

# Behandlung freier Oberflächen

Moira Peter, Marcel Messer, Paul Volland

Projektvortrag im Rahmen der Vorlesung Numerische Simulation

07. Februar 2019



# Hintergrund

- ▷ etliche reale Vorgänge, bei denen Fluid mit Gas mit vergleichsweise deutlich geringerer Dichte (bspw. Luft) interagiert
- ▷ physikalische Wirkung des Gases vernachlässigt  
→ sog. freie Oberflächen, da Fluid ungehindert in „leeren“ Raum einströmen kann
- ▷ Beispiele: Bruch eines Damms, Tropfen fällt auf Wasseroberfläche, ...

# Modellierung freier Oberflächen

- ▷ mögliche Krafteinflüsse an der Oberfläche
  - ↪ Oberflächenspannung
  - ↪ Gravitation
  - ↪ innere Spannungen des Fluids

# Modellierung freier Oberflächen

- ▷ mögliche Krafteinflüsse an der Oberfläche
  - ↪ Oberflächenspannung → vernachlässigt
  - ↪ Gravitation
  - ↪ innere Spannungen des Fluids → innere Spannung an Oberfläche muss verschwinden
- ▷ Aufspaltung der Spannung in Normal- und Tangentialrichtung



# Implementierungsaspekte

## Übersicht über grundlegend neue Routinen

- 1 Zelltypen des Gitters dynamisch über Simulationszeit; Typen werden über zeitliche Evolution von Partikeln bestimmt → Partikelkriterium
- 2 Update von Gitterzellen an Oberfläche zwischen Fluid und leerem Gebiet (vgl. Modellierung)
- 3 Partikel-Tracing: Partikel werden per Bewegungsgleichung innerhalb des Gitters bewegt

# Implementierungsaspekte

## Zelltypen & Partikelkriterium

- ▶ Einführung neuer Zelltypen: *typeEmpty* für leeren Raum, *typeSurf* für Zellen der Fluidoberfläche
- ▶ Zelle ist *typeSurf*  $\Leftrightarrow$  mindestens eine Nachbarzelle ist *typeEmpty*
- ▶ *typeSurf*-Zellen benötigen Wissen über Nachbarschaft, relative Positionen von umgebenden leeren Zellen werden gespeichert (alle Kombinationen aus {N, W, S, E} möglich)
- ▶ dynamisches Feld der Zelltypen, d.h. in jedem Zeitschritt werden sowohl alle Zelltypen als auch deren Nachbarschaft geupdatet

# Implementierungsaspekte

## Zelltypen & Partikelkriterium

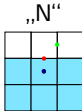
- ▶ Einführung neuer Zelltypen: *typeEmpty* für leeren Raum, *typeSurf* für Zellen der Fluidoberfläche
- ▶ Zelle ist *typeSurf*  $\Leftrightarrow$  mindestens eine Nachbarzelle ist *typeEmpty*
- ▶ *typeSurf*-Zellen benötigen Wissen über Nachbarschaft, relative Positionen von umgebenden leeren Zellen werden gespeichert (alle Kombinationen aus {N, W, S, E} möglich)
- ▶ dynamisches Feld der Zelltypen, d.h. in jedem Zeitschritt werden sowohl alle Zelltypen als auch deren Nachbarschaft geupdatet
- ▶ Wann ist Zelle Fluid bzw. leer?
  - $\Rightarrow$  Partikelkriterium: befindet sich mindestens ein Partikel in einer Zelle, so ist diese Fluid, ansonsten leer
  - $\rightarrow$  nach diesem binären Kriterium werden anschließend wieder die Oberflächenzellen bestimmt



# Implementierungsaspekte

## Update von Oberflächenzellen

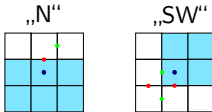
- ▷ für *typeSurf*-Zellen ist gesonderte Behandlung unabhängig vom Lösen für innere Fluidzellen notwendig, um die Modellgleichungen zu erfüllen
- ▷ je nach vorliegender Nachbarschaft werden Werte für  $u, v, p$  in nächster Nachbarschaft der betreffenden Zelle gesetzt
- ▷ 5 grundsätzlich verschiedene Fälle zu betrachten, die eine weitere Untergliederung aufweisen



# Implementierungsaspekte

## Update von Oberflächenzellen

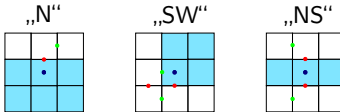
- ▷ für *typeSurf*-Zellen ist gesonderte Behandlung unabhängig vom Lösen für innere Fluidzellen notwendig, um die Modellgleichungen zu erfüllen
- ▷ je nach vorliegender Nachbarschaft werden Werte für  $u, v, p$  in nächster Nachbarschaft der betreffenden Zelle gesetzt
- ▷ 5 grundsätzlich verschiedene Fälle zu betrachten, die eine weitere Untergliederung aufweisen



# Implementierungsaspekte

## Update von Oberflächenzellen

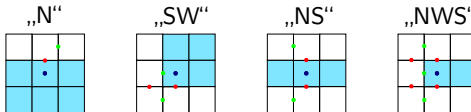
- ▷ für *typeSurf*-Zellen ist gesonderte Behandlung unabhängig vom Lösen für innere Fluidzellen notwendig, um die Modellgleichungen zu erfüllen
- ▷ je nach vorliegender Nachbarschaft werden Werte für  $u, v, p$  in nächster Nachbarschaft der betreffenden Zelle gesetzt
- ▷ 5 grundsätzlich verschiedene Fälle zu betrachten, die eine weitere Untergliederung aufweisen



# Implementierungsaspekte

## Update von Oberflächenzellen

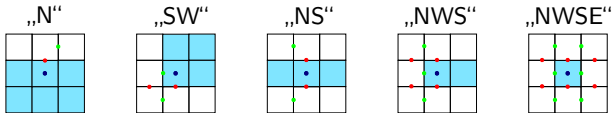
- ▷ für *typeSurf*-Zellen ist gesonderte Behandlung unabhängig vom Lösen für innere Fluidzellen notwendig, um die Modellgleichungen zu erfüllen
- ▷ je nach vorliegender Nachbarschaft werden Werte für  $u, v, p$  in nächster Nachbarschaft der betreffenden Zelle gesetzt
- ▷ 5 grundsätzlich verschiedene Fälle zu betrachten, die eine weitere Untergliederung aufweisen



# Implementierungsaspekte

## Update von Oberflächenzellen

- ▷ für *typeSurf*-Zellen ist gesonderte Behandlung unabhängig vom Lösen für innere Fluidzellen notwendig, um die Modellgleichungen zu erfüllen
- ▷ je nach vorliegender Nachbarschaft werden Werte für  $u, v, p$  in nächster Nachbarschaft der betreffenden Zelle gesetzt
- ▷ 5 grundsätzlich verschiedene Fälle zu betrachten, die eine weitere Untergliederung aufweisen



# Implementierungsaspekte

## Partikel-Tracing

- ▷ gebe Partikel mit zufälliger Position in innere Fluidzellen nur zu Beginn + in jedem Zeitschritt an Kanten der Inflow-Zellen
- ▷ Ausbreitung der Partikel definiert dynamische Zelltypen
- ▷ Partikel außerhalb des physikalischen Gebiets bzw. in innere Obstacle-Zelle gelangt → Löschung
- ▷ Datenstruktur: dynamische Liste an Arrays; jeder Listeneintrag  $i$  enthält die Positionen  $x, y$  des  $i$ -ten Partikels
- ▷ Bewegung über Bewegungsgleichung für einzelne Teilchen bestimmt

# Implementierungsaspekte

## Partikel-Tracing

- ▷ Bewegung über Bewegungsgleichung für einzelne Teilchen bestimmt  
→ zwei unterschiedliche Varianten ausprobiert:

↪ Leap-Frog

$$x^{(i+1)} = x^{(i)} + \Delta t \cdot u^{(i+\frac{1}{2})}$$

$$\text{mit } u^{(i+\frac{1}{2})} := \frac{1}{2} \left( u^{(i)} + u^{(i+1)} \right)$$

# Implementierungsaspekte

## Partikel-Tracing

- ▷ Bewegung über Bewegungsgleichung für einzelne Teilchen bestimmt  
→ zwei unterschiedliche Varianten ausprobiert:

↪ Leap-Frog

$$x^{(i+1)} = x^{(i)} + \Delta t \cdot u^{(i+\frac{1}{2})}$$

$$\text{mit } u^{(i+\frac{1}{2})} := \frac{1}{2} \left( u^{(i)} + u^{(i+1)} \right)$$

↪ Taylor-Entwicklung

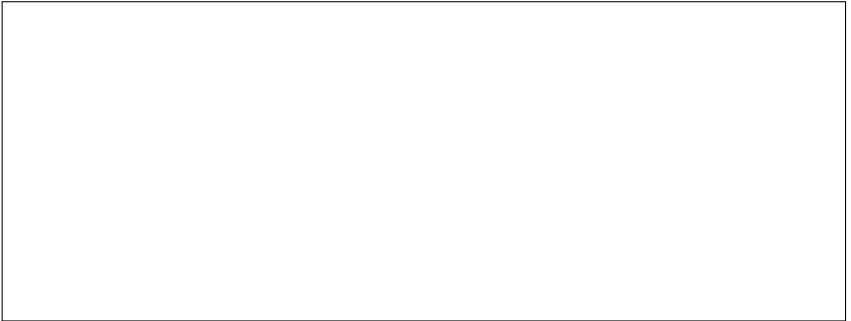
$$x_k^{(i+1)} = x_k^{(i)} + \Delta t \cdot u_k^{(i+\frac{1}{2})} + \sum_{\substack{l=0 \\ l \neq k}}^L \frac{\Delta t^2}{2} \frac{F_{x,kl}}{m_k m_l}$$

$$\text{mit } u_k^{(i+\frac{1}{2})} := \frac{1}{2} \left( u_k^{(i)} + u_k^{(i+1)} \right)$$



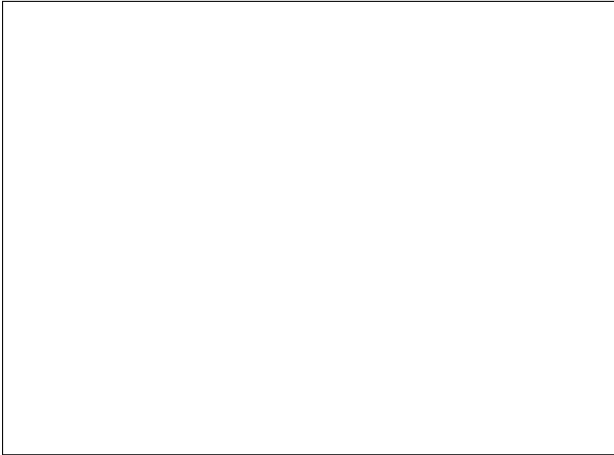
# Resultate

## Beispiel 1: „Aorta“



# Resultate

## Beispiel 2: „Einguss“



Vielen Dank fürs Kommen  
und die Aufmerksamkeit!

Fragen?!