

Institut für Visualisierung und Interaktive Systeme
Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Bachelorarbeit Nr. 10

Punktbasierte Animation deformierbarer Objekte mit der SPH-Methode

Karsten Schatz

Studiengang: Informatik
Prüfer: Prof. Dr. Daniel Weiskopf
Betreuer: Dipl.-Inf. Markus Huber

begonnen am: 21. Mai 2012
beendet am: 20. November 2012

CR-Klassifikation: I.3.7

Kurzfassung

Diese Bachelorarbeit präsentiert einen Ansatz um elastische Materialien punktbasiert zu animieren. Um dies zu erreichen, kommt die SPH-Methode zum Einsatz. Während andere Methoden weitaus zeitraubendere Verfahren wie etwa Moving Least Squares verwenden, können mit diesem Ansatz sehr gute Rechenzeiten erzielt werden. Vor allem besticht das Verfahren durch seine Erweiterbarkeit, insbesondere im Hinblick auf die Interaktion mit Stoffen, die sich in anderen Aggregatzuständen befinden. Dies alles ist durch eine Neuformulierung der Ruhelage eines elastischen Objekts möglich, bei der statt der Startposition des Objekts nur lokale Partikel-Nachbarschaften im Speicher gehalten werden. Diese Arbeit beschreibt die Umsetzung und Evaluation dieses Verfahrens, insbesondere im Hinblick auf dessen Robustheit und Effizienz. Diese Eigenschaften werden in einigen Experimenten, welche verschiedenste Parameter variieren, bestätigt.

Inhaltsverzeichnis

1	Einleitung	9
1.1	Zielsetzung	10
1.2	Überblick über die Arbeit	10
2	Verwandte Arbeiten	13
2.1	Gitterlose Methoden	13
2.2	Gitterbasierte Methoden	14
3	Grundlagen	15
3.1	Punktbasierte Simulation	15
3.1.1	Partikel	17
3.1.2	Newtonsche Gesetze	18
3.1.3	Zeitintegration	19
3.2	Smoothed Particle Hydrodynamics	21
3.2.1	Grundidee	21
3.2.2	Diskretisierung der Interpolation	22
3.2.3	Kernel-Funktion	23
3.3	Elastizitätstheorie	25
3.3.1	Spannung	25
3.3.2	Verzerrung	27
3.3.3	Hookesches Gesetz	28
3.3.4	Elastizitätsmodul und Poissonzahl	29
3.3.5	Verallgemeinerung des Hookeschen Gesetzes	29
4	Modell	33
4.1	Grundannahmen	33
4.1.1	Lokale Undeformiertheit	33
4.2	Verwendung von SPH im Elastizitätsmodell	34
4.2.1	Partikelvolumen	34
4.2.2	Verschiebungsfeld	34
4.2.3	Kernel-Funktion	36
4.3	Verzerrung und Spannung	36
4.4	Elastische Kraft	37
4.5	Zeitintegration	38
5	Implementierung	39
5.1	Umgebung	39

5.2	Simulationsablauf	40
5.2.1	Initialisierung	40
5.2.2	Ablauf eines Zeitschritts	42
5.3	Besondere Details	42
5.3.1	Gradient des Verschiebungsfeldes	43
5.3.2	Anrechnung der Kräfte	43
5.3.3	Grenzbehandlung	44
6	Ergebnisse	45
6.1	Variation des Elastizitätsmoduls	45
6.2	Variation der Poissonzahl	47
6.3	Variation von SPH-Grundparametern	48
6.4	Laufzeiten	49
7	Diskussion	53
7.1	Wahl von Elastizitätsmodul und Poissonzahl	53
7.2	Vor- und Nachteile bezüglich anderer Methoden	54
7.3	Erweiterbarkeit	55
8	Zusammenfassung und Ausblick	57
	Literaturverzeichnis	59

Abbildungsverzeichnis

3.1	Vergleich zwischen Eulerschen (links) und Lagrange-Methoden (rechts).	16
3.2	Spiky-Kernel (spitz) und Poly 6-Kernel (flach) für $h = 1$	23
3.3	Verhalten von SPH.	24
3.4	Kraft, welche in Richtung der Normale auf einen Zylinderschnitt wirkt (nach [Slao2]).	26
3.5	Longitudinale Verzerrung eines Quaders (nach [Slao2]).	27
3.6	Orthogonale Verzerrung eines Quaders (nach [Slao2]).	28
4.1	Beispiel für einen Verschiebungsdifferenzvektor	35
4.2	Elastizitätskernel (niedrige Kurve) und sein Gradient (hohe Kurve) für $h = 1$	36
5.1	Simulationsumgebung, in die das Verfahren integriert wurde	39
6.1	Startkonfiguration und Versuchsparameter Versuch 1	45
6.2	Ergebnisse Versuch 1 bei $E \in [50, 300]$	46
6.3	Ergebnisse Versuch 1 bei $E \in [500, 1000]$	46
6.4	Startkonfiguration und Versuchsparameter Versuch 2	47
6.5	Ergebnisse Versuch 2	48

Tabellenverzeichnis

5.1	Simulationsparameter	40
6.1	Laufzeitmessungen Total für 200 Zeitschritte	50
6.2	Laufzeitmessungen aufgeschlüsselt für 200 Zeitschritte	50

Verzeichnis der Algorithmen

- 5.1 Initialisierung der Simulation 41
- 5.2 Einzelner Zeitschritt der Simulation 42
- 5.3 Berechnung der Kräfte 44

1 Einleitung

Die Simulation und Animation von Materialien aller Art und deren Interaktionen untereinander hat in den vergangenen Jahren immer mehr an Bedeutung gewonnen. Kaum jemand wird die starke Zunahme von zumindest zum Teil animierten Filmen und Werbespots im Fernsehen nicht bemerkt haben. Der Anspruch besteht darin, Photorealismus zu erzeugen, wo keine echte Kamera verwendet werden kann oder soll. Bei Einzelbildern stellt dies ein eher geringes Problem dar. Soll jedoch eine Abfolge von Bildern, also eben ein Film etc., erzeugt werden, müssen auch die Übergänge zwischen diesen Bildern realistisch erscheinen. Wasser soll physikalisch korrekt fließen, ein Blatt soll, verlangsamt durch den Luftwiderstand, vom Baum fallen. Betrachtet man Einzelbilder dieser Szenen, kann jedes für sich genommen real aussehen, die zeitliche Abfolge mehrerer Bilder muss dies aber nicht. Es muss also gewährleistet werden, dass im Zeitverlauf physikalische Korrektheit gewahrt bleibt.

Physikalisch korrekte Simulationsmethoden von Materialien aller Art sind schon seit Jahrzehnten sowohl in der Industrie als auch in der Forschung in Verwendung. Mit deren Hilfe kann auch bei einer Computeranimation der Realitätseindruck gewahrt bleiben, indem man das korrekte Verhalten aller in der Szene vorhandenen Gegenstände simuliert. Die Computergrafik hat jedoch spezielle Anforderungen an diese Simulationsmethoden. Die dort verwendeten Verfahren müssen vor allem schnell sein. Insbesondere für interaktive Anwendungen wie etwa Computerspiele ist dies von Relevanz. Hier müssen mindestens 25 Einzelbilder pro Sekunde erreicht werden. Eine Methode welche das nächste Bild nach 5 Minuten berechnet hat, ist in diesem Anwendungsbereich also unnützlich. Dabei werden gerne auch Kompromisse im Bezug auf den physikalische Korrektheit eingegangen, solange das Ergebnis nur korrekt aussieht. Rein rechnerisch muss es dies aber nicht unbedingt sein. Gesucht sind also Simulationsmethoden, die schnell sind und ein gewisses Maß an realistischem Eindruck wahren.

Viele in der Computergrafik verwendeten Simulationsmethoden sind partikelbasiert. Das heißt, dass zu simulierende Objekte in kleine Partikel aufgeteilt werden, die dann untereinander, je nach Modell unterschiedlich, miteinander interagieren können. Eine dieser Methoden, die insbesondere bei Gasen zu guten Ergebnissen führt ist Smoothed Particle Hydrodynamics (SPH). SPH wurde ursprünglich entwickelt, um die Simulation von Gasdynamik in Weltraumnebeln zu vereinfachen, deren Berechnung mit Hilfe der damals existierenden Methoden, wie beispielsweise der Finite-Elemente-Methode (FEM), recht zeitintensiv waren. Einige Jahre nach dessen Veröffentlichung wurden auch zunehmend Verfahren entwickelt, welche SPH in der Fluidsimulation zur Anwendung bringen. Fluide können mit dieser Methode weitaus zeiteffizienter als bisher simuliert und animiert werden, was natürlich für Branchen wie die Animationsfilmindustrie recht interessant ist. Vor der Simulation von Feststoffen mittels SPH wurde bis vor kurzem jedoch zurückgeschreckt. Während bei Gasen

und Flüssigkeiten die Partikel nicht fest untereinander verbunden sind, erfordern Feststoffe meist eine bestimmtere Partikelanordnung. Diese ist weitaus schwerer zu modellieren als beispielsweise die recht freie Bewegung von Gasparkeln im Vakuum. Besonders elastische Materialien verursachen große Schwierigkeiten, da hier die Partikel innerhalb eines Objekts ihre Position ändern können, ohne dass das Objekt seinen Zusammenhalt verlieren darf.

Trotz all dieser Schwierigkeiten gelang es in den letzten Jahren, Verfahren zu entwickeln, welche elastische Feststoffe mit sehr zufriedenstellenden Ergebnissen modellieren und die den speziellen Anforderungen der Computergrafik genügen. In dieser Arbeit wird daher ein Verfahren präsentiert, welches in der Lage ist, elastische Materialien unter Verwendung von Smoothed Particle Hydrodynamics zu simulieren und zu animieren. Das physikalische Verhalten der simulierten Objekte muss dabei, wie schon erwähnt, nicht perfekt sein, jedoch muss es ästhetischen Ansprüchen genügen. Es sollen so gut wie alle elastischen Materialien vom Radiergummi bis hin zu Wackelpudding animiert werden können. Wie bereits ausgeführt, soll dieses Verfahren sowohl schneller als bereits existierende Methoden, als auch leicht erweiterbar sein. Dabei bedient es sich dem Elastizitätsmodell der Kontinuumsmechanik. Die genauere Zielsetzung dieser Arbeit soll nun im nächsten Abschnitt erläutert werden.

1.1 Zielsetzung

Das Ziel dieser Bachelorarbeit ist es, ein Verfahren zur Simulation deformierbarer Objekte umzusetzen und zu analysieren. Dieses soll auf einer Methode basieren, welche im Jahre 2007 von Solenthaler und Kollegen veröffentlicht wurde. Sie beschreiben in ihrer Arbeit „A Unified Particle Model for Fluid-Solid Interactions“ [SSP07] ein Modell, welches die Interaktion zwischen Feststoffen und Flüssigkeiten auf sehr einfache Weise umsetzt. Insbesondere gehen sie auf die elastischen Materialien ein und zeigen ein Verfahren auf, wie diese mittels SPH simuliert werden können. Auf genau dieses Verfahren stützt sich diese Bachelorarbeit.

Nach der Implementierung des Grundverfahrens für elastische Körper soll dieses auf seine Robustheit und seine Effizienz gegenüber anderen Verfahren untersucht werden. Es soll auch die physikalische Plausibilität der Simulation evaluiert werden.

1.2 Überblick über die Arbeit

Im Folgenden soll nun ein kurzer Überblick über die Arbeit gegeben werden. Kapitel 2 präsentiert einige verwandte Arbeiten mit ähnlichem Themengebiet. Themengebiete, die für diese Arbeit relevant sind. In Kapitel 3 soll dann zunächst auf einige für diese Arbeit wichtigen Grundlagen eingegangen werden. So behandelt Abschnitt 3.1 die Grundzüge der punktbasierten Simulation. Eine Methode, die dazu dient in einer punktbasierten Simulation einige nötige Werte zu bestimmen, namentlich Smoothed Particle Hydrodynamics, wird dann in Abschnitt 3.2 eingeführt. Abschnitt 3.3 behandelt dann die Elastizitätstheorie der Kontinuumsmechanik, soweit sie für diese Arbeit von Belang ist.

Kapitel 4 führt daraufhin das Simulationsmodell für elastische Körper ein, welches in [SSP07] präsentiert wird. Insbesondere wird auch auf die Anwendung von Smoothed Particle Hydrodynamics in diesem Falle eingegangen. In Kapitel 5 werden einige Implementierungsdetails erläutert.

Kapitel 6 präsentiert einige der Ergebnisse, die bei Anwendung dieser Simulationsmethode im Laufe dieser Arbeit erreicht worden sind. Die Abschnitte 6.2 und 6.1 zeigen die Resultate auf, die bei Variation von Parametern für die Elastizität auftreten. Zu Details zu diesen Materialeigenschaften sei auf Abschnitt 3.3.4 verwiesen. In Abschnitt 6.3 wird auf die verschiedenen SPH-Grundparameter eingegangen, beziehungsweise auf die Effekte ihrer Veränderung (siehe auch Abschnitt 3.2).

Die Diskussion der aufgezeigten Ergebnisse findet dann in Kapitel 7 statt. Eingangs wird dort die Plausibilität der Ergebnisse für bestimmte elastische Parameter diskutiert. Daraufhin wird dann noch auf die Vor- und Nachteile der Methode eingegangen und abschließend ihre Erweiterbarkeit aufgezeigt. Kapitel 8 fasst diese Arbeit dann schlussendlich zusammen und gibt einen kurzen Ausblick auf mögliche künftige Weiterentwicklungen.

2 Verwandte Arbeiten

Die Forschung in den Bereichen der Simulation von Elastizität hat im Laufe der Jahre eine Vielzahl von Arbeiten und Modellen hervorgebracht. Die Elastizitätstheorie selbst ist sehr weit ausgearbeitet. Zusammenfassungen, insbesondere mit Fokus auf die Anwendbarkeit in der Computergraphik, sind beispielsweise [TPBF87], [GM97] oder [NMK⁺06]. Welche Arbeiten noch von Relevanz sind, soll in den folgenden beiden Abschnitten gezeigt werden.

Bei den präsentierten Arbeiten kann zwischen den sogenannten gitterbasierten und den gitterlosen Methoden unterschieden werden. Während bei ersteren zwischen den simulierten Objekten ein gewisser räumlicher Zusammenhang, also ein Gitter, besteht, können sie sich bei letzteren weitestgehend frei bewegen, ohne durch eine Gitterstruktur eingeschränkt zu sein.

2.1 Gitterlose Methoden

Bei den meisten gitterlosen Methoden handelt es sich um Partikelsysteme. Deren ursprüngliche Grundlagen liegen in der newtonschen Punktmechanik, die schon im 17. Jahrhundert von Isaac Newton ausgearbeitet wurde [New87]. In der Computergraphik kam diese Art der Modellierung jedoch erst recht spät an. 1983 wurde in [Ree83] ein Verfahren präsentiert, das etwa Feuer mittels Partikelsystemen anschaulich visualisieren kann. [LW85] führt Punkte als Bildschirmprimitive ein und ermöglicht somit, dass Simulationsergebnisse von punktbasierten Methoden anschaulich präsentiert werden können. Auf dem Gebiet der Simulation hat sich vor allem Smoothed Particle Hydrodynamics hervorgetan. Diese Methode wurde in [GM77] und [Luc77] erstmals präsentiert. Reviews der Methode stammen vor allem von Monaghan ([Mon92], [Mon05a], [Mon05b]). In [SM93] wird versucht mittels einer eigenen Implementierung die Grenzen von SPH aufzuzeigen. In den folgenden Jahren kam diese Methode immer mehr in der Computergrafik an. So präsentierten Desbrun und Kollegen ([DC96]) eine Methode, welche in der Lage ist verschiedenste einfache Materialien zu animieren.

Infolgedessen erschien eine Vielzahl von Arbeiten mit unterschiedlichen Schwerpunkten. In [MCG03] wird eine Methode präsentiert, mit der Fluide mittels SPH simuliert werden können. Die Simulation elastischer Körper wird etwa in [MKN⁺04] oder [SSP07] thematisiert. Während sich erstere Arbeit auf feste, elastische und schmelzende Körper konzentriert, versucht letztere, ein universales Partikelmodell zu entwickeln, um die Interaktion von Objekten verschiedener Aggregatzustände zu vereinfachen. Erstere verwendet dazu Moving Least Squares, was recht zeitintensiv sein kann, bei dem universalen Partikelmodell kommt

das SPH-Modell zum Einsatz. Wie schon erwähnt, basiert das in dieser Arbeit präsentierte Verfahren auf der Arbeit von Solenthaler und Kollegen.

Ebenfalls relevant ist [KAG⁺05], welche ein ähnliches Verfahren präsentiert, das jedoch mehr auf Zustandsübergänge zwischen fest und flüssig fokussiert ist. Elastizität ist dort nur ein Nebenprodukt. Das Elastizitätsmodell von [SSP07] wird in [BIT09] erweitert, um Rotationsinvarianz bei elastischen Körpern zu ermöglichen. In [GBB09] konzentriert man sich, ebenfalls basierend auf [SSP07], ausschließlich auf elastische Materialien und es wird keine initiale Referenzkonfiguration von Objekten verwendet, wie es bei vorhergehenden Modellen der Fall war. Ein wichtiger Aspekt der Simulation elastischer Materialien, das Kontaktverhalten mehrerer Objekte, wird in [KMH⁺04] thematisiert. Zusätzlich zu den Methoden zur Simulation sind auch Methoden zur Oberflächenrekonstruktion von Relevanz. In [BGB11] wird z. B. eine solche Methode präsentiert.

2.2 Gitterbasierte Methoden

Neben den gitterlosen Methoden sind die gitterbasierten von hoher Relevanz. Vor allem in der Industrie sind letztere beliebter bzw. weiter verbreitet, da dort eher auf Korrektheit denn auf Schnelligkeit Wert gelegt wird. Das soll jedoch nicht heißen, dass gitterbasierte Methoden zwangsläufig korrekter sind als gitterlose. Zu ihnen gehören etwa die Finite-Elemente-Methode (FEM) oder Masse-Feder-Systeme. Eine Einführung in die FEM wird beispielsweise in [Bra03] vorgenommen. Dank der natürlichen elastischen Eigenschaften von Federn, werden Masse-Feder-Systeme vor allem zur Simulation von elastischen Materialien oder Stoffen verwendet. Masse-Feder-Systeme sind jedoch zur Simulation einiger Eigenschaften von elastischen Objekten ungeeignet. In [CZKM98] wird ein solches System zur Animation fallender Gegenstände verwendet. Speziell mit den Eigenschaften von elastoplastischen Materialien beschäftigen sich zwei Arbeiten von O'Brian, welche die FEM verwenden. In [OH99] werden vor allem spröde Materialien thematisiert, [OBH02] präsentiert ein Verfahren für dehnbare Objekte. In beiden Arbeiten wird speziell auf das Bruchverhalten unter großen Kräften eingegangen. Ein Verfahren, das sich vor allem für sehr starke Deformationen eignet, wird in [WTo8] präsentiert. Da insbesondere FEM im Gegensatz zu vielen gitterlosen Methoden recht zeitintensiv ist, findet es in der Computergrafik seltener Anwendung.

3 Grundlagen

Dieses Kapitel beschäftigt sich mit den nötigen Grundlagen, die zum Verständnis der Simulationsmethode dieser Arbeit vonnöten sind. Abschnitt 3.1 führt den Begriff der Punktbasierten Simulation ein. Abschnitt 3.2 führt dies dann weiter und erläutert die Funktionsweise von Smoothed Particle Hydrodynamics, welches eine Methode darstellt, die bestimmte benötigte Größen in einer Punktbasierten Simulation berechnen kann. Abschließend wird noch auf die Elastizitätstheorie der Kontinuumsmechanik eingegangen, welche in vielen Simulationsmodellen Anwendung findet (Abschnitt 3.3).

3.1 Punktbasierte Simulation

Seit jeher ist es das Ziel von Physikern und anderen Wissenschaftlern das Verhalten von allen Dingen zu 100% genau vorhersagen zu können. Die kontinuierliche Natur des Universums ist hierbei jedoch höchst hinderlich. Hat man genaue Aussagen über zwei Punkte im Raum getroffen, heißt dies nicht zwangsläufig, dass man daraus das dazwischen liegende einfach folgern kann. Auch für Computer-Simulationen ist dies sehr hinderlich, da Rechner nur mit einer endlichen Datenmenge umgehen können. Somit ist also eine Vereinfachung der allgemeinen Modelle notwendig, um effizient Vorhersagen treffen zu können. Dies kann getan werden, indem man nur die Werte von endlich vielen Messstellen berücksichtigt. Hierfür gibt es zwei grundlegend unterschiedliche Taktiken.

Es scheint naheliegend, dass man den Raum einfach aufteilen kann, wenn man beispielsweise jeden Millimeter einen Messpunkt anbringt. So kann ein würfelförmiger Raum mit 10cm Kantenlänge mit Hilfe von 1000000 Messpunkten beschrieben werden. Der Unterschied der beiden erwähnten Taktiken besteht nun darin, wie man diese Messpunkte behandelt. Will man in diesem Würfel etwa die Bewegung von Wasser simulieren, kann man die Messpunkte an ihrer Ursprungsposition belassen und etwa die Druckverteilung zwischen diesen Messpunkten simulieren. Die Punkte sind, wie beschrieben, in einem Gitter angeordnet [NMK⁺06].

Die zweite Möglichkeit besteht darin, jeden Punkt als Bestandteil des zu simulierenden Gegenstands zu betrachten (hier: als Wasserteilchen). Bewegt sich das Wasser im Würfel, bewegen sich die das Wasser repräsentierenden Teilchen ebenfalls, gleichförmig mit dem Wasser. So besteht die Möglichkeit, dass bestimmte Teile des Raums nicht betrachtet werden (jene ohne Wasser darinnen), andere jedoch schon. Jeder Punkt, bzw. jedes Partikel repräsentiert einen Teil der Eigenschaften des gesamten Wassers im simulierten Raum [NMK⁺06].

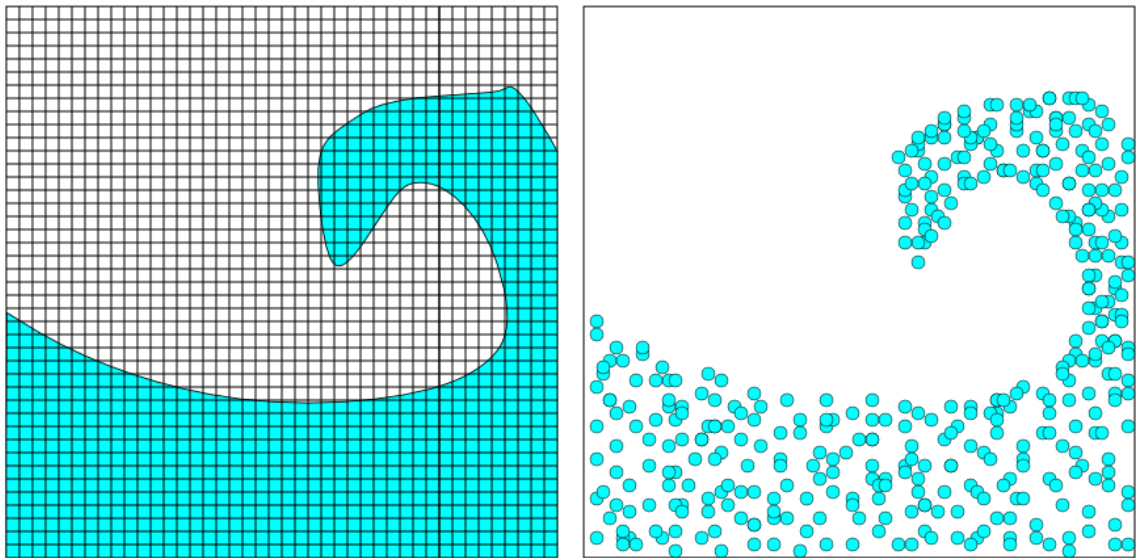


Abbildung 3.1: Vergleich zwischen Eulerschen (links) und Lagrange-Methoden (rechts).

Hier ist also leicht der Unterschied zwischen beiden Herangehensweisen sichtbar. Bei dem ersten sogenannten Eulerschen Ansatz wird der Raum diskretisiert. Bei letzterem, dem Lagrange-Ansatz diskretisiert man das zu simulierende Material [NMK⁺06]. Dies sei an Abbildung 3.1 verdeutlicht.

Während links beim Eulerschen Verfahren bei vielen Messpunkten kein Wasser vorhanden ist, ist dies rechts beim Lagrange-Ansatz nicht der Fall. Somit wird vermieden, leeren Raum simulieren zu müssen, was Rechenleistung spart. Im Gegenzug ist natürlich in einem Raum mit höherer Partikeldichte entsprechend mehr Leistung erforderlich. Eine komplett gefüllte Domäne wird gegenüber dem Eulerschen Ansatz eventuell sogar im Nachteil sein, was die reine Rechendauer angeht, da somit der Vorteil des Lagrange-Ansatzes verloren geht.

Um dem Lagrange-Ansatz gerecht zu werden kann man das zu simulierende Medium beliebig unterteilen. Die einfachste Möglichkeit dazu besteht darin, dies mittels einer endlichen Menge von Punkten zu tun. Bei der weiteren Verfahrensweise hat man nun wiederum zwei Möglichkeiten. Entweder dürfen die Punkte sich frei im Raum bewegen, oder sie werden gitterartig untereinander verbunden. Bei ersteren handelt es sich um die punktbasierten, bei letzteren um die gitterbasierten Methoden.

Die punktbasierten Systeme wurden im Jahre 1983 von William T. Reeves in [Ree83] eingeführt. Damals dienten sie jedoch eher zur eher ästhetisch motivierten Animation denn zur Simulation von physikalischen Phänomenen. Mit ihnen wurde im Film „Star Trek II: Der Zorn des Khan“ die Ausbreitung einer Schockwelle eines Bombeneinschlags dargestellt.

Für beide Ansätze existieren die unterschiedlichsten Verfahren die auf ihnen basieren. Während beispielsweise die Finite-Elemente-Methode (FEM) [Bra03] zu den gitterbasierten Methoden gezählt wird, gehören beispielsweise Smoothed Particle Hydrodynamics (siehe Abschnitt 3.2) oder eben die von Reeves beschriebenen „Loosely coupled particle systems“ [Ree83] zu den punktbasierten Methoden.

Im Folgenden sollen nun die grundlegenden Methodiken und Annahmen erläutert werden, die für eine punktbasierte physikalisch motivierte Simulation notwendig sind.

3.1.1 Partikel

In punktbasierten Systemen sind, wie zu erwarten, die Punkte bzw. die Partikel die wichtigsten Bestandteile. Sie repräsentieren die zu simulierenden Materialien. Die Beschreibung eines Objekts in einem solchen System wird also durch die Beschreibung all seiner zugehörigen Partikel ersetzt. Um dies leisten zu können, werden jedem Partikel diverse Eigenschaften mitgegeben [Ree83]. Diese Eigenschaften können variieren, je nachdem zu welcher Aufgabe das System in der Lage sein soll. Die gebräuchlichsten sind hierbei:

Position \mathbf{x} : Wird im Normalfall im dreidimensionalen Raum angegeben, also $\mathbf{x} \in \mathbb{R}^3$. Sie beschreibt den genauen Punkt im gegebenen Koordinatensystem, in dem sich das Partikel aktuell befindet. Natürlich auch zweidimensional möglich ($\mathbf{x} \in \mathbb{R}^2$).

Geschwindigkeit \mathbf{v} : Im dreidimensionalen Raum gilt, analog zur Position, $\mathbf{v} \in \mathbb{R}^3$. Die Richtung des Vektors beschreibt die aktuelle Bewegungsrichtung des Partikels, die Länge beschreibt den Betrag der momentanen Geschwindigkeit.

Zu diesen sehr grundlegenden Eigenschaften können noch einige hinzukommen, die speziell für physikalische Simulationen von Belang sind:

Masse m : Wird als Skalar $m \in \mathbb{R}$ angegeben. Beschreibt die Masse des Raumteils, für den das Partikel zuständig ist. Summiert man die Masse aller Partikel eines Objekts, erhält man die Objektmasse.

Volumen v : Beschreibt die Größe des Raumteils, für den das Partikel zuständig ist. Das Volumen ist besonders in [SSP07] von Relevanz. Im Gegensatz zur Geschwindigkeit handelt es sich hierbei nicht um einen Vektor sondern um ein Skalar, trotz desselben Symbols.

Temperatur T : Repräsentiert die aktuelle Temperatur an der Partikelposition.

Da Partikelsysteme nicht nur für physikalisch motivierte Simulationen von Belang sind, sondern auch im Bereich der Visualisierung eine große Rolle spielen, können noch folgende Eigenschaften Anwendung finden:

Farbe: Typischerweise als RGB-Farbwert angegeben. Ist vor allem bei der Visualisierung von Relevanz. Kann sich im Zeitverlauf ändern um beispielsweise Temperaturänderungen visuell zu verdeutlichen.

Transparenz: Ist gewissermaßen der α -Kanal zu der Farbe des Partikels. Bei der Flüssigkeitsvisualisierung kann ein Partikel evtl. leicht durchsichtig sein. Diese Durchsichtigkeit wird im Transparenzwert festgelegt.

Form: Auch meist für die Visualisierung von Relevanz, beschreibt das äußere Erscheinungsbild eines Partikels.

Aus allen Eigenschaften, die die Partikel eines Systems besitzen, können dann die neue Position bzw. die neuen Eigenschaften errechnet werden, die ein Partikel zu einem späteren Zeitpunkt innehat. So gut wie alle punktbasierten Methoden gehen nach diesem Schema vor. Der Unterschied zwischen ihnen besteht zumeist darin, *wie* diese Eigenschaften berechnet werden. Im Regelfall wird dazu eine Interpolationsmethode verwendet, da eine endliche Anzahl von Stützstellen (die Partikel) existieren, auf deren Basis die Berechnung durchgeführt werden kann. Aufgrund der hohen Anzahl unterschiedlicher existierender Interpolationsalgorithmen ist die Anzahl der Möglichkeiten für solche Simulationen sehr hoch. Zu den bekanntesten und verbreitetsten zählen unter anderem SPH oder Moving Least Squares (MLS). Diese nehmen die Eigenschaften benachbarter Partikel als Eingabewerte, um daraus andere Eigenschaften des betrachteten Partikels zu berechnen.

3.1.2 Newtonsche Gesetze

Die Diskretisierung mittels Partikeln kann in unterschiedlichen Größenordnungen durchgeführt werden. Je nach Anwendungsfall kann man etwa einen $1m^2$ -Gummiwürfel mit einem oder einer Milliarde Partikeln darstellen. Punktbasierte Simulationen verwenden häufig ein makroskopisches Modell. Das heißt, dass unter anderem Effekte auf atomarer Ebene vernachlässigt werden. Natürlich könnten diese auf irgendeine Weise ebenfalls modelliert werden, jedoch wird in diesem Zusammenhang auf die Beschreibung dieser Fälle verzichtet, da sie für diese Arbeit nicht relevant sind. Zudem spielen auch relativistische Effekte meist keine Rolle. Das immer noch beliebteste Modell für diese Fälle ist die Newtonsche oder Klassische Mechanik, die zu großen Teilen noch aus dem 17. Jahrhundert stammt. Insbesondere die Berechnung der Partikelpositionen und -geschwindigkeiten geschieht mit Hilfe von Gleichungen aus der klassischen Mechanik. Im Regelfall wird in den meisten Arbeiten nur eine Berechnungsformel für die auftretenden Kräfte zwischen den Partikeln gegeben. Die Umrechnung der Kräfte in resultierende Partikelgeschwindigkeiten und neue Positionen im Zeitverlauf geschieht, fast schon selbstverständlich, durch die eben erwähnten Gleichungen.

Angenommen man versucht, Wasserbewegungen zu simulieren. In einer solchen Simulation werden die einzelnen Wasserteilchen (nicht zu verwechseln mit H_2O -Molekülen, da es sich um ein makroskopisches Modell handelt) untereinander mittels der auftretenden Kräfte wechselwirken. Sammeln sich viele Teilchen nahe beieinander, kann dies als erhöhter Druck

bzw. erhöhte Dichte an dieser Stelle interpretiert werden. Da Wasser für die meisten Berechnungen als inkompressibles Fluid angesehen werden kann, wird das System nun versuchen, die erhöhte Dichte auszugleichen. Dies wird modelliert, indem man Kräfte einführt, die von einem Partikel auf andere, sich zu nahe befindende, Partikel wirken. Die Kraft wird zu nahe Partikel von dem Partikel, von dem sie ausgeht, wegschieben, um den erhöhten Druck auszugleichen.

Seien alle Partikel von 1 bis $n \in \mathbb{N}$ durchnummeriert. Dann steht P_1 für das erste Partikel und P_i für das i -te, wobei $1 \leq i \leq n$ und $i \in \mathbb{N}$. Die Kraft, die von Partikel P_j auf P_i wirkt, wird dann als F_{ij} bezeichnet. Zusätzlich zu den Kräften zwischen den Partikeln können noch externe Kräfte auftreten, wie beispielsweise die Gravitation, die meist als konstante, nach unten wirkende Kraft modelliert wird. Diese seien als F_{i_ext} bezeichnet. Die Summe aller Kräfte, die auf P_i wirken, ist F_i . Als Anmerkung sei gesagt, dass es sich bei der Kraft um eine vektorielle Größe handelt, der Vektorpfeil wird jedoch der Übersicht halber weggelassen und durch fetten Druck ersetzt. Mit Hilfe der drei Newtonschen Gesetze [New87] können dann aus den Kräften und den schon bekannten Positionen und Geschwindigkeiten die neuen Positionen und Geschwindigkeiten der Partikel berechnet werden. Das zweite dieser drei Gesetze (das so genannte Aktionsprinzip), wobei m_i die Masse des i -ten Partikels angibt und x_i dessen Position, lautet:

$$\mathbf{F}_i = m_i * \ddot{\mathbf{x}}_i \quad (3.1)$$

Umgeformt nach der Position x_i kann nun mit Hilfe dieses Gesetzes die neue Position des Partikels im Raum berechnet werden. Der Wert $\dot{\mathbf{x}}_i = \mathbf{v}_i$ wird gemeinhin als Geschwindigkeit von P_i bezeichnet. Des weiteren wird $\ddot{\mathbf{x}}_i = \dot{\mathbf{v}}_i = \mathbf{a}_i$ als dessen Beschleunigung bezeichnet.

Bevor jedoch die neuen Positionen bestimmt werden können, kann es von Vorteil sein sicherzustellen, dass das erste und das dritte Newtonsche Gesetz erfüllt ist. Das erste ist das der Trägheit. Es besagt, dass ein Partikel, auf das keine Kräfte wirken, seine aktuelle Geschwindigkeit bzw. seinen aktuellen Impuls beibehält. Das dritte Gesetz ist das von *actio* und *reactio*. Es besagt, dass Kräfte immer in Paaren auftreten, die sich gegenseitig auslöschen. Übt P_j auf P_i eine Kraft F_{ij} aus, so gilt:

$$\mathbf{F}_{ij} = -\mathbf{F}_{ji} \quad (3.2)$$

Somit wirkt von P_i auf P_j eine betragsmäßig gleich große Gegenkraft. Final bleibt nur noch die Lösung einiger Differentialgleichung übrig, um aus der umgeformten Gleichung 3.1 die neue Geschwindigkeit und Position zu errechnen. Dieses Verfahren wird auch Zeitintegration genannt, wobei der Zustand des Systems nach endlichen, jeweils gleich großen Zeitschritten bestimmt wird.

3.1.3 Zeitintegration

Die Zeitintegration ist ein Faktor für die Stabilität bzw. Robustheit einer Simulation. Wird der Zeitschritt zu groß gewählt, oder das falsche Verfahren angewandt, kann die Simulation unter Umständen nicht beabsichtigte Ergebnisse liefern.

Formt man nun die Gleichung $F_i = m_i * a$ nach der Beschleunigung a um, erhält man eine Differentialgleichung zweiter Ordnung. Hierbei handelt es sich wirklich um eine Differentialgleichung, da F_i von der aktuellen Position x_i abhängig ist:

$$\ddot{x}_i = a = \frac{F_i}{m_i} \quad (3.3)$$

Dies ist ein Spezialfall einer Differentialgleichung, ein sogenanntes Anfangswertproblem, da die Position x_i^0 , die Geschwindigkeit \dot{x}_i^0 und die Beschleunigung \ddot{x}_i^0 zum Zeitpunkt $t = 0$ gegeben sind. Diese können nämlich zum Start der Simulation frei gewählt werden, je nachdem wie das zu simulierende Objekt aussehen soll und welche Bewegung es zum Start auszuführen hat. Allgemein soll der Wert x_i^t die Position des Partikels P_i zum Zeitpunkt t bezeichnen.

Dieses Anfangswertproblem könnte beispielsweise mit dem expliziten Euler-Verfahren (siehe u.a. [AP98], [WB97]) gelöst werden. Neben dem Expliziten Euler-Verfahren existiert eine ganze Reihe von anderen Verfahren, die ebenfalls in der Lage sind, lineare Differentialgleichungen numerisch zu lösen. Das Explizite Euler-Verfahren zählt darunter noch zu den einfachen. Da seine Anwendung die physikalischen Beziehungen fast 1 : 1 wiedergibt, wird es sehr gerne der Anschaulichkeit halber verwendet, auch wenn zur eigentlich Integration dann ein anderes Verfahren zum Zuge kommt. Zu anderen Lösern für Differentialgleichungen kann ebenfalls [AP98] zu Rate gezogen werden. Unter Verwendung des Euler-Verfahrens ergeben sich dann folgende Beziehungen, welche leicht gelöst werden können, da alle Werte bekannt sind:

$$\begin{aligned} \dot{x}_i^{t+\delta t} &= \dot{x}_i^t + \ddot{x}_i^t * \delta t \\ x_i^{t+\delta t} &= x_i^t + \dot{x}_i^{t+\delta t} * \delta t \end{aligned} \quad (3.4)$$

Mit diesen Gleichungen kann die neue Position eines Partikels P_i , bezeichnet mit $x_i^{t+\delta t}$ aus dessen alter Position x_i^t berechnet werden. Der Wert δt bezeichnet den zeitlichen Abstand, mit dem das System abgetastet werden soll. Die Wahl der Größe dieses Zeitschritts ist essentiell für die Stabilität des Systems. In den meisten physikalischen Simulationen wird ein zu großes δt zu Instabilität führen. Wird es sehr klein gewählt, wirkt sich dies auf die Rechendauer aus, da dann für die selbe Simulationsdauer mehr Zeitschritte absolviert werden müssen. Es existiert auch die Möglichkeit, die Länge des Zeitschritts adaptiv zu bestimmen, indem man eine Stabilitätsbedingung formuliert, die vom System eingehalten werden muss. Dann kann δt gerade klein genug gewählt werden, dass das System nicht instabil wird. Somit könnte dann jeden Zeitschritt ein anderes δt verwendet werden.

Das endgültige Schema für eine Punktbasierte Simulation lautet schließlich:

1. Berechnung der Kräfte F_i^t
2. Berechnung der neuen Geschwindigkeiten $v_i^{t+\delta t}$
3. Berechnung der neuen Positionen $x_i^{t+\delta t}$

Dieses Schema kann so oft wiederholt werden, wie nötig. Will man z.B. eine Simulation von 1s Länge ausführen, sind $1s/\delta t$ Wiederholungen notwendig. So gut wie jede physikalisch basierte Simulation folgt einem Schema dieser Art. Je nach Differentialgleichungslöser kann jedoch die Schrittreihenfolge variieren. Im folgenden Kapitel soll nun ein Verfahren vorgestellt werden, welches bestimmte benötigte Werte in einer punktbasierten Simulation diskret annähern kann: Smoothed Particle Hydrodynamics.

3.2 Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics wurde ursprünglich von L. Lucy ([Luc77]), R. A. Gingold und J. J. Monaghan ([GM77]) entwickelt, um die Gasdynamik in Sternensystemen zu simulieren. Die beide Arbeiten, in denen diese Systematik zuerst publiziert wurde, entstanden unabhängig voneinander, auch wenn Gingold und Monaghan in [GM77] klarstellen, dass die Grundidee von Lucy in einer seiner Vorlesungen in Cambridge 1976 bereits erwähnt wurde und sie diese nur weitergeführt hatten. Gemeinhin wird jedoch meist Monaghan als „Vater“ von SPH dargestellt, da zum einen der Begriff SPH in [GM77] geprägt wurde und er zum anderen viele weitere Forschungen in dem Gebiet angestellt hat.

Im Laufe der Jahre hat sich herausgestellt, dass das ursprüngliche Verfahren allgemein und simpel genug gehalten war, um einfach für andere Probleme adaptiert werden zu können. Beispielsweise wird von Desbrun und Kollegen in [DC96] die Simulation unterschiedlichster Stoffe mit stark variierenden Parametern präsentiert. Darunter befinden sich sowohl Flüssigkeiten als auch Feststoffe. Von Müller und Kollegen werden in [MCG03] optisch sehr ansprechende Ergebnisse mit Fluiden erzielt.

Wie genau SPH funktioniert, soll im Folgenden geklärt werden.

3.2.1 Grundidee

Smoothed Particle Hydrodynamics basiert auf einem Interpolationsverfahren, mit dem man einen skalaren Wert mit Hilfe eines Integrals interpoliert. Die Herleitung der SPH-Grundgleichung beginnt bei der folgenden Identität:

$$A(\mathbf{r}) = \int_{\Omega} A(\mathbf{x})\delta(\mathbf{r} - \mathbf{x})d\mathbf{x} \quad (3.5)$$

Hier bezeichnet $\delta(x)$ die so genannte Diracsche Delta-Funktion, welche unter anderem wie folgt definiert werden kann [Wal94]:

$$\delta(x) = \begin{cases} \infty & , \text{wenn } x = 0 \\ 0 & , \text{wenn } x \neq 0 \end{cases} \quad (3.6)$$

Zusätzlich gilt noch die Bedingung:

$$\int_{-\infty}^{\infty} \delta(x) dx = 1 \quad (3.7)$$

Somit ist offensichtlich, dass die Identität 3.5 Gültigkeit besitzen muss, denn $\delta(\mathbf{r} - \mathbf{x})$ nimmt nur einen Wert $\neq 0$ an, wenn \mathbf{r} gleich \mathbf{x} ist. $A(\mathbf{r})$ kann als eine Größe angesehen werden, die wir approximieren wollen, sie könnte beispielsweise für die Dichte bei einem bestimmten Partikel im Partikelsystem stehen. Um nun aus Gleichung 3.5 eine diskretisierbare Interpolationsformel zu erhalten, ersetzt man $\delta(x)$ durch $W(x, h)$, die so genannte Kernel-Funktion. Zu näheren Details zur Kernel-Funktion, siehe auch Abschnitt 3.2.3. Sie sollte für gültige Ergebnisse folgende Eigenschaften besitzen:

$$\begin{aligned} \lim_{h \rightarrow 0} W(\mathbf{r}, h) &= \delta(\mathbf{r}) \\ \int W(\mathbf{r}, h) dx &= 1 \end{aligned} \quad (3.8)$$

Zudem sollte sie symmetrisch zur y -Achse sein und in einem endlichen Bereich um $x = 0$ ungleich 0 sein. Der Wert h ist die so genannte Glättungslänge (engl. Smoothing Length). Ihre Größenordnung spielt eine wichtige Rolle bei der Genauigkeit der nun aus Gleichung 3.5 erstellten Interpolationsformel, bei der $\delta(r)$ durch $W(r, h)$ ersetzt wird:

$$A(\mathbf{r}) \approx A_I(\mathbf{r}) = \int A(\mathbf{x}) W(\mathbf{r} - \mathbf{x}, h) d\mathbf{x} \quad (3.9)$$

Anhand der Definition von $W(\mathbf{r}, h)$ kann man erkennen, dass die Genauigkeit der Interpolation zunimmt, je kleiner die Glättungslänge h ist, denn dann wird die Delta-Funktion besser approximiert. Diese Interpolationsformel muss jedoch noch diskretisiert werden, da wir eine diskrete Zahl von Partikeln besitzen, deren Werte wir approximieren wollen.

3.2.2 Diskretisierung der Interpolation

Wie Gleichung 3.9 diskretisiert wird, variiert je nach den Anforderungen, die gestellt werden. In der Literatur wird dies gerne exemplarisch für den Anwendungsfall eines Fluids getan [Mono5b]. Dies soll hier nun auch geschehen. Ein Fluid wird in eine Menge kleiner Volumenelemente unterteilt. Wie schon in Abschnitt 3.1 dargelegt, kann man diese Elemente als Partikel ansehen. Ein Partikel P_x habe dabei Masse m_x , Dichte ρ_x und Position \mathbf{r}_x . Mit Hilfe dieser Werte kann das Interpolationsintegral wie folgt geschrieben werden [Mono5b]:

$$\int \frac{A(\mathbf{r}')}{\rho(\mathbf{r}')} \rho(\mathbf{r}') d\mathbf{r}' \quad (3.10)$$

Mit der Zusatzinformation, dass die Masse eines Volumens dem Integral über die Dichte des Volumens entspricht, kann Gleichung 3.9 mit einer Summe über die Massen approximiert werden. Die finale, diskretisierte SPH-Gleichung lautet also:

$$A_S(\mathbf{r}) = \sum_b m_b \frac{A_b}{\rho_b} W(\mathbf{r} - \mathbf{r}_b, h) \quad (3.11)$$

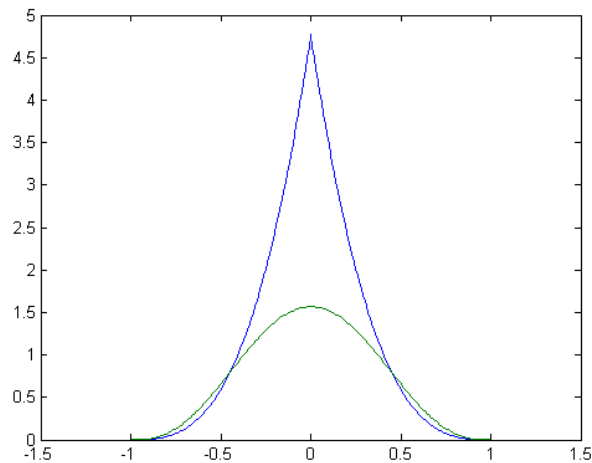


Abbildung 3.2: Spiky-Kernel (spitz) und Poly 6-Kernel (flach) für $h = 1$.

Für A kann hier so gut wie jede benötigte Partikeleigenschaft eingesetzt werden, sei es die Dichte oder sonstige Eigenschaften die für jeweilig verwendete Modell nötig sind. Aufgrund dessen gilt SPH als sehr flexibel und vielseitig anwendbar, da so gut wie alle zu berechnenden Werte mit Hilfe der Formeln approximiert werden können. Sei einer dieser benötigten Werte die Dichte. In diesem Fall kürzt sich Gleichung 3.11 zu folgendem:

$$\rho(\mathbf{r}) = \sum_b m_b W(|\mathbf{r} - \mathbf{r}_b|, h) \quad (3.12)$$

Somit wird die Dichte bei einem bestimmten Partikel allein aus den Massen und den Abständen zu den umliegenden Partikeln berechnet (h wird als gegeben angenommen). Abschließend bleibt nur noch zu klären, welchen Einfluss die Kernel-Funktion $W(\mathbf{r}, h)$ besitzt.

3.2.3 Kernel-Funktion

Wie schon angeführt, besitzt die Kernel-Funktion $W(\mathbf{r}, h)$ und einer ihrer Eingabeparameter, die Glättungslänge h , einen enormen Einfluss auf die Güte und die Performance der Approximation. Man rufe sich noch mal die Eigenschaften der Funktion aus Gleichung 3.8 in Erinnerung. Der Zweck der Kernel-Funktion ist es, eine Gewichtung aufgrund eines gegebenen Partikelabstandes \mathbf{r} bereitzustellen, abhängig von der Glättungslänge h . Da sie die Diracsche Delta-Funktion annähern soll, befindet sich ihr Maximum bei $W(0, h)$. Das Maximum der Delta-Funktion ist ebenfalls bei $\delta(0)$. Es sei noch angemerkt, dass der Eingabeparameter \mathbf{r} eher als $|\mathbf{r}|$ zu verstehen ist, da man einzig und allein am Abstand zweier Partikel interessiert ist, nicht an deren Relativposition. Dennoch ist auch eine Abhängigkeit von der Relativposition denkbar. Diese Schreibweise hat sich jedoch so eingebürgert.

In der Praxis wählt man für $W(\mathbf{r}, h)$ eine Funktion, die von ihrem Maximum bei $x = 0$ nach außen hin abfällt und symmetrisch zur y -Achse ist. Um die Performance der Approximation

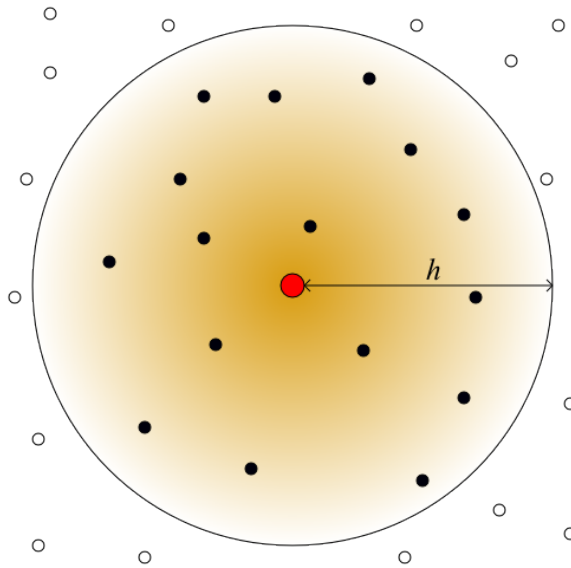


Abbildung 3.3: Verhalten von SPH.

zu verbessern, sind diese Funktionen so designt, dass sie bei Werten von $|\mathbf{r}| \geq 2h$ konstant 0 sind [Mong92]. In den Anfangszeiten nahm man dazu meist eine abgeschnittene Gauss-Kurve. Jedoch stellte sich schon bald heraus, dass eine Gauss-Kurve nicht immer den geeignetsten Kernel darstellt. Daher ging man dazu über, für jeden zu approximierenden Wert eigene Funktionen zu entwickeln. So entwickelten etwa Desbrun und Kollegen folgende Kernel-Funktion, den so genannten Spiky-Kernel [DC96]:

$$W(|\mathbf{r}|, h) = \frac{15}{\pi h^6} \begin{cases} (h - |\mathbf{r}|)^3 & ; 0 \leq |\mathbf{r}| \leq h \\ 0 & ; \text{sonst} \end{cases} \quad (3.13)$$

Der Graph für diese Funktion ist in Abbildung 3.2 gegeben, wobei es sich um die spitz zulaufende Funktion handelt. Bei der flacheren Funktion handelt es sich um den Poly 6-Kernel von Müller und Kollegen [MCG03]. Während ersterer speziell für die Berechnung der Dichte entwickelt wurde, handelt es sich bei letzterem um einen Kernel, der in multiplen Gebieten zum Einsatz kommt. Wie an den beiden Graphen erkennbar ist, sinken die Funktionswerte schon bei $|\mathbf{r}| \geq h$ auf 0 und nicht erst bei $|\mathbf{r}| \geq 2h$. Dieses Verhalten wurde erst in neueren Arbeiten eingeführt ([DC96], [MCG03], etc.). Man wollte wohl direktere Kontrolle haben, ab welchem Abstand zum betrachteten Partikel die Werte der anderen Partikel an Relevanz verlieren.

Das Gesamtverhalten von SPH sei auch an Abbildung 3.3 verdeutlicht. Angenommen, man will die Dichte an der Position des mittleren Partikels (rot, groß) berechnen. In diesem Fall sind nur alle Partikel von Relevanz, die sich innerhalb der Glättungslänge h um das betrachtete Partikel befinden (schwarz). Bei allen anderen (weiß) beträgt der Rückgabewert der Kernel-Funktion 0 und sie sind somit nicht relevant. Je näher sich ein Partikel an

dem betrachteten befindet, desto relevanter ist es für die Dichte an dessen Position. Es sei angemerkt, dass das Partikel, für das die Berechnung durchgeführt wird, selbst auch noch mit einfließt. Insgesamt sähe dann ein kompletter SPH Simulations-Zeitschritt so aus:

1. Berechnung der Nachbarschaften aller Partikel (alles mit Abstand $> h$ soll nicht betrachtet werden)
2. Berechnung aller relevanten Werte für die Partikel (also z.B. Dichte, ...)
3. Daraus: Berechnung der wirkenden Kräfte (z.B. Druckkraft, ...)
4. Zeitintegration

Der SPH-Formalismus wird in diesem Schema nur in Punkt 2 und evtl. in Punkt 3 zum Einsatz kommen, um aus den Materialeigenschaften resultierende Werte zu approximieren. Alles andere wird mit anderen Verfahren berechnet.

3.3 Elastizitätstheorie

Da es das Ziel dieser Arbeit ist, elastische Materialien zu modellieren, werden noch einige Anleihen aus der Elastizitätstheorie der Kontinuumsmechanik benötigt. Die Frage die man sich stellen muss, ist: Welche Prozesse finden innerhalb eines elastischen Körpers statt und wie kann man diese modellieren? Da es sich hierbei, wie gesagt, um ein Teilgebiet der Kontinuumsmechanik handelt, sind alle Aussagen, die getroffen werden, für das Kontinuum relevant. Dies können wir uns für die Punktbasierte Simulation zu Nutze machen, indem wir die bestehenden Kontinuumsgleichungen, bei denen es sich meist um Differentialgleichungen handelt, diskretisieren. Dies kann durch Unterteilung des Raums auf mehrere Partikel geschehen.

Während man sich bei der Simulation von Starrkörpern um deren Deformation weniger Gedanken machen muss, stößt man bei elastischen Körpern durch die vielschichtigen internen Prozesse auf größere Schwierigkeiten. Zieht man beispielsweise einen Gummiblock in einer Richtung auseinander, wird er sich in der Regel in Richtung der beiden anderen Raumdimensionen zusammenziehen. Dies ist nur eines der Phänomene, die modelliert werden müssen. Die für diese Arbeit relevanten Verhaltensweisen elastischer Körper sollen nun im Folgenden geklärt werden.

3.3.1 Spannung

Wirkt man auf einen elastischen Körper von außen her eine Kraft aus, wird diese zum Teil in das innere des Körpers abgeleitet und führt zu internen Spannungen. Zieht man etwa an einem Kaugummi, wird durch die wirkenden Kräfte nicht nur die Fläche, an der gezogen wird, ihre Position ändern, sondern auch Teile des Kaugummis, welche nicht in direktem Kontakt mit dieser Kraft stehen. Anhand von Abbildung 3.4 sei eine dieser Kräfte, die Normalkraft visualisiert.

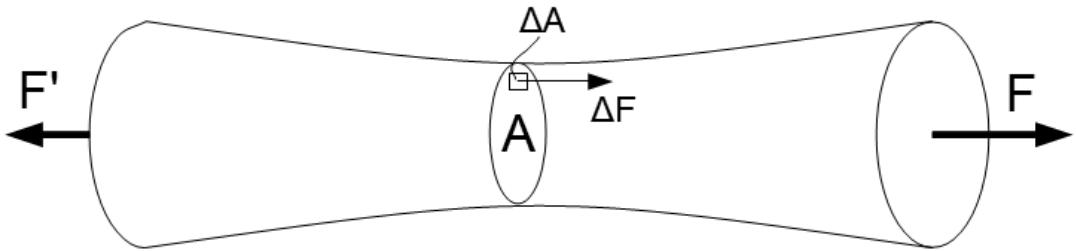


Abbildung 3.4: Kraft, welche in Richtung der Normale auf einen Zylinderschnitt wirkt (nach [Slao2]).

Der Zylinder wird mit Hilfe der Kräfte \mathbf{F} und \mathbf{F}' an den Grundflächen auseinander gezogen. Ein Teil dieser Kraft überträgt sich auch ins Innere des Zylinders auf die Fläche A . Wiederum ein Teil dieser Kraft, namentlich $\Delta\mathbf{F}$ überträgt sich auf eine Teilfläche von A , also ΔA . Die Normalenspannung σ an einem Punkt ist dann wie folgt definiert [Slao2]:

$$\sigma = \lim_{\Delta A \rightarrow 0} \frac{\Delta\mathbf{F}}{\Delta A} \quad (3.14)$$

Gewissermaßen kann also die Normalenspannung als Gegenteil des physikalischen Drucks gesehen werden, nur die Kraft wirkt in die entgegengesetzte Richtung. Sie wird offensichtlich Normalenspannung genannt, weil die Kraft, über die sie definiert ist, in Richtung der Ebenennormalen wirkt. Aus den Normalenspannungen aller Punkte in Fläche A kann dann noch die durchschnittliche Spannung für den Zylinderschnitt A berechnet werden [Slao2]:

$$\sigma_{avg} = \frac{1}{A} \int_A \sigma dA = \frac{|\mathbf{F}|}{A} \quad (3.15)$$

Neben der Normalenspannung existiert jedoch noch eine andere Art von Spannung, die so genannte Schubspannung. Sie wirkt nicht entlang der Normalen, sondern entlang der Ebenentangenten. So etwas ist auch bei dem Beispiel des Kaugummis beobachtbar. Zieht man ihn entlang einer Raumachse auseinander, wird er sich entlang der anderen Raumachsen zusammenziehen. Die Spannungen, die für dieses Zusammenziehen verantwortlich sind, sind die Schubspannungen. Sie sind analog zu der Normalenspannung definiert. Sei \mathbf{F}_x die Normalkraft in Richtung der x -Achse an einem Punkt, dann bezeichnet \mathbf{F}_{xy} die Kraft, die sich, verursacht durch die Materialeigenschaften, von der x -Achse auf die y -Achse überträgt. Aus diesen Kräften \mathbf{F}_x , \mathbf{F}_{xy} und \mathbf{F}_{xz} lassen sich dann die Normalenspannung σ_x und die Schubspannungen τ_{xy} und τ_{xz} berechnen:

$$\sigma_x = \lim_{\Delta A \rightarrow 0} \frac{\Delta\mathbf{F}_x}{\Delta A}, \quad \tau_{xy} = \lim_{\Delta A \rightarrow 0} \frac{\Delta\mathbf{F}_{xy}}{\Delta A}, \quad \tau_{xz} = \lim_{\Delta A \rightarrow 0} \frac{\Delta\mathbf{F}_{xz}}{\Delta A} \quad (3.16)$$

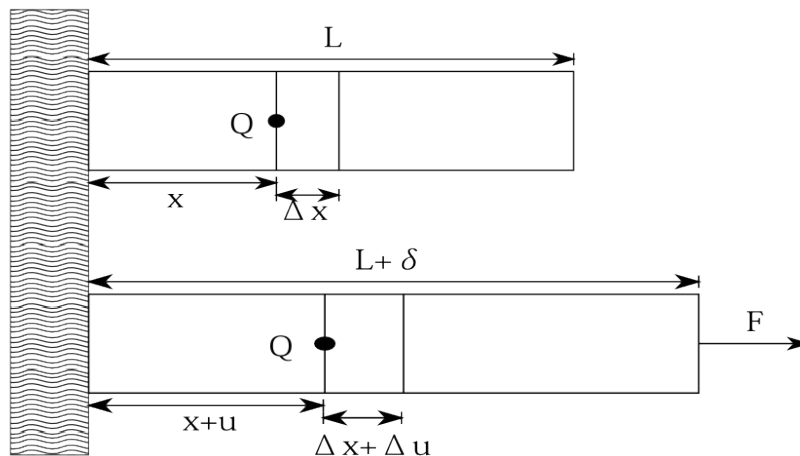


Abbildung 3.5: Longitudinale Verzerrung eines Quaders (nach [Slao2]).

Insgesamt existieren im dreidimensionalen Raum pro Raumpunkt drei unterschiedliche Normalspannungen und sechs unterschiedliche Schubspannungen. Diese Spannungen können im Spannungstensor, der genau für einen Punkt im Kontinuum Gültigkeit besitzt, angeordnet werden:

$$\left[\sigma \right] = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix} \quad (3.17)$$

In diesem sind alle möglichen Spannungen in den drei Raumdimensionen abgetragen.

3.3.2 Verzerrung

Neben den Spannungen existiert eine weitere wichtige Größe im Bezug auf elastische Körper. Diese ist die so genannte Verzerrung ϵ . Sie entsteht durch die in der Größe der Spannung σ enthaltenen Kräfte. Diese Kräfte ziehen an einem bestimmten Punkt im Objekt und verursachen somit, dass sich seine Position ändert. Abbildung 3.5 zeigt die Verzerrung eines Quaders (die dritte Dimension wird nicht abgetragen). Vor der Auswirkung der Kraft F besitzt es Länge L , danach Länge $L + \delta$. Betrachtet man nun einen Teilabschnitt der Länge Δx , besitzt dieser nach der Auswirkung der Kraft die Länge $\Delta x + \Delta u$. Dieser Teilabschnitt hat sich also durch die Auswirkungen der Kraft um Δu gedehnt. Mit Hilfe dieser Größen kann jetzt die Definition der Verzerrung ϵ geschehen [Slao2]:

$$\epsilon = \lim_{\Delta x \rightarrow 0} \frac{\Delta u}{\Delta x} = \frac{du}{dx} \quad (3.18)$$

Der Wert ϵ bezeichnet dann die Verzerrung an einem bestimmten Punkt Q . Ähnlich zur Spannung σ kann auch hier eine Übertragung der Verzerrung in andere Raumrichtungen stattfinden, je nachdem wie das Material beschaffen ist.

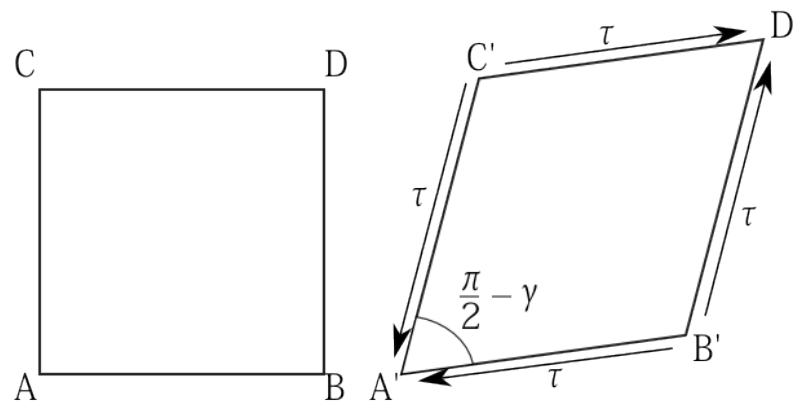


Abbildung 3.6: Orthogonale Verzerrung eines Quaders (nach [Slao2]).

Betrachten wir Abbildung 3.6. Der Winkel $\angle BAC$ beträgt initial $90^\circ = 0,5\pi$. Nach einer Verformung durch eine Schubspannung τ ändert sich dieser Winkel. Anders als die longitudinale Verzerrung ϵ wird die orthogonale Verzerrung γ mit Hilfe dieses Winkels bestimmt:

$$\gamma = \frac{\pi}{2} - \angle B'A'C' \quad (3.19)$$

Ebenfalls analog zur Spannung können auch die Verzerrungen als Tensor geschrieben werden:

$$[\epsilon] = \begin{bmatrix} \epsilon_x & \gamma_{xy} & \gamma_{xz} \\ \gamma_{yx} & \epsilon_y & \gamma_{yz} \\ \gamma_{zx} & \gamma_{zy} & \epsilon_z \end{bmatrix} \quad (3.20)$$

Diese Tensorschreibweisen werden später noch wichtig werden, um das allgemeine Hookesche Gesetz zu formulieren (siehe Abschnitt 3.3.5).

3.3.3 Hookesches Gesetz

Aufgrund der Definitionen von Spannung und Verzerrung erscheint es logisch, dass zwischen diesen beiden Größen irgendeine Beziehung bestehen muss. Besteht eine Spannung an einem Punkt des Objekts, versucht dieses, diese durch Deformation, also Verzerrung, abzubauen. Aufgrund der Bewegungs- und Kraftgesetze, die teilweise auch in Abschnitt 3.1.3 beschrieben werden, erscheint ein linearer Zusammenhang korrekt. Dieser konnte auch experimentell bestätigt werden. Also kann man eine Gesetzmäßigkeit formulieren [Slao2]:

$$\sigma = E\epsilon \quad (3.21)$$

Dieses Gesetz ist als das Hookesche Gesetz bekannt. Diese Beziehung gilt ausschließlich zwischen der Normalenspannung und der longitudinalen Verzerrung. E ist ein Proportionalitätsfaktor. Unter zu großen Kräften jedoch kann selbst ein elastisches Material dieses Gesetz nicht mehr erfüllen, da jedes Material irgendwann reißt oder ausleiert.

3.3.4 Elastizitätsmodul und Poissonzahl

Nach der Formulierung der allgemeinen Gesetzmäßigkeiten bleibt noch die Frage offen, worin sich verschiedene Materialien in ihrem Elastizitätsverhalten unterscheiden. Wir sind also auf der Suche nach Materialkonstanten, die für jeden Werkstoff spezifisch sind. Eine relevante ist der im Hookeschen Gesetz (Gleichung 3.21) vorkommende Proportionalitätsfaktor E . Dieser ist auch als Elastizitätsmodul oder Youngsches Modul bekannt. Das Elastizitätsmodul ist ein Maß für die Steifigkeit eines Materials, wenn auch nicht der einzige Faktor, der diese bestimmt. Je steifer ein Material ist, desto höher fällt E aus. Die Einheit des Elastizitätsmoduls ist Pascal. Beispielsweise hat Gummi ein Elastizitätsmodul von ca. $0,01 - 0,1 GPa$, je nach genauer Zusammensetzung [The12].

Zusätzlich zum Elastizitätsmodul existiert noch eine weitere grundlegende Größe. Weiter oben wurde schon erwähnt, dass sich Spannungen und Verzerrungen in einer Richtung in andere Raumrichtungen fortpflanzen können. In welcher Stärke sie das tun, ist abhängig vom spezifischen Material. Daher definiert man die so genannte Poissonzahl ν (engl. Poisson's ratio). Sie ist gegeben als Verhältnis von axialer ϵ_x zu dazu orthogonaler (lateraler) Verzerrung $\epsilon_y = \epsilon_z$ [Slao2].

$$\nu = \left| \frac{\text{laterale Verzerrung}}{\text{axiale Verzerrung}} \right| = -\frac{\epsilon_y}{\epsilon_x} = -\frac{\epsilon_z}{\epsilon_x} \quad (3.22)$$

Für Materialien mit $\nu \neq 0$ gilt also, dass sich Verzerrungen in einer Raumdimension gleichmäßig in andere Raumdimensionen fortpflanzen. Wie weiter oben schon beschrieben wird sich also ein Gummiblock, der auseinandergezogen wird, in den anderen Raumrichtungen zusammenziehen. Dieser hat folglich eine Poissonzahl $\nu > 0$. Bei den meisten Materialien sind für ν Werte zwischen 0 und 0,5 bestimmt worden. Ein Wert von $\nu = 0,5$ bedeutet hierbei, dass sich eine Verzerrung voll in die anderen beiden Raumdimensionen überträgt. Bei $\nu = 0$ findet keinerlei Übertragung statt. Materialien mit $\nu > 0,5$ existieren auch, jedoch zählen diese nicht zu den elastischen Stoffen. Auch wenn dies etwas kontraintuitiv wirken mag: Es sind auch Materialien mit $-1 \leq \nu < 0$ bekannt. Werden sie in einer Dimension zusammengedrückt, ziehen sie sich in den anderen Dimensionen ebenfalls zusammen [Kölo4].

3.3.5 Verallgemeinerung des Hookeschen Gesetzes

Mit der Kenntnis der unterschiedlichen Materialeigenschaften fällt bald auf, dass das Hookesche Gesetz $\sigma = E\epsilon$ (siehe auch Abschnitt 3.3.3) längst nicht immer Gültigkeit besitzt. Es ist nur bei Materialien mit $\nu = 0$ anwendbar. Daher wurde eine Gesetzmäßigkeit formuliert, die als verallgemeinertes Hookesches Gesetz bekannt ist [Slao2]. Es lautet:

$$\begin{bmatrix} \sigma \end{bmatrix} = \begin{bmatrix} C \end{bmatrix} \begin{bmatrix} \epsilon \end{bmatrix} \quad (3.23)$$

Der Unterschied zum einfachen Hookeschen Gesetz besteht darin, dass es sich bei den Werten für Verzerrung und Spannung um die kompletten Tensoren handelt und nicht nur um einzelne Werte. Zusätzlich wird das Elastizitätsmodul E durch den Steifigkeitstensor C

ersetzt. Bei σ und ϵ handelt es sich um Tensoren zweiter Stufe, also um normale Matrizen. Bei C handelt es sich um einen Tensor vierter Stufe. Mit einsteinscher Summenkonvention kann die Gleichung also auch als $\sigma_{ij} = C_{ijkl}\epsilon_{kl}$ geschrieben werden. Das sieht auf den ersten Blick sehr schwer handhabbar aus, da allein C 81 Komponenten besitzt.

Für die Zwecke dieser Arbeit können wir uns auf die so genannten isotropen Materialien beschränken, was die Gleichung vereinfacht. Isotrope Materialien sind Stoffe die ein drehungsinvariantes Verhalten aufweisen [Sla02]. Stoffe wie Holz sind das beispielsweise nicht, weil die Fasern in eine bestimmte Richtung laufen und dies somit verhindern. Dadurch, dass sich Stoffe gedreht gleich Verhalten, gilt für σ also:

$$\tau_{xy} = \tau_{yx}, \quad \tau_{xz} = \tau_{zx}, \quad \tau_{yz} = \tau_{zy} \quad (3.24)$$

Entsprechend hat auch ϵ ebenfalls nur noch sechs unabhängige Komponenten. C vereinfacht sich durch die Symmetrie auf 36 unabhängige Komponenten. Nun macht man sich die so genannte Voigtsche Notation für Tensoren zu Nutze, mit der man symmetrische Tensoren als Tensoren niedrigerer Ordnung als ihrer ursprünglichen auffassen kann [Voig10]. Somit werden aus σ und ϵ Vektoren der Länge 6, aus C wird eine 6x6-Matrix. Mit Voigtscher Notation sieht die Gleichung wie folgt aus:

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{yz} \\ \tau_{xz} \\ \tau_{xy} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{bmatrix} * \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{yz} \\ \gamma_{xz} \\ \gamma_{xy} \end{bmatrix} \quad (3.25)$$

Setzt man dann die Bedingungen für Isotrope Materialien ein, erhält C bestimmte Koeffizienten und die Gleichung erhält folgende Gestalt:

$$\begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{yz} \\ \tau_{xz} \\ \tau_{xy} \end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} \end{bmatrix} * \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ 2\gamma_{yz} \\ 2\gamma_{xz} \\ 2\gamma_{xy} \end{bmatrix} \quad (3.26)$$

Der Vorfaktor von 2 bei den γ -Werten sorgt dafür, dass das Skalarprodukt aus Spannung und Verzerrung gleich dem inneren Tensorprodukt ist. Dies ist nötig, damit die Kräfteberechnung für die in σ verborgenen Kräfte korrekt verläuft. Man beachte, dass wenn man für ν den Wert 0 einsetzt, sich C auf die Identitätsmatrix mit dem Elastizitätsmodul E als Vorfaktor vereinfacht. Somit fällt die Gleichung auf die aus Abschnitt 3.3.3 bekannte Formel $\sigma = E\epsilon$ zurück. Speziell Gleichung 3.26 wird in späteren Kapiteln noch von großer Wichtigkeit sein, da unter anderem mit ihrer Hilfe die wirkenden elastischen Kräfte bestimmt werden können.

Dies waren soweit die für diese Arbeit nötigen Grundlagen der Elastizitätstheorie. Diese kann hier natürlich nicht vollständig behandelt werden. Der interessierte Leser sei daher auf „The Linearized Theory of Elasticity“ von William S. Slaughter verwiesen, welches weite Teile des Themengebiets abdeckt [Sla02].

4 Modell

Da nun die nötigen Grundlagen geklärt wurden, wollen wir klären, wie wir das elastische Verhalten von Materialien mit Hilfe einer punktbasierten Simulation modellieren können, bei der dann bestimmte Werte mit der SPH-Methode approximiert werden. Um das alles zu beschreiben, werden in Abschnitt 4.1 zunächst die Annahmen geklärt, die gemacht werden müssen, um ein Modell zu erhalten, das einfach genug ist, um es effizient berechnen zu können. Abschnitt 4.2 klärt, wie SPH in unserem Modell verwendet wird, um bestimmte nötige Werte zu berechnen. In den Abschnitten 4.3 und 4.4 wird dann herausgestellt, wie die Verzerrungen und Spannungen berechnet werden und aus diesen und den anderen Werten dann die elastische Kraft bestimmt werden kann, die zwischen objektinternen Partikeln wirkt. In Sektion 4.5 wird abschließend noch das verwendete Zeitintegrationsverfahren beschrieben.

Das Modell stammt zum Hauptteil aus der Arbeit von Solenthaler und Kollegen, die schon weiter oben erwähnt wurde ([SSP07]). Jedoch werden auch Anleihen bei der Arbeit von Becker und Kollegen gemacht, die die eben genannte Arbeit um bestimmte Aspekte erweitern ([BIT09]).

4.1 Grundannahmen

Wie bei jedem Modell ist es notwendig, im Voraus bestimmte Annahmen zu treffen. So wollen wir uns auf isotrope Materialien beschränken, um die Berechnung von Spannung bzw. Verzerrung zu vereinfachen, wie in 3.3.5 beschrieben. Natürlich wäre es möglich, die Spannungsberechnung mit dem vollbesetzten Steifigkeitstensor C durchzuführen, jedoch sehr aufwändig.

4.1.1 Lokale Undeformiertheit

Eine weitere, sehr wichtige Annahme ist die der lokalen Undeformiertheit. In Arbeiten wie beispielsweise [KAG⁺05] wird zur Berechnung der Verzerrung in einem Körper mit einem globalen Referenzobjekt gearbeitet. Dieses Referenzobjekt repräsentiert die undeformierte Variante des zu simulierenden Körpers. Bei der Verzerrungsberechnung wird dann der aktuelle Zustand des Objekts mit dem Referenzobjekt verglichen, um die aktuelle Verzerrung zu bestimmen. Im Kontrast dazu wird hier ein anderes Modell verwendet. Dies ist nötig, weil es auch dazu entwickelt wurde, Aggregatzustandsübergänge abzubilden. Dies ist mit einem globalen Referenzobjekt sehr schwer umsetzbar [SSP07].

Daher speichert jedes Partikel initial einen Distanzvektor zu seinen Nachbarn. Seine Nachbarschaft ist definiert als alle Partikel, die sich im Radius h um das zu betrachtende Partikel befinden. Der Wert h ist die Glättungslänge, die auch für die SPH-Berechnungen verwendet wird. Da alle Partikel außerhalb der Glättungslänge keinen Einfluss auf das betrachtete Partikel haben, erscheint diese Einschränkung der gespeicherten Vektoren plausibel [SSP07]. Während der Simulation kann dann der aktuelle Distanzvektor bestimmt werden. Weicht dieser vom gespeicherten Distanzvektor ab, ist eine Verzerrung aufgetreten. Die Position, die ein Partikel beim Start der Simulation einnimmt und die für die Berechnung der Nachbarschaften herangezogen wird, wird als die Referenzposition \mathbf{x}^0 bezeichnet. Für Partikel P_i ist die Referenzposition entsprechend \mathbf{x}_i^0 .

4.2 Verwendung von SPH im Elastizitätsmodell

Im Folgenden soll geklärt werden, bei der Berechnung welcher Größen im Modell SPH verwendet wird. Wie beispielsweise die Spannung in einem Objekt berechnet werden kann wurde schon im Grundlagen-Kapitel geklärt, SPH wäre hier von keinem großen Nutzen. Es sind aber noch einige Werte übrig, die für die Berechnung der elastischen Kraft relevant sind.

4.2.1 Partikelvolumen

Wie in den meisten punktbasierenden Modellen besitzt auch hier jedes Partikel eine Masse. Diese ist vor allem für die Beschleunigungsberechnung mit Hilfe der Formel $F = m \cdot a$ vonnöten. Genauso wie ein Partikel einen Teil der Masse eines Objekts repräsentieren kann, kann es auch einen Teil des Volumens repräsentieren. Um dieses zu berechnen, greift man zunächst auf die einfache Formel $v = \frac{m}{\rho}$ zurück. Volumen entspricht Masse geteilt durch Dichte. Wie die Dichte mittels SPH berechnet werden kann, wurde schon in Abschnitt 3.2.2 gezeigt. Die Gleichung, um das Volumen eines Partikels P_i zu berechnen lautet wie folgt:

$$\bar{v}_i = \bar{v}(\mathbf{x}_i) = \frac{m_i}{\rho_i} = \frac{m_i}{\sum_j m_j W(|\mathbf{x}_j^0 - \mathbf{x}_i^0|, h)} \quad (4.1)$$

Da sich Deformationen von Objekten häufig auch in Volumenänderungen niederschlagen, ist ein bekanntes Partikelvolumen für die Berechnung einiger Größen von Vorteil. Es sei noch angemerkt, dass die P_j über welche summiert wird, alle Teil desselben Objekts sind, wie P_i . Der Wert \bar{v}_i ist für die Berechnung der elastischen Kraft von Relevanz und diese tritt nur innerhalb eines Objektes auf.

4.2.2 Verschiebungsfeld

Um später Verzerrungen und Spannungen errechnen zu können, benötigt man ein Maß, inwieweit das Objekt von seiner ursprünglichen Form abweicht. Diese Abweichung wird

auch Verschiebung (engl. displacement) genannt. Die Gesamtverschiebung ist jedoch nicht wirklich nützlich um den aktuellen Zustand im Objekt, also auch die aktuell wirkenden Kräfte zu berechnen. Wichtiger ist, inwieweit sich die Verschiebung im Moment ändert. Gefragt ist also der Gradient der Verschiebung von etwa Partikel P_i , welcher mit ∇u_i bezeichnet wird. Um ∇u_i zu berechnen, wird wieder einmal die SPH-Methode verwendet [SSPo7]:

$$\nabla u_i = \sum_j \bar{v}_j \nabla W(|\mathbf{x}_j - \mathbf{x}_i|, h) (\mathbf{u}_{ji})^T \quad (4.2)$$

Ebenso wie beim Partikelvolumen wird hier nur über die Partikel desselben Objekts summiert. Der Vektor \mathbf{u}_{ji} wird der Verschiebungsdifferenzvektor (engl. displacement difference vector) genannt. Er ist wie folgt definiert:

$$\mathbf{u}_{ji} = \mathbf{x}_j - \mathbf{x}_i - (\mathbf{x}_j^0 - \mathbf{x}_i^0) \quad (4.3)$$

Er bezeichnet, wie auch der Name suggeriert, die Differenz der Verschiebungen der Partikel P_i und P_j . Dies sei an Abbildung 4.1 gezeigt.

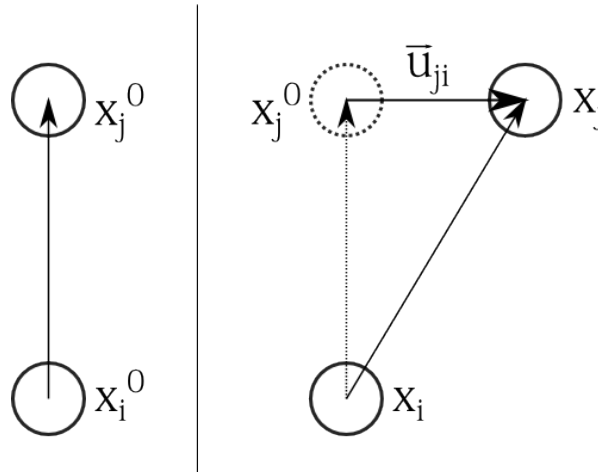


Abbildung 4.1: Beispiel für einen Verschiebungsdifferenzvektor

Links ist die Startkonfiguration zum Zeitpunkt $t = 0$ abgetragen. Der Vektor zwischen den beiden Partikeln ist somit einer der initial berechneten Distanzvektoren, die weiter oben beschrieben wurden. Nach einigen Zeitschritten befindet sich P_j an einer anderen Position relativ zu P_i . Der gesuchte Verschiebungsdifferenzvektor ist dann der Vektor, welcher sich von der Position, an der sich P_j laut der Bedingung für die lokale Undeformiertheit befinden müsste, zu der eigentlichen Position von P_j aufspannt.

Die Gleichung für ∇u_i macht sich zusätzliche eine spezielle Eigenschaft von SPH zu Nutze. Der Gradient einer bestimmten Größe kann mittels SPH bestimmt werden, indem man zunächst die Gleichung für die Größe selbst aufstellt. Dann kann einfach die Kernel-Funktion durch ihren Gradienten ersetzt werden und schon erhält man die Gleichung für den Gradienten der gewollten Größe [Mono5b].

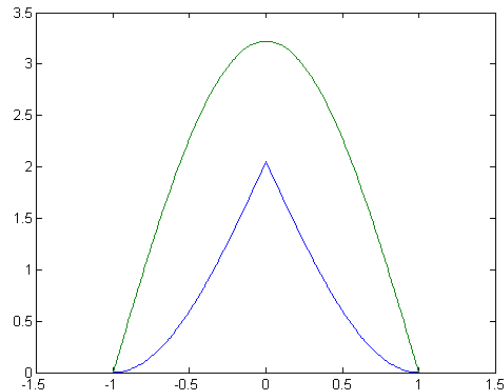


Abbildung 4.2: Elastizitätskernel (niedrige Kurve) und sein Gradient (hohe Kurve) für $h = 1$.

Der Gradient des Verschiebungsfeldes ∇u_i enthält alle partiellen Ableitungen für die Verschiebung des Punktes P_i . Bei ihm handelt es sich um eine 3×3 -Matrix.

4.2.3 Kernel-Funktion

Bis jetzt wurde noch nicht geklärt, von welcher Gestalt die Kernel-Funktion sein soll, um ein möglich realistisches elastisches Verhalten zu gewährleisten. Solenthaler und Kollegen entwickelten speziell für elastische Objekte den so genannten Elastizitätskernel (engl. elasticity kernel). Dieser ist wie folgt definiert [SSP07]:

$$W(r, h) = \begin{cases} c \frac{2h}{\pi} \cos\left(\frac{(r+h)\pi}{2h}\right) + c \frac{2h}{\pi} & , \text{wenn } 0 \leq r \leq h \\ 0 & , \text{ansonsten} \end{cases} \quad (4.4)$$

Um die Bedingung zu gewährleisten, dass die Fläche unter der Kurve gleich 1 ist, ist c so bestimmt:

$$c = \frac{\pi}{8h^4 \left(\frac{\pi}{3} - \frac{8}{\pi} + \frac{16}{\pi^2} \right)} \quad (4.5)$$

Der Graph des Kernels und seines Gradienten für Glättungslänge $h = 1$ ist in Abbildung 4.2 abgetragen. Er ist nicht ganz so spitz wie beispielsweise der Spiky-Kernel [DC96]. Das heißt, dass Partikel, die sich weiter weg befinden, stärker gewichtet werden.

4.3 Verzerrung und Spannung

Um letztendlich die elastischen Kräfte berechnen zu können, müssen erst die Spannungen und Verzerrungen innerhalb eines Objekts bekannt sein. Wie Spannungen und Verzerrungen zueinander in Verhältnis stehen wurde bereits in Abschnitt 3.3 erläutert. Jedoch ist noch

nicht klar, wie man an eine der beiden Größen kommt, um mit ihr die andere zu berechnen. Dabei trifft es sich gut, dass aus dem Gradient des Verschiebungsfeldes ∇u_i sehr leicht der Verzerrungstensor ϵ berechnen lässt. Dazu existiert jedoch mehr als eine Möglichkeit. Hier wird der so genannte *Green-Saint-Venant* Verzerrungstensor verwendet. Dieser ist so definiert [NMK⁺06]:

$$\epsilon = \frac{1}{2}(\nabla u + \nabla u^T + \nabla u^T \nabla u) \quad (4.6)$$

In der Arbeit [BIT09] wird auch demonstriert, wie sich das Verfahren verhält, wenn man einen anderen Verzerrungstensor wie den *Cauchy-Green* Tensor verwendet. Bei diesem handelt es sich um einen linearen Verzerrungstensor. Bei starken Deformationen können lineare Verzerrungstensoren ungenaue Ergebnisse liefern, daher wird ein nichtlinearer gewählt. Die Definition des *Green-Saint-Venant* Tensors variiert an einigen Punkten. So wird in [MKN⁺04] der Faktor $\frac{1}{2}$ weggelassen. Wir jedoch legen uns auf obige Definition fest.

Da ϵ nun bekannt ist, kann die Berechnung der Spannung wie in Abschnitt 3.3.5 erfolgen. Insbesondere kann Gleichung 3.26 verwendet werden, da wir uns auf isotrope Materialien beschränken.

4.4 Elastische Kraft

Die Berechnung der Kräfte erfolgt über die Verzerrungsenergiedichte (engl. strain energy density) U . Es gilt:

$$U_i = \bar{v}_i \frac{1}{2} (\epsilon_i * \sigma_i) \quad (4.7)$$

An einem bestimmten Punkt P_i entspricht die wirkende Kraft dem negativen Gradienten der Verzerrungsenergiedichte. Hierbei wird noch die Verschiebung des Punktes berücksichtigt [MKN⁺04]. Folglich kann auch die elastische Kraft berechnet werden, die ein Partikel P_i auf P_j auswirkt. Diese Kraft sei mit $\mathbf{F}_{ji}^{elastic}$ bezeichnet.

$$\mathbf{F}_{ji}^{elastic} = -\nabla_{u_j} U_i = -2\bar{v}_i (\mathbb{I} + \nabla u_i^T) \sigma_i \mathbf{d}_{ij} \quad (4.8)$$

Hierbei ist \mathbb{I} die Identitätsmatrix, bei \mathbf{d}_{ij} handelt es sich um den nach der Verschiebung des j -ten Partikels abgeleiteten Gradienten des Verschiebungsfeldes ∇u_i . Dieser ist wie folgt definiert. Diese Definition gilt jedoch nur für $i \neq j$:

$$\mathbf{d}_{ij} = \bar{v}_j \nabla W(\left| \mathbf{x}_j^0 - \mathbf{x}_i^0 \right|, h) \quad (4.9)$$

Anders als bei anderen Modellen üblich, wird hier nicht die Kraft vom j -ten auf das i -te Partikel angegeben, sondern umgekehrt.

4.5 Zeitintegration

Um letzten Endes die neuen Partikelpositionen für den aktuellen Zeitschritt berechnen zu können, benötigt man ein Zeitintegrationsverfahren. Hier wird jedoch nicht, wie in Abschnitt 3.1.3 beschrieben, das explizite Euler-Verfahren verwendet. Es kommt das so genannte Leapfrog-Verfahren zur Anwendung [AP98]. Hierbei handelt es sich um ein Verfahren zweiter Ordnung, welche im Allgemeinen genauer sind als Verfahren erster Ordnung. Beim Leapfrog-Verfahren wird dies dadurch erreicht, dass Geschwindigkeiten und Positionen zeitversetzt berechnet werden. Sei δt die Länge eines Zeitschritts und x_i die Position eines Partikels zum Zeitschritt i (entsprechend für die Geschwindigkeit v und Kraft F).

$$\begin{aligned} \mathbf{x}_{n+1} &= \mathbf{x}_n + \delta t * \mathbf{v}_{n+\frac{1}{2}} \\ \mathbf{v}_{n+\frac{3}{2}} &= \mathbf{v}_{n+\frac{1}{2}} + \delta t * F_{n+1} \end{aligned} \tag{4.10}$$

Man sieht, dass die Geschwindigkeiten gewissermaßen in einem „Zwischenzeitschritt“ berechnet werden. Der einzige Wert, der auf diese Weise nicht berechnet werden kann, ist $\mathbf{v}_{\frac{1}{2}}$, da $\mathbf{v}_{-\frac{1}{2}}$ nicht existiert, weil es zu Simulationsstart schon in der Vergangenheit liegt. Man behilft sich hier, indem man initial einen halben Zeitschritt mit Hilfe des Expliziten Euler-Verfahrens bestimmt.

5 Implementierung

Dieses Kapitel soll Details zur Implementierung des Modells aus [SSP07] erläutern. In Abschnitt 5.1 wird die Simulationsumgebung präsentiert, in die dieses Verfahren im Rahmen dieser Arbeit integriert wurde. Abschnitt 5.2 erläutert den Ablauf der Simulation selbst, insbesondere den eines einzelnen Zeitschritts. Abschnitt 5.3 schlussendlich zeigt einige Details bei der Implementierung auf. Es werden Teile des Modells erläutert, bei der die tatsächliche Implementierung von den in Kapitel 4 aufgestellten Gleichungen abweichen.

5.1 Umgebung

Das Verfahren von Solenthaler und Kollegen wurde im Rahmen dieser Arbeit in eine schon bestehende Simulationsumgebung integriert bzw. diese erweitert. Diese ist in C++ implementiert und verwendet Qt zur Darstellung der Benutzeroberfläche. Zur Visualisierung der Simulation selbst ist ein OpenGL-Fenster in die Oberfläche integriert. Abbildung 5.1

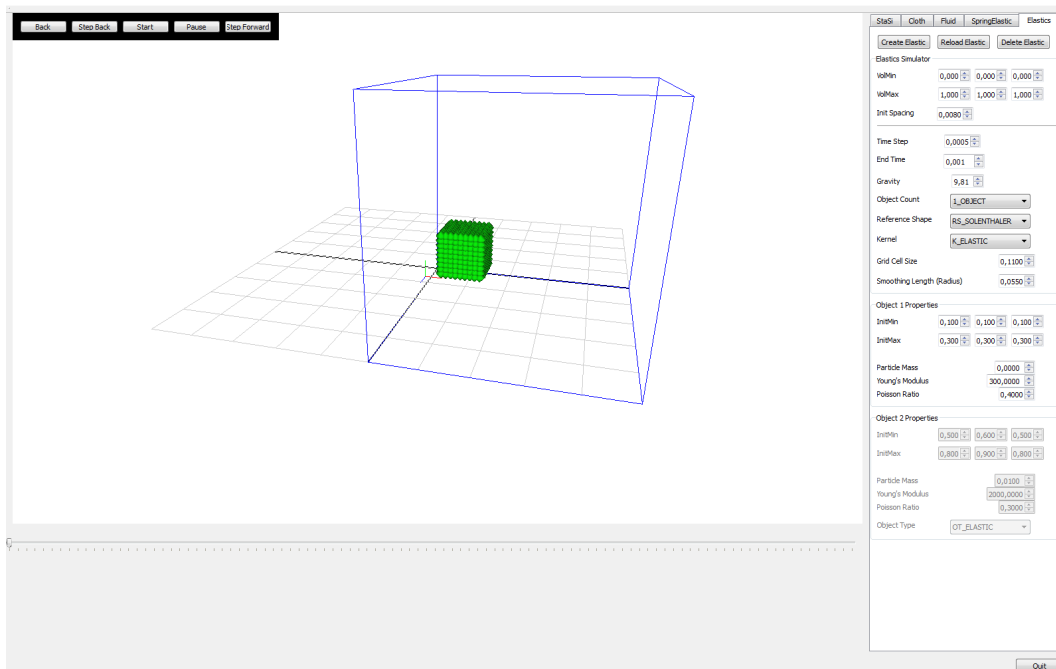


Abbildung 5.1: Simulationsumgebung, in die das Verfahren integriert wurde

Beschreibung	Symbol	Typische Werte
Gravitationskonstante	g	9,81
Länge eines Zeitschritts	δt	0,001
Simulationsdauer	t_{max}	2,00
Glättungslänge	h	0,01 – 0,1
Partikelmasse	m	0,00001
Elastizitätsmodul	E	50 – 10000
Poissonzahl	ν	0,0 – 0,5
Objektanmessungen u. Position	$\mathbf{x}_{min}, \mathbf{x}_{max}$	beliebig, solange im Raum
Partikelabstand	δx	0.025
Größe des zu simulierenden Raums	$\mathbf{V}_{min}, \mathbf{V}_{max}$	beliebig, meist Einheitswürfel

Tabelle 5.1: Simulationsparameter

zeigt die Oberfläche der Simulationsumgebung. Links ist das *OpenGL*-Fenster zu erkennen. Rechts die einstellbaren Simulationsparameter. Welche genau dies sind, sei in Abschnitt 5.2.1 geklärt. Bei dem grünen Würfel im *OpenGL*-Fenster handelt es sich um den zu simulierenden Partikelsatz. Zusätzlich zu der Benutzeroberfläche besitzt die Umgebung auch noch eine Konsolenausgabe, die genauere Details der Simulation ausgibt, wie z.B. berechnete Zwischenergebnisse oder den aktuellen Zeitschritt. Zusätzlich zu der eigentlichen Simulation wurde in die Anwendung noch ein Oberflächenrekonstruktionsverfahren integriert, um aus den berechneten Punktdaten eine Polygongraphik extrahieren zu können. Bei diesem Verfahren handelt es sich um das von Bhattacharya und Kollegen ([BGB11]). Dieses wurde jedoch nicht von Hand implementiert, es wurde lediglich der von den Entwicklern der Methode bereitgestellte Code in die Umgebung integriert.

5.2 Simulationsablauf

Dieser Abschnitt soll den Ablauf der Simulation erläutern. In 5.2.1 wird zunächst auf die Initialisierung der Simulation eingegangen um dann mit dem eigentlichen Ablauf eines Zeitschritts in 5.2.2 fortzufahren.

5.2.1 Initialisierung

Zur Durchführung einer Simulation sind natürlich einige Simulationsparameter notwendig, die im Vorhinein bestimmt werden müssen. Die wichtigsten sind in Tabelle 5.1 aufgeführt. Hierbei können einige Werte wie die Partikelmasse, die Poissonzahl usw. mehrfach vorhanden sein, für den Fall, dass man mehrere Objekte mit unterschiedlichen Materialparametern zu simulieren wünscht. Die Objektanmessungen und die Raumgröße werden als je ein paar von Vektoren angegeben. Somit kann man zwischen den beiden Vektoren ein Quader

Algorithmus 5.1 Initialisierung der Simulation

```

procedure INITIALIZE
  READPARAMETERS
  INSERTPARTICLES( $m, \mathbf{x}_{min}, \mathbf{x}_{max}, \delta x$ )
  COMPUTEKERNELVALUES( $h$ )
  COMPUTEREFERENCEDISTANCES
  COMPUTEVOLUMES( $m$ )
  COMPUTEFORCES
  INITVELOCITIES
end procedure

```

aufspannen. Die Kanten befinden sich dann parallel zu den Raumachsen. Natürlich ist die Objektgröße nur relevant, wenn man ein quaderförmiges Objekt wünscht. Andernfalls muss die Geometrie alternativ in die Umgebung geladen werden. Die Raumgröße ist vor allem für Aufprallsimulationen relevant, wenn ein Objekt auf den Boden treffen soll. Der Partikelabstand bestimmt den Abstand der Partikel innerhalb des gegebenen Quaders.

Der komplette Initialisierungsvorgang sei in Algorithmus 5.1 gegeben. Welche Parameter eingelesen werden und wie die Partikel eingefügt werden sollen, wurde schon geklärt. Als nächster Schritt wird die Konstante c der Kernel-Funktion (siehe Abschnitt 4.2.3) vorberechnet. Genauso wie die später folgende Berechnung der Partikelvolumen \bar{v}_i (siehe Abschnitt 4.2.1) kann dies getan werden, weil sich diese Werte im Verlauf der Simulation nicht ändern. Passt man die Partikelvolumen immer dem aktuellen Zustand des Objekts an, wirken überhaupt keine elastischen Kräfte mehr, da dann davon ausgegangen wird, dass sich das Objekt im Ruhezustand befindet.

Zwischen der Berechnung des c 's und der der Volumen müssen noch die Referenzdistanzen zwischen den Partikeln bestimmt werden, also jene, die für die Annahme der lokalen Undeformiertheit vonnöten sind (siehe auch Abschnitt 4.1.1). Zusätzlich wird die lokale Nachbarschaft für jedes Partikel berechnet. Um, besonders bei großen Objekten bzw. großem h , Speicherplatz zu sparen, wird anstatt der Referenzdistanz selbst nur die initiale Position eines jeden Partikels abgespeichert. Aus diesen kann dann, falls benötigt, die Referenzdistanz berechnet werden. Anstatt also bei jedem Partikel für jeden Nachbarn einen Distanzvektor zu speichern, was bei ungünstigen Parametern schnell den Arbeitsspeicher sprengen kann, wird nur ein einzelner Vektor pro Partikel im Speicher gehalten.

Die folgenden beiden Schritte, die Kraft- und Geschwindigkeitsberechnung, sind dem Leapfrog-Schema geschuldet, das zur Zeitintegration verwendet wird. Da sich das Objekt zu diesem Zeitpunkt noch in seiner Ursprungslage befindet, wirken nur externe Kräfte wie die Gravitation. Mit diesen Kräften wird dann die Geschwindigkeit zum Zeitpunkt $t = \frac{1}{2}\delta t$ mittels dem Euler-Verfahren berechnet. Damit ist dann die Initialisierung abgeschlossen.

Algorithmus 5.2 Einzelner Zeitschritt der Simulation

```
procedure TIMESTEP
  COMPUTE NEW PARTICLE POSITIONS
  COMPUTE PARTICLE NEIGHBORHOODS
  COMPUTE DISPLACEMENT FIELD GRADIENTS
  COMPUTE STRAINS
  COMPUTE STRESSES
  COMPUTE FORCES
  COMPUTE VELOCITIES
end procedure
```

5.2.2 Ablauf eines Zeitschritts

Nachdem alle relevanten Werte initialisiert wurden, kann die vorher eingestellte Anzahl an Zeitschritten berechnet werden. Ein kompletter Zeitschritt ist in Algorithmus 5.2 gegeben. Die Berechnung der neuen Partikelpositionen erfolgt gleich zu Beginn, um dem Leapfrog-Schema gerecht zu werden. Nachdem dies geschehen ist, können die aktuellen Nachbarschaften der Partikel errechnet werden. Dies kann entweder einmal pro Zeitschritt geschehen oder aber implizit durch die Anwendung der Kernel-Funktion getan werden. Letztere Möglichkeit würde jedoch dazu führen, dass die Nachbarschaften öfter als einmal pro Zeitschritt errechnet würden. Nachdem all dies getan wurde, kann man beginnen, die notwendigen Werte für die elastischen Kräfte zu berechnen.

Zunächst wird für jedes Partikel der Gradient des Verschiebungsfeldes bestimmt. Daraus kann dann, wie in den Abschnitten 4.3 und 3.3.5 beschrieben, seine Verzerrung und Spannung errechnet werden. Zu Details der Bestimmung des Gradienten des Verschiebungsfeldes sei auf den folgenden Abschnitt verwiesen, da sich hierbei einige Schwierigkeiten ergeben. Nachdem alle nötigen Größen feststehen, können die wirkenden Kräfte errechnet werden. Zu Details hierzu sei ebenfalls auf den nächsten Abschnitt verwiesen. Schlussendlich werden dann aus den Kräften noch die Geschwindigkeiten bestimmt. Einfacher dargestellt werden kann ein Zeitschritt auch durch dieses Schema:

$$\mathbf{x}_t \rightarrow \nabla u_t \rightarrow \epsilon_t \rightarrow \sigma_t \rightarrow F_t \rightarrow \mathbf{v}_{t+\frac{1}{2}\delta t} \quad (5.1)$$

Nachdem dies ausgeführt wurde, kann ein neuer Zeitschritt, beginnend mit der Berechnung von $\mathbf{x}_{t+\delta t}$ durchgeführt werden.

5.3 Besondere Details

Dieser Abschnitt soll auf besondere Problematiken der Implementierung und deren Lösung eingehen, insbesondere auf solche, die nicht direkt aus dem Modell ersichtlich sind.

5.3.1 Gradient des Verschiebungsfeldes

Um den Gradienten des Verschiebungsfeldes ∇u korrekt berechnen zu können, benötigt man zunächst einmal den Gradienten der Kernel-Funktion. Der Übersichtlichkeit halber sei die Berechnungsvorschrift noch einmal erwähnt:

$$\nabla u_i = \sum_j \bar{v}_j \nabla W(|\mathbf{x}_j - \mathbf{x}_i|, h) (\mathbf{u}_{ji})^T \quad (5.2)$$

Da $\mathbf{d}_{ij} = \bar{v}_j \nabla W(|\mathbf{x}_j - \mathbf{x}_i|, h)$ gilt, ist dies auch für die Berechnung der elastischen Kräfte relevant. Da es sich bei der Glättungslänge h quasi um eine Konstante handelt, genügt es, zur Berechnung des Gradienten des Kernels $W(|\mathbf{r}|, h)$ partiell nach \mathbf{r} abzuleiten und das Ergebnis mit dem normalisierten \mathbf{r} zu multiplizieren. Leitet man $W(|\mathbf{r}|, h)$ partiell nach \mathbf{r} ab, erhält man folgende Formel:

$$\nabla W(|\mathbf{r}|, h) = \begin{cases} c * \sin\left(\frac{(r+h)\pi}{2h}\right) & , \text{wenn } 0 \leq r \leq h \\ 0 & , \text{ansonsten} \end{cases} \quad (5.3)$$

Wie beschrieben muss der Rückgabewert dieser Funktion noch mit $\frac{\mathbf{r}}{|\mathbf{r}|}$ multipliziert werden, um den eigentlichen Gradienten zu erhalten. Es sei außerdem angemerkt, dass die obige Formel eigentlich mit -1 multipliziert werden müsste, um korrekt zu sein, jedoch wäre die Gradientenfunktion dann nicht mehr positiv, was jedoch für unsere Zwecke benötigt wird. Wendet man all diese Modifikationen an, erhält man denselben Funktionsgraphen wie in Abbildung 4.2. Dieses Ergebnis stimmt mit dem Graphen aus [SSP07] überein.

5.3.2 Anrechnung der Kräfte

Berechnet man mit Hilfe der oben genannten Formeln alle elastischen Kräfte, fällt auf, dass $\mathbf{F}_{ji}^{elastic} \neq -\mathbf{F}_{ij}^{elastic}$. Dies widerspricht dem in Abschnitt 3.1.2 angeführten dritten newtonschen Gesetz. Der Grund dafür ist, dass bei der Berechnung des \mathbf{d}_{ij} welches in der Berechnungsformel für die elastische Kraft auftritt, bei Kraft und Gegenkraft nicht von derselben Nachbarschaft ausgegangen werden kann. Partikel P_i besitzt in der Regel eine andere Nachbarschaft als P_j . Man behilft sich hier, indem man zwischen Kraft und Gegenkraft mittelt und das Ergebnis dann den Partikeln als eigentliche elastische Kraft anrechnet. Wie die Kräfte insgesamt berechnet werden sei in Algorithmus 5.3 gegeben.

Es wird vermutet, dass der Faktor 2 in Gleichung 4.8, der Berechnung der elastischen Kraft, deshalb eingeführt wurde, um den Faktor $\frac{1}{2}$ in Algorithmus 5.3 auszulöschen. In späteren Arbeiten wie [BIT09] wurde er in der Formulierung der elastischen Kraft weggelassen. Es ist zu erkennen, dass die elastische Kraft nicht für das aktuelle Partikel berechnet wird, sondern für dessen aktuellen Nachbarn. Dies ist der Formulierung der Formel für $\mathbf{F}_{ji}^{elastic}$ geschuldet. Es würde eventuell mehr Aufwand bedeuten, die Kräfte jeweils für das aktuelle Partikel zu berechnen.

Die Gravitationskraft kann natürlich auch weggelassen werden. Dies kann auch implizit erreicht werden, indem man als Eingabeparameter die Gravitationskonstante auf 0 setzt. Das

Algorithmus 5.3 Berechnung der Kräfte

```
procedure COMPUTEFORCES
  for all Particles  $i$  do
     $P_i.Force \leftarrow 0$  // Reset Forces
  end for
  for all Particles  $i$  do
    for all Neighbors  $j$  of Particle  $i$  do
       $P_j.Force \leftarrow P_j.Force + \frac{1}{2}(\mathbf{F}_{ji}^{elastic} - \mathbf{F}_{ij}^{elastic})$  // Elastic Forces
    end for
     $P_i.Force \leftarrow P_i.Force + \begin{pmatrix} 0 \\ -m_i * g \\ 0 \end{pmatrix}$  // Gravity Force
  end for
end procedure
```

Zurücksetzen der Kräfte zu Beginn ist aber in jedem Fall erforderlich, da sonst die Kräfte des letzten Zeitschritts mit einberechnet würden.

5.3.3 Grenzbehandlung

Da die Grenzbehandlung weder im Modell in Abschnitt 4 thematisiert wird, noch in [SSP07] erwähnt wird, wurde hierzu ein simples Verfahren entwickelt, damit Objekte nicht aus dem selbst bestimmten Simulationsraum austreten können. Das ist vor allem für Aufprallsimulationen von Relevanz. Das Verfahren wird in die Prozedur zur Errechnung der neuen Partikelpositionen x_i integriert. Zunächst werden dort die neuen Partikelpositionen berechnet. Stellt sich dann heraus, dass sich das aktuell betrachtete Partikel außerhalb des Simulationsraums befindet, wird es so lange auf der Raumachse verschoben, auf der es ausgetreten war, bis es sich wieder im Raum befindet. Jedoch wird es nicht genau auf der Grenze platziert, um eventuelle Fehler durch die Verwendung von Fließkommazahlen zu vermeiden. Es wird dann ca. 0.00001 von der Stelle platziert, an der es die Raumgrenze durchdrungen hat. Dieser Wert kann natürlich je nach Raumgröße variiert werden. Er sollte so klein gewählt werden, dass er das Partikel nicht zu weit verschiebt.

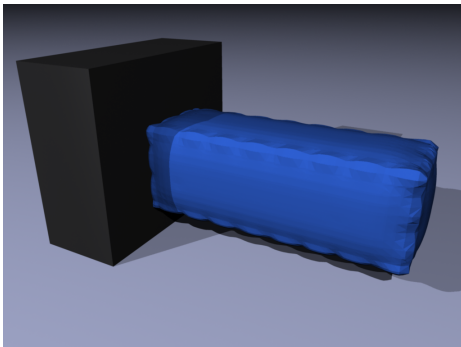
Zusätzlich zur Rückverschiebung des betrachteten Partikels in den Raum hinein muss seine Geschwindigkeit in der Raumrichtung, in der er die Raumgrenze durchdrungen hat, auf 0 gesetzt werden. Geschieht dies nicht, kann das Partikel regelrecht an der Raumgrenze „kleben“. Dies ist vor allem bei elastischen Materialien wichtig, welche beispielsweise auf den Boden fallen und dann durch die elastischen Kräfte wieder in die Luft geschleudert werden. Werden die nach unten gerichteten Geschwindigkeiten der auf den Boden aufprallenden Partikel nicht zurückgesetzt, springt das Objekt nicht nach oben, obwohl es das eigentlich müsste.

6 Ergebnisse

In diesem Kapitel sollen einige Experimente vorgestellt werden, die mit der eigenen Implementierung der Methode von Solenthaler und Kollegen [SSP07] durchgeführt wurden. Vor allem soll dabei ihre Stabilität und Korrektheit überprüft werden. Zunächst wird dabei die Variation der Parameter für elastische Materialien untersucht (Abschnitte 6.1 und 6.2). Danach folgt die Untersuchung der Variation von anderen Simulationsparametern (Abschnitt 6.3) und eine Untersuchung der Laufzeiten der Implementierung (Abschnitt 6.4). Wie die Ergebnisse der Untersuchungen genau zu interpretieren sind, folgt später in Kapitel 7.

6.1 Variation des Elastizitätsmoduls

Die erste der beiden grundlegenden materialabhängigen Variablen isotroper elastischer Körper ist das Elastizitätsmodul E . Wie in Abschnitt 3.3.4 schon beschrieben, ist es ein Maß für die Steifigkeit eines Objekts. Je höher das Elastizitätsmodul, desto schwerer deformierbar ist ein Körper. Ob dies vom präsentierten Modell ebenso abgebildet wird, soll folgender Versuchsaufbau klären.



Beschreibung	Symbol	Gewählter Wert
Gravitationskonstante	g	9,81
Länge eines Zeitschritts	δt	0,001s
Simulationsdauer	t_{max}	0,3
Glättungslänge	h	0,055
Partikelmasse	m	0,00001
Elastizitätsmodul	E	variabel
Poissonzahl	ν	0,0
Partikelabstand	δx	0,025
Anzahl der Partikel	N	2106

Abbildung 6.1: Startkonfiguration und Versuchsparameter Versuch 1

Simuliert wird ein quaderförmiges Objekt mit den Abmessungen von $0,2 * 0,2 * 0,6$ Raumeinheiten, welches aus ca 2100 Partikeln besteht. Dieses wird an einem der beiden quadratischen Enden $0,1$ Einheiten über dem „Boden“ des Simulationsraums fixiert. Der initiale Aufbau soll anhand von Abbildung 6.1 verdeutlicht werden. Das linke Ende des Objekts ist hier fixiert. Nach dem Start der Simulation wird das Objekt von der Gravitation nach unten

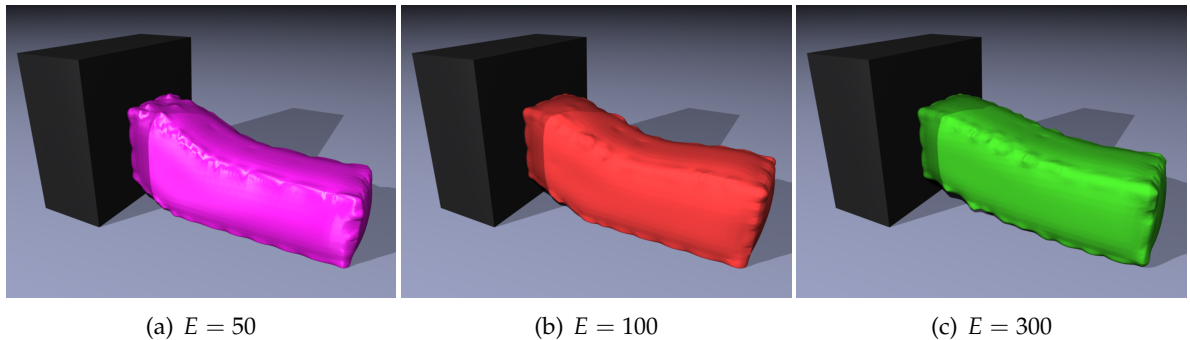


Abbildung 6.2: Ergebnisse Versuch 1 bei $E \in [50, 300]$

gezogen werden. Da die linke Seite fixiert ist, wird dies größeren Effekt auf die rechte Seite haben. Nach einigen Zeitschritten wird das rechte Ende des Quaders auf den Boden des Simulationsraums treffen. Daraufhin beginnt das Objekt, sich in der Mitte durchzubiegen. Wie stark es dies tut, soll ein Maß für die jeweilige Steifheit sein. Ein steifes Material wird den Boden kaum berühren und danach streben, die Ursprungsconfiguration möglichst beizubehalten. Weniger steife Materialien werden eine größere Auflagefläche besitzen, da die geringere Steifigkeit des Objekts eine größere Deformation erlaubt. Die Poissonzahl ν wird in diesem Versuch konstant bei 0 belassen, um den Versuch nicht durch den Einfluss eines weiteren, für die Elastizität relevanten, Parameters zu verfälschen. Das Elastizitätsmodul E wird in einem Bereich von 50 (sehr weich) bis 1000 (hart) variiert.

Die Ergebnisse dieses Aufbaus zeigen die Abbildungen 6.2 und 6.3. Sie geben den aktuellen Zustand des Systems nach 200 Zeitschritten wieder. Wie erwartet sind Objekte mit höherem Elastizitätsmodul steifer. Bei $E = 50$ liegen annähernd 80% der Grundfläche des Objekts auf dem Boden auf. Zudem wird es, im Gegensatz zu den Objekten mit höherem Elastizitätsmodul, vertikal stark gestaucht. Dies ist ebenfalls der geringen Steifigkeit geschuldet. Das genaue Gegenteil bildet das Objekt mit $E = 1000$. Es wird so gut wie gar nicht gestaucht

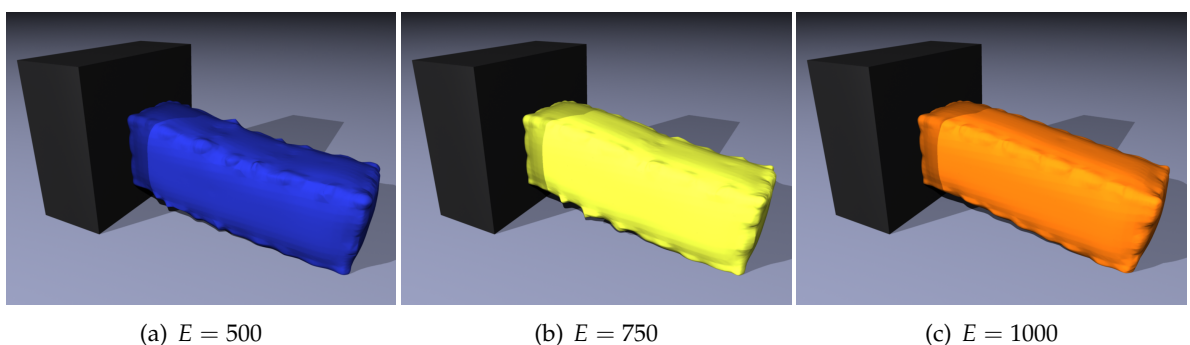


Abbildung 6.3: Ergebnisse Versuch 1 bei $E \in [500, 1000]$

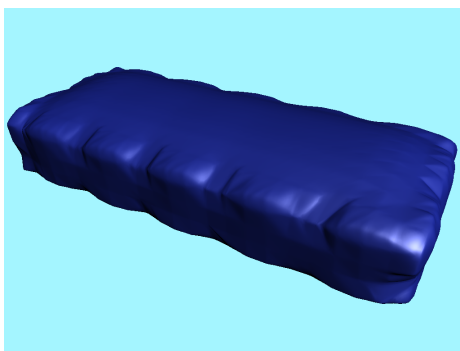
und liegt nur mit der Kante auf dem Boden auf. Die anderen gewählten Werte zeigen Zwischenstufen dieser beiden Extreme. So liegen bei $E = 300$ ca. 30% der Grundfläche auf dem Boden auf, bei $E = 100$ sind es schon 60%.

Objekte mit höherem Elastizitätsmodul E erlauben also weniger absolute Deformation. Sie tendieren eher dazu, an jeder Stelle leichte Deformationen zuzulassen, als an einigen wenigen große. Bei kleinem Modul ist das Gegenteil der Fall. Hier sind große lokale Deformationen möglich. Der Rest des Objekts bleibt dabei möglichst undeformiert. Es soll noch kurz auf ein kleines Phänomen eingegangen werden, welches in Abbildung 6.2(b) zu erkennen ist. Die Höhe des rechten Endes des Objekts ist dort größer, als bei Objekten mit höherem E , was eigentlich nicht sein dürfte. Dies ist dem Aufprallverhalten des Materials geschuldet. Trifft es mit einer gewissen Geschwindigkeit auf den Boden auf, so prallt es von dort ab und schnell leicht nach oben. Zum 200. Zeitschritt war das Objekt aus Abbildung 6.2(b) gerade am höchsten Punkt einer solchen Bewegung. Bei allen anderen Objekten war dies nicht der Fall, was deren geringere Höhe erklärt.

Weitere Beobachtungen zur Variation des Elastizitätsmoduls, insbesondere solche mit Blick auf die Stabilität, sollen in Kapitel 7 erläutert werden.

6.2 Variation der Poissonzahl

Der zweite wichtige Materialparameter für isotrope Materialien neben dem Elastizitätsmodul ist die Poissonzahl ν . Sie ist ein Maß dafür, wie stark sich Verformungen in einer Raumrichtung in die anderen Raumrichtungen fortpflanzen. Um den Effekt dieser Größe zu visualisieren wird folgender Versuchsaufbau angewandt:



Beschreibung	Symbol	Gewählter Wert
Gravitationskonstante	g	0
Länge eines Zeitschritts	δt	0,001s
Simulationsdauer	t_{max}	0,3
Glättungslänge	h	0,055
Partikelmasse	m	0,00001
Elastizitätsmodul	E	300
Poissonzahl	ν	variabel
Partikelabstand	δx	0,025
Anzahl der Partikel	N	357

Abbildung 6.4: Startkonfiguration und Versuchsparameter Versuch 2

Die Startkonfiguration besteht aus einem Quader mit den Maßen $0,2 * 0,1 * 0,5$, welches aus ca. 350 Partikeln aufgebaut ist. Eine der beiden Flächen mit den Maßen $0,2 * 0,1$ ist fixiert. Die gegenüberliegende Fläche wird dann mit $0,7$ Raumeinheiten pro Sekunde auf

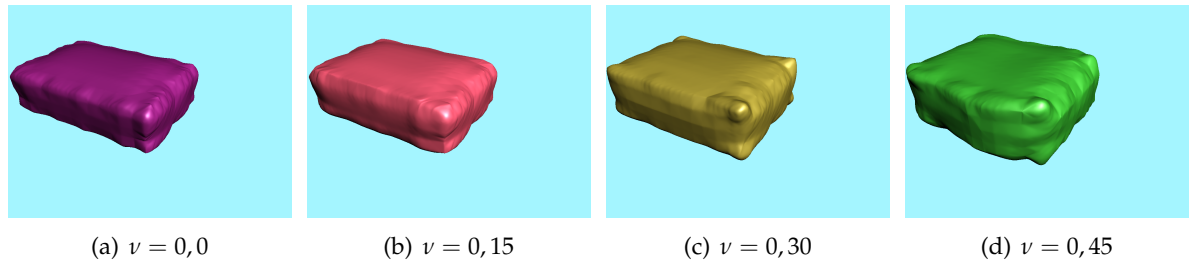


Abbildung 6.5: Ergebnisse Versuch 2

die fixierte Fläche zubewegt. Das Objekt wird also in einer Raumrichtung gestaucht. Dies wird mehrmals mit variierender Poissonzahl ν durchgeführt. Diese bewegt sich in einem Bereich von 0 bis 0,45. Ein Wert von 0,5, welcher in der Realität auch auftreten kann, wird jedoch nicht überprüft, da dieser mit den Formeln für isotrope Materialien nicht vereinbar ist. Setzt man in Gleichung 3.26 den Wert $\nu = 0,5$ ein, würde man durch 0 teilen, was nicht erlaubt ist.

Abbildung 6.5 zeigt die Ergebnisse dieses Experiments nach 200 Zeitschritten. Die Abbildung der Initialkonfiguration zeigt den selben Bildausschnitt wie die Abbildungen der Ergebnisse. Es ist also zu erkennen, dass das Objekt wirklich gestaucht wurde. Das Maß, in dem es sich orthogonal zur Stauchrichtung verbreitert, variiert je nach Poissonzahl. Bei $\nu = 0$ verbreitert sich der Körper überhaupt nicht, was eben so zu erwarten ist. Je höher die Poissonzahl gewählt wird, desto höher bzw. breiter wird das Objekt, wenn es in Längsrichtung zusammengedrückt wird. Dies entspricht der Definition der Poissonzahl und war somit so zu erwarten.

Zusätzlich sei noch auf ein weiteres Phänomen hingewiesen, welches durch diese Simulationmethode abgedeckt wird. Dieses ist jedoch in Render-Szenen sehr schwer zu erkennen, deshalb muss eine schriftliche Beschreibung genügen. Wird die Stauchung des Objektes bei hoher Poissonzahl ruckartig genug ausgeführt, verbreitert sich das Objekt zunächst nur an dem Ende, von dem aus die Stauchung ausgeführt wurde. Diese Verbreiterung pflanzt sich dann schneller als die Stauchung bis hin zum anderen Ende des Objekts fort. Dieses Verhalten ähnelt beispielsweise dem von Wackelpudding. Es ist aber nur zu beobachten, wenn der Zeitschritt nicht zu klein gewählt wurde, da sonst die auftretenden Kräfte zu gering sind, bzw. sich diese besser über das Objekt verteilen.

6.3 Variation von SPH-Grundparametern

Außer den grundlegenden Parametern für elastoplastische Objekte sind auch noch andere Werte interessant. So interessiert beispielsweise, wie sich die Wahl der Länge des Zeitschritts auf die Simulation auswirkt. Weitere interessante Parameter sind die Partikelmasse und die Glättungslänge. Zunächst sei auf die Glättungslänge eingegangen. In vielen Arbeiten, welche

sich auf Smoothed Particle Hydrodynamics stützen, wird die Glättungslänge so gewählt, dass sie in etwa dem doppelten initialen Partikelabstand entspricht [NMK⁺06]. Dies konnte auch hier bestätigt werden. Je größer die Glättungslänge ist, desto mehr Nachbarn besitzt jedes Partikel. Da für jeden Nachbarn Rechenoperationen ausgeführt werden müssen, führen große Nachbarschaften zu hoher Rechenzeit. In Versuchen zeigt sich auch hier, dass die Stabilität der Simulation ab einer Glättungslänge von ca. dem doppelten initialen Partikelabstand nicht mehr wirklich zunimmt. Bei kleineren Glättungslängen wirken entweder gar keine Kräfte oder sie wirken einseitig in eine Richtung, so dass das simulierte Objekt regelrecht explodiert. Die Empfehlung von Solenthaler und Kollegen ist, die Glättungslänge so zu wählen, dass jedes Partikel im Schnitt 40 Nachbarn besitzt [SSP07]. Dies ist in etwa beim doppelten Initialabstand der Fall.

Variiert man die Partikelmasse m stellen sich gänzlich andere Effekte ein. Wird sie erhöht, wird das Objekt weicher. Erniedrigt man sie, wird das simulierte Objekt steifer. Sie scheint also den inversen Effekt des Elastizitätsmoduls zu besitzen. Betrachtet man die Formeln der Modelle für die Elastizität aus Abschnitt 3.3 fällt schnell auf, dass zwischen Masse m und Elastizitätsmodul E ein Zusammenhang besteht. Per Definition ist das Elastizitätsmodul proportional zu der Kraft, die im Innern eines elastischen Körpers wirkt. Je stärker diese Kraft ist, desto steifer ist das Objekt. Lässt man jedoch diese Kraft gleich und erhöht die Partikelmassen, muss plötzlich dieselbe Kraft mehr Masse bewegen. Dies kann nicht so leicht gelingen, deshalb wird hier ein Körper bei Erhöhung der Partikelmassen weniger steif. Bei Verringerung der Partikelmassen ist natürlich das Gegenteil der Fall.

Der letzte interessante Parameter ist die Länge eines Zeitschritts. Dessen Einfluss auf die Stabilität scheint jedoch kaum von anderen Werten abzuhängen. Egal wie andere Werte gewählt waren, bei allen Zeitschrittlängen über $0,001s$ tendiert die Simulation zur Instabilität. Kleinere Längen scheinen sie jedoch nicht stabiler zu machen, was den Beobachtungen von Solenthaler und Kollegen widerspricht [SSP07]. Vor allem können zu kleine Zeitschritte bestimmte Details der Simulation verwischen, wie in Abschnitt 6.2 ausgeführt. Aufgrund dessen wurde in so gut wie jedem Versuch $\delta t = 0,001s$ gewählt.

6.4 Laufzeiten

Insbesondere für Anwendungen in der Computergraphik sind die Laufzeiten der Simulation interessant. Es wird immer danach gestrebt, möglichst interaktive Anwendungen zu entwickeln, d.h. es soll dem Nutzer möglich sein, bestimmte Parameter während der Laufzeit zu ändern und diese Änderungen sofort in der Ausgabe zu beobachten. Für diese ist zumeist erforderlich, dass die Berechnung eines Zeitschritts nur so lange dauert, dass mindestens 25 Zeitschritte pro Sekunde berechnet werden können. Inwiefern dies mit dieser Implementierung möglich ist, soll im Folgenden geklärt werden.

Die Laufzeitmessungen wurden auf einer Testmaschine mit einem Intel Core *i7-2600K* bei $4GHz$ Taktfrequenz durchgeführt, welche $16GB$ Arbeitsspeicher besitzt. Als Grafikkarte ist eine *GeForce GTX580* verbaut. In der Simulation wird 200 Zeitschritte lang das Fallen eines Würfels simuliert. Dieser besitzt die Partikelmasse $m = 0.00001$, das Elastizitätsmodul

N	h	#n	Dauer
729	0,055	42	10,36s
729	0,075	81	20,44s
2197	0,055	46	40,62s
2197	0,075	93	78,47s
4913	0,055	49	126,4s
4913	0,075	99	221,4s

Tabelle 6.1: Laufzeitmessungen Total für 200 Zeitschritte

$E = 500$ und Poissonzahl $\nu = 0,4$. Die Partikelanzahl des Würfels und die Glättungslänge werden variiert. Durch die Änderung der Glättungslänge verändert sich natürlich auch die durchschnittliche Anzahl an Nachbarn, die die Partikel besitzen. Diese ist ausschlaggebend für die Laufzeit, da viele Berechnungen für alle Nachbarn eines Partikels erfolgen. Die Laufzeitmessungen für variierte Partikelanzahl N und Glättungslänge h ist in Tabelle 6.1 gegeben, die genauere Aufschlüsselung nach einzelnen Berechnungsschritten in Tabelle 6.2. Hierbei bezeichnet $\#n$ die durchschnittliche Anzahl an Nachbarn und $\text{comp } n$ die Berechnungszeit, die benötigt wird, um die Nachbarschaften der Partikel zu bestimmen. Die Messwerte geben die gesamt benötigte Zeit für 200 Zeitschritte an.

Es ist zu erkennen, dass die Erhöhung der Glättungslänge um ca. 35% bereits zu einer Verdoppelung der Laufzeit führen kann, ohne dass die Qualität der Simulation zunimmt. Der größten Anteil an der Rechendauer fällt auf die letztendliche Kraftberechnung und auf die Berechnung der Nachbarschaften ab. Die Bestimmung von Verzerrung, Spannung Geschwindigkeit und neuer Position ist nicht von der Glättungslänge abhängig. Dem zufolge ändert sich die benötigte Zeit kaum, wenn h variiert wird. Da diese Werte nur linear von der Partikelanzahl abhängen, ist ihr Einfluss auf die Gesamtlaufzeit sehr gering. Die Berechnung der Nachbarschaften verhält sich schon per Definition quadratisch zur Anzahl der Partikel. Da die Kraftberechnung und die der Verschiebungsgradienten im Normalfall nicht für alle anderen Partikel erfolgen sondern nur für die eigenen Nachbarn, steigt deren Laufzeit nicht so stark mit der Anzahl der Partikel. Jedoch sind diese von der durchschnittlichen Zahl der Nachbarn abhängig.

N	h	x	$\text{comp } n$	∇u	ϵ	σ	F	v
729	0,055	0,003457s	1,34s	2,017s	0,01551s	0,01013s	6,228s	0,00277s
729	0,075	0,003319s	1,398s	3,963s	0,01536s	0,01007s	14,49s	0,00261s
2197	0,055	0,01205s	11,88s	6,764s	0,04664s	0,03049s	20,72s	0,00807s
2197	0,075	0,01295s	12,12s	13,63s	0,04698s	0,0305s	51,78s	0,008699s
4913	0,055	0,03651s	59,47s	15,71s	0,1086s	0,07143s	49,23s	0,02138s
4913	0,075	0,03704s	59,63s	32,77s	0,1064s	0,06852s	127,7s	0,02662s

Tabelle 6.2: Laufzeitmessungen aufgeschlüsselt für 200 Zeitschritte

Bei einer Partikelanzahl von ca. 730 benötigt ein Zeitschritt in etwa 0,05s, was nur das 50-fache der Zeitschrittdauer ist. Da diese Implementierung nicht auf Parallelität ausgelegt ist, wäre mit ausreichender Parallelisierung, insbesondere auf Grafikprozessoren, eine interaktive Anwendung möglich. Je größer die Partikelanzahl ist, desto weniger wahrscheinlich ist, dass die Dauer eines Zeitschritts so gedrückt werden kann, dass dies möglich ist. Insgesamt ist dies wohl bei Zahlen unter 1000 realistisch. Was für Parallelisierungsraten mit den neuesten Grafikkartengenerationen wirklich erreicht werden können, könnte hier nur geschätzt werden, deshalb wollen wir uns eher auf diesen eher konservativen Wert festlegen.

7 Diskussion

Wie jede Simulationsmethode besitzt auch die hier präsentierte Vor- und Nachteile. Je nach Eigenschaften des Modells können nur bestimmte Eigenschaften von wirklichen Materialien abgebildet werden. Welche genau mit dem vorliegenden Modell simuliert werden können und was für weitere Vor- und Nachteile die Methode besitzt, soll im Folgenden erörtert werden.

7.1 Wahl von Elastizitätsmodul und Poissonzahl

Eine der wichtigen Einschränkungen des Modells ist die auf isotrope Materialien. Deren elastische Materialeigenschaften können nur abhängig von Elastizitätsmodul und Poissonzahl beschrieben werden. Die Problematik an dem bestehenden Ansatz besteht nun darin, dass einige Werte für die beiden Größen schon rein mathematisch auszuschließen sind, solange man sich auf die Formeln für isotrope Materialien beschränkt. So ist, wie in Abschnitt 6.2 erläutert, eine Poissonzahl von 0,5 oder nahe 0,5 gar nicht oder kaum möglich. Dasselbe Problem besteht für, zugegeben exotische, Materialien mit $\nu = -1$. Da viele Gummisorten eine Poissonzahl nahe 0,5 besitzen, schließt das eine große Klasse von Materialien aus.

Zu den isotropen Materialien gehören unter anderem auch Metalle. Viele Metalle sind sehr starr und kaum deformierbar, besitzen also ein recht hohes Elastizitätsmodul. In den Versuchen hat sich gezeigt, dass die präsentierte Methode mit steigendem E tendenziell immer instabiler wird, da die auftretenden Kräfte mit dieser Größe ebenfalls steigen. Die weiter oben durchgeführten Experimente wurden nur für zumindest leicht elastische Materialien vorgenommen. Es ist anzunehmen, dass die Methode sehr instabil werden würde, wenn man versucht das Modell auf eher starre Stoffe wie etwa Aluminium anzuwenden. Diese Problematik ist jedoch als eher nicht kritisch zu betrachten, da Solenthaler und Kollegen ebenfalls in [SSP07] einen Ansatz präsentiert haben, um starre Stoffe zu simulieren. Natürlich könnte man auch in die Richtung argumentieren, dass es der Zweck, des Modells ist, für Laien auf den ersten Blick elastisch erscheinende Stoffe (z.B. Gummi) zu simulieren und nicht solche, deren elastische Eigenschaften nur unter Laborbedingungen gezeigt werden können (wie etwa Platin).

Es ist des Weiteren anzumerken, dass generell eine Erhöhung der Poissonzahl die Stabilität zu erhöhen scheint. Da diese generell gefährdet ist, wenn zu große Kräfte auftreten, scheint dies plausibel. Eine Kraft, die in eine bestimmte Richtung auftritt, wird durch eine hohe Poissonzahl in andere Richtungen abgelenkt bzw. verteilt. Somit nimmt der Betrag der Kraft in der initialen Raumrichtung ab, was förderlich für die Stabilität sein kann.

7.2 Vor- und Nachteile bezüglich anderer Methoden

Wie schon erwähnt besitzt auch diese Methode bestimmte Vor- und Nachteile. Da der präsentierte Ansatz primär auf Smoothed Particle Hydrodynamics fußt, wird er sowohl positive als auch negative Eigenschaften des Grundverfahrens übernommen haben. SPH wird gern als schnelle Methode angesehen, jedoch kann diese Behauptung laut [SM93] nicht wirklich gehalten werden. Wenn hohe Genauigkeit, also hohe Partikelzahlen gefordert sind, steigt die Rechendauer auf ähnliche Werte wie z.B. die der Finite-Elemente-Methode. Seit Steinmetz und Kollegen diese Arbeit veröffentlichten hat sich einiges getan. Vor allem im Bereich des parallelen Rechnens gab es enorme Fortschritte. Da sehr viele Operationen bei einer SPH-Methode gleichförmig für alle Partikel ausgeführt werden müssen und diese sich nicht unbedingt gegenseitig bedingen, ist Smoothed Particle Hydrodynamics geradezu prädestiniert für Parallelisierung. In [HKK07] wurde eine SPH-Implementierung für Fluide präsentiert, welche noch bei etwa 60000 Partikeln interaktive Applikationen zulässt, was weit mehr ist, als die hier geschätzten 1000. Es muss jedoch beachtet werden, dass die elastischen Berechnungen eventuell komplexer sein könnten als solche für Fluide.

Während die potentielle Schnelligkeit einen Vorteil darstellt, ist die Genauigkeit eher ein Nachteil von SPH im Allgemeinen und von diesem Modell im Speziellen. Begeht man bei einer FEM einen Fehler in der Implementierung, tendieren diese meist dazu, sehr schnell instabil zu werden. Bei SPH ist dies oft nicht der Fall ([SM93]). Es werden dann eher physikalisch unkorrekte Ergebnisse geliefert, die aber eventuell nicht als solche zu erkennen sind. Da für diese Arbeit keine genauen Materialdaten sondern nur maximal die Dichte, das Elastizitätsmodul und die Poissonzahl vorlagen, ist sehr schwer zu sagen, inwieweit die Implementierung oder das Modell physikalisch korrekt sind. Es konnte nur beobachtet werden, dass es das Verhalten von weicheren elastischen Materialien sehr gut imitiert, wenn nicht gar wiedergibt.

Das präsentierte Modell hat anderen Methoden gegenüber, was die physikalischen Eigenschaften betrifft, auch Vorteile. So können mit einem Masse-Feder-System keine Materialien mit $\nu = 0$ modelliert werden. Dort verhindern die für die Modellierung essentiellen Strukturfedern eine verschwindende Poissonzahl. Mit den Finite-Elemente-Methoden kann wohl im Allgemeinen eine größere Bandbreite an Eigenschaften simuliert werden, als hier möglich, jedoch sind viele Verfahren so spezialisiert, dass die wirklichen Anwendungsbereiche einzelner darunter ähnliche Einschränkungen besitzen dürften.

Der Größte Vorteil dieser Methode besteht in der hohen Erweiterbarkeit. Bei SPH handelt es sich nach allgemeiner Meinung ([NMK⁺06]) um eine sehr leicht wart- und erweiterbare Methode. Für neue Eigenschaften, die modelliert werden sollen, können einfach neue Partikeleigenschaften definiert werden, die dann ebenfalls eventuell mittels SPH approximierbar sind. Speziell das vorgestellte Modell wurde schon um einige Eigenschaften erweitert, zu Details hierzu sei auf den nächsten Abschnitt verwiesen.

Als Fazit lässt sich also sagen, dass die vorliegende Methode, mangels genauerer Untersuchungen, noch nicht als wirklich physikalisch korrekt angesehen werden kann. Sollten diese aber zeigen, dass Korrektheit gegeben ist, ist diese Behauptung natürlich nichtig. Aufgrund

dessen und auch vor allem aufgrund der hohen Parallelisierbarkeit von SPH-Methoden im Allgemeinen ist diese Methode wohl eher zur Animation in der Computergrafik denn zur physikalischen Simulation geeignet. Während bei der Simulation physikalische Korrektheit das Hauptaugenmerk darstellt, ist es bei der Animation zum einen der Anschein der physikalischen Korrektheit (welche aber nicht wirklich gegeben sein muss) und zum anderen die Schnelligkeit. Beides erfüllt diese Methode.

7.3 Erweiterbarkeit

Wie schon erwähnt, besticht das präsentierte Modell durch seine Erweiterbarkeit, nicht zuletzt, weil es fast primär für diesen Zweck entwickelt wurde. In [SSP07] wurde ein Verfahren entwickelt, welches die Interaktionen von Objekten, die mittels dieses Modells simuliert werden, mit Flüssigkeiten und starren Objekten ermöglicht. Es wurden auch Zustandsübergänge vom elastischen Objekt zur Flüssigkeit, verursacht durch Temperaturerhöhung, möglich gemacht.

Die für diese Arbeit vorgenommene Implementierung besitzt, wie schon gesagt, noch hohes Potential für Parallelisierung. Insbesondere die Berechnung der Nachbarschaften und der elastischen Kräfte könnte dadurch signifikant beschleunigt werden. Um dies zu ermöglichen müsste jedoch die Formulierung der elastischen Kraft angepasst werden, damit weniger Synchronisation bei der parallelen Berechnung nötig ist. Bei der aktuellen Formulierung könnte es vorkommen, dass mehrere Kraftberechnungen gleichzeitig versuchen, schreibend auf dieselbe Variable zuzugreifen, wenn diese für verschiedene Partikel gleichzeitig ausgeführt wird. Die Bestimmung der Nachbarschaften könnte jedoch nicht nur mit Parallelisierungen beschleunigt werden. So genannte Hash Grids wären hier ebenfalls eine Möglichkeit um schnellere Nachbarschaftssuche zu gewährleisten.

In ihrer Arbeit „Corotated SPH for deformable solids“ [BIT09] decken Becker und Kollegen eine Problematik in der Formulierung der Bedingung für lokale Undeformiertheit auf. Diese führt bei reiner Rotation eines Körpers, die jedoch keine Deformation beinhaltet, trotzdem zu elastischen Kräften die ungleich 0 sind. Anders formuliert: Das präsentierte Modell ist nicht rotationsinvariant. Becker und Kollegen erweitern das präsentierte Modell, um dieser Problematik Herr zu werden. Dazu wird in jedem Zeitschritt mittels SPH eine Rotationsmatrix berechnet, die auf die Partikelpositionen in Ruhelage angewandt wird. Dadurch kann die Rotationsinvarianz hergestellt werden. Das in deren Arbeit beschriebenen Verhalten konnte auch hier teils reproduziert werden. Ließ man die in Abschnitt 6.1 beschriebene Simulation weiter laufen, als beschrieben, traten nach einigen Hundert Zeitschritten bei Partikeln von rotierten Objektteilen plötzlich sehr starke Kräfte in Richtung ihrer ursprünglichen Position auf. Bei nicht rotierten Objektteilen war dies nicht der Fall. Auf jeden Fall ist das Problem bekannt und es ist ebenfalls bekannt, wie ihm mittels einer Erweiterung beizukommen ist.

Nicht zuletzt wäre auch eine Erweiterung des Modells für andere elastische Materialien außer den isotropen möglich. Dazu wäre nur eine andere Berechnungsvorschrift für den Spannungstensor notwendig. Dies kann jedoch bei bestimmten Materialparametern zu erhöhter Rechendauer führen, da man dann evtl. die Tensorschreibweise der Formel (siehe

Abschnitt 3.3.5) nicht mittels Voigtscher Notation vereinfachen kann und man deshalb plötzlich mit Tensoren vierter Stufe zurechtkommen muss. Dies wäre aber nur bei sehr exotischen Materialien der Fall. Für die meisten Arten von Stoffen existieren einfachere Berechnungsvorschriften (siehe u.a. [Slao2]).

Alles in allem lässt sich feststellen, dass das präsentierte Modell durch seine Erweiterbarkeit besticht und auch schon diverse Erweiterungen daran durchgeführt wurden. Die Möglichkeiten wurden jedoch bei weitem nicht ausgeschöpft, vor allem in Hinblick auf die Schnelligkeit der Simulation und der Abdeckung verschiedenster Materialien.

8 Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde ein Verfahren zur punktbasierten Animation von elastischen Körpern implementiert und evaluiert, welches durch seine hohen Erweiterungsmöglichkeiten besticht. Diese Verfahren basiert auf der Smoothed Particle Hydrodynamics-Methode. Bei dieser werden bestimmte Eigenschaften der Partikel errechnet, indem über Partikeleigenschaften der Nachbarpartikel geglättet wird. So kann etwa der Druck an einer bestimmten Stelle errechnet werden, wenn man die Massen der Nachbarschaft gewichtet aufsummiert. Zunächst wurden in dieser Arbeit die Grundlagen der punktbasierten Simulation, der SPH-Methode und der Elastizitätstheorie besprochen. Daraufhin wurde das Modell von Solenthaler und Kollegen eingeführt, welches die Simulation von elastischen Materialien ermöglicht. Es ändert die gängige Formulierung der Ruhelage von elastischen Objekten, um einfachere Erweiterbarkeit zu ermöglichen bzw. um Vereinigungen mit anderen Objekten oder Spaltungen zu vereinfachen. Daraufhin folgte eine Beschreibung von bestimmten Implementierungsdetails, wie etwa die Kräfte auf die verschiedenen Partikel angerechnet werden. Im Folgenden wurde dann die implementierte Simulationemethode in Hinblick auf Korrektheit und Leistungsfähigkeit evaluiert. Dabei stellte sich heraus, dass sie großes Potential besitzt, durch Parallelisierung und andere Erweiterungen verbessert zu werden. In einer abschließenden Diskussion wurden weiter Vor- und Nachteile erörtert. Insbesondere im Hinblick auf physikalische Korrektheit bestehen noch Verbesserungsmöglichkeiten.

Ausblick

Die Simulation und Animation verschiedenster Materialien hat in den letzten Jahren große Fortschritte gemacht. Jedoch haben beide Bereiche unterschiedliche Anforderungen. Die präsentierte Methode erfüllt nur die eines Bereichs. Die zukünftige Entwicklung wird wohl auf die Kombination beider Richtungen hinauslaufen. Während es für Entwickler von Simulationemethoden wichtig ist, ihre Ergebnisse ansprechend zu präsentieren, ist es für solche von Animationsmethoden ebenfalls wichtig, physikalische Korrektheit zu wahren. Insbesondere durch den raschen Fortschritt der Parallelisierungstechniken wird es möglich sein, kompliziertere Simulationsverfahren auch in der Animation zur Anwendung zu bringen, ohne dass der zeitliche Aufwand explodiert. Inwieweit dabei gitterbasierte Verfahren zur Anwendung kommen können, bleibt abzuwarten, da punktbasierte, wie das präsentierte, für die Zwecke der Animation bessere Grundvoraussetzungen liefern.

Literaturverzeichnis

- [AP98] U. M. Ascher, L. R. Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*. Society for Industrial and Applied Mathematics, Philadelphia, 1998. (Zitiert auf den Seiten 20 und 38)
- [BGB11] H. Bhattacharya, Y. Gao, A. Bargteil. A level-set method for skinning animated particle data. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, S. 17–24. 2011. (Zitiert auf den Seiten 14 und 40)
- [BIT09] M. Becker, M. Ihmsen, M. Teschner. Corotated SPH for deformable solids. In *Proceedings of the Fifth Eurographics conference on Natural Phenomena*, S. 27–34. Eurographics Association, 2009. (Zitiert auf den Seiten 14, 33, 37, 43 und 55)
- [Bra03] D. Braess. *Finite Elemente*. Springer DE, 2003. (Zitiert auf den Seiten 14 und 17)
- [CZKM98] Y. Chen, Q.-H. Zhu, A. Kaufman, S. Muraki. Physically-based animation of volumetric objects. In *Computer Animation 98. Proceedings*, S. 154 –160. 1998. (Zitiert auf Seite 14)
- [DC96] M. Desbrun, M.-P. Cani. Smoothed Particles: A new paradigm for animating highly deformable bodies. In *Eurographics Workshop on Computer Animation and Simulation (EGCAS)*, S. 61–76. 1996. (Zitiert auf den Seiten 13, 21, 24 und 36)
- [GBB09] D. Gerszewski, H. Bhattacharya, A. W. Bargteil. A point-based method for animating elastoplastic solids. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, S. 133–138. 2009. (Zitiert auf Seite 14)
- [GM77] R. A. Gingold, J. J. Monaghan. Smoothed particle hydrodynamics - Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–389, 1977. (Zitiert auf den Seiten 13 und 21)
- [GM97] S. F. F. Gibson, B. Mirtich. A survey of deformable modeling in computer graphics. Technischer Bericht, Citeseer, TR-97-1, 1997. (Zitiert auf Seite 13)
- [HKK07] T. Harada, S. Koshizuka, Y. Kawaguchi. Smoothed particle hydrodynamics on GPUs. In *Computer Graphics International*, S. 63–70. 2007. (Zitiert auf Seite 54)
- [KAG⁺05] R. Keiser, B. Adams, D. Gasser, P. Bazzi, P. Dutre, M. Gross. A unified lagrangian approach to solid-fluid animation. In *Eurographics/IEEE VGTC Symposium Proceedings*, S. 125–148. 2005. (Zitiert auf den Seiten 14 und 33)

- [KMH⁺04] R. Keiser, M. Mueller, B. Heidelberger, M. Teschner, M. Gross. Contact handling for deformable point-based objects. In *Proceedings on Vision, Modeling, Visualization (VMV)*, S. 315–322. 2004. (Zitiert auf Seite 14)
- [Kölo4] K. Kölln. *Morphologie und mechanische Eigenschaften von Zellulosefasern: Untersuchungen mit Röntgen- und Neutronenstreuung*. Dissertation, Christian-Albrechts-Universität Kiel, 2004. (Zitiert auf Seite 29)
- [Luc77] L. Lucy. A numerical approach to the testing of the fission hypothesis. *The astronomical journal*, 82:1013–1024, 1977. (Zitiert auf den Seiten 13 und 21)
- [LW85] M. Levoy, T. Whitted. *The use of points as a display primitive*. University of North Carolina, Department of Computer Science, 1985. (Zitiert auf Seite 13)
- [MCG03] M. Müller, D. Charypar, M. Gross. Particle-based fluid simulation for interactive applications. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, S. 154–159. 2003. (Zitiert auf den Seiten 13, 21 und 24)
- [MKN⁺04] M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, M. Alexa. Point based animation of elastic, plastic and melting objects. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, S. 141–151. 2004. (Zitiert auf den Seiten 13 und 37)
- [Mon92] J. J. Monaghan. Smoothed particle hydrodynamics. *Annual review of astronomy and astrophysics*, 30:543–574, 1992. (Zitiert auf den Seiten 13 und 24)
- [Mono5a] J. J. Monaghan. Smoothed particle hydrodynamics. *Reports on progress in physics*, 68(8):1703, 2005. (Zitiert auf Seite 13)
- [Mono5b] J. J. Monaghan. Smoothed Particle Hydrodynamics. Notes, 2005. URL http://www.ipam.ucla.edu/publications/pcatut2005/pcatut_5472_preprint.pdf. (Zitiert auf den Seiten 13, 22 und 35)
- [New87] I. Newton. *Philosophiae Naturalis Principia Mathematica*, Band 1. Halley, E., 1687. (Zitiert auf den Seiten 13 und 19)
- [NMK⁺06] A. Nealen, M. Müller, R. Keiser, E. Boxerman, M. Carlson. Physically based deformable models in computer graphics. In *Computer Graphics Forum*, Band 25, S. 809–836. 2006. (Zitiert auf den Seiten 13, 15, 16, 37, 49 und 54)
- [OBH02] J. F. O’Brien, A. W. Bargteil, J. K. Hodgins. Graphical modeling and animation of ductile fracture. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, S. 291–294. 2002. (Zitiert auf Seite 14)
- [OH99] J. F. O’Brien, J. K. Hodgins. Graphical modeling and animation of brittle fracture. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, S. 137–146. 1999. (Zitiert auf Seite 14)
- [Ree83] W. T. Reeves. Particle systems - a technique for modeling a class of fuzzy objects. In *Proceedings of the 10th annual conference on Computer graphics and interactive techniques*, S. 359–375. 1983. (Zitiert auf den Seiten 13, 16 und 17)

- [Sla02] W. S. Slaughter. *The Linearized Theory of Elasticity*. Birkhäuser, Boston, 2002. (Zitiert auf den Seiten 7, 26, 27, 28, 29, 30, 31 und 56)
- [SM93] M. Steinmetz, E. Müller. On the capabilities and limits of smoothed particle hydrodynamics. *Astronomy and Astrophysics*, 268:391–410, 1993. (Zitiert auf den Seiten 13 und 54)
- [SSP07] B. Solenthaler, J. Schläfli, R. Pajarola. A unified particle model for fluid-solid interactions. *Computer Animation and Virtual Worlds*, 18(1):69–82, 2007. (Zitiert auf den Seiten 10, 11, 13, 14, 17, 33, 34, 35, 36, 39, 43, 44, 45, 49, 53 und 55)
- [The12] The Engineering Toolbox. Elastic Properties and Young Modulus for some Materials. Webpage, 2012. URL http://www.engineeringtoolbox.com/young-modulus-d_417.html. (Zitiert auf Seite 29)
- [TPBF87] D. Terzopoulos, J. Platt, A. Barr, K. Fleischer. Elastically deformable models. *ACM SIGGRAPH Computer Graphics*, 21(4):205–214, 1987. (Zitiert auf Seite 13)
- [Voi10] W. Voigt. *Lehrbuch der Kristallphysik: (mit Ausschluss der Kristalloptik)*. Teubner, Leipzig, 1910. (Zitiert auf Seite 30)
- [Wal94] W. Walter. *Einführung in die Theorie der Distributionen*. BI-Wissenschaftsverlag, Mannheim, 1994. (Zitiert auf Seite 21)
- [WB97] A. Witkin, D. Baraff. Physically Based Modeling: Principles and Practice Differential Equation Basics. *SIGGRAPH Course notes*, 1:B1–B8, 1997. (Zitiert auf Seite 20)
- [WTo8] C. Wojtan, G. Turk. Fast viscoelastic behavior with thin features. In *ACM SIGGRAPH 2008 papers*, S. 47:1–47:8. 2008. (Zitiert auf Seite 14)

Alle URLs wurden zuletzt am 18. 11. 2012 geprüft.