

Analyse der Druckberechnung mithilfe einer Zustandsgleichung im Vergleich zur Lösung eines Gleichungssystems in SPH- Flüssigkeitssimulationen

Pascal Hunkler

May 2022

Inhaltsverzeichnis

1	Abstract	4
2	Einleitung	5
3	Grundlagen	6
3.1	Navier-Stokes-Gleichung und Flüssigkeitssimulationen	6
3.1.1	Partikelbasierte Simulation	6
3.1.2	Gitterbasierte Simulation	7
3.2	SPH	7
3.2.1	Diskretisierung mit SPH	7
3.2.2	SPH in partikelbasierten Simulationen	8
4	Druckberechnung	9
4.1	Druckberechnung mit einer Zustandsgleichung	9
4.2	Druckberechnung mit IISPH	9
5	Implementierung	10
5.1	Programmierungsumgebung	10
5.2	Architektur der Software	10
5.3	Kernelfunktion, Kernelgradient	10
5.4	Nachbarschaftssuche	10
5.4.1	Uniformes Gitter Aufbau	10
5.4.2	Bestimmung der Nachbarn mithilfe des uniformen Gitters	10
5.5	Simulationsschritt	10
5.5.1	Berechnung der Dichte	10
5.5.2	Berechnung des Drucks	10
5.5.3	Berechnung der Druckbeschleunigung	10
5.5.4	Berechnung der restlichen Beschleunigungen	10
5.6	Visualisierung	10
6	Analyse	11
6.1	Szenarien	11
6.2	Rechen- und Speicheraufwand	11
6.3	Einfluss des Zeitschritts	11
7	Fazit und Ausblick	12

1 Abstract

2 Einleitung

3 Grundlagen

3.1 Navier-Stokes-Gleichung und Flüssigkeitssimulationen

Dieser Abschnitt behandelt die Navier-Stokes-Gleichung, welche die Grundlage dafür bildet, Flüssigkeitssimulationen zu realisieren. Sie beschreibt die Änderungsrate der Geschwindigkeit eines kleinen volumetrischen Flüssigkeitselements. Für die Navier-Stokes-Gleichung gibt es verschiedene Formen, die auf den eulerschen Ansatz oder den lagrangeschen Ansatz beruhen. Diese Arbeit beschäftigt sich im Wesentlichen auf lagrangesche, partikelbasierte Simulationen, der Vollständigkeit wegen wird jedoch auch noch kurz auf eulersche, gitterbasierte Ansätze eingegangen. Die Inhalte dieses Abschnitts beziehen sich auf Ihmsen et al. [IOS⁺14].

3.1.1 Partikelbasierte Simulation

In einer partikelbasierten Flüssigkeitssimulation wird die Flüssigkeit in so genannte Partikel unterteilt, die sich mit der Flüssigkeit mitbewegen. Jedes Partikel nimmt ein Teilvolumen der Flüssigkeit ein und besitzt eine Masse. In einem partikelbasierten Ansatz kann die Navier-Stokes-Gleichung wie folgt beschrieben werden:

$$\frac{d\mathbf{v}_i}{dt} = -\frac{1}{\rho_i}\nabla p_i + \nu\nabla^2\mathbf{v}_i + \frac{\mathbf{F}_i^{\text{other}}}{m_i} \quad (3.1)$$

Sie zeigt die verschiedenen Beschleunigungen, die auf ein Partikel i wirken, und damit die Geschwindigkeit des Partikels beeinflussen: Die Druckbeschleunigung $-\frac{1}{\rho_i}\nabla p_i$, die Viskositätsbeschleunigung $\nu\nabla^2\mathbf{v}_i$ und der Einfluss anderer Kräfte $\frac{\mathbf{F}_i^{\text{other}}}{m_i}$, wie beispielsweise die Gravitation. Ziel einer partikelbasierten Simulation ist es, die verschiedenen Beschleunigungen zu berechnen, um die Geschwindigkeiten und Positionen der Partikel stets anzupassen. Eine Methode, um die Beschleunigungen an den Partikeln zu berechnen ist Smoothed Particle Hydrodynamics (SPH), welches im nächsten Abschnitt vorgestellt wird.

3.1.2 Gitterbasierte Simulation

3.2 SPH

Das Konzept Smoothed Particle Hydrodynamics (SPH), ursprünglich formuliert von Lucy [Luc77] und unabhängig davon von Monaghan und Gingold [GM77], entstand ursprünglich aus dem Bereich der Astrophysik, wird aber heute auch in diversen anderen Bereichen, unter anderem auch der Computergrafik angewandt. Mithilfe von SPH kann durch Diskretisierung und Berechnung von Größen wie der Druckbeschleunigung oder der Viskositätsbeschleunigung die Navier-Stokes-Gleichung gelöst werden. Es eignet sich daher gut für partikelbasierte Simulationen.

3.2.1 Diskretisierung mit SPH

Die Inhalte dieses Abschnittes basieren hauptsächlich auf den Arbeiten von Monaghan [Mon05], von Price [Pri12] und von Koshier et al. [KBST20]. Für eine beliebige skalare Variable A gilt die Identität

$$A(\mathbf{x}) = \int A(\mathbf{x}') \delta(\mathbf{x} - \mathbf{x}') d\mathbf{x}' \quad (3.2)$$

δ ist hierbei die Dirac'sche Deltafunktion, die definiert ist als

$$\delta(\mathbf{x}) = \begin{cases} \infty, & \text{falls } \mathbf{x} = 0 \\ 0, & \text{sonst} \end{cases} \quad (3.3)$$

Die Dirac'sche Deltafunktion in Gleichung 3.2 kann mithilfe einer glättenden Kernelfunktion W mit endlicher Breite h approximiert werden.

$$A(\mathbf{x}) = \int A(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' + O(h^2) \quad (3.4)$$

Damit die Approximation aus Gleichung 3.4 gültig ist, muss W folgende Eigenschaften besitzen:

$$\int_{\mathbb{R}^d} W(\mathbf{x}', h) dv' = 1 \quad (\text{Normalisierung}) \quad (3.5)$$

$$\lim_{h \rightarrow 0} W(\mathbf{x}, h) = \delta(\mathbf{x}) \quad (\text{Dirac-}\delta) \quad (3.6)$$

Weitere wünschenswerte Eigenschaften der Kernelfunktion W sind:

$$W(\mathbf{x}, h) \geq 0 \quad (\text{Positivität}) \quad (3.7)$$

$$W(\mathbf{x}, h) = W(-\mathbf{x}, h) \quad (\text{Symmetrie}) \quad (3.8)$$

$$W(\mathbf{x}, h) = 0 \text{ für } \|\mathbf{x}\| \geq h \quad (\text{Kompakter Support}) \quad (3.9)$$

$\forall \mathbf{x} \in \mathbb{R}^d, h \in \mathbb{R}^+$, während h der Kernsupport ist. Eine beliebte Kernelfunktion ist der Cubic Spline Kernel [Mon92]. Er ist definiert als

$$W(\mathbf{x}, h) = \sigma_d \begin{cases} (2-q)^3 - 4(1-q)^3, & \text{für } 0 \leq q \leq 1 \\ (2-q)^3, & \text{für } 1 \leq q \leq 2 \\ 0, & \text{sonst} \end{cases} \quad (3.10)$$

mit $q = \frac{1}{h} \|\mathbf{x}\|$. Der Kernnormalisierungsfaktor σ_d ist abhängig von der Dimension d und beträgt für $d = 1, 2, 3$ $\sigma_1 = \frac{1}{6h}$, $\sigma_2 = \frac{5}{14\pi h^2}$, $\sigma_3 = \frac{1}{4\pi h^3}$. In der Literatur gibt es verschiedene Formulierungen für den Cubic Spline Kernel, die sich im Wesentlichen in der Parametrisierung unterscheiden. Der Vorteil dieser Kernelfunktion ist, dass die Eigenschaften zu Positivität, Symmetrie, und kompakten Support, erfüllt werden. Zudem erzielt Cubic Spline trotz seiner Einfachkeit gute Ergebnisse.

Um die Interpolation aus Gleichung 3.4 bei einer Flüssigkeit zu diskretisieren, wird die Flüssigkeit in mehrere Partikel unterteilt. Jedes Partikel f besitzt eine Masse m_f , Dichte ρ_f und Position \mathbf{x}_f . Der Wert von A an einem Partikel f wird notiert als A_f . Der Integral aus Gleichung 3.4 kann nun durch eine Summe, und die Masse ρdV durch die Partikelmasse m_f ersetzt werden.

$$A(\mathbf{x}) = \int \frac{A(\mathbf{x}')}{\rho(\mathbf{x}')} W(\mathbf{x} - \mathbf{x}', h) \rho(\mathbf{x}') d\mathbf{x}' \quad (3.11)$$

$$\approx \sum_{f_f} m_{f_f} \frac{A_{f_f}}{\rho_{f_f}} W(\mathbf{x}_f - \mathbf{x}_{f_f}, h) = A_f \quad (3.12)$$

f_f sind hierbei alle Partikel der Flüssigkeit. Da beispielsweise der Cubic Spline Kernel einen Kernsupport von $2h$ besitzt, muss hier nur über Partikel f_f summiert werden, die in direkter Nachbarschaft sind, da die Kernelfunktion für Partikel, die weiter als $2h$ von dem Partikel f entfernt sind, null ist. Daher ist die Eigenschaft, dass der Support der Kernelfunktion kompakt ist, wünschenswert.

3.2.2 SPH in partikelbasierten Simulationen

4 Druckberechnung

4.1 Druckberechnung mit einer Zustandsgleichung

4.2 Druckberechnung mit IISPH

5 Implementierung

5.1 Programmierumgebung

5.2 Architektur der Software

5.3 Kernelfunktion, Kernelgradient

5.4 Nachbarschaftssuche

5.4.1 Uniformes Gitter Aufbau

5.4.2 Bestimmung der Nachbarn mithilfe des uniformen Gitters

5.4.3

5.5 Simulationsschritt

5.5.1 Berechnung der Dichte

5.5.2 Berechnung des Drucks

5.5.3 Berechnung der Druckbeschleunigung

5.5.4 Berechnung der restlichen Beschleunigungen

5.6 Visualisierung

6 Analyse

6.1 Szenarien

6.2 Rechen- und Speicheraufwand

6.3 Einfluss des Zeitschritts

6.4

7 Fazit und Ausblick

8 Literaturverzeichnis

Literaturverzeichnis

- [GM77] Robert A. Gingold and Joseph J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, 181(3):375–389, 1977. ISBN: 1365-2966 Publisher: Oxford University Press Oxford, UK.
- [IOS⁺14] Markus Ihmsen, Jens Orthmann, Barbara Solenthaler, Andreas Kolb, and Matthias Teschner. SPH fluids in computer graphics. In *EUROGRAPHICS 2014/S. LEFEBVRE AND M. SPAGNUOLO*. Citeseer, 2014.
- [KBST20] Dan Koschier, Jan Bender, Barbara Solenthaler, and Matthias Teschner. Smoothed particle hydrodynamics techniques for the physics based simulation of fluids and solids. *arXiv preprint arXiv:2009.06944*, 2020.
- [Luc77] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82:1013, December 1977.
- [Mon92] Joe J. Monaghan. Smoothed particle hydrodynamics. *Annual review of astronomy and astrophysics*, 30:543–574, 1992. ISBN: 0066-4146.
- [Mon05] Joe J. Monaghan. Smoothed particle hydrodynamics. *Reports on progress in physics*, 68(8):1703, 2005. ISBN: 0034-4885 Publisher: IOP Publishing.
- [Pri12] Daniel J. Price. Smoothed particle hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics*, 231(3):759–794, February 2012.

Algorithm 1 Simulationsschritt

```
1: Determine neighbors of each particle
2: Compute density  $\rho_f$  of each fluid particle using algorithm 2
3: Compute non-pressure accelerations  $\mathbf{a}_f^n$  using algorithm 3
4: for all fluid particle f do
5:    $\mathbf{v}_f^* \leftarrow \mathbf{v}_f + \Delta t \mathbf{a}_f^n$ 
6: end for
7: Compute pressure  $p_f$  of each fluid particle using algorithm 5
8: Compute pressure accelerations  $\mathbf{a}_f^p$  using algorithm 4
9: for all fluid particle f do
10:   $\mathbf{v}_f \leftarrow \mathbf{v}_f^* + \Delta t \mathbf{a}_f^p$ 
11: end for
12: for all fluid particle f do
13:   $\mathbf{x}_f \leftarrow \mathbf{x}_f + \Delta t \mathbf{v}_f$ 
14: end for
15:
```

Algorithm 2 Berechnung der Dichte der Partikel

```
for all particle i do
  if particle i belongs to the boundary then
    continue
  end if
   $\rho_i \leftarrow 0$ 
  for all neighbor j of particle i do
     $\rho_i \leftarrow \rho_i + W_{ij}$ 
  end for
   $\rho_i \leftarrow \rho_i \cdot m_f$ 
end for
```

Algorithm 3 Berechnung der restlichen Beschleunigungen

```
for all particle i do
  if particle i belongs to the boundary then
     $\mathbf{a}_i^n \leftarrow (0 \ 0)^\top$ 
    continue
  end if
   $\mathbf{acc}_g \leftarrow (0 \ -9.81)^\top$ 
   $\mathbf{acc}_v \leftarrow (0 \ 0)^\top$ 

  // Viskositätsbeschleunigung an Partikel i
  for all neighbor j of particle i do
    if particle j belongs to the boundary then
       $\mathbf{acc}_v \leftarrow \mathbf{acc}_v + \frac{1}{\rho_i} \frac{(\mathbf{v}_i - \mathbf{v}_j)(\mathbf{x}_i - \mathbf{x}_j)}{(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j) + 0.01h^2} \cdot \nabla W_{ij}$ 
    else
       $\mathbf{acc}_v \leftarrow \mathbf{acc}_v + \frac{1}{\rho_j} \frac{(\mathbf{v}_i - \mathbf{v}_j)(\mathbf{x}_i - \mathbf{x}_j)}{(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j) + 0.01h^2} \cdot \nabla W_{ij}$ 
    end if
  end for
   $\mathbf{acc}_v \leftarrow 2\nu m_f \cdot \mathbf{acc}_v$ 

   $\mathbf{a}_i^n \leftarrow \mathbf{acc}_g + \mathbf{acc}_v$ 
end for
```

Algorithm 4 Berechnung der Druckbeschleunigungen

```
for all particle i do
  if particle i belongs to the boundary then
     $\mathbf{a}_i^p \leftarrow (0 \ 0)^\top$ 
    continue
  end if
   $\mathbf{acc}_p \leftarrow (0 \ 0)^\top$ 

  // Druckbeschleunigung an Partikel i
  for all neighbor j of particle i do
    if particle j belongs to the boundary then
       $\mathbf{acc}_p \leftarrow \mathbf{acc}_p - \left( \frac{p_i}{\rho_i^2} + \frac{p_j}{(\rho_f^0)^2} \right) \cdot \nabla W_{ij}$ 
    else
       $\mathbf{acc}_p \leftarrow \mathbf{acc}_p - \left( \frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \cdot \nabla W_{ij}$ 
    end if
  end for
   $\mathbf{acc}_p \leftarrow \mathbf{acc}_p \cdot m_f$ 

   $\mathbf{a}_i^p \leftarrow \mathbf{acc}_p$ 
end for
```

Algorithm 5 Berechnung des Drucks der Partikel

```
1: for all fluid particle  $f$  do
2:    $A_{ff} \leftarrow -\Delta t^2 \frac{m_f^2}{\rho_f^2} \cdot \left( \sum_{f_f} \nabla W_{ff_f} \nabla W_{ff_f} + \sum_{f_b} \nabla W_{ff_b} \nabla W_{ff_b} \right)$ 
3:    $s_f \leftarrow \rho_f^0 - \rho_f - m_f \Delta t \left( \sum_{f_f} (\mathbf{v}_f^* - \mathbf{v}_{f_f}^*) \nabla W_{ff_f} + \sum_{f_b} \mathbf{v}_f^* \nabla W_{ff_b} \right)$ 
4:    $p_f \leftarrow 0$ 
5: end for
6:  $e \leftarrow \infty$ 
7: while  $e \geq 0.001$  do
8:    $e \leftarrow 0$ 
9:   Compute pressure accelerations  $\mathbf{a}_f^p$  using algorithm 4
10:  for all fluid particle  $f$  do
11:     $(\mathbf{A}p)_f \leftarrow m_f \Delta t^2 \left( \sum_{f_f} (\mathbf{a}_f^p - \mathbf{a}_{f_f}^p) \nabla W_{ff_f} + \sum_{f_b} \mathbf{a}_f^p \nabla W_{ff_b} \right)$ 
12:    if  $f$  has no neighbors then
13:       $(\mathbf{A}p)_f \leftarrow 0$ 
14:    end if
15:     $p_f \leftarrow \max(p_f + \omega \frac{s_f - (\mathbf{A}p)_f}{\mathbf{A}_{ff}}, 0)$ 
16:     $e \leftarrow e + \frac{(\mathbf{A}p)_f - s_f}{\rho_f^0}$ 
17:  end for
18:   $e \leftarrow \frac{e}{n}$ 
19: end while
```
