

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	9
4	Druckberechnung	11
4.1	Druckberechnung mit einer Zustandsgleichung	11
4.2	Druckberechnung mit IISPH	11
5	Implementierung	12
5.1	Programmierungsumgebung	12
5.2	Architektur der Software	12
5.3	Kernelfunktion, Kernelgradient	12
5.4	Nachbarschaftssuche	12
5.4.1	Uniformes Gitter Aufbau	12
5.4.2	Bestimmung der Nachbarn mithilfe des uniformen Gitters	12
5.5	Simulationsschritt	12
5.5.1	Berechnung der Dichte	12
5.5.2	Berechnung des Drucks	12
5.5.3	Berechnung der Druckbeschleunigung	12
5.5.4	Berechnung der restlichen Beschleunigungen	12
5.6	Visualisierung	12
6	Analyse	13
6.1	Szenarien	13
6.2	Rechen- und Speicheraufwand	13
6.3	Einfluss des Zeitschritts	13
7	Fazit und Ausblick	14

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 lagrangeschen, partikelbasierten Flüssigkeitssimulation wird die Flüssigkeit in so genannte Partikel unterteilt, die sich mit der Flüssigkeit mitbewegen. Jedes Partikel nimmt ein Teilvervolumen 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 gibt die Änderungsrate der Geschwindigkeit $\frac{d\mathbf{v}_i}{dt}$ eines beweglichen Partikels mit Positionen \mathbf{x}_i nach Zeit an, welche durch verschiedene Beschleunigungen beeinflusst wird: Die Druckbeschleunigung $-\frac{1}{\rho_i}\nabla p_i$, die Viskositätsbeschleunigung $\nu\nabla^2\mathbf{v}_i$, die durch die Viskosität ν beeinflusst wird, 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. In partikelbasierten Simulationen muss regelmäßig die Nachbarschaft eines Partikels bestimmt werden, da diese frei beweglich sind und sich somit die Nachbarschaft ändert. Die Nachbarschaft eines Partikels wird benötigt, um die Beschleunigungen an einem Partikel mit SPH zu bestimmen. Im Gegensatz zu gitterbasierten Simulationen,

welche im nächsten Unterabschnitt vorgestellt werden, muss hier jedoch ein Term aus der Navier-Stokes-Gleichung weniger berechnet werden.

3.1.2 Gitterbasierte Simulation

In eulerschen, gitterbasierten Simulationen wird im Gegensatz zu partikelbasierten Simulationen mit Gittern statt Partikeln gearbeitet. Hier wird die Flüssigkeit nicht in frei bewegliche Partikel unterteilt, sondern in Teilvolumina, die fest an einem Ort sind und meist ein Gitter bilden. Die Navier-Stokes-Gleichung hat durch diesen anderen Ansatz folgende Form:

$$\frac{\delta \mathbf{v}_i}{\delta t} = -\frac{1}{\rho_i} \nabla p_i + \nu \nabla^2 \mathbf{v}_i + \frac{\mathbf{F}_i^{\text{other}}}{m_i} - \mathbf{v}_i \cdot \nabla \mathbf{v}_i \quad (3.2)$$

Betrachtet wird hier die Änderungsrate der Geschwindigkeit $\frac{\delta \mathbf{v}_i}{\delta t}$ an einer festen Position \mathbf{x}_i . Dadurch, dass die Positionen fest sind, kommt in der Navier-Stokes-Gleichung neben den Beschleunigungen noch der Term $-\mathbf{v}_i \cdot \nabla \mathbf{v}_i$ vor, welcher die Bewegung der Flüssigkeit ausgleicht. Im Gegensatz zu partikelbasierten Simulationen muss hier jedoch direkt erst einmal keine Nachbarschaftssuche stattfinden, da die Punkte nicht frei beweglich sind. Generell kann jedoch die Performance und Genauigkeit zwischen partikelbasierten und gitterbasierten nicht allgemein verglichen werden.

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

$$A(\mathbf{x}) = \int A(\mathbf{x}') \delta(\mathbf{x} - \mathbf{x}') d\mathbf{x}' \quad (3.3)$$

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

$$\delta(\mathbf{x}) = \begin{cases} \infty, & \text{falls } \mathbf{x} = 0 \\ 0, & \text{sonst} \end{cases} \quad (3.4)$$

Die Dirac'sche Deltafunktion in Gleichung 3.3 kann mithilfe einer glättenden Kernelfunktion W mit endlicher Breite h approximiert werden.

$$A(\mathbf{x}) = \int A(\mathbf{x}') W(\mathbf{x} - \mathbf{x}', h) d\mathbf{x}' + O(h^2) \quad (3.5)$$

Damit die Approximation aus Gleichung 3.5 gültig ist, muss W folgende Eigenschaften besitzen:

$$\int_{\mathbb{R}^d} W(\mathbf{x}', h) d\mathbf{x}' = 1 \quad (\text{Normalisierung}) \quad (3.6)$$

$$\lim_{h \rightarrow 0} W(\mathbf{x}, h) = \delta(\mathbf{x}) \quad (\text{Dirac-}\delta) \quad (3.7)$$

Weitere wünschenswerte Eigenschaften der Kernelfunktion W sind:

$$W(\mathbf{x}, h) \geq 0 \quad (\text{Positivität}) \quad (3.8)$$

$$W(\mathbf{x}, h) = W(-\mathbf{x}, h) \quad (\text{Symmetrie}) \quad (3.9)$$

$$W(\mathbf{x}, h) = 0 \text{ für } \|\mathbf{x}\| \geq h \quad (\text{Kompakter Support}) \quad (3.10)$$

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

$$W(\mathbf{x}, h) = \sigma_d \begin{cases} (2 - q)^3 - 4(1 - q)^3, & \text{für } 0 \leq q \leq 1 \\ (2 - q)^3, & \text{für } 1 \leq q \leq 2 \\ 0, & \text{sonst} \end{cases} \quad (3.11)$$

mit $q = \frac{1}{h} \|\mathbf{x}\|$. Der Kernelnormierungsfaktor 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 Einfachheit gute Ergebnisse.

Um die Interpolation aus Gleichung 3.5 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 , die Approximation davon als $\langle A_f \rangle$. Der Integral aus Gleichung 3.5 kann nun durch eine Summe, und die Masse ρdV durch die Partikelmasse m_f ersetzt werden.

$$A(\mathbf{x}) = \int \frac{A(\mathbf{x}')}{\rho(\mathbf{x}')} W(\mathbf{x} - \mathbf{x}', h) \rho(\mathbf{x}') d\mathbf{x}' + O(h^2) \quad (3.12)$$

$$\approx \sum_{f_f} m_{f_f} \frac{A_{f_f}}{\rho_{f_f}} W(\mathbf{x} - \mathbf{x}_{f_f}, h) =: \langle A_f \rangle \quad (3.13)$$

f_f sind hierbei alle Partikel der Flüssigkeit. $W(\mathbf{x}_f - \mathbf{x}_{f_f}, h)$ wird ab nun durch W_{ff_f} abgekürzt. 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 Kernfunktion für Partikel, die weiter als $2h$ von dem Partikel f entfernt sind, null ist. Daher ist die Eigenschaft, dass der Support der Kernfunktion kompakt ist, wünschenswert.

Häufig ist es auch nötig, die Differentialoperatoren zu diskretisieren. Der Gradient einer Variable A kann durch folgende Approximation bestimmt werden:

$$\nabla A_f \approx \sum_{f_f} A_{f_f} \frac{m_{f_f}}{\rho_{f_f}} \nabla W_{ff_f} \quad (3.14)$$

Diese Approximation hat eine Genauigkeit 0-ter Ordnung, wenn folgende Bedingung erfüllt wird:

$$\sum_{f_f} \frac{m_{f_f}}{\rho_{f_f}} \nabla W_{ff_f} = \mathbf{0} \quad (3.15)$$

Wird zusätzlich die Bedingung

$$\sum_{f_f} \frac{m_{f_f}}{\rho_{f_f}} (\mathbf{x}_{f_f} - \mathbf{x}_f) \otimes \nabla W_{ff_f} = \mathbb{K} \quad (3.16)$$

erfüllt, hat sie eine Genauigkeit erster Ordnung. Ist \mathbf{A} höherdimensional, so können komplexere Differentialoperatoren wie folgt diskretisiert werden:

$$\nabla \mathbf{A}_f \approx \sum_{f_f} \frac{m_{f_f}}{\rho_{f_f}} \mathbf{A}_{f_f} \otimes \nabla W_{ff_f} \quad (3.17)$$

$$\nabla \cdot \mathbf{A}_f \approx \sum_{f_f} \frac{m_{f_f}}{\rho_{f_f}} \mathbf{A}_{f_f} \cdot \nabla W_{ff_f} \quad (3.18)$$

$$\nabla \times \mathbf{A}_f \approx - \sum_{f_f} \frac{m_{f_f}}{\rho_{f_f}} \mathbf{A}_{f_f} \times \nabla W_{ff_f} \quad (3.19)$$

wobei $a \otimes b = ab^T$ das dyadische Produkt ist. Leider führen diese „direkten“ Ableitungen zu einer schlechten Approximationsqualität und instabilen Simulationen. Daher gibt es mittlerweile viele alternative Formulierungen für diese Differentialoperatoren. Zwei davon werden im nächsten Unterabschnitt gezeigt.

3.2.2 SPH in partikelbasierten Simulationen

Ziel einer partikelbasierten Simulation ist es, die Beschleunigungen zu bestimmen, die in der Navier-Stokes-Gleichung angegeben sind. Die Navier-Stokes-Gleichung für einen partikelbasierten Ansatz hat folgende Form:

$$\frac{d\mathbf{v}_i}{dt} = -\frac{1}{\rho_i} \nabla p_i + \nu \nabla^2 \mathbf{v}_i + \frac{\mathbf{F}_i^{\text{other}}}{m_i} \quad (3.20)$$

Es muss also in erster Linie die Druckbeschleunigung $-\frac{1}{\rho_i}\nabla p_i$ und die Viskositätsbeschleunigung $\nu\nabla^2\mathbf{v}_i$ bestimmt werden. Die SPH Diskretisierung einer Variable A sieht wie folgt aus:

$$A_f \approx \sum_{ff} m_{ff} \frac{A_{ff}}{\rho_{ff}} W_{fff} \quad (3.21)$$

Hieraus ist ersichtlich, dass zur Berechnung von A die Dichte ρ_f aller Partikel benötigt wird. Diese kann jedoch leicht ermittelt werden, indem in Gleichung 3.21 A durch ρ ersetzt wird. Der Faktor $\frac{\rho_{ff}}{\rho_{ff}}$ kürzt sich weg und man erhält:

$$\rho_f \approx \sum_{ff} m_{ff} \frac{\rho_{ff}}{\rho_{ff}} W_{fff} = \sum_{ff} m_{ff} W_{fff} \quad (3.22)$$

Für die Berechnung der Druckbeschleunigung $-\frac{1}{\rho_i}\nabla p_i$ wird die Diskretisierung des Druckgradienten benötigt. Wie der Druck berechnet werden kann, wird im nächsten Kapitel vorgestellt. Ist der Druck berechnet, ist es naheliegend, die normale SPH Diskretisierung für den Gradienten zu verwenden:

$$-\frac{1}{\rho_i}\nabla p_f \approx -\frac{1}{\rho_i} \sum_{ff} A_{ff} \frac{m_{ff}}{\rho_{ff}} \nabla W_{fff} \quad (3.23)$$

Das Problem bei dieser Näherung ist jedoch, dass die daraus resultierende Beschleunigung nicht den Impuls und Drehimpuls der Flüssigkeit erhält, was zu einer instabilen Simulation führt. Für die Druckberechnung wird häufig stattdessen eine alternative Approximation verwendet, die auf folgender Diskretisierung des Gradienten einer Variable A beruht:

$$\nabla A_f \approx \rho_f \sum_{ff} m_{ff} \left(\frac{A_f}{\rho_f^2} + \frac{A_{ff}}{\rho_{ff}^2} \right) \nabla W_{fff} \quad (3.24)$$

Der Vorteil an dieser Approximation ist, dass das eben genannte Problem, dass daraus resultierende Kräfte oder Beschleunigungen den Impuls und Drehimpuls der Flüssigkeit nicht erhalten, behoben wird. Dies ist essentiell für robuste, stabile Simulationen. Der Nachteil, der in Kauf genommen wird, ist, dass nun konstante oder lineare Gradienten nicht mehr exakt berechnet werden können. Berechnet man die Druckbeschleunigung anhand dieser Diskretisierung erhält man:

$$-\frac{1}{\rho_i}\nabla p_f \approx -\sum_{ff} m_{ff} \left(\frac{p_f}{\rho_f^2} + \frac{p_{ff}}{\rho_{ff}^2} \right) \nabla W_{fff} \quad (3.25)$$

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
```
