A Lattice Boltzmann Method for immiscible multiphase flow simulations using the Level Set Method

Lorenz Hufnagel, Daniel Zint

BGCE Student Project

August 20th, 2015

Multiphase flow

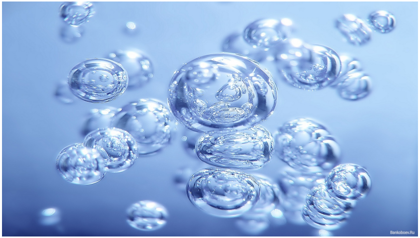
Examples

- Liquid-liquid mixtures (e.g. oil in water)
- Gas-liquid mixtures (e.g. bubble dynamics)

Multiphase flow

Examples

- Liquid-liquid mixtures (e.g. oil in water)
- Gas-liquid mixtures (e.g. bubble dynamics)



Macroscopic fluid mechanics

- N immiscible fluids.
- Each has own ρ_i, ν_i
- Hydrodynamics described by (incompressible) NSE

Macroscopic fluid mechanics

- N immiscible fluids.
- Each has own ρ_i, ν_i
- Hydrodynamics described by (incompressible) NSE

Macroscopic fluid mechanics

- N immiscible fluids.
- Each has own ρ_i, ν_i
- Hydrodynamics described by (incompressible) NSE

$$\nabla \cdot \vec{v} = 0$$

$$\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla)\, \vec{v} = -\frac{1}{\varrho_i} \nabla p + \nu_i \nabla^2 \vec{v}$$

Mathematic foundation

Interface conditions

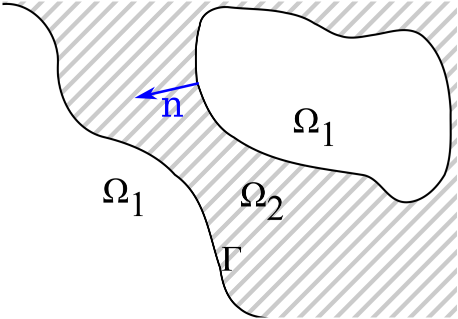


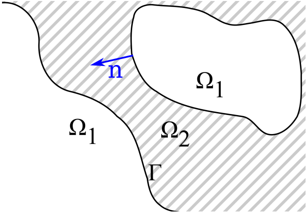
Figure: Two fluid domains Ω_i and interface Γ inbetween

- Velocity across interface is C_0-continous

$$[v] = \lim_{\epsilon \to 0} (\vec{v}\,(x + \epsilon\vec{n}) - \vec{v}(x - \epsilon\vec{n})) = 0$$

Mathematic foundation

Interface conditions cont'd

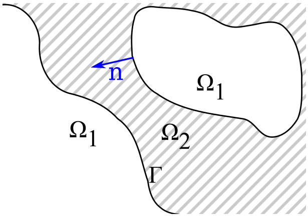


- Normal stress is balanced by surface tension → pressure jump

$$[T] \cdot \vec{n} = \lim_{\epsilon \to 0} (\mathbf{T}_2(x + \epsilon \vec{n}) - \mathbf{T}_1(x - \epsilon \vec{n})) \cdot \vec{n} = 2\sigma\kappa\vec{n}$$

Mathematic foundation

Interface conditions cont'd



- Normal stress is balanced by surface tension → pressure jump

$$[T] \cdot \vec{n} = \lim_{\epsilon \to 0} (\mathbf{T}_2(x + \epsilon \vec{n}) - \mathbf{T}_1(x - \epsilon \vec{n})) \cdot \vec{n} = 2\sigma\kappa\vec{n}$$

where $\mathbf{T}_i$ is the stress tensor $\mathbf{T}_i = 2\nu_i \rho_i \mathbf{S}_i - p\mathbb{I}$ and κ is the curvature of the interface $\nabla \cdot \vec{n}$. $\mathbf{S}_{ij} = \frac{1}{2}(\partial_i v_j + \partial_j v_i)$.

To solve the two-phase problem we need to:

- solve the flow equations $\rightarrow$ LBM
- compute the motion of the interface $\rightarrow$ Level Set Method
- couple the two algorithms $\rightarrow$ according BC's

To solve the two-phase problem we need to:

- solve the flow equations $\rightarrow$ LBM
- compute the motion of the interface $\rightarrow$ Level Set Method
- couple the two algorithms $\rightarrow$ according BC's

To solve the two-phase problem we need to:

- solve the flow equations $\rightarrow$ LBM
- compute the motion of the interface $\rightarrow$ Level Set Method
- couple the two algorithms $\rightarrow$ according BC's

Mathematic foundation

Interface capturing

The interface between fluid phases is captured by a Level-Set Method. I.e. a *level set function* $\varphi := \varphi(x, t) \to \mathbb{R}$ is tracked through the fluid domain. The interface is given (implicitly) by the zero-isosurface of this function. It holds:

$$\frac{\partial \varphi}{\partial t} + \vec{v} \cdot \nabla \varphi = 0$$

where $\vec{v}$ is the velocity at the interface, given by NSE.
Benefit of Level-Set Method:

- Implicit surface description eases bubble coalescence/breakup in code
- High density and viscosity ratios ($> 10^3$) possible for simulation

Mathematic foundation

Interface capturing

The interface between fluid phases is captured by a Level-Set Method. I.e. a *level set function* $\varphi := \varphi(x,t) \to \mathbb{R}$ is tracked through the fluid domain. The interface is given (implicitly) by the zero-isosurface of this function. It holds:

$$\frac{\partial \varphi}{\partial t} + \vec{v} \cdot \nabla \varphi = 0$$

where $\vec{v}$ is the velocity at the interface, given by NSE.
Benefit of Level-Set Method:

- Implicit surface description eases bubble coalescence/breakup in code
- High density and viscosity ratios ($> 10^3$) possible for simulation

Mathematic foundation

Interface capturing

The interface between fluid phases is captured by a Level-Set Method. I.e. a *level set function* $\varphi := \varphi(x, t) \to \mathbb{R}$ is tracked through the fluid domain. The interface is given (implicitly) by the zero-isosurface of this function. It holds:

$$\frac{\partial \varphi}{\partial t} + \vec{v} \cdot \nabla \varphi = 0$$

where $\vec{v}$ is the velocity at the interface, given by NSE.
Benefit of Level-Set Method:

- Implicit surface description eases bubble coalescence/breakup in code
- High density and viscosity ratios ($> 10^3$) possible for simulation

Mathematic foundation

Interface implementation (Methods Coupling)

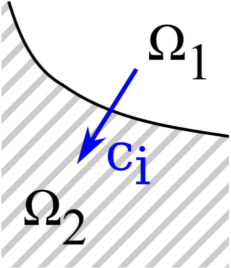
Hydrodynamics are solved by LBM (D2Q9, SRT for our first test)

- Interface becomes a new boundary condition for LBM

$$f_i(x, t+1) = f_{i^*}^{+}(x, t) + 6\Delta x f_i^* c_i \cdot \tilde{v} + R_i$$

- $\tilde{v}$ is the velocity on the interface along the direction c_i

- R_i ensures the jump conditions of the normal stress and corrects the error terms resulting from the bounce back treatment



Mathematic foundation

Interface implementation (Methods Coupling)

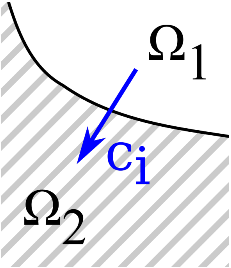
Hydrodynamics are solved by LBM (D2Q9, SRT for our first test)

- Interface becomes a new boundary condition for LBM

$$f_i(x, t+1) = f_{i^*}^{+}(x, t) + 6\Delta x f_i^* c_i \cdot \tilde{v} + R_i$$

- $\tilde{v}$ is the velocity on the interface along the direction c_i

- R_i ensures the jump conditions of the normal stress and corrects the error terms resulting from the bounce back treatment



Mathematic foundation

Interface implementation (Methods Coupling)

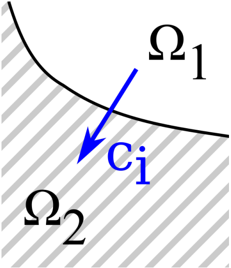
Hydrodynamics are solved by LBM (D2Q9, SRT for our first test)

- Interface becomes a new boundary condition for LBM

$$f_i(x, t+1) = f_{i^*}^{+}(x, t) + 6\Delta x f_i^* c_i \cdot \tilde{v} + R_i$$

- $\tilde{v}$ is the velocity on the interface along the direction c_i

- R_i ensures the jump conditions of the normal stress and corrects the error terms resulting from the bounce back treatment



$\tilde{v}$ is calculated by linear interpolation:

$$\tilde{v} = qv(x_2, t) + (1 - q)v(x_1, t)$$

$\tilde{v}$ is calculated by linear interpolation:

$$\tilde{v} = qv(x_2, t) + (1 - q)v(x_1, t)$$

R_i consists of several parts:

$$R_i = 6\Delta x^2 f_i^* \Lambda_i : A$$

$\tilde{v}$ is calculated by linear interpolation:

$$\tilde{v} = qv(x_2, t) + (1 - q)v(x_1, t)$$

R_i consists of several parts:

$$R_i = 6\Delta x^2 f_i^* \Lambda_i : A$$

- with:

$$\Lambda_i = c_i \otimes c_i - \frac{1}{3}|c_i|^2 \mathbb{I}$$

$$A = -q(1 - q)[S] - (q - 1/2)S^2 + O(\Delta x)$$

- $S^{(k)}$ velocity gradient at x_k
- $[S] = \lim_{\epsilon \to 0}(S(x + \epsilon\vec{n}) - S(x - \epsilon\vec{n}))$ jump of velocity gradient at the interface. Depends on normal, tangent and curvature.

$\tilde{v}$ is calculated by linear interpolation:

$$\tilde{v} = qv(x_2, t) + (1 - q)v(x_1, t)$$

R_i consists of several parts:

$$R_i = 6\Delta x^2 f_i^* \Lambda_i : A$$

- with:

$$\Lambda_i = c_i \otimes c_i - \frac{1}{3}|c_i|^2 \mathbb{I}$$

$$A = -q(1 - q)[S] - (q - 1/2)S^2 + O(\Delta x)$$

- $S^{(k)}$ velocity gradient at x_k
- $[S] = \lim_{\epsilon \to 0}(S(x + \epsilon\vec{n}) - S(x - \epsilon\vec{n}))$ jump of velocity gradient at the interface. Depends on normal, tangent and curvature.

$\tilde{v}$ is calculated by linear interpolation:

$$\tilde{v} = qv(x_2, t) + (1 - q)v(x_1, t)$$

R_i consists of several parts:

$$R_i = 6\Delta x^2 f_i^* \Lambda_i : A$$

- with:

$$\Lambda_i = c_i \otimes c_i - \frac{1}{3}|c_i|^2 \mathbb{I}$$

$$A = -q(1-q)[S] - (q - 1/2)S^2 + O(\Delta x)$$

- $S^{(k)}$ velocity gradient at x_k
- $[S] = \lim_{\epsilon \to 0}(S(x + \epsilon \vec{n}) - S(x - \epsilon \vec{n}))$ jump of velocity gradient at the interface. Depends on normal, tangent and curvature.

Mathematic foundation

Algorithm for LBM with level set

- Create initial interface
- Run level set method to calculate surface description
- Run LBM for a prescribed number of steps
- Run level set method for the same number of steps
- Repeat till end of simulation

Mathematic foundation

Algorithm for LBM with level set

- Create initial interface
- Run level set method to calculate surface description
- Run LBM for a prescribed number of steps
- Run level set method for the same number of steps
- Repeat till end of simulation

Mathematic foundation

Algorithm for LBM with level set

- Create initial interface
- Run level set method to calculate surface description
- Run LBM for a prescribed number of steps
- Run level set method for the same number of steps
- Repeat till end of simulation

Mathematic foundation

Algorithm for LBM with level set

- Create initial interface
- Run level set method to calculate surface description
- Run LBM for a prescribed number of steps
- Run level set method for the same number of steps
- Repeat till end of simulation

Mathematic foundation

Algorithm for LBM with level set

- Create initial interface
- Run level set method to calculate surface description
- Run LBM for a prescribed number of steps
- Run level set method for the same number of steps
- Repeat till end of simulation

Validation

Validation setups

TODO: Hier könnte man Bilder von unserer ersten Simulation zeigen. Leider können wir die Richtigkeit bisher nicht mit Zahlen belegen.

Outlook

$\rightarrow$ *Conclusion habe ich rausgenommen. Ich wüsste zumindest nicht was wir da reinschreiben könnten. Wenn du was weißt, darfst du das aber sehr gerne wieder einfügen.*

Outlook:

- Add correction term to prevent mass loss
- Reduce computational effort: Store Level-Set function only in narrow band around interface, Adalsteinsson and Sethian TODO: Quellen als Footnotes + Uebersichtsfolie
- Include thermal flow (simulate e.g. lava lamp) / Include gravity

Outlook

$\rightarrow$ *Conclusion habe ich rausgenommen. Ich wüsste zumindest nicht was wir da reinschreiben könnten. Wenn du was weißt, darfst du das aber sehr gerne wieder einfügen.*

Outlook:

- Add correction term to prevent mass loss
- Reduce computational effort: Store Level-Set function only in narrow band around interface, Adalsteinsson and Sethian TODO: Quellen als Footnotes + Uebersichtsfolie
- Include thermal flow (simulate e.g. lava lamp) / Include gravity

Outlook

$\rightarrow$ *Conclusion habe ich rausgenommen. Ich wüsste zumindest nicht was wir da reinschreiben könnten. Wenn du was weißt, darfst du das aber sehr gerne wieder einfügen.*

Outlook:

- Add correction term to prevent mass loss
- Reduce computational effort: Store Level-Set function only in narrow band around interface, Adalsteinsson and Sethian TODO: Quellen als Footnotes + Uebersichtsfolie
- Include thermal flow (simulate e.g. lava lamp) / Include gravity

References

- Thömmes, Guido, et al. "A lattice Boltzmann method for immiscible multiphase flow simulations using the level set method." Journal of Computational Physics 228.4 (2009): 1139-1156
- Mitchell, Ian, A Toolbox of Level Set Methods