The Lattice Boltzmann method for hyperbolic systems

Benjamin Graille
October 19, 2016
Framework

The Lattice Boltzmann method

1. Description of the lattice Boltzmann method
   - Link with the kinetic theory
   - Classical schemes
   - Analysis methods
   - Boundary conditions

2. pyLBM (collaboration with Loïc Gouarin)
   - Presentation
   - Examples

3. The vectorial schemes
   - Presentation
   - Numerical tests

4. Conclusion
Description of the lattice Boltzmann method

The Lattice Boltzmann method

1. Description of the lattice Boltzmann method
   - Link with the kinetic theory
   - Classical schemes
   - Analysis methods
   - Boundary conditions

2. pyLBM (collaboration with Loïc Gouarin)

3. The vectorial schemes

4. Conclusion
Description of the lattice Boltzmann method – Link with the kinetic theory

Different scales for modeling

- **microscopic models**
  - particles
  - positions, velocities

- **lattice Boltzmann**
  - Taylor expansion
  - $\Delta t \to 0$

- **mesoscopic models**
  - statistical descriptions
  - mean distribution functions

- **macroscopic models**
  - Chapman-Enskog
  - $\epsilon \to 0$

- **observable quantities**
  - density, velocity, temperature
Description of the lattice Boltzmann method – Link with the kinetic theory

From mesoscopic to macroscopic

Mesoscopic scale: the Boltzmann equation

\[ \partial_t f(t, x, c) + c \cdot \nabla_x f(t, x, c) = \frac{1}{\epsilon} Q(f), \]

where \( \epsilon \) is the Knudsen number.

Chapman-Enskog method: formal asymptotic expansion according to \( \epsilon \) ⇒ the macroscopic equations on the moments.

\[
\begin{align*}
\rho &= \int f(t, x, c) \, dc, \quad \text{(mass)} \\
q &= \int cf(t, x, c) \, dc, \quad \text{(momentum)} \\
E &= \int \frac{1}{2} c^2 f(t, x, c) \, dc, \quad \text{(energy)}
\end{align*}
\]
Description of the lattice Boltzmann method – Link with the kinetic theory

Mimic the kinetic to simulate the macroscopic

Example in 2 dimensions
  - a uniform cartesian mesh
Mimic the kinetic to simulate the macroscopic

Example in 2 dimensions
- a uniform cartesian mesh
- on each spot, “particles” with adapted discrete velocities
Description of the lattice Boltzmann method – Link with the kinetic theory

Mimic the kinetic to simulate the macroscopic

Example in 2 dimensions

- a uniform cartesian mesh
- on each spot, “particles” with adapted discret velocities
- the transport phase
Description of the lattice Boltzmann method – Link with the kinetic theory

Mimic the kinetic to simulate the macroscopic

Example in 2 dimensions

- a uniform cartesian mesh
- on each spot, “particles” with adapted discreet velocities
- the transport phase
- the relaxation phase
Description of the lattice Boltzmann method – Link with the kinetic theory

Mimic the kinetic to simulate the macroscopic

Example in 2 dimensions

- a uniform cartesian mesh
- on each spot, “particles” with adapted discret velocities
- the transport phase
- the relaxation phase
- do it again!
The collision operator is a linear relaxation of the vector $f$ toward an equilibrium value $f^{eq}$

$$f^* = f + M^{-1} S M (f^{eq} - f).$$

conserved moments

even non conserved moments

odd non conserved moments
The collision operator is a linear relaxation of the vector $f$ toward an equilibrium value $f^{eq}$.

$$f^* = f + M^{-1} SM (f^{eq} - f).$$

The value of $f^{eq}$ is function of the conserved moments.
Define a lattice Boltzmann scheme

A lattice Boltzmann scheme is given by

- a set of $q$ velocities adapted to the mesh, $c_0, \ldots, c_{q-1}$,
- an invertible matrix $M$ that transforms the densities into the moments:

$$m_k = \sum_{j=0}^{q-1} M_{kj} f_j = \sum_{j=0}^{q-1} P_k(c_j)f_j, \quad P_k \in \mathbb{R}[X],$$

- functions defining the equilibrium $m_k^{eq}$,
- relaxation parameters $s_k$.

The $n$th first moments are necessary the unknowns of the PDEs: these moments are conserved, that is

$$m_k^{eq} = m_k, \quad 0 \leq k \leq n - 1.$$

The next moments are not conserved and their equilibrium value depends on the conserved moments:

$$m_k^{eq} = m_k^{eq}(m_0, \ldots, m_{n-1}), \quad n \leq k \leq q - 1.$$
Description of the lattice Boltzmann method – Classical schemes

Examples

<table>
<thead>
<tr>
<th>$D_1 Q_2$</th>
<th>$D_1 Q_3$</th>
<th>$D_1 Q_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>advection, heat</td>
<td>advection, wave, heat</td>
<td>Euler</td>
</tr>
<tr>
<td><img src="image1.png" alt="Diagram" /></td>
<td><img src="image2.png" alt="Diagram" /></td>
<td><img src="image3.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$D_2 Q_4$</th>
<th>$D_2 Q_5$</th>
<th>$D_2 Q_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>advection, heat</td>
<td>advection, wave, heat</td>
<td>Navier-Stokes</td>
</tr>
<tr>
<td><img src="image4.png" alt="Diagram" /></td>
<td><img src="image5.png" alt="Diagram" /></td>
<td><img src="image6.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>
The $D_1Q_2$ can be used to simulate the mono-dimensional hyperbolic conservative equation

$$\partial_t u(t, x) + \partial_x \varphi(u)(t, x) = 0, \quad t > 0, \ x \in \mathbb{R},$$

where $u : \mathbb{R} \to \mathbb{R}$ is the unknown.

The $D_1Q_2$ is given by

- two velocities $\{-\lambda, \lambda\}$ with $\lambda = \Delta x / \Delta t$ and the associated densities $(f_-, f_+)$
- the matrix $M$ and its inverse
  $$M = \begin{pmatrix} 1 & 1 \\ -\lambda & \lambda \end{pmatrix} \quad M^{-1} = \begin{pmatrix} 1/2 & -1/(2\lambda) \\ 1/2 & 1/(2\lambda) \end{pmatrix}$$
- the conserved moment $u$ and the non conserved moment $v$, where $u = f_- + f_+$ and $v = -\lambda f_- + \lambda f_+$.
- the equilibrium value $v^{eq} = v^{eq}(u)$ and the relaxation parameter $s$. 
Description of the lattice Boltzmann method – Analysis methods

One time step of the linear $D_1Q_2$

In the linear case ($v^{eq} = \alpha u$), one time step of the scheme reads

\[
\begin{align*}
    v^*(x, t) &= (1 - s)v(x, t) + s\alpha u(x, t), & \text{relaxation,} \\
    f_-(x, t + \Delta t) &= f_*(x + \Delta x, t), & \text{transport to the left,} \\
    f_+(x, t + \Delta t) &= f_*(x - \Delta x, t), & \text{transport to the right.}
\end{align*}
\]

In terms of densities, it reads

\[
\begin{align*}
    f_-(x, t + \Delta t) &= \frac{1}{2} \left( 2 - s - \frac{s\alpha}{\lambda} \right) f_-(x + \Delta x, t) + \frac{1}{2} \left( s - \frac{s\alpha}{\lambda} \right) f_+(x + \Delta x, t), \\
    f_+(x, t + \Delta t) &= \frac{1}{2} \left( s + \frac{s\alpha}{\lambda} \right) f_-(x + \Delta x, t) + \frac{1}{2} \left( 2 - s + \frac{s\alpha}{\lambda} \right) f_+(x + \Delta x, t)
\end{align*}
\]

In terms of moments, it reads

\[
\begin{align*}
    u_{j+1}^{n+1} &= \frac{1}{2} \left( 1 - s \frac{\alpha}{\lambda} \right) u_{j+1}^n + \frac{1}{2} \left( 1 + s \frac{\alpha}{\lambda} \right) u_{j-1}^n - \frac{1-s}{2\lambda} (v_{j+1}^n - v_{j-1}^n), \\
    v_{j+1}^{n+1} &= \frac{1-s}{2} (v_{j+1}^n + v_{j-1}^n) - \frac{\lambda}{2} \left( 1 - s \frac{\alpha}{\lambda} \right) u_{j+1}^n + \frac{\lambda}{2} \left( 1 + s \frac{\alpha}{\lambda} \right) u_{j-1}^n.
\end{align*}
\]
Assuming that the densities are regular functions, the Taylor expansion method for small $\Delta t$ and $\Delta x$ yields

**Zeroth order**

At the order $o$, the distribution are at equilibrium

$$f_j = f_j^{eq} + O(\Delta t), \quad f_j^* = f_j^{eq} + O(\Delta t), \quad j \in \{-, +\}.$$  

We have

$$f_j(t + \Delta t, x) = f_j^*(t, x - v_j \Delta t), \quad v_j = j \lambda, \quad j \in \{-, +\},$$

$$f_j + O(\Delta t) = f_j^* + O(\Delta t),$$

$$v = v^* + O(\Delta t) = (1 - s)v + sv^{eq} + O(\Delta t),$$

$$v = v^{eq} + O(\Delta t) \quad \text{and} \quad v^* = v^{eq} + O(\Delta t).$$
**First order**

The first moment $u$ satisfies the partial differential equation
\[ \partial_t u + \partial_x v^{\text{eq}} = \mathcal{O}(\Delta t). \]

The choice $v^{\text{eq}} = \varphi(u)$ is then done so that $u$ satisfies the conservative equation at order 1.

---

**Transition lemma**

The second moment $v$ satisfies
\[ v = v^{\text{eq}} - \frac{\Delta t}{s} \theta + \mathcal{O}(\Delta t^2), \]
\[ v^* = v^{\text{eq}} + \Delta t \left(1 - \frac{1}{s}\right) \theta + \mathcal{O}(\Delta t^2), \]
with
\[ \theta = \partial_t v^{\text{eq}} + \lambda^2 \partial_x u. \]

---

We have
\[ f_j (t + \Delta t, x) = f_j^* (t, x - v_j \Delta t), \]
\[ f_j + \Delta t \partial_t f_j = f_j^* - v_j \Delta t \partial_x f_j^* + \mathcal{O}(\Delta t^2), \]
\[ f_j + \Delta t \partial_t f_j^{\text{eq}} = f_j^* - v_j \Delta t \partial_x f_j^{\text{eq}} + \mathcal{O}(\Delta t^2). \]

Taking the first moment:
\[ u + \Delta t \partial_t u = u - \Delta t \partial_x v^{\text{eq}} + \mathcal{O}(\Delta t^2). \]

Taking the second moment:
\[ v + \Delta t \partial_t v^{\text{eq}} = v^* - \lambda^2 \Delta t \partial_x u + \mathcal{O}(\Delta t^2) \]
\[ = (1 - s) v + s v^{\text{eq}} - \lambda^2 \Delta t \partial_x u + \mathcal{O}(\Delta t^2) \]
Equivalent equations (2)

**Second order**

The first moment $u$ satisfies the second-order partial differential equation

$$\partial_t u + \partial_x \varphi(u) = \Delta t \sigma \partial_x \left[ (\lambda^2 - \varphi'(u)^2) \partial_x u \right] + \mathcal{O}(\Delta t^2),$$

with $\sigma = 1/s - 1/2$.

Taking the first moment at second order:

$$u + \Delta t \partial_t u + \frac{1}{2} \Delta t^2 \partial_{tt} u = u - \Delta t \partial_x v^* + \frac{1}{2} \lambda^2 \Delta t^2 \partial_{xx} u + \mathcal{O}(\Delta t^3)$$

$$\partial_t u + \partial_x \varphi(u) = \partial_x (v^\text{eq} - v^*) + \frac{1}{2} \Delta t (\lambda^2 \partial_{xx} u - \partial_{tt} u) + \mathcal{O}(\Delta t^2)$$

$$v^\text{eq} - v^* = \Delta t (\frac{1}{s} - 1) \theta \quad \theta = \varphi'(u) \partial_t u + \lambda^2 \partial_{xx} u = (\lambda^2 - \varphi'(u)^2) \partial_{xx} u$$
Description of the lattice Boltzmann method – Analysis methods

Maximum principle

**Theorem**

We assume that \( s \in [0, 1], u_j^0 \in [\alpha, \beta], v_j^0 = \varphi(u_j^0), \forall j, \) and \( \lambda \geq \max_{\alpha \leq u \leq \beta} |\varphi'(u)|. \) Then \( \forall n \geq 0 \ \forall j \ u_j^n \in [\alpha, \beta]. \)

The functions \( h^\pm(u) = \frac{\lambda u^\pm \varphi(u)}{2\lambda} \) are increasing. We then have

\[
f_{\pm,j}^0 \in [h^\pm(\alpha), h^\pm(\beta)].
\]

The relaxation phase reads as a linear convex combination:

\[
f_{\pm,j}^{n*} = \frac{\lambda u_j^n \pm v_j^n}{2\lambda} + \frac{v_j^{n*} - v_j^n}{2\lambda} = f_{\pm,j}^n \pm s \frac{\varphi(u_j^n) - v_j^n}{2\lambda} = (1 - s)f_{\pm,j}^n + sh^\pm(u_j^n) \in [h^\pm(\alpha), h^\pm(\beta)].
\]

And

\[
u_j^n = f_{-,j}^n + f_{+,j}^n \in [h^-(\alpha) + h^+(\alpha), h^-(\beta) + h^+(\beta)] = [\alpha, \beta].
\]
Theorem
We assume that \( v^{eq} = \alpha u \), \( \lambda \geq |\alpha| \), and \( s \in [0, 2] \). Then the D_1Q_2 scheme is stable for the \( L^2 \)-norm.

The amplification matrix of the linear D_1Q_2 scheme is given by

\[
G(\Delta x, \xi) = \begin{pmatrix}
(1 - \frac{s}{2} (1 + \frac{\alpha}{\lambda})) e^{-i \Delta x \xi} & \frac{s}{2} (1 - \frac{\alpha}{\lambda}) e^{-i \Delta x \xi} \\
\frac{s}{2} (1 + \frac{\alpha}{\lambda}) e^{i \Delta x \xi} & (1 - \frac{s}{2} (1 - \frac{\alpha}{\lambda})) e^{i \Delta x \xi}
\end{pmatrix}.
\]
Description of the lattice Boltzmann method – Boundary conditions

Bounce Back and anti Bounce Back

The black point has outside neighbors. For each one, we have to specify the distribution functions with incoming velocities.
Bounce Back and anti Bounce Back

When a fluid particle (discrete distribution function) reaches a boundary node, the particle will scatter back to the fluid along with its incoming direction. For anti Bounce Back condition, the sign is changed. The bound of the domain is then pixelized.
In order to improve the accuracy of the boundary conditions, the interface need to be localized more precisely.
**Description of the lattice Boltzmann method – Boundary conditions**

**Bouzidi type boundary conditions**

Case $s < 1/2$ (explicit interpolation).

We have

$$x_{i - \frac{1}{2}} = (1 - 2s)x_{i - 1} + 2sx_i.$$  

Then

$$f_{i+1}^- = \tilde{f}_i^- = f_{i - \frac{1}{2}}^+ = (1 - 2s)f_{i - 1}^+ + 2sf_i^+.$$
Description of the lattice Boltzmann method – Boundary conditions

Bouzidi type boundary conditions

Case $s > 1/2$ (implicit interpolation).

We have

$$x_i = \frac{1}{2s} x_{i+\frac{1}{2}} + \frac{2s - 1}{2s} x_{i-1}.$$ 

Then

$$f_{i+1}^- = \tilde{f}_i^- = \frac{1}{2s} \tilde{f}_{i+\frac{1}{2}}^- + \frac{2s - 1}{2s} \tilde{f}_{i-1}^- = \frac{1}{2s} f_i^- + \frac{2s - 1}{2s} f_i^-.$$
The Lattice Boltzmann method

1. Description of the lattice Boltzmann method

2. pyLBM (collaboration with Loïc Gouarin)
   - Presentation
   - Examples

3. The vectorial schemes

4. Conclusion
Motivations

pyLBM

- academic and flexible code
  - to test and to compare the schemes: simple and brief implementation
  - can be used by non specialists or students
- optimisations: mpi, openmp, cython (cuda and opencl in test)

$\Rightarrow$ python: compromise efficiency / simplicity

- fine management of the geometry built from simple objects
  - 2D: circle, ellipse, triangle, parallelogram
  - 3D: sphere, ellipsoid, parallelepiped, cylinder

The user defines
- the geometry of the domain and the scheme through a dictionary,
- initial conditions and boundary conditions through functions,
- parameters for the optimization (optional),

the code is then generated and executed.
pyLBM (collaboration with Loïc Gouarin) – Presentation

Structure of the code

- Circle
- Ellipse
- Parallelogram
- Triangle
- Sphere
- Ellipsoid
- Cylinder
- Parallelepiped

Dictionary

Simulation

Geometry
- Circle
- Ellipse
- Parallelogram
- Triangle
- Sphere
- Ellipsoid
- Cylinder
- Parallelepiped

Stencil

Velocity

Domain

Scheme

Generator

Array: f, m

Functions:
- transport
- relaxation
- m2f f2m

Result

Initialization

Boundary conditions
Geometry

```python
import pyLBM

xmin, xmax = 0., 1.  # domain
dic = {
    'box': {'x': [xmin, xmax]},
}

geom = pyLBM.Geometry(dic)
print(geom)
geom.visualize()
```

Geometry informations
- spatial dimension: 1
- bounds of the box:
  - [[ 0.  1.]]
**Stencil**

```python
import pyLBM
dic = {
    'dim': 1,
    'schemes': [
        {'velocities': [1, 2],
        ]},
}
sten = pyLBM.Stencil(dic)
print(sten)
sten.visualize()
```

**Stencil informations**
- spatial dimension: 1
- maximal velocity in each direction: [1]
- minimal velocity in each direction: [-1]
- Informations for each elementary stencil:
  - stencil 0
    - number of velocities: 2
    - velocities: (1: 1), (2: -1),
```

```

```

```

```
Domain informations
spatial dimension: 1
space step: dx= 1.000e-01

D₁Q₂ for the advection

```python
import pyLBM
xmin, xmax = 0., 1. # domain
N = 10 # number of points

dic = {
    'box': {'x': [xmin, xmax]},
    'space_step': (xmax-xmin)/N,
    'scheme_velocity': 1.,
    'schemes': [
        {'velocities': [1, 2],
        }
    ],
}

dom = pyLBM.Domain(dic)
print(dom)
dom.visualize()
```
import pyLBM
import sympy as sp

u, X = sp.symbols('u, X')
c = .25  # velocity
s = 1.  # relaxation parameter
dic = {
    'dim': 1,
    'scheme_velocity': 1.,
    'schemes': [
        {'velocities': [1, 2],
         'conserved_moments': u,
         'polynomials': [1, X],
         'equilibrium': [u, c*u],
         'relaxation_parameters': [0., s],
         'init': {u: 0.}
    ]
},
}
scheme = pyLBM.Scheme(dic)
print(scheme)
pyLBM (collaboration with Loïc Gouarin) – Examples

D_1Q_2 for the advection (numerical results)

<table>
<thead>
<tr>
<th>k</th>
<th>s</th>
<th>2.000</th>
<th>1.900</th>
<th>1.750</th>
<th>1.000</th>
<th>0.750</th>
<th>0.500</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td>1.536e-01</td>
<td>1.416e-01</td>
<td>1.256e-01</td>
<td>8.104e-02</td>
<td>7.881e-02</td>
<td>8.113e-02</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>1.733e-01</td>
<td>1.714e-01</td>
<td>1.712e-01</td>
<td>2.062e-01</td>
<td>2.288e-01</td>
<td>2.550e-01</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>1.319e-01</td>
<td>1.153e-01</td>
<td>1.073e-01</td>
<td>1.495e-01</td>
<td>1.757e-01</td>
<td>2.100e-01</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>4.897e-02</td>
<td>4.697e-02</td>
<td>5.138e-02</td>
<td>1.145e-01</td>
<td>1.405e-01</td>
<td>1.719e-01</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>1.254e-02</td>
<td>1.429e-02</td>
<td>2.162e-02</td>
<td>7.983e-02</td>
<td>1.049e-01</td>
<td>1.357e-01</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>3.113e-03</td>
<td>4.850e-03</td>
<td>9.913e-03</td>
<td>4.990e-02</td>
<td>7.081e-02</td>
<td>9.927e-02</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>7.761e-04</td>
<td>1.991e-03</td>
<td>4.836e-03</td>
<td>2.863e-02</td>
<td>4.329e-02</td>
<td>6.599e-02</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>1.943e-04</td>
<td>9.263e-04</td>
<td>2.412e-03</td>
<td>1.551e-02</td>
<td>2.448e-02</td>
<td>3.990e-02</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>4.863e-05</td>
<td>4.522e-04</td>
<td>1.208e-03</td>
<td>8.096e-03</td>
<td>1.311e-02</td>
<td>2.233e-02</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>1.216e-05</td>
<td>2.241e-04</td>
<td>6.041e-04</td>
<td>4.138e-03</td>
<td>6.794e-03</td>
<td>1.188e-02</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>3.039e-06</td>
<td>1.117e-04</td>
<td>3.022e-04</td>
<td>2.092e-03</td>
<td>3.461e-03</td>
<td>6.136e-03</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>7.598e-07</td>
<td>5.577e-05</td>
<td>1.512e-04</td>
<td>1.052e-03</td>
<td>1.747e-03</td>
<td>3.121e-03</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>1.900e-07</td>
<td>2.787e-05</td>
<td>7.559e-05</td>
<td>5.277e-04</td>
<td>8.778e-04</td>
<td>1.574e-03</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>4.749e-08</td>
<td>1.393e-05</td>
<td>3.780e-05</td>
<td>2.642e-04</td>
<td>4.040e-04</td>
<td>7.904e-04</td>
</tr>
</tbody>
</table>

Smooth solution

slope 2.000e00 1.000e00 9.999e-01 9.979e-01 9.965e-01 9.937e-01

<table>
<thead>
<tr>
<th>k</th>
<th>s</th>
<th>2.000</th>
<th>1.900</th>
<th>1.750</th>
<th>1.000</th>
<th>0.750</th>
<th>0.500</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td>2.722e-01</td>
<td>2.657e-01</td>
<td>2.590e-01</td>
<td>2.649e-01</td>
<td>2.758e-01</td>
<td>2.893e-01</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>8.353e-02</td>
<td>8.611e-02</td>
<td>9.415e-02</td>
<td>1.696e-01</td>
<td>2.027e-01</td>
<td>2.389e-01</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>1.488e-01</td>
<td>1.372e-01</td>
<td>1.304e-01</td>
<td>1.434e-01</td>
<td>1.587e-01</td>
<td>1.832e-01</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>1.055e-01</td>
<td>9.036e-02</td>
<td>8.323e-02</td>
<td>1.066e-01</td>
<td>1.225e-01</td>
<td>1.444e-01</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>8.651e-02</td>
<td>7.416e-02</td>
<td>7.188e-02</td>
<td>9.591e-02</td>
<td>1.082e-01</td>
<td>1.251e-01</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>6.158e-02</td>
<td>4.995e-02</td>
<td>5.070e-02</td>
<td>7.838e-02</td>
<td>8.932e-02</td>
<td>1.038e-01</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>5.668e-02</td>
<td>4.470e-02</td>
<td>4.497e-02</td>
<td>6.609e-02</td>
<td>7.494e-02</td>
<td>8.675e-02</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>4.421e-02</td>
<td>3.434e-02</td>
<td>3.570e-02</td>
<td>5.515e-02</td>
<td>6.270e-02</td>
<td>7.268e-02</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>3.460e-02</td>
<td>2.684e-02</td>
<td>2.954e-02</td>
<td>4.657e-02</td>
<td>5.289e-02</td>
<td>6.125e-02</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>2.710e-02</td>
<td>2.089e-02</td>
<td>2.424e-02</td>
<td>3.909e-02</td>
<td>4.422e-02</td>
<td>5.146e-02</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>2.230e-02</td>
<td>1.732e-02</td>
<td>2.043e-02</td>
<td>3.288e-02</td>
<td>3.735e-02</td>
<td>4.326e-02</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>1.783e-02</td>
<td>1.406e-02</td>
<td>1.707e-02</td>
<td>2.763e-02</td>
<td>3.140e-02</td>
<td>3.637e-02</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>1.403e-02</td>
<td>1.151e-02</td>
<td>1.432e-02</td>
<td>2.324e-02</td>
<td>2.641e-02</td>
<td>3.059e-02</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>1.111e-02</td>
<td>9.517e-03</td>
<td>1.202e-02</td>
<td>1.954e-02</td>
<td>2.220e-02</td>
<td>2.572e-02</td>
</tr>
</tbody>
</table>

Discontinuous solution

slope 3.374e-01 2.746e-01 2.527e-01 2.502e-01 2.501e-01 2.501e-01
The vectorial schemes

The Lattice Boltzmann method

1. Description of the lattice Boltzmann method

2. pyLBM (collaboration with Loïc Gouarin)

3. The vectorial schemes
   • Presentation
   • Numerical tests

4. Conclusion
The vectorial schemes – Presentation

Motivations

We focus on the numerical simulation of hyperbolic systems of pde’s by the lattice Boltzmann method. In this presentation, we deal with mono dimensional conservation laws:

\[
\partial_t u(t, x) + \partial_x \varphi(u)(t, x) = 0, \quad t > 0, \ x \in \mathbb{R},
\]

where \( u \) is the unknown vector of size \( n \).

Goals:
- generic scheme for all flux functions \( \varphi \),
- identifiable stability conditions,
- straightforward treatment of the boundary conditions.
For a system of $n$ equations, we duplicate $n$ times the $D_1Q_2$:

- the velocity of the scheme: $\lambda = \Delta x / \Delta t$
- the velocities of the particles: $v_0 = -\lambda$, $v_1 = \lambda$
- the particles distributions: $f_{1,0}, f_{1,1}, \ldots, f_{n,0}, f_{n,1}$
- the moments: $u_k = f_{k,0} + f_{k,1}$, $v_k = \lambda(-f_{k,0} + f_{k,1})$, $1 \leq k \leq n$
- the relaxation phase: $u_k^* = u_k$, $v_k^* = v_k + s_k(v_{eq} - v_k)$, $1 \leq k \leq n$

**Theorem (Equivalent equation)**

Taking $v_{eq}^k = \varphi_k(u)$, the vector of the first moments $u$ satisfy

$$
\partial_t u + \partial_x \varphi(u) = \Delta t \mathcal{S} \partial_x \left( \left( \lambda^2 I_n - (d \varphi(u))^2 \right) \partial_x u \right) + O(\Delta t^2),
$$

with $\mathcal{S} = \text{diag}(\sigma_1, \ldots, \sigma_n)$, $\sigma_k = 1/s_k - 1/2$, $1 \leq k \leq n$. 
The vectorial schemes – Presentation

The $D_1Q_2,\ldots,2$ as a relaxation scheme

The relaxation system proposed by Jin and Xin reads

$$\begin{cases}
\partial_t u^\varepsilon + \partial_x v^\varepsilon = 0, \\
\partial_t v^\varepsilon + A \partial_x u^\varepsilon = \frac{1}{\varepsilon} (\varphi(u^\varepsilon) - v^\varepsilon).
\end{cases}$$

Denoting $u^n_i = u(x_i, t^n), v^n_i = v(x_i, t^n), x_i \in \mathcal{L}, t^n = n\Delta t$, the $D_1Q_2,\ldots,2$ can be rewritten in the form (if all relaxation parameters are equal to $s$)

$$\begin{align*}
v^{n*}_i &= v^n_i - s(v^n_i - \varphi(u^n_i)), \\
u^{n+1}_i &= \frac{1}{2}(u^{n+1}_i + u^n_{i-1}) - \frac{\Delta t}{2\Delta x} (v^{n*}_{i+1} - v^{n*}_{i-1}), \\
v^{n+1}_i &= \frac{1}{2}(v^{n*}_{i+1} + v^{n*}_{i-1}) - \lambda^2 \frac{\Delta t}{2\Delta x} (u^n_{i+1} - u^n_{i-1}).
\end{align*}$$

Splitting between

- the relaxation part treated by the explicit Euler method with $\varepsilon = \Delta t/s$,
- the hyperbolic part treated by the Lax-Friedrichs method with $A = \lambda^2 I_n$. 

The vectorial schemes – Numerical tests

Numerical tests done with pyLBM

- **dimension 1**
  - advection with constant velocity
  - advection-diffusion-reaction equation
  - Burgers
  - \( p \)-system
  - shallow water system
  - full compressible Euler system

- **dimension 2**
  - advection with constant velocity
  - shallow water system
  - incompressible thermo-hydrodynamic (Boussinesq approximation)
  - magneto-hydrodynamic system

- **dimension 3**
  - advection with constant velocity
  - incompressible thermo-hydrodynamic (Boussinesq approximation)
The Lattice Boltzmann method

1. Description of the lattice Boltzmann method
2. pyLBM (collaboration with Loïc Gouarin)
3. The vectorial schemes
4. Conclusion
Conclusion

The Lattice Boltzmann method

- The lattice Boltzmann method and the Boltzmann equation
  - very rough discretization in term of velocities
  - no claim to solve the Boltzmann equation
  - a taylor expansion to mimic the Chapman-Enskog expansion

- The software pyLBM
  - flexible and simple interface to create a simulation
  - performances of a compiled langage
  - some features to add: optimization with graphics cards, relative velocity schemes (Tony Février), mesh refinement...

- The vectorial schemes
  - simple way to simulate conservative hyperbolic problems
  - link with the relaxation methods
  - to do: obtain a convergence theorem with this restrictive framework