Link-wise boundary conditions for LBM

Francesco Marson

University of Geneva

Palabos Summer School 2021

8th June 2021

The goals of this session

PALABOS

Goal

To understand and use *link-wise (directional) boundary conditions.*

Ingredients

1. General link-wise equation;
2. Generalized geometrical interpretation.

- Derive specific schemes in the literature from the formula;
- Know how analyze them and analyze the errors;
- "Move" boundaries and compute the momentum exchange;
- Use some schemes in Palabos.

Difference with the 2020 session

2020

- Overview of most off-lattice conditions;
- Classification;
- Interpolated bounce-back [LW];
- Partially saturated BB;
- Overview of ghost methods [LW];
- Computation of the drag;
- Immersed Boundary Method.

Download the 2020 slides: ▸ Slides 2020
Watch the 2020 recording: ▸ Youtube, lecture 2020

2021

- Equation of the link-wise approach;
- Overview of link-wise schemes (LW);
- Numerical analysis;
- Palabos implemenation:
  - → Voxelization;
  - → Boundary from stl file;
  - → Generate simple triangulate surfaces.

Useful material

▸ ELI article, RG ▸ ELI article, download! ▸ LBM book
▸ Palabos forum

Do we need curved off-lattice boundary conditions?

PALABOS

- The LBM is inherently connected with its *structured lattice*;
  - → we need a curved boundary condition (off-lattice scheme);
  - → curved boundary conditions integrate off-lattice information in the LBM nodes.
- NO off-lattice scheme ⇒ stair-cased boundary.

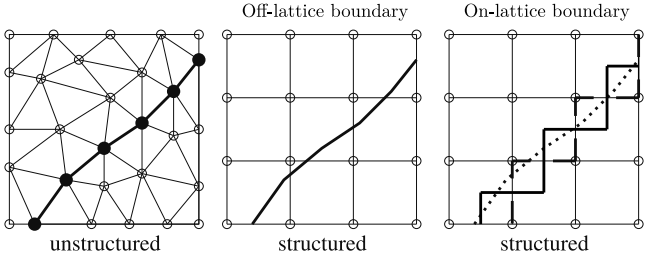


Figure: Structured and unstructured meshes (inspired by Krüger et al., 2017)

The Boltzmann Equation (repetita iuvant)

PALABOS

Continuum BGK-Boltzmann Equation

$$\partial_t f + \partial_\alpha \xi_\alpha f = \frac{f - f^{\mathrm{eq}}}{\tau}$$
$$f^{\mathrm{eq}}(\rho, \boldsymbol{u}, \theta, \boldsymbol{\xi}) = \frac{\rho}{(2\pi\theta)^{d/2}} \mathrm{e}^{-(\boldsymbol{\xi}-\boldsymbol{u})^2/(2\theta)} \quad (1)$$

BGK-Discrete Velocities Boltzmann Equation

$$\partial_t f_i + \partial_\alpha \xi_\alpha f_i = \frac{f_i - f_i^{\mathrm{eq}}}{\tau}\,. \quad (2)$$

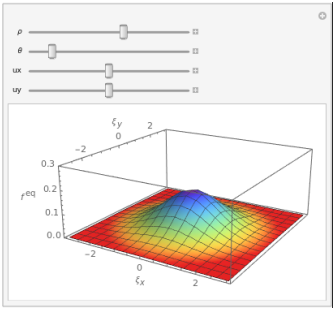


Figure: Maxwell-Boltzamann 2D

The Boltzmann Equation (repetita iuvant)

PALABOS

Continuum BGK-Boltzmann Equation

$$\partial_t f + \partial_\alpha \xi_\alpha f = \frac{f - f^{\mathrm{eq}}}{\tau}$$
$$f^{\mathrm{eq}}(\rho, \boldsymbol{u}, \theta, \boldsymbol{\xi}) = \frac{\rho}{(2\pi\theta)^{d/2}} \mathrm{e}^{-(\boldsymbol{\xi}-\boldsymbol{u})^2/(2\theta)} \quad (1)$$

BGK-Discrete Velocities Boltzmann Equation

$$\partial_t f_i + \partial_\alpha \xi_\alpha f_i = \frac{f_i - f_i^{\mathrm{eq}}}{\tau}\,. \quad (2)$$

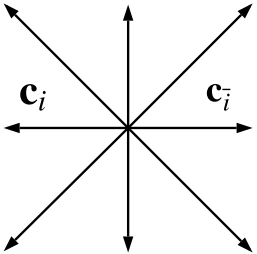


Figure: Discrete Velocities

The Lattice Boltzmann Equation (repetita iuvant)

PALABOS

BGK-LBE

$$f_i(\boldsymbol{x} + \boldsymbol{c}_i,\ t+1) = f_i(\boldsymbol{x}, t) - \frac{f_i - f_i^{\mathrm{eq}}}{\tau} \tag{3a}$$

$$f_{i,2}^{\mathrm{eq}} = w_i \rho \left[1 + \frac{c_{i\alpha_1} u_{\alpha_1}}{c_{\mathrm{s}}^2} + \frac{u_{\alpha_1} u_{\alpha_2} \left(c_{i\alpha_1} c_{i\alpha_2} - c_{\mathrm{s}}^2 \delta_{\alpha_1 \alpha_2} \right)}{2 c_{\mathrm{s}}^4} \right] \tag{3b}$$

TRT-LBE

$$f_i^{\pm}(\boldsymbol{x} + \boldsymbol{c}_i,\ t+1) = f_i^{\pm}(\boldsymbol{x}, t) - \frac{f_i^{\pm} - f_i^{\pm \mathrm{eq}}}{\tau^{\pm}} \tag{4a}$$

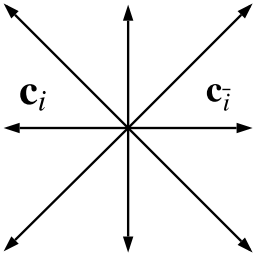


Figure: Discrete Velocities

The Lattice Boltzmann Equation (repetita iuvant)

PALABOS

BGK-LBE

$$f_i(\boldsymbol{x} + \boldsymbol{c}_i,\ t+1) = f_i(\boldsymbol{x}, t) - \frac{f_i - f_i^{\mathrm{eq}}}{\tau} \tag{3a}$$

$$f_{i,2}^{\mathrm{eq}} = w_i \rho \left[1 + \frac{c_{i\alpha_1} u_{\alpha_1}}{c_{\mathrm{s}}^2} + \frac{u_{\alpha_1} u_{\alpha_2} \left(c_{i\alpha_1} c_{i\alpha_2} - c_{\mathrm{s}}^2 \delta_{\alpha_1 \alpha_2} \right)}{2 c_{\mathrm{s}}^4} \right] \tag{3b}$$

TRT-LBE

$$f_i^{\pm}(\boldsymbol{x} + \boldsymbol{c}_i,\ t+1) = f_i^{\pm}(\boldsymbol{x}, t) - \frac{f_i^{\pm} - f_i^{\pm \mathrm{eq}}}{\tau^{\pm}} \tag{4a}$$

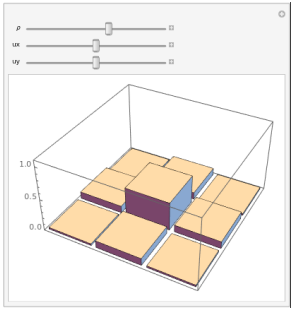


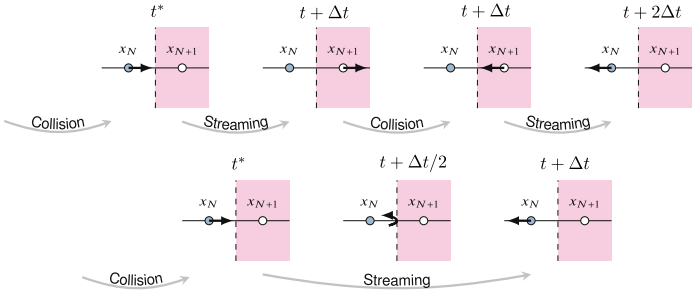
Figure: Stokes Maxwell-Boltzamann 2D

Summary of straight Bounce-back (repetita iuvant)

PALABOS

Bounce-back (boundary located at half-way)

1. On-lattice bounce-back: first order accurate (NB: the at half-way nevertheless!)
2. Half-way bounce-back: up to third order accurate (Ginzburg & Adler, 1994)



(inspired by Krüger et al., 2017)

Half-Way bounce-back with curved geometries

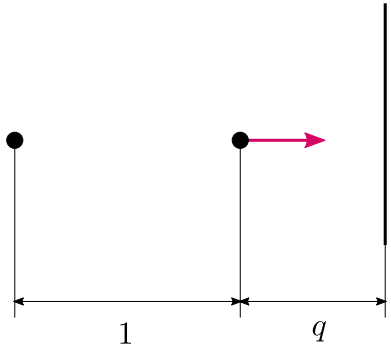


Figure: Linearly interpolated bounce back.

Half-Way bounce-back with curved geometries

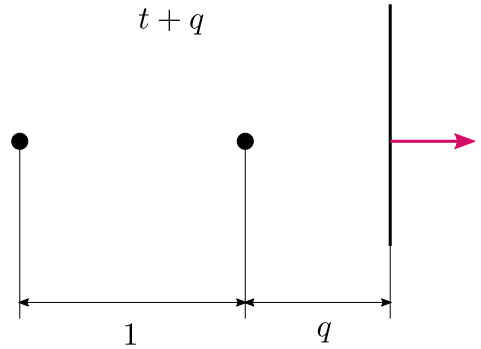


Figure: Linearly interpolated bounce back.

Half-Way bounce-back with curved geometries

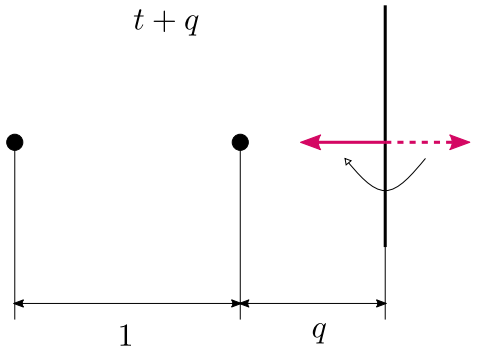


Figure: Linearly interpolated bounce back.

Half-Way bounce-back with curved geometries

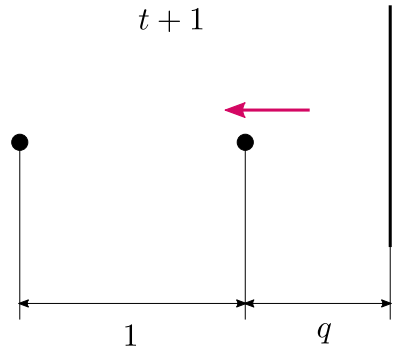


Figure: Linearly interpolated bounce back.

Half-Way bounce-back with curved geometries

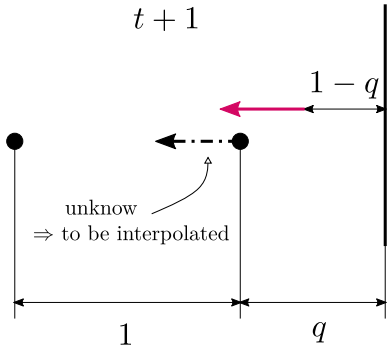


Figure: Linearly interpolated bounce back.

General formula (1)

$$f_i(\boldsymbol{x}_F, t+1) = \sum_k a_k f_i^*(\boldsymbol{x}_k, t) + \sum_j a_j f_i^*(\boldsymbol{x}_j, t) + a_w f_W^{\mathrm{eq}-} + K \quad (5)$$

🕮 Bibliografy: seminal works

Ginzburg and d'Humières (2003); Ginzburg and Verhaeghe (2008)

⚠ Notation

The notation of the indexes i and $\overline{i}$ it is not unique. Some articles use the opposite notation!

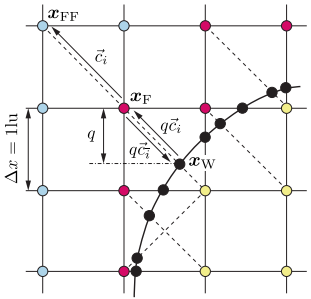


Figure: (Marson et al., 2021)

General formula (1)

$$f_i(\boldsymbol{x}_F, t+1) = \sum_k a_k f_i^*(\boldsymbol{x}_k, t) + \sum_j a_j f_i^*(\boldsymbol{x}_j, t) + a_w f_W^{eq-} + K \quad (5)$$

🕮 Bibliografy: seminal works

Ginzburg and d'Humières (2003); Ginzburg and Verhaeghe (2008)

⚠ Notation

The notation of the indexes i and $\overline{i}$ it is not unique. Some articles use the opposite notation!

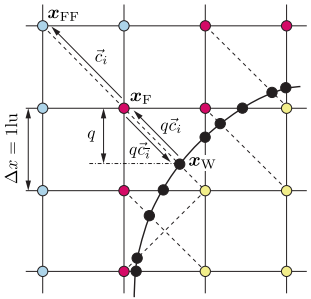


Figure: (Marson et al., 2021)

General formula (1)

$$f_i(\boldsymbol{x}_F, t+1) = \sum_k a_k f_{\bar{i}}^*(\boldsymbol{x}_k, t) + \sum_j a_j f_i^*(\boldsymbol{x}_j, t) + a_w f_W^{eq-} + K \quad (5)$$

🕮 Bibliografy: seminal works

Ginzburg and d'Humières (2003); Ginzburg and Verhaeghe (2008)

⚠ Notation

The notation of the indexes i and $\bar{i}$ it is not unique. Some articles use the opposite notation!

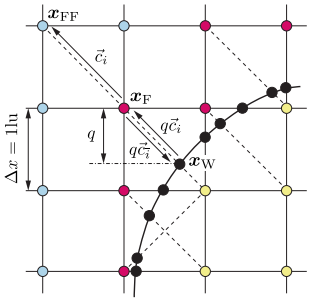


Figure: (Marson et al., 2021)

General formula (1)

PALABOS

$$f_i(\boldsymbol{x}_F, t+1) = \sum_k a_k f_i^*(\boldsymbol{x}_k, t) + \sum_j a_j f_i^*(\boldsymbol{x}_j, t) + a_w f_W^{\mathrm{eq}-} + K \quad (5)$$

🕮 Bibliografy: seminal works

Ginzburg and d'Humières (2003); Ginzburg and Verhaeghe (2008)

⚠ Notation

The notation of the indexes i and $\overline{i}$ it is not unique. Some articles use the opposite notation!

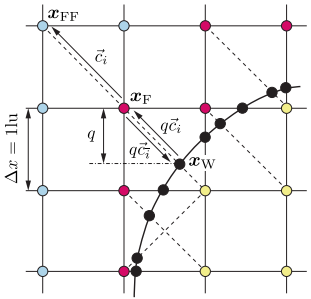


Figure: (Marson et al., 2021)

General formula (1)

$$f_i(\boldsymbol{x}_F, t+1) = \sum_k a_k f_i^*(\boldsymbol{x}_k, t) + \sum_j a_j f_i^*(\boldsymbol{x}_j, t) + a_w f_W^{eq} + K \quad (5)$$

Bibliografy: seminal works

Ginzburg and d'Humières (2003); Ginzburg and Verhaeghe (2008)

Notation

The notation of the indexes i and $\overline{i}$ it is not unique. Some articles use the opposite notation!

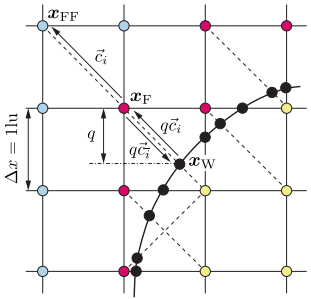


Figure: (Marson et al., 2021)

General formula (1)

$$f_i(\boldsymbol{x}_F, t+1) = \sum_k a_k f_i^*(\boldsymbol{x}_k, t) + \sum_j a_j f_i^*(\boldsymbol{x}_j, t) + a_w f_W^{\mathrm{eq}-} + K \quad (5)$$

🕮 Bibliografy: seminal works

Ginzburg and d'Humières (2003); Ginzburg and Verhaeghe (2008)

⚠ Notation

The notation of the indexes i and $\overline{i}$ it is not unique. Some articles use the opposite notation!

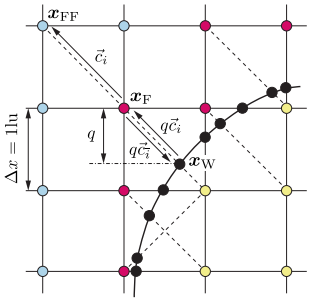
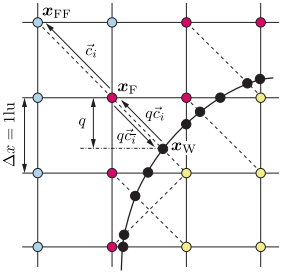


Figure: (Marson et al., 2021)

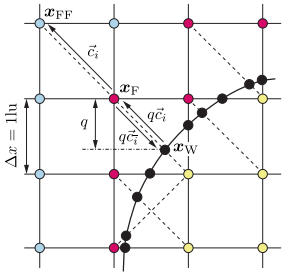
General formula (2)

$$
\begin{aligned}
f_i(\boldsymbol{x}_F, t+1) = \; & a_{k+1}\left[f_i^*(\boldsymbol{x}_{FF}) + 2f^{\mathrm{eq}-}(\boldsymbol{x}_W)\right] + \\
& + a_k\left[f_i^*(\boldsymbol{x}_F) + 2f^{\mathrm{eq}-}(\boldsymbol{x}_W)\right] + \\
& + a_{k-1}\underbrace{\left[\tilde{f}_i^*(\boldsymbol{x}_S) + 2f^{\mathrm{eq}-}(\boldsymbol{x}_W)\right]}_{\text{FH,MLS}} \\
& + a_j f_i^*(\boldsymbol{x}_F) + a_{j+1} f_i^*(\boldsymbol{x}_{FF}) \\
& + a_{j+2}\underbrace{\tilde{f}_i(\boldsymbol{x}_W, t+1)}_{\text{Tao et al.}} + a_{j+3}\underbrace{\tilde{f}_i^*(\boldsymbol{x}_W)}_{\text{ELI}} \\
& - \underbrace{K\left(\tau^-, f_i^{\mathrm{neq}-}(\boldsymbol{x}_F), \ldots\right)}_{\text{parametrization}},
\end{aligned}
$$



General formula (2)

$$
\begin{aligned}
f_i(\boldsymbol{x}_F, t+1) = \; & a_{k+1}\left[f_i^*(\boldsymbol{x}_{FF}) + 2f^{\mathrm{eq}-}(\boldsymbol{x}_W)\right] + \\
& + a_k\left[f_i^*(\boldsymbol{x}_F) + 2f^{\mathrm{eq}-}(\boldsymbol{x}_W)\right] + \\
& + a_{k-1}\underbrace{\left[\tilde{f}_i^*(\boldsymbol{x}_S) + 2f^{\mathrm{eq}-}(\boldsymbol{x}_W)\right]}_{\text{FH,MLS}} \\
& + a_j f_i^*(\boldsymbol{x}_F) + a_{j+1} f_i^*(\boldsymbol{x}_{FF}) \\
& + a_{j+2}\underbrace{\tilde{f}_i(\boldsymbol{x}_W, t+1)}_{\text{Tao et al.}} + a_{j+3}\underbrace{\tilde{f}_i^*(\boldsymbol{x}_W)}_{\text{ELI}} \\
& - \underbrace{K\left(\tau^-, f_i^{\mathrm{neq}-}(\boldsymbol{x}_F), \ldots\right)}_{\text{parametrization}},
\end{aligned}
$$

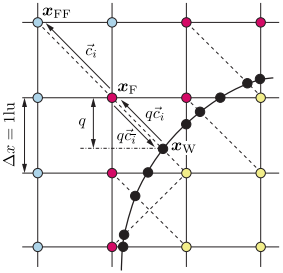


⚠ Warnings

- The notation for the coefficients $a_{j,k}$ is varies a lot in the literature.
- The term $a_W f_W^{\mathrm{eq}-}$ has been distributed among multiple terms.

General formula (2)

$$
\begin{aligned}
f_i(\boldsymbol{x}_F, t+1) = \; & a_{k+1}\left[f_i^*(\boldsymbol{x}_{FF}) + 2f^{\mathrm{eq}-}(\boldsymbol{x}_W)\right] + \\
& + a_k\left[f_i^*(\boldsymbol{x}_F) + 2f^{\mathrm{eq}-}(\boldsymbol{x}_W)\right] + \\
& + a_{k-1}\underbrace{\left[\tilde{f}_i^*(\boldsymbol{x}_S) + 2f^{\mathrm{eq}-}(\boldsymbol{x}_W)\right]}_{\text{FH,MLS}} \\
& + a_j f_i^*(\boldsymbol{x}_F) + a_{j+1} f_i^*(\boldsymbol{x}_{FF}) \\
& + a_{j+2}\underbrace{\tilde{f}_i(\boldsymbol{x}_W, t+1)}_{\text{Tao et al.}} + a_{j+3}\underbrace{\tilde{f}_i^*(\boldsymbol{x}_W)}_{\text{ELI}} \\
& - \underbrace{K\left(\tau^-, f_i^{\mathrm{neq}-}(\boldsymbol{x}_F), \ldots\right)}_{\text{parametrization}},
\end{aligned}
$$



To think about

Is there a better way of identifying populations and coefficients?

Geometrical interpretation: simple example, linear interpolation

Consider the linear case

$$f_i(\boldsymbol{x}_F, t+1) = a_1 f_i^*(\boldsymbol{x}_{FF}) + \underbrace{a_2 f_i^*(\boldsymbol{x}_F)}_{\text{HW}} + a_3 f_i^*(\boldsymbol{x}_F) + K \tag{6}$$

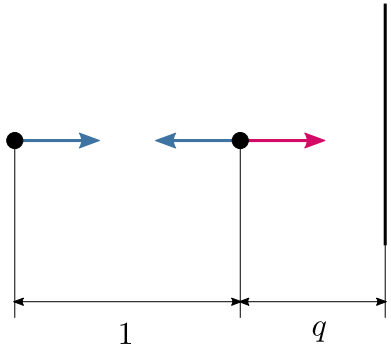


Figure: Linearly interpolated bounce back.

Geometrical interpretation: simple example, linear interpolation

Consider the linear case

$$f_i(\boldsymbol{x}_F, t+1) = a_1 f_i^*(\boldsymbol{x}_{FF}) + \underbrace{a_2 f_i^*(\boldsymbol{x}_F)}_{\text{HW}} + a_3 f_i^*(\boldsymbol{x}_F) + K \tag{6}$$

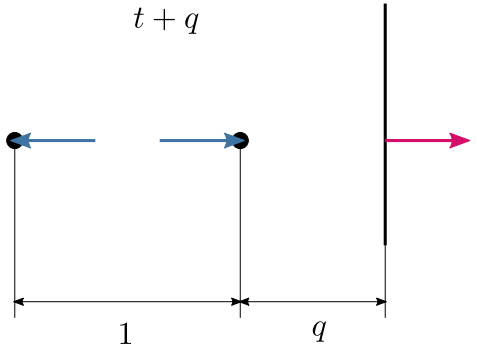


Figure: Linearly interpolated bounce back.

Geometrical interpretation: simple example, linear interpolation

Consider the linear case

$$f_i(\boldsymbol{x}_F, t+1) = a_1 f_i^*(\boldsymbol{x}_{FF}) + \underbrace{a_2 f_i^*(\boldsymbol{x}_F)}_{\text{HW}} + a_3 f_i^*(\boldsymbol{x}_F) + K \tag{6}$$

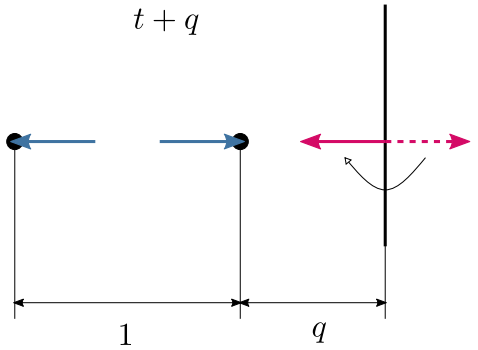


Figure: Linearly interpolated bounce back.

Geometrical interpretation: simple example, linear interpolation

PALABOS

Consider the linear case

$$f_i(\boldsymbol{x}_F, t+1) = a_1 f_i^*(\boldsymbol{x}_{FF}) + \underbrace{a_2 f_i^*(\boldsymbol{x}_F)}_{\text{HW}} + a_3 f_i^*(\boldsymbol{x}_F) + K \tag{6}$$

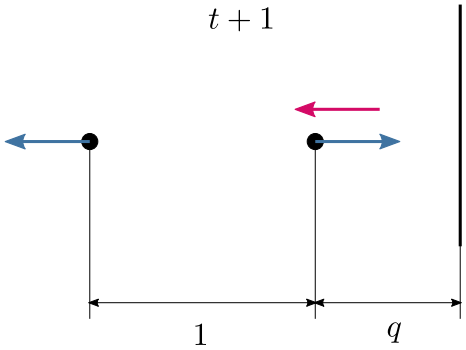


Figure: Linearly interpolated bounce back.

Geometrical interpretation: simple example, linear interpolation

Consider the linear case

$$f_i(\boldsymbol{x}_F, t+1) = a_1 f_i^*(\boldsymbol{x}_{FF}) + \underbrace{a_2 f_i^*(\boldsymbol{x}_F)}_{\text{HW}} + a_3 f_i^*(\boldsymbol{x}_F) + K \tag{6}$$

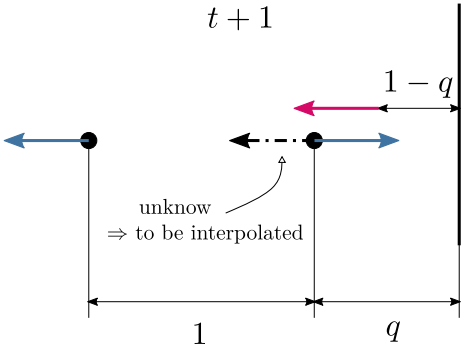
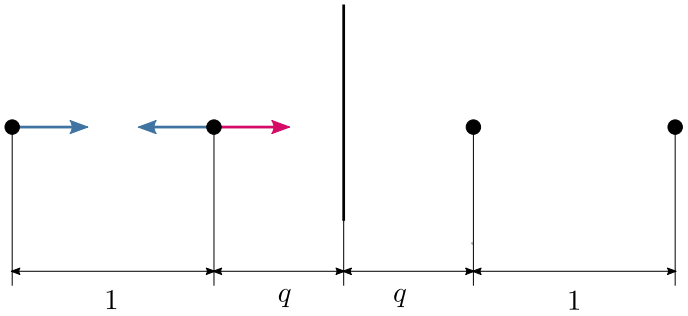


Figure: Linearly interpolated bounce back.

Signed distance from the wall (Marson et al., 2021)

PALABOS

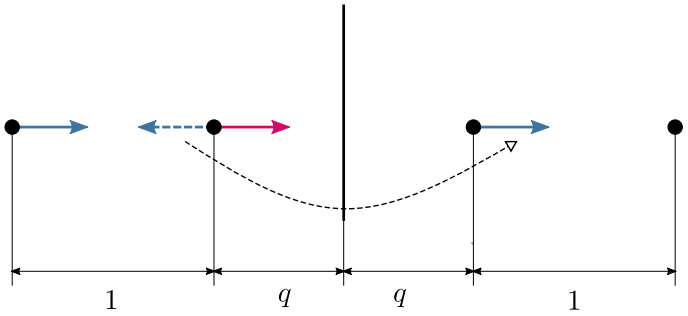
$$f_i(\boldsymbol{x}_F, t+1) = a_1 f_i^*(\boldsymbol{x}_{FF}) + a_2 \underbrace{f_i^*(\boldsymbol{x}_F)}_{\text{HW}} + a_3 f_i^*(\boldsymbol{x}_F)$$



Signed distance from the wall (Marson et al., 2021)

PALABOS

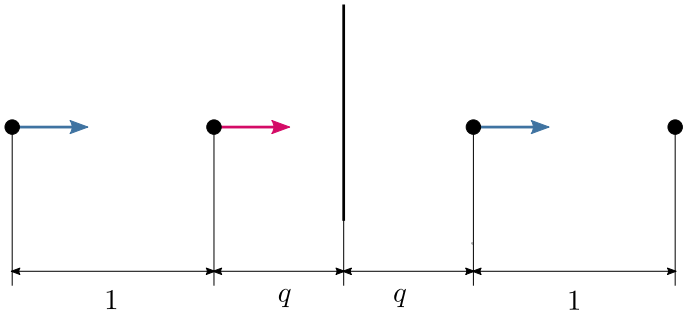
$$f_i(\boldsymbol{x}_F, t+1) = a_1 f_i^*(\boldsymbol{x}_{FF}) + a_2 \underbrace{f_i^*(\boldsymbol{x}_F)}_{\text{HW}} + a_3 f_i^*(\boldsymbol{x}_F)$$



Signed distance from the wall (Marson et al., 2021)

PALABOS

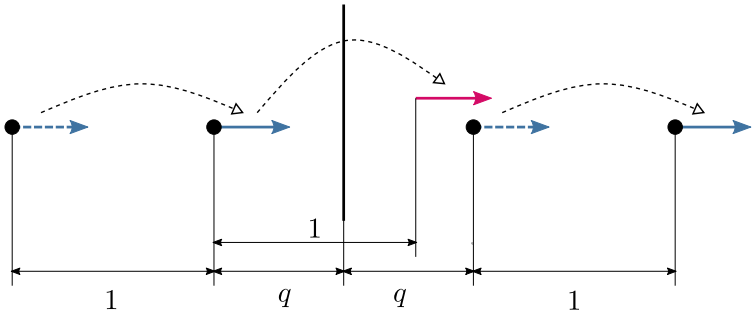
$$f_i(\boldsymbol{x}_F, t+1) = a_1 f_i^*(\boldsymbol{x}_{FF}) + a_2 \underbrace{f_i^*(\boldsymbol{x}_F)}_{\text{HW}} + a_3 f_i^*(\boldsymbol{x}_F)$$



Signed distance from the wall (Marson et al., 2021)

PALABOS

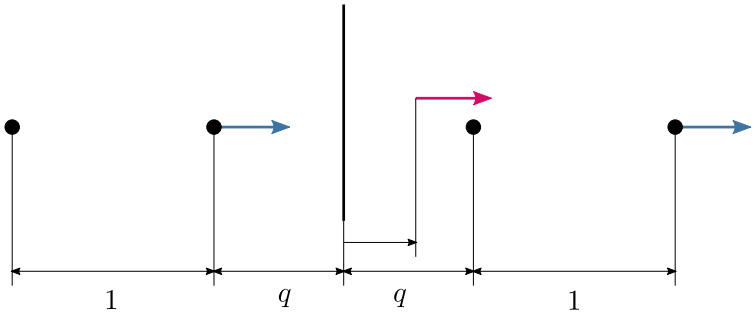
$$f_i(\boldsymbol{x}_F, t+1) = a_1 f_i^*(\boldsymbol{x}_{FF}) + a_2 \underbrace{f_i^*(\boldsymbol{x}_F)}_{\text{HW}} + a_3 f_i^*(\boldsymbol{x}_F)$$



Signed distance from the wall (Marson et al., 2021)

PALABOS

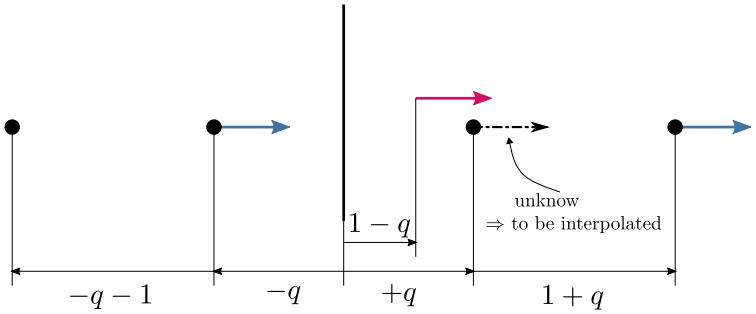
$$f_i(\boldsymbol{x}_F, t+1) = a_1 f_i^*(\boldsymbol{x}_{FF}) + a_2 \underbrace{f_i^*(\boldsymbol{x}_F)}_{\text{HW}} + a_3 f_i^*(\boldsymbol{x}_F)$$



Signed distance from the wall (Marson et al., 2021)

PALABOS

$$f_i(\boldsymbol{x}_F, t+1) = a_1 f_i^*(\boldsymbol{x}_{FF}) + a_2 \underbrace{f_i^*(\boldsymbol{x}_F)}_{\text{HW}} + a_3 f_i^*(\boldsymbol{x}_F)$$



Signed distance from the wall (Marson et al., 2021)

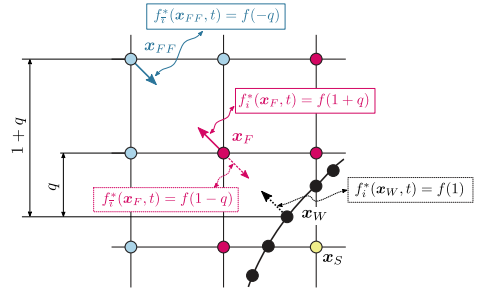
$$f_i\left(\boldsymbol{x}_F, t+1\right) = a_1 f_i^*\left(\boldsymbol{x}_{FF}\right) + a_2 \underbrace{f_i^*\left(\boldsymbol{x}_F\right)}_{\text{HW}} + a_3 f_i^*\left(\boldsymbol{x}_F\right)$$

We call the signed distance from the wall the generalized s coordinate (Marson et al., 2021)

Comparison of the metrics (Marson et al., 2021)

$$f_i(\boldsymbol{x}_F, t+1) = a_1 f_i^*(\boldsymbol{x}_{FF}) + a_2 \underbrace{f_i^*(\boldsymbol{x}_F)}_{\text{HW}} + a_3 f_i^*(\boldsymbol{x}_F)$$

$$\Rightarrow f(q) = a_{-q} f(-q) + a_{1-q} \underbrace{f(1-q)}_{\text{HW}} + a_{1+q} f(1+q)$$



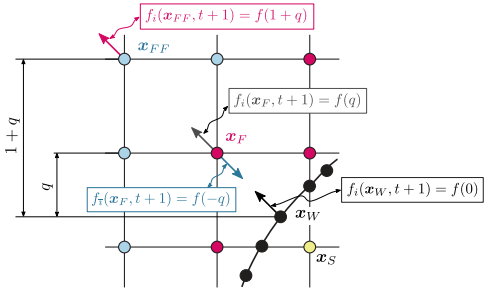


Figure: (Marson et al., 2021)

Comparison of the metrics (Marson et al., 2021)

$$f_i(\boldsymbol{x}_F, t+1) = a_1 f_i^*(\boldsymbol{x}_{FF}) + a_2 \underbrace{f_i^*(\boldsymbol{x}_F)}_{\text{HW}} + a_3 f_i^*(\boldsymbol{x}_F)$$

$$\Rightarrow f(q) = a_{-q} f(-q) + a_{1-q} \underbrace{f(1-q)}_{\text{HW}} + a_{1+q} f(1+q)$$

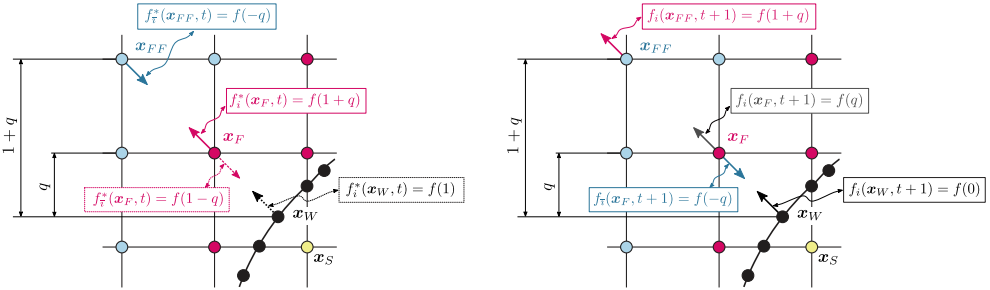


Figure: (Marson et al., 2021) ▸ ELI article 2021, PRE

General formula (2)

PALABOS

$$
\begin{aligned}
f_i(\boldsymbol{x}_F, t+1) = \; & a_{k+1}\left[f_{\bar{i}}^{*}\left(\boldsymbol{x}_{FF}\right) + 2f^{\mathrm{eq}-}(\boldsymbol{x}_W)\right] + \\
& + a_k\left[f_{\bar{i}}^{*}\left(\boldsymbol{x}_F\right) + 2f^{\mathrm{eq}-}(\boldsymbol{x}_W)\right] + \\
& + a_{k-1}\underbrace{\left[\tilde{f}_{\bar{i}}^{*}\left(\boldsymbol{x}_S\right) + 2f^{\mathrm{eq}-}(\boldsymbol{x}_W)\right]}_{\text{FH,MLS}} \\
& + a_j f_i^{*}\left(\boldsymbol{x}_F\right) + a_{j+1} f_i^{*}\left(\boldsymbol{x}_{FF}\right) \\
& + a_{j+2}\underbrace{\tilde{f}_i\left(\boldsymbol{x}_W, t+1\right)}_{\text{Tao et al.}} + a_{j+3}\underbrace{\tilde{f}_i^{*}\left(\boldsymbol{x}_W\right)}_{\text{ELI}} \\
& - \underbrace{K\left(\tau^{-}, f_i^{\mathrm{neq}-}(\boldsymbol{x}_F), \ldots\right)}_{\text{parametrization}},
\end{aligned}
$$

$$
\begin{aligned}
f(q) = \; & a_{-q}\left[f\left(-q\right) + 2f^{\mathrm{eq}-}(1)\right)\Big] \\
& + a_{1-q}\left[f\left(1-q\right) + 2f^{\mathrm{eq}-}(1)\right)\Big] \\
& + a_{2-q}\underbrace{\left[\tilde{f}\left(2-q\right) + 2f^{\mathrm{eq}-}(1)\right]}_{\text{FH,MLS}} \\
& + a_{1+q} f\left(1+q\right) + a_{2+q} f\left(2+q\right) \\
& + a_0\underbrace{\tilde{f}\left(0\right)}_{\text{Tao et al.}} + a_1\underbrace{\tilde{f}\left(1\right)}_{\text{ELI}} \\
& - \underbrace{K\left(\tau^{-}, f_i^{\mathrm{neq}-}(\boldsymbol{x}_F), \ldots\right)}_{\text{parametrization}},
\end{aligned}
$$

The Bouzidi Method (1)

The Bouzidi's method (linear) uses different f_i for the interpolation depending on the value of q.

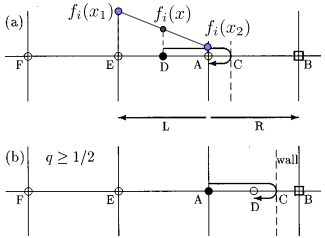
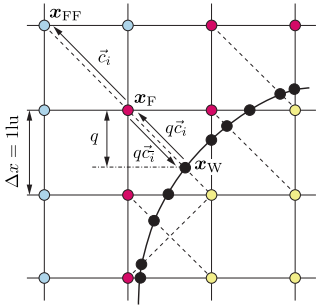


Figure: Bouzidi method (from the original paper).



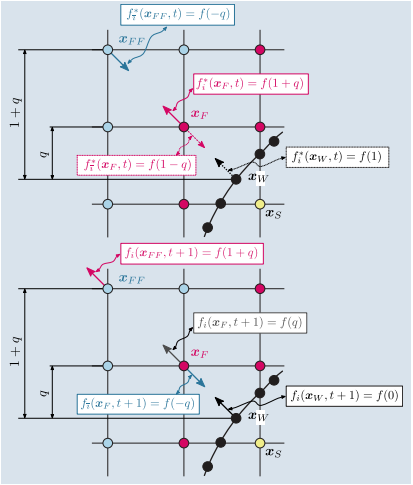
$$f_L^{t+1}(\boldsymbol{x}_A) = (1-2q) f_R^*(\boldsymbol{x}_E) + 2q f_R^*(\boldsymbol{x}_A) \qquad q < 0.5$$

$$f_L^{t+1}(\boldsymbol{x}_A) = \frac{1}{2q} f_R^*(\boldsymbol{x}_A) + \frac{2q-1}{2q} f_L^*(\boldsymbol{x}_A) \qquad q \geq 0.5$$

The Bouzidi Method (2): exercise

PALABOS

Using

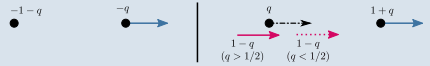


Try to

Convert the classical Bouzidi's equation

$$f_i^{t+1}(\boldsymbol{x}_F) = \boxed{?} f_{\bar{i}}^*(\boldsymbol{x}_{FF}) + \boxed{?} f_{\bar{i}}^*(\boldsymbol{x}_F) \qquad q < 0.5$$

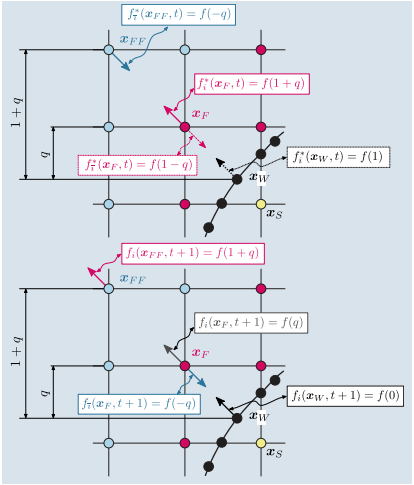
$$f_i^{t+1}(\boldsymbol{x}_F) = \boxed{?} f_{\bar{i}}^*(\boldsymbol{x}_F) + \boxed{?} f_i^*(\boldsymbol{x}_F) \qquad q \geq 0.5$$



The Bouzidi Method (2): exercise

PALABOS

Using

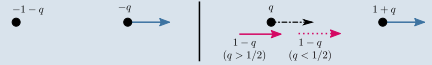


Try to

Convert the classical Bouzidi's equation

$$f_i^{t+1}(\boldsymbol{x}_F) = \boxed{?} f_{\bar{i}}^*(\boldsymbol{x}_{FF}) + \boxed{?} f_{\bar{i}}^*(\boldsymbol{x}_F) \qquad q < 0.5$$

$$f_i^{t+1}(\boldsymbol{x}_F) = \boxed{?} f_{\bar{i}}^*(\boldsymbol{x}_F) + \boxed{?} f_i^*(\boldsymbol{x}_F) \qquad q \geq 0.5$$



Solution

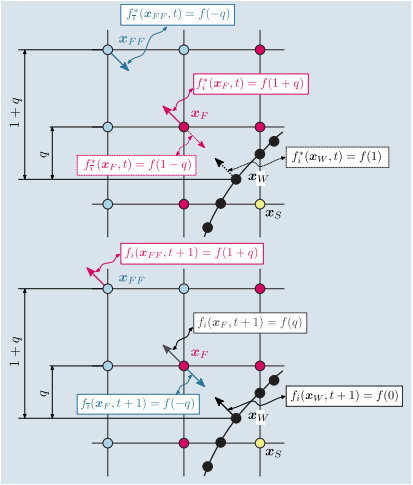
$$f(q) = \boxed{1-2q} f(-q) + \boxed{2q} f(1-q) \qquad q < 0.5$$

$$f(q) = \boxed{\frac{1}{2q}} f(1-q) + \boxed{\frac{2q-1}{2q}} f(1+q) \qquad q \geq 0.5$$

The Bouzidi Method (2): exercise

PALABOS

Using

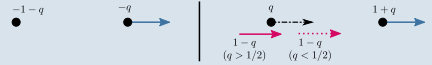


Try to

Convert the classical Bouzidi's equation

$$f_i^{t+1}(\boldsymbol{x}_F) = \boxed{?} f_{\bar{i}}^*(\boldsymbol{x}_{FF}) + \boxed{?} f_{\bar{i}}^*(\boldsymbol{x}_F) \qquad q < 0.5$$

$$f_i^{t+1}(\boldsymbol{x}_F) = \boxed{?} f_{\bar{i}}^*(\boldsymbol{x}_F) + \boxed{?} f_i^*(\boldsymbol{x}_F) \qquad q \geq 0.5$$



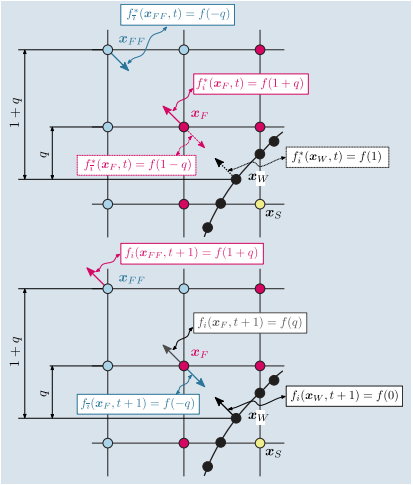
Trick

Use a translated coordinate $s' = s - q$ to simplify the interpolation and to see symmetry!

The Bouzidi Method (2): exercise

PALABOS

Using

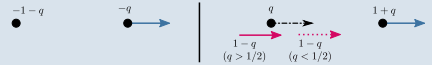


Try to

Convert the classical Bouzidi's equation

$$f_i^{t+1}(\boldsymbol{x}_F) = \boxed{?} f_{\bar{i}}^*(\boldsymbol{x}_{FF}) + \boxed{?} f_{\bar{i}}^*(\boldsymbol{x}_F) \qquad q < 0.5$$

$$f_i^{t+1}(\boldsymbol{x}_F) = \boxed{?} f_{\bar{i}}^*(\boldsymbol{x}_F) + \boxed{?} f_i^*(\boldsymbol{x}_F) \qquad q \geq 0.5$$



$$f(s'=0) = \boxed{1-2q} f(-2q) + \boxed{2q} f(1-2q) \qquad q < 0.5$$

$$f(s'=0) = \boxed{\frac{1}{2q}} f(1-2q) + \boxed{\frac{2q-1}{2q}} f(1) \qquad q \geq 0.5$$

The Bouzidi Method (3): Palabos Implementation

palabos/src/offLattice/bouzidiOffLatticeModel3D.hh → BouzidiOffLatticeModel3D<T,Descriptor>::cellCompletion()

```
if (bdType==OffBoundary::dirichlet) {
    T u_ci = D::c[iPop][0]*wall_vel[0]+D::c[iPop][1]*wall_vel[1]
                                      +D::c[iPop][2]*wall_vel[2];
    plint numUnknown = 0;
    if (q<(T)0.5) {
        if (hasFluidNeighbor[i]) {
           cell[oppPop] = 2.*q*iCell[iPop] + (1.-2.*q)*cell[iPop];
        }
        else {
            ++numUnknown;
            cell[oppPop] = iCell[iPop];
        }
        cell[oppPop] -= 2.* u_ci*D::t[iPop]*D::invCs2;
    }
    else {
        cell[oppPop] = 1./(2.*q)*iCell[iPop]+(2.*q-1)/(2.*q)*jCell[oppPop];
        cell[oppPop] -= 1./q* u_ci*D::t[iPop]*D::invCs2;
    }
}
```

The CLI scheme (Ginzburg & Verhaeghe, 2008)

Is a "unified" scheme like the Yu's one, but has an advantage!

Advantage

It has viscosity independent accuracy at steady state!

$$f(q) = \frac{1-2q}{1+2q} f(-q) + f(1-q) - \frac{1-2q}{1+2q} f(1+q)$$

Exercise

Find the traditional form of the CLI scheme.

Answer

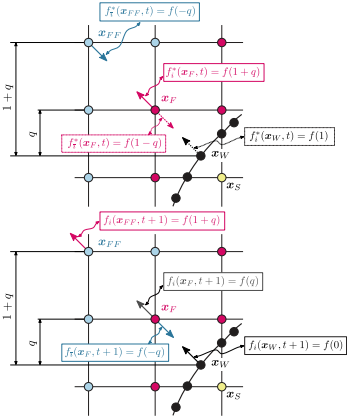


Figure: (Marson et al., 2021)

The CLI scheme (Ginzburg & Verhaeghe, 2008)

PALABOS

Is a "unified" scheme like the Yu's one, but has an advantage!

Advantage

It has viscosity independent accuracy at steady state!

$$f(q) = \frac{1-2q}{1+2q} f(-q) + f(1-q) - \frac{1-2q}{1+2q} f(1+q)$$

Exercise

Find the traditional form of the CLI scheme.

Answer

$$f_i^{t+1}(\boldsymbol{x}_F) = \frac{1-2q}{1+2q} f_{\bar{i}}^*(\boldsymbol{x}_{FF}, t) + f_{\bar{i}}^*(\boldsymbol{x}_F, t) - \frac{1-2q}{1+2q} f_i^*(\boldsymbol{x}_F, t)$$

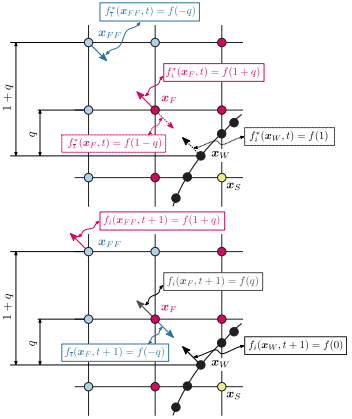


Figure: (Marson et al., 2021)

Filippova Hänel (Filippova & Hänel, 1997) (1)

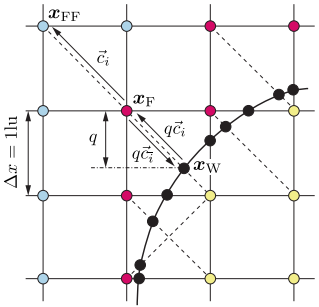
PALABOS

$$f_i(\boldsymbol{x}_F, t+1) = (1-\chi) f_i^*(\boldsymbol{x}_F, t) + \chi f_i^{\mathrm{eq}}(\boldsymbol{x}_S, t) \tag{9a}$$

$$\chi = \begin{cases} \frac{1}{\tau}\frac{2q-1}{1-\frac{1}{\tau}} & q < \frac{1}{2} \\ \frac{1}{\tau}(2q-1) & q \geq \frac{1}{2} \end{cases} \tag{9b}$$

$$f_i^{\mathrm{eq}}(\boldsymbol{x}_S, t) = w_i \left(\frac{p(\boldsymbol{x}_F, t)}{c_s^2} + \frac{\boldsymbol{c}_i \cdot \boldsymbol{u}_S}{c_s^2} + \frac{(\boldsymbol{c}_i \cdot \boldsymbol{u}_F)^2}{2c_s^4} - \frac{\boldsymbol{u}_F \cdot \boldsymbol{u}_F}{2c_s^2} \right) \tag{9c}$$

$$\boldsymbol{u}_S = \begin{cases} \boldsymbol{u}_F & q < \frac{1}{2} \\ \frac{q-1}{q}\boldsymbol{u}_F + \frac{1}{q}\boldsymbol{u}_W & q \geq \frac{1}{2} \end{cases} \tag{9d}$$



Note

Filippova Hänel is both algorithmically and physically local!

Filippova Hänel (2): Palabos Implementation

PALABOS

palabos/src/offLattice/filippovaHaenelOffLatticeModel3D.hh → FilippovaHaenelLocalModel3D<T,Descriptor>::cellCompletion()

```
if (delta < 0.5) {
    wf_j = f_j;
    kappa = (omega * (2.0 * delta - 1.0)) / (1.0 - omega);

} else {
    wf_j = f_j * ((delta - 1.0) / delta) + w_j / delta;
    kappa = omega * (2.0 * delta - 1.0);
}

T c_i_wf_j_f_j = D::c[iPop][0]*(wf_j[0]-f_j[0]) +
D::c[iPop][1]*(wf_j[1]-f_j[1]) +
D::c[iPop][2]*(wf_j[2]-f_j[2]) ;

T c_i_w_j      = D::c[iPop][0]*w_j[0] + D::c[iPop][1]*w_j[1] + D::c[iPop][2]*w_j[2];


T f_ieq = f_cell.getDynamics().computeEquilibrium(iPop, f_rhoBar, f_j, f_jSqr) +
D::t[iPop]*D::invCs2*c_i_wf_j_f_j;
s_cell[iOpp] = (1.0-kappa)*collidedCell[iPop]+kappa*f_ieq+2.0*D::t[iPop]*D::invCs2*c_i_w_j;
```

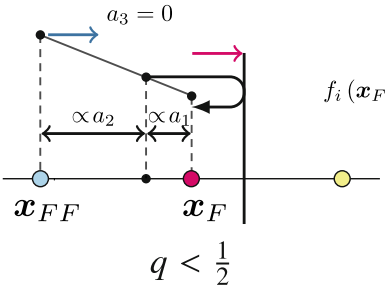
Table of methods

$$\uparrow = \begin{cases} 1 & q < 0.5 \\ 0 & q \geq 0.5 \end{cases} \qquad \downarrow = \begin{cases} 0 & q < 0.5 \\ 1 & q \geq 0.5 \end{cases}$$

coeff. $\rightarrow$ / Name	a_{-1-q}	a_{-q}	a_0	a_{1-q}	a_1	a_{2-q}	a_{1+q}	a_{2+q}	K
HW				1					not needed
BFL		$(1-2q)\uparrow$		$2q\uparrow + \frac{1}{2q}\downarrow$			$\frac{2q-1}{2q}\downarrow$		available
CLI		$\frac{1-2q}{1+2q}$		1			$-\frac{1-2q}{1+2q}$		not needed
NELIUL			$1-q$		q				available
NELIULT			$\frac{1}{1+q}$				$\frac{q}{1+q}$		available
MR	$\frac{q^2}{(1+q)^2}$	$\frac{1-2q(1+q)}{(1+q)^2}$		1			$-\frac{1-2q(1+q)}{(1+q)^2}$	$-\frac{q^2}{(1+q)^2}$	available

Algorithmic locality vs Physical locality

- Can we implement Bouzidi *et al.* method locally?
- We can apply the algorithm after streaming;



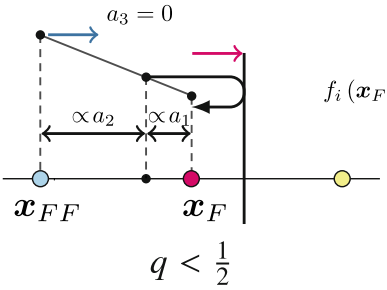
$$f_i(\boldsymbol{x}_F, t+1) = a_1 \; f_{\bar{i}}^*(\boldsymbol{x}_{FF}) \; + a_2 \; f_{\bar{i}}^*(\boldsymbol{x}_F)$$

Figure: Can we implement Bouzidi *et al.* method locally?

We get algorithmic local, but not physically local method.⇒ This eventually cause issues.

Algorithmic locality vs Physical locality

- Can we implement Bouzidi *et al.* method locally?
- We can apply the algorithm after streaming;



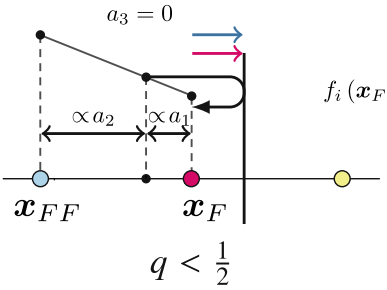
$$f_i(\boldsymbol{x}_F, t+1) = a_1 \; f_{\bar{i}}^*(\boldsymbol{x}_{FF}) \; + a_2 \; f_{\bar{i}}^*(\boldsymbol{x}_F)$$

Figure: Can we implement Bouzidi *et al.* method locally?

We get algorithmic local, but not physically local method.⇒ This eventually cause issues.

Algorithmic locality vs Physical locality

- Can we implement Bouzidi *et al.* method locally?
- We can apply the algorithm after streaming;



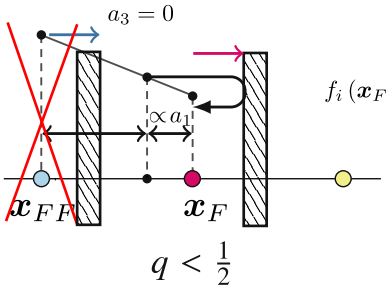
$$f_i(\boldsymbol{x}_F, t+1) = a_1 \underbrace{f_{\bar{i}}^*(\boldsymbol{x}_{FF})}_{\text{stream to } \boldsymbol{x}_F} + a_2 \underbrace{f_{\bar{i}}^*(\boldsymbol{x}_F)}_{\text{save in } \boldsymbol{x}_F}$$
$$\| $$
$$f_{\bar{i}}(\boldsymbol{x}_F, t+1)$$

Figure: Can we implement Bouzidi *et al.* method locally?

We get algorithmic local, but not physically local method.⇒ This eventually cause issues.

Algorithmic locality vs Physical locality

- Can we implement Bouzidi *et al.* method locally?
- We can apply the algorithm after streaming;



$$f_i(\boldsymbol{x}_F, t+1) = a_1 \underbrace{f_{\bar{i}}^*(\boldsymbol{x}_{FF})}_{\text{stream to } \boldsymbol{x}_F} + a_2 \underbrace{f_{\bar{i}}^*(\boldsymbol{x}_F)}_{\text{save in } \boldsymbol{x}_F}$$

Figure: Can we implement Bouzidi *et al.* method locally?

We get algorithmic local, but not physically local method.⇒ This eventually cause issues.

Local (single-node) and non-local link-wise methods

PALABOS

Non-local

- BFL (Bouzidi et al., 2001)
- CLI (Ginzburg & Verhaeghe, 2008)
- MR (Ginzburg & d'Humières, 2003)
- Yu (Yu et al., 2003)
- MLS (Mei et al., 1999)
- ...

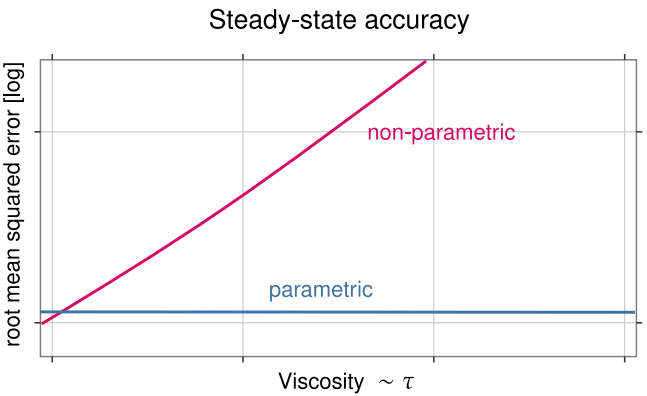
Local schemes

- HW (Ginzburg & Adler, 1994)
- FH (Filippova & Hänel, 1997)
- ZY
- ELI schemes (Marson et al., 2021)

Advantage

Uniform implementation in the whole domain: also in narrow gaps and corners!

Viscosity independence at steady state



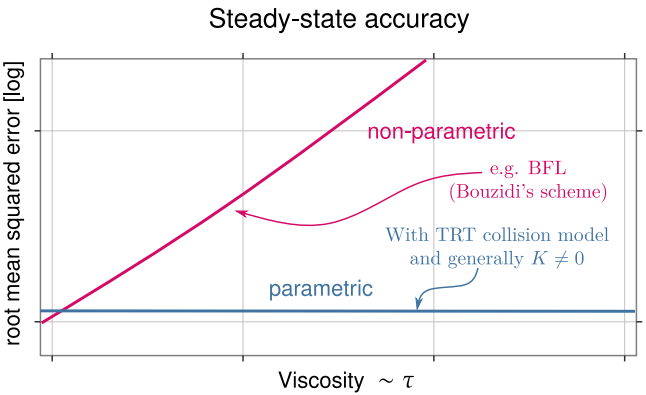
Why it is important?

- Low Re $\Rightarrow$ slow convergence
- increase τ to increase $\mathrm{d}t$

without parametrization:

1. increasing error at steady state;
2. virtual position of the boundary changes with τ
3. in porous media permeability is a function of τ

Viscosity independence at steady state



Why it is important?

- Low Re $\Rightarrow$ slow convergence
- increase τ to increase $\mathrm{d}t$

without parametrization:

1. increasing error at steady state;
2. virtual position of the boundary changes with τ
3. in porous media permeability is a function of τ

The closure relation (1)

PALABOS

Why we need to write a closure relation?

1. to correct the steady-state viscosity-dependence;
2. to analyze the errors introduced by the scheme.

Ingredients

1. link-wise formula;
2. TRT approach $\Rightarrow$ $f_i = f_i^+ + f_i^-$;
3. Taylor expansion;
4. Chapmann-Enskog expansion;
5. Diffusive scaling
   $Re_1 = Re_2$; $Ma_1 \neq Ma_2$; $Kn \propto \mathrm{dx}$

⚠ Erratum (Marson et al., 2021)

Equation (B3)

$$\mathrm{Re} = \mathrm{Re}_1 = \mathrm{Re}_2,$$
$$\mathrm{Kn} \neq \mathrm{Ma}_1 \neq \mathrm{Ma}_2 \neq \mathrm{Kn}_1 \neq \mathrm{Kn}_2$$

The closure relation (2)

PALABOS

$$\alpha^+\left(\cancel{1} + \partial_t + qc_{\bar{\imath},\alpha}\partial_\alpha + \cancel{\frac{\partial_t^2}{2} + qc_{\bar{\imath},\alpha}\partial_t\partial_\alpha} + \frac{q^2}{2}(c_{\bar{\imath},\alpha}\partial_\alpha)^2\right) f_{\bar{\imath}}^{\mathrm{eq}+}$$

$$\approx \left(\cancel{\alpha^+} + \tau^+\partial_t + \beta^+ c_{\bar{\imath},\alpha}\partial_\alpha + \cancel{\upsilon^+\partial_t^2 + \theta^+ c_{\bar{\imath},\alpha}\partial_t\partial_\alpha} + \gamma^+(c_{\bar{\imath},\alpha}\partial_\alpha)^2\right) f_{\bar{\imath}}^{\mathrm{eq}+},$$

$$\alpha^-\left(\cancel{1} + \partial_t + \cancel{gc_{\bar{\imath},\alpha}\partial_\alpha} + \cancel{\frac{\partial_t^2}{2} + qc_{\bar{\imath},\alpha}\partial_t\partial_\alpha} + \frac{q^2}{2}(c_{\bar{\imath},\alpha}\partial_\alpha)^2\right) f_{\bar{\imath}}^{\mathrm{eq}-}$$

$$\approx \left(\cancel{\alpha^-} + \tau^-\partial_t + \cancel{g\alpha^- c_{\bar{\imath},\alpha}\partial_\alpha} + \cancel{\upsilon^-\partial_t^2 + \theta^- c_{\bar{\imath},\alpha}\partial_t\partial_\alpha} + \gamma^-(c_{\bar{\imath},\alpha}\partial_\alpha)^2\right) f_{\bar{\imath}}^{\mathrm{eq}-},$$

$$\alpha^+ = a_2 + a_3 - 1, \qquad \tau^+ = \beta^- - 1,$$

$$\alpha^- = a_2 - a_3 + 1, \qquad \tau^- = \beta^+ + 1,$$

$$\alpha_W^+ = -a_4 - a_5, \qquad \gamma^+ = -\Lambda^-\beta^-,$$

$$\alpha_W^- = a_4 + a_5 + 2a_2, \qquad \gamma^- = -\Lambda^+\beta^+.$$

$$\beta^+ = -a_2(\tau^- - 1) + a_3(\tau^- - 1) + \tau^- a_4 + a_5(\tau^- - 1) - \tau^- + K^-,$$

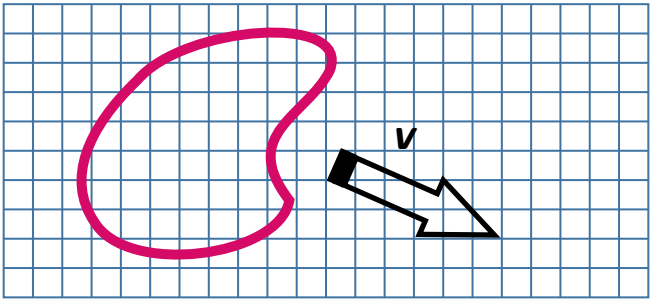
$$\beta^- = -a_2(\tau^+ - 1) - a_3(\tau^+ - 1) - \tau^+ a_4 - a_5(\tau^+ - 1) + \tau^+,$$

Figure: (Marson et al., 2021)

Move the boundaries

Move a boundary surface inside the static and structured LBM lattice adds some problems:

1. What happens when the boundary crosses the lattice nodes?
2. Which kind of errors does this operation introduce?



Creation and destruction of Fluid Nodes

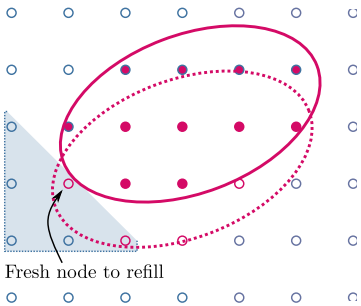
The main problem is with newly created fluid nodes → a refill of fluid is needed.
Tecniques to perform the refill:

- equilibrium refill;
- equilibrium + non equilibrium;
- Grad based approaches;
- Local Iteration approach;

(Tao et al., 2016) showed that the best performing techniques are the Grad-based and the Local Iteration (Chen et al., 2014).

Local iteration refill:

1. detect fresh nodes;
2. consider fresh node + envelope subdomain;
3. do until stop condition
   - 3.1 virtual collide&stream + boundary condition in the subdomain; this operation writes new values only on the fresh nodes!
   - 3.2 check stop condition



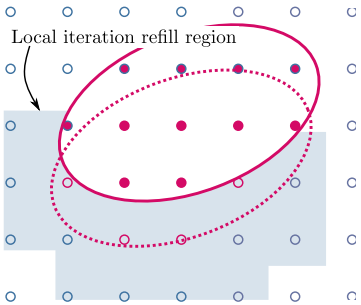
Creation and destruction of Fluid Nodes

The main problem is with newly created fluid nodes → a refill of fluid is needed.
Tecniques to perform the refill:

- equilibrium refill;
- equilibrium + non equilibrium;
- Grad based approaches;
- Local Iteration approach;

(Tao et al., 2016) showed that the best performing techniques are the Grad-based and the Local Iteration (Chen et al., 2014).
Local iteration refill:

1. detect fresh nodes;
2. consider fresh node + envelope subdomain;
3. do until stop condition
   - 3.1 virtual collide&stream + boundary condition in the subdomain; this operation writes new values only on the fresh nodes!
   - 3.2 check stop condition



Momentum Exchange Algorithm

Standard method (Ladd, 1994):

$$F = \sum_i -\left(\boldsymbol{c}_i f_i\left(\boldsymbol{x}_f\right) - \boldsymbol{c}_{\bar{i}} f_{\bar{i}}^*\left(\boldsymbol{x}_f\right)\right) \tag{10}$$

It exists also an improved version for galelian invariance (Wen et al., 2014; Tao et al., 2016):

$$F = \sum_i -\left(\left(\boldsymbol{c}_i - \boldsymbol{u}_w\right) f_i\left(\boldsymbol{x}_f\right) - \left(\boldsymbol{c}_{\bar{i}} - \boldsymbol{u}_w\right) f_{\bar{i}}^*\left(\boldsymbol{x}_f\right)\right) \tag{11}$$

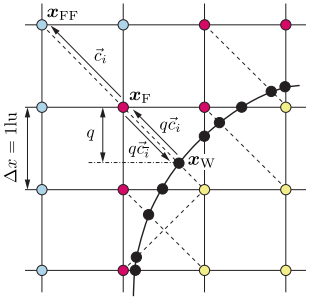


Figure: (Marson et al., 2021)

Integration of the stresses

PALABOS

Depending on the application and on the type of interpolation/extrapolation used, it can results in better or worse results of the ME.

$$\sigma_{\alpha\beta} = -c_s^2 \rho \boldsymbol{I} - \left(1 - \frac{1}{2\tau}\right) \sum_i f_i^{\mathrm{neq}}(\boldsymbol{x}, t)(\boldsymbol{c}_i - \boldsymbol{u}_\alpha)(\boldsymbol{c}_i - \boldsymbol{u}_\alpha) \tag{12}$$

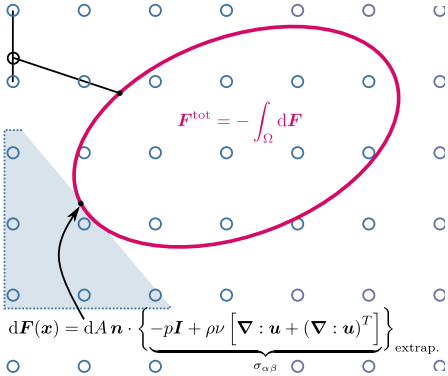
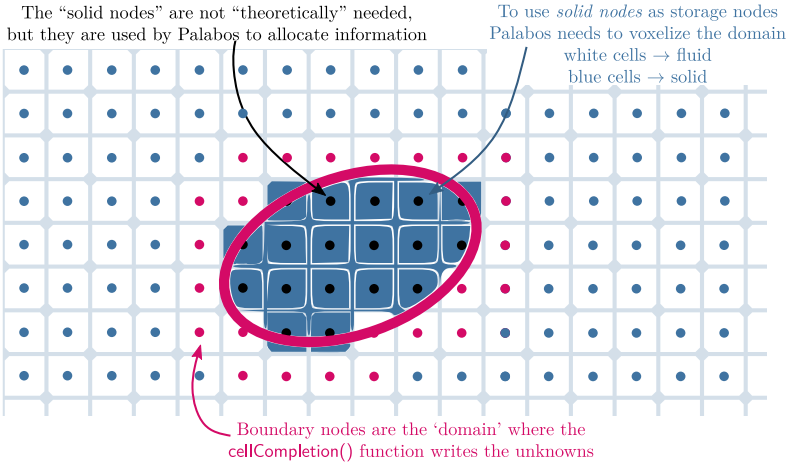


Figure:

Palabos Implementation



Read a stl file, or generate it

PALABOS

1.a Read an .stl file

Palabos holds all the information of a .stl file in objects of class type TriangleSet<Real>.
So, a stl file can be read as:

```
TriangleSet<Real> object_name("file_name.stl");
```

1.b Generate simple geometry TriangleSet

Palabos can also generate TriangleSets of simple geometries, like spheres:

```
TriangleSet<Real> sphere = constructSphere<Real>(
        center, radius, minNumOfTriangles);
```

Create the mesh, voxelize the domain

PALABOS

2. Voxelize the domain

The boundary condition is implemented as a *data processing functional* (DPF). This functional needs to know:

1. geometry of the stl
2. intersections with *links*
3. voxel matrix (fluid/solid)

$\Rightarrow$ a more complex data structure is needed: the VoxelizedDomain3D<Real>. The latter is generated starting from TriangleSet<Real> after some steps.

```
auto defMesh = new DEFscaledMesh<Real>
                (*triangle_set, 0, xDirection, margin,
                                        Dot3D(0, 0, 0));
auto boundary = new TriangleBoundary3D<Real>(*defMesh);
// 2. Create the voxel matrix
auto bounding_box =
    Box3D(0, parameters.getNx()-1, 0,
             parameters.getNy()-1, 0,
           parameters.getNz() - 1);
auto voxalized_domain = new VoxelizedDomain3D<Real>
(*boundary, voxelFlag::outside, bounding_box, borderWidth,
    extendedEnvelopeWidth, blockSize);
```

Allocate the OffLatticeModel and inject the dataprocessor

PALABOS

7. Use the geometry to create the DPF

The voxel matrix and the other geometrical information contained in VoxelizedDomain3D<Real>, is then used to create the actual OffLatticeModel3D<Real, Array3D> and the relative DPF that operates in the lattice (OffLatticeBoundaryCondition3D<Real, Descriptor, Array3D>),

```
offLatticeModel=
    new BouzidiOffLatticeModel3D<Real, Descriptor>(
new TriangleFlowShape3D<Real, Array<Real, 3> >(
    voxalized_domain->getBoundary(), *profiles),
    voxelFlag::outside);
boundaryCondition =
new OffLatticeBoundaryCondition3D<Real,Descriptor,Array3D>(
    offLatticeModel, *voxalized_domain, *target_lattice);

defineDynamics(*target_lattice,
        voxalized_domain->getVoxelMatrix(),
           target_lattice->getBoundingBox(),
           new NoDynamics<Real, Descriptor>(),
voxelFlag::inside);

boundaryCondition->insert();
```

The Bouzidi Method (4): exercise

PALABOS

Using

```
T u_ci = D::c[iPop][0]*wall_vel[0]+D::c[iPop][1]*wall_vel[1]
                                  +D::c[iPop][2]*wall_vel[2];
plint numUnknown = 0;
if (q<(T)0.5) {
    if (hasFluidNeighbor[i]) {
       cell[oppPop] = 2.*q*iCell[iPop] + (1.-2.*q)*cell[iPop];
    }
    else {
        ++numUnknown;
        cell[oppPop] = iCell[iPop];
    }
    cell[oppPop] -= 2.* u_ci*D::t[iPop]*D::invCs2;
}
else {
    cell[oppPop] = 1./(2.*q)*iCell[iPop]+
                        (2.*q-1)/(2.*q)*jCell[oppPop];
    cell[oppPop] -= 1./q* u_ci*D::t[iPop]*D::invCs2;
}
```

Try to

Write in mathematical notations the following:

- iPop
- oppPop
- u_ci
- cell[oppPop]
- cell[iPop]
- jcell[oppPop]
- icell[iPop]

Thank you! Questions?

THANK YOU FOR YOU ATTENTIONS! QUESTIONS?

Arcknowdlegments

References I

Bouzidi, M., Firdaouss, M., & Lallemand, P. (2001, October). Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Physics of Fluids*, *13*(11), 3452–3459. Retrieved 2019-10-18, from https://aip.scitation.org/doi/abs/10.1063/1.1399290 doi: 10.1063/1.1399290

Chen, L., Yu, Y., Lu, J., & Hou, G. (2014). A comparative study of lattice Boltzmann methods using bounce-back schemes and immersed boundary ones for flow acoustic problems. *International Journal for Numerical Methods in Fluids*, *74*(6), 439–467. Retrieved 2019-06-17, from https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.3858 doi: 10.1002/fld.3858

Filippova, O., & Hänel, D. (1997, September). Lattice-Boltzmann simulation of gas-particle flow in filters. *Computers & Fluids*, *26*(7), 697–712. Retrieved 2020-06-25, from http://www.sciencedirect.com/science/article/pii/S0045793097000091 (162 citations (Crossref) [2021-05-08]) doi: 10.1016/S0045-7930(97)00009-1

References II

Ginzburg, I., & Adler, P. M. (1994, February). Boundary flow condition analysis for the three-dimensional lattice Boltzmann model. *Journal de Physique II*, *4*(2), 191–214. Retrieved from http://www.edpsciences.org/10.1051/jp2:1994123 doi: 10.1051/jp2:1994123

Ginzburg, I., & d'Humières, D. (2003, December). Multireflection boundary conditions for lattice Boltzmann models. *Physical Review E*, *68*(6), 066614. Retrieved from https://link.aps.org/doi/10.1103/PhysRevE.68.066614 doi: 10.1103/PhysRevE.68.066614

Ginzburg, I., & Verhaeghe, F. (2008). Two-Relaxation-Time Lattice Boltzmann Scheme: About Parametrization, Velocity, Pressure and Mixed Boundary Conditions. *Commun. Comput. Phys.*, *3*(2), 427–478. Retrieved from https://global-sci.org/intro/article_detail/cicp/7862.html

Krüger, T., Kusumaatmaja, H., Kuzmin, A., Shardt, O., Silva, G., & Viggen, E. M. (2017). *The Lattice Boltzmann Method: Principles and Practice*. Springer International Publishing. Retrieved from https://www.springer.com/gp/book/9783319446479

References III

Ladd, A. J. C. (1994, July). Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation. *Journal of Fluid Mechanics*, *271*, 285–309. Retrieved 2019-10-17, from https://www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/numerical-simulations-of-particulate-suspensions-via-a-discretized-boltzmann-equation-part-1-theoretical-foundation/7AB6FF4ADE0133108C88992CFA9B6AC7 doi: 10.1017/S0022112094001771

Marson, F., Thorimbert, Y., Chopard, B., Ginzburg, I., & Latt, J. (2021, May). Enhanced single-node lattice Boltzmann boundary condition for fluid flows. *Physical Review E*, *103*(5), 053308. Retrieved 2021-05-22, from https://link.aps.org/doi/10.1103/PhysRevE.103.053308 doi: 10.1103/PhysRevE.103.053308

References IV

Mei, R., Luo, L.-S., & Shyy, W. (1999, November). An Accurate Curved Boundary Treatment in the Lattice Boltzmann Method. *Journal of Computational Physics*, *155*(2), 307–330. Retrieved 2019-06-17, from http://www.sciencedirect.com/science/article/pii/S0021999199963349 (377 citations (Crossref) [2021-05-08]) doi: 10.1006/jcph.1999.6334

Tao, S., Hu, J., & Guo, Z. (2016, July). An investigation on momentum exchange methods and refilling algorithms for lattice Boltzmann simulation of particulate flows. *Computers & Fluids*, *133*, 1–14. Retrieved 2019-06-17, from https://linkinghub.elsevier.com/retrieve/pii/S0045793016301116 doi: 10.1016/j.compfluid.2016.04.009

Wen, B., Zhang, C., Tu, Y., Wang, C., & Fang, H. (2014, June). Galilean invariant fluid–solid interfacial dynamics in lattice Boltzmann simulations. *Journal of Computational Physics*, *266*, 161–170. Retrieved from http://www.sciencedirect.com/science/article/pii/S0021999114001387 doi: 10.1016/j.jcp.2014.02.018

References V

Yu, D., Mei, R., & Shyy, W. (2003). A Unified Boundary Treatment in Lattice Boltzmann Method. *41st Aerospace Sciences Meeting and Exhibit*. Retrieved 2019-08-22, from https://arc.aiaa.org/doi/abs/10.2514/6.2003-953 doi: 10.2514/6.2003-953

The Yu's Method (Yu et al., 2003)

PALABOS

The Yu's method yuaunified (linear) uses the same f_is for the independently from the value of q.

It applys a double linear interpolation to avoid defining two separate case depending on q.

$$f_i^{t+1}(\boldsymbol{x}_F) = \frac{\left(q f_i^*(\boldsymbol{x}_F) + q f_i^*(\boldsymbol{x}_F) + (1-q) f_i^*(\boldsymbol{x}_{FF})\right)}{1+q} \tag{13}$$

- Is is slightly less accurate of Bouzidi;
- the advantage of having an unique interpolation scheme independent from q is debated.

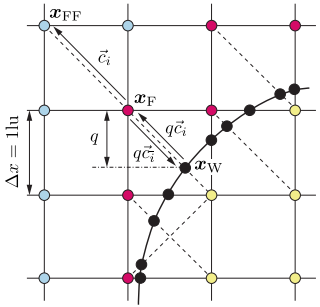


Figure: (Marson et al., 2021)

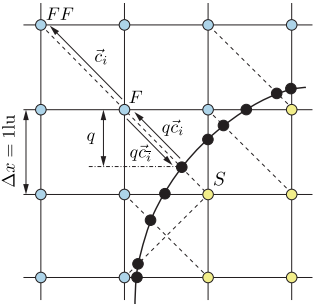
Mei-Luo-Shyy method (Mei et al., 1999) (1)

$$f_i\left(\boldsymbol{x}_F, t+1\right) = (1-\chi) f_i^*\left(\boldsymbol{x}_F, t\right) + \chi f_i^{\mathrm{eq}}\left(\boldsymbol{x}_S, t\right) \tag{14a}$$

$$\chi = \begin{cases} \frac{1}{\tau}\frac{2q-1}{1-\frac{2}{\tau}} & q < \frac{1}{2} \\ \frac{1}{\tau}(2q-1) & q \geq \frac{1}{2} \end{cases} \tag{14b}$$

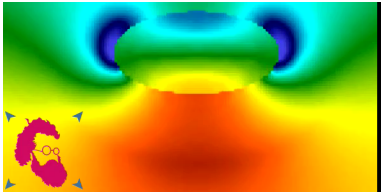
$$f_i^{\mathrm{eq}}\left(\boldsymbol{x}_S, t\right) = w_i \left(\frac{p\left(\boldsymbol{x}_F, t\right)}{c_s^2} + \frac{\boldsymbol{c}_i \cdot \boldsymbol{u}_S}{c_s^2} + \frac{\left(\boldsymbol{c}_i \cdot \boldsymbol{u}_F\right)^2}{2c_s^4} - \frac{\boldsymbol{u}_F \cdot \boldsymbol{u}_F}{2c_s^2} \right) \tag{14c}$$

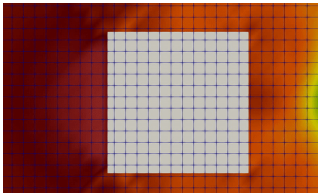
$$\boldsymbol{u}_S = \begin{cases} \boldsymbol{u}_{FF} & q < \frac{1}{2} \\ \frac{q-1}{q}\boldsymbol{u}_F + \frac{1}{q}\boldsymbol{u}_W & q \geq \frac{1}{2} \end{cases} \tag{14d}$$



Refilling is not the end of the story

I the numerical viscosity is sufficiently high ($\tau \sim 0.6$), the refilling is a sufficient counter measure to remove spurious pressure oscillations.





Performed by Dimitrios Kontaxakis

If the viscosity is low ($\tau \sim 0.5$) the refilling is not enough, and spurios oscillations appears also in the front.