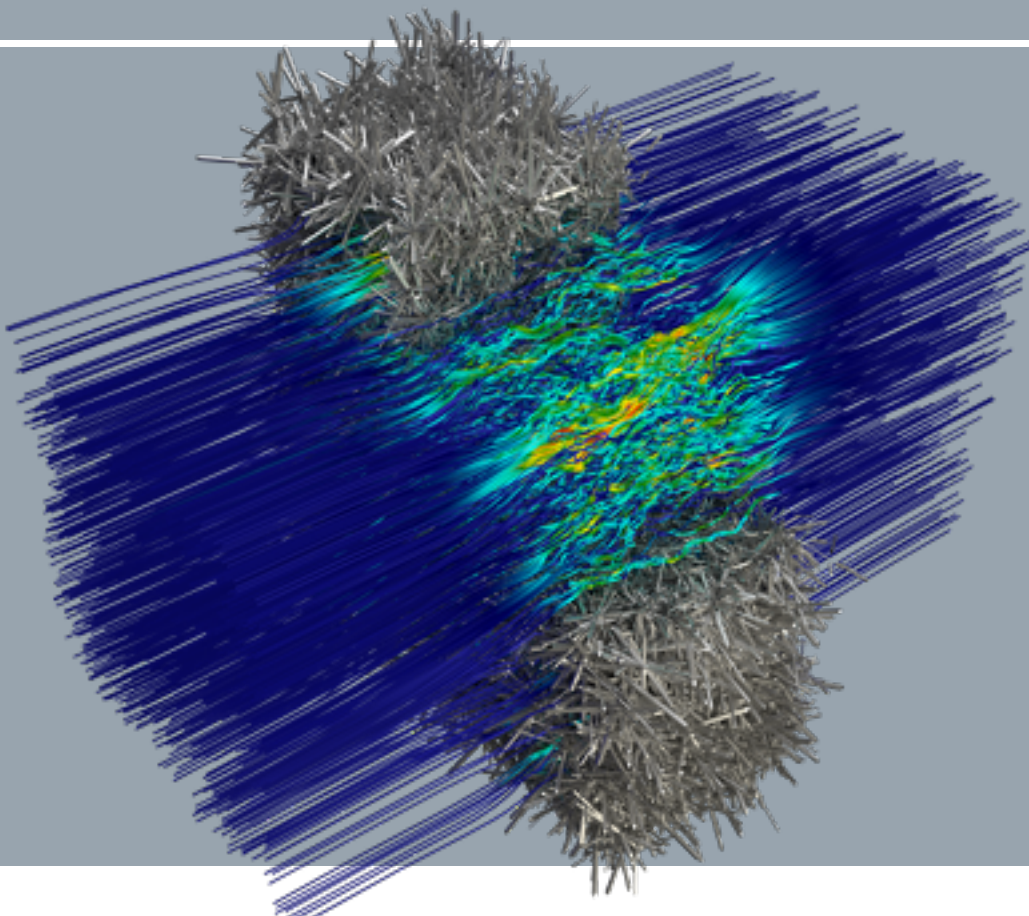
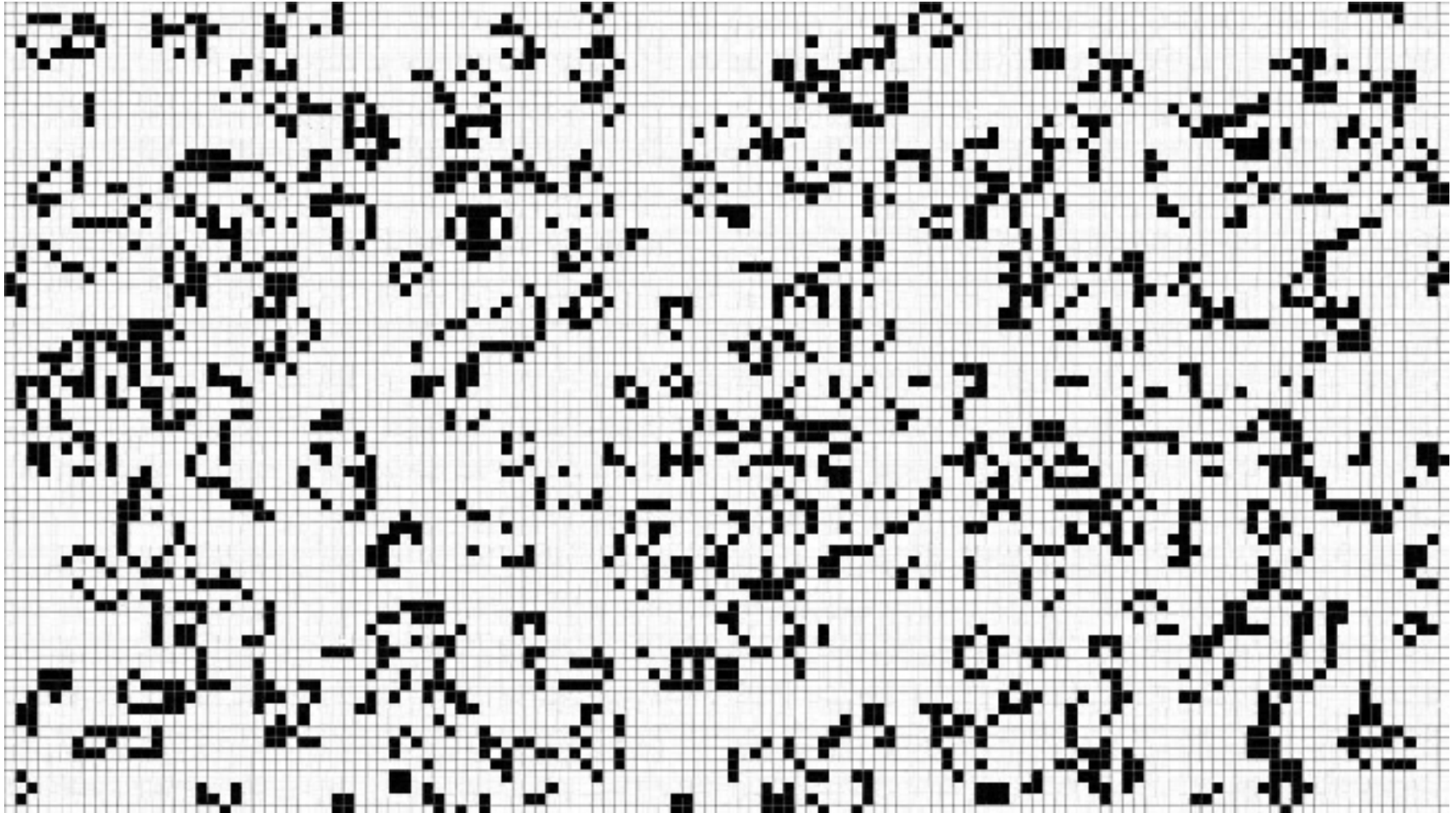


Algorithmik kontinuierlicher Systeme

Zelluläre Automaten und die Lattice-Boltzmann-Methode



Zelluläre Automaten



Geschichte der Zellulären Automaten



Stanislaw Ulam

- Monte-Carlo-Methode
- Manhattan-Projekt



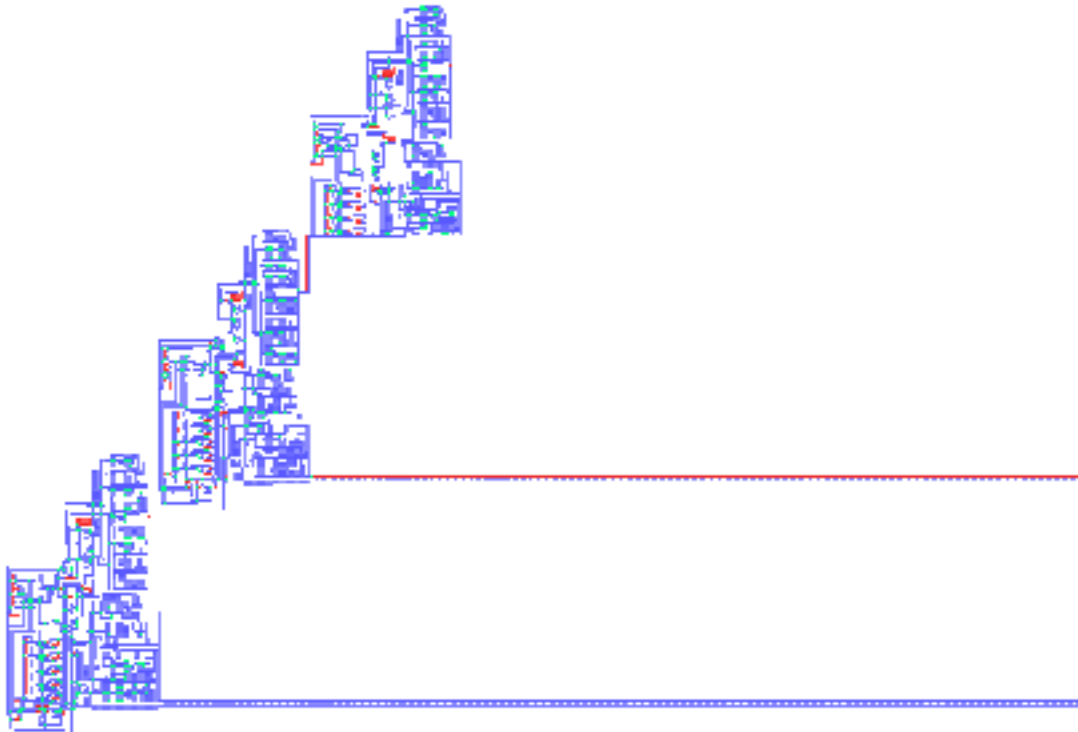
John von Neumann

- Von-Neumann-Architektur
- Spieltheorie
- Manhattan-Projekt



Von-Neumann-Universalkonstruktor

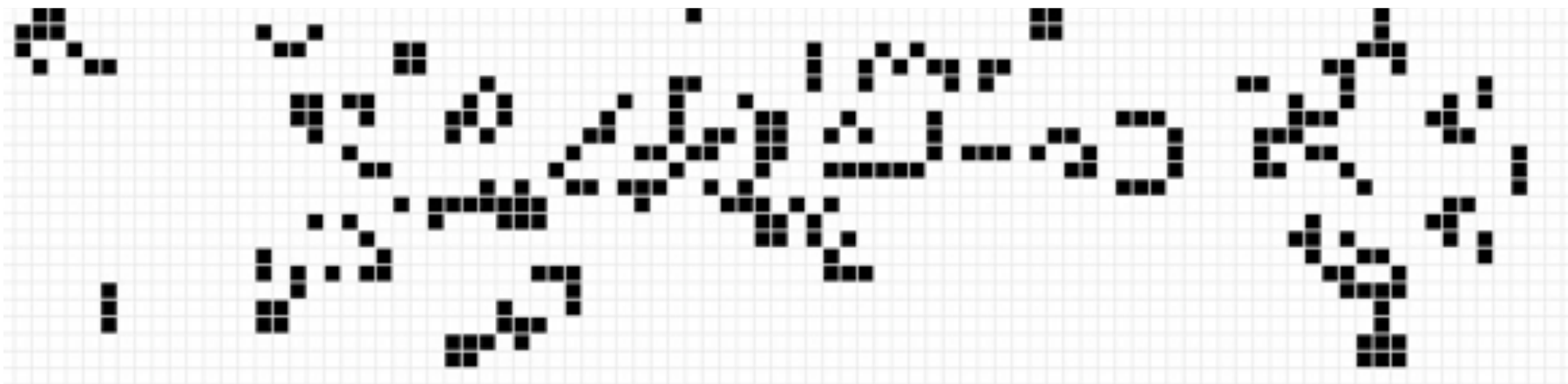
- Ein selbstreplizierender Zellulärer Automat.
- Zweidimensionales Gitter mit 29 Zuständen.
- Entwickelt 1940, erstmals implementiert 1995



Was sind Zelluläre Automaten?

- Reguläre Gitter aus Zellen.
- Endlich viele Zustände pro Zelle (oft nur zwei).
- Simultane Updates anhand einfacher Regeln
- Regeln verwenden nur Nachbarzellen.

Berühmtestes Beispiel: Conway "Game of Life"



Game of Life

- Zweidimensionales Gitter beliebiger Größe
- Jede Zelle hat zwei Zustände: Tot oder Lebendig

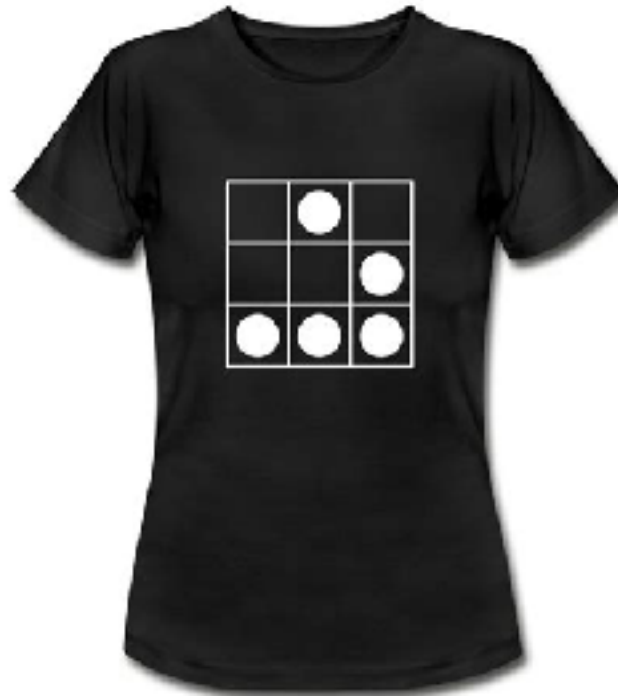
Die Regeln:

1. Genau drei lebende Nachbarn: Zelle wird lebendig
2. Weniger als zwei lebende Nachbarn: Zelle stirbt
3. Mehr als drei lebende Nachbarn: Zelle stirbt

Überraschende Dynamik, hat bis heute viele Fans:

www.conwaylife.com/wiki

Beispiel 1 - Der Gleiter



<http://www.conwaylife.com/wiki/Glider>

Beispiel 2 - Sir Robin



http://www.conwaylife.com/wiki/Sir_Robin

Zusammenfassung

Zelluläre Automaten beschreiben mit einfachen, lokalen und inhärent parallelen Regeln komplexe Sachverhalte

Extrembeispiel: **Rule 110**

111	110	101	100	011	010	001	000
0	1	1	0	1	1	1	0

- Eindimensional
- Turing-Vollständig (Matthew Cook, 2000)

Kann man das Verhalten von Flüssigkeiten und Gasen durch Zelluläre Automaten beschreiben?

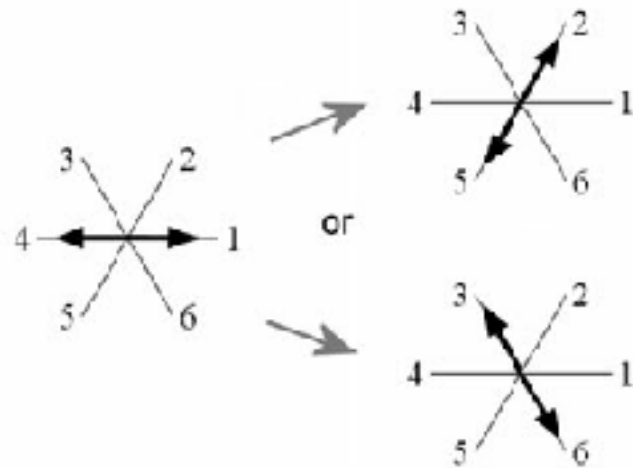
Lattice-Gas Zelluläre Automaten (LGCA)

- Jede Zelle enthält mehrere (z.B. 4 oder 6) gerichtete Einträge. Jeder Eintrag ist entweder 0 oder 1.
- Zweistufige Update-Regel:
 - **Stream:** Jeder Eintrag "wandert" in die entsprechende Nachbarzelle
 - **Collide:** Innerhalb einer Zelle werden Einträge getauscht.

FHP-1 Schema

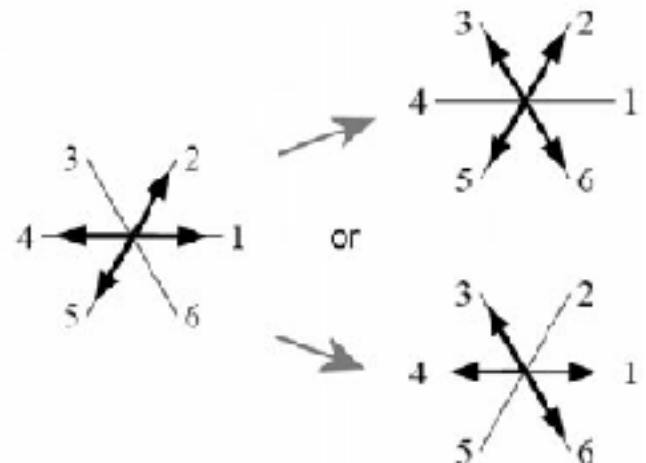
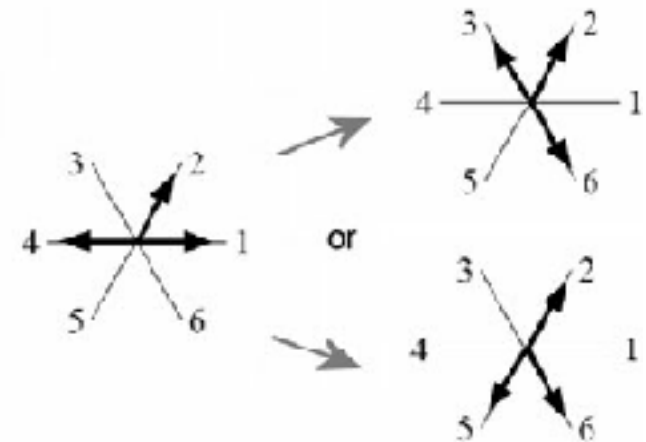
before collision
(after propagation step)

after collision



before collision
(after propagation step)

after collision



Vor- und Nachteile von LGCA

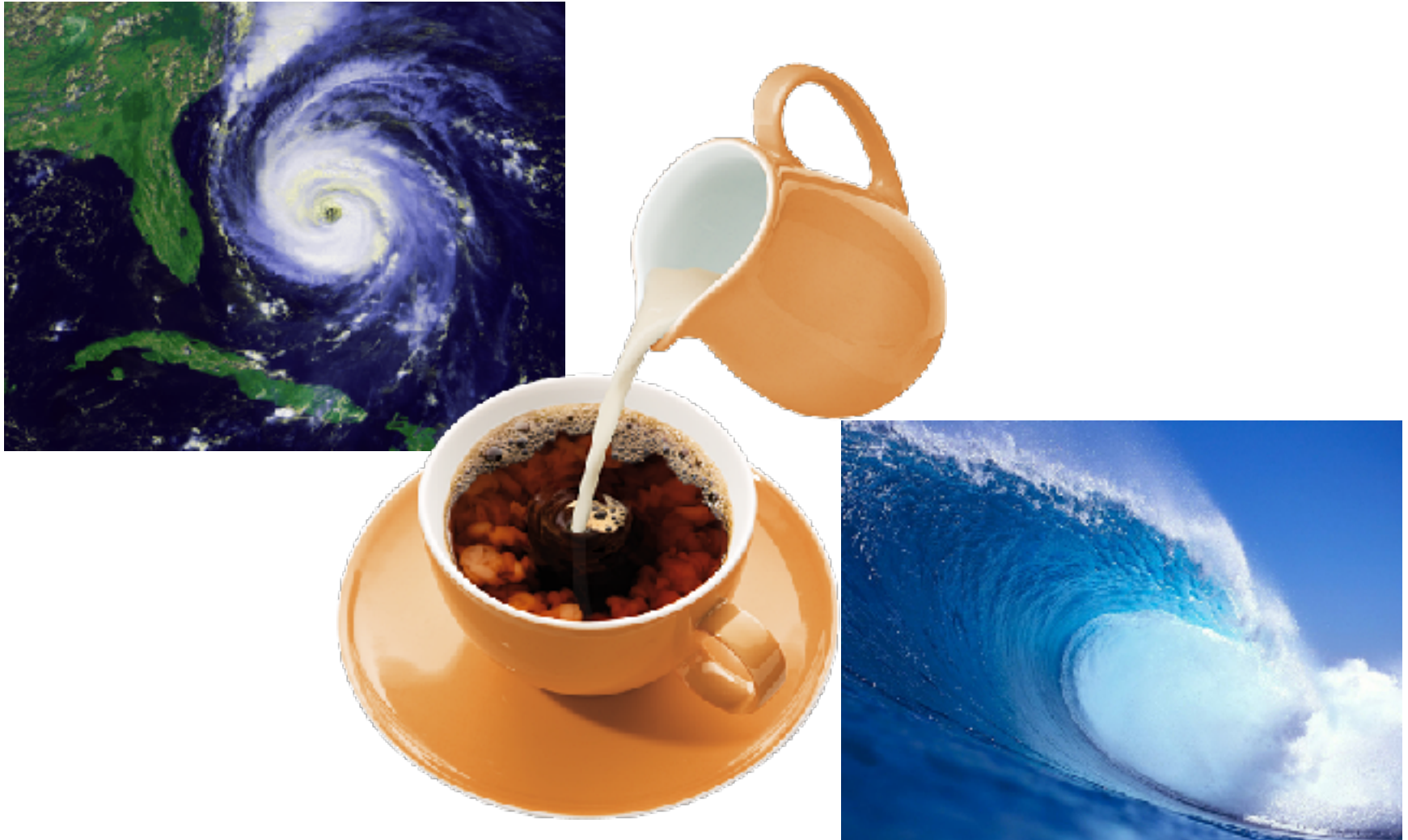
Vorteile

- Einfach zu implementieren.
- Inhärent parallel.
- Trivial Massen- und Energieerhaltend.

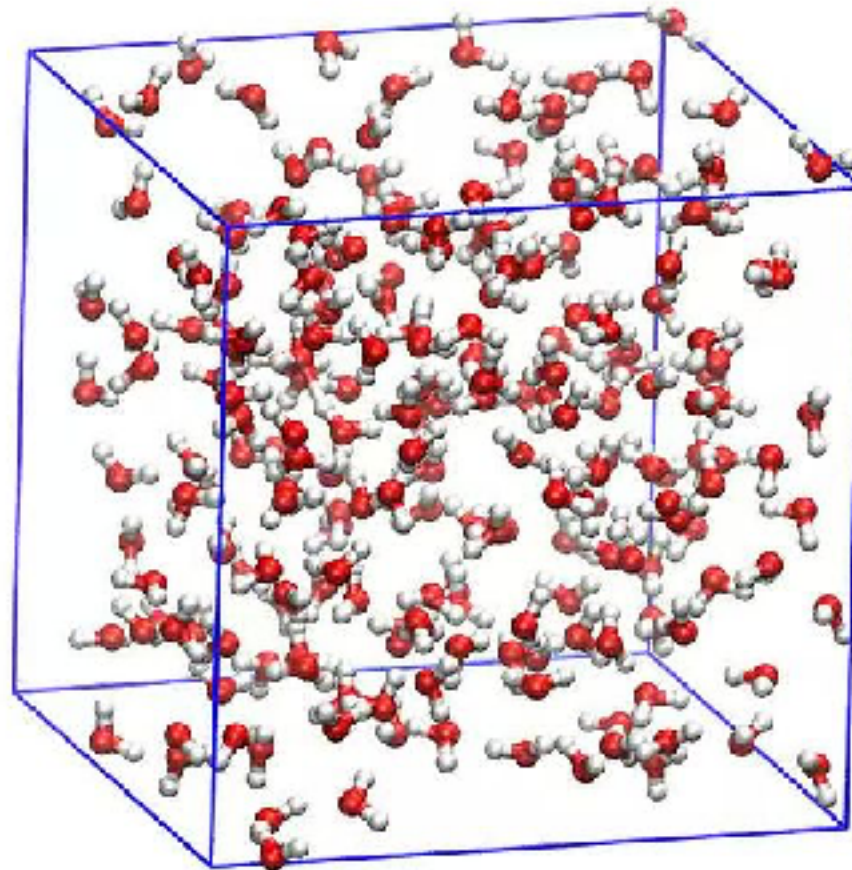
Nachteile

- "spurious invariants".
- Auf modernen Computern sehr ineffizient.
- Physikalisch nicht immer korrekt.

Mathematische Modellierung von Fluiden



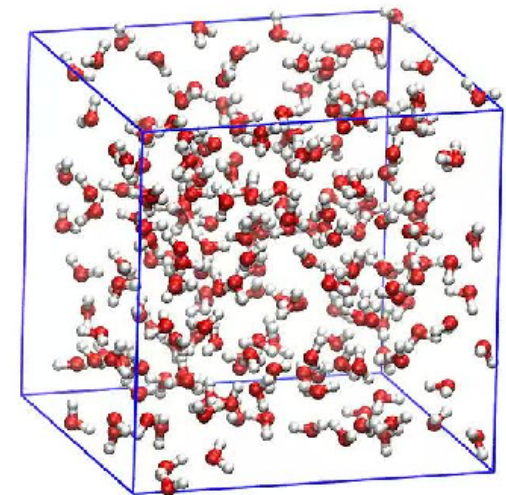
Modell 1 - Molekulardynamik



Modell 1 - Molekulardynamik

1. Newtonsches Gesetz (Trägheit)

Corpus omne perseverare in statu suo quiescendi vel movendi uniformiter in directum, nisi quatenus illud a viribus impressis cogitur statum suum mutare.



2. Newtonsches Gesetz ($F = m a$)

Mutationem motus proportionalem esse vi motrici impressae, et fieri secundum lineam rectam qua vis illa imprimitur.

Modell 1 - Molekulardynamik

Simulationsgrößen:

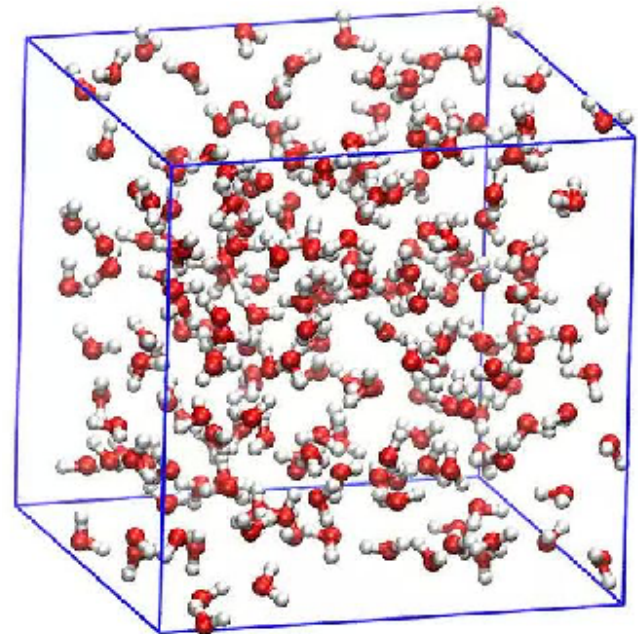
- Position x_i
- Geschwindigkeit v_i

Leapfrog-Verfahren:

$$x_i = x_{i-1} + v_{i-\frac{1}{2}} \Delta t$$

$$a_i = F(x_i)$$

$$v_{i+\frac{1}{2}} = v_{i-\frac{1}{2}} + a_i \Delta t$$



Modell 1 - Molekulardynamik

Vorteile

- Einfaches Modell.
- Beschreibt auch komplizierte Sachverhalte (chemische Reaktionen, Katalyse, nichtnewtonsche Fluide).
- Relativ einfach zu implementieren und parallelisieren.

Nachteile

- $1\text{mol} = 6.022 \times 10^{23}$ Teilchen
- Derzeit schnellster Supercomputer: 200 Petaflops
(1 Petaflop = 10^{15} Gleitkommaoperationen/Sekunde)
- ca. ein Zeitschritt pro Monat ...

Modell 2 - Die Navier-Stokes-Gleichungen



Die Navier-Stokes-Gleichungen

Impulsgleichungen:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \mu \Delta \mathbf{u} + (\lambda + \mu) \nabla (\nabla \cdot \mathbf{u}) + \rho g$$

Kontinuitätsgleichung:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

Der Einfachheit halber betrachten wir im Folgenden nur den inkompressiblen Fall...

Die inkompressiblen Navier-Stokes-Gleichungen

Impulsgleichungen:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \mu \Delta \mathbf{u} + \rho g$$

Kontinuitätsgleichung:

$$\nabla \cdot \mathbf{u} = 0$$

Physikalische Größen: Geschwindigkeit \mathbf{u} , Dichte ρ , Zeit t , Druck p , dynamische Viskosität μ , Gravitationskonstante g

Entdimensionalisierung

Wichtige Erkenntnis:

Die Navier-Stokes-Gleichungen beschreiben beliebige Fluide, unabhängig von der Größe des Systems.

Insbesondere verhalten sich Systeme mit gleicher **Reynoldszahl** ähnlich.

Reynoldszahl:
$$\text{Re} = \frac{vd}{\nu}$$

Geschwindigkeit v , charakteristische Länge d , kinematische Viskosität ν

Die inkompressiblen, entdimensionalisierten Navier-Stokes-Gleichungen

Impulsgleichungen:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \frac{1}{\text{Re}} \Delta \mathbf{u} + \rho g$$

Kontinuitätsgleichung:

$$\nabla \cdot \mathbf{u} = 0$$

Alle Größen (Geschwindigkeit, Dichte, Druck,...) sind hierbei dimensionslos.

Lösen der Navier-Stokes-Gleichungen

Die Navier-Stokes Gleichungen sind ein gekoppeltes System von Differentialgleichungen.

Für solche Systeme gibt es etablierte Lösungsverfahren:

- Finite-Differenzen-Methode
- Finite-Elemente-Methode
- Finite-Volumen-Methode
- ...

Lösen der Navier-Stokes-Gleichungen

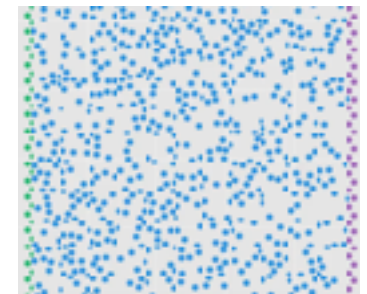
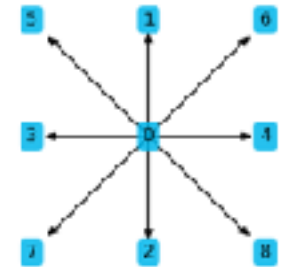
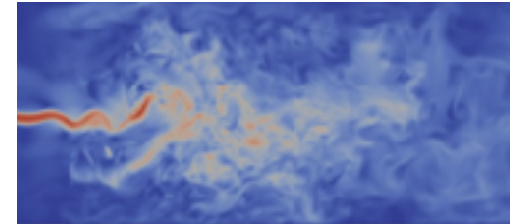
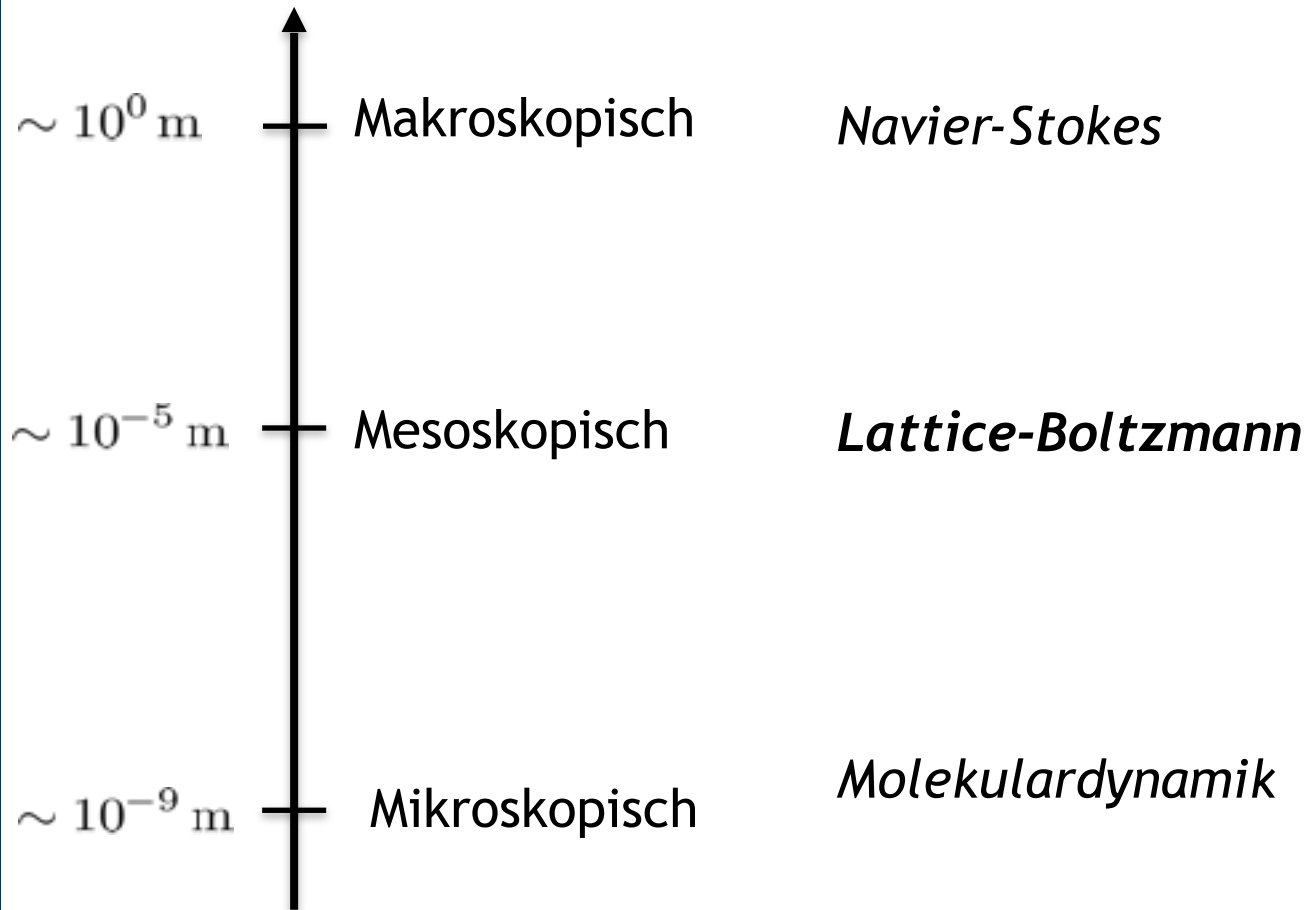
Lösen mit FEM, FDM oder FVM:

- Das Simulationsgebiet wird durch ein Gitter beschrieben
- Die Differentialgleichungen werden auf diesem Gitter diskretisiert
- Das resultierende Gleichungssystem wird mit etablierten Lösungsverfahren (direkte Löser, iterative Löser, ...) gelöst

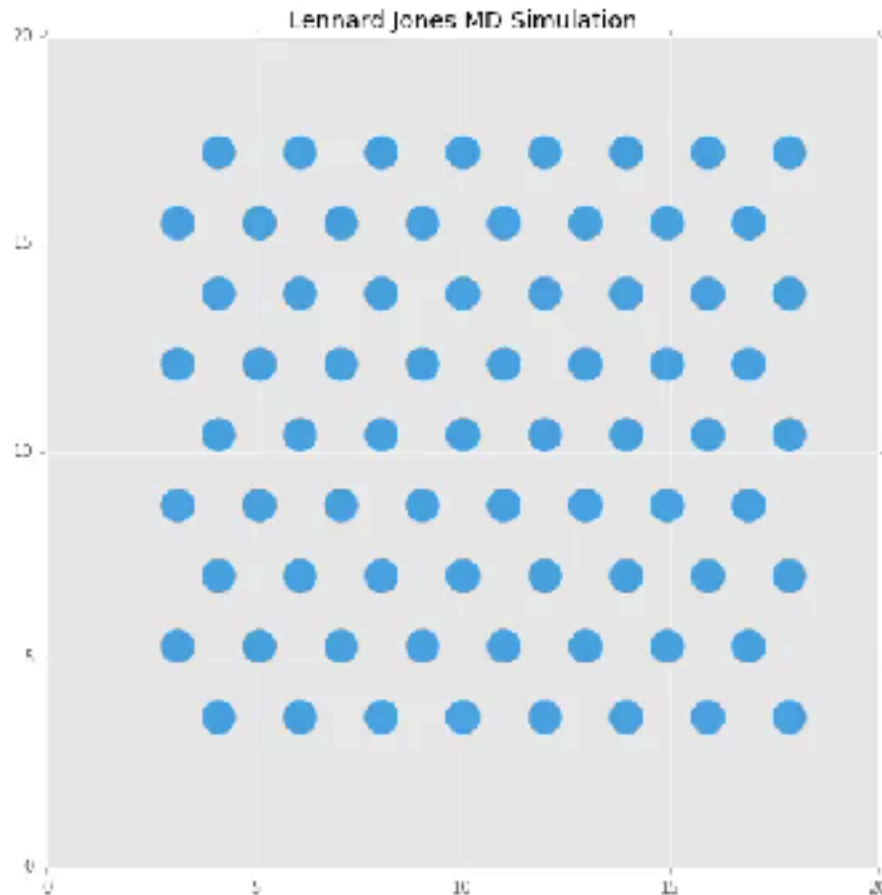
Probleme:

- Gittergenerierung ist eine Kunst für sich
- Behandlung von Rändern ist kompliziert
- Schwierig zu optimieren und zu parallelisieren

Modell 3 - Lattice-Boltzmann-Methode



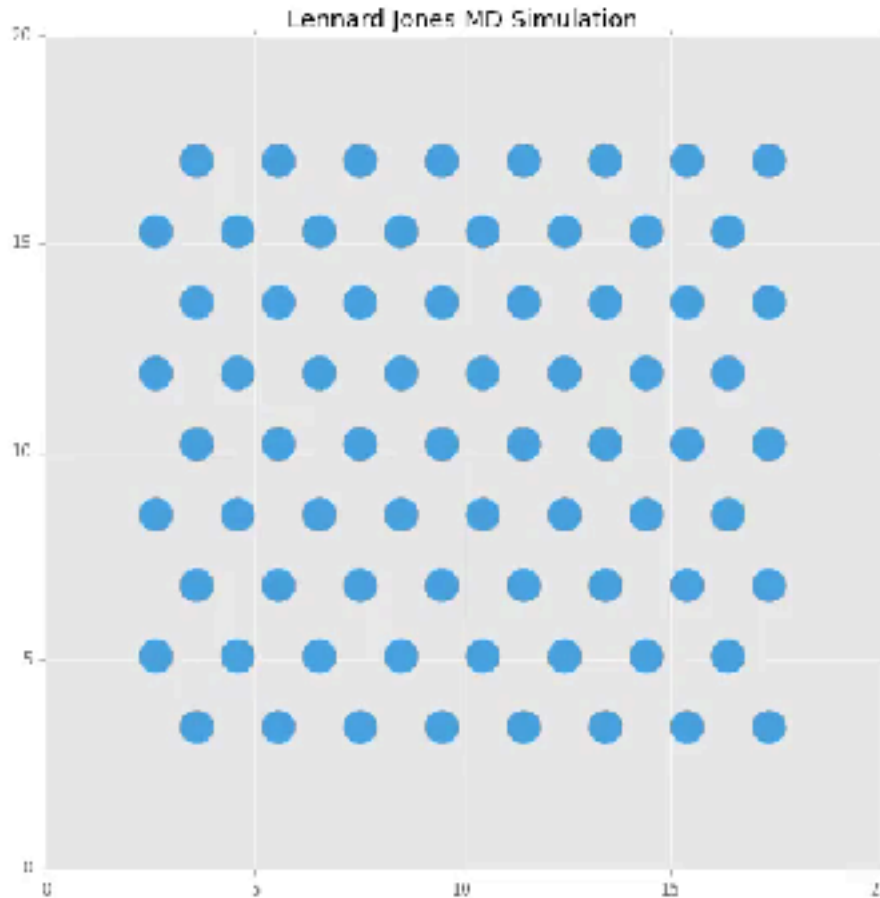
Statistische Mechanik



- klassische MD Simulation
- vollelastische Stöße
- Anfangsbedingungen:
 - gleiche Richtung (rechts oben)
 - gleiche Geschwindigkeit

Ergebnis: Chaos

Statistische Mechanik



- klassische MD Simulation
- vollelastische Stöße
- Anfangsbedingungen:
 - gleiche Richtung (links)
 - gleiche Geschwindigkeit

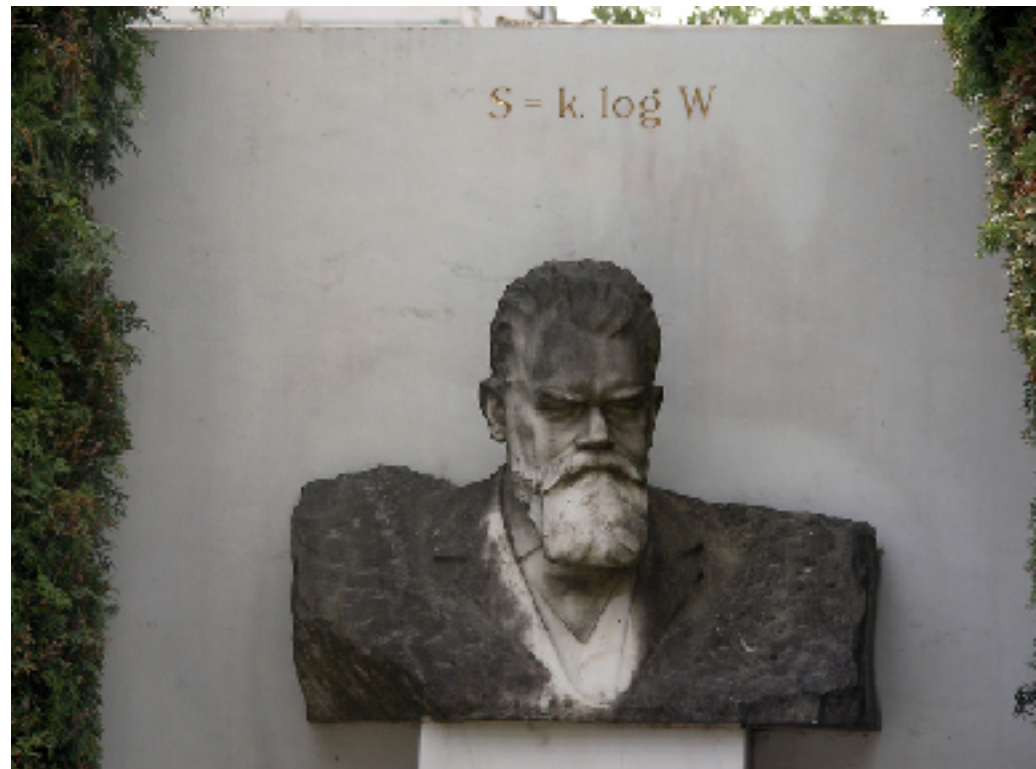
Ergebnis: Chaos

Statistische Mechanik - Zwischenfazit

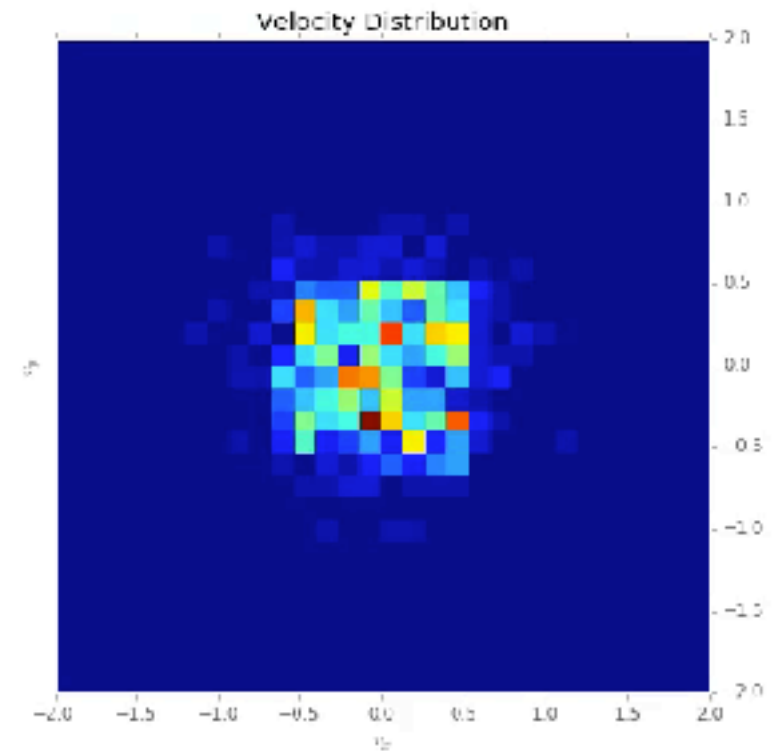
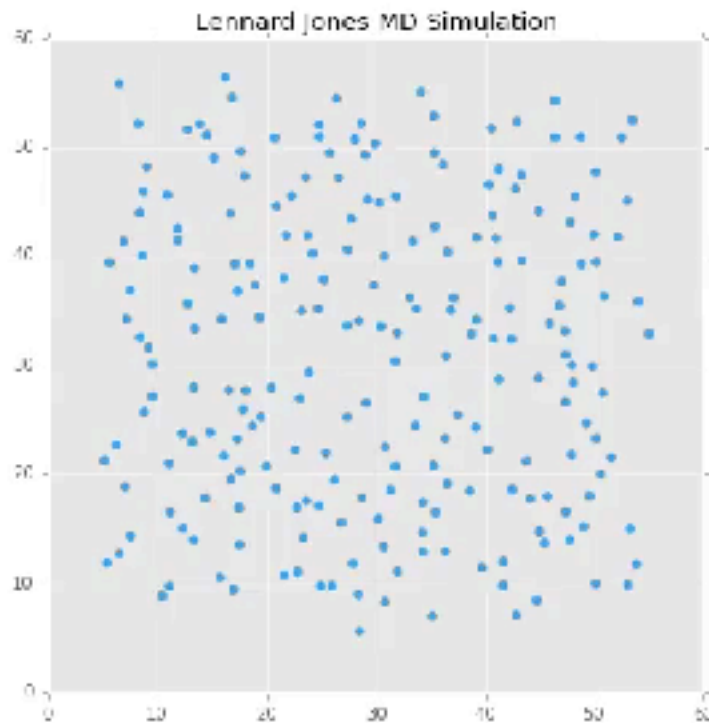
- Systeme mit gleicher Energie (= Anfangsgeschwindigkeit) enden im selben **Gleichgewichtszustand**

Der Grund:

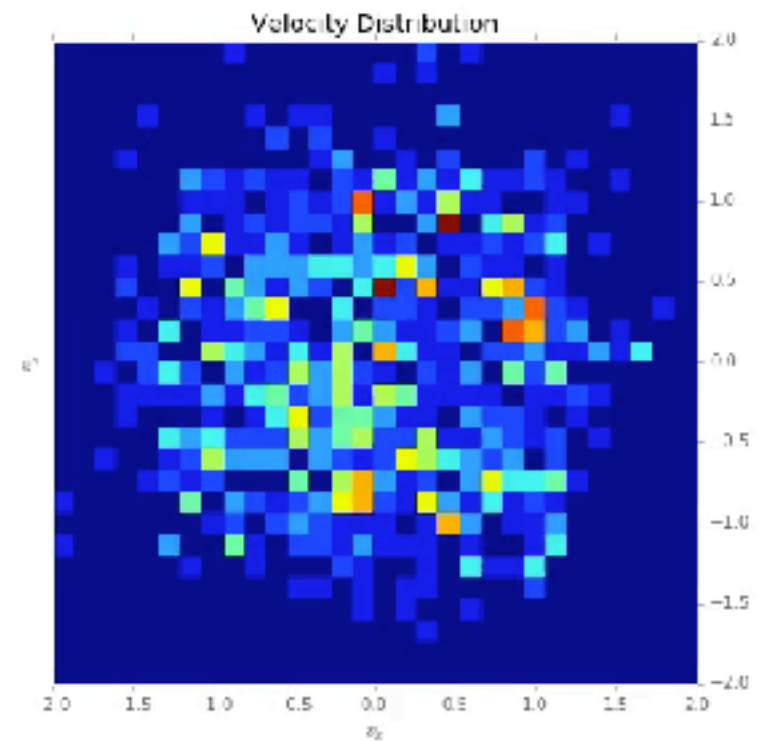
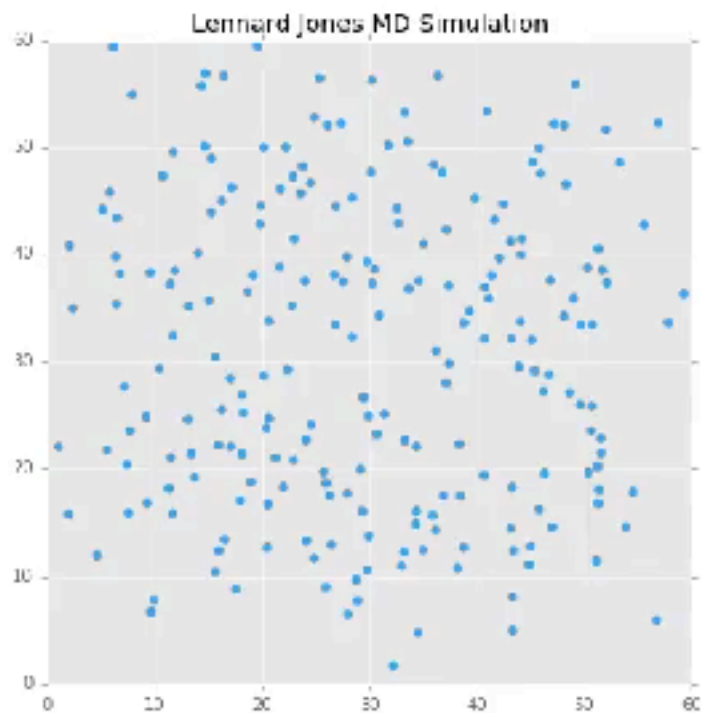
- Die Wahrscheinlichkeit für "besondere" Zustände nimmt rapide ab.
- Die Entropie (Wahrscheinlichkeit aller Mikrozustände) nimmt stetig zu.



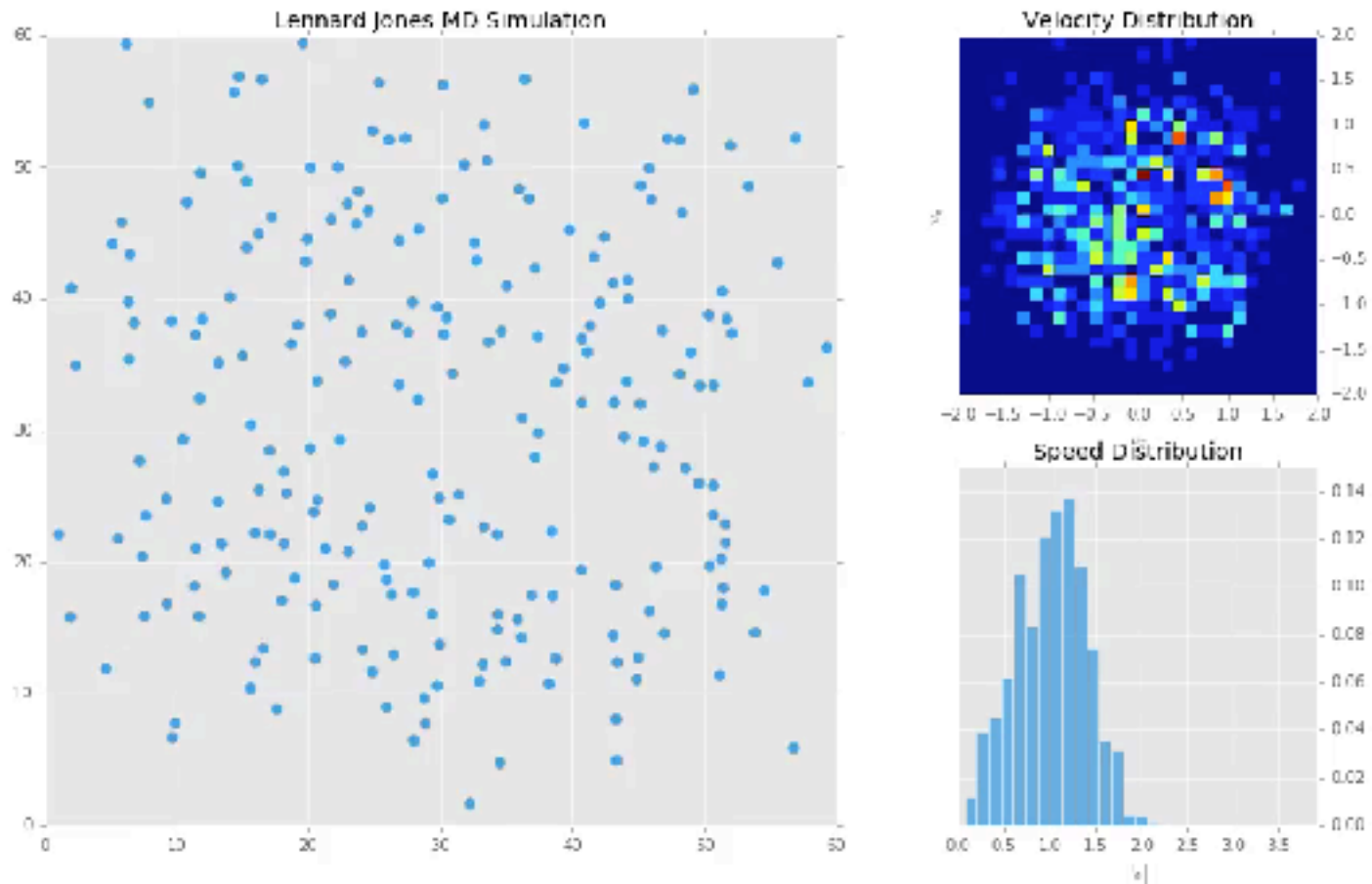
Geschwindigkeitshistogramm 1 (langsam)



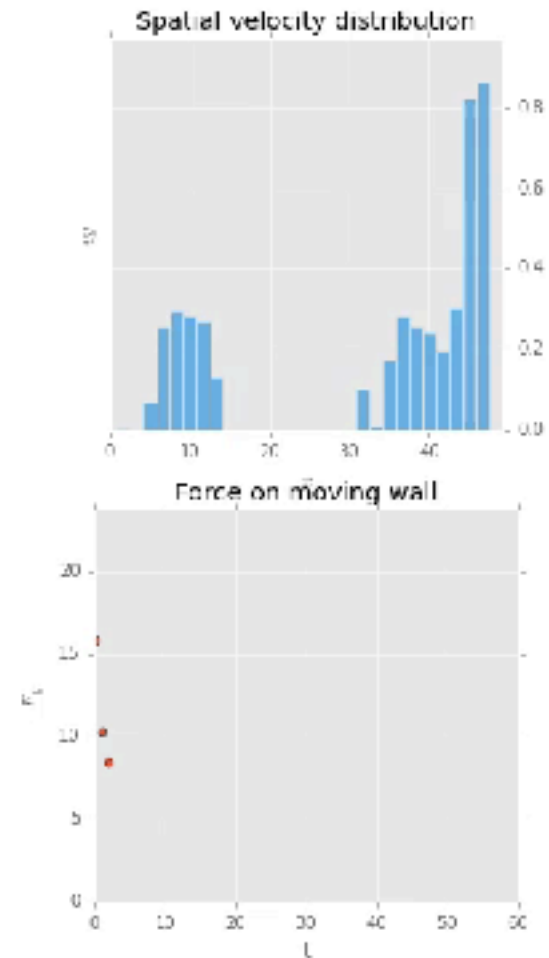
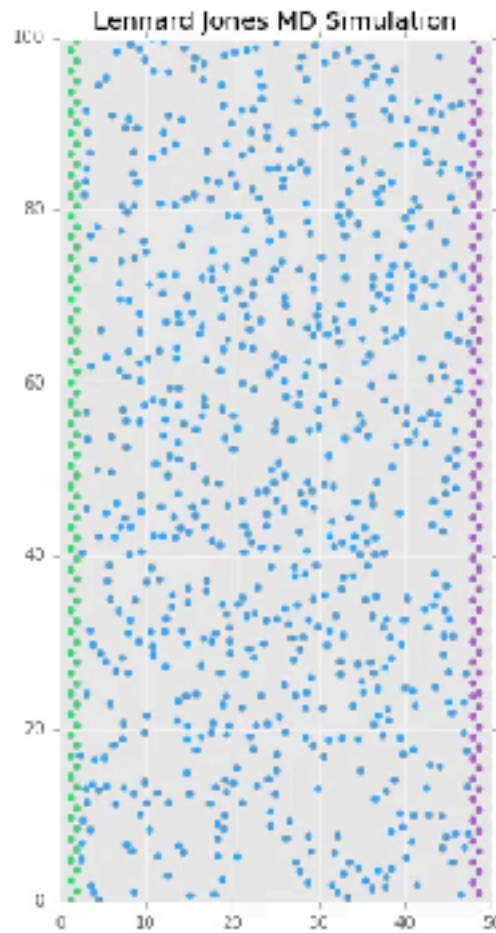
Geschwindigkeitshistogramm 2 (schnell)



Geschwindigkeitsverteilung



Fluss in einem Kanal



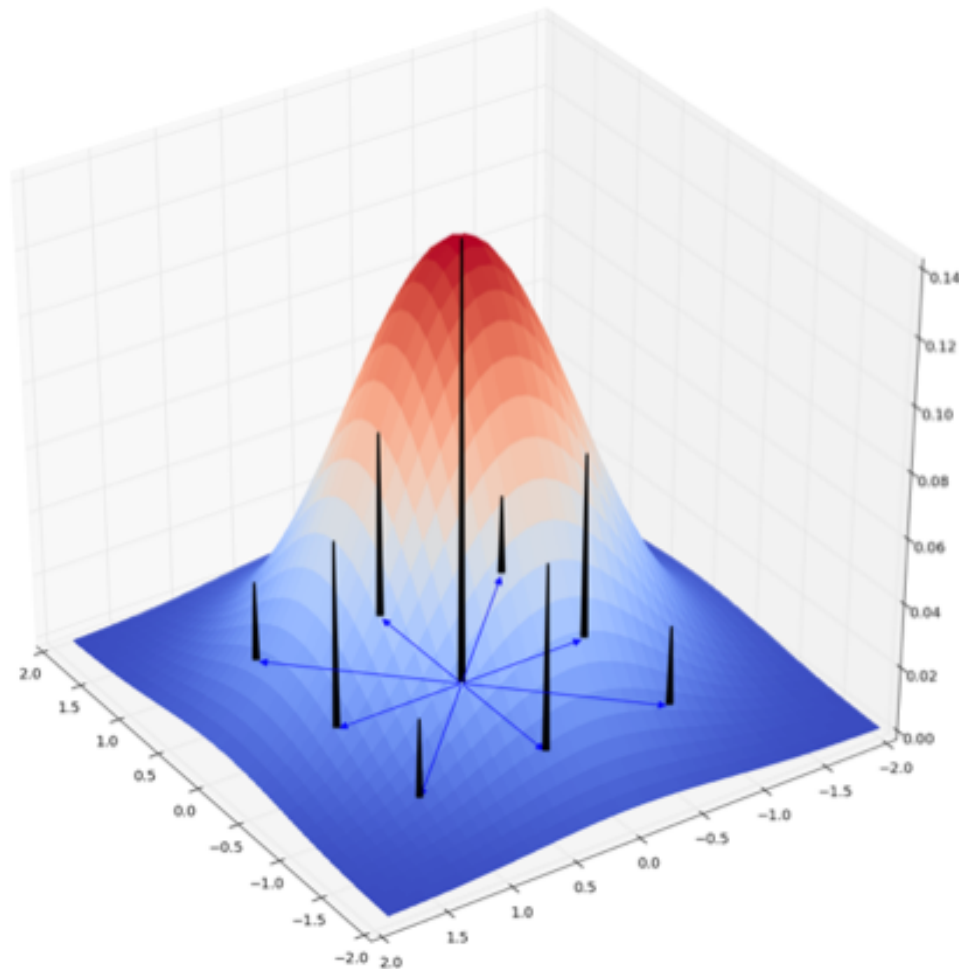
Fazit

- Verschiedene Anfangsbedingungen (mit gleicher Energie) liefern gleiche Geschwindigkeitsverteilungen.
- Die Geschwindigkeitsverteilung ist radialsymmetrisch.
- Je größer die Systemenergie, desto breiter die Geschwindigkeitsverteilung.

Idee:

- Statt der Position und Geschwindigkeit von tausenden Partikeln speichern wir lediglich die Parameter der Verteilung (Radius, Mittelpunkt, Energie, ...).
- Zurück zu Zellulären Automaten!

Die Lattice-Boltzmann-Methode



Die Lattice-Boltzmann-Methode

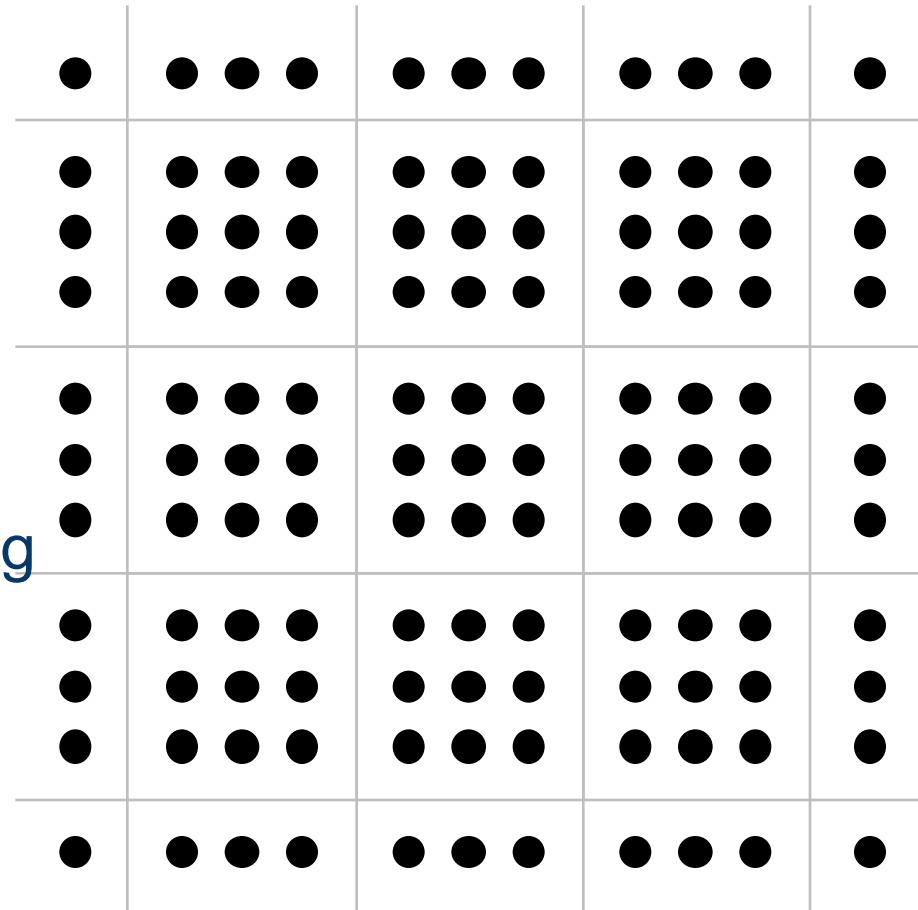
Beispiel: D2Q9 Methode

- 2D Strömungssimulation
- Jede Zelle hat 9 Einträge
- Jeder Eintrag ist eine Wahrscheinlichkeitsverteilung (Gleitkommazahl)

Update-Regel wie bei LGCA:

1. Stream

2. Collide



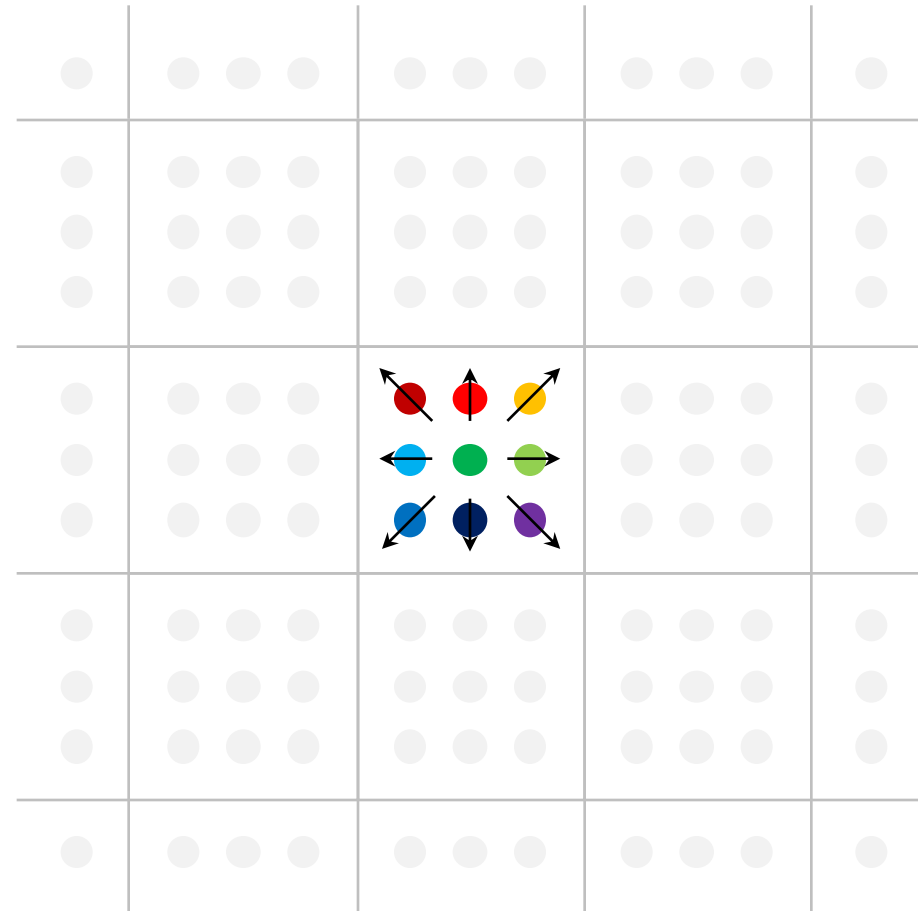
Die Lattice-Boltzmann-Methode

Beispiel: D2Q9 Methode

Update-Regel wie bei LGCA:

1. Stream

Jeder Eintrag wird in eine Nachbarzelle propagiert.



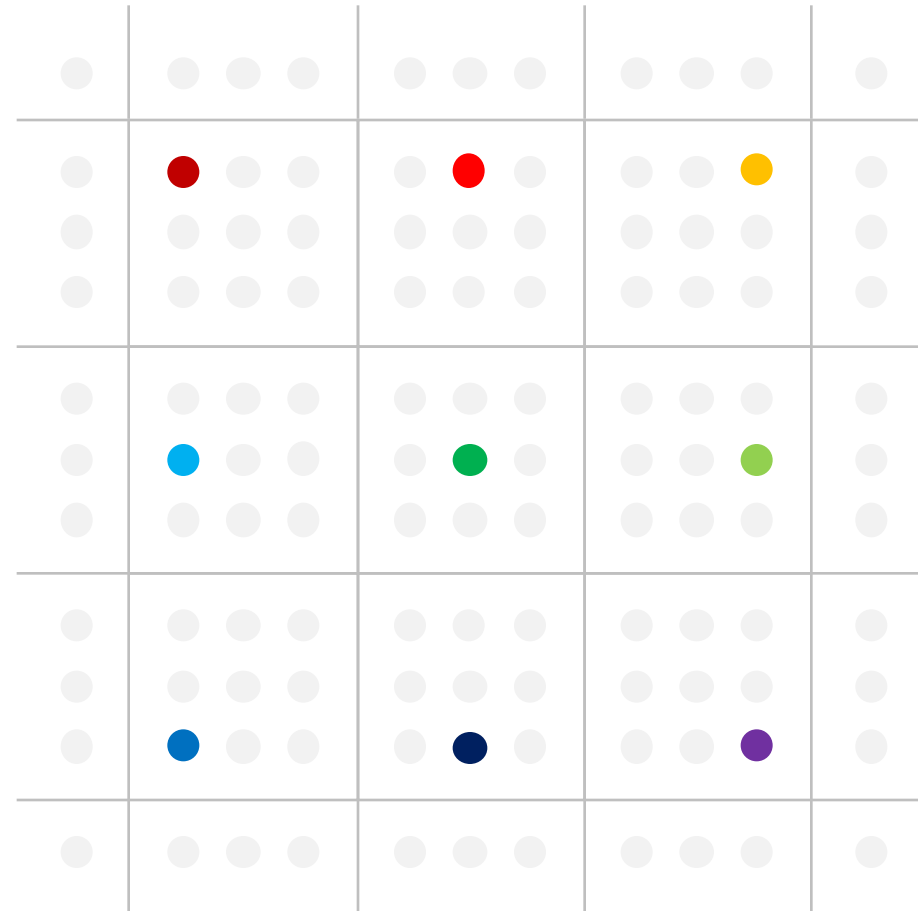
Die Lattice-Boltzmann-Methode

Beispiel: D2Q9 Methode

Update-Regel wie bei LGCA:

1. Stream

Jeder Eintrag wird in eine Nachbarzelle propagiert.



Die Lattice-Boltzmann-Methode

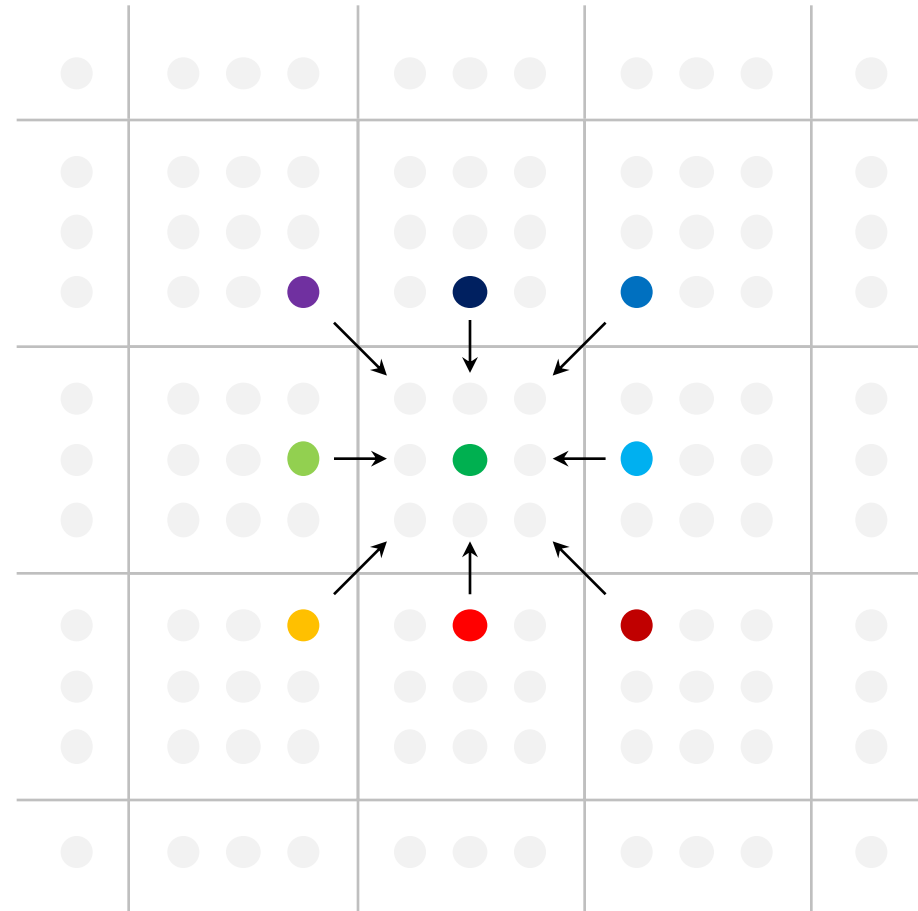
Beispiel: D2Q9 Methode

Update-Regel wie bei LGCA:

1. Stream

Jeder Eintrag wird in eine Nachbarzelle propagiert.

Im Austausch erhält die Zelle die Einträge der Nachbarzellen.



Die Lattice-Boltzmann-Methode

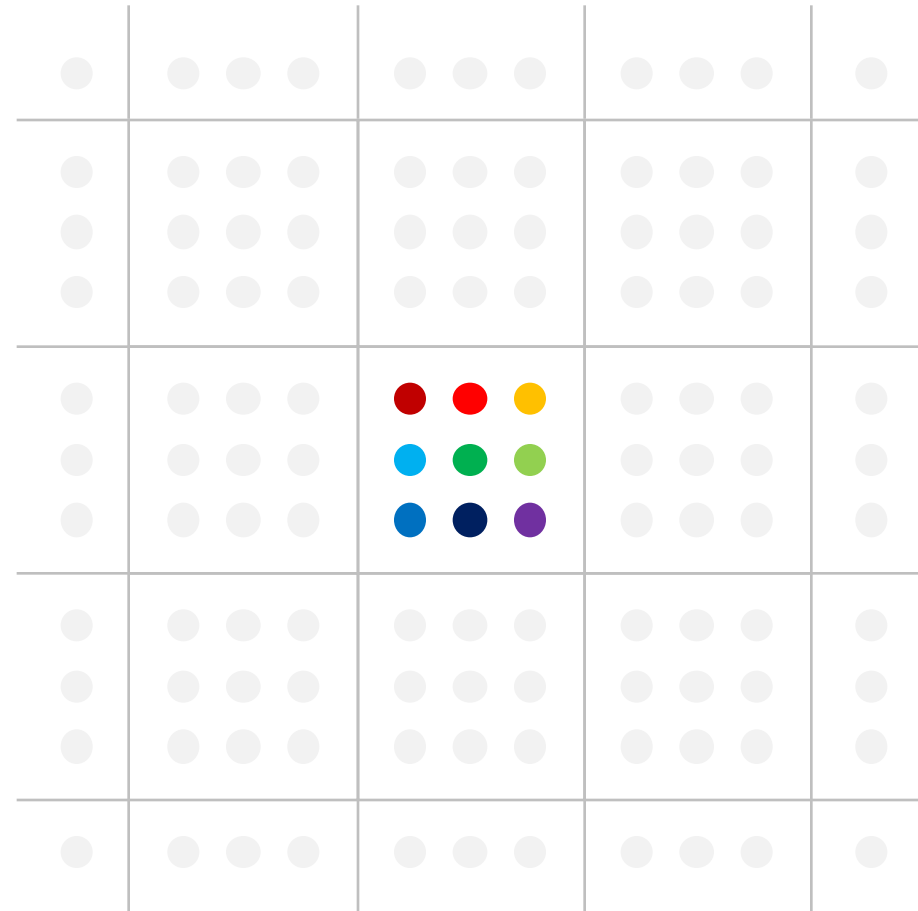
Beispiel: D2Q9 Methode

Update-Regel wie bei LGCA:

1. Stream

Jeder Eintrag wird in eine Nachbarzelle propagiert.

Im Austausch erhält die Zelle die Einträge der Nachbarzellen.



Die Lattice-Boltzmann-Methode

Beispiel: D2Q9 Methode

Update-Regel wie bei LGCA:

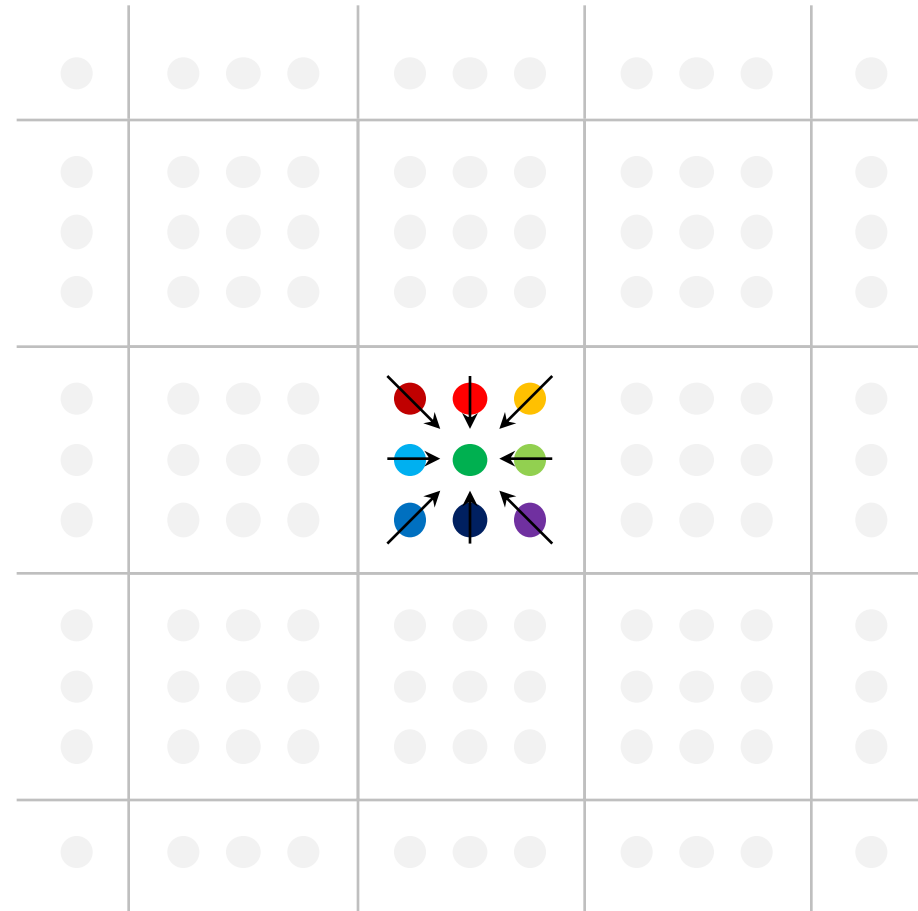
1. Stream

2. Collide

Einträge werden lokal ins Gleichgewicht gebracht

Hierfür gibt es zahlreiche Techniken

(SRT, TRT, MRT, ...)



Der Kollisionsoperator

- **Oft verwendet:** Der BGK-Operator (lineare Relaxation)

$$C_{BGK} = -\omega \left(f - f^{(eq)} \right)$$

Die Gleichgewichtsverteilung $f^{(eq)}$ ist dabei:

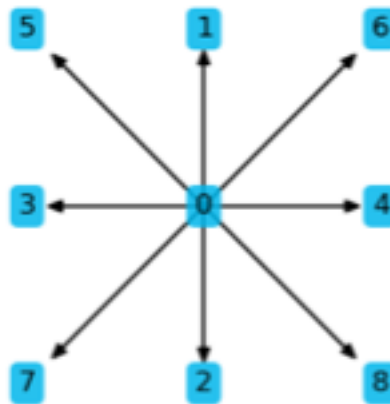
$$f^{(eq)}(\rho, \mathbf{u}) = \rho (2\pi c_s^2)^{-\frac{D}{2}} e^{-\frac{1}{2c_s^2} (\mathbf{v} - \mathbf{u})^2}$$

(näheres dazu auf dem nächsten Übungsblatt ...)

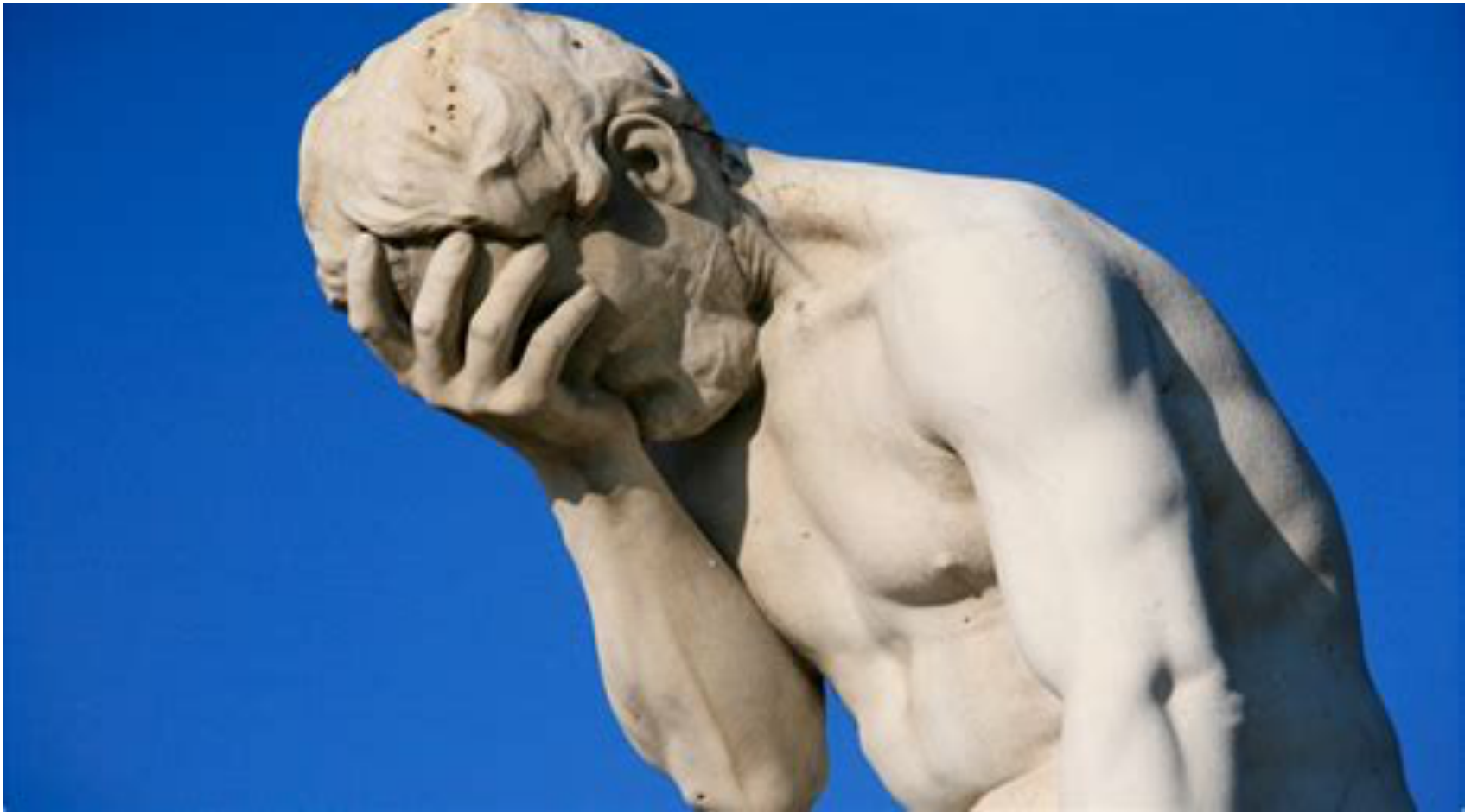
Berechnung makroskopischer Größen

$$\int f d\mathbf{v} = \rho = \sum_q f_q = f_0 + f_1 + f_2 + \cdots + f_8$$

$$\int v_x f d\mathbf{v} = \rho u_x = \sum_q c_{qx} f_q = f_6 + f_4 + f_8 - f_5 - f_3 - f_7$$



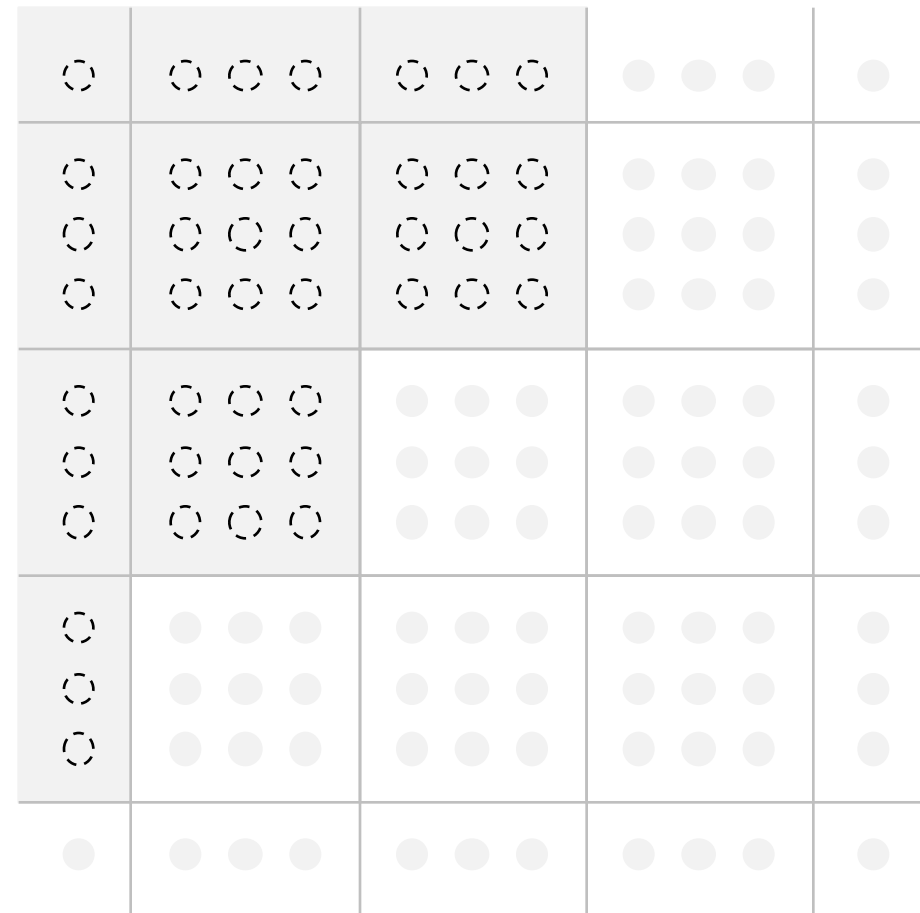
LBM Randbehandlung



LBM Randbehandlung

Beispiel: D2Q9 Methode

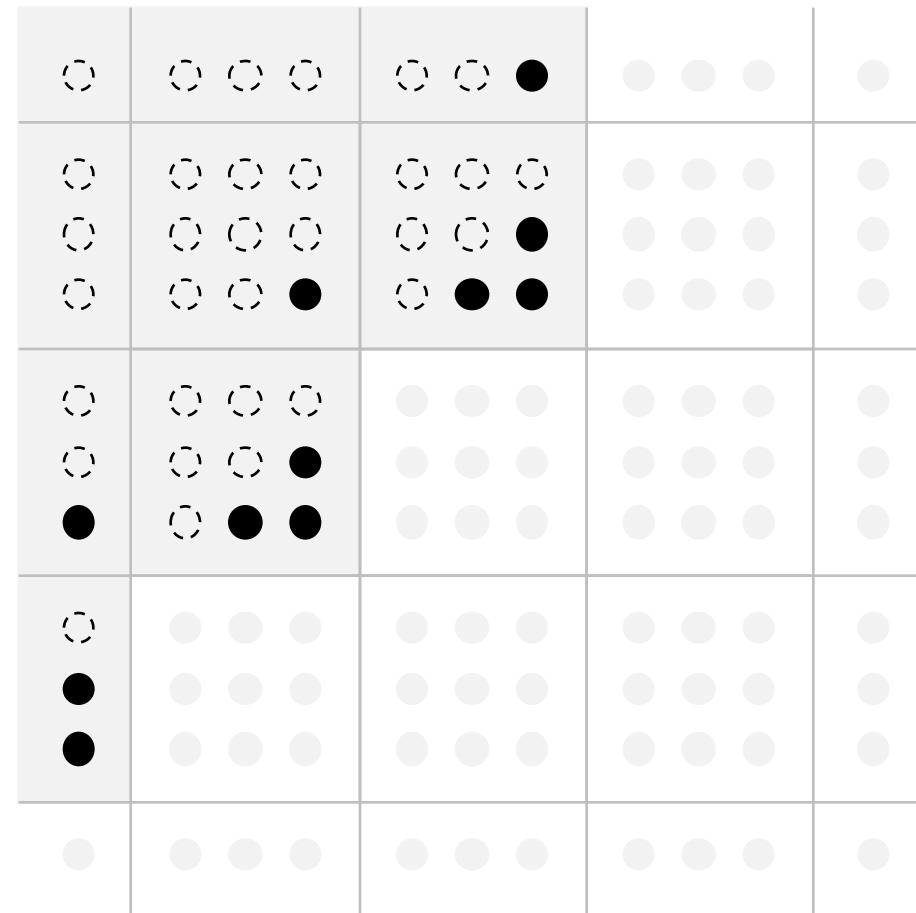
- Randbehandlung bei LBM ist einfach!



LBM Randbehandlung

Beispiel: D2Q9 Methode

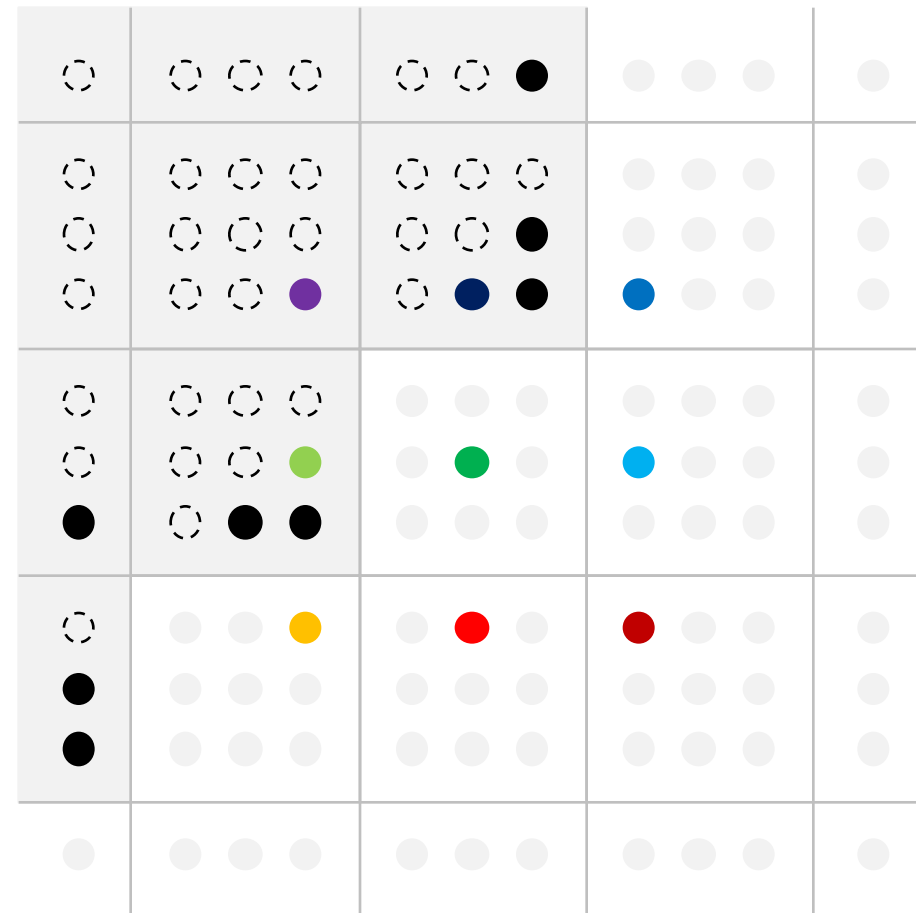
- Randbehandlung bei LBM ist einfach!
- Direkte Randzellen bekommen ebenfalls Verteilungsfunktionen.



LBM Randbehandlung

Beispiel: D2Q9 Methode

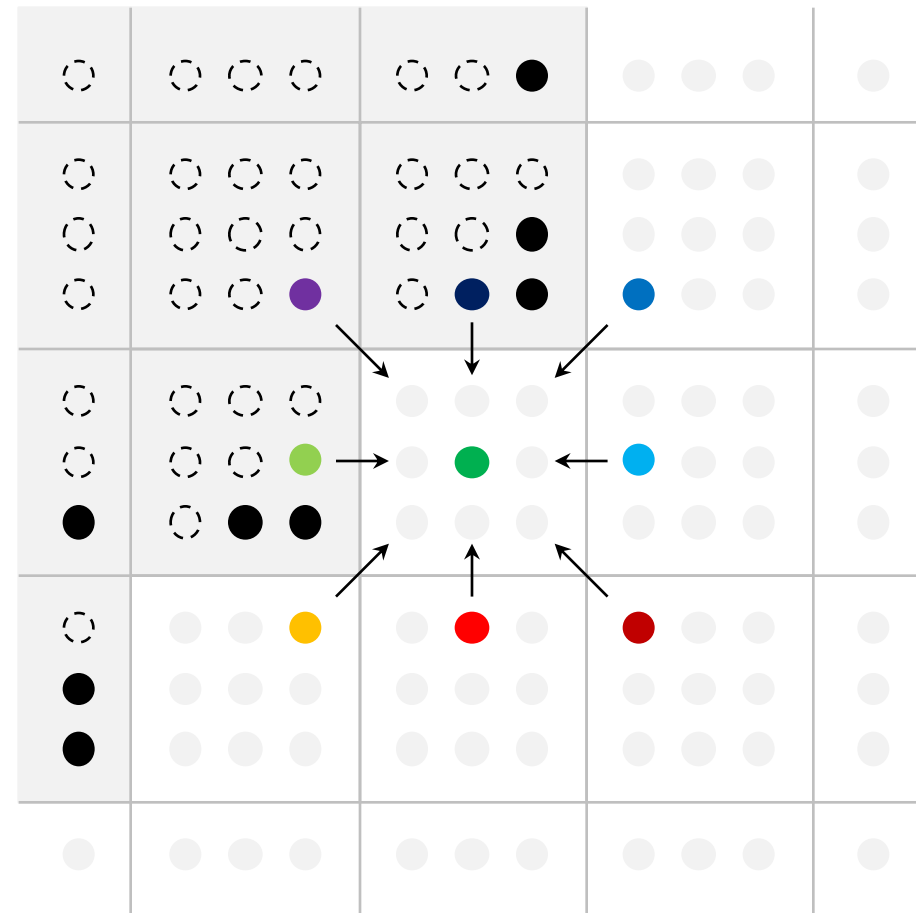
- Randbehandlung bei LBM ist einfach!
- Direkte Randzellen bekommen ebenfalls Verteilungsfunktionen (je nach Randtyp)



LBM Randbehandlung

Beispiel: D2Q9 Methode

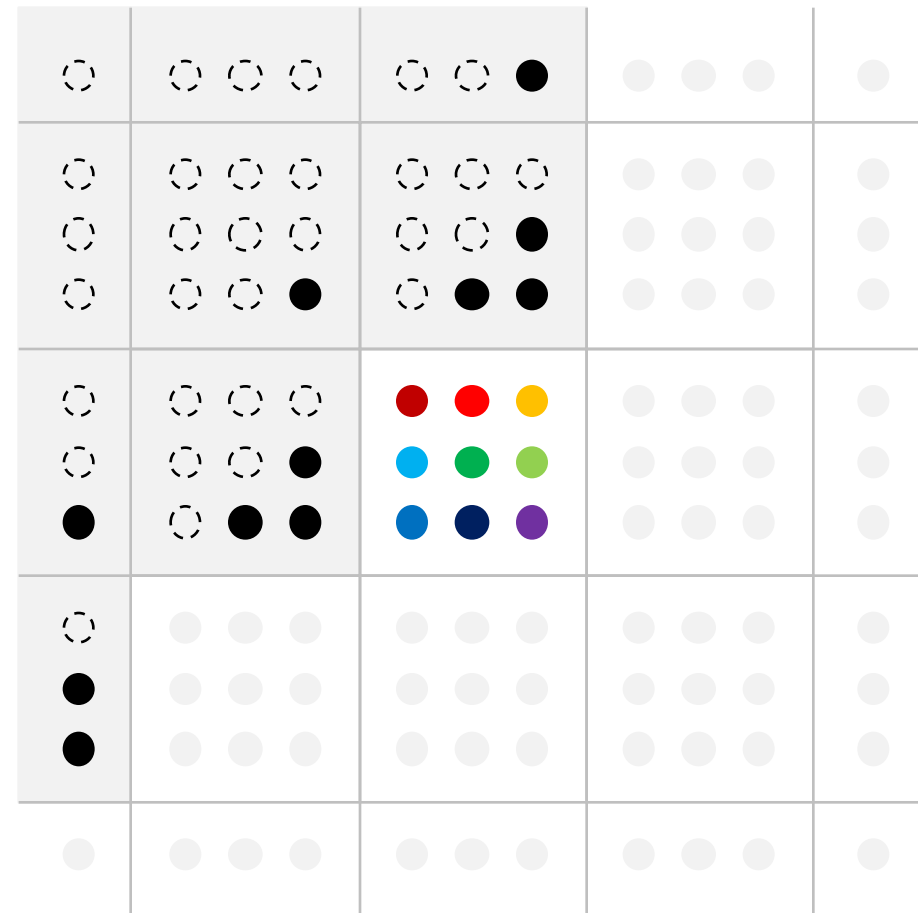
- Randbehandlung bei LBM ist einfach!
- Direkte Randzellen bekommen ebenfalls Verteilungsfunktionen (je nach Randtyp)
- Diese Werte werden bei **Stream** ebenfalls verteilt.



LBM Randbehandlung

Beispiel: D2Q9 Methode

- Randbehandlung bei LBM ist einfach!
- Direkte Randzellen bekommen ebenfalls Verteilungsfunktionen (je nach Randtyp)
- Diese Werte werden bei **Stream** ebenfalls verteilt.



LBM Randbehandlung - Implementierung

- Jede Zelle speichert (zusätzlich zu den Verteilungsfunktionen) ihren Typ
- Zulässige Typen sind: Raue Wand, Glatte Wand, Fluid, ...
- Vor dem **Stream**-Schritt werden alle Randpunkte speziell aktualisiert.

Randbehandlung ist bei LBM **deutlich** einfacher als z.B. beim Lösen der Navier-Stokes-Gleichungen.

Damit ist LBM eine attraktive Methode für Simulationen auf komplexen Geometrie (freie Oberflächen, Blutgefäße, ...)

LBM - Ausblick

Die gute Nachricht:

- Man kann zeigen, dass die Lattice-Boltzmann Methode die Navier-Stokes Gleichungen richtig löst.
(Stichwort: Chapman-Enskog Erweiterung)
- Wie so oft kommt die Lösung erst aus der Praxis und wird erst nachträglich theoretisch aufgearbeitet.

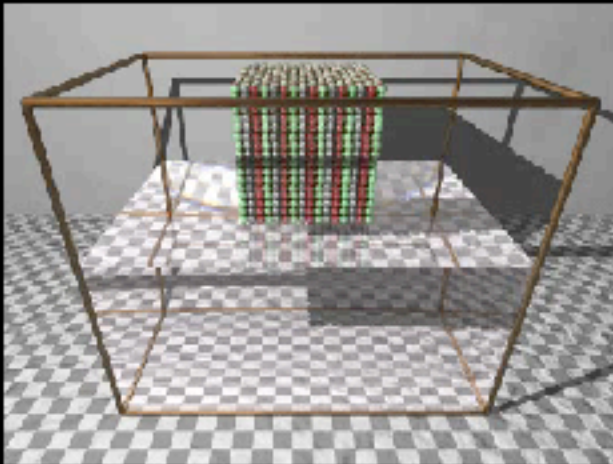
LBM - Ausblick

- Komplexere Zellen, 3 Dimensionen (D3Q19, D3Q27)
- Freie Oberflächen
- Kopplung von LBM und Partikelsimulation
- Temperatur und chemische Reaktionen
- Genauere Kollisionsmodelle (PyStencils, PyLBM)
- Lokale Verfeinerung
- Effiziente Implementierung, insbesondere Lastenverteilung auf Supercomputern.

Einige LSS-Videos zum Schluss...

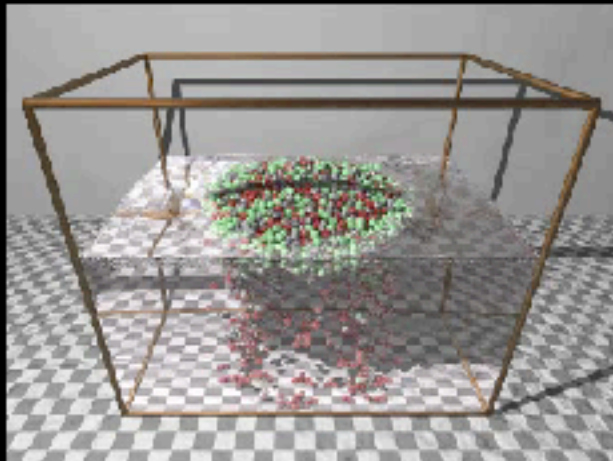


Department of Computer Science
Chair for System Simulation
University of Erlangen-Nürnberg



Simon Bogner, Ulrich Rüde

Particles in Free-Surface Flow



March 2011

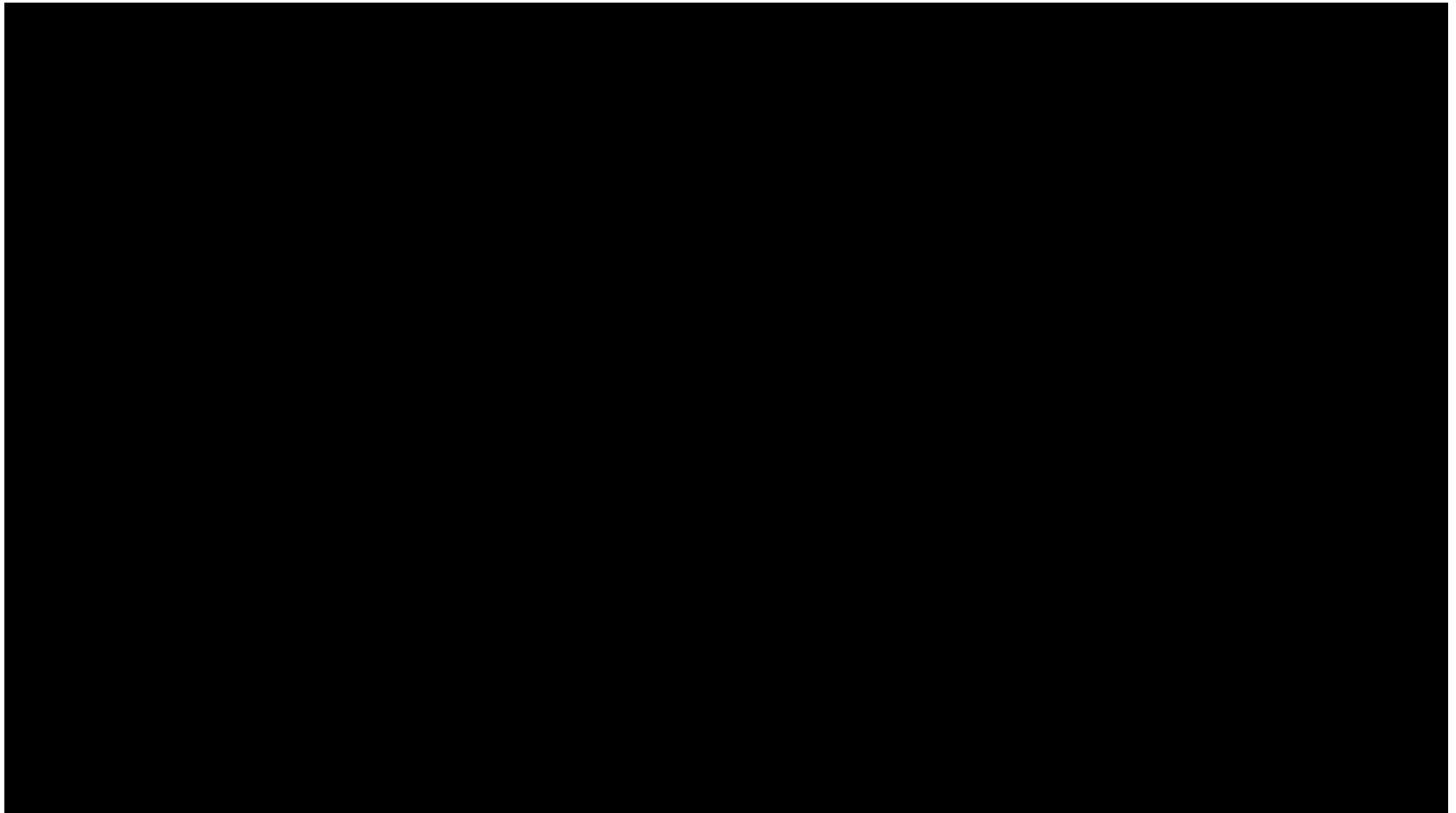
Blutfluss am Herzen

Sedimentablagerung in einem Fluss



Simulation von Stimmlippen

4300 Prozesse auf SuperMUC, 101.466.432 LBM Zellen



Vielen Dank für die Aufmerksamkeit!