

TECHNISCHE UNIVERSITÄT DRESDEN

FAKULTÄT INFORMATIK
INSTITUT FÜR SOFTWARE- UND MULTIMEDIATECHNIK
PROFESSUR FÜR COMPUTERGRAPHIK UND VISUALISIERUNG
PROF. DR. STEFAN GUMHOLD

Diplomarbeit

zur Erlangung des akademischen Grades
Diplom-Medieninformatiker

Computergraphische Simulation dynamischer Bruchbildung in starr-plastischen Gemengen

David Körner
(Geboren am 13. September 1980 in Dresden)

Betreuer: Dipl.-Phys. Niels von Festenberg

Dresden, 31. August 2008

Aufgabenstellung

Ziel dieser Arbeit ist die Verbindung eines Nasssandsimulators mit der Modellierung der Bruchbildung in starren Körpern. Dazu sollen bekannte Simulationsansätze für nassen Sand und für zerbrechliche Starrkörper

als Ausgangspunkt verwendet werden. Das Ergebnis soll verwendet werden, um Animationen granularer Materialien

mit hoher Realitätsnähe zu verwirklichen

Im Rahmen der Diplomarbeit sind im Einzelnen folgende Grundaufgaben

zu lösen:

- Erarbeitung bekannter Verfahren zur Sandsimulation und zur Bruchbildungsmodellierung
- Entwicklung einer theoretischen Grundlage zur Verbindung von Fluid-, Starrkörper- und Bruchmodellierung. Hierbei soll auch untersucht werden, welche natürlichen Phänomene außer nassem Sand noch mit der entwickelten Grundlage beschrieben werden können (z.B. Lava, Gletscher, Schnee oder Schlammlawinen etc.)
- Implementierung des theoretischen Ansatzes mit der Möglichkeit zum Speichern und Laden der simulierten Szene
- Dokumentation der Implementierung
- Erstellung mehrerer Ergebnisvideos für verschiedene starr-flüssige Materialien

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die von mir am heutigen Tag dem Prüfungsausschuss der Fakultät Informatik eingereichte Diplomarbeit zum Thema:

Computergraphische Simulation dynamischer Bruchbildung in starr-plastischen Gemengen

vollkommen selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Dresden, den 31. August 2008

David Körner

Kurzfassung

Die Fluidsimulation ist ein populärer Bereich der Computergraphik mit welchem spektakuläre Effekte aus der Natur nachgebildet werden können. Trotz der großen Fortschritte der vergangenen Jahre gibt es allerdings einen nicht abreißen Bedarf nach noch realistischeren und detailgetreueren Bildern. Diese Arbeit untersucht, inwieweit sich Modelle der Bruchbildung und des Materialversagens in Fluidsimulationen verwenden lassen, um neue bisher nicht darstellbare Phänomene aus der Natur computergraphisch zu modellieren. Dabei wird ein neuer Ansatz vorgestellt mit dem sich Brüche auf starren Oberflächen plastischer Gemenge simulieren lassen.

Abstract

Fluidsimulation is a popular area in computergraphics which enables the reproduction of spectacular effects in nature. Although much advancement was achieved over the past decade, there is still a high demand for more realistic and more breathtaking imagery. This work explores, how models of fracture and material failure can be used in fluidsimulation to recreate natural phenomena for which no approaches exist up to date. A new method is proposed which can be used to model cracks on rigid surfaces of plastic flowing media.

Inhaltsverzeichnis

1	Einleitung	3
1.1	Zielstellung	3
1.2	Übersicht	4
2	Grundlagen	5
2.1	Physikalischer Kontext	5
2.2	Elastizitätstheorie	10
2.2.1	Spannungen, Verformungen und Tensoren	10
2.2.2	Verallgemeinerung des Hookeschen Gesetzes	12
2.2.3	Bruchkriterien	12
2.2.3.1	Rankine Bruchkriterium	13
2.2.3.2	Mohr-Coulomb Bruchkriterium	13
2.3	Strömungsmechanik	17
2.4	Numerische Methoden	18
2.4.1	Numerische Methoden der Elastizitätstheorie	19
2.4.2	Numerische Methoden der Strömungsmechanik	21
2.4.2.1	Der Fluidsimulator	24
2.5	Verwandte Arbeiten	26
2.5.1	Modellierung und Animation von Bruchbildung in steifen Körpern	28
2.5.2	Sand als Fluid	30
2.5.2.1	Sandmodell	31
2.5.2.2	Bruchkriterium	32
2.5.2.3	Starrbereiche	33
2.5.2.4	Plastische Bereiche	34
2.5.2.5	Algorithmus	35
2.5.2.6	Ergebnisse	37
3	Bruchbildung auf starren Fluidoberflächen	39

3.1	Modellbildung	39
3.2	Elasto-plastischer Fluss	40
3.3	Oberflächenbruchbildung	43
3.3.1	Bruchkriterium	45
3.3.2	Vertexsplit	47
3.3.3	Relaxation	48
3.3.4	Stabilität	49
3.3.5	Algorithmus	50
3.3.6	Oberflächenspannungen	51
3.4	Re-tiling	55
3.5	Zur Animation von Oberflächenbrüchen	60
3.6	Zur Darstellung und Modellierung von Oberflächenbrüchen	63
4	Zusammenfassung und Schlussfolgerung	66
	Literaturverzeichnis	69

1 Einleitung

1.1 Zielstellung

Die Simulation und Modellierung natürlicher Phänomene ist ein wichtiger Teilbereich in der Computergraphik. Insbesondere die Unterhaltungsindustrie hat hier einen enormen Bedarf an effizient erzeugten Bildern wirklichkeitsgetreuer Darstellungen der Natur. Dazu gehört insbesondere das Thema der Flüssigkeits- oder allgemeiner Fluidsimulation. Innerhalb der vergangenen 10 Jahren wurde die Fluidsimulation aus dem Bereich der Computational Fluid Dynamics in die Computergraphik übernommen. Seitdem wird sie in der Computergraphik zur Animation verschiedenster plastischer Erscheinungen verwendet.

In diesem Zusammenhang ist auch die vorliegende Arbeit zu betrachten, die sich als Ziel gesetzt hat zu untersuchen inwieweit sich die Darstellung und Animation verschiedener plastischer Phänomene durch Berücksichtigung von Bruchbildung in ihrem Realismusgrad steigern lassen. Zur Verdeutlichung des Sachverhalts sei Abbildung 1.1 herangezogen auf welcher Gletscherspalten als Ergebnis von Bruchbildung in einem plastisch fließenden Gletscher zu sehen sind.

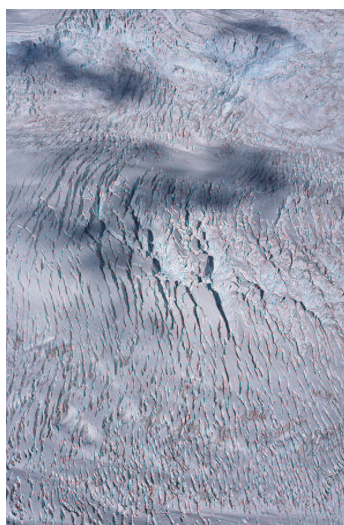


Abbildung 1.1: Gletscherspalten in Neuseeland (Photo von Meredith Lewis)

Während es in der Computergraphik eine Fülle von Arbeiten gibt die von Honig über Schlamm bis

zu zähestem Gummi die verschiedenen Spielarten des Fließverhaltens von Fluiden untersuchen, ist bisher das Thema des Materialversagens in solchen Fluiden fast gar nicht behandelt worden. Es ist meine Überzeugung, dass die Berücksichtigung von Bruchbildung zu einer neuen Qualität von Fluidsimulationen beitragen kann. In diesem Zusammenhang ist diese Arbeit auch als Untersuchung der Frage zu verstehen, in welche Richtung sich die Fluidsimulation in der Computergraphik weiterentwickeln kann. Welche Aspekte können den Realismus der jetzt schon atemberaubenden Ergebnissen noch weiter steigern? Ausgangspunkt der Arbeit ist daher ein Fluidsimulator wie er in ähnlicher Form zur Simulation von Fluiden in der Computergraphik eingesetzt wird. Ziel ist es, den Fluidsimulator so zu erweitern das Bruchbildung berücksichtigt wird, und zu zeigen, dass sich damit Phänomene und Bilder erzeugen lassen, für die es momentan noch keinen Ansatz in der Computergraphik gibt.

1.2 Übersicht

Der Einleitung folgt ein Kapitel zu den Grundlagen der Bruchbildung. Nach einer Beleuchtung des Themas von einem rein physikalischen Standpunkt aus folgt einem aufbauenden Kapitel zu den numerischen Methoden das Kapitel 2.5, in welchem themenverwandte Arbeiten vorgestellt und diskutiert werden. Die zum Verständnis des Kapitel 3 notwendigen Methoden werden dabei vertieft.

Im Kapitel 3 wird ein neuer Ansatz vorgestellt in welchem Brüche auf einer starren Oberfläche eines plastisch fließenden Gemenges simuliert werden können. Dieser Ansatz verbindet zwei Arbeiten von welchem eine aus dem Bereich der Bruchbildung und die andere aus dem Bereich der Fluiddynamik stammt.

Die schriftliche Ausarbeitung wird mit einem zusammenfassenden Kapitel abgeschlossen in welchem auch einige Ergebnisse präsentiert und mögliche Ansätze für weiterführende Arbeiten diskutiert werden.

2 Grundlagen

In diesem Kapitel wird der theoretische Unterbau dieser Arbeit aufbereitet. Über das zentrale Thema der Bruchbildung wird zuerst im folgenden Teilkapitel eine Übersicht erarbeitet. Zur Vertiefung bestimmter Aspekte der Bruchbildung werden dann im Kapitel 2.2 die notwendigen Grundlagen aus der Elastizitätstheorie erläutert und die offenen Aspekte darauf aufbauend erklärt. Da für diese Arbeit von Bedeutung und zur Vollständigkeit wird im darauf folgenden Teilkapitel 2.3 die Strömungsmechanik kurz angerissen.

Im Hinblick auf die verwandten Arbeiten in der Computergraphik, mit welchen sich Kapitel 2.5 auseinandersetzt, werden in Kapitel 2.4 die grundlegenden numerischen Methoden erarbeitet. Dabei wird in Abschnitt 2.4.2.1 der Fluidsimulator erläutert, welcher die Grundlage und Ausgangsbasis dieser Arbeit darstellte. In Abschnitt 2.5.2 wird die im Rahmen dieser Arbeit implementierte Erweiterung des Fluidsimulators zur Sandsimulation vorgestellt.

2.1 Physikalischer Kontext

Das Thema der Bruchbildung ist sehr umfassend. Eine Verortung dieser Thematik in ein bestimmtes Fachgebiet ist im Prinzip nicht möglich, da das Thema des Bruchs sehr viele Wissenschaften berührt und in diesen jeweils in eigene Fachtermini und Bedeutungen eingebettet ist. In der Geologie zum Beispiel setzt man sich unter anderem mit Brüchen in großen Gesteins- und Felsformationen bis hin zu Brüchen bei den kontinentalen Plattenbewegungen auseinander. Hier werden Brüche bei Gesteinen Verwerfungen genannt und allgemeiner als Störung bezeichnet. Damit wird auch die Wissenschaft der Rheologie vom Thema berührt, denn neben dem klassischen Bruch in Abbildung 2.1, werden mit diesen Störungen auch Bereiche plastischer Deformation bezeichnet. Für die Material- und Werkstoffwissenschaften ist die Bruchbildung der verschiedenen Stoffe unter verschiedenen Einflüssen interessant. Für die Fertigung verschiedenster Dinge, wie zum Beispiel Häuser, Kraftfahrzeuge etc. ist es notwendig zu wissen, wann die verschiedenen Materialien brechen und welche Gründe es dafür gibt. In der Kristallographie untersucht man Brüche in ihrer mikroskopischen Beschaffenheit. Dabei sind Brüche auf kleinste Dislokationen in der kristallinen Struktur zurückzuführen die sich zu Mikro-Rissen entwickeln, ausbreiten

und vereinen.



Abbildung 2.1: Sprödbbruch (Abbildung aus dem Skript zu Vorlesung Strukturgeologie der ETH Zürich)

Dieser Querschnitt durch die verschiedenen Größenordnungen in denen Brüche von Bedeutung sind, zeigt sehr gut die Spannweite des Themas. Konsequenterweise ist der Begriff vom Bruch sehr diffus und nur wenig abgegrenzt. Eine präzise Definition gibt es nicht. In seiner allgemeinsten Form kann der Bruch einfach als Deformation aufgrund von Materialversagen verstanden werden. Grundsätzlich wird der Bruch in den verschiedenen Fachbereichen in zwei Bruchtypen unterteilt. Beim auseinander brechen eines Materials, wie in Abbildung 2.1, spricht man vom Kohäsionsverlust aufgrund von Deformation. Diese Form wird Sprödbbruch genannt. Dem gegenüber steht der so genannte duktile Bruch, zu sehen in Abbildung 2.2, bei dem das Material plastisch fließt. Dies ist auch der Berührungspunkt zur Strömungsmechanik, welche sich mit der plastischen Deformation beschäftigt. Ob es zur spröden oder duktilen Verformung kommt hängt von dem Material und einigen äußeren Faktoren ab. Glas zum Beispiel bricht spröde, ohne das eine duktile Verformung erkennbar ist. Stahl hingegen verformt sich in großem Maße plastisch bevor es reißt. Die erwähnten äußeren Faktoren sind unter anderem Druck und Temperatur. Gestein verformt sich bei geringem Druck und geringer Temperatur wie in Abbildung 2.1 spröde. Bei hohem Druck hingegen verformt es sich duktil wie in Abbildung 2.2 zu sehen. Diese plastische Verformung setzt auch bei hohen Temperaturen ein, wie am Beispiel von Lava bekannt ist.

Die Materialdeformation allgemein ist ein zentrales Thema der Kontinuumsmechanik. Hier wird die Deformation grundlegend unterschieden in elastische Deformation, welche reversibel ist, und plastische Deformation, die irreversibel ist. Die Elastizitätstheorie als Teilbereich der Kontinuumsmechanik beschreibt das Verformungsverhalten von elastischen Materialien unter Einwirkung von äußeren Kräften. Die durch die Elastizitätstheorie behandelten Materialien unterliegen bei äußerer Krafteinwirkung einer elastischen Deformation. Also einer Deformation, welche sich bei Wegfall der Krafteinwirkung voll-



Abbildung 2.2: duktiler Bruch (Abbildung aus dem Skript zu Vorlesung Strukturgeologie der ETH Zürich)

ständig in den ursprünglichen, verformungsfreien Zustand zurückversetzt. Diese Eigenschaft ist dadurch beschrieben, dass der Körper durch die Deformation potentielle Energie bzgl. des Ausgangszustandes aufnimmt. Im Kontext wird diese auch elastische Energie genannt und ist durch Spannungen im inneren des Körpers angegeben bzw. gespeichert. Bei überschreiten einer so genannten Streckgrenze - je nach Kontext auch Bruchpunkt, Fließgrenze, Elastizitätsgrenze oder plastisches Fließvermögen bezeichnet - ist die Verformung nicht mehr nur elastisch, sondern auch plastisch und damit zu Teilen nicht mehr vollständig rückkehrbar. Materialien mit diesem Anteil der plastischen Verformung werden als visko-elastische Materialien bezeichnet.

Veranschaulicht wird dieses Verhalten durch das Spannungs-Dehnungs-Diagramm. Um Aussagen über bestimmte Eigenschaften von Materialien machen zu können gibt es den einachsigen Zugversuch. Dabei wird der Werkstoff an zwei Enden eingespannt und immer stärker werdenden Zugkräften ausgesetzt. Die Verformung des Körpers entspricht in diesem Versuch einer Dehnung, also Verlängerung des Körpers. Die Entwicklung der Dehnung bei steigender Zugkraft wird im Spannungs-Dehnungs-Diagramm beschrieben. Das elastisch-plastische Materialverhalten wird in diesem deutlich sichtbar.

Vereinfacht kann das Materialverhalten in drei Bereiche eingeteilt werden. Im verformungsfreien Zustand entspricht das Verhalten der Starrkörperdynamik und das Material ist völlig steif. Mit Beginn der Einwirkung von Oberflächenkräften ist das Materialverhalten durch den linear-elastischen Bereich beschrieben. Dieser Bereich ist durch das Hooksche-Gesetz bestimmt, welches einen proportionalen Zusammenhang zwischen Spannung und Verformung konstatiert und ein zentraler Bestandteil der Elastizitätstheorie ist. Danach folgt der plastische Bereich, welcher vom linear-elastischen Bereich durch die Elastizitätsgrenze getrennt wird. Am Ende des plastischen Bereiches ist der Sprödpoint der angibt, ab

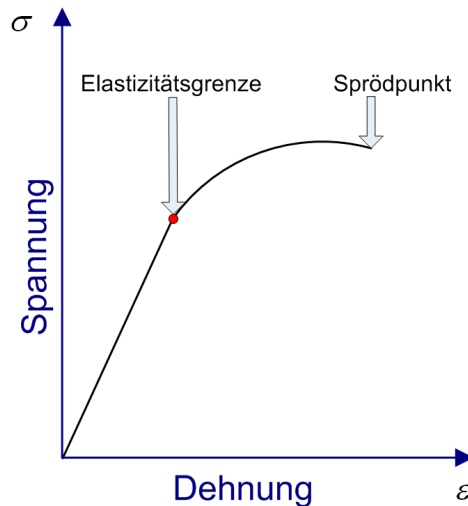


Abbildung 2.3: Spannungs-Dehnungs-Diagramm

wann es zum Kohäsionsverlust und damit zum Spröbruch kommt. Anhand dieses Diagramms sind die zwei verschiedenen Bruchtypen - Spröbruch und duktiler Bruch - zu erkennen. Der Bruchpunkt des duktilen Bruchs ist die Elastizitätsgrenze. Nach der duktilen Verformung kommt es zum Spröbruch, dessen Bruchgrenze der Sprödpunkt ist. Um die zwei Brüche besser voneinander zu unterscheiden wird in der Literatur beim überschreiten des Sprödpunktes auch oft vom reißen statt vom brechen gesprochen. Die Elastizitätsgrenze und der Sprödpunkt sind Materialkennwerte die durch den einachsigen Zugversuch ermittelt werden können und das Materialverhalten bei Deformation maßgeblich bestimmen.

Durch die elastische Deformation grenzt sich die Elastizitätstheorie deutlich von der Strömungsmechanik ab. Die Strömungsmechanik ist ebenfalls ein Gebiet der Kontinuumsmechanik und beschreibt die plastische Deformation von Materialien die sich nicht elastisch verformen. Diese Materialien werden Fluide genannt und konsequenterweise als Materialien definiert, die einer äußeren, beliebig kleinen Kraft (genauer: Scherkraft) keinen Widerstand entgegensetzen - also keine Spannungen aufbauen. Zur Beschreibung visko-elastischer Materialien müssen beide Theorien (Elastizitätstheorie und Strömungsmechanik) vereint werden. Dies geschieht in der Rheologie, der Wissenschaft von fließenden Stoffen.

Die bisherigen Betrachtungen und insbesondere das Spannungs-Dehnungs-Diagramm lassen sich zu folgenden Punkten verallgemeinern:

- Es gibt ein Kriterium welches über Materialeigenschaften ausgewertet wird und angibt ob eine Bruchgrenze überschritten wurde, oder nicht.
- Ist ein Bruch durch das Kriterium festgestellt worden, muss der Bruch modelliert bzw. ausgeführt werden.

- Bestehende Brüche müssen im Simulationsverlauf aktualisiert werden, da sie eine Entwicklung erfahren. So breiten sich Brüche zum Beispiel immer weiter aus, oder weiten sich.

Zur näheren Diskussion der Bruchkriterien ist ein genaueres Verständnis bestimmter Aspekte der Elastizitätstheorie und den Spannungen notwendig. Daher werden die Bruchkriterien erst später in Kapitel 2.2.3, nach Einführung der Spannungen und dem Spannungstensor, behandelt.

Die Bruchmodellierung entspricht beim duktilen Bruch zu einem bestimmten Anteil dem plastischen Fluss. Dieser wird bestimmt durch den plastischen Anteil an der Änderung der Verformung des Materials. Dieses Inkrement wird Fließgesetz (im engl: flow rule) genannt und ist für die Modellierung der duktilen Brüche wichtig. Bei einem Sprödbbruch bildet sich dagegen, in Abhängigkeit vom Bruchkriterium, eine Bruchebene aus. Diese Bruchebene ist durch die Bruchspitze begrenzt. Der Kohäsionsverlust tritt entlang dieser Ebene auf und das Material bewegt sich in drei grundlegenden Arten voneinander weg. Diese Arten sind die drei in 2.4 abgebildeten Bruchmodi. Entweder bewegt sich das Material orthogonal zur Bruchebene auseinander (Bruchmodus I), tangential zur Bruchebene und orthogonal zur Bruchspitze (Bruchmodus II) oder tangential zur Bruchebene und parallel zur Bruchspitze (Bruchmodus III).

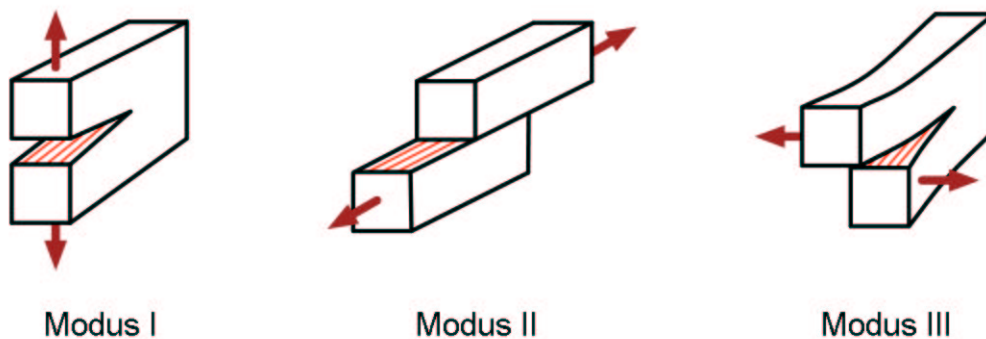


Abbildung 2.4: Die verschiedenen Bruchmodi bei Sprödbrüchen

Anhand dieser drei Bruchmodi werden die Sprödbrüche nochmals in zwei Bruchformen mechanisch klassifiziert:

- Dehnungsbrüche (Bruchmodus I)
- Scherbrüche (Bruchmodus II und Bruchmodus III)

Damit ist ein erster grober Überblick über das Thema der Bruchbildung gegeben. Um nun die Theorie weiter bis hin zu den praktischen Umsetzungen zu entwickeln, ist ein Verständnis von Teilen der Elastizitätstheorie notwendig.

2.2 Elastizitätstheorie

Wie bereits im Kapitel 2.1 erwähnt, ist das Hookesche Gesetz der zentrale Bestandteil der Elastizitätstheorie. Dieses beschreibt in der einfachsten Form den Zusammenhang zwischen Federkraft F und Federlänge Δl und konstatiert, dass es eine streng lineare Beziehung zwischen beiden gibt:

$$F = D \cdot \Delta l \quad (2.1)$$

Die Kraft F wird pro Fläche angegeben und ist daher eine Spannung. Je stärker die Spannung F , desto stärker ist die Dehnung der Feder. D ist die Federkonstante bzw. allgemeiner die Proportionalitätskonstante. In allgemeiner Form setzt das Hookesche Gesetz die wirkenden Spannungen und die bestehende Verformung in einem Körper in Zusammenhang. Um die allgemeine Notation angeben zu können müssen Spannungen im mehrdimensionalen Fall und deren Notation durch Tensoren eingeführt werden.

2.2.1 Spannungen, Verformungen und Tensoren

Für den Fall der eindimensionalen Feder kann die Spannung in Form eines Skalars angegeben werden. Dies ist deshalb möglich, weil es durch den eindimensionalen Charakter nur eine Achse gibt, auf welcher die Spannung wirkt und auch die Fläche gerichtet ist, bezüglich welcher die Spannung beschrieben wird. Im Falle von zwei Dimensionen ist das nicht mehr der Fall. Die Spannung F besteht hier aus einem gerichteten Vektor $\vec{F} = (F_x, F_y)^T$ und auch die Angriffsfläche, bezüglich welcher die Spannung wirkt, hat eine Richtung \vec{n} . Würde man nun alle wirkenden Kräfte in einem Punkt angeben wollen, müsste man für jede Angriffsfläche \vec{n} in diesem Punkt den wirkenden Spannungsvektor \vec{F} explizit angeben. Um das zu vermeiden, wird der Spannungsvektor \vec{F} jeweils für die Ebenen angegeben, welche durch die Einheitsvektoren \vec{e}_1 und \vec{e}_2 definiert sind. Dabei wird der Spannungsvektor der auf die Ebene mit der Normalen \vec{e}_1 wirkt, mit $\vec{\sigma}_x$ bezeichnet und analog für die Ebene mit der Normalen \vec{e}_2 mit $\vec{\sigma}_y$. Nach der Kontinuumsmechanik können die Spannungsvektoren für alle Ebenen \vec{n} mit der Linearkombination

$$\vec{F}_{\vec{n}} = n_x \cdot \vec{\sigma}_x + n_y \cdot \vec{\sigma}_y \quad (2.2)$$

berechnet werden. Wie bei Linearkombinationen üblich kann (2.2) auch in Matrixschreibweise dargestellt werden:

$$\vec{F}_{\vec{n}} = \begin{pmatrix} \sigma_{xx} & \sigma_{yx} \\ \sigma_{xy} & \sigma_{yy} \end{pmatrix} \cdot \vec{n} = \sigma \cdot \vec{n} \quad (2.3)$$

Aus der Kontinuumsmechanik folgt weiter, dass σ immer symmetrisch ist (siehe [Chu07]). Intuitiv gesprochen bedeutet dies, dass dieselbe auf der Ebene \vec{n} wirkende Spannung auch auf die oppositional gerichtete Ebene $-\vec{n}$ wirkt. Bildlich vergegenwärtigt bedeutet die Symmetrieeigenschaft zusammen mit (2.3), dass der Betrag von σ_x und σ_y für alle \vec{n} eine beliebig rotierte Ellipse im Raum beschreibt.

σ ist der Spannungstensor. Für den dreidimensionalen Fall ergibt sich entsprechend ein 3×3 Spannungstensor.

Die auf der Hauptdiagonale liegenden Komponenten des Spannungstensors sind die Komponenten, welche den Anteil am resultierenden Kraftvektor beisteuern, der normal zur Ebene \vec{n} wirkt. Auf den Nebendiagonalen liegen entsprechend die Komponenten, welche die Anteile am Kraftvektor besteuern, die tangential zur Ebene \vec{n} wirken. Diese tangential wirkenden Spannungen werden Schubspannungen oder auch Scherspannungen genannt. In Abbildung 2.5 ist der Spannungstensor veranschaulicht.

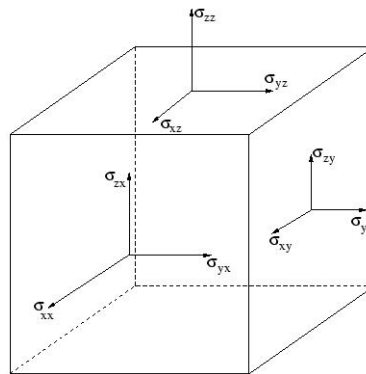


Abbildung 2.5: Der Spannungstensor mit Normal- und Schubkräften

Der durch die neun Komponenten von σ beschriebene Spannungszustand lässt sich vereinfachen, indem die Normalspannungen maximiert werden. Dies wird mit der Eigenwertanalyse durchgeführt und entspricht der Suche nach dem Koordinatensystem, in welchem die Scherspannungen verschwinden und die wirkende Spannungen nur normal zur Ebene wirken. Die Achsen dieses Koordinatensystems sind die Eigenvektoren und die dazugehörigen Eigenwerte sind die Hauptspannungen σ_1 und σ_2 (im zweidimensionalen Fall). Diese werden meist der Größe nach abfallend sortiert, so dass σ_1 die größte Hauptspannung und σ_2 die kleinste ist. Wichtig ist auch, dass der durch Hauptspannungen beschriebene Spannungszustand äquivalent zum ursprünglichen Spannungszustand σ ist.

Analog zur Spannung kann eine gleiche Tensordarstellung auch für die Dehnung hergeleitet werden. Dabei wird die Dehnung durch den Verzerrungstensor

$$\epsilon = \begin{pmatrix} \epsilon_{xx} & \epsilon_{yx} \\ \epsilon_{xy} & \epsilon_{yy} \end{pmatrix} \quad (2.4)$$

angegeben. Dieser wird auch Verformungstensor genannt und ist ebenfalls wie σ symmetrisch.

2.2.2 Verallgemeinerung des Hookeschen Gesetzes

Mit Hilfe der Tensornotation von Spannung und Verformung kann das Hookesche Gesetz in allgemeiner Form für den mehrdimensionalen Fall folgendermaßen Angegeben werden:

$$\tilde{\sigma} = C\tilde{\epsilon} \quad (2.5)$$

Dies ist analog zur eindimensionalen Form aus Gleichung (2.1). Dabei beschreibt $\tilde{\sigma}$ die wirkende Spannung, welche durch den Zusammenhang der vorhandenen Verformung $\tilde{\epsilon}$ und einer Proportionalitätskonstanten C beschrieben werden kann. Die Tilde über den Tensoren σ und ϵ gibt an, dass hier die Tensoren nicht in Matrixschreibweise angegeben sind, sondern dass die Komponenten in Vektorform verwendet werden. C ist ein Tensor, der für jeden möglichen Verzerrungszustand $\tilde{\epsilon}$ die Elastizität des Materials angibt und damit die kristalline Struktur des Materials repräsentiert. C ist entsprechend im zweidimensionalen Fall ein 4×4 Tensor mit 16 und im dreidimensionalen Fall ein 9×9 Tensor mit 81 Komponenten. Aufgrund der Symmetrieeigenschaft von σ und ϵ lassen sich die Vektoren $\tilde{\sigma}$ und $\tilde{\epsilon}$ im dreidimensionalen Fall auf sechs Komponenten reduzieren und die Elastizitätskonstante entsprechend auf 36 Komponenten. Durch weitere energetische Betrachtungen und dem Betrachten von isotropen Materialien vereinfacht sich das Hookesche Gesetz auf:

$$\sigma_{ij} = 2\mu\epsilon_{ij} + \lambda\delta_{ij}\epsilon_{mn} \quad (2.6)$$

λ und μ sind die so genannten Lameschen-Konstanten.

2.2.3 Bruchkriterien

Zur Beschreibung des Verhaltens eines Materials im linear-elastischen Bereich des Spannungs-Dehnungs-Diagramms aus Abbildung 2.3, kann die Elastizitätstheorie eingesetzt werden. Zur Bruchbildung ist nun

zu klären, wie ein Materialversagen festgestellt werden kann. Dabei spielen die Spannungen und materialabhängige Kennwerte eine Rolle. Diese werden durch ein Bruchkriterium in jedem Ort des zu analysierenden Körpers ausgewertet. Das Bruchkriterium, oder allgemeiner Versagenskriterium, ist eine implizite Funktion im Spannungsraum mit der Form (hier für den dreidimensionalen Fall)

$$f(\sigma_1, \sigma_2, \sigma_3) = 0 \quad . \quad (2.7)$$

Der Spannungsraum ist ein kartesisches Koordinatensystem mit den Hauptspannungen als Komponenten. Die implizite Oberfläche dieser Funktion im Spannungsraum ist die so genannte Fließortfläche und gibt demnach an, bei welchen Konstellationen von Hauptspannungen das Material von einem stabilen in einen instabilen Zustand übergeht, also Materialversagen eintritt. Die Fließortfläche bildet eine abgeschlossene Form. Im inneren der Form befinden sich die durch die entsprechenden Hauptspannungskordinaten im Spannungsraum äquivalent beschriebenen Spannungen σ , bei denen kein Materialversagen eintritt. Alle Bereiche außerhalb des von der Fließortfläche beschriebenen Körpers sind die Spannungen, bei denen das Material versagt und es zum Bruch bzw. zum fließen kommt.

Es gibt verschiedene solcher Versagenskriterien. Zwei davon werden im folgenden vorgestellt.

2.2.3.1 Rankine Bruchkriterium

Das Rankine-Bruchkriterium ist das wohl einfachste Bruchkriterium der Art

$$\sigma_v \leq \sigma_G \quad . \quad (2.8)$$

Dabei ist σ_v die Rankine-Vergleichsspannung und σ_G ein skalarer Materialkennwert, der aus einem einachsigen Zugversuch ermittelt wurde und angibt, ab welcher Zugspannung das Material zerreißt. Die Rankine Vergleichsspannung entspricht der größten Normalspannung von σ :

$$\sigma_v = \max(\sigma_1, \sigma_2, \sigma_3) \quad (2.9)$$

2.2.3.2 Mohr-Coulomb Bruchkriterium

Das bekannteste und auch für diese Arbeit wichtigste, ist das Mohr-Coulomb Bruchkriterium, welches nun vorgestellt wird.

Neben Festkörpern ist das Mohr-Coulomb Bruchkriterium auch für Böden und granulare Stoffe allgemein von Bedeutung. Das Kriterium lässt sich intuitiv anhand des Falles eines einfachen geometrischen

Körpers der an einer abfallenden Fläche hinabgleitet veranschaulichen. Wie in Abbildung 2.6 zu erkennen sagt das Bruchkriterium aus ab wann der Körper sich auf einer Bruchfläche bewegt. Dies tut er wenn die Scherspannung relativ zur Normalspannung steigt oder die Normalspannung relativ zur Scherspannung sinkt. Ab welcher Konstellation von Scher- und Normalspannung genau die Bewegung eintritt hängt vom Reibungswiderstand ab welcher der Scherspannung entgegenwirkt. Je größer der Reibungswiderstand desto größer muss die Scherspannung bezüglich der Normalspannung sein um eine Bewegung zu verursachen. Wie in Abbildung 2.6 auf der rechten Seite zu sehen stellt das Mohr-Coulomb Bruchkriterium also ein Verhältnis zwischen Normal- und Scherspannung dar dessen Anstieg vom Reibungswiderstand μ abhängig ist. Weiterhin wirken eventuell noch Haftungskräfte zwischen den Körpern welche als Kohäsionskonstante c zum Wertebereich hinzu addiert wird:

$$\tau = c + \mu\sigma_n \quad (2.10)$$

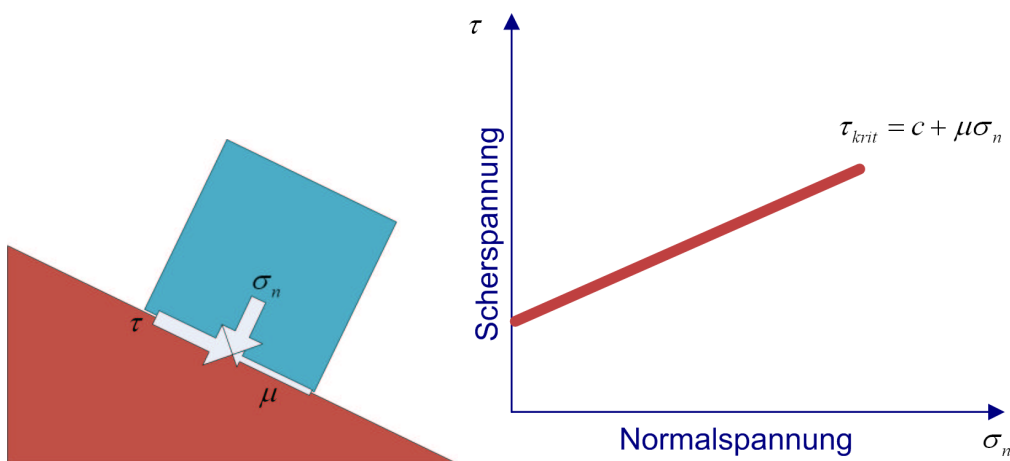


Abbildung 2.6: Mohr-Coulomb Bruchkriterium intuitiv im einachsigen Fall (*links*) und als Funktion im Raum der Normal- und Scherspannung (*rechts*).

Im Kontinuum gibt es nicht nur eine konkrete Bruchebene mit eindeutigen skalaren Werten für Normal- und Scherspannungen wie in Abbildung 2.6 links. Stattdessen gibt es für jeden Ort unendlich viele mögliche Bruchebenen für welche jede einzelne eine konkrete Normal- und Scherspannung hat. Die Menge aller Spannungen für jede mögliche Ebene in einem Punkt ist der Spannungszustand dieses Punktes und wird mit Hilfe eines Tensors σ angegeben.

Will man nun das Mohr-Coulomb Bruchkriterium für den kontinuierlichen Fall auswerten, dann muss man das Kriterium für Tensoren generalisieren. Denn statt einem Wert für Normalspannungen und einem Wert für Scherspannung gibt es nun je nach Dimension eine ganze Reihe von Tensorkomponenten. Diese können zwar äquivalent je nach Dimension auf zwei oder drei Hauptspannungen reduziert werden, müssen dann aber immer noch in irgend einer Weise zu einem einzigen Skalar jeweils für Normal-

und Scherspannung verrechnet werden. Für die Scherspannung ist dies schwierig, da hier aus den Tensorkomponenten ein skalarer Wert σ_v für das Maß der Scherspannung extrahiert werden muss. Dies ist das Prinzip der Vergleichsspannung welche eine fiktive einachsige Spannung darstellt, die möglichst gut die tatsächliche mehrachsige Spannung beschreibt. Es gibt verschiedene Möglichkeiten σ_v aus σ zu berechnen. Diese Rechenvorschriften nennt man Vergleichsspannungshypothesen. Jede dieser Rechenvorschriften kommt auf seine eigene Weise zu σ_v indem es die Tensorkomponenten auf verschiedene Arten in der Berechnung berücksichtigt. Dies bedeutet, dass die Vergleichsspannungshypothese den Versagensmechanismus impliziert. Daher nennt man sie auch Versagensregel.

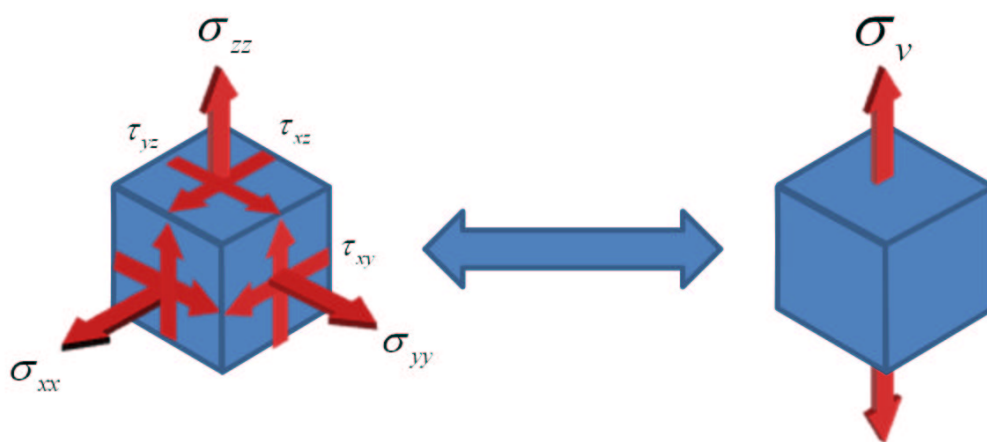


Abbildung 2.7: Prinzip der Vergleichsspannung

Das klassische Mohr-Coulomb Bruchkriterium wird in der Literatur meist mit der Schubspannungshypothese, auch Tresca Vergleichsspannung genannt, angegeben. Diese ergibt sich aus der größten Hauptspannungsdifferenz und wird im dreidimensionalen Fall folgendermaßen berechnet:

$$\sigma_v = \max(|\sigma_1 - \sigma_2|, |\sigma_2 - \sigma_3|, |\sigma_3 - \sigma_1|) \quad (2.11)$$

Bei Verwendung der Schubspannungshypothese wird also die größte Scherspannung als Versagenskriterium angenommen. In dieser Arbeit wird später die Gestaltänderungshypothese verwendet, welche auch oft von-Mises Vergleichsspannung genannt wird und zu den am häufigst verwendeten Vergleichsspannungen zählt:

$$\sigma_v = \sqrt{\frac{1}{2} \left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2 \right]} \quad (2.12)$$

Während die Rankine-Vergleichsspannung aus Gleichung 2.9 ein Materialversagen aufgrund der stärksten Normalspannung bedeutet, impliziert die Formulierung der von-Mises Vergleichsspannung die Ver-

zerrungsenergie als die Größe, welche als Kriterium für ein Versagen des Materials herangezogen werden kann.

Der Zusammenhang zwischen Versagenskriterium und Vergleichsspannung bedarf einer besonderen Aufmerksamkeit, da diese beiden Begriffe in der Literatur generell etwas unscharf gebraucht werden. Zum Beispiel findet man in einigen Büchern den Begriff der von-Mises Vergleichsspannung und in anderer Literatur ist vom von-Mises Bruchkriterium die Rede. Der Grund dieser Unschärfe liegt darin, dass Bruchkriterium und Vergleichsspannung sehr eng aneinander gekoppelt sind. Die Versagenskriterien bauen stark auf den Vergleichsspannungen auf, da immer in irgend einer Weise die mehr-achsigen Spannungen reduziert werden müssen. Dies sorgt dafür, dass die Versagenskriterien maßgeblich von den Vergleichsspannungen bestimmt und charakterisiert werden. So wird in Kapitel 2.5.2 statt der Tresca-Vergleichsspannung die von-Mises Vergleichsspannung für das Mohr-Coulomb Bruchkriterium verwendet. Dadurch liegt rein formell nicht mehr das Mohr-Coulomb Bruchkriterium zu Grunde. Im Bezug auf die Vergleichsspannungen lässt sich die begriffliche Unschärfe auch dadurch erklären, dass sich die Bruchkriterien in fast allen Fällen auf die Form

$$\sigma_v \leq \sigma_G \quad (2.13)$$

reduzieren. Wobei σ_G ein Materialkennwert und σ_v die aus σ berechnete Vergleichsspannung ist. Damit wird die Vergleichsspannung zum tragenden Element des Bruchkriteriums und indirekt zum Bruchkriterium selbst.

Zum Abschluss des Kapitels und zur Verdeutlichung des eben diskutierten Sachverhalts seien in Abbildung 2.8 die Fließortflächen eines Bruchkriteriums welches nur durch die Tresca Vergleichsspannung gegeben ist und die Fließortfläche des Mohr-Coulomb Bruchkriteriums, welches auf der Tresca-Vergleichsspannung basiert, im zweidimensionalen Spannungsraum angegeben.

Ist letztendlich ein Bruchkriterium gegeben, so wird dieses in jedem Ort im zu analysierenden Körper ausgewertet. In den Fällen, wo das Bruchkriterium null ist, ändert sich die Stabilität des Materials. An diesen Punkten kommt es zum Materialversagen. Die Menge all dieser Punkte bilden Bruch- bzw. Fließflächen im Körper aus. Die konkrete Auswertung des Bruchkriteriums und die Behandlung erkannter Brüche sind stark vom numerischen Verfahren abhängig. Bevor nun diese Seite beleuchtet wird, folgt ein kurzes Kapitel welches die für diese Arbeit relevanten Inhalte der Strömungsmechanik kurz rekapituliert.

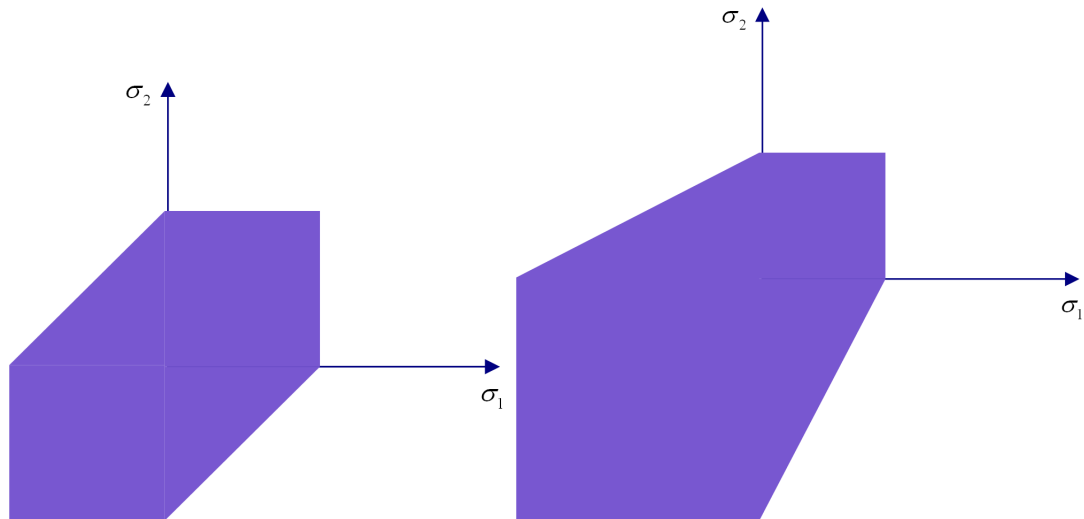


Abbildung 2.8: Fließortfläche im zweidimensionalen Spannungsraum. Tresca (*links*) Mohr-Coulomb (*rechts*).

2.3 Strömungsmechanik

Die Strömungsmechanik ist ein wichtiger Teilbereich der Kontinuumsmechanik und beschreibt den plastischen Fluss von Fluiden. Fluide sind Stoffe, welche einer beliebig kleinen Scherspannung keinen Widerstand entgegensetzen. Das bedeutet, dass in Fluiden keine Spannungen auftreten bzw. diese sofort in plastische Deformation umgesetzt werden. Wie bei der Elastizitätstheorie auch gilt hier die Kontinuums-hypothese, welche besagt das sich die Eigenschaften auf Teilchenebene bei einem hinreichend kleinem Kontrollvolumen durch konstante Werte annähern lassen.

Grundlage der Strömungsmechanik ist eine Erhaltungsgleichung aus der Physik:

$$\begin{aligned} &\text{Zeitliche \u00c4nderung der Masse im Volumenelement} = \\ &\sum \text{der einstr\u00f6menden Massenstr\u00f6me in das Volumenelement} - \\ &\sum \text{der ausstr\u00f6menden Massenstr\u00f6me in das Volumenelement} \end{aligned}$$

Die Masse wird dabei pro Volumen, also in Dichte ρ angegeben. $\vec{v}(x)$ ist ein Vektorfeld im Raum und beschreibt die Geschwindigkeit innerhalb des Fluids als kontinuierliche Variable. Die Komponenten $\vec{v} = (u, v, w)$ beschreiben die Geschwindigkeit des Fluids in die Richtungen x, y, z . F\u00fcr den allgemeinen Fall der instation\u00e4ren Fluide beschreibt t die Zeit. Durch die Annahme, dass das Fluid inkompressibel ist, also das dessen Dichte sich nicht \u00e4ndert, kommt man nach umstellen auf die Konstinit\u00e4tsgleichung:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (2.14)$$

Ein weiterer Erhaltungssatz, der ber\u00fccksichtigt werden muss, ist der Impulserhaltungssatz, welcher nach

theoretischen Überlegungen und Umstellungen letztendlich in die Gleichung

$$\rho \cdot \left(\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \nabla) \vec{v} \right) = \vec{f} - \nabla \cdot p + \mu \nabla^2 \vec{v} \quad (2.15)$$

mündet. Diese Gleichung besagt, dass die zeitliche Änderung des Impulses $\delta \vec{v} / \delta t$ abhängig ist, von:

- Volumenkräften \vec{f} wie zum Beispiel der Gravitation
- den Oberflächenkräften, welche durch Druckunterschiede $-\nabla p$ in der Umgebung von \vec{x} existieren
- den Oberflächenkräften, welche aufgrund von koppelnden Kräften zwischen den Teilchen entstehen, den Reibungskräften $\mu \nabla^2 \vec{v}$, die durch den Reibungskoeffizienten μ parametrisiert sind
- der Änderung des Impulses durch Fortbewegung von Teilchen aus dem Kontrollvolumen, $(\vec{v} \cdot \nabla) \vec{v}$, welche ihren eigenen Impuls “mitnehmen“ und Teilchen die sich in das Kontrollvolumen hineinbewegen und einen eigenen Impuls “mitbringen“

Für eine sehr ausführliche und leicht verständliche Herleitung der Gleichungen sei das dritte Kapitel in [Oer99] empfohlen.

2.4 Numerische Methoden

In den vorangegangenen Teilkapiteln wurden die für diese Diplomarbeit grundlegenden physikalischen Modelle eingeführt. Nun werden bestimmte Fragen an diese Modelle gestellt. Zum Beispiel nach der Verformung eines Körpers unter Einwirkung einer bestimmten Kraft, oder dem Fließverhalten einer bestimmten fluiden Masse. Die Beantwortung dieser Fragen wirft Probleme auf:

- Für manche Modelle, wie zum Beispiel den Navier-Stokes-Gleichungen aus Kapitel 2.3, gibt es keine analytische Lösung. Fragen an das Modell können also nicht einfach durch Umstellen und Einsetzen von Werten in eine Gleichung gelöst werden. Es müssen *Näherungslösungen* gefunden werden.
- Die Modelle arbeiten mit kontinuierlichen Werten. So ist die Verformung und Spannung im Hookeschen Gesetz aus Kapitel 2.2 zum Beispiel ein kontinuierliches Tensorfeld. Diese kontinuierlichen Variablen können nicht exakt angegeben bzw. dargestellt werden, da es Beschränkungen von Speicherplatz und Rechenleistungen gibt die sich nicht abschaffen lassen. Daher müssen für diese kontinuierlichen Variablen *näherungsweise Darstellungen* gefunden werden. Dieses Problem gilt

nicht alleine für kontinuierliche Variablen, sondern generell für alle Zahlen, da das Rechnen mit dem Computer (und auch auf dem Papier) ungenau ist.

- Mit *Näherungslösungen* und *näherungsweise Darstellungen* können Antworten aus den Modellen gewonnen werden. Diese sind allerdings nicht exakt sondern fehlerbehaftet. Zu jeder Antwort muss deshalb auch heraus gefunden werden, wie ungenau diese ist.

All diese genannten Punkte gehören in das Gebiet der Numerik. Ein Teilbereich der Mathematik, der die Brücke zwischen den in den vorangegangenen Kapiteln vorgestellten physikalischen Modellen, und den im nächstfolgenden Kapitel 2.5 vorgestellten praktischen Umsetzungen schlägt. Die Numerik beschäftigt sich mit der näherungsweise Lösung kontinuierlicher Probleme. Eine erschöpfende Behandlung des Themas kann zum Beispiel in [HB06] gefunden werden. Im folgenden sollen die für diese Diplomarbeit wichtigen Aspekte anhand der beiden Modelle aus der Elastizitätstheorie und der Strömungsmechanik beleuchtet werden. Diese numerischen Algorithmen bilden in den meisten Fällen die Grundlage für die physikalisch plausiblen Methoden in der Computergraphik.

2.4.1 Numerische Methoden der Elastizitätstheorie

In der Elastizitätstheorie ist die Methode der finiten Elemente die de-facto Standardmethode zur Lösung entsprechender Probleme. Sie fand in den 50iger Jahren ihre erste Anwendung zur Berechnung der Belastung von Flugzeugtragflächen. Erst Jahre später wurde ein mathematischer Unterbau durch die Variationsformulierung geschaffen, der die Korrektheit des Verfahrens bewies.

Die kontinuierlichen Variablen in der Elastizitätstheorie sind die Zeit t , die Verformung und der Spannungstensor. Die Diskretisierung der Zeit erfolgt durch gleichmäßige Zeitschritte Δt . Die Tensoren sind kontinuierlich über das Volumen eines Körpers im Raum \mathbb{R}^3 definiert. Diskretisiert werden sie, indem sie als konstant über kleine Teilvolumina innerhalb des Volumens angegeben werden. Der Körper wird aus diesen Teilvolumina aufgebaut und die kontinuierlichen Werte sind damit stückweise konstant im Raum definiert. Diese Teilvolumina können aus den verschiedensten Primitiven, wie zum Beispiel Octaedern oder Hexaedern bestehen. Am meisten wird eine Diskretisierung auf Basis von Tetraedern verwendet. Der Körper wird also durch ein Tetraedernetz beschrieben. Die Punkte der Tetraeder sind die Knoten. Über die Volumina der einzelnen Tetraeder werden die kontinuierlichen Variablen, wie zum Beispiel die Spannung und die Verformung als konstant angenommen und diskret angegeben.

Der Körper ist im Raum der Materialkoordinaten \mathbf{u} definiert. Dieses Koordinatensystem entspricht dem lokalen Koordinatensystem des Körpers im undeformierten Zustand. Die Verformung des Körpers kann durch eine kontinuierliche Funktion $\mathbf{p}(\mathbf{u})$, also eine Verschiebung jedes einzelnen Punktes im Raum

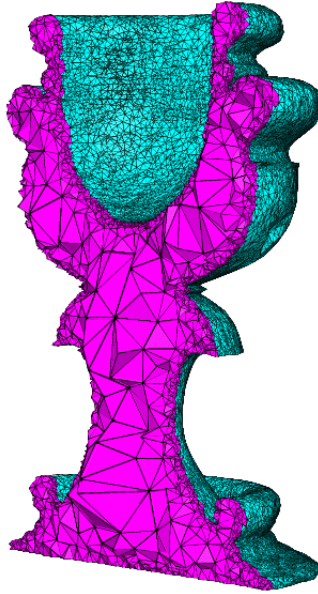


Abbildung 2.9: Querschnitt durch die Tetraederdiskretisierung eines Körpers

dargestellt werden. Die Funktion $\mathbf{p}(\mathbf{u})$ wird diskretisiert, indem die Verschiebung eines jeden einzelnen Knotens angegeben wird. Die Verschiebung eines beliebigen Punktes x im Körper kann dann aus der Linearkombination aller vier angrenzenden Knoten des Tetraeders, in welchem sich x befindet, bestimmt werden. Ziel ist es $\mathbf{p}(\mathbf{u})$, und damit die Verformung des Körpers, unter Einfluss bekannter externer Kräfte \mathbf{f}_{ext} zu bestimmen. Dazu wird folgender Gleichgewichtssatz der Mechanik verwendet:

$$\mathbf{innere\ Arbeit} = \mathbf{äußere\ Arbeit} \quad (2.16)$$

Die innere Arbeit wird durch die elastischen Kräfte formuliert die im inneren des Körpers wirken. Die äußere Arbeit durch die extern angreifenden Kräfte. Beides wird an den Knoten des Körpers bestimmt. Bei den äußeren Kräften ist dies einfach da die externen Kräfte meist schon durch Kraftvektoren, die an den Knoten wirken, angegeben sind. Die elastischen Kräfte im Körper werden berechnet, indem für jeden Knoten im Körper die an diesen Knoten anliegenden elastischen Kräfte aller angrenzenden Tetraeder bestimmt und aufsummiert werden. Die Rückstellkraft einer Feder berechnet sich aus der negativen Ableitung der Verformungsenergie. Analog kann die Elastizitätskraft $\mathbf{f}_{[i]}$, welche ein Tetraeder auf einen seiner Knoten i ausübt durch die über das Tetraedervolumen integrierte partielle Ableitung der Verformungsenergie bzgl. der Position des Knotens dargestellt werden:

$$\mathbf{f}_{[i]} = -A \sum_{j=1}^4 p_{[j]} \sum_{k=1}^3 \sum_{l=1}^3 \beta_{jl} \beta_{ik} \sigma_{kl} \quad (2.17)$$

A ist der Flächeninhalt des Tetraeders und bewerkstelligt die numerische Integration über das Volumen.

$p_{[j]}$ ist die Weltposition des entsprechenden indizierten Knotens des Tetraeders. β ist eine elementbezogene Matrix, welche Weltkoordinaten $\mathbf{p}(\mathbf{u})$ in die entsprechenden baryzentrischen Koordinaten für dieses Element transformiert.

Der Tensor σ , welcher die Spannungen in einem Element repräsentiert, berechnet sich nach dem Hooke'schen Gesetz (siehe Gleichung (2.6)) wie folgt:

$$\sigma_{ij} = \sum_{k=1}^3 \lambda \epsilon_{kk} \delta_{ij} + 2\mu \epsilon_{ij} \quad (2.18)$$

Der Verformungstensor ϵ eines Elements stellt die Verformung im Element dar. Es gibt verschiedene Arten die Verformung in einem Tensor zu repräsentieren: Gängig in der Literatur zur Elastizitätstheorie ist der Greensche Verformungstensor:

$$\epsilon = \mathbf{J}_u(p) \mathbf{J}_u^T(p) - \mathbf{I} \quad (2.19)$$

$\mathbf{J}_u(p)$ ist die Jakobimatrix der Vektorfunktion \mathbf{p} bzgl. des Vektors u . Der Tensor ϵ ist nicht linear.

Stellt man die Gleichung (2.17) um, so entsteht für jeden Knoten des Körpers eine Gleichung mit den über angrenzende Tetraeder benachbarten Knoten als abhängige Variablen. Das Störglied dieser Gleichung ist die an dem Knoten wirkende externe Kraft. Damit ist ein schwach besetztes Gleichungssystem entstanden, mit dessen Lösung die Verschiebung eines jeden Knoten bestimmt werden kann. Da dieses Gleichungssystem nicht linear ist, muss ein entsprechendes Verfahren wie zum Beispiel das Newton-Raphson Verfahren zur Lösung verwendet werden.

Dieses umfangreiche Thema kann durch dieses Kapitel nur oberflächlich skizziert werden. In [OH99], [MMDJ01a] und [JL01] können die theoretischen Hintergründe und Details der finiten Elemente Methode weiter vertieft werden.

2.4.2 Numerische Methoden der Strömungsmechanik

In der Strömungsmechanik sind die kontinuierlichen Variablen neben der Zeit, der Druck und das Vektorfeld der Geschwindigkeiten $v(x)$. In den klassischen Ingenieurwissenschaften wird $v(x)$ ebenfalls durch ein Tetraedergitter diskretisiert. In der Computergraphik ist die Diskretisierung durch ein achsenparalleles Quadrigitter gebräuchlich. Der Grund dafür liegt hauptsächlich in der einfacheren Umsetzung und handhabung, denn die Ableitungen von $v(x)$ können durch einfache Differenzbildung approximiert werden. Deswegen wird diese Methode finite Differenzen Methode genannt.

Die Eigenschaften des Fluids sind in den einzelnen Gitterzellen definiert und konstant. Die Gitterzellen werden durch einen Index ijk eindeutig identifiziert. Die Ableitung dv_{ijk}/dx kann mit zentralen Differenzen angenähert werden:

$$\frac{dv_{ijk}}{dx} = \frac{u_{(i+1)jk} - u_{(i-1)jk}}{2} \quad (2.20)$$

Wichtig hierbei ist, dass die Geschwindigkeitskomponente u_{ijk} der Zelle ijk selbst nicht in die Annäherung zur Ableitung der Geschwindigkeit in dieser Zelle einfließt. Am Beispiel der Funktion

$$\frac{dv_{ijk}}{dx} = 0 \quad (2.21)$$

ist leicht zu sehen dass diese Approximation die in Abbildung 2.10 dargestellte falsche numerische Lösung haben kann, obwohl die einzige korrekte Lösung $u_{ijk} = \text{constant}$ ist. Es entstehen Schwingungen, die geglättet oder anderweitig behandelt werden müssen.

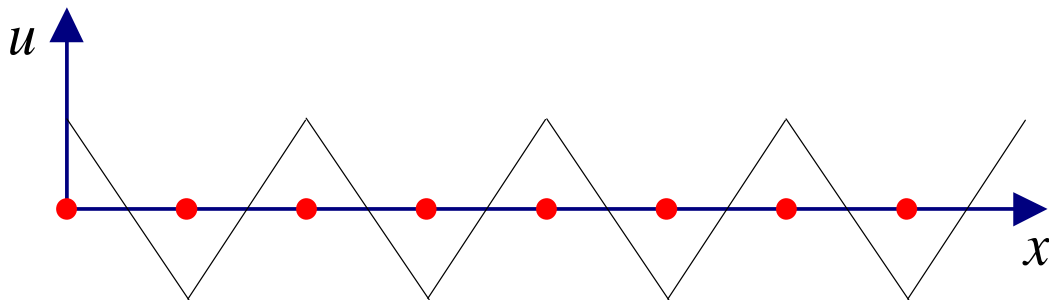


Abbildung 2.10: Falsche numerische Lösung von $du_{ijk}/dx = 0$ durch Annäherung der Ableitung mit zentralen Differenzen

Werden statt der zentralen Differenzen Vorwärtsdifferenzen verwendet, fließt die Geschwindigkeit der Zelle ijk nun mit in die Ableitung ein und die Probleme treten nicht auf:

$$\frac{dv_{ijk}}{dx} = u_{(i+1)jk} - u_{(i)jk} \quad (2.22)$$

Da die Geschwindigkeiten über die Größe einer Zelle konstant sind, entspricht dies einem versetzten Gitter und die Geschwindigkeiten liegen (je nach Art der Differenz) auf den entsprechenden Seitenflächen der Zelle.

Über dieser Diskretisierung und ausgehend von einem Initialzustand werden die Navier-Stokes-Gleichungen gelöst. Dabei wird das so genannte Operator-Splitting durchgeführt, was bedeutet, dass das Integral in seine Einzeltermine zerlegt wird, und diese einzeln gelöst werden. Das führt zu folgenden aufeinander aufbauenden Teillösungen w_i :

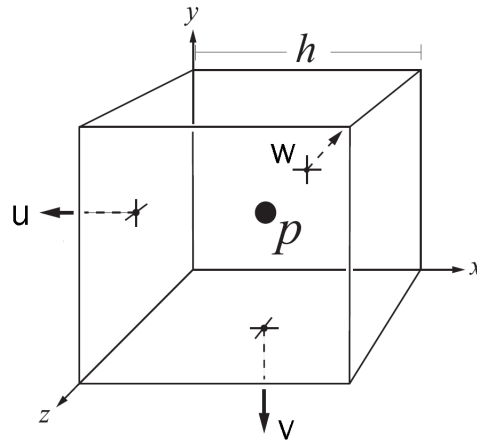


Abbildung 2.11: Versetztes Gitter als Variante der finiten Differenzen Diskretisierung

$$w_0(x) \xrightarrow{\vec{f}} w_1(x) \xrightarrow{-(\vec{v} \cdot \nabla) \vec{v}} w_2(x) \xrightarrow{\mu \nabla^2 \vec{v}} w_3(x) \xrightarrow{P(w_3)} w_4(x) \quad (2.23)$$

Die Pfeile zwischen den Zuständen w_i repräsentieren einen zu lösenden Term der Navier-Stokes Gleichungen. \vec{f} beschreibt die im Fluid wirkenden Volumenkräfte und wird durch einen expliziten Eulerschritt integriert:

$$w_1 = w_0 + \Delta t \vec{f} \quad (2.24)$$

Zur Lösung des Advektionsterm hat sich in der Computergraphik das Semi-Lagrange-Verfahren durchgesetzt welches in [Sta99] eingeführt wurde. Dabei werden Partikel in den Positionen erzeugt, in dem der Term gelöst werden soll, und anhand eines Einschrittverfahrens in der Zeit zurückbewegt um die Position zu ermitteln welche die Partikel vor dem aktuellen Zeitschritt inne hatten. Die Fluideigenschaften an diesen Positionen können dann für den aktuellen Zeitschritt an den Ausgangspositionen der Partikel angenommen werden.

Die Lösung des Reibungsterms kann mit einer Glättung verglichen werden, welche die Diffusion simuliert.

Letztendlich wird das Geschwindigkeitsfeld so korrigiert, dass die Divergenzfreiheit gewährleistet ist. Dabei muss die Poisson-Gleichung

$$\nabla \cdot w_3 = \frac{\delta^2 p}{\delta x^2} + \frac{\delta^2 p}{\delta y^2} + \frac{\delta^2 p}{\delta z^2} \quad (2.25)$$

gelöst werden. Durch Diskretisierung der Differentialoperatoren mit Differenzenbildung und umstellen

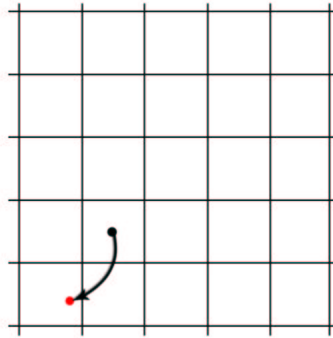


Abbildung 2.12: Semi-Lagrange-Advektion durch Partikelrückverfolgung

von (2.25) kann für jede Zelle eine Gleichung zur Berechnung von p aufgestellt werden. Dabei entsteht ein schwach besetztes Gleichungssystem mit einer Gleichung für jede Zelle, welches mit einem Standardverfahren gelöst werden kann.

Mit den angegebenen Lösungsschritten kann die Navier-Stokes Gleichung für einen Zeitschritt Δt näherungsweise berechnet werden. Dabei ist das Fluid in allen Zellen welche in die Berechnung eingehen vorhanden. Für ein Free-Surface-Fluid ist die Kombination von Fluidzellen und Nichtfluidzellen zusätzlich zu berücksichtigen. Auch die Verfolgung der Fluidmasse bzw. dessen Oberfläche ist dabei ein weiteres Problem. Ausgangsbasis dieser Diplomarbeit war ein Fluidsimulator der verwendet werden sollte. Im folgenden Kapitel wird die Funktionalität und der Aufbau des Simulators kurz vorgestellt, da dieser in Kapitel 3 benötigt und um Funktionalität erweitert wird.

2.4.2.1 Der Fluidsimulator

Der Fluidsimulator, welcher dieser Diplomarbeit zur Verfügung stand, ist ganz im Geiste der Arbeiten von [EMF02], [FM96], [FF01] und [FSJ01] implementiert und ist in der Lage Fluide mit freien Oberflächen zu simulieren. Die Lösung aus [Sta99] steht dem beispielsweise gegenüber. Dort können Eigenschaften innerhalb eines Fluides wie Farbe oder Dichte nur "unter Wasser" verfolgt werden, da die Simulation nur Fluidzellen und Randzellen kennt. Ein Algorithmus zur Simulation von freien Fluidoberflächen kennt darüber hinaus auch Luft- bzw. Vakuumzellen und ist in der Lage die Masse des Fluids im Raum zu verfolgen, da nicht mehr angenommen werden kann das der ganze Raum mit Fluid gefüllt ist. Bis auf diesen Aspekt arbeitet der Algorithmus allerdings genauso, wie in Abschnitt 2.4.2 beschrieben.

Zur Oberflächen bzw. Fluidverfolgung gibt es in der Computergraphik zwei grundlegende Verfahren. Zum einen die Level-Set Methode welche Ende der 90iger Jahre für den Bereich der Fluidodynamik entdeckt wurde. Kerngedanke ist die mathematische Beschreibung der Fluidoberfläche durch eine vorzeichenbehaftete Distanzfunktion welche ebenfalls auf dem Gitter diskretisiert wird. Die Fluidzellen sind

durch negative Entfernungen zur Oberfläche gekennzeichnet.

Die alternative Methode, welche 1965 von Harlow et. al. in [HW65] auf den Weg gebracht wurde, diskretisiert die Fluidmasse durch eine Partikelwolke. Alle Fluidzellen sind dadurch gekennzeichnet, dass sich mindestens ein Fluidpartikel in der Zelle befindet. Diese Partikel markieren das Fluid im Simulationsraum und werden daher auch Markerpartikel genannt. Das Verfahren selbst heist Marker-and-Cell Method (MAC).

In beiden Verfahren hat die Oberflächenrepräsentation keinen Einfluss auf die Lösung der Gleichungen. Bei beiden Methoden muss die Darstellung nach Lösung der Gleichung aktualisiert werden. Das passiert bei der MAC-Methode durch Advektion der Partikel durch das Geschwindigkeitsfeld (welches auch außerhalb der Fluidzellen definiert ist). Bei den Levelsets wird ebenfalls eine Advektion durchgeführt und eine Reihe von Nachbearbeitungs- und Fehlerbehandlungsschritten, da die Advektion der Levelsets leicht zu Inkonsistenzen führen kann.

Im Fluidsimulator, welcher der Diplomarbeit zur Verfügung stand, wurde die MAC-Methode verwendet. Zur Veranschaulichung wird der Algorithmus zur Lösung eines Zeitschrittes in Abbildung 2.13 angegeben.

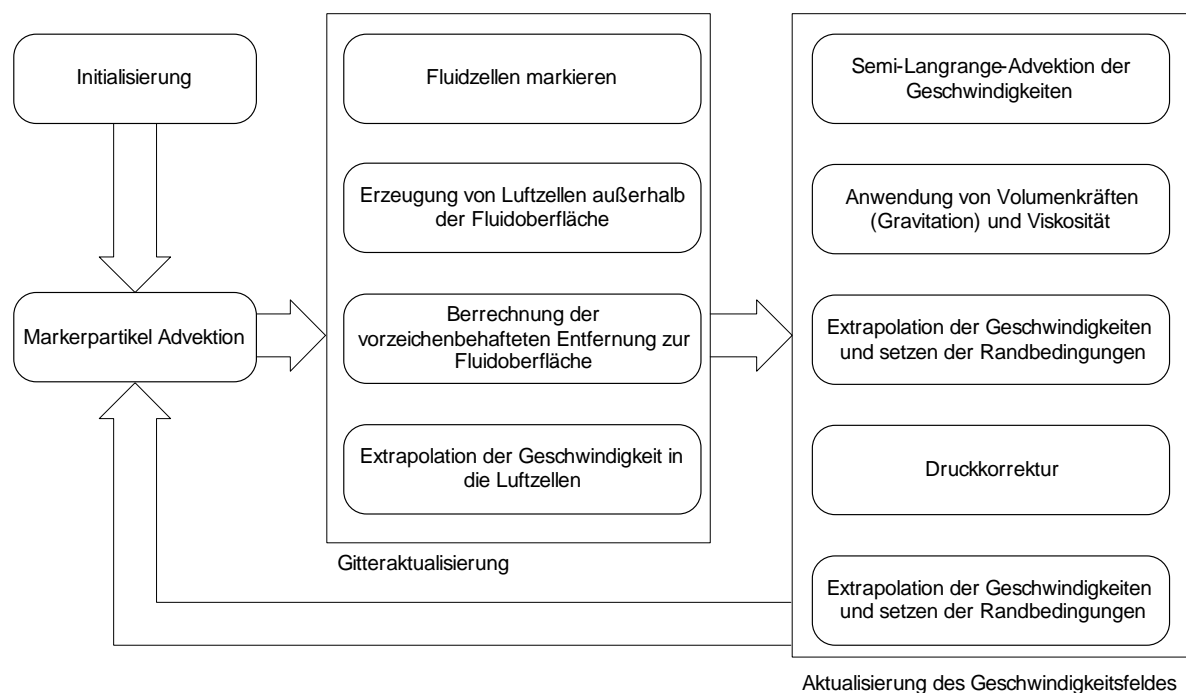


Abbildung 2.13: Der Algorithmus des Fluidsimulators in schematischer Ansicht

Aus der Oberflächenrepräsentation muss für die Visualisierung ein Dreiecksnetz erzeugt werden. Dazu muss aus den Marker-Partikeln eine Distanzfunktion errechnet werden, mit welcher dann ein Algorithmus zur Tessellierung gespeist werden kann. Im vorhandenen Fluidsimulator kam hier eine in [ZB05]

vorgestellte Berechnung zum Einsatz, welche einen Distanzwert für einen Punkt im Raum aus den in der Nähe befindlichen Partikeln berechnet. Tesseliert wurde diese Funktion dann mit dem Marching Cubes Algorithmus.

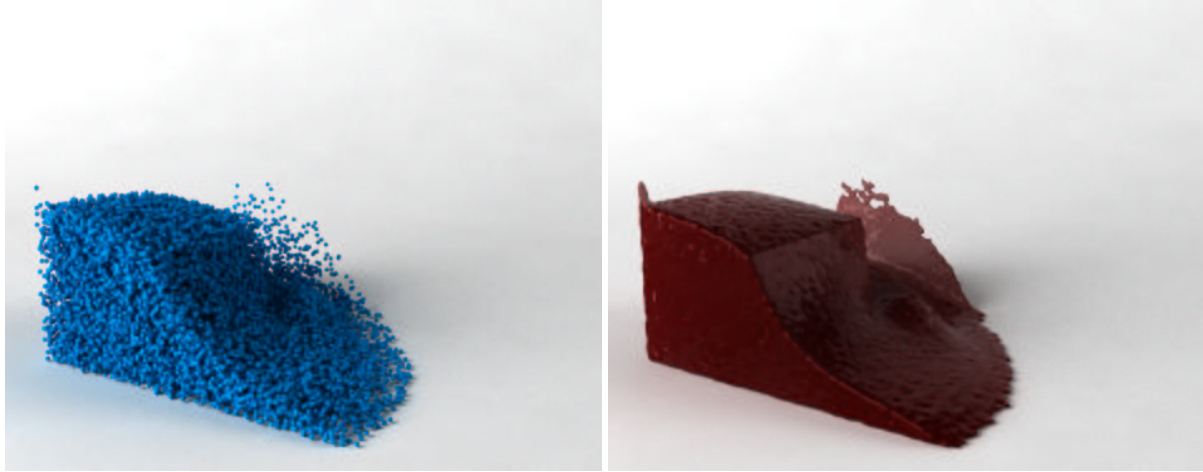


Abbildung 2.14: Marker-Partikel des Fluidsimulators (*links*) und die nach [ZB05] erzeugte Dreiecksnetzrepräsentation (*rechts*)

2.5 Verwandte Arbeiten

Dieser Abschnitt bietet einen Überblick über wissenschaftliche Arbeiten, die für das Thema dieser Diplomarbeit relevant sind. Das sind im besonderen Arbeiten zu Themen starr-plastischer Phänomene wie Lava oder Gletscher. Weiter sind Arbeiten im Bezug auf Bruchbildung interessant.

Starr-plastische Phänomene wie Lava und Gletscher haben in der Computergraphik bisher noch keine große Aufmerksamkeit erfahren. In [SAC⁺99] wird versucht die charakteristische Fließeigenschaft von Lava durch einen Smoothed Particle Hydrodynamics ([Kel06]) basierten Algorithmus anzunähern. Die durch Brüche in den durch Abkühlung verfestigten Bereichen an der Oberfläche des Lavastroms charakterisierten Muster, werden durch eine prozedurale Textur erzeugt. Weitere Publikationen zum Thema finden sich in den Geowissenschaften. Diese untersuchen vor allem die Physik der Bruchbildung und das Fließverhalten von Lavamassen. [Spi20] bietet einen interessanten Einblick in jene Problemsphären. Die Darstellung visueller Eindrücke wird allerdings nicht berührt.

In [LM98] wird das Phänomen des Erdbebens simuliert. Beim Erdbeben handelt es sich um eine Bruchbildung an Hängen oder Abfällen, bei dem sich ein massiver Teil an Gestein, Geröll oder Schlamm löst und abgleitet. Zur Beschreibung des Terrains wird eine Heightmap verwendet. Große Steine werden als Starrkörper simuliert. Das auseinander brechen dieser Steine wird erzeugt in dem die Gesteinsbrocken vor dem Simulationsstart manuell zerbrochen werden. Beim Beginn der Simulation sind die vorgebro-

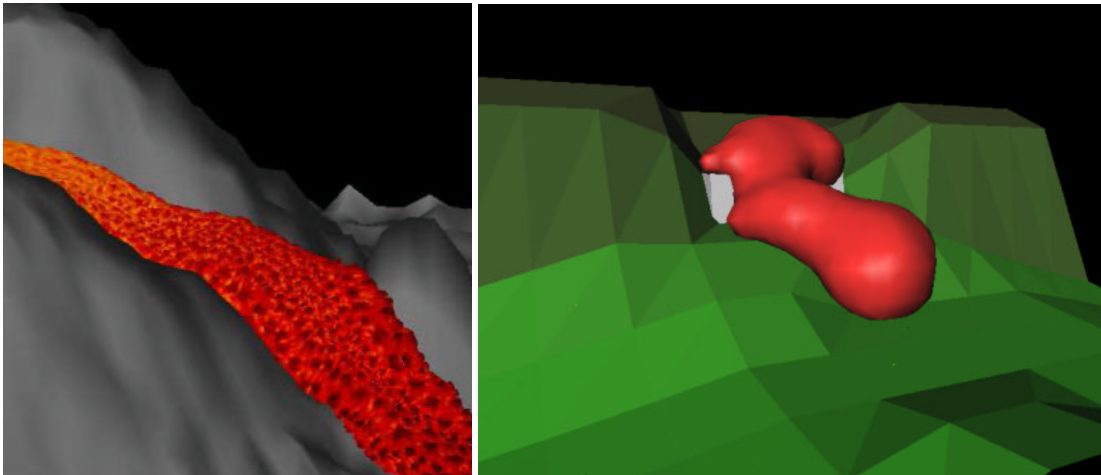


Abbildung 2.15: Ergebnis aus [SAC⁺99](links) und [LM98](rechts)

chenen Teile durch eine starre Federkraft miteinander verbunden. Beim Überschreiten einer bestimmten Spannung wird diese Feder gelöscht und der Stein fällt auseinander. Bei der Kollision mit dem Boden wird das elastisch-plastische Verhalten des Bodens berücksichtigt. Für die Modellierung des Schlammes in Abbildung 2.16 rechts, kamen ebenfalls die Smoothed Particle Hydrodynamics zum Einsatz. Auch hier finden die Brüche, bis auf den pre-fracturing Mechanismus der Steine, keine Beachtung. Die Erdrutschmasse selbst ist schon bei Simulationsbeginn vom Terrain gelöst.

Zur Simulation von Gletschern insbesondere zusammen mit der Gletscherspaltenbildung gibt es in der Computergraphik noch keine Arbeiten. Wissenschaftliche Werke auf diesem Gebiet finden sich im Fachbereich der Bodenmechanik und der Geologie. Diese setzen sich primär mit dem Fließverhalten der Gletscher auseinander. Hierbei wird eine möglichst exakte Voraussage über das Verhalten des Gletschers angestrebt. Vorort erhobene Messdaten spielen dabei immer eine wichtige Rolle. Entweder um die berechneten Daten zu validieren oder um Berechnungen auf Basis dieser Daten durchzuführen. So zum Beispiel in [Hed04] und [Hed99], wo die Force-Budget Methode verwendet wurde um das Fließverhalten von Gletschern zu berechnen. Die Force-Budget Methode ist ein Ansatz zur Bestimmung der Spannungen und Verformungen innerhalb des Gletschers aufgrund von Messdaten an der Gletscheroberfläche. Das gewünschte Maß an Exaktheit geht weit über das hinaus, was für diese Diplomarbeit angestrebt ist. Zur visuellen Reproduktion der optischen Phänomene von Gletschern leisten diese Arbeiten zudem keinen Beitrag.

Die wenigen themanahen Veröffentlichungen in der Computergraphik beschäftigen sich vorrangig mit dem Fließverhalten der simulierten Medien und ignorieren den Effekt der Bruchbildung. Zum Fließverhalten im allgemeinen sind dem gegenüber bereits eine Reihe von Arbeiten erschienen. Exemplarisch sei hier auf [GBO04], [CMRBVHT02], [Car04] und [MKN⁺04] verwiesen. Diese Arbeiten präsentieren

Erweiterungen und Modifikationen bekannter Algorithmen und Verfahren um das große Spektrum der verschiedenen Flusseigenschaften zu simulieren.

Das Thema der Bruchbildung alleine ist auch bereits Gegenstand intensiver computergraphischer Betrachtung gewesen. In [TF88] wurde das reißen eines Masse-Feder Netzes simuliert, indem die Federn bei Überschreitung eines elastischen Limits gelöscht wurden. Neff et. al. haben diesen Ansatz in [NF99] für Starrkörper, die durch Federn fest miteinander verbunden sind, erweitert und konnten somit zerbrechende, vorgebrochene Objekte simulieren. O'Brien et. al. gelang 1999 in [OH99] der Durchbruch, indem er die Methode der finiten Elemente aus den klassischen Ingenieurwissenschaften (siehe Kapitel 2.4.1) um Bruchkriterium und Bruchmodellierung erweiterte. In Kombination mit einem Starrkörpersimulator wurden so sehr spektakuläre Ergebnisse erzielt. Dieses Verfahren ist in seiner Grundform der de-facto Standard zur Bruchmodellierung in der Computergraphik. Eine Optimierung zu einem Echtzeitverfahren ist in [MMDJ01b] veröffentlicht.

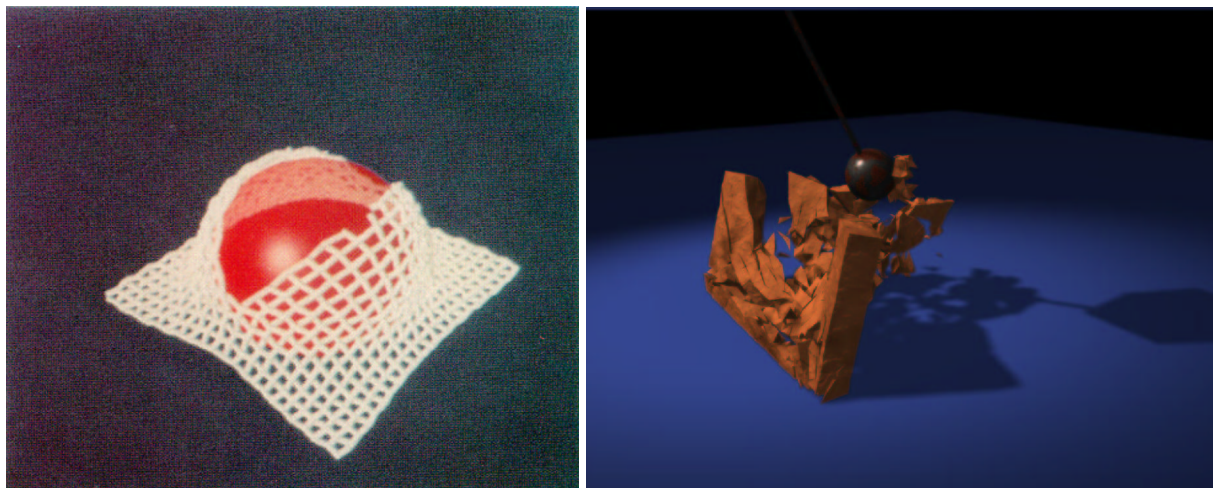


Abbildung 2.16: Ergebnis zur Bruchbildung aus [TF88](links) und [OH99](rechts)

Im folgenden Teilkapitel wird die Publikation von [OH99] genauer vorgestellt. Dies deshalb, da sie für das theoretische Verständnis des Kapitels 3 von großer Bedeutung ist und zudem gut zeigt, wie die Theorie zur Bruchbildung aus Kapitel 2.2 praktisch umgesetzt werden kann.

2.5.1 Modellierung und Animation von Bruchbildung in steifen Körpern

Der in [OH99] vorgestellten Methode liegt das in Kapitel 2.4.1 erklärte FEM-Verfahren zugrunde. Der Körper wird durch ein Tetraedernetz diskretisiert und die Verschiebungen der Knoten werden auf Basis einer Erhaltungsgleichung und dem Hookeschen Gesetz berechnet. Nachdem diese Verschiebungen in einem Zeitschritt ermittelt wurden, wird die Auswertung eines Bruchkriteriums in das Verfahren ein-

gefügt. Dieses Bruchkriterium wird an den Knoten ausgewertet. Gemäß Kapitel 2.5.2.2 sind dazu die Spannungen σ in diesem Punkt notwendig. Diese Spannungen sind allerdings nur über die Volumina der Tetraeder gegeben und an den Positionen der Knoten nicht definiert. Um die Spannungen an den Knoten zu ermitteln können die Spannungen der angrenzenden Tetraeder gemittelt werden. Ein einfaches Bruchkriterium könnte den größten Eigenwert des so ermittelten Spannungstensors mit einem Materialhärte- wert vergleichen. Bei Überschreitung wäre ein Bruch erkannt (siehe [Bri04]). Dieses Bruchkriterium entspricht einem einfachen Bruchkriterium auf Basis der Hauptnormalspannungshypothese (Rankine- Vergleichsspannung):

$$\sigma_v = \max(\sigma_1, \sigma_2, \sigma_3) \quad (2.26)$$

Das Bruchkriterium in [OH99] unterscheidet sich zu dem eben vorgestellten nur in der Art und Weise wie der Spannungstensor in den Knoten berechnet wird. O'Brien verwendet eine etwas komplexere Methode, in welcher die Spannungen, der angrenzenden Tetraeder, in ihre Zug- und Druckanteile σ^+ und σ^- zerlegt werden:

$$\sigma^+ = \sum_{i=1}^3 \max(0, v^i(\sigma)) m(\hat{\mathbf{n}}^i(\sigma)) \quad (2.27)$$

$$\sigma^- = \sum_{i=1}^3 \min(0, v^i(\sigma)) m(\hat{\mathbf{n}}^i(\sigma)) \quad (2.28)$$

$v^i(\sigma)$ sind die Eigenwerte oder auch Hauptspannungen von σ und $\hat{\mathbf{n}}^i(\sigma)$ die dazu gehörigen Eigenvektoren. Durch Substitution von σ mit σ^+ bzw. σ^- in Gleichung (2.17), wird $\mathbf{f}_{[i]}^+$ und $\mathbf{f}_{[i]}^-$ entsprechend berechnet.

Danach wird aus diesen zerlegten Kräften ein Spannungstensor ζ in den Knoten neu zusammengestellt. Dabei werden alle wirkenden Kräfte, welche für einen Überschuss an Zug- bzw. Druckkraft sorgen, nicht mehr berücksichtigt. Dies stellt sicher, dass im finalen Spannungstensor ζ jeder Spannung eine entgegengesetzt wirkende Spannung vorliegt. Das Material versagt also nicht mehr wegen zu hoher Spannung alleine, sondern wegen einem zu hohen wirken zweier entgegengesetzter Spannungen. Dieser Separationstensor wird folgendermaßen berechnet:

$$\zeta = \frac{1}{2} \left(-m(f^+) + \sum_{f \in \{f^+\}} m(f) + m(f^-) - \sum_{f \in \{f^-\}} m(f) \right) \quad (2.29)$$

Die Funktion $m(\vec{x})$ ist das dyadische Produkt des Argumentvektors mit sich selbst und erzeugt somit eine Matrix bzw. einen Tensor der den Argumentvektor als Eigenvektor mit dessen Betrag als Eigenwert

enthält. f ist die Summe aller anliegenden inneren Kräfte und f^+ bzw. f^- der Anteil an Zug- bzw. Druckspannung.

Nachdem ζ in jedem Knoten berechnet und eine Eigenwertanalyse durchgeführt wurde, wird der größte positive Eigenwert mit einem Materialhärtewert τ verglichen. Wird dieser überschritten, so kommt es zum Materialversagen und der Knoten bricht. Die Beschränkung auf positive Eigenwerte hat zur Folge das nur Brüche aufgrund von Zugspannungen gebildet werden.

Bricht ein Knoten, so werden alle angrenzenden Tetraedernetze wie in Abbildung 2.17 gespalten. Die Ebene an der gespalten wird, ist durch den zugehörigen Eigenvektor gegeben. Ein abschließendes Remeshing stellt sicher, dass alle neuen Elemente ebenfalls Tetraeder sind.

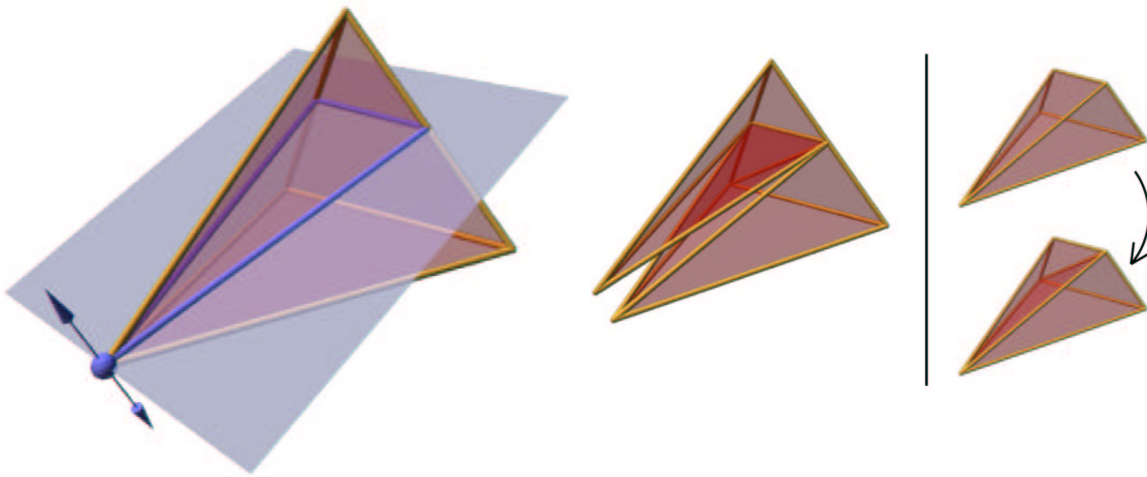


Abbildung 2.17: Bruchmodellierung durch Spaltung eines Tetraeders (*links*) mit anschließendem Remeshing (*rechts*)

Damit ist eine grobe Skizze des Verfahrens gegeben.

2.5.2 Sand als Fluid

Eine wichtige verwandte Arbeit ist die Publikation [ZB05] in welcher die Animation von Sand die primäre Zielstellung ist. Der Arbeit zugrunde liegt die Hypothese, dass die Dynamik von Gemengen mit hinreichend feiner Granularität immer mehr einem reibungsbehafteten plastischem Fluss gleichen. Ausgehend von dieser Annahme werden nun Methoden aus der Fluidodynamik eingesetzt um das Verhalten von Sand zu simulieren. Grundlage ist hier ebenfalls der in Kapitel 2.4.2.1 vorgestellte Fluidsimulator. Der Fluidsimulator wurde im Rahmen dieser Diplomarbeit um den in [ZB05] vorgestellten Algorithmus erweitert.

2.5.2.1 Sandmodell

Bei dem zugrunde liegenden Sandmodell werden einige wichtige Annahmen und Vereinfachungen getroffen auf die nun kurz im einzelnen eingegangen wird.

Beim Modell wird von einem homogenen einphasigen Material ausgegangen. In der Realität existiert in den Zwischenräumen der Sandkörner Luft, Wasser oder ein anderes Medium. Dieses hat mitunter erheblichen Einfluss auf das Fließverhalten des gesamten Materials. Hier wird dieser zwei-phasige Zustand ignoriert.

Da ein Fluidsimulator eingesetzt wird gilt die Kontinuumshypothese welche besagt, dass bei immer kleiner werdendem Kontrollvolumen die Ableitungen der Fluideigenschaften, genauer der Geschwindigkeiten und des Drucks, immer kleiner werden. Es gilt also das die Eigenschaften des Fluids als stetig angenommen werden. Das bedeutet das intermolekulare Wechselwirkungen und die Bewegung einzelner Moleküle nicht untersucht werden.

Es gilt ebenfalls die Annahme das Sand einem starr-plastischem Fluss gleicht. Das also das Fluid sich am Ort wie ein Starrkörper verhält und dann sprunghaft in einen Zustand des plastischen fließens übergeht. Das heist das die geringen Elastizitätskräfte im Sand vernachlässigt werden.

Des Weiteren wird vereinfachend angenommen das der Druck im Fluid mit dem im Sand identisch ist. Dies ist tatsächlich nicht der Fall. Während bei einer Fluidsäule der Druck mit der Tiefe linear steigt, steigt der Druck in einer gleichen Säule aus Sand zuerst stärker und mit zunehmender Tiefe immer weniger stark an. Dies passiert aufgrund von so genannten Spannungsketten die sich durch die Kornkontakte ausbilden und dafür sorgen das der Druck wie bei einem Gewölbe in gotischen Kathedralen auf die Seitenwände abgeleitet wird. Was zu einer Entlastung in den tieferen Regionen führt.

Das die bei granularer Materie vorhandene Dilatanz nicht berücksichtigt wird ist ebenfalls eine wichtige Vereinfachung des Sandmodells. Dilatanz ist eine Eigenschaft die besagt, dass sich verdichtete körnige Materie nur dann verformen kann wenn sich gleichzeitig mit einer Scherkraft auch eine Auflockerung der Materie einstellt. Dies ist so zu verstehen das die einzelnen Körner Spielraum brauchen um sich aneinander vorbei bewegen zu können. Die Vereinfachung des Modells so das Dilatanz nicht berücksichtigt wird, hat zur Konsequenz, dass nur Scherbrüche modelliert werden und keine Dehnungsbrüche Beachtung finden.

Diese Annahmen ermöglichen es, den Fluidsimulator als Grundlage zur Simulation zu verwenden. Eine Erweiterung des Fluidsimulators für Sandsimulation besteht aus drei Teilen. Zuerst muss ein Kriterium eingeführt werden mit dem starre bzw. plastische Bereiche innerhalb des Fluids bestimmt werden können. Danach müssen diese beiden Bereiche gesondert behandelt werden.

2.5.2.2 Bruchkriterium

Als Bruchkriterium wird das Mohr-Coulomb Bruchkriterium verwendet, welches in der Bodenmechanik oft genutzt wird um die Dynamik granularer Gemenge zu beschreiben. Der Normaldruck σ_n ist im Kontinuum des Fluids durch den hydrostatischen Druck p gegeben, welcher durch den Fluidsimulator berechnet wird. Das klassische Mohr-Coulomb Bruchkriterium basiert auf der Tresca-Vergleichsspannung (größte Differenz der Hauptspannungen). In [ZB05] wird stattdessen die von-Mises Vergleichsspannung verwendet, um den mehrachsigen Spannungszustand auf einen einachsigen Spannungszustand zu reduzieren.

Der dazu benötigte Spannungstensor ist definiert als die Summe aus Normal- und Scherspannungen:

$$\sigma = -p\mathbf{I} + \tau \quad (2.30)$$

Der erste Teil der Summe ist durch den Druck p im Fluid bereits ermittelt. Was fehlt ist der Tensor der Scherspannungen τ . Diesen zu finden ist problematisch da ein Fluid per Definition eine Substanz ist, welche einer beliebig kleinen Scherspannung keinen Widerstand entgegen setzt. Sobald also im Fluid Scherspannungen anliegen werden diese sofort umgesetzt und damit aufgehoben.

Spannungen verursachen Deformationen. Wäre in jedem Ort innerhalb des Fluids die Verformung ϵ gegeben, könnte daraus die Scherspannung berechnet werden, da zwischen Verformung und Spannung ein gesetzmäßiger Zusammenhang besteht. Im Kapitel 3.2 später dazu mehr.

Im Fluidsimulator wird die Verformung nicht verfolgt aber aus dem Geschwindigkeitsfeld lässt sich ermitteln wie sich die Deformation im aktuellen Zeitschritt ändert. Die Änderung der Deformation im Zeitschritt t ist die Verformungsgeschwindigkeit und wird durch die symmetrisierte Jakobimatrix innerhalb des Vektorfeldes der Geschwindigkeit im Fluid bestimmt:

$$D = \frac{1}{2}(\nabla u + \nabla u^T) \quad (2.31)$$

In [ZB05] wird die Verformungsgeschwindigkeit verwendet um daraus die Vergleichsspannung für die Scherspannung im Fluid zu berechnen. Dies ist möglich da empirisch gezeigt wurde das es einen proportionalen Zusammenhang zwischen Spannung und Verformungsgeschwindigkeit gibt. Grundlage ist dabei der Newtonsche Zähigkeitsansatz. Nach diesem sind die Teilchen in zähen Flüssigkeiten stärker aneinander gebunden. Wirkt nun eine Verformung in einer hohen Geschwindigkeit, werden diese Bindungen stärker beansprucht was im Kontinuum zu proportional höherer Spannung führt. Mit dieser Vergleichsspannung kann dann das Mohr-Coulomb Bruchkriterium bei gegebenen Materialkonstanten Kohäsion c

und Reibungswiderstand μ für jede Fluidzelle im Simulationsgitter ausgewertet werden.

Die Zellen werden anhand dieser Auswertung als stabil (kein Bruch) oder instabil (Bruch) gekennzeichnet.

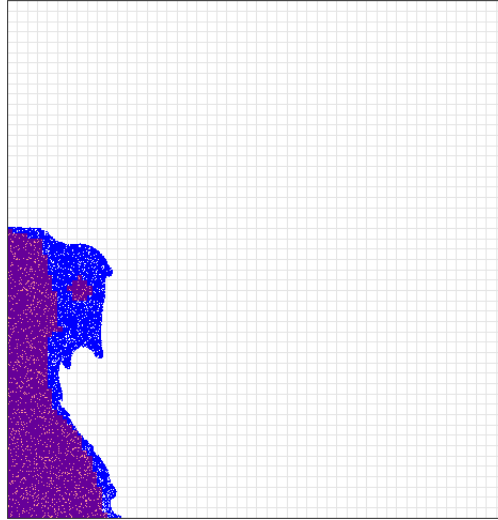


Abbildung 2.18: stabile(rot) und instabile(blau) Bereiche nach Auswertung des Bruchkriteriums

2.5.2.3 Starrbereiche

Nach Auswertung des Kriteriums sind nun, wie in Abbildung 2.18 zu sehen, starre Bereiche \mathbb{R} und plastische Bereiche \mathbb{F} gefunden worden. Dabei gilt für den Simulationsbereich \mathbb{C} das $\mathbb{C} = \mathbb{R} \cup \mathbb{F}$. Weiter ist $\mathbb{R} = \bigcup \mathbb{R}_i$. \mathbb{R}_i sind die einzelnen voneinander getrennten Starrkörperbereiche die entstehen können. Jeder einzelne Starrkörperbereich ist ein Verband aus starren Zellen der nicht mit starren Zellen anderer Starrkörper verbunden ist. \mathbb{R}_i kann über einen Füllalgorithmus ermittelt werden.

Für \mathbb{R} muss nun Starrkörperverhalten hergestellt werden. Dies geschieht mit der in [CMT04] vorgestellten Rigid Fluid Methode deren Idee es ist, die Starrkörper zuerst so zu behandeln als wären sie genauso wie ein Fluid und dann in einem weiteren Schritt analog zur Druckkorrektur einen Projektionsschritt durchzuführen welcher die Geschwindigkeiten in \mathbb{R} in den Raum der Starrkörperbewegung abbildet. Diese Rigiditätsbedingung wird mit Hilfe eines Deformationsoperators $\mathbf{D}[\mathbf{u}]$ ausgedrückt. Dabei handelt es sich um die Verformungsgeschwindigkeit aus Gleichung (2.31). Dieser symmetrische 3×3 Tensor misst die räumliche Deformation des Geschwindigkeitsfeldes \mathbf{u} . Damit kann die Rigiditätsbedingung durch

$$\mathbf{D}[\mathbf{u}] = 0 \quad \text{in } \mathbb{R} \quad (2.32)$$

ausgedrückt werden. Diese Bedingung fordert, dass sich die Geschwindigkeiten in \mathbb{R} wie bei der Starr-

körperbewegung durch eine Translationsgeschwindigkeit \mathbf{v} und eine Winkelgeschwindigkeit ω ausdrücken lassen:

$$\mathbf{u} = \dot{\mathbf{x}} = \mathbf{v} + \omega \times \mathbf{r}_x \quad \text{in } \mathbb{R} \quad (2.33)$$

Der Vektor \mathbf{r}_x ist ein Vektor vom Masseschwerpunkt des Starrkörpers zum Ort \mathbf{x} .

Seien $\mathbf{u}_{\mathbb{R}}$ die Geschwindigkeiten in \mathbb{R} für welche die Gleichung (2.32) erfüllt ist. Es gilt

$$\mathbf{u}_{\mathbb{R}} = \bigcup_{\mathbb{R}_i} (\hat{\mathbf{v}}_i + \hat{\omega}_i \times \mathbf{r}_i). \quad (2.34)$$

$\hat{\mathbf{v}}_i$ ist die Translationsgeschwindigkeit und $\hat{\omega}_i$ die Winkelgeschwindigkeit des Starrkörpers \mathbb{R}_i . Diese können direkt ermittelt werden:

$$M_i \hat{\mathbf{v}}_i = \int_{\mathbb{R}_i} \rho_i \hat{\mathbf{u}} \quad (2.35)$$

$$I_i \hat{\omega}_i = \int_{\mathbb{R}_i} \mathbf{r}_i \times \rho_i \hat{\mathbf{u}} \quad (2.36)$$

Die Integrale können einfach durch Aufsummierung über alle Zellen eines Starrkörpers \mathbb{R}_i berechnet werden. Um die Bedingung (2.32) zu erfüllen werden die Geschwindigkeiten \mathbf{u} in \mathbb{R} durch die Geschwindigkeiten $\mathbf{u}_{\mathbb{R}}$ ersetzt.

2.5.2.4 Plastische Bereiche

Nachdem nun die starren Bereiche behandelt wurden müssen noch die plastisch fließenden Bereiche bearbeitet werden um eine plausible Sandanimation zu erzeugen. Unbehandelt fließen diese bisher wie ein ideales Fluid was zur Bildung von Spritzern und turbulenten Bewegungen führt - für Sand untypische Phänomene. Für den plastischen Bereich muss daher berücksichtigt werden, dass es bei Sand zu einer inneren Reibung kommt welche dadurch entsteht das die einzelnen Sandkörner miteinander kollidieren und sich aneinander reiben wenn sie sich vorbei bewegen.

Die Verformungsgeschwindigkeit aus Gleichung (2.31) gibt an wie die Materie in einem Punkt aneinander reibt. Die Stärke der Reibung wird vom Reibungskoeffizienten μ und dem Druck p beeinflusst. Je mehr Druck bzw. je höher der Reibungskoeffizient, desto höher ist die innere Reibung. Das führt zu folgender Gleichung:

$$\sigma_f = -\mu p \frac{D}{|D|} \quad (2.37)$$

Zur Berücksichtigung der inneren Reibung muss das Geschwindigkeitsfeld folgendermaßen aktualisiert werden:

$$\mathbf{u}_{neu} = \mathbf{u} + \Delta t / \rho \nabla \cdot \sigma_f \quad (2.38)$$

Dies entspricht einem expliziten Eulerschritt mit Δt als Zeitschritt.

2.5.2.5 Algorithmus

Im diesem Teilkapitel soll nun der von [ZB05] veröffentlichte Algorithmus zusammengefasst werden. Ausgehend von der in Abbildung (2.13) gezeigten Arbeitsweise des Fluidsimulators wird nun die Aktualisierung des Geschwindigkeitsfeldes \mathbf{u} um einige Schritte erweitert.

Zuerst werden alle bisherigen Schritte der Fluidsimulation einschließlich Druckkorrektur durchgeführt.

Für jede Fluidzelle wird mit Hilfe der Verformungsgeschwindigkeit unter Berücksichtigung der Zellgröße und der Größe des Zeitschrittes ein Spannungstensor berechnet. Von diesem wird dann die von Mises Vergleichsspannung ermittelt. Wenn diese kleiner ist als τ_{krit} , welches durch das Bruchkriterium gegeben ist, wird die Fluidzelle als starr markiert.

Die als starr markierten Zellen werden in die verschiedenen disjunkten Bereiche aufgeteilt. Ein Füllalgorithmus wird verwendet um über Zellnachbarschaften zusammenhängende starre Bereiche zu finden.

Für jeden einzelnen starren Bereich wird dann nach Gleichung (2.35) und (2.36) die Translations- und Rotationsgeschwindigkeit berechnet und das Geschwindigkeitsfeld entsprechend angepasst um jegliche Deformation in diesen Bereichen zu verhindern.

Zuletzt wird die innere Reibung mittels Gleichung (2.38) in allen Fluidzellen angewandt welche nicht als starr markiert wurden. Dabei ist zu beachten das es sich um einen einfachen expliziten Euler-Schritt handelt, was bei einem zu hohen Zeitschritt bzw. bei zu hohem Reibungskoeffizienten zu Instabilität in der Simulation führen kann.

Die Erweiterungen zur Sandsimulation sind in Abbildung 2.19 schematisch dargestellt.

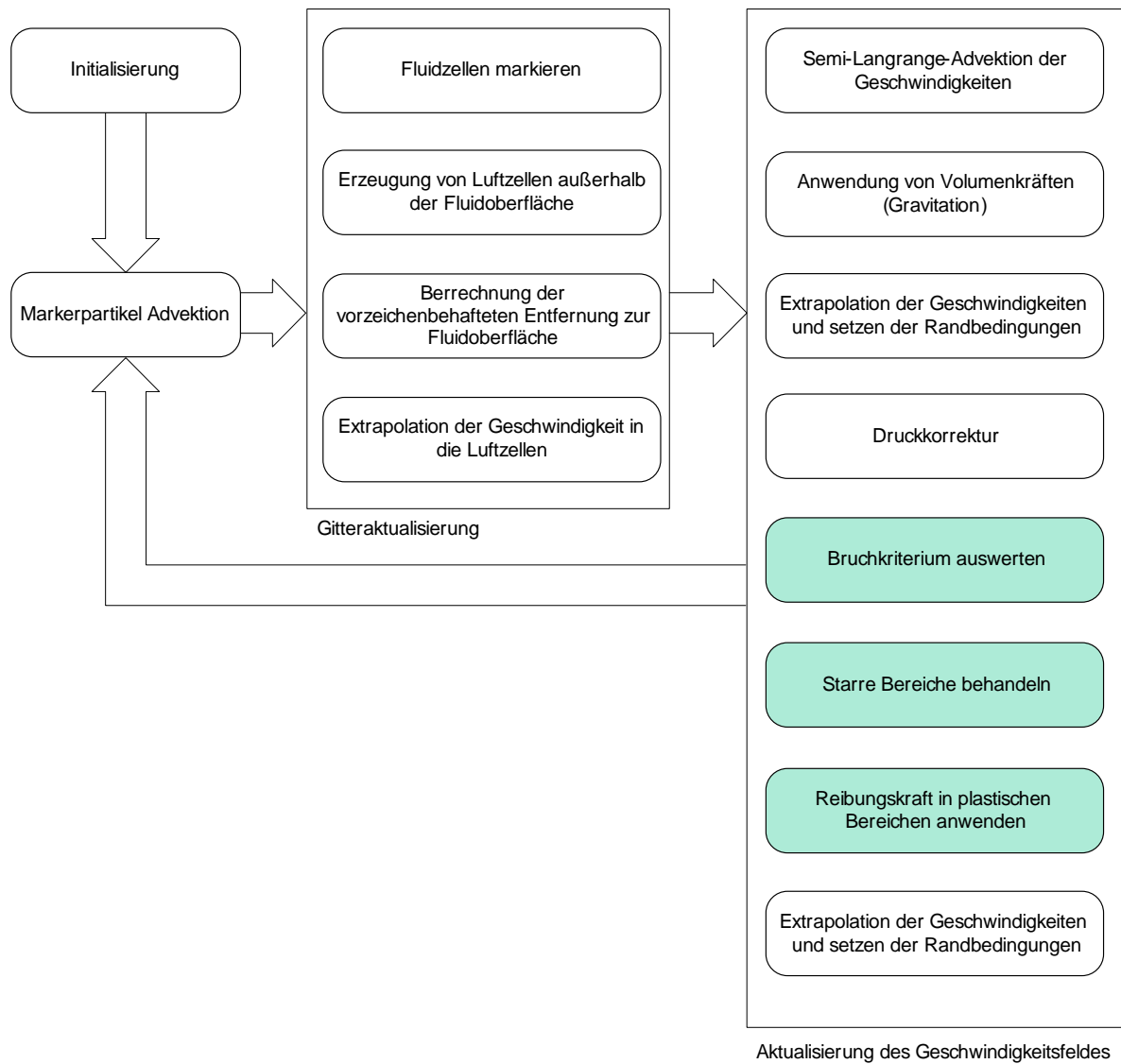


Abbildung 2.19: schematische Ansicht des Fluidsimulators und den Erweiterungen zum Sandsimulator

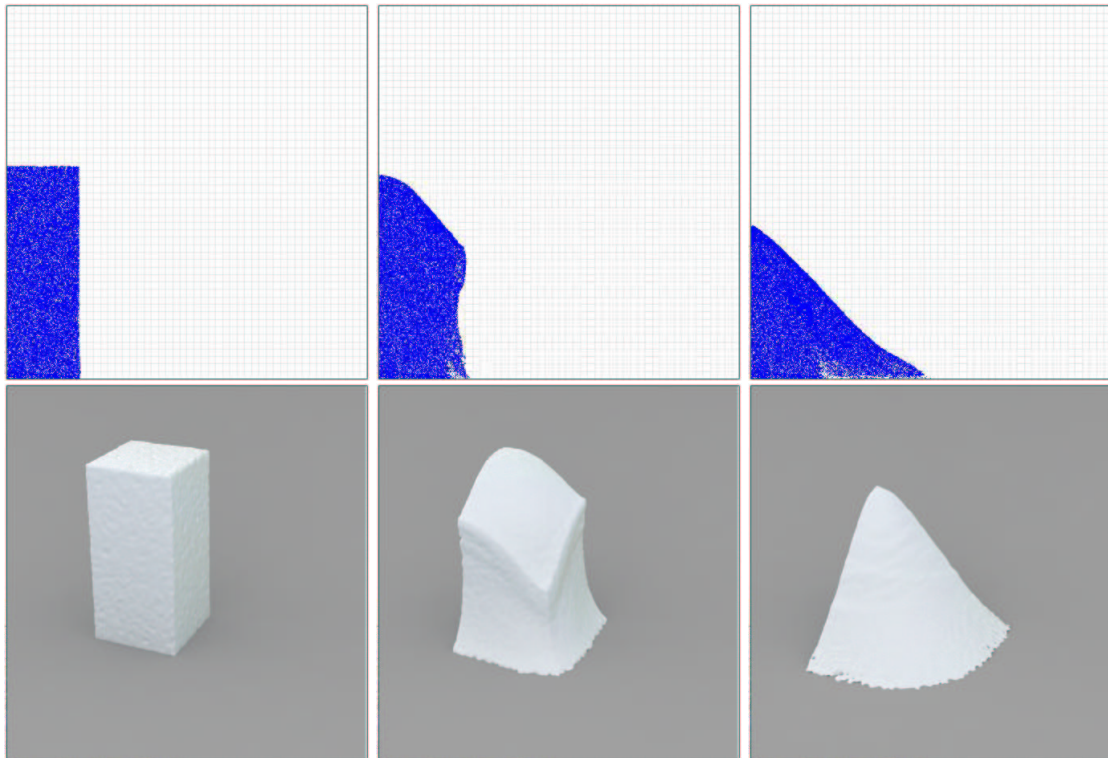


Abbildung 2.20: Ergebnisse des Sandfluidsimulators

2.5.2.6 Ergebnisse

Wie in den Ergebnissen in Abbildung 2.20 zu sehen, erzeugt die Berücksichtigung des Bruchkriteriums und der inneren Reibung den Eindruck von granularem Material. Durch die Verfestigung an entsprechenden Stellen in der Simulation wird der Charakter von Sand glaubhaft erzeugt. Wichtig hierbei ist, dass die Brüche, also die Stellen wo starre Bereiche in plastische Bereiche übergehen, nur implizit durch die unterschiedliche Behandlung von starren und plastischen Zellen erzeugt werden. Es gibt keine direkte Behandlung des Fluids am Bruchort. Zudem ist festzuhalten das aufgrund der Vernachlässigung der Dilatation bei der Bruchbildung nur Scherbrüche ausgebildet werden bei denen sich die Bruchteile entlang der Bruchebene aneinander vorbei bewegen und nicht voneinander weg.

Eine sorgsame Wahl der Simulationsparameter ist zu beachten da sonst die Simulation aufgrund des expliziten Euler-Schritts bei Anwendung der inneren Reibung instabil wird.

Bei einer Kohäsionskonstanten von $c = 0$ im Bruchkriterium kommt die Simulation nicht zur Ruhe weil kleinere numerische Fehler immer zu Bewegungen auf der Sandoberfläche führen. [ZB05] schlagen daher auch für kohäsionslose Materialien, wie reiner und trockener Sand eines ist, eine Kohäsionskonstante $c > 0$ vor, was zu einer Beruhigung der Simulation führt.

Weiterhin Berücksichtigt werden muss die Reibung zwischen Sand und soliden Körpern bzw. dem Rand

der Simulationsdomäne, was in [ZB05] genauer beschrieben wird.

3 Bruchbildung auf starren Fluidoberflächen

Nachdem nun die Bruchbildung von verschiedenen Perspektiven aus beleuchtet und ein Überblick über bisherige Arbeiten und Methoden in der Computergraphik gegeben wurde, wird nun ein neuer Ansatz vorgestellt, mit welchem sich Brüche auf starren Oberflächen plastischer Gemenge in mehreren Schritten berechnen und animieren lassen.

Ausgehend von der Modellbildung werden danach die einzelnen Aspekte vertieft, welche zur Umsetzung des Ansatzes notwendig sind. Phänomene, die mit diesem Ansatz erfasst werden können sind zum Beispiel Lava, nasser Sand mit trockener, verkrusteter Oberfläche, oder die Gletscherspalten in Gletschern. Obwohl Schnee und Eis zum Beispiel in der Computergraphik schon untersucht wurde gibt es keine Arbeit die sich mit dem Phänomen der Gletscher und der Gletscherspalten beschäftigt und auch für Lavaoberflächen oder Sandkrusten gibt es bisher keine vergleichbaren Ansätze.

3.1 Modellbildung

Bei hohem Druck oder hohen Temperaturen verformen sich Festkörper nicht spröde, sondern plastisch. Dies ist auch bei Gletschern zu beobachten bei denen in entsprechender Tiefe ein so hoher Druck herrscht, das dort der Gletscher fließt. Die Bewegung des Gletschers gleicht in diesen Regionen dem plastischen Fluss. An der Oberfläche ist das Eis hingegen weiterhin fest. Unebenheiten im Boden und an den Rändern des Gletschers können in den plastischen Regionen ausgeglichen werden. An der festen Eisschicht über dem plastischen Bereich hingegen entstehen Spannungen welche zu Sprödbrüchen führen, den Gletscherspalten. In Abbildung 3.1 ist dieser Sachverhalt veranschaulicht.

Das Modell nimmt an, dass die Tiefe der spröden Eisschicht an der Oberfläche des Gletschers im Vergleich zur Tiefe des gesamten Gletschers vernachlässigt werden kann. Dem gesamten Gletscher wird also ein plastischer Fluss unterstellt und nur an einer sehr dünnen Schicht auf der Oberfläche werden Sprödbrüche aufgrund von Spannungen gebildet.

Für die Simulation des plastischen Flusses des Gletschers kommt der in Abschnitt (2.4.2.1) vorgestellte Fluidsimulator zum Einsatz. Dieser muss jedoch für das Fließverhalten eines Gletschers erweitert werden, was im nächsten Kapitel 3.2 behandelt wird. Auf der konstruierten Oberfläche wird ein von [IO06]

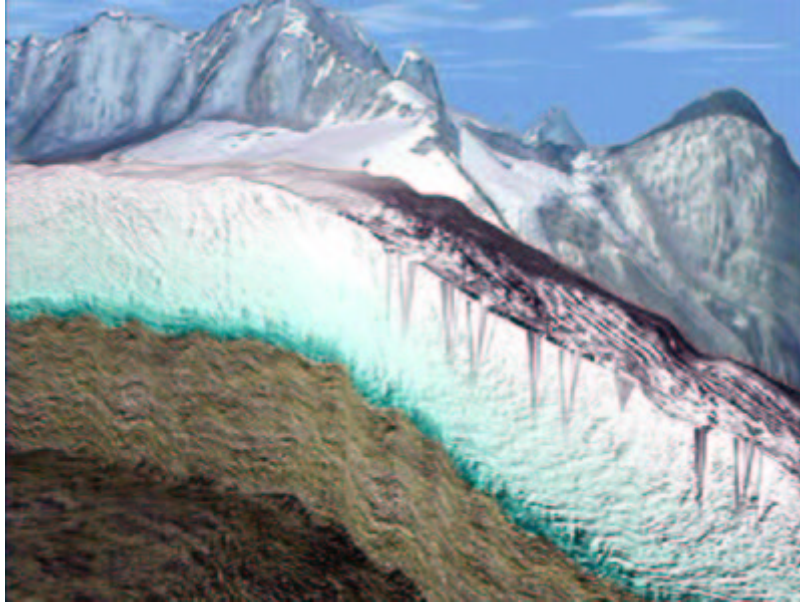


Abbildung 3.1: schematische Ansicht eines Gletschers (Grafik aus der Sendung “Quarks & Co“ vom 03.Januar 2006 des WDR)

vorgestellter Algorithmus zur Berechnung von Brüchen auf Oberflächen angewandt, was in Kapitel 3.3 genau beschrieben wird. Die darauf folgenden Kapitel gehen dann näher auf verschiedene Probleme ein die sich mit der Animation und Darstellung beschäftigen.

3.2 Elasto-plastischer Fluss

Bevor die Bruchbildung an der Gletscheroberfläche berechnet werden kann, müssen die charakteristischen Fließeigenschaften einer solchen Masse möglichst gut angenähert werden. Ein naheliegender Ansatz ist, den Gletscher als besonders zäh fließende Flüssigkeit aufzufassen und diese mit einer entsprechend hohen Viskositätskonstante im Fluidsimulator zu simulieren. Dieser Versuch ist von verschiedenen Seiten her problematisch. Sollte der Viskositätsterm

$$\mu \nabla^2 \vec{v} \quad (3.1)$$

mit einem expliziten Verfahren gelöst werden, kommt es schon bei geringfügig hohen Werten schnell zur Instabilität in der Simulation. Dies kann durch den Einsatz eines impliziten Verfahrens wie in [CMRBVHT02] vorgestellt behoben werden. Der Nachteil eines solchen impliziten Verfahrens ist die längere Simulationsdauer. Denn bei so hohen Viskositätskonstanten wie sie zur plausiblen Animation von Gletschern nötig wären, würde der Simulator mehrere tausend Zeitschritte rechnen, ohne das eine merkbare visuelle Veränderung in der Simulation sichtbar wäre. Dies ist unpraktikabel wenn zur Berechnung eines

Zeitschritten mehrere Minuten nötig sind. Ein weiteres Problem was aufkommt, wenn der Gletscher als enorm hoch viskose Flüssigkeit simuliert wird, ist das die Deformationsgeschwindigkeit verschwindet, also die Tensorkomponenten gegen null gehen. In Kapitel 2.5.2.2 wurde die Verformungsgeschwindigkeit zur Berechnung der Spannung herangezogen. Mit einer enorm hohen Viskosität sind die Spannungen welche mit Hilfe der Verformungsgeschwindigkeit abgeschätzt werden nahezu null. Das macht den Einsatz eines Bruchkriteriums schwierig. Darüber hinaus ist letztendlich auch wichtig, dass Gletscher elastisch sind und der Fluss nicht nur einen plastischen Anteil sondern auch einen elastischen Anteil hat. Neben all diesen Aspekten führt insbesondere das Bedürfnis Spannungen innerhalb des Fluids zu speichern und zu verfolgen zu einer notwendigen Erweiterung des Fluidsimulators aus Kapitel 2.4.2.1. Spannungen im Kontinuum werden durch den Tensor σ dargestellt und aus dem Verformungstensor ϵ mit Hilfe des Hookschen Gesetzes berechnet:

$$\sigma = \hat{C}\epsilon \quad (3.2)$$

\hat{C} ist der Elastizitätstensor vierter Stufe mit 81 Komponenten. Dieser kann unter Berücksichtigung der Tatsache das σ und ϵ symmetrisch sind und unter der Annahme das das Material isotrop ist bis zu einer Konstante, dem Elastizitätsmodul, reduziert werden. Zur Berechnung der Spannungen an einer Stelle im Fluid sind also eine Materialkonstante und der Verformungstensor ϵ an der entsprechenden Stelle notwendig. Die Verformung berechnet sich durch folgendes Integral:

$$\epsilon = \int \dot{\epsilon} dt \quad (3.3)$$

Dabei ist $\dot{\epsilon}$ die Verformungsgeschwindigkeit D aus Gleichung (2.31). In Kapitel 2.5.2.2 wird die gesamte Verformung also durch $\epsilon = \dot{\epsilon}$ ermittelt. Für die Verfolgung von Spannungen über die Zeit hinweg brauchen wir eine Historie der Verformung des Fluids vom Simulationsstart an. Wie in [GBO04] wird dafür in den Fluidzellen ein Verformungstensor gespeichert. Da der Verformungstensor symmetrisch ist reichen 6 skalare Werte aus. Diese werden analog zu den Geschwindigkeiten an verschiedenen Stellen in der Zelle gespeichert. Die Werte auf der Hauptdiagonale im Zellmittelpunkt und die Werte auf den Nebendiagonalen entsprechend auf einer der vier Kanten der Zelle, welche normal zur Krafrichtung und Ebenennormale stehen.

Der gespeicherte Tensor wird nur die elastische Verformung speichern, also den Anteil der Verformung welcher für die Spannungen im Material sorgt. Da dieser Elastizitätstensor nicht dem eigentlichen Verformungstensor entspricht wird der Tensor mit \mathbf{E} bezeichnet. Im Initialzustand sind alle Elemente des Tensors 0. Die Änderung des Tensors über die Zeit ergibt sich aus:

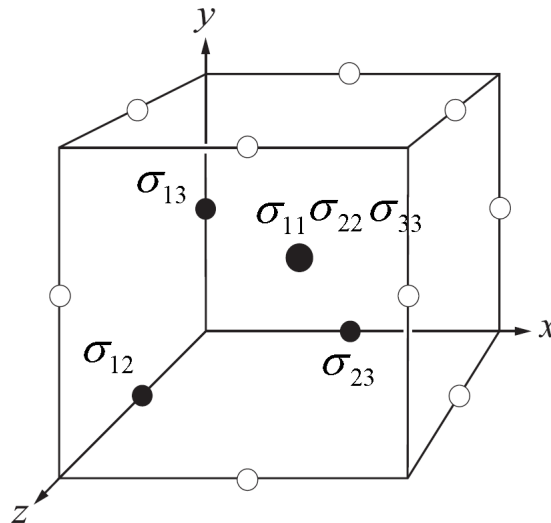


Abbildung 3.2: Elemente des Verformungstensors ϵ werden an verschiedenen Stellen in der Fluidzelle gespeichert

$$\frac{\partial \mathbf{E}}{\partial t} = D - k_{yr} \max(0, \|\mathbf{E}\| - k_{yp}) \frac{\mathbf{E}}{\|\mathbf{E}\|} - (\nabla \mathbf{E}) \cdot \mathbf{u} \quad (3.4)$$

Bei der Lösung des sich ergebenden Integrals kommt wieder das Operator-splitting zum Einsatz was bedeutet, dass das Integral in seine einzelnen Terme zerlegt wird und diese einzeln gelöst und dann zusammengerechnet werden.

Der erste Term D entspricht der Deformationsgeschwindigkeit welche durch einen einfachen Eulerschritt $\Delta t D$ integriert wird. Das entspricht der Akkumulation der gesamten Verformung des aktuellen Zeitschrittes auf den Tensor.

Der zweite Term modelliert den plastischen Fluss des Materials. Plastische Verformung ist im Gegensatz zur elastischen Verformung irreversibel und sorgt auch nicht für Spannungen im Material. In Worten ausgedrückt zieht dieser Term den Anteil der Verformung des aktuellen Zeitschrittes ab, welcher plastisch ist. Der Elastizitätstensor “vergisst“ die Verformung welche irreversibel ist und daher für die Elastizität keine Rolle mehr spielt. Der Term zieht den normierten Elastizitätstensor ab welcher mit zwei Faktoren gewichtet wird. Zuerst mit der Materialkonstante k_{yr} welches die plastische Fließgeschwindigkeit des Materials angibt. Also wie schnell sich das Material nach Überschreitung der Fließgrenze plastisch verformt. Der zweite Faktor ist der Anteil, den die Frobenius-Norm des Elastizitätstensors die Fließgrenze k_{yp} überschreitet. Je mehr die Grenze überschritten wurde, desto stärker kommt es zum plastischen Fluss. Für eine genauere Herleitung des Terms sei auf [GBO04] verwiesen. Integriert wird auch hier mit einem expliziten Eulerschritt.

Der letzte Term, $-(\nabla \mathbf{E}) \cdot \mathbf{u}$, beschreibt die Advektion des Tensors durch das Geschwindigkeitsfeld des Fluids. Wie auch bei allen anderen advektierten Fluidkomponenten wird die Semi-lagrange Advektion aus [Sta99] verwendet. Dabei sind vier Rückverfolgungen für die an vier verschiedenen Stellen in der Zelle gespeicherten Tensorkomponenten notwendig.

Die Integration von Gleichung (3.4) wird verwendet um den Tensor der elastischen Verformung im Laufe der Simulation zu verfolgen. Damit die elastische Verformung auch tatsächlich Effekt hat, muss das Geschwindigkeitsfeld durch eine entsprechende Elastizitätskraft verändert werden. Diese Elastizitätskraft ist identisch mit den Spannungen σ im Material und wird durch das Hooksche Gesetz berechnet:

$$\sigma = k_{em} \mathbf{E} \quad (3.5)$$

Diese Spannungen müssen nun noch auf das Geschwindigkeitsfeld angewendet werden. Dies geschieht analog zur inneren Reibung in Gleichung (2.38) durch Anwendung eines Tensorgradienten:

$$\mathbf{u} = \mathbf{u} + \Delta t \nabla \cdot \sigma \quad (3.6)$$

Die in Abbildung 3.3 noch einmal schematisch zusammengefassten Änderungen am Fluidsimulator helfen nun die aufgrund von Elastizität auftretenden Spannungen im Fluid zu verfolgen und auszuwerten.

Wie auch in Kapitel 2.5.2.4 kann es bei der Anwendung der Elastizitätskraft in Gleichung (3.6) zu Problemen mit der numerischen Stabilität kommen. Weiterhin ist die Wahl der Materialkonstanten k_{yp} und k_{yr} für das Fließverhalten des Fluids wichtig. In [GBO04] sind die verschiedenen Möglichkeiten durch die unterschiedlichen Werte von k_{yp} und k_{yr} gegenübergestellt.

3.3 Oberflächenbruchbildung

Das Ergebnis der Fluidsimulation sind die Simulationsdaten welche in jedem Zeitschritt gespeichert wurden. Dazu gehören die Positionen aller Markerpartikel und die verschiedenen in den Zellen gespeicherten Fluidkomponenten wie Geschwindigkeit, Druck und Spannungstensor. Aus den Markerpartikeln wird nun in jedem Zeitschritt ein Dreiecksnetz, wie in Kapitel 2.4.2.1 beschrieben, gebildet. Dieses Netz ist die Grundlage für den weiteren Verfahrensschritt zur Berechnung von Oberflächenbrüchen der in diesem Kapitel beschrieben wird. Das Ergebnis dieses Schrittes sollen physikalisch plausible Brüche auf der Oberfläche des Dreiecksnetzes sein.

Kerngedanke ist, eine modifizierte Abwandlung des in [IO06] vorgestellten Algorithmus zur Berechnung von Oberflächenbrüchen auf das vorliegende Dreiecksnetz anzuwenden. Zum besseren Verständnis wird

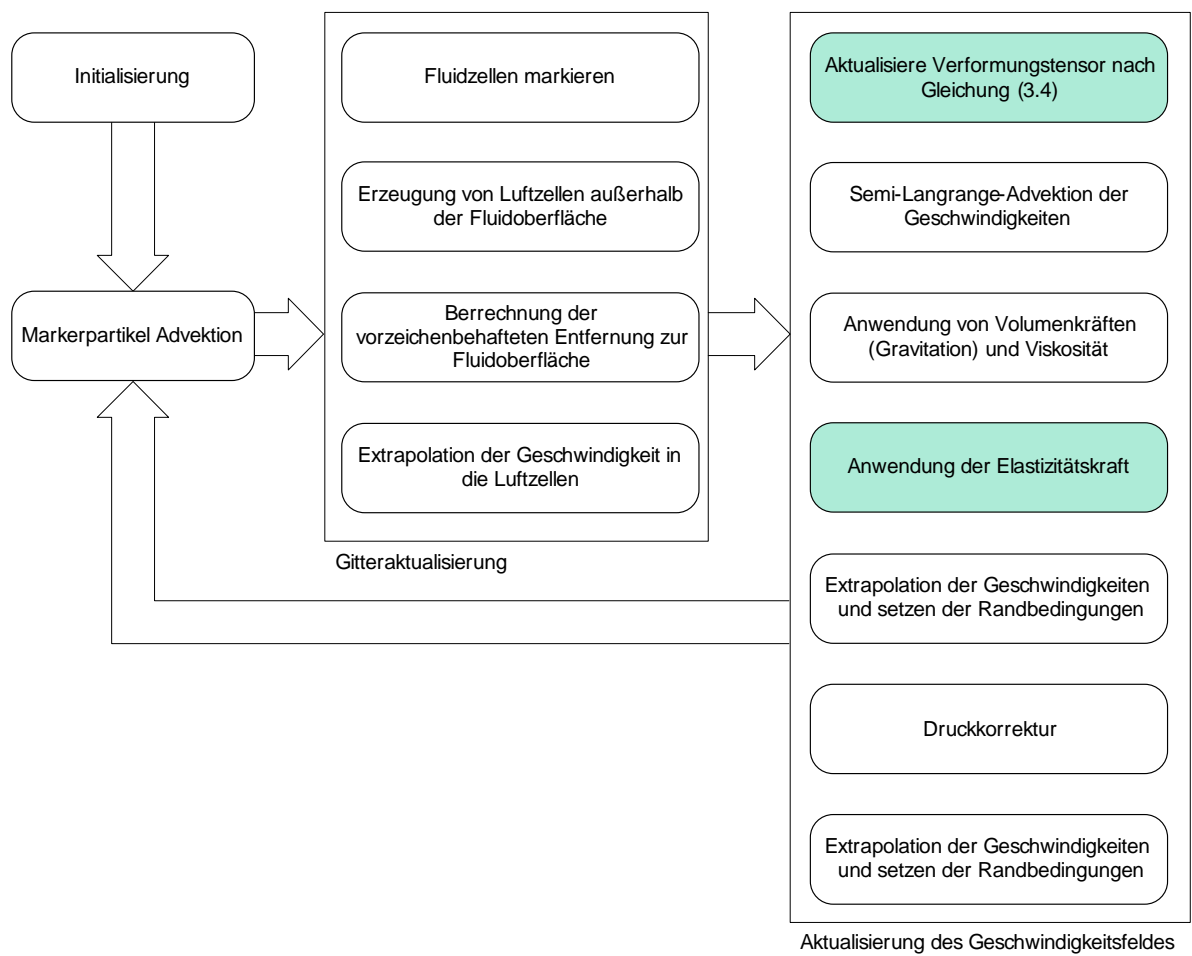


Abbildung 3.3: Die Erweiterung des Fluidsimulators für elasto-plastischen Fluss

im Folgenden der Algorithmus und seine Änderungen genauer beschrieben.

Der Algorithmus ist der in Kapitel 2.5.1 vorgestellten Methode sehr ähnlich. Die essenziellen Unterschiede bestehen darin, dass statt einer Diskretisierung des Volumens durch ein Tetraedernetz nur die Oberfläche durch ein Dreiecksnetz diskretisiert wird. Weiterhin werden die Spannungen in den diskreten Elementen (den Dreiecken) nicht durch lösen einer Erhaltungsgleichung wie in [OH99] bzw. Kapitel 2.4.1 ermittelt, sondern auf Basis von Heuristiken gesetzt. Beide Unterschiede vereinfachen und reduzieren die Anzahl an Berechnungen. Auch muss kein lineares Gleichungssystem ausgewertet werden. Konsequenz ist, dass die Brüche nur auf der Oberflächen ermittelt werden können.

Für den Algorithmus werden Oberflächenspannungen in den Dreiecken definiert. Diese Spannungen werden in den Dreiecken als konstant angenommen und mit einem Spannungstensor beschrieben. Dieser Tensor beschreibt den ebenen Spannungszustand des Dreiecks und ist daher ein 2×2 Tensor. Dabei ist der Tensor relativ zum lokalen Koordinatensystem des Dreiecks angegeben, welches durch eine Kante des Dreiecks und einen in der Ebene des Dreiecks orthogonal dazu liegenden Vektor gegeben ist.

Bevor der Algorithmus ausgeführt wird werden notwendige Werte in den Elementen berechnet. Dann werden die Spannungen heuristisch in den Elementen gesetzt (mehr dazu in Kapitel 3.3.6). Der Algorithmus arbeitet iterativ mit zwei sich abwechselnden Schritten. Zuerst wird mehrmals das Bruchkriteriums ausgewertet und der Vertexsplit durchgeführt. Danach die Relaxation der Oberflächenspannungen welche ebenfalls in einer bestimmten Anzahl ausgeführt wird.

3.3.1 Bruchkriterium

Die Matrix β transformiert einen Punkt innerhalb des lokalen Koordinatensystems eines Dreiecks in seine baryzentrische Koordinaten und ist definiert durch

$$\beta = \begin{bmatrix} m_{[1]} & m_{[2]} & m_{[3]} \\ 1 & 1 & 1 \end{bmatrix}^{-1}. \quad (3.7)$$

$m_{[i]}$ mit $i \in \{1, 2, 3\}$ sind die baryzentrischen Koordinaten der Eckpunkte des Dreiecks.

Die Spannungen σ welche für jedes Element definiert sind können zu an den Eckpunkten des Dreiecks wirkenden Kräften umformuliert werden. Sei $\mathbf{f}_{[i]}$ die Kraft welches ein Element auf einen anliegenden Knoten i ausübt, so kann diese mit

$$\mathbf{f}_{[i]} = -A \sum_{j=1}^3 p_{[j]} \sum_{k=1}^2 \sum_{l=1}^2 \beta_{jl} \beta_{ik} \sigma_{kl} \quad (3.8)$$

beschrieben werden (genauer dazu in Kapitel 2.5.1). A ist der Flächeninhalt des Dreiecks, $p_{[j]}$ ist die Position im globalen Koordinatensystem \mathbb{R}^3 . Die gesamte Kraft welche auf einen Knoten i wirkt kann als Summe aller Kräfte $\mathbf{f}_{[i]}$ der Elemente, welche i enthalten, erfasst werden. Da zur Berechnung von $\mathbf{f}_{[i]}$ der Tensor σ zugrunde liegt, welcher alle verschiedenen Spannungen (Zug- und Druckspannung) enthält, repräsentiert $\mathbf{f}_{[i]}$ ebenfalls die Zug- und Druckspannungen, welche von einem Element aus auf den Knoten i ausgeübt werden.

Das Bruchkriterium wird in den Knoten des Dreiecksnetzes ausgewertet. Dazu wird wie in Kapitel 2.5.1 ein Separationstensor ζ aus den umliegenden Elementen ermittelt. Dieser Tensor wird gebildet, indem die Kräfte $\mathbf{f}_{[i]}$, welche die Elemente auf die anliegenden Knoten ausüben, in Zug- und Druckkräfte $\mathbf{f}_{[i]}^+$ und $\mathbf{f}_{[i]}^-$ zerlegt werden. Dafür wird zuerst der Tensor σ in den Tensor σ^+ , welcher den Anteil an Zugkräften enthält, und den Tensor σ^- folgendermaßen zerteilt: Der 2×2 Spannungstensor σ wird in den Hauptspannungszustand mit Hilfe der Routine aus [VTPF07] transformiert (Eigenwertdekomposition). Aus den positiven Hauptspannungen $v^i(\sigma)$ wird dann der Tensor der Zugspannungen σ^+ , und aus den negativen Hauptspannungen der Tensor σ^- konstruiert.

$$\sigma^+ = \sum_{i=1}^2 \max(0, v^i(\sigma)) m(\hat{\mathbf{n}}^i(\sigma)) \quad (3.9)$$

$$\sigma^- = \sum_{i=1}^2 \min(0, v^i(\sigma)) m(\hat{\mathbf{n}}^i(\sigma)) \quad (3.10)$$

$v^i(\sigma)$ sind die Eigenwerte oder auch Hauptspannungen von σ und $\hat{\mathbf{n}}^i(\sigma)$ die dazu gehörigen Eigenvektoren. Durch Substitution von σ mit σ^+ bzw. σ^- in Gleichung (3.8), wird $\mathbf{f}_{[i]}^+$ und $\mathbf{f}_{[i]}^-$ entsprechend berechnet.

Der Separationstensor ζ ist ein 3×3 Tensor dessen positive Eigenwerte die im Knoten i wirkenden Zugspannungen repräsentieren und dessen negative Eigenwerte entsprechend die wirkenden Druckspannungen. Dabei ist ζ so konstruiert, dass jeder Zug- bzw. Druckspannung eine entsprechend gleich große Oppositionalspannung entgegen wirkt. Dabei kommt die Funktion $m(\vec{x})$ zum Einsatz welche als das dyadische Produkt des Argumentvektors mit sich selbst definiert ist und somit eine Matrix bzw. einen Tensor erzeugt der den Argumentvektor als Eigenvektor mit dessen Betrag als Eigenwert enthält. ζ wird aus den Tensoren der Druck- und Zugspannungen $m(\mathbf{f}^-)$ und $m(\mathbf{f}^+)$ wie in Gleichung (2.29) für jeden Knoten berechnet. Der größte positive Eigenwert v^+ und der dazugehörige Eigenvektor \mathbf{n}^+ von ζ wird bestimmt. v^+ wird mit einem Schwellwert, der Materialhärte τ , verglichen. Wird die Materialhärte überschritten, dann kommt es am Knoten zum Materialversagen. Die versagenden Knoten werden nach Größe von v^+ in eine Prioritätsliste einsortiert. Am Knoten mit dem größten v^+ wird der nächste Bruch mit dem im folgenden Kapitel beschriebenen Vertexsplit modelliert.

3.3.2 Vertexsplit

Nachdem nun der Knoten mit dem größten Materialversagen bestimmt wurde, wird ein Bruch in diesem Knoten gebildet. Die Bruchrichtung ist dabei orthogonal zum Eigenvektor \hat{n}^+ , welcher der Normalen der Bruchebene entspricht. Um den Bruch im Dreiecksnetz zu modellieren müssen Kanten vom Bruchknoten aus entlang der Bruchebene eingefügt werden. Dazu werden die Kanten aus den Nachbardreiecken des Bruchknotens mit der Bruchebene geschnitten, welche nicht den Bruchknoten enthalten. An den Schnittpunkten wird ein neuer Knoten eingefügt und die der Kante benachbarten Dreiecke entsprechend durch Kanteneinfügen geteilt. Um die Bildung von extrem spitzwinkligen oder degenerierten Dreiecken zu unterbinden werden die Kanten nicht gebildet, wenn der Winkel zwischen der neuen Bruchkante und einer der beiden Seitenkanten des Nachbardreiecks einen bestimmten Toleranzwert unterschreitet. Stattdessen wird dann die entsprechende Seitenkante als Bruchkante angenommen. Der Bruchknoten wird durch zwei neue Knoten ersetzt welche die selbe Position haben wie der Bruchknoten. Je einem dieser zwei neuen Knoten wird eine der beiden Seiten der Bruchebene zugeordnet. Die Referenz auf den Bruchknoten wird in dessen Nachbardreiecken durch die Referenz auf jenen der beiden neuen Knoten ersetzt, welche der Seite der Bruchebene zugeordnet wurde auf welcher sich das entsprechende Nachbardreieck befindet. Die einzelnen Bruchkanten werden dabei durch zwei neue Bruchkanten für je eine Seite der Bruchebene ersetzt. Das führt dazu, dass diese neuen Bruchkanten Randkanten sind, welche nur noch ein Dreieck als Nachbarn haben.

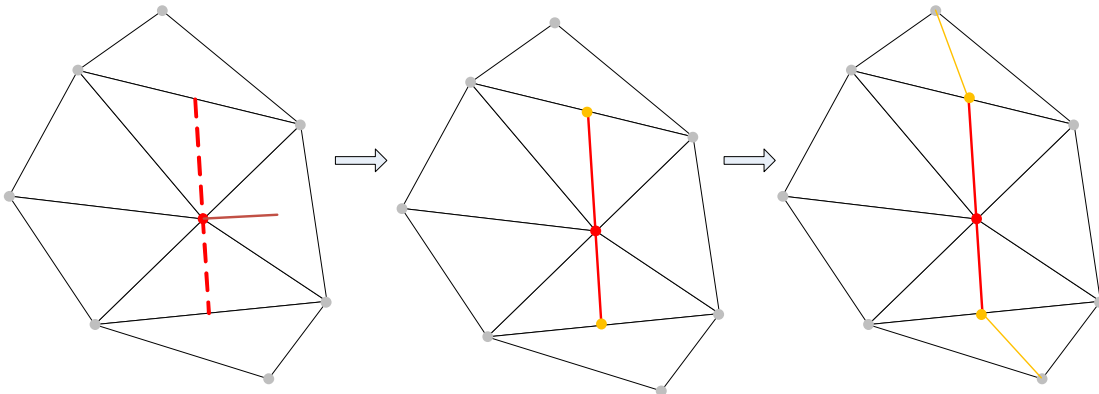


Abbildung 3.4: Der Vertexsplit am Bruchknoten erzeugt Randkanten entlang der Bruchebene

Da die Spannungen als konstant über der Fläche der Dreiecke definiert sind, können bei dem Aufteilen eines Dreiecks dessen Eigenschaften einfach in die zwei neuen Dreiecke übernommen werden. Die Änderung der Topologie allerdings macht das zweidimensionale Koordinatensystem des ursprünglichen Dreiecks ungültig. Für die neuen Dreiecke werden entsprechend neue lokale Koordinatensysteme bestimmt. Alle lokal definierten Größen, wie zum Beispiel der Spannungstensor müssen deshalb aktuali-

sirt werden. Da sich die Ebene nicht ändert, kann der Übergang vom lokalen Koordinatensystem des ursprünglichen Dreiecks in je eines der neuen Dreiecke durch eine Rotation R beschrieben werden. Der Spannungstensor σ' des neuen Dreiecks beispielsweise ergibt sich dann aus

$$\sigma' = R^T \sigma R \quad (3.11)$$

Die Theorie der Bruchmechanik besagt, dass sich Brüche nicht nur bilden sondern auch weiter ausbreiten. Dies kommt daher, dass bei Dehnungsbrüchen die größten Zugspannungen innerhalb eines kleinen Radius um die Bruchspitze auftreten. Diese Belastung sorgt dafür, dass es an der Bruchspitze zu weiterem Materialversagen und damit weiterer Bruchbildung kommt. Um diesen Effekt zu berücksichtigen werden die Separationstensoren der Knoten an den Bruchspitzen folgendermaßen angepasst:

$$\zeta' = \zeta + \alpha v^* \mathbf{m}(\hat{\mathbf{n}}^+) \quad (3.12)$$

Dabei ist ζ' der aktualisierte und ζ der alte Separationstensor der Knoten an den Bruchspitzen. $\mathbf{m}(\hat{\mathbf{n}}^+)$ ist ein Tensor welcher die Normale der Bruchebene $\hat{\mathbf{n}}^+$ als Eigenvektor hat. Dieser Tensor wird mit dem Koeffizienten v^* skaliert der dem Überschuss des für das Materialversagen verantwortlichen Eigenwertes über der Materialhärte entspricht. In einer Formel ausgedrückt:

$$v^* = v^+ - \tau \quad (3.13)$$

Der Koeffizient α in Formel (3.12) ist ein Parameter zur Steuerung des Effektes der Bruchausbreitung und ist auf das Intervall $[0, 1]$ beschränkt. Ein Wert von null bewirkt, dass keinerlei Bruchausbreitung berücksichtigt wird. Kleine Werte für α erzeugen kurze, raue und verzackte Brüche, während höhere Werte zu langen und mehr geradlinigen Brüchen führen.

3.3.3 Relaxation

Nachdem nun das Bruchkriterium ausgewertet und der Bruch behandelt wurde, werden die Spannungen relaxiert. Dies geschieht in [IO06] ähnlich einer Netz-diffusion. Die Relaxation bewirkt, dass Spannungen von Bereichen mit großer Spannung in Bereiche mit weniger Spannung umverteilt werden. Dafür wird zuerst die Verschiebung der Knoten aus den Spannungen der umliegenden Dreiecke berechnet. Sei $\mathbf{F}_{[n]}$ die Summe aller Kräfte welche von benachbarten Elementen auf einen Knoten n ausgeübt werden. Die Kraft die ein einzelnes Element auf einen seiner Knoten ausübt hängt von der Spannung in diesem Element ab und wird mit (3.8) berechnet. Die Verschiebung $\Delta p_{[n]}$ des Knotens errechnet sich aus

$$\Delta p_{[n]} = \Delta t \mathbf{F}_{[n]} . \quad (3.14)$$

Wobei Δt die Relaxationsgeschwindigkeit und damit ein Parameter des Algorithmus ist.

Nachdem so alle Knotenverschiebungen berechnet wurden, werden diese genutzt um die Änderung der Spannung $\Delta \sigma$ in den einzelnen Elementen zu bestimmen. Dies geschieht indem die Verformung eines jeden Elementes, repräsentiert durch den Greenschen Verformungstensor ϵ , berechnet wird. Da die Verformung nach dem Hookeschen Gesetz proportional zur Spannung ist, muss auch die Änderung der Verformung Proportional zur Änderung der Spannung sein. Mit der Annahme einer Proportionalitätskonstante von 1, kann demnach bei Kenntnis der Änderung der Verformung bzgl. der Knotenpositionen $\frac{\delta \epsilon}{\delta p_{[n]}}$ auf die Änderung der Spannung mit

$$\Delta \sigma = \Delta \epsilon = \frac{\delta \epsilon}{\delta p_{[n]}} \Delta p_{[n]} \quad (3.15)$$

zurück geschlossen werden. Für eine genauere Herleitung von $\Delta \epsilon$ sei auf [IO06] verwiesen.

In jedem Durchgang werden also die Kräfte aus den Spannungen der Elemente berechnet die auf die Knoten wirken. Aus diesen werden dann die Knotenverschiebungen berechnet. Mit Gleichung (3.15) werden dann über die Ableitung der Verformung nach den Knotenverschiebungen die Änderung der Spannungen in den Elementen berechnet. Nach deren Aktualisierung werden die Kräfte neu berechnet, die auf die Knoten wirken. Dieser Relaxationszyklus wird eine festgelegte Anzahl oft hintereinander ausgeführt. Die Anzahl der Relaxationsschritte zusammen mit der Relaxationsgeschwindigkeit Δt sind Parameter des Algorithmus und beeinflussen die Ergebnisse. Die charakteristischen Bruchmuster entstehen durch das Zusammenspiel vom Einfügen der Randkanten entlang der Brüche und der anschließenden Relaxation.

3.3.4 Stabilität

Sehr kleine Dreiecke, degenerierte Dreiecke oder so genannte Splitterdreiecke (slivers) können zu numerischen Instabilitäten führen. Dies bewirkt meistens das die Spannungstensoren der Dreiecke extrem große Werte bekommen. Ein Fehler welcher durch die Relaxation auf andere Elemente übertragen wird. Die Folge ist, dass jeder Knoten bricht und das Ergebnis unplausibel wird.

Durch den Toleranzwinkel beim Einfügen der Bruchkanten in Abschnitt 3.3.2 wird versucht die Bildung solcher missgebildeten Dreiecke zu verhindern. Doch gerade bei sehr dichten und hochaufgelösten Ausgangsnetzen gibt es topologische Situationen die trotzdem leicht zu numerischen Problemen führen.

[IO06] führen diese Instabilitäten auf sehr große Singulärwerte in der Matrix der Basisvektoren β der einzelnen Elemente zurück. Um die Instabilitäten zu verhindern wird vor Beginn des Algorithmus ein Singulärschwellwert κ_{avg} aus

$$\kappa_{avg} = c \left(\frac{\sum_{i=1}^N \kappa_i}{N} \right) \quad (3.16)$$

berechnet. Wobei N die Anzahl der Elemente ist und κ_i der größte Singulärwert des i -ten Elementes ist. κ_{avg} ist also der Mittelwert aller größten Singulärwerte aller Elemente. c ist ein konstanter Multiplikator der verhindern soll, dass der berechnete Schwellwert nicht zu klein bleibt. [IO06] nutzen einen Wert von $c = 8$.

Wenn immer die Matrix β eines Elementes berechnet wird, wird die Singulärwertzerlegung mit Hilfe der entsprechenden Routine aus [VTPF07] durchgeführt:

$$\beta = U \Sigma V^* \quad (3.17)$$

Die Matrix Σ ist eine Diagonalmatrix und enthält die Singulärwerte von β auf der Hauptdiagonalen. Diese Werte werden ausgewertet und jeweils auf den Wert κ_{avg} beschränkt falls dieser überschritten wird. Ist es auf diesem Weg zu einer Änderung von Σ gekommen, wird die Matrix β nach Gleichung (3.17) mit den beschränkten Singulärwerten neu berechnet. Für eine weiterführende Diskussion der Singulärwertzerlegung sei [HB06] Seite 113 ff. empfohlen.

Mit dieser Berücksichtigung der Singulärwerte kann der Algorithmus numerisch stabil gehalten werden ohne die erzeugten Bruchmuster stark in ihrer visuellen Erscheinung zu beeinflussen.

3.3.5 Algorithmus

Mit der Erweiterung zur Sicherstellung der numerischen Stabilität ist der Algorithmus vollständig beschrieben. Hier sei der Algorithmus noch einmal kurz zusammengefasst.

In der Initialisierungsphase werden die Eigenschaften der Elemente wie zum Beispiel β berechnet und die Spannungen σ in den Elementen nach einer Heuristik gesetzt. Zusätzlich wird κ_{avg} zur Beschränkung der Singulärwerte der neu entstehenden Elemente berechnet.

Der Algorithmus kann beliebig oft ausgeführt werden, wobei kein Bruch mehr stattfindet wenn der Algorithmus oft genug durchlaufen wurde. Ein einzelner Schritt besteht aus mehrfacher Auswertung des Bruchkriteriums und dem Vertexsplit des Knotens mit dem größten Eigenwert des Separationstensors, der die Materialhärte τ überschreitet. Danach wird in einer bestimmten Anzahl der Relaxationsschritt

durchgeführt. Die den Algorithmus parametrisierenden Variablen sind in Tabelle (3.1) zusammengefasst. Der Algorithmus ist noch einmal in der schematischen Darstellung 3.5 dargestellt.

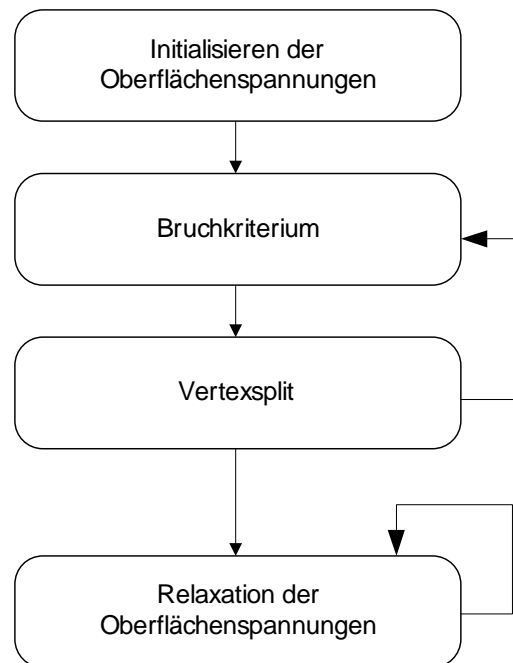


Abbildung 3.5: Der Algorithmus zur Berechnung von Oberflächenbrüchen

Tabelle 3.1: Übersicht der Algorithmusparameter

Parameter	Beschreibung
σ	Oberflächenspannungen der Elemente
τ	Materialhärte für das Bruchkriterium
N_T	Gesamtanzahl an Iterationen
η	Anzahl Brüche(Vertexsplits) pro Iteration
R	Anzahl Relaxationsschritte pro Iteration
Δt	Relaxationsgeschwindigkeit
α	Steuert Bruchausbreitung beim Vertexsplit

3.3.6 Oberflächenspannungen

Von den in Tabelle (3.1) aufgeführten Parametern des Algorithmus sind die Spannungen σ , welche für jedes Element angegeben werden, die wichtigsten. Diese Spannungen bestimmen maßgeblich wo und

in welcher Richtung sich Brüche ausbilden. [IO06] stellen einige Heuristiken vor, mit denen sie die Spannungen zu Beginn gesetzt und während der Iterationen aktualisiert haben.

Für zufällige und völlig irreguläre Bruchmuster wird vorgeschlagen, den Spannungstensor nach folgender Gleichung zu initialisieren:

$$\sigma = c\mathbf{I} \quad (3.18)$$

\mathbf{I} ist dabei die Identitätsmatrix und c eine Konstante welche für jedes Element verschieden oder gleich groß sein kann. In der linken Abbildung 3.6 sind die so entstehenden Bruchmuster gezeigt. Obwohl jedes Element mit dem selben Spannungstensor initialisiert wurde, entstehen völlig irreguläre Strukturen. Dies kommt daher das der Spannungstensor relativ zum lokalen Koordinatensystem des Elementes angegeben ist. Die irreguläre Beschaffenheit des Dreiecksnetzes sorgt für eine quasi-zufällige Ausrichtung der Spannungen. Dieser Aspekt macht deutlich, dass die Ergebnisse des Algorithmus auch von dem Dreiecksnetz, also der Form wie die Oberfläche diskretisiert ist, abhängig sind.

Für gerichtete Brüche, wie zum Beispiel die Längsbrüche in alten Holzdielen, wird in [IO06] eine weitere Heuristik vorgeschlagen. Dabei beschreibt der Vektor \vec{d} die globale Richtung der Belastung. Da sich bildende Brüche orthogonal zur Richtung der größten Zugspannung ausbilden, werden gerichtete Brüche erzeugt, indem die Eigenvektoren von σ entsprechend nach \vec{d} ausgerichtet werden. Um die Spannungen in den einzelnen Elementen zu setzen muss der globale Richtungsvektor in das lokale Koordinatensystem des Elementes zu \vec{d}_{loc} transformiert werden. Eine Matrix mit dem Eigenvektor \vec{d}_{loc} wird durch das dyadische Produkt des Vektors mit sich selbst erzeugt. Gerichtete Brüche können somit durch folgende Gleichung für die Oberflächenspannungen der Elemente erzeugt werden:

$$\sigma = \left(\vec{d}_{loc}\right)^T \circ d_{loc} \quad (3.19)$$

Die Länge von \vec{d}_{loc} steuert die Stärke der Spannung. Ergebnisse dieser Heuristik sind in Abbildung 3.6 rechts zu sehen. In [IO06] werden noch weitere Heuristiken vorgestellt. So kann zum Beispiel die Krümmung auf der Oberfläche in der Bildung von σ berücksichtigt werden. Auch wird gezeigt, wie sich Aufschlagmuster nachbilden lassen.

Nach der Konstruktion der Fluidoberfläche aus den Markerpartikeln wird diese Oberfläche dem Algorithmus zur Bildung von Oberflächenbrüchen eingegeben. Damit sich die für Gletscher typischen Muster ausbilden müssten die Spannungen von einem Gestalter manuell gesetzt werden. Dies wäre enorm aufwendig. Und selbst dann ist eine physikalische Plausibilität nicht gewährleistet. Spätestens bei der Animation des Gletschers ist eine Steuerung des Algorithmus von Hand nicht mehr machbar, da hier die

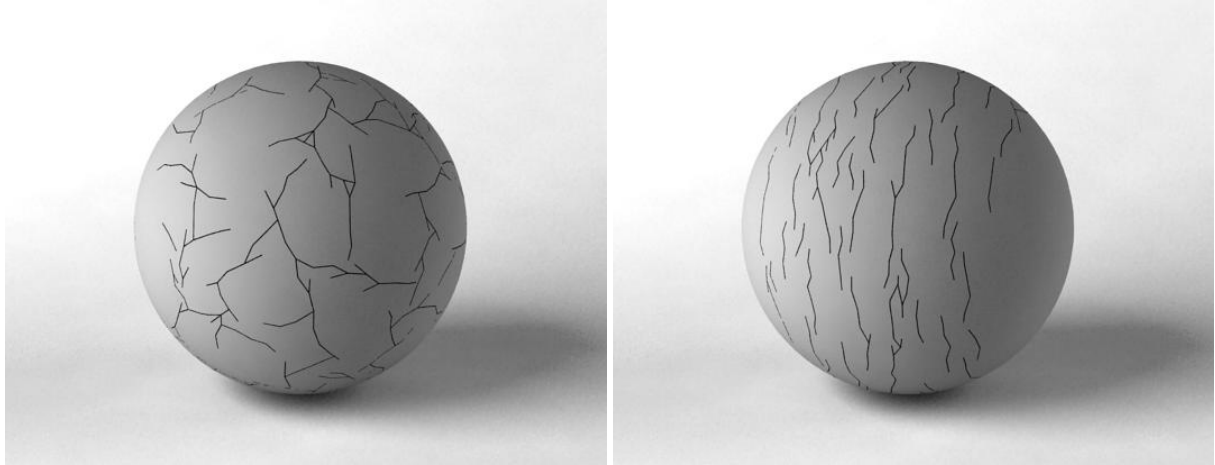


Abbildung 3.6: Ergebnisse auf Basis heuristischer Spannungen. Nach Gleichung (3.18) (*links*) und nach Gleichung (3.19) (*rechts*)

Änderungen der Spannungen ebenfalls eingegeben werden müssten. Das Problem besteht also darin die Oberflächenspannungen so zu setzen das sich physikalisch plausible Bruchmuster ergeben. Die Idee ist, die Spannungen aus der Fluidsimulation auf das Dreiecksnetz zu übertragen. In Kapitel 3.2 wurde erklärt wie der Fluidsimulator erweitert wurde, so dass Spannungen im Fluid verfolgt und angegeben werden können. Diese Spannungen sind in Form eines Verformungstensors \mathbf{E} in den einzelnen Fluidzellen gespeichert und damit für jeden Ort im Simulationsbereich definiert. Unter Kenntnis des Elastizitätsmoduls des Materials kann daraus mit Hilfe des Hookschen Gesetzes ein Spannungstensor berechnet werden welcher die Spannungen in der jeweiligen Zelle beschreibt. Der Verformungstensor wird ebenfalls zusammen mit den anderen Simulationsdaten in jedem Zeitschritt abgespeichert. Da die Spannungen aus der Fluidsimulation im globalen Koordinatensystem definiert sind, müssen diese analog zur Gleichung (3.19) in das lokale Koordinatensystem des Elementes transformiert werden:

$$\xi' = R^T \xi R \quad (3.20)$$

Dabei ist ξ der 3×3 Spannungstensor aus der Fluidsimulation. Dieser ist bezüglich des dreidimensionalen kartesischen Raums \mathbb{R}^3 mit den Basisvektoren \vec{e}_1 , \vec{e}_2 und \vec{e}_3 angegeben. Im Mittelpunkt des Elementes wird der Tensor durch trilineare Interpolation der Tensoren umliegender Zellen berechnet. Der 3×3 Tensor ξ' beschreibt den selben Spannungszustand wie ξ bezüglich des lokalen Koordinatensystems des Elementes welches durch die drei Vektoren \vec{u} , \vec{v} und \vec{n} beschrieben ist. \vec{n} ist die Normale des Dreiecks. Die Transformationsmatrix R berechnet sich aus dem Richtungskosinus:

$$R = \begin{pmatrix} \vec{e}_1 \cdot \vec{u} & \vec{e}_1 \cdot \vec{v} & \vec{e}_1 \cdot \vec{n} \\ \vec{e}_2 \cdot \vec{u} & \vec{e}_2 \cdot \vec{v} & \vec{e}_2 \cdot \vec{n} \\ \vec{e}_3 \cdot \vec{u} & \vec{e}_3 \cdot \vec{v} & \vec{e}_3 \cdot \vec{n} \end{pmatrix} \quad (3.21)$$

$\vec{x} \cdot \vec{y}$ ist hier das Skalarprodukt zwischen zwei Vektoren. Der lokale Spannungstensor σ des Elementes ergibt sich, indem die Spannungen in Richtung der Elementnormalen \vec{n} verschwinden:

$$\sigma = \begin{pmatrix} \xi'_{11} & \xi'_{12} \\ \xi'_{21} & \xi'_{22} \end{pmatrix} \quad (3.22)$$

In Abbildung 3.7 rechts ist das Ergebnis des Algorithmus mit Oberflächenspannungen aus der Fluidsimulation zu erkennen. Die Bruchstruktur ist sehr regulär und gerichtet. Die Brüche sind sehr klein, geradlinig und dicht beieinander.

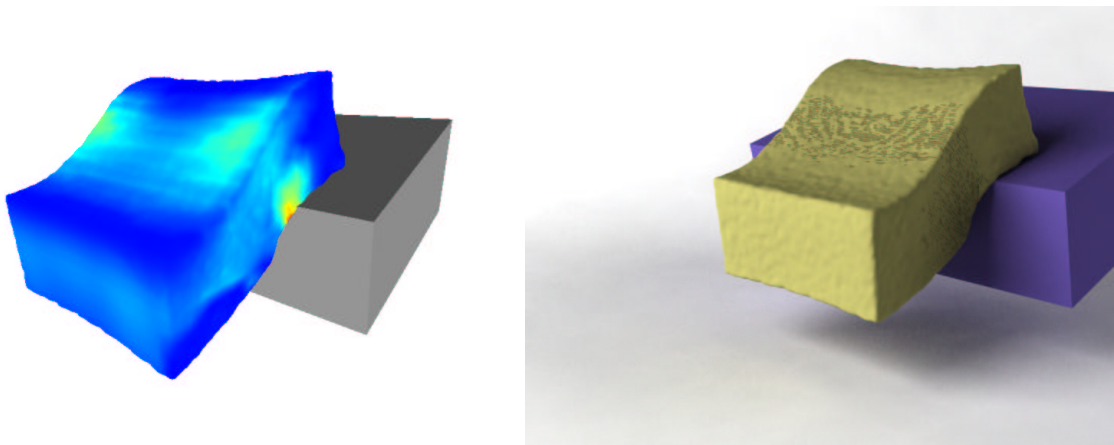


Abbildung 3.7: Oberflächenspannungen aus der Fluidsimulation (*links*) und das Ergebnis der Bruchbildung nach 150 Iterationen (*rechts*)

Dieses unbefriedigende Ergebnis resultiert aus der regulären Struktur der konstruierten Oberfläche. In Abbildung 3.8 ist eine Drahtgitteransicht der Oberfläche, wie sie das Ergebnis des Marching Cubes Algorithmus' aus [LC87] ist. Der generische Charakter des Konstruktionsalgorithmus spiegelt sich in den regulären gitterartigen Strukturen wieder.

Um die Ergebnisse der Bruchgenerierung zu verbessern muss daher die reguläre Struktur des Dreiecksnetzes in eine irreguläre Struktur umgewandelt werden, bevor der Algorithmus zur Berechnung der Oberflächenbrüche angewendet wird. Dieser Schritt wird im folgenden Kapitel umgesetzt und erläutert.

Ein weiteres Problem sind die hohen Rechenzeiten für den Algorithmus. In [IO06] werden für ein Drahtgitternetz aus etwa 30 000 Elementen Zeiten bis zu 1,5h erzielt. Die Implementierung dieser Arbeit kommt bei dem in Abbildung 3.8 angegebenen Netz aus ungefähr 106 000 Dreiecken auf eine Berech-

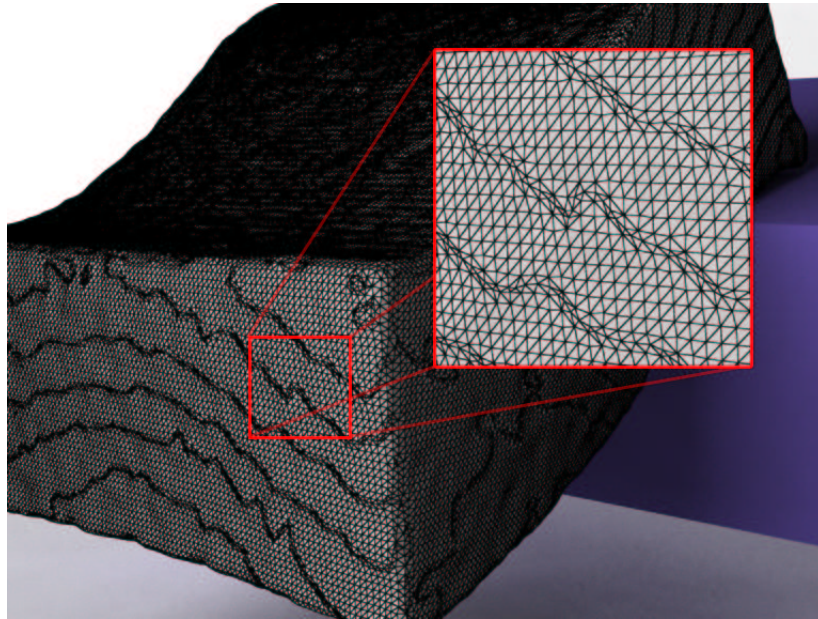


Abbildung 3.8: Drahtgitteransicht des vom Marching Cubes Algorithmus erzeugten Dreiecksnetzes

nungszeit von ca. acht Minuten. Auch dies ist zuviel wenn der Zeitrahmen von fünf Minuten pro Bild für Simulation, IO, Oberflächenkonstruktion und Bruchflächengenerierung nicht überschritten werden soll. Neben der Optimierung des Verfahrens ist eine Vereinfachung des Netzes zur Beschleunigung daher zusätzlich wünschenswert.

3.4 Re-tiling

Ein das Dreiecksnetz vorbereitender Algorithmus, der die im letzten Kapitel verfassten Anforderungen erfüllt, ist der in [Tur92a] vorgestellte. Damit kann ein beliebiges reguläres Netz in ein irreguläres Netz umgewandelt werden, mit der Möglichkeit, gleichzeitig eine Vereinfachung des Netzes zu bewirken. Der Algorithmus arbeitet in drei Schritten. Zuerst wird die Anzahl N an Eckpunkten bestimmt, die das resultierende Netz haben soll. Auf der Oberfläche werden diese Anzahl an Punkten zufällig platziert. In einem weiteren Schritt werden die Punkte so auf dem Netz entlang bewegt, dass sie möglichst gleichmäßig auf dem Netz verteilt sind. Im nächste Schritt werden die neuen Punkte in die Dreiecke hinein trianguliert, auf denen sie sich befinden. Danach werden alle anderen Eckpunkte des Netzes, die vorher schon zum Netz gehörten, so entfernt, dass nur noch die neuen Eckpunkte übrig bleiben.

Zufälliges Verteilen der Eckpunkte

Zuerst werden N Eckpunktkandidaten auf dem Netz verteilt. Dies geschieht in Abhängigkeit des Oberflächeninhalts der Dreiecke, so dass die Wahrscheinlichkeit das ein Eckpunkt auf einem großen Dreieck

erzeugt wird auch tatsächlich dem Anteil des Dreiecks an der Gesamtoberfläche entspricht.

Eine Referenz auf das Dreieck auf dem der Eckpunktkandidat erzeugt wurde, wird mit selbigem gespeichert. Dies ist notwendig da in späteren Schritten bekannt sein muss, auf welchem Dreieck die Eckpunktkandidaten sich befinden.

Repulsionsschritt

Die Punkte, welche nun zufällig auf dem Netz verteilt wurden, bestimmen, wie gut die verschiedenen Bereiche des Objektes erhalten werden. Regionen mit weniger Eckpunktkandidaten werden stärker vereinfacht als Regionen wo sich die Eckpunktkandidaten häufen. Um das gesamte Objekt möglichst gut zu erfassen werden nach der Erzeugung die Punkte gleichmäßig auf dem Netz verteilt.

Für jeden Punkt p werden die nächsten Nachbarn gefunden. Die Suche der Nachbarn findet im globalen Koordinatensystem \mathbb{R}^3 statt. Der Suchradius r wird heuristisch festgelegt. In [Tur92b] wird $r = 2\sqrt{\frac{A}{N}}$ vorgeschlagen, wobei A dem gesamten Oberflächeninhalt des Dreiecksnetzes entspricht. Sind die Nachbarn q_i gefunden worden, muss der Abstand auf der Oberfläche zu jedem dieser Nachbarn berechnet werden. Idealerweise wäre dieser Abstand der kürzeste Pfad auf der Netzoberfläche zwischen beiden Punkten. Um diese aufwendige Berechnung zu umgehen, wird der euklidische Abstand verwendet. Damit dieser möglichst gut den tatsächlichen Abständen auf der Oberfläche entspricht, werden die Nachbarn in die Ebene des Dreiecks t_p , auf dem der Kandidat p liegt, transformiert.

Für die Nachbarn welche auf dem Dreieck t_p liegen, und damit auch in dessen Ebene, erübrigt sich eine Transformation. Nachbarn, welche auf einem zu t_p benachbarten Dreieck liegen, werden um die gemeinsame Kante der beiden Dreiecke in die Ebene von t_p rotiert. Alle anderen Nachbarn werden um eine der Kanten von t_p rotiert und abschließend in die Ebene von t_p projiziert. Wie die Kante, um welche der Nachbar rotiert wird, gefunden wird ist offen. In der diese Arbeit begleitenden Implementierung wird die Kante gewählt, dessen zusammen mit der Normale von t_p aufgespannten Ebene den kleinsten Abstandswert zum Nachbarn hat. In Abbildung 3.9 sind diese Transformationen veranschaulicht.

Der auf diesen Weg ermittelte Abstand zwischen p und seinen Nachbarn wird dazu verwendet um die von den Nachbarn auf p wirkenden Repulsionskräfte zu berechnen. Diese bewirken das sich näher liegende Punkte stärker voneinander abstoßen als weiter entfernt liegende. Die Kraft die ein Nachbar q auf p ausübt berechnet sich aus:

$$\mathbf{f}_{q \rightarrow p} = (r - |\vec{p} - \vec{q}|) \frac{\vec{p} - \vec{q}}{|\vec{p} - \vec{q}|} \quad (3.23)$$

Damit werden für jeden Eckpunktkandidaten die von allen Nachbarn ausgeübten Kräfte folgendermaßen

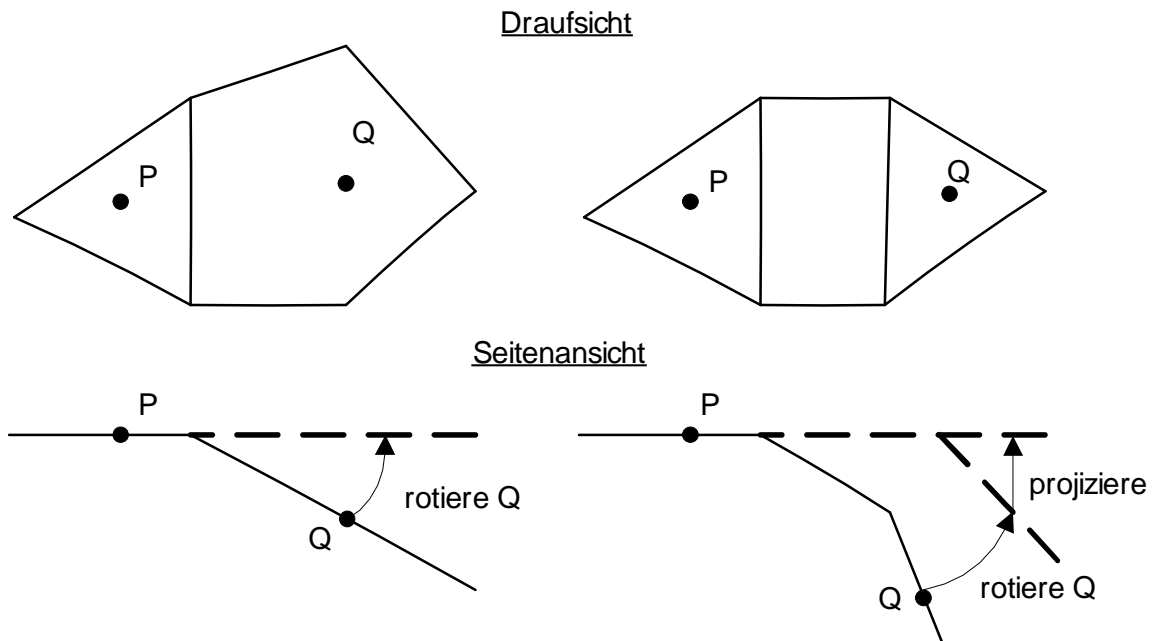


Abbildung 3.9: Abstand zweier Punkte auf einem Dreiecksnetz für benachbarte Dreiecke (*links*) und nicht benachbarte Dreiecke (*rechts*)

superponiert:

$$\mathbf{F}_p = \varpi \sum_i \mathbf{f}_{q_i \rightarrow p} \quad (3.24)$$

ϖ ist ein Dämpfungsfaktor zwischen $[0, 1]$. Da q die in die Ebene von t_p projizierten Nachbarn sind liegt auch \mathbf{F}_p in der Ebene von t_p . Sind so die Kräfte für alle p berechnet worden, werden die Eckpunktkandidaten anhand dieser Kräfte auf dem Netz bewegt.

Dabei wird überprüft ob der Vektor \mathbf{F}_p sich mit einer der Kanten von t_p schneidet. Falls nicht, wird p einfach durch $p = p + \mathbf{F}_p$ aktualisiert. Andernfalls wird der Punkt p aus dem Dreieck hinausbewegt und in die Ebene des über die Schnittkante benachbarten Dreiecks, wie in Abbildung 3.9 links, rotiert. \mathbf{F}_p verkürzt sich entsprechend und t_p wird ebenfalls aktualisiert. Dieser Schritt wird so lange wiederholt bis es nicht mehr zu einem Schnitt mit einer Kante kommt.

Dieser Repulsionschritt (Kräfte \mathbf{F}_p berechnen und p auf dem Netz bewegen) wird mehrmals ausgeführt, so dass es zu einer allmählichen Relaxation der Positionen der Eckpunktkandidaten auf dem Netz kommt.

Erhaltende Retriangulierung

Die gleichmäßig verteilten Eckpunktkandidaten werden nun in die Dreiecke auf denen sie nach der Repulsion liegen geblieben sind hinein trianguliert. Aus den Eckpunktkandidaten werden somit neue Eck-

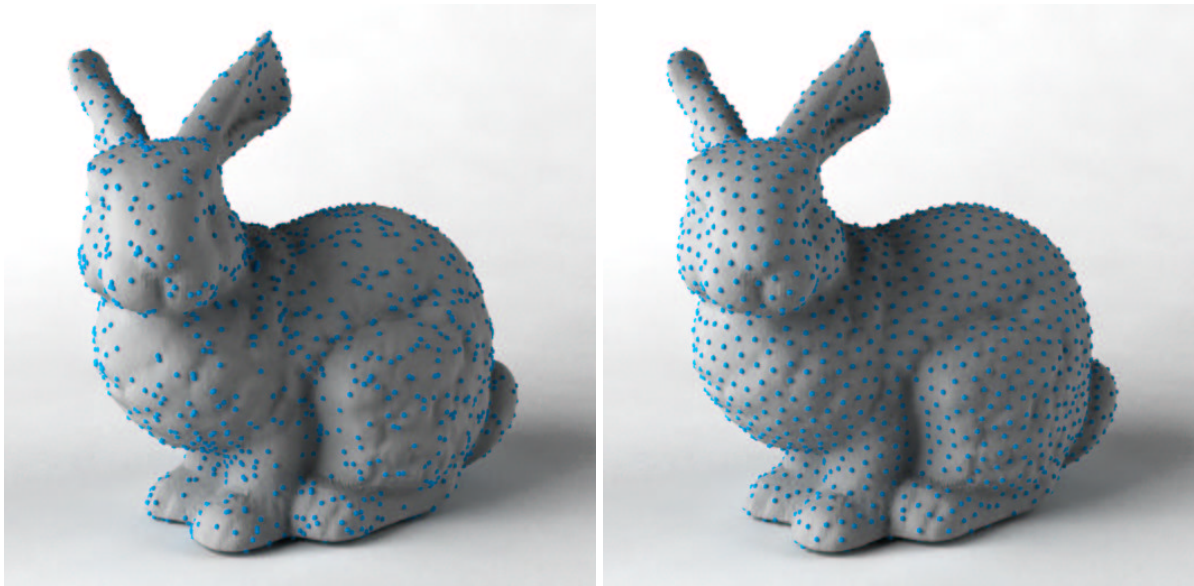


Abbildung 3.10: Das Ergebnis vor (*links*) und nach 20 Repulsionsschritten (*rechts*).

punkte des Netzes. Da diese neuen Eckpunkte alle in den Dreiecken des Originalnetzes liegen, wird das Objekt durch diese Retriangulierung nicht in seiner Form verändert. Daher wird hier von Form erhaltender Retriangulierung gesprochen.

Befindet sich mehr als ein Eckpunktkandidat auf einem Dreieck ist diese Retriangulierung nicht mehr trivial. Zur Lösung kann ein beliebiger Triangulierungsalgorithmus eingesetzt werden.

Eckpunkte entfernen

Im letzten Schritt werden die alten Eckpunkte, also jene welche schon vor der erhaltenden Retriangulierung Bestandteil des Netzes waren, entfernt. Dies geschieht schrittweise pro Eckpunkt. Jener wird zusammen mit allen Dreiecken welchen den zu entfernenden Eckpunkt enthalten aus dem Dreiecksnetz gelöscht. Das entstandene Loch im Netz wird danach durch eine Retriangulierung gefüllt.

Damit die Topologie des ursprünglichen Netzes erhalten bleibt ist die Art und Weise der Retriangulierung entscheidend. Die Randkanten des entstandenen Loches müssen dafür so in eine Ebene projiziert werden, dass ein einfaches Polygon entsteht. Dies stellt die Bedingung an die Projektion, dass sich die in eine Ebene transformierten Randkanten nicht schneiden dürfen (außer an den Eckpunkten benachbarter Randkanten). In den meisten Fällen genügt schon die durch die Normale des Eckpunktes definierte Ebene diesem Kriterium. Sollte die Projektion in diese Ebene einmal nicht gültig sein, werden noch zwölf weitere Ebenen getestet, deren Normalen sich aus gleichmäßig verteilten Punkten auf der Einheitskugel ergeben. Wenn keine Ebene gefunden wird, wird der zu entfernende Punkt vorläufig belassen.

Das durch den Zug der Randkanten in der Ebene liegende einfache Polygon wird dann trianguliert. Damit

im Resultat eine möglichst gleichmäßige Vernetzung herrscht und die neuen Eckpunkte mit den nächstliegenden Eckpunkten verbunden sind, ist eine Triangulierung nach dem Greedy-Algorithmus notwendig. Die Greedy Triangulation sortiert alle Eckpunktpaare des einfachen Polygons welche keine gemeinsame Kante haben nach Abständen. Beginnend vom geringsten Abstand wird getestet ob beim Einfügen einer Kante zwischen den beiden Paarpunkten andere Kanten geschnitten würden. Wenn kein Schnitt stattfindet und die Kante nicht den Außenbereich des Polygons kreuzt, wird sie eingefügt. Dieser Schritt wird für alle Paarpunkte ausgeführt.

Neben der Bedingung, dass es sich bei der Triangulierung um ein einfaches Polygon handeln muss, sind noch einige weitere topologische Sonderfälle zu beachten. Das Ausschließen dieser kritischen Fälle lässt sich in der Bedingung zusammenfassen, dass wenn zwei Eckpunkte des Randkantenzugs über eine Kante miteinander verbunden sind, welche nicht zum Randkantenzug gehört, diese Kante nicht das Innere des einfachen Polygons schneiden darf.

Stabilität

Wie für Algorithmen im Bereich "Computational Geometry" üblich, ist die Stabilität auch beim Algorithmus des Re-tilings ein großes Problem. Durch die hohe Auflösung des Marching Cube Netzes entstehen extrem spitze Splitterpolygone für welche die Präzision der Fließkommazahlen nicht mehr ausreicht. So kann es zum Beispiel vorkommen, dass das zur Bestimmung der Windung eines Dreiecks nötige Kreuzprodukt ein falsches Vorzeichen hat, was zu falschen Entscheidungen des Algorithmus, einem missgebildeten Ergebnisnetz und schlimmstenfalls zum Absturz des Programms führt.

Um den Algorithmus möglichst stabil und robust zu gestalten wurde daher an verschiedenen kritischen Stellen die CGAL-Bibliothek [BCD⁺99] eingesetzt. Diese Bibliothek ist direkt für die Umsetzung solcher Algorithmen entworfen worden. Kernkonzept ist der konsequente Einsatz des Paradigmas der generischen Programmierung, welches die Bibliothek sehr flexibel und auch leicht erweiterbar macht. Alle Funktionen und Strukturen sind durch C++-Templates von dem zugrunde liegenden Zahlentyp abstrahiert. So können neben der klassischen Fließkommazahl, durch austauschen des entsprechenden Template-Kernels andere Datentypen wie zum Beispiel den für beliebig präzise Fließkommazahlen aus [FHL⁺07] eingesetzt werden. Das zweite Kernkonzept sind die so genannten *predicates*. Das sind Aussagen über geometrische Sachverhalte, wie zum Beispiel die Windung eines Polygons, welche von der Bibliothek in jedem Fall absolut korrekt beantwortet werden. Im Falle der Windung beispielsweise, für dessen Bestimmung das Kreuzprodukt nötig ist, wird der Sachverhalt ausgenutzt, dass nur das Vorzeichen des Kreuzproduktes entscheidend ist. Daher wird statt der klassischen, numerisch problematischen Berechnung eine robuste und exakte Alternative verwendet.

Durch Einsatz von CGAL konnte der Algorithmus enorm robust gemacht werden. Für einen reibungslosen Einsatz ist allerdings auch erforderlich, dass der Marching Cubes Algorithmus robust und zuverlässig arbeitet. Es dürfen keine Dreiecksnetze mit missgebildete Topologien, wie zum Beispiel Kanten mit mehr als zwei benachbarten Dreiecken oder Löcher im Netz, produziert werden. Für die mit 2000 Eckpunkten vereinfachte Version des in Abbildung 3.8 angegebenen Dreiecksnetzes kommt der Algorithmus zur Bruchbildung nun auf weniger als eine Minute. Das Re-tiling braucht für dieses Netz etwa zwei Minuten, was zu einer gesamten Arbeitszeit von ungefähr drei Minuten führt.

Ergebnis

In Abbildung 3.11 ist der Einsatz des Re-tiling Algorithmus demonstriert.

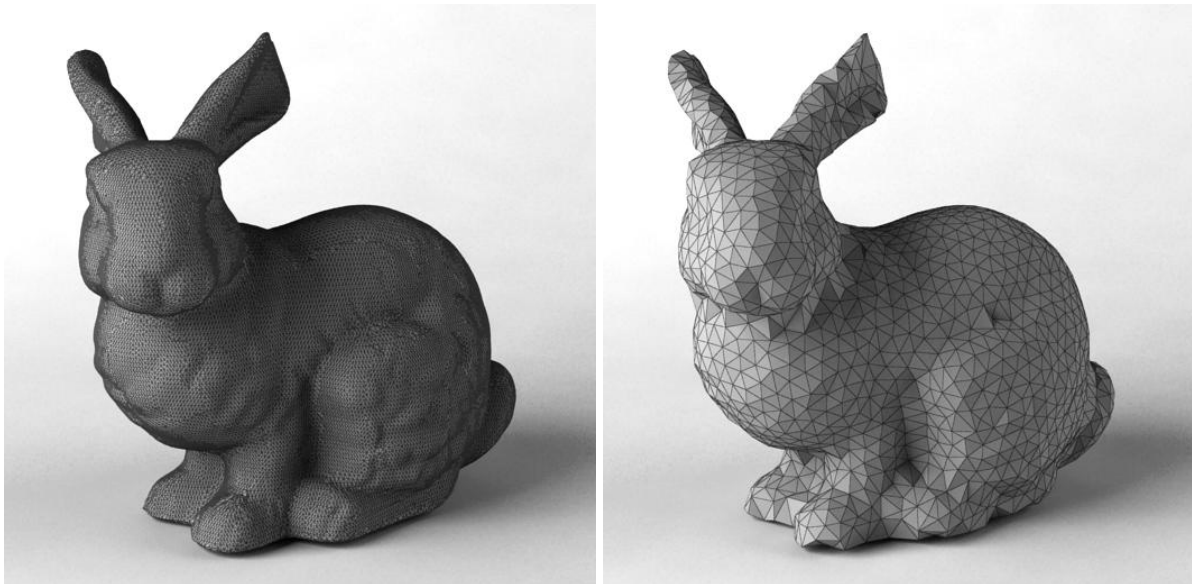


Abbildung 3.11: Das Ergebnis des Re-tiling Algorithmus vor(*links*) und nach der Ausführung(*rechts*).

Die Abbildung 3.12 zeigt das Ergebnis der Oberflächenbruchbildung auf einem Dreiecksnetz, welches vorher durch das Re-tiling irregularisiert wurde. Die visuelle Erscheinung der Brüche konnte wesentlich verbessert werden und entspricht viel mehr den erwarteten Bruchmustern. Auch die Geschwindigkeit beim erzeugen der Brüche ist durch das einfachere Netz gesteigert worden.

3.5 Zur Animation von Oberflächenbrüchen

Mit der Bruchbildung und dem Re-tiling ist es also möglich die charakteristischen Bruchmuster auf einer Fluidoberfläche unter Kenntnis der Oberflächenspannungen für einen bestimmten Simulationszustand zu erzeugen. Eine zeitliche Dynamik äußert sich bei Gletschern darin, dass sich Gletscherspalten in ihrer

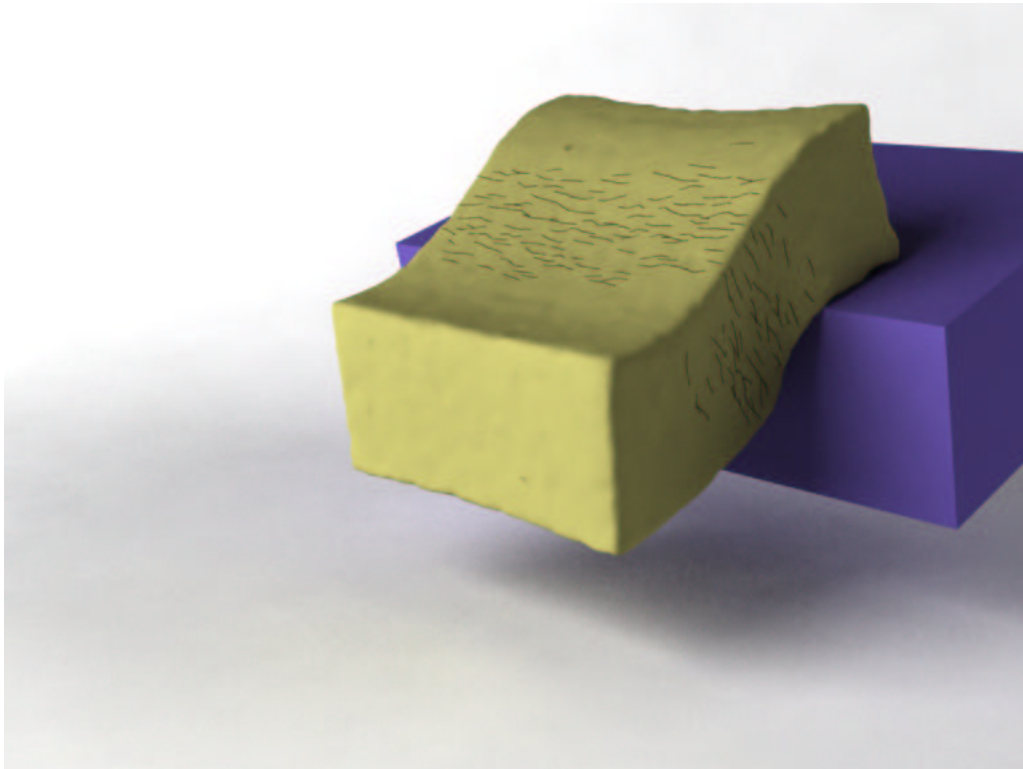


Abbildung 3.12: Ergebnis der Bruchbildung auf dem irregularisierten Netz

Länge ausbreiten und auch in Abhängigkeit vom plastischen Fluss des Gletschers weiten oder zusammenfallen. Für diese zeitliche Dynamik der Bruchsimulation ist es zuerst notwendig, bereits gebildete Brüche über den Verlauf der Simulation auf der Fluidoberfläche zu verfolgen. Dies ist problematisch da es zwischen zwei aufeinander folgenden Simulationsschritten keine Kohärenz bzgl. der Fluidoberfläche gibt. Durch das immer wieder neue abtasten der Isofläche des Fluids entsteht immer wieder ein völlig neues Netz mit gänzlich verschiedener Topologie.

Um die Brüche in irgend einer Art zu verfolgen müssen diese erst aus dem Dreiecksnetz extrahiert und beschrieben werden. Dies geschieht, indem bei der Bruchbildung protokolliert wird, welche Knoten versagen und welche Bruchkanten beim Vertexsplit gebildet werden. Diese Information ist später auch für die Darstellung der Brüche in Abschnitt 3.6 nützlich. Aus diesem Protokoll können dann die Brüche in Form von Knoten und Kantenzügen beschrieben und auch abgespeichert werden.

Zur Verfolgung der Brüche liegen mehrere Ansätze nahe. Ein Weg wäre, eine Textur auf der Fluidfläche zu verfolgen und die generierten Brüche in diese Textur zu rendern. Zur Verfolgung von Texturkoordinaten auf Fluidoberflächen sind Verfahren in [KAK⁺06], [SS07] und [BGOS06] vorgestellt worden. Neben der mangelnden Stabilität bzgl. der Kohärenz spricht besonders die Tatsache gegen diesen Weg, dass es keine mögliche Rückkopplung der Brüche auf die Simulation bzw. keine Möglichkeit der weiteren Animation der Brüche besteht, wenn sie erst einmal in eine Textur gezeichnet wurden.

Der Weg, welcher in dieser Arbeit beschrrieben wurde um das Problem zu lösen, besteht darin, die Bruchpunkte und Bruchkanten mit dem Geschwindigkeitsfeld zu advektieren. Danach werden diese auf das Dreiecksnetz des nächsten Simulationsschrittes projiziert. Die Bruchpunkte und Bruchkanten werden nicht direkt in das Dreiecksnetz aus dem Marching Cubes Algorithmus hinein modelliert, da dies die Topologie des ursprünglichen Bruches zerstören würde. Stattdessen werden die Bruchpunkte zu Eckpunktkandidaten des Re-tiling Algorithmus aus Kapitel 3.4. Dies stellt sicher das die Bruchpunkte als Eckpunkte im Dreiecksnetz enthalten bleiben. Damit die Positionen der Bruchpunkte durch das Re-tiling nicht verändert werden, muss der Algorithmus angepasst werden. Dazu wird die Struktur der Eckpunktkandidaten um eine Variable erweitert, welche angibt ob dieser Eckpunkt verschoben werden darf oder nicht. Diese Variable wird bei Ausführung des Repulsionsschrittes aus Abschnitt 3.4 berücksichtigt. Durch markieren der Eckpunktkandidaten, welche von Bruchpunkten stammen, wird erreicht, dass an den Stellen der Bruchpunkte nach Ausführung des Re-tiling wieder Eckpunkte zu finden sind. Durch die Repulsionskräfte welche von den fixen Bruchpunkten ausgehen, werden andere Eckpunktkandidaten weg gestoßen was auch begünstigt, dass durch eine Bruchkante benachbarte Bruchpunkte auch nach dem Re-tiling wieder durch eine Kante verbunden sind. Bei den seltenen Fällen, wo dies nicht der Fall ist, kann durch einen Nachbearbeitungsschritt, in dem das Viereck der zwei benachbarten Dreiecke neu tesseliert wird, die verbindende Bruchkante wieder hergestellt werden.

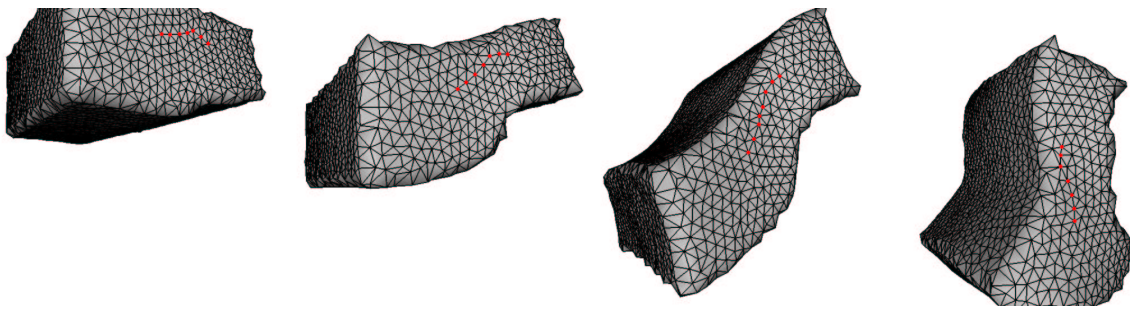


Abbildung 3.13: Verfolgung einer Bruchkante in einer Fluidsimulation - die roten Punkte markieren den Kantenzug, welcher über den gesamten Verlauf der Animation erhalten bleibt.

Auch bei dieser Methode ist die Stabilität der Kohärenz nicht in jedem Fall gewährleistet. In hochturbulenten und stark fragmentierenden Fluidsimulationen wird es schwierig die topologischen Features zu erhalten. Die Schwierigkeit liegt in der Rückprojektion der advectierten Bruchpunkte auf das Oberflächennetz des nächsten Simulationsschrittes. Hat sich hier eine zu starke Änderung ergeben, ist diese Projektion möglicherweise fehlerhaft, was zu unplausiblen Resultaten führt. Ein weiteres Problem ist der schwimmende Effekt an den Krümmungen der Fluidoberfläche. Dieses schwimmen entsteht dadurch, dass das Dreiecksnetz beim Re-tiling, durch die in jedem Bild erneut zufällig verteilten Eckpunktkandi-

daten, eine völlig andere Topologie aufweist, was sich an den Krümmungen besonders bemerkbar macht.

3.6 Zur Darstellung und Modellierung von Oberflächenbrüchen

Zur Darstellung der Brüche gibt es verschiedene Wege. Eine zentrale Eigenschaft die dabei genutzt werden kann ist, dass die Brüche selbst durch Bildung von Randkanten erzeugt werden. Dabei liegen zwei Randkanten eines Bruchs direkt übereinander. Dies deshalb, weil beim Vertexsplit in Kapitel 3.3.2 die zwei neuen Eckpunkte welche aus dem Bruchpunkt entstehen die selbe Position des Bruchpunktes erhalten.

Für eine Darstellung in Echtzeit bietet sich daher an, die Randkanten visuell hervorzuheben. Die OpenGL basierte Echtzeitdarstellung in Abbildung 3.14 wurde erzeugt, indem die Randkanten mit schwarzen Linien nachgezogen wurden. Dabei muss beachtet werden, dass die Z-Werte der Dreiecke mit einem bestimmten Offset in den Tiefenpuffer geschrieben werden müssen, um Artefakte durch Z-Fighting zu vermeiden.

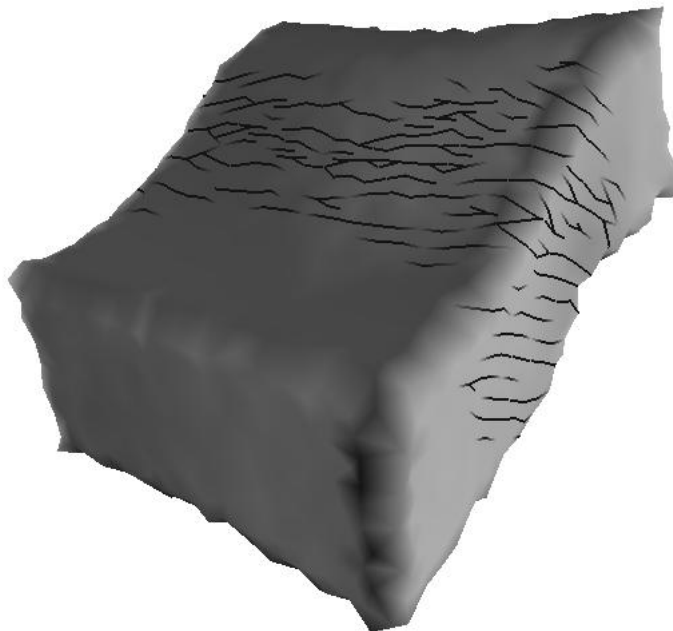


Abbildung 3.14: Echtzeitdarstellung der Brüche mit OpenGL

Um die Erzeugung qualitativ hochwertiger Bilder in DCC-Anwendungen zu demonstrieren ist im Rahmen dieser Arbeit ein Plugin für die Anwendung 3d-Studio-Max von Autodesk entwickelt worden. Dieses Plugin nutzt Klassen aus dem 3dsmax-SDK um die Anwendung mittels einer DLL um eine spezielle

Textur zu erweitern. Diese spezielle Textur kann von anderen Materialien in 3dsmax verwendet werden. Beim shading erzeugt die Textur Farbwerte in Abhängigkeit von der Entfernung des zu schattierenden Punktes von der nächsten Bruchkante. Unterschreitet die Entfernung einen vom Nutzer gesteuerten Schwellwert, wird eine ebenfalls parametrisierte Farbe zurückgegeben. Andernfalls wird ein Texturwert mit einer Deckkraft von null erzeugt, was dazu führt, dass die Textur den Ergebnisfarbwert nicht weiter beeinflusst.

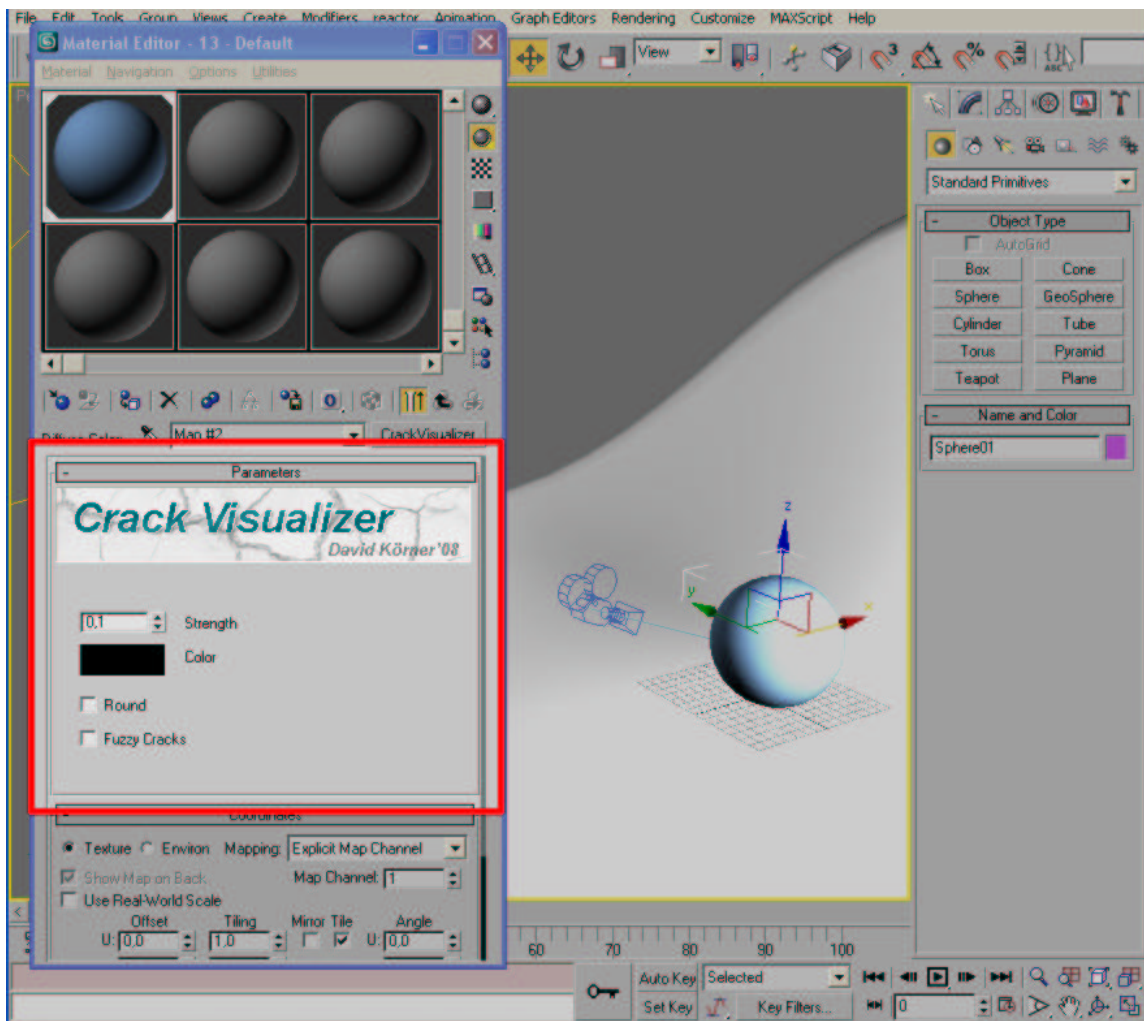


Abbildung 3.15: Das CrackVisualizer-Plugin für 3D-Studio-Max im Einsatz

Alle in dieser Arbeit gezeigten Ergebnisse sind mit dieser Methode erzeugt worden.

Die beiden vorgestellten Methoden stellen die Brüche als Linien dar. Das ist für Bilder bei denen die Entfernung zu den Brüchen so groß ist, dass die tatsächliche Ausdehnung der Bruchspalten vernachlässigt werden kann, eine effektive Lösung. Für Detailaufnahmen oder andere Szenarien ist das allerdings nicht ausreichend und die Bruchspalten sollten tatsächlich vorhanden sein. Dazu ist es möglich durch die während der Bruchbildung bekannten Knotenverschiebungen der beiden neuen Knoten des gespaltenen

Bruchknotens auch tatsächlich auseinander zu bewegen. In [IO06] wird das genutzt um zerbrochenes Glas und aufbrechende Lehmböden zu modellieren.

4 Zusammenfassung und Schlussfolgerung

In Abbildung 4.1 ist eine Gletschersituation simuliert und die Bruchbildung an der Gletscheroberfläche mit Hilfe des hier vorgestellten Verfahrens erzeugt worden. Die Bilder wurden wie in Abschnitt 3.6 berechnet.

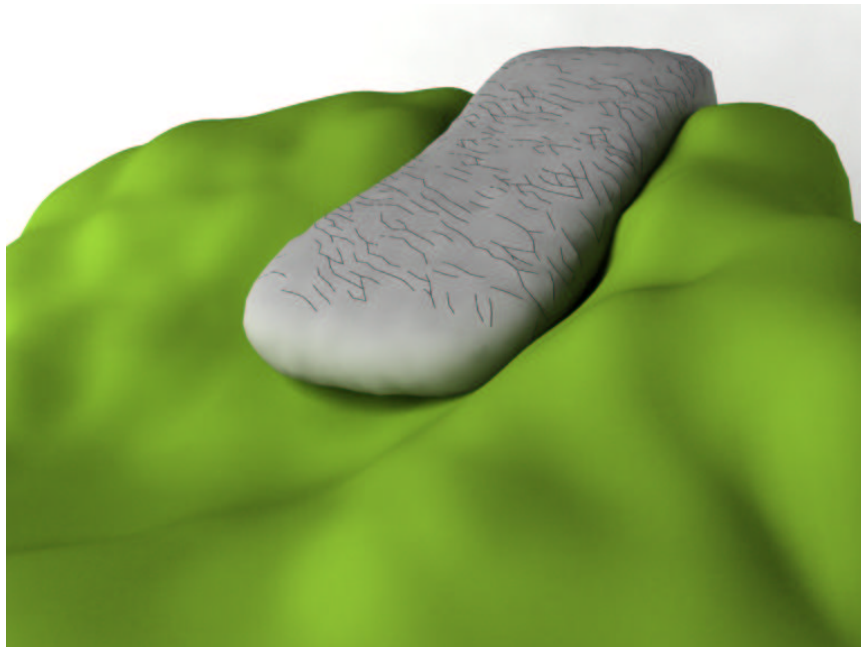


Abbildung 4.1: Ergebnis der Bruchbildung des Beispiels “Gebirge“ nach dem Retiling

Es gibt verschiedene Aspekte an denen der hier vorgestellte Ansatz sicher erweitert und verbessert werden kann. Am wichtigsten wäre dabei die Rückkopplung der Bruchbildung auf die Simulation. Momentan bewirken Brüche auf der Oberfläche keine Änderung der Fluidsimulation. Dies ist aber notwendig, damit Brüche nicht nur in einem Einzelbild, sondern auch über den gesamten Zeitverlauf der Simulation hinweg berechnet werden können. Weiterhin nachteilhaft ist der Grad des Aufwandes dieses Verfahrens. Mehrere Algorithmen, die jeweils eine sehr geringe Fehlertoleranz bezüglich ihrer Eingabedaten haben, werden zu einem Mehrschrittverfahren kombiniert. Die Dauer, welche sich bei der Szene “Gebirge“ aus Abbildung 4.1 auf ca. sechs Minuten pro Bild (inklusive Fluidsimulation, Retiling und Bruchbildung) subsumiert, machen eine flexible und intuitive Steuerung der Ergebnisse schwierig. Durch Optimierung alleine wird die notwendige Reduzierung der Berechnungsdauer sicher nicht zu bewerkstelligen sein.

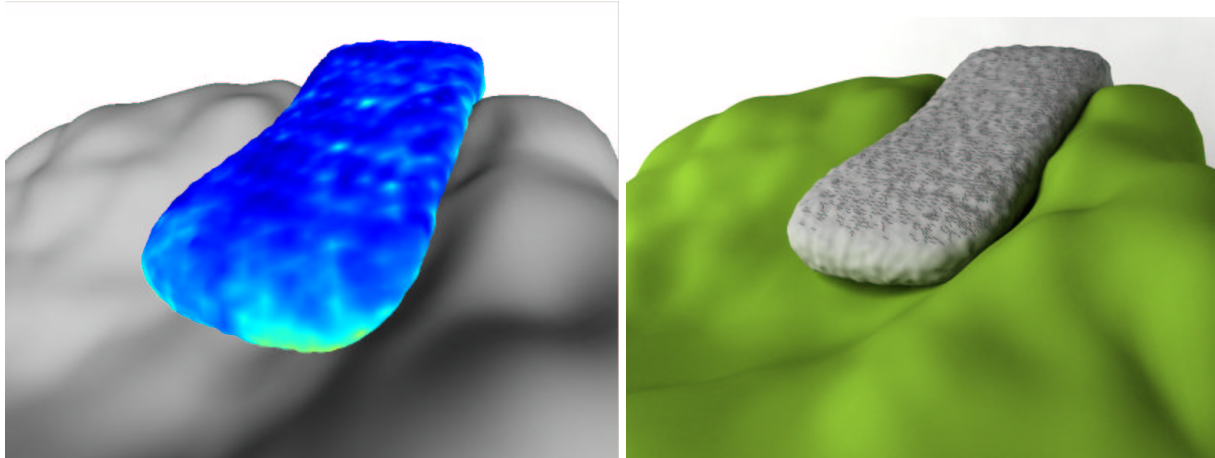


Abbildung 4.2: Ergebnis “Gebirge“ mit den Verteilungen der Spannungen(*links*) und den Ergebnissen der Bruchbildung vor dem Retiling (*rechts*)

Zur Erzeugung eines irregulären Oberflächennetzes zum Beispiel müssen vielleicht ganz andere Wege beschritten werden.

Über den hier vorgestellten Ansatz hinausgedacht gibt es einige Richtungen in die gegangen werden kann. Denkbar wäre vielleicht die Umstellung des Fluidsimulators auf eine Finite Elemente Methode. Damit könnte das größte Problem bei der Bruchbildung, nämlich die Modellierung der Diskontinuitäten besser angegangen werden. Die finiten Differenzen stellen diesbezüglich ein großes Hindernis dar und es könnte daher auch untersucht werden, inwieweit es möglich wäre solche Diskontinuitäten in einer auf finiten Differenzen basierenden Methode zu erzeugen. Eine finite Elemente Methode hingegen könnte unter Umständen auch direkt mit einem Verfahren zur Bruchbildung in Starrkörpern kombiniert werden, welches ebenfalls auf finiten Elementen basiert. Damit wäre es sicher leichter die Algorithmen aus der Starrkörperbruchbildung anzuwenden.

Schlussfolgerung

Das Thema der Bruchbildung in Fluiden ist meiner Meinung nach ein Thema, bei dem in der Computergraphik noch viel entdeckt und getan werden kann. Der Trend in der computergraphischen Fluidodynamik ist sehr stark die Untersuchung des Fließverhaltens von Fluiden. Realistische Animation und beeindruckende Bilder werden so erzeugt. Dies spiegelt sich auch in der jüngsten Siggraph (2008) wieder, bei welcher sich die Veröffentlichungen zum Thema mit der Viskosität und der Simulation zähflüssiger und dünnwandiger Fluidmassen beschäftigen ([BB08], [WT08]). Ich bin überzeugt das der Aspekt der Bruchbildung zu einer neuen Generation von Qualität in der computergraphischen Fluidsimulation führen kann. Nasser Schlamm ist eben nicht glatt, sondern bildet Risse aufgrund von Materialinhomoge-

nitäten aus. Nasser Sand zerbricht wenn man ihn auf den Boden fallen lässt und fließt nicht nur davon. Es kommt nicht nur auf das fließen an sich an, sondern auch auf das brechen. Materialversagen ist ein entscheidender Faktor für hochqualitative und überzeugende Bilder und Animationen. [OH99] haben mit ihrer Arbeit die Starrkörpersimulation auf ein völlig neues Qualitätsniveau gehoben. Und genau diese Möglichkeit sehe ich auch für die Fluidodynamik. Daher sollte in der Fluidodynamik der Computergraphik dieser Aspekt mehr untersucht werden. Das das nicht passiert liegt nicht zuletzt auch daran, dass das Thema sehr unübersichtlich ist. Verschiedene Theorien der Kontinuumsmechanik laufen hier zusammen und die Studie der Literatur kann in dieser Grauzone nur fragmentarisch Antworten geben. Auch das Spannungsverhältnis zwischen CFD und der Fluidsimulation in der Computergraphik, wo die Grenzen des physikalisch plausiblen ausgereizt werden, trägt zur Schwierigkeit dieses Themas bei.

Ich denke mit dieser Arbeit ist mir zumindest gelungen das Potential dieses Themas aufzuzeigen. Schon die einfache Kombination zweier Arbeiten jeweils aus dem Bereich der Bruchbildung und der Fluidodynamik haben es mir ermöglicht Bilder eines Phänomens zu ersimulieren, für welches es bisher in der Computergraphik noch gar keinen Ansatz gab. Dies ist ein erster Schritt und es ist meine volle Überzeugung, dass eine gemeinschaftliche, ausdauernde Auseinandersetzung mit dem Thema sicher noch mehr spannende Ergebnisse zu Tage fördern wird.

Literaturverzeichnis

- [BB08] BATTY, Christopher ; BRIDSON, Robert: Accurate Viscous Free Surfaces for Buckling, Coiling, and Rotating Liquids. In: *Proceedings of the 2008 ACM/Eurographics Symposium on Computer Animation*, 2008
- [BCD⁺99] BOISSONNAT, Jean-Daniel ; CAZALS, Frédéric ; DA, Frank ; DEVILLERS, Olivier ; PION, Sylvain ; REBUFAT, François ; TEILLAUD, Monique ; YVINEC, Mariette: Programming with CGAL: the example of triangulations. In: *SCG '99: Proceedings of the fifteenth annual symposium on Computational geometry*. New York, NY, USA : ACM, 1999. – ISBN 1-58113-068-6, S. 421–422
- [BGOS06] BARGTEIL, Adam W. ; GOKTEKIN, Tolga G. ; O'BRIEN, James F. ; STRAIN, John A.: A semi-Lagrangian contouring method for fluid simulation. In: *ACM Trans. Graph.* 25 (2006), Nr. 1, S. 19–38. – ISSN 0730-0301
- [Bri04] BRIDSON, Robert. *CS533b Course Notes: Animating Physics*. 2004
- [Car04] CARLSON, Mark T.: *Rigid, melting, and flowing fluid*. Atlanta, GA, USA, Diss., 2004. – Director-Greg Turk and Director-Peter J. Mucha
- [Chu07] CHUNG, T. J.: *General continuum mechanics*. Cambridge [u.a.] : Cambridge University Press, Jun 2007. – ISBN 978-0-521-87406-9
- [CMRBVHT02] CARLSON, Mark ; MUCHA, Peter J. ; R. BROOKS VAN HORN, III ; TURK, Greg: Melting and flowing. In: *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*. New York, NY, USA : ACM, 2002. – ISBN 1-58113-573-4, S. 167–174
- [CMT04] CARLSON, Mark ; MUCHA, Peter J. ; TURK, Greg: Rigid fluid: animating the interplay between rigid bodies and fluid. In: *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*. New York, NY, USA : ACM, 2004, S. 377–384
- [EMF02] ENRIGHT, Douglas ; MARSCHNER, Stephen ; FEDKIW, Ronald: Animation and rendering of complex water surfaces. In: *ACM Trans. Graph.* 21 (2002), Nr. 3, S. 736–744.

- ISSN 0730–0301
- [FF01] FOSTER, Nick ; FEDKIW, Ronald: Practical animation of liquids. In: *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. New York, NY, USA : ACM, 2001. – ISBN 1–58113–374–X, S. 23–30
- [FHL⁺07] FOUSSE, Laurent ; HANROT, Guillaume ; LEFÈVRE, Vincent ; PÉLISSIER, Patrick ; ZIMMERMANN, Paul: MPFR: A Multiple-Precision Binary Floating-Point Library with Correct Rounding. In: *ACM Transactions on Mathematical Software* 33 (2007), Juni, Nr. 2, S. 13. – Article 13, 15 pages
- [FM96] FOSTER, Nick ; METAXAS, Dimitri: Realistic animation of liquids. In: *Graph. Models Image Process.* 58 (1996), Nr. 5, S. 471–483. – ISSN 1077–3169
- [FSJ01] FEDKIW, Ronald ; STAM, Jos ; JENSEN, Henrik W.: Visual simulation of smoke. In: *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. New York, NY, USA : ACM, 2001. – ISBN 1–58113–374–X, S. 15–22
- [GBO04] GOKTEKIN, Tolga G. ; BARGTEIL, Adam W. ; O'BRIEN, James F.: A method for animating viscoelastic fluids. In: *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH 2004)* 23 (2004), Nr. 3, S. 463–468
- [HB06] HANKE-BOURGEOIS, Martin: *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens*. 2. Auflage. Teubner Verlag, Feb 2006. – ISBN 3–8351–0090–4
- [Hed99] HEDFORDS, Jim. *Simulating Landslides for Natural Disaster Prevention Jean-Dominique Gascuel, Marie-Paule Cani-Gascuel, Mathieu Desbrun* y. Jun 1999
- [Hed04] HEDFORDS, Jim: Force Budget Analysis of Glacier Flow. In: *Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology*, ISSN 1104-232X ; 963 (2004), S. 71. ISBN 91–554–5936–6
- [HW65] HALOW, F. H. ; WELCH, J.E. *Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface*. 1965
- [IO06] IBEN, Hayley N. ; O'BRIEN, James F.: Generating Surface Crack Patterns. In: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2006, S. 177–185
- [JL01] JUNG, Michael ; LANGER, Ulrich: *Methode der finiten Elemente für Ingenieure: Ei-*

- ne Einführung in die numerischen Grundlagen und Computersimulation.* 1. Auflage. Teubner Verlag, Jun 2001. – ISBN 978–3519029731
- [KAK⁺06] KWATRA, Vivek ; ADALSTEINSSON, David ; KWATRA, Nipun ; CARLSON, Mark ; LIN, Ming C.: Texturing fluids. In: *SIGGRAPH '06: ACM SIGGRAPH 2006 Sketches*. New York, NY, USA : ACM, 2006. – ISBN 1–59593–364–6, S. 63
- [Kel06] KELAGER, Micky. *Lagrangian Fluid Dynamics Using Smoothed Particle Hydrodynamics*. 2006
- [LC87] LORENSEN, William E. ; CLINE, Harvey E.: Marching cubes: A high resolution 3D surface construction algorithm. In: *SIGGRAPH Comput. Graph.* 21 (1987), Nr. 4, S. 163–169. – ISSN 0097–8930
- [LM98] LEROI, Eric ; MIRGON, Carola. *Simulating Landslides for Natural Disaster Prevention* Jean-Dominique Gascuel, Marie-Paule Cani-Gascuel, Mathieu Desbrun y. 1998
- [MKN⁺04] MÜLLER, M. ; KEISER, R. ; NEALEN, A. ; PAULY, M. ; GROSS, M. ; ALEXA, M.: Point based animation of elastic, plastic and melting objects. In: *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Aire-la-Ville, Switzerland, Switzerland : Eurographics Association, 2004. – ISBN 3–905673–14–2, S. 141–151
- [MMDJ01a] MÜLLER, Matthias ; MCMILLAN, Leonard ; DORSEY, Julie ; JAGNOW, Robert: Real-time simulation of deformation and fracture of stiff materials. In: *Proceedings of the Eurographic workshop on Computer animation and simulation*. New York, NY, USA : Springer-Verlag New York, Inc., 2001. – ISBN 3–211–83711–6, S. 113–124
- [MMDJ01b] MÜLLER, Matthias ; MCMILLAN, Leonard ; DORSEY, Julie ; JAGNOW, Robert: Real-time simulation of deformation and fracture of stiff materials. In: *Proceedings of the Eurographic workshop on Computer animation and simulation*. New York, NY, USA : Springer-Verlag New York, Inc., 2001. – ISBN 3–211–83711–6, S. 113–124
- [NF99] NEFF, Michael ; FIUME, Eugene: A visual model for blast waves and fracture. In: *Proceedings of the 1999 conference on Graphics interface '99*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 1999. – ISBN 1–55860–632–7, S. 193–202
- [Oer99] OERTEL, Herbert: *Strömungsmechanik*. Braunschweig : Vieweg Verlag, 1999
- [OH99] O'BRIEN, James F. ; HODGINS, Jessica K.: Graphical modeling and animation of brittle fracture. In: *Proceedings of ACM SIGGRAPH 1999*, ACM Press/Addison-Wesley

- Publishing Co., August 1999. – ISBN 0–201–48560–5, S. 137–146
- [SAC⁺99] STORA, Dan ; OLIVIER AGLIATI, Pierre ; PAULE CANI, Marie ; NEYRET, Fabrice ; DOMINIQUE GASCUEL, Jean: Animating lava flows. In: *in Graphics Interface*, 1999, S. 203–210
- [Spi20] SPIELER, Oliver: Die Fragmentierung hochviskoser Magmen. In: *PhD Thesis, Ludwig-Maximilians-Universität München, Geowissenschaften* (2001-08-20)
- [SS07] SHEN, Chen ; SHAH, Apurva: Extracting and Parametrizing Temporally Coherent Surfaces from Particles. In: *Pixar Technical Memo 07-05* (2007)
- [Sta99] STAM, Jos: Stable fluids. In: *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. New York, NY, USA : ACM Press/Addison-Wesley Publishing Co., 1999. – ISBN 0–201–48560–5, S. 121–128
- [TF88] TERZOPOULOS, Demetri ; FLEISCHER, Kurt: Modeling inelastic deformation: viscoelasticity, plasticity, fracture. In: *SIGGRAPH Comput. Graph.* 22 (1988), Nr. 4, S. 269–278. – ISSN 0097–8930
- [Tur92a] TURK, Greg: Re-tiling polygonal surfaces. In: *SIGGRAPH Comput. Graph.* 26 (1992), Nr. 2, S. 55–64. – ISSN 0097–8930
- [Tur92b] TURK, Greg: *Texturing surfaces using reaction-diffusion*. Chapel Hill, NC, USA, Diss., 1992
- [VTPF07] VETTERLING, William T. ; TEUKOLSKY, Saul A. ; PRESS, William H. ; FLANNERY, Brian P.: *Numerical Recipes in C++, The Art of Scientific Computing*. Third Edition. Cambridge : Cambridge University Press, 2007. – ISBN 0521880688
- [WT08] WOJTAN, Chris ; TURK, Greg: Fast viscoelastic behavior with thin features. In: *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*. New York, NY, USA : ACM, 2008, S. 1–8
- [ZB05] ZHU, Yongning ; BRIDSON, Robert: Animating sand as a fluid. In: *ACM Trans. Graph.* 24 (2005), Nr. 3, S. 965–972. – ISSN 0730–0301