA Ames Research Center, January 22nd, 2026







Introduction

Lattice Boltzmann schemes *versus* traditional numerical methods for PDEs

Lattice Boltzmann schemes

End of the 1980s [McNamara & Zanetti, '**88**]: fast and physically-driven alternative to traditional numerical methods for PDEs (conservation laws):

- Incompressible Navier-Stokes.
- Hyperbolic systems (shallow water, Euler, *etc.*).
- Diphasic systems (diffuse interface).
- ...

Traditional numerical methods

Finite Diff., Finite Volume, Finite Elem., *etc.*

N PDEs
$\downarrow$
N numerical schemes (unknowns)

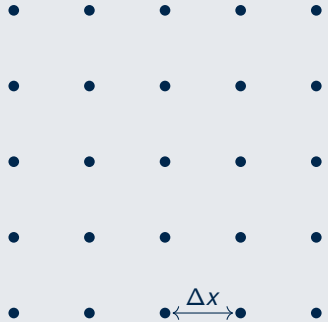
Lattice Boltzmann methods

$q > N$ numerical schemes (unknowns)
$\downarrow$
N PDEs

... and a peculiar structure.

Lattice Boltzmann: algorithmic point of view

Lattice



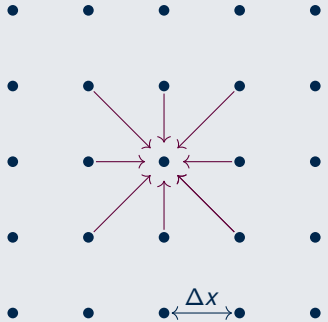
Boltzmann

Imitation of the Boltzmann equation

$$\partial_t f + \underbrace{\boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} f}_{\text{transport}} = \underbrace{Q(f)}_{\text{collision}},$$

with a finite number of (well-selected) velocities.

Lattice



Boltzmann

Imitation of the Boltzmann equation

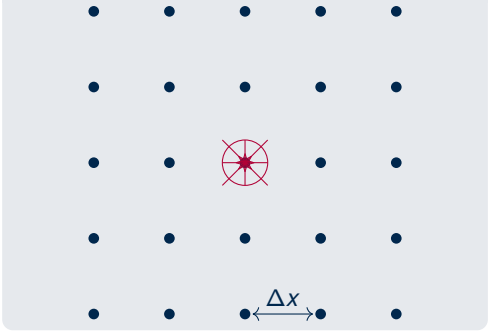
$$\partial_t f + \underbrace{\boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} f}_{\text{transport}} = \underbrace{Q(f)}_{\text{collision}},$$

with a finite number of (well-selected) velocities.

- Transport: non-local but linear

$$\mathrm{f}_j(t + \tfrac{\Delta t}{2}, \boldsymbol{x}) = \mathrm{f}_j(t, \boldsymbol{x} - \Delta t \boldsymbol{v}_j), \qquad j = 1, \ldots, q.$$

Lattice



Boltzmann

Imitation of the Boltzmann equation

$$\partial_t f + \underbrace{\boldsymbol{v} \cdot \nabla_{\boldsymbol{x}} f}_{\text{transport}} = \underbrace{Q(f)}_{\text{collision}},$$

with a finite number of (well-selected) velocities.

- Transport: non-local but linear

$$\mathrm{f}_j(t + \tfrac{\Delta t}{2}, \boldsymbol{x}) = \mathrm{f}_j(t, \boldsymbol{x} - \Delta t \boldsymbol{v}_j), \qquad j = 1, \ldots, q.$$

- Collision: local but non-linear

$$\mathrm{f}_j(t + \Delta t, \boldsymbol{x}) = Q_j(\mathrm{f}_1(t + \tfrac{\Delta t}{2}, \boldsymbol{x}), \ldots, \mathrm{f}_q(t + \tfrac{\Delta t}{2}, \boldsymbol{x})), \quad j = 1, \ldots, q.$$

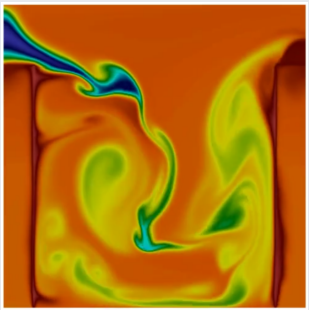
Quite often Q_j is the BGK operator $Q_j = (1 - \omega)\mathrm{f}_j + \omega \mathrm{f}_j^{\mathrm{eq}}$.

Pros and cons of lattice Boltzmann

They all come from this stream-and-collide algorithmic origin on Cartesian meshes using many unknowns.

Pros

- Ease of implementation (solver free, *etc.*).
- Fast.
- Embarassingly and easily parallel.



Temperature field in an AC system by `pyLBM`.

Cons

All coming from overabundant unknowns: $q > N$.

- Difficult to design.
- **Lack of mathematical understanding** (*e.g.* consistency, stability, convergence; initialization; boundary conditions).
- **Increase order**.
- Boundary conditions.
- **Memory consuming** ($q \gg N$).
- **Fixed uniform Cartesian meshes**.

Plan of the talk

1. Introduction
2. Dynamic mesh adaptation with error control
3. Construction of fourth-order accurate methods
4. Numerical analysis: connection between LBM and multi-step Finite Difference
5. Stability of boundary conditions

Dynamic mesh adaptation with error control

Problem

Joint work with B. Graille (Orsay, France), L. Gouarin, and M. Massot (Ecole polytechnique, France).

Why?

- Fine resolution needed in a few spots in the domain (shocks, *etc.*).
- Lattice Boltzmann is memory expensive when $q \gg N$.

  For example: 2D Euler ($N = 4$ and $q = 16$), 2D incompressible Navier-Stokes ($N = 3$ and $q = 9$), and 3D incompressible Navier-Stokes ($N = 4$ and $q = 27$).

Existing approaches to the problem.

Fixed meshed

[Filippova & Hänel, '**98**], [Dupuis & Chopard, '**03**], [Rohde *et al.*, '**06**].

- ☺ Efficient (no mesh to modify).
- ☹ *A priori* knowledge of the flow (steady regime and problem-dependent).
- ☹ No error control.

Time-moving meshes with AMR

[Crouse *et al.*, '**03**], [Fakhari & Lee, '**14**].

- ☺ Dynamic.
- ☹ Mesh modifications needed.
- ☹ Heuristic criteria (problem-dependent and no error control).

Problem and idea

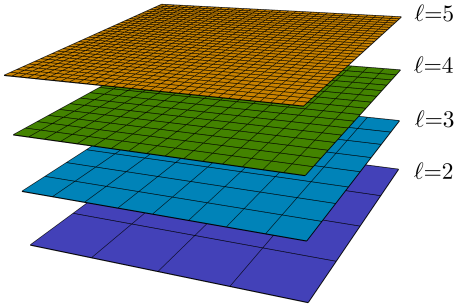
Criteria

- Dynamic adaptation in time $\rightarrow$ moving meshes.
- Independent of the lattice Boltzmann scheme at hand $\rightarrow$ global time-step.
- Ensure error control $\rightarrow$ adaptive multiresolution [Cohen *et al.*, '**03**].

You can find the details in [B. *et al.*, *SIAM J. Sci. Comput.* ('**22**)] and [B. *et al.*, *J. Comput. Phys.* ('**22**)].

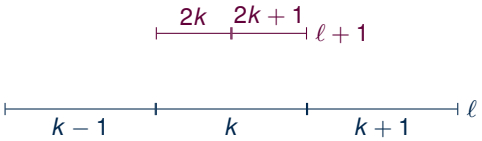
Holistic employ of multiresolution:

1. Adapt the mesh.
2. Adapt lattice Boltzmann schemes.



Adaptive multiresolution

Decomposition on a wavelet basis [Daubechies, '**88**] to measure local smoothness (or lack thereof).



- Projection $\ell+1 \mapsto \ell$. Only one way

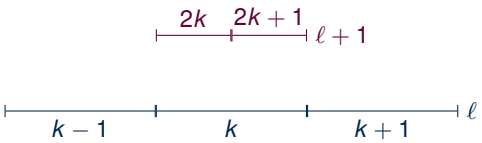
$$\mathsf{f}_{\ell,k} = \frac{1}{2}(\mathsf{f}_{\ell+1,2k} + \mathsf{f}_{\ell+1,2k+1}).$$

- Prediction $\ell \mapsto \ell+1$. Many ways, for example

$$\hat{\mathsf{f}}_{\ell+1,2k} = \mathsf{f}_{\ell,k} + \dots$$

Adaptive multiresolution

Decomposition on a wavelet basis [Daubechies, '88] to measure local smoothness (or lack thereof).



Example: we face a change of basis

Take $\hat{\mathsf{f}}_{\ell+1,2k} = \mathsf{f}_{\ell,k}$, then we see that

$$\begin{pmatrix} \mathsf{f}_{\ell,k} \\ \mathsf{d}_{\ell+1,2k} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \mathsf{f}_{\ell+1,2k} \\ \mathsf{f}_{\ell+1,2k+1}. \end{pmatrix}$$

- Projection $\ell+1 \mapsto \ell$. Only one way

$$\mathsf{f}_{\ell,k} = \frac{1}{2}(\mathsf{f}_{\ell+1,2k} + \mathsf{f}_{\ell+1,2k+1}).$$

- Prediction $\ell \mapsto \ell+1$. Many ways, for example

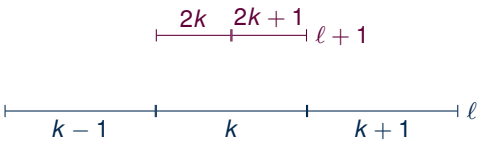
$$\hat{\mathsf{f}}_{\ell+1,2k} = \mathsf{f}_{\ell,k} + \dots$$

- Detail: difference between actual and predicted datum

$$\mathsf{d}_{\ell+1,2k} = \mathsf{f}_{\ell+1,2k} - \hat{\mathsf{f}}_{\ell+1,2k}.$$

Adaptive multiresolution

Decomposition on a wavelet basis [Daubechies, '88] to measure local smoothness (or lack thereof).



- Projection $\ell + 1 \mapsto \ell$. Only one way

$$\mathsf{f}_{\ell,k} = \frac{1}{2}(\mathsf{f}_{\ell+1,2k} + \mathsf{f}_{\ell+1,2k+1}).$$

- Prediction $\ell \mapsto \ell + 1$. Many ways, for example

$$\hat{\mathsf{f}}_{\ell+1,2k} = \mathsf{f}_{\ell,k} + \dots$$

- Detail: difference between actual and predicted datum

$$\mathsf{d}_{\ell+1,2k} = \mathsf{f}_{\ell+1,2k} - \hat{\mathsf{f}}_{\ell+1,2k}.$$

Example: we face a change of basis

Take $\hat{\mathsf{f}}_{\ell+1,2k} = \mathsf{f}_{\ell,k}$, then we see that

$$\begin{pmatrix} \mathsf{f}_{\ell,k} \\ \mathsf{d}_{\ell+1,2k} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \mathsf{f}_{\ell+1,2k} \\ \mathsf{f}_{\ell+1,2k+1}. \end{pmatrix}$$

Fast wavelet transform

$$\mathbf{f}_{\overline{\ell}} \quad \leftrightarrows \quad (\mathbf{f}_{\underline{\ell}}, \mathbf{d}_{\underline{\ell}+1}, \dots, \mathbf{d}_{\overline{\ell}}).$$

small detail = locally smooth datum.

Adaptive multiresolution: Mesh adaptation

Let $\epsilon > 0$ be a global error tolerance that we can tune.

Mesh coarsening (static)

$$\text{Small detail } (\textit{i.e.}\ |\mathrm{d}_{\ell,k}| < \epsilon_\ell(\epsilon)) \quad \rightarrow \quad \text{put } \mathrm{d}_{\ell,k} = 0 \ (\textit{i.e.}\ \text{erase the corresponding cell})$$

Even discarding details, we have error control

$$\|\mathbf{f}_{\overline{\ell}} - \mathbf{f}_{\overline{\ell}} \text{ with discarded details}\| \lesssim \epsilon.$$

Mesh refinement (dynamics)

Mesh is updated using old information at time t to store the still-unknown solution at time $t + \Delta t$.

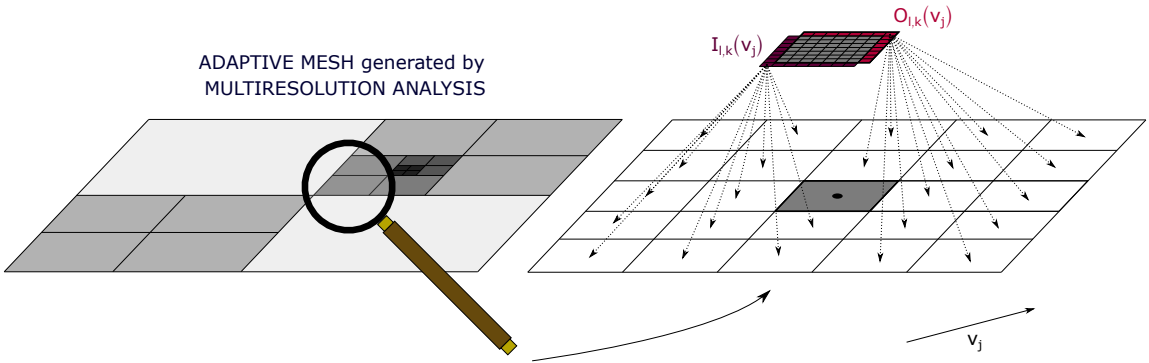
Phenomena to keep in mind:

1. Propagation of information (transport) $\rightarrow$ add security cells.
2. Smoothness decreases (collision) $\rightarrow$ additional criterion on the details.

$$\text{Large detail } (\textit{i.e.}\ |\mathrm{d}_{\ell,k}| > C \times \epsilon_\ell(\epsilon)) \quad \rightarrow \quad \text{locally refine the mesh.}$$

These are essential: think about the Burgers equation.

Adaptive lattice Boltzmann methods



- Transport: as if we were on the finest level $\overline{\ell}$ using multiresolution

$$f_{j,\ell,\boldsymbol{k}}(t+\tfrac{\Delta t}{2}) = f_{j,\ell,\boldsymbol{k}}(t) + \frac{1}{2^{d(\overline{\ell}-\ell)}}\Big(\sum_{\overline{\boldsymbol{k}}\in\mathcal{I}_{\ell,\boldsymbol{k}}(\boldsymbol{v}_j)} \hat{f}_{j,\overline{\ell},\overline{\boldsymbol{k}}}(t) - \sum_{\overline{\boldsymbol{k}}\in\mathcal{O}_{\ell,\boldsymbol{k}}(\boldsymbol{v}_j)} \hat{f}_{j,\overline{\ell},\overline{\boldsymbol{k}}}(t)\Big).$$

- Collision: untouched since local $f_{j,\ell,\boldsymbol{k}}(t+\Delta t) = Q_j(f_{1,\ell,\boldsymbol{k}}(t+\tfrac{\Delta t}{2}),\dots,f_{q,\ell,\boldsymbol{k}}(t+\tfrac{\Delta t}{2}))$.

Numerical experiment: 2D Euler system – qualitative features

We want to check that

- Dynamic adaptation in time works.
- It follows the shocks.
- Mesh is coarsened where fine resolution is not needed.

Riemann problems by [Lax and Liu, '**98**] and implosion by [Liska & Wendroff, '**03**].

Movies.

Numerical experiment: 2D Euler system – quantitative features

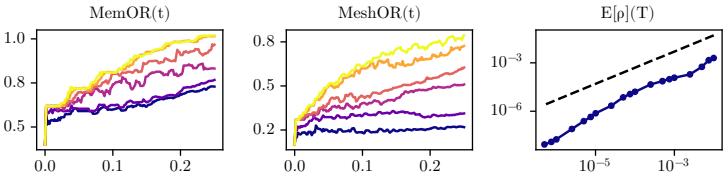
We want to check

- Low memory occupation.
- Error control by ϵ.

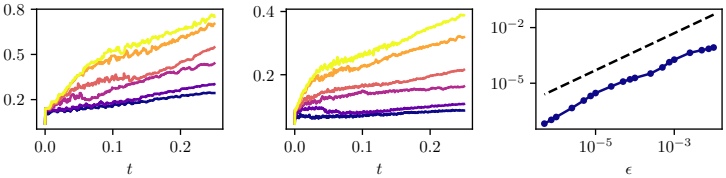
We use [Lax and Liu, '**98**] configuration N. 12.

- $\epsilon = 1E-2$
- $\epsilon = 5E-3$
- $\epsilon = 1E-3$
- $\epsilon = 5E-4$
- $\epsilon = 1E-4$
- $\epsilon = 5E-5$

$\underline{\ell} = 2, \overline{\ell} = 7$



$\underline{\ell} = 2, \overline{\ell} = 9$



Implementation: SAMURAI



SAMURAI

Structured Adaptive mesh and MUlti-Resolution based on Algebra of Intervals

github.com/hpc-maths/samurai

Features

- Handle adaptive Cartesian meshes without constructing the associated tree.
- Other numerical schemes with simplicity: Finite Volume, *etc.*

Construction of fourth-order accurate methods

Problem

Joint work with P. Helluy and L. Navoret (Strasbourg, France).
Important kind of equations: systems of 1st order conservation laws

$$\partial_t \begin{pmatrix} u_1 \\ \vdots \\ u_N \end{pmatrix} = \mathbf{\Phi}_{\mathrm{PDE}} \left(\begin{pmatrix} u_1 \\ \vdots \\ u_N \end{pmatrix} \right), \qquad \text{where} \qquad \mathbf{\Phi}_{\mathrm{PDE}}(\boldsymbol{u}) = -\mathrm{div}_{\boldsymbol{x}}(\boldsymbol{\varphi}(\boldsymbol{u})).$$

This setting encompasses the shallow water system, the Euler system, *etc.*
Standard lattice Boltzmann schemes can be at most second-order accurate for this problem, with

$$\underbrace{\sum_j \mathsf{f}_j = \sum_j \mathsf{f}_j^{\mathrm{eq}} = \boldsymbol{u}, \qquad \sum_j v_j \mathsf{f}_j^{\mathrm{eq}} = \mathbf{\Phi}_{\mathrm{PDE}},}_{\text{consistency}} \qquad \underbrace{Q_j = (1-\omega)\mathsf{f}_j + \omega \mathsf{f}_j^{\mathrm{eq}}|_{\omega=2} = -\mathsf{f}_j + 2\mathsf{f}_j^{\mathrm{eq}}}_{\text{second-order accuracy}}.$$

The idea to achieve this splits in three steps, see [B., Helluy, Navoret, *SIAM SISC* (’**24**)]:

1. Construct a 2nd-order lattice Boltzmann scheme as above.
2. Symmetrize to obtain a 2nd-order time-symmetric lattice Boltzmann scheme.
3. Reach fourth-order by composing 2nd-order time-symmetric lattice Boltzmann schemes, see [McLachlan & Quispel, ’**02**].

More details on the procedure

1. Let $\boldsymbol{Q} = \boldsymbol{Q}_{\omega=2}$, then
$$\boldsymbol{\Phi}^{\text{2nd}}_{\text{LBM}}(\Delta t) = \boldsymbol{QT}(\Delta t).$$
In this case $\boldsymbol{QQ} = \boldsymbol{Id}$, the collision is an involution.
2. We create a time-symmetric scheme $\overline{\boldsymbol{\Phi}}^{\text{2nd}}_{\text{LBM}}(\Delta t)$, that is—such that
$$\overline{\boldsymbol{\Phi}}^{\text{2nd}}_{\text{LBM}}(0) = \boldsymbol{Id}, \qquad \overline{\boldsymbol{\Phi}}^{\text{2nd}}_{\text{LBM}}(\Delta t)\overline{\boldsymbol{\Phi}}^{\text{2nd}}_{\text{LBM}}(-\Delta t) = \boldsymbol{Id}.$$
The easiest way of doing this is to take
$$\overline{\boldsymbol{\Phi}}^{\text{2nd}}_{\text{LBM}}(\Delta t) = \overbrace{\boldsymbol{T}\Big(\frac{\Delta t}{4}\Big)\boldsymbol{QT}\Big(\frac{\Delta t}{4}\Big)}^{\boldsymbol{\Phi}^{\text{2nd}}_{\text{LBM}}(\Delta t/2) \text{ à la Strang}} \boldsymbol{T}\Big(\frac{\Delta t}{4}\Big)\boldsymbol{QT}\Big(\frac{\Delta t}{4}\Big).$$
3. We increase to order four by creating
$$\overline{\boldsymbol{\Phi}}^{\text{4th}}_{\text{LBM}}(\Delta t) = \overline{\boldsymbol{\Phi}}^{\text{2nd}}_{\text{LBM}}(\alpha\Delta t)^n\overline{\boldsymbol{\Phi}}^{\text{2nd}}_{\text{LBM}}(\beta\Delta t)\overline{\boldsymbol{\Phi}}^{\text{2nd}}_{\text{LBM}}(\alpha\Delta t)^n, \qquad \text{where} \qquad n = 1, 2, \ldots, \quad \begin{cases} 2n\alpha + \beta = 1, \\ 2n\alpha^3 + \beta^3 = 0. \end{cases}$$

More details on the procedure

As we want to stick with transport phases made up of shifts on the regular grid: $\alpha, \beta \in \mathbb{Q}$.

- $n = 1, 2, 3$, we fail, for $\alpha, \beta \in \mathbb{R} \setminus \mathbb{Q}$.
- $n = 4$, we succeed, as $\alpha = 1/6 \in \mathbb{Q}$ and $\beta = -1/3 \in \mathbb{Q}$.

We therefore utilize

$$\overline{\mathbf{\Phi}}^{\text{4th}}_{\text{LBM}}(\Delta t) = \underbrace{\overline{\mathbf{\Phi}}^{\text{2nd}}_{\text{LBM}}\left(\frac{\Delta t}{6}\right)^4}_{\boldsymbol{\tau}\left(\frac{\Delta t}{24}\right)\boldsymbol{a}\cdots} \overline{\mathbf{\Phi}}^{\text{2nd}}_{\text{LBM}}\left(-\frac{\Delta t}{3}\right) \overline{\mathbf{\Phi}}^{\text{2nd}}_{\text{LBM}}\left(\frac{\Delta t}{6}\right)^4, \qquad \text{with} \qquad \frac{\Delta t}{24\Delta x} = 1.$$

Rationale

To gain fourth-order accuracy, we must both advance and go back in time.

Numerical experiment: order

1D shallow water system ($N = 2$)

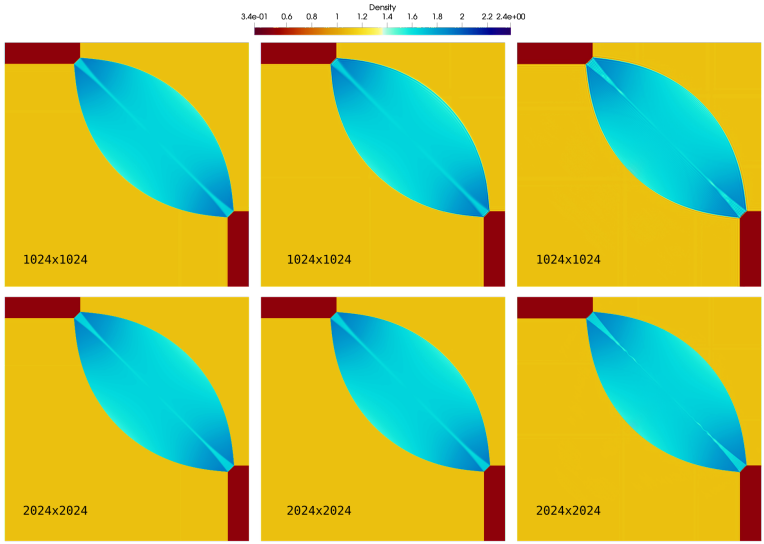
- $\varphi(h, hu) = (h, hu^2 + \frac{g}{2}h^2)$.
- Smooth solutions.
- $q = 2$

	h		u	
Δx	Error	Order	Error	Order
7.8125E-03	5.8333E-06		2.9538E-05	
3.9063E-03	7.9483E-07	**2.88**	1.6474E-06	**4.16**
1.9531E-03	1.0703E-07	**2.89**	4.8759E-08	**5.08**
9.7656E-04	7.6700E-09	**3.80**	2.9001E-09	**4.07**
4.8828E-04	4.9440E-10	**3.96**	1.8273E-10	**3.99**
2.4414E-04	3.1134E-11	**3.99**	1.1456E-11	**4.00**
1.2207E-04	1.9495E-12	**4.00**	7.1665E-13	**4.00**
6.1035E-05	1.2202E-13	**4.00**	4.5492E-14	**3.98**

Numerical experiment: 2D Riemann problems for the Euler equations

2D Euler system ($N = 4$)

- Initial datum [Lax and Liu, '**98**], configuration #4.
- $q = 4$.
- Implemented and parallelized on GPUs (`OpenCL`).



Some open questions

- **Shocks**: with high-order schemes, Gibbs phenonenon and instability.
  This calls for limitation procedures.
  - *A priori*, in the spirit of slope limiters for Finite Volume. Careful: retain computational efficiency.
  - *A posteriori*, in the spirit of MOOD.

- **Boundary conditions**: the scheme stencil over Δt goes quite far.

- **Sixth-order**: possible, keeping in mind that we need limiters.

- **Systems with dissipation**: in this case $\boldsymbol{QQ} \neq \boldsymbol{Id}$.

Numerical analysis: connection between LBM and multi-step Finite Difference

Problem and idea to fix it

Joint work with B. Graille and M. Massot.

Continuous problem (PDEs) Discrete problem (LBM scheme)

$$\partial_t \begin{pmatrix} u_1 \\ \vdots \\ u_N \end{pmatrix} = \mathbf{\Phi}_{\text{PDE}} \left(\begin{pmatrix} u_1 \\ \vdots \\ u_N \end{pmatrix} \right) \qquad \begin{pmatrix} \mathrm{m}_1 \\ \vdots \\ \mathrm{m}_N \\ \mathrm{m}_{N+1} \\ \vdots \\ \mathrm{m}_q \end{pmatrix} (t + \Delta t) = \mathbf{\Phi}_{\text{LBM}} \left(\begin{pmatrix} \mathrm{m}_1 \\ \vdots \\ \mathrm{m}_N \\ \mathrm{m}_{N+1} \\ \vdots \\ \mathrm{m}_q \end{pmatrix} (t) \right)$$

However, we only care about $\mathrm{m}_1 \approx u_1, \ldots, \mathrm{m}_N \approx u_N$: we eliminate $\mathrm{m}_{N+1}, \ldots, \mathrm{m}_q$ to study

- Consistency: what are the equations (indeed $\mathbf{\Phi}_{\text{PDE}}$) that the scheme solves?
- Stability: are round-off errors kept under control?
- Initialization: how to take $\mathrm{m}_{N+1}, \ldots, \mathrm{m}_q$ at the beginning?

Keep in mind : in the linear setting, the Lax theorem ensures that a consistent scheme converges if and only if it is stable.

How to eliminate the non-conserved moments

The lattice Boltzmann operator splits in the following way:

$$\mathbf{\Phi}_{\text{LBM}}\left(\begin{pmatrix}\mathrm{m}_1\\ \vdots\\ \mathrm{m}_N\\ \mathrm{m}_{N+1}\\ \vdots\\ \mathrm{m}_q\end{pmatrix}\right) = \mathbf{\Phi}^{\text{linear}}_{\text{LBM}}\left(\begin{pmatrix}\mathrm{m}_1\\ \vdots\\ \mathrm{m}_N\\ \mathrm{m}_{N+1}\\ \vdots\\ \mathrm{m}_q\end{pmatrix}\right) + \mathbf{\Phi}^{\text{non-linear}}_{\text{LBM}}\left(\begin{pmatrix}\mathrm{m}_1\\ \vdots\\ \mathrm{m}_N\end{pmatrix}\right).$$

The linear operator $\mathbf{\Phi}^{\text{linear}}_{\text{LBM}}$ is a $q \times q$ matrix, so we can use the tools from linear algebra.

The one we use is the Cayley-Hamilton theorem, which states that the matrix $\mathbf{\Phi}^{\text{linear}}_{\text{LBM}}$ satisfies its own characteristic equation, and obtain

$$\begin{pmatrix}\mathrm{m}_1\\ \vdots\\ \mathrm{m}_N\end{pmatrix}(t+\Delta t) = \sum_{k=0}^{q-N} \mathbf{\Phi}^{(k)}_{\text{FD}}\left(\begin{pmatrix}\mathrm{m}_1\\ \vdots\\ \mathrm{m}_N\end{pmatrix}(t-k\Delta t)\right).$$

Some intuition using ODEs

Harmonic oscillator

$$\begin{cases} \dot{x} = v, \\ \dot{v} = -\omega^2 x \end{cases} \qquad \rightarrow \qquad \ddot{x} + \omega^2 x = 0.$$

Advancements: We analyze $\mathbf{\Phi}_{\mathrm{LBM}}$ using $\mathbf{\Phi}_{\mathrm{FD}}^{(k)}$, without computing the latter explicitly

Consistency: what are the equations that the scheme $\mathbf{\Phi}_{\mathrm{LBM}}$ solves?

Let $\Delta x/\Delta t$ be constant. The modified equations of $\mathbf{\Phi}_{\mathrm{LBM}}$, computed using $\mathbf{\Phi}_{\mathrm{FD}}^{(k)}$, are

$$\partial_t \begin{pmatrix} u_1 \\ \vdots \\ u_N \end{pmatrix} = \underbrace{\mathbf{\Phi}_{\mathrm{PDE}}\left(\begin{pmatrix} u_1 \\ \vdots \\ u_N \end{pmatrix}\right)}_{\text{1st-order diff. op.}} + \Delta x \underbrace{\mathbf{\Psi}_{\text{num. diff.}}\left(\begin{pmatrix} u_1 \\ \vdots \\ u_N \end{pmatrix}\right)}_{\text{2nd-order diff. op.}} + \mathcal{O}(\Delta x^2), \text{ where } \begin{cases} \mathbf{\Phi}_{\mathrm{PDE}} = \mathbf{\Phi}_{\mathrm{PDE}}(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_q, \boldsymbol{Q}) \\ \mathbf{\Psi}_{\text{num. diff.}} = \mathbf{\Psi}_{\text{num. diff.}}(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_q, \boldsymbol{Q}), \end{cases}$$

are explicitly known. See [B., *ESAIM-M2AN* (’**23**)] for all the details.

Stability: are round-off errors kept under control?

For scalar ($N = 1$) and linear schemes, the lattice Boltzmann scheme $\mathbf{\Phi}_{\mathrm{LBM}}$ is spectrally (L^2) stable if and only if the Finite Difference scheme $\mathbf{\Phi}_{\mathrm{FD}}^{(k)}$ is spectrally stable. See [B., Massot, Graille, *Numer. Math.* (’**22**)].

Initialization: how to take $\mathrm{m}_{N+1}, \ldots, \mathrm{m}_q$ at the beginning?

A well-known choice of $\mathrm{m}_{N+1}(t = 0), \ldots, \mathrm{m}_q(t = 0)$, called equilibrium, ensures first-order consistency with $\partial_t \boldsymbol{u} = \mathbf{\Phi}_{\mathrm{PDE}}(\boldsymbol{u})$, see [B., *J. Comput. Phys* (’**24**)].

Some open questions

- **Stability for non-linear schemes**: the spectral norm is no longer useful.
  Introduce the concept of monotone scheme, as for Finite Volume.
  The schemes $\Phi_{\mathrm{FD}}^{(k)}$ do not behave well with respect to this. Difficult to conclude without computations.
  It seems better to study things directly on Φ_{LBM}.

- **Boundary conditions**: the scheme Φ_{LBM} is no longer the same for every grid-point.
  I do not know how to compute $\Phi_{\mathrm{FD}}^{(k)}$ in general. Is it possible?
  Devise tools to study boundary conditions directly on Φ_{LBM}.

Stability of boundary conditions

Problem

Mathematically, stability with boundary conditions (no initial data) is

$$\|\text{numerical solution}\| \leq C\|\text{boundary data}\|.$$

Question: is the numerical solution only $m_1, \ldots, m_N$ or also $m_{N+1}, \ldots, m_q$?

This question is crucial as eliminating $m_{N+1}, \ldots, m_q$ in presence of boundaries can be difficult (if not impossible), although we did it in a simple setting [B., *IMA J. Numer. Anal.* ('**25**)].

Strong stability

We consider $N = 1$, linear problem on a half-space ($x > 0$). Essentially drawn from the seminal paper of [Gustafsson, Kreiss, Sundstrom, '**72**], but differences with FD exist.

For all $\alpha > 0$:

$$\text{(SS)} \qquad \frac{\alpha}{1+\alpha\Delta t} \underbrace{\sum_{t\in\Delta t\mathbb{N}} \sum_{x\in\Delta x\mathbb{N}} \Delta t\Delta x e^{-2\alpha t} |\mathbf{m}(t,x)|^2}_{\text{norm of the solution}} \leq C \overbrace{\sum_{t\in\Delta t\mathbb{N}} \underbrace{\sum_{v_k>0}}_{\text{incoming}} \sum_{j=1}^{v_k} \Delta t e^{-2\alpha t} g_k(t,-j\Delta x)^2}^{\text{norm of the boundary data}},$$

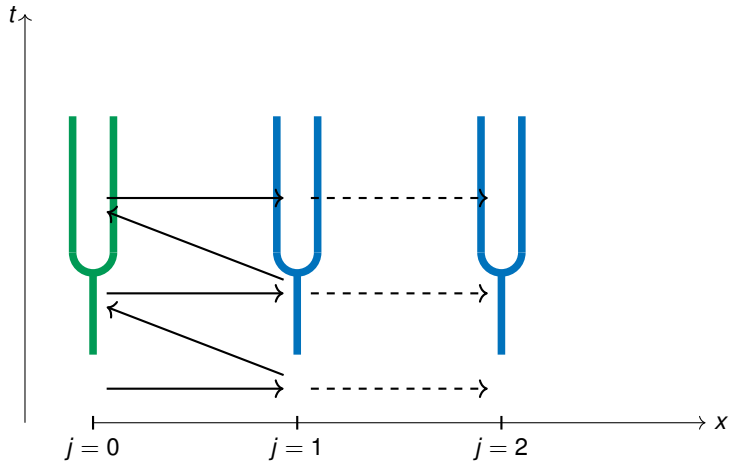
hence we take $\mathrm{m}_1, \ldots, \mathrm{m}_q$ to be the numerical solution of interest. If we consider only m_1 be interesting:

$$\text{(SSOO)} \qquad \frac{\alpha}{1+\alpha\Delta t} \sum_{t\in\Delta t\mathbb{N}} \sum_{x\in\Delta x\mathbb{N}} \Delta t\Delta x e^{-2\alpha t} \mathrm{m}_1(t,x)^2 \leq C \sum_{t\in\Delta t\mathbb{N}} \sum_{v_k>0} \sum_{j=1}^{v_k} \Delta t e^{-2\alpha t} g_k(t,-j\Delta x)^2.$$

Involved but convenient as it can be checked with algebraic conditions (i.e. eigenvalue problem).

Instability as resonance

The boundary scheme and the bulk scheme form a possibly resonant system (think about an optical cavity).



Stability = absence of resonant modes (eigenvalues) between boundary and bulk.

Eigenvalue problem

Here, z is the time-eigenvalue and κ the space-eigenvalue.

We consider $|z| \geq 1$ (otherwise the mode exponentially damps):

$$(z\boldsymbol{I}_q - \boldsymbol{\Phi}^{\text{bulk}}_{\text{LBM}}(\kappa))\boldsymbol{v}^{\text{bulk}}(z) = \boldsymbol{0}_q \qquad \text{and} \qquad (z\boldsymbol{I}_q - \boldsymbol{\Phi}^{\text{bd}}_{\text{LBM}}(\kappa))\boldsymbol{v}^{\text{bd}}(z) = \boldsymbol{0}_q$$

Notice that the eigenvector is not necessarily shared!
Hence we study

$$\text{Find} \quad (z, \kappa) \quad \text{such that} \quad \det(z\boldsymbol{I}_q - \boldsymbol{\Phi}^{\text{bulk}}_{\text{LBM}}(\kappa)) = 0 \qquad \text{and} \qquad \det(z\boldsymbol{I}_q - \boldsymbol{\Phi}^{\text{bd}}_{\text{LBM}}(\kappa)) = 0$$

and we show that instabilities can happen only if this problem has a solution (shared eigenvalue).

Eigenvalue problem

The z-transformed (Laplace transformed) numerical solution reads

$$\tilde{\mathbf{m}}(z,x) = C(z)\boldsymbol{v}^{\text{bulk}}(z)\kappa(z)^{x/\Delta x}$$

- $C(z)$ comes from injecting into the boundary scheme.
- $\boldsymbol{v}^{\text{bulk}}(z)$ eigenvector in the bulk.
- $\kappa(z)$ solution of $\det(z\boldsymbol{I}_q - \boldsymbol{\Phi}^{\text{bulk}}_{\text{LBM}}(\kappa)) = 0$.

Strong stability comes from having $\tilde{\mathbf{m}}(z,x)$ depending continuously (no poles) on the boundary data for $|z| \geq 1$. One easily sees that $\kappa(z)$ cannot be a source of singularities: we are left with $C(z)$ and $\boldsymbol{v}^{\text{bulk}}(z)$ (this is the difference with FD, independent of the boundary conditions).

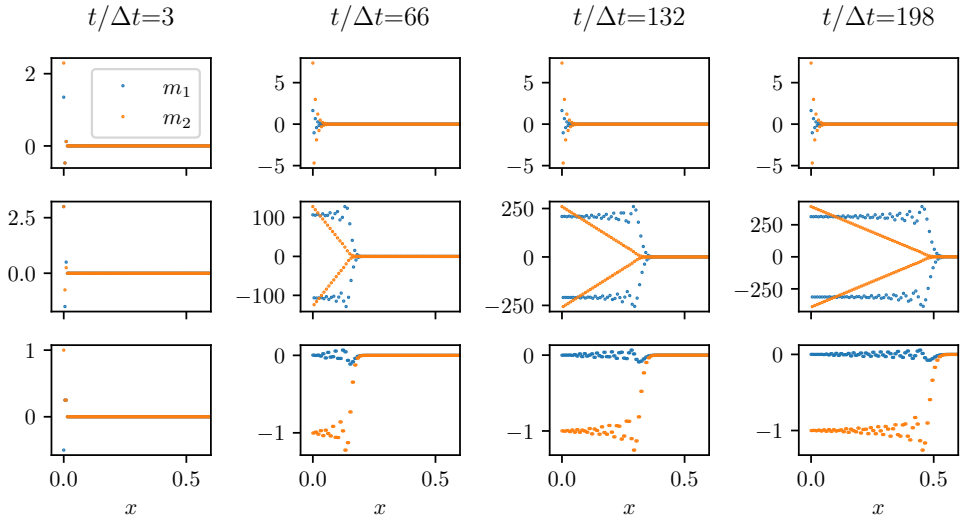
Finally

$$\Delta_{\text{KL}}(z)C(z) = \text{boundary data}(z),$$

where the so-called Kreiss-Lopatinskii determinant $\Delta_{\text{KL}}(z)$ depends on bulk and boundary scheme, as well as $\boldsymbol{v}^{\text{bulk}}(z)$ (thus it can have poles).

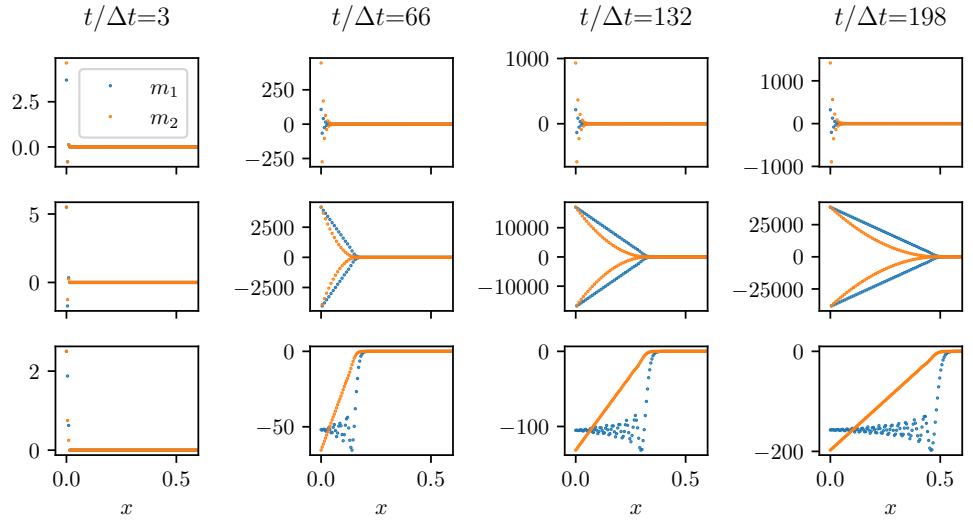
Numerical results on unstable cases: boundary datum = Dirac delta at initial time

Shared eigvl. 1st: no pole $\boldsymbol{v}^{\text{bulk}}$, zero Δ_{KL}. 2nd: pole v_2^{bulk}, zero Δ_{KL}. 3rd: pole v_2^{bulk}, $0 < \Delta_{\text{KL}} < \infty$ (SSOO).



Numerical results on unstable cases: resonant boundary data

Shared eigvl. 1st: no pole $\boldsymbol{v}^{\text{bulk}}$, zero Δ_{KL}. 2nd: pole v_2^{bulk}, zero Δ_{KL}. 3rd: pole v_2^{bulk}, $0 < \Delta_{\text{KL}} < \infty$ (SSOO).



Conclusions on stability for boundary conditions

More details in [B., *Accepted in ESAIM: M2AN* (’**25**)].

Main message

- We do not have to eliminate **“numerical” moments**, but...
- We can have **radically different behaviors** between different components of the numerical solution. This is different from **scalar Finite Difference schemes**.

Drawbacks and future directions

- Only linear theory $\mapsto$ for non-linear, we can do something using specific boundary conditions.
- Embed this in a larger proof of convergence.
- Add initial data.

General recap

1. Introduction
2. Dynamic mesh adaptation with error control
3. Construction of fourth-order accurate methods
4. Numerical analysis: connection between LBM and multi-step Finite Difference
5. Stability of boundary conditions

More information and papers-preprints on my website: `https://thomasbellotti.altervista.org`

Merci!