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

Lorenz Hufnagel, Daniel Zint

BGCE Student Project

x y, 2015

# Multiphase flow - Examples

- Examples
- e.g. Oil in water
- ...

# Multiphase flow - Examples

- Examples
- e.g. Oil in water
- ...

# Multiphase flow - Examples

- Examples
- e.g. Oil in water
- ...

# Macroscopic fluid mechanics

- $N$ immiscible fluids.
- Each has own $\rho_i, \nu_i$
- Hydrodynamics described by (incompressible) NSE

# Macroscopic fluid mechanics

- $N$ immiscible fluids.
- Each has own $\rho_i, \nu_i$
- Hydrodynamics described by (incompressible) NSE

# Macroscopic fluid mechanics

- $N$ immiscible fluids.
- Each has own $\rho_i, \nu_i$
- 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

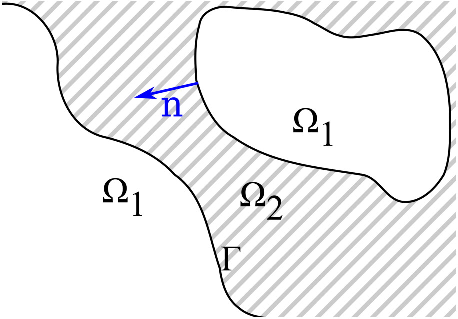


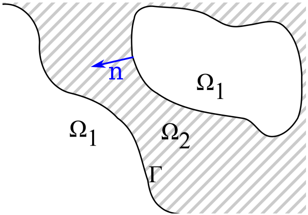

Figure: Two fluid domains $\Omega_i$ and interface $\Gamma$ inbetween

- Velocity across interface is $C_0$-continous

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

# Mathematic foundation

## Interface conditions cont'd

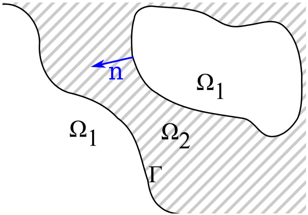



- Normal stress is balanced by surface tension

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

# Mathematic foundation

## Interface conditions cont'd




- Normal stress is balanced by surface tension

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

where $\mathbf{T}_i$ is the stress tensor $\mathbf{T}_i = 2\nu_i\rho_i\mathbf{S}_i - p\mathbf{Id}$ and $\kappa$ 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
- couple the two algorithms

To solve the two-phase problem we need to:

- solve the flow equations $\rightarrow$ LBM
- compute the motion of the interface $\rightarrow$ level set
- couple the two algorithms

To solve the two-phase problem we need to:

- solve the flow equations $\rightarrow$ LBM
- compute the motion of the interface $\rightarrow$ level set
- couple the two algorithms

# Mathematic foundation

## Interface capturing

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

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

# Mathematic foundation

## Interface capturing

Hydrodynamics are solved by LBM.

- Interface becomes a new boundary condition for LBM

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

- $\tilde{u}$ 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

TODO: Bild vom Interface <Coupling und BC's erklären!!...>

# Mathematic foundation

## Interface capturing

Hydrodynamics are solved by LBM.

- Interface becomes a new boundary condition for LBM

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

- $\tilde{u}$ 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

TODO: Bild vom Interface <Coupling und BC's erklären!!...>

# Mathematic foundation

## Interface capturing

Hydrodynamics are solved by LBM.

- Interface becomes a new boundary condition for LBM

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

- $\tilde{u}$ 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

TODO: Bild vom Interface <Coupling und BC's erklären!!...>

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

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

$R_i$ consists of several parts:

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

- with:

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

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

- $S^{(k)}$ velocity gradient at $x_k$
- $[S]$ jump of velocity gradient at the interface. Depends on normal, tangent and curvature.

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

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

$R_i$ consists of several parts:

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

- with:

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

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

- $S^{(k)}$ velocity gradient at $x_k$
- $[S]$ jump of velocity gradient at the interface. Depends on normal, tangent and curvature.

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

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

$R_i$ consists of several parts:

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

- with:

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

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

- $S^{(k)}$ velocity gradient at $x_k$
- $[S]$ 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

# Conclusion & Outlook

$\rightarrow \ldots$

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
- ...

# Conclusion & Outlook

$\rightarrow \ldots$

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
- ...

# Conclusion & Outlook

$\rightarrow \ldots$

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