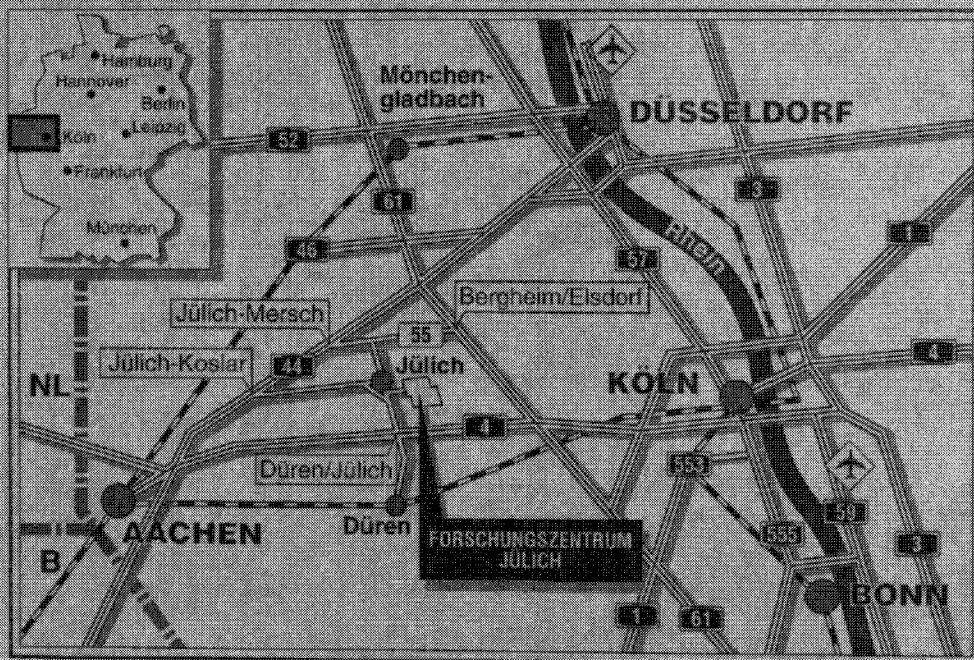


Höchstleistungsrechenzentrum

**Molekulardynamik-Simulationen
beliebig geformter Teilchen
in zwei Dimensionen**

Hans-Jürgen Tillemans



Berichte des Forschungszentrums Jülich ; 3038

ISSN 0944-2952

Höchstleistungsrechenzentrum Jüli-3038

D38 (Diss. Universität Köln)

Zu beziehen durch: Forschungszentrum Jülich GmbH · Zentralbibliothek
D-52425 Jülich · Bundesrepublik Deutschland

Telefon: 02461/61-6102 · Telefax: 02461/61-6103 · Telex: 833556-70 kfa d

**Molekulardynamik-Simulationen
beliebig geformter Teilchen
in zwei Dimensionen**

Hans-Jürgen Tillemans

Inhaltsverzeichnis

1	Einleitung	3
2	Molekulardynamik beliebig geformter Teilchen	11
2.1	Integration der Bewegungsgleichungen	11
2.2	Kraftberechnung	14
2.3	Bestimmung der Nachbarn	21
2.4	Verbinden der Zellen mit Stäben	22
3	Voronoi Gitter	27
3.1	Einleitung	27
3.2	Voronoi- und Delaunay-Gitter	28
3.3	Algorithmen	29
3.3.1	Vektorisierbare Zufallsgitter	29
3.3.2	Das Poisson'sche Zufallsgitter	32
3.3.3	Dynamische Zufallsgitter	36
3.4	Ergebnisse	42
3.5	Zusammenfassung	42
4	Simulation von Scherzellen	43
4.1	Einleitung	43
4.2	Das numerische Experiment	44
4.3	Scherkraft und Dilatanz	48
4.4	Scherbänder	57
4.5	Zusammenfassung	62
5	Simulation von Erdbeben	65
5.1	Einleitung	65
5.2	Das Modell	67
5.3	Ergebnisse	70
5.4	Ausblick	72
6	Der granulare Fluß durch Trichter	75
6.1	Einleitung	75

6.2	Das numerische Modell	76
6.3	Die granularen Flußregime	77
6.4	Flußprofile	83
6.5	Diskussion	84
7	Schüttgut in einer sich drehenden Trommel	91
7.1	Einleitung	91
7.2	Das Modell	93
7.3	Ergebnisse	95
8	Weitere Anwendungen	99
8.1	Bruchexperimente	99
8.2	Kristalle aus Pentagonen	103
9	Schlußbetrachtungen und Ausblick	107
A	Das Simulationsprogramm	111
A.1	Das Parameterfile	113
A.2	Das Hauptprogramm	114
A.3	Berechnung des Überlappungspolygons	120
A.4	Berechnung der Stabkräfte	127
A.5	Bewegung der Stäbe	129

Kapitel 1

Einleitung

In dieser Arbeit werden einige Aspekte des physikalischen Verhaltens granularer Medien beleuchtet. Dies sind Substanzen, die intern auf mesoskopischer Größenskala aus Teilchen bestehen und deren physikalisches Verhalten aufgrund von Energiedissipation der Teilchen untereinander zwischen dem von Festkörpern und Flüssigkeiten liegt. Es handelt sich dabei um so verschiedene Materialien wie Sand, Kies, Getreide, Felsgestein, Schnee oder auch auf einem Fluß treibende Eisschollen, um nur einige zu nennen. Solche Materialien sind nicht nur von theoretischer Bedeutung, sondern spielen auch im Alltagsleben eine große Rolle. Der Fluß granularer Medien ist wichtig, wenn man Phänomene wie das Auslaufen von Schüttgütern aus Bunkern und Trichtern, Schnee- und Steinlawinen im Gebirge oder Erdbeben näher verstehen möchte. Aufgrund dieser Bedeutung bezeichnet man die granularen Medien seit einiger Zeit auch als den fünften Aggregatzustand (die ersten vier sind Festkörper, Flüssigkeit, Gas und Plasma).

Unser Hauptanliegen ist es, die Gesetzmäßigkeiten im Zusammenhang mit der Scherung granularer Medien zu verstehen. Wir verwenden zur Untersuchung dieser Frage ein ComputermodeLL, in dem wir die Teilchen als beliebig geformte Polygone betrachten. In anderen Ansätzen werden die Teilchen oft als sphärisch angenommen. Solche Modelle sind sowohl einfacher zu realisieren, als auch schneller im Hinblick auf die verwendete Rechenzeit.

Wir distanzieren uns von anderen Zugängen, die die Teilchen als sphärisch annehmen. Der Preis, den wir dafür zu zahlen haben, daß wir diese komplizierteren aber realistischeren Formen annehmen, ist etwa ein Faktor zehn an Rechenzeit gegenüber den Simulationen mit runden Teilchen. Auf der anderen Seite bieten unregelmäßig geformte Polygone gegenüber runden Teilchen einige Vorteile. Zum einen ist unser Modell in der Lage, die sterischen Effekte korrekt wiederzugeben, die durch die geometrische Reibung der Teilchen zustandekommen. Die Teilchen können sich ineinander verkanten und verhaken. Diese geometrisch bedingte Reibung ist eigentlich immer wichtig, wenn man es mit Teilchen zu tun hat, die nicht sphärische Formen haben, wie dies bei den allermeisten Schüttgütern der Fall ist. Ein ähnliches Simulationsmodell wie das von uns verwendete wurde erstmals von Cundall eingeführt [41, 42]. Er bezeichnete es als DEM (Discrete Element Method). Später haben auch O. Walton [182, 183, 184] und

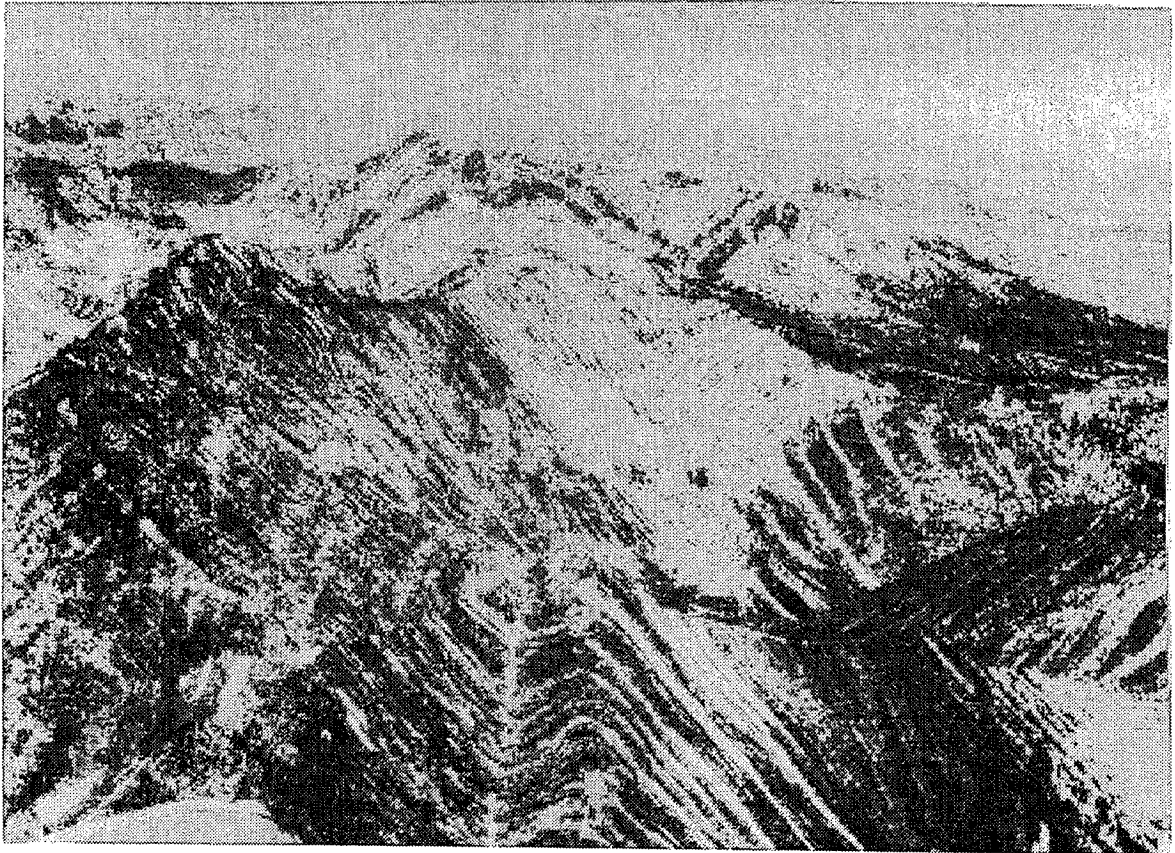


Abbildung 1.1: Große Scherbänder nördlich des Borah Peak, Idaho. Blick in nordwestlicher Richtung, entnommen aus Ludman 1982, S. 371.

A. V. Potapov [133, 134] Molekulardynamik-Simulationen mit unregelmäßig geformten Polygonen durchgeführt.

Ein weiterer Nachteil von sphärischen Teilchen, der sich bei der Verwendung von komplexeren Teilchen wie wir sie benutzen nicht ergibt, tritt zum Vorschein, wenn man sehr dicht gepackte Systeme untersuchen will. Sphärische Teilchen ordnen sich nämlich bevorzugt in Form eines regelmäßigen Gitters an (s. auch Kapitel 7.3), so daß man ähnliche Nachteile in Kauf nehmen muß wie bei Simulationen, denen ein Gitter zugrunde liegt. Es ist leicht möglich, daß man die Gittersymmetrie in den Simulationsergebnissen wiederfindet [50]. Um solche Anisotropieeffekte zu vermeiden, benutzen wir als Ausgangspunkt unserer Simulationen Zufallsgitter, in unserem Fall Voronoi-Gitter, die wir in Kapitel 3 beschreiben werden. Wir verwenden für die Form unserer Teilchen die Zellen solcher Voronoi-Gitter. Dadurch können wir Anisotropien weitgehend ausschließen.

Das Hauptanwendungsgebiet unseres Modells besteht in der Untersuchung großer Deformationen, insbesondere der Scherung dicht gepackter granularer Medien und Fels-
gestein. Die Deformationen, die bei solchen Scherungen auftreten, finden normalerweise



Abbildung 1.2: Scherbänder aus Sandstein auf kleiner geophysikalischer Skala. Aufnahme nördlich des Mt. Carmel, Utah, entnommen aus Ludman 1982, S. 371.

nicht gleichmäßig im Material statt, sondern konzentrieren sich in lokalisierten Zonen, die auch als Scherbänder bezeichnet werden. Von den Metallen sind solche Scherbänder schon lange bekannt. Die mikroskopischen Mechanismen, die diesen zugrunde liegen, sind jedoch in beiden Fällen unterschiedlich. Während bei den Metallen die Versetzungen eine wichtige Rolle spielen, liegen den Deformationen bei granularen Medien und Gestein lokale Rotationen zugrunde wie sie auch in der Cosserat-Theorie berücksichtigt werden. Das Material bricht bei Deformation an einer Stelle auf, dilatiert und gleitet dann über einer Schicht von rotierenden Teilchen. Wirkt auf das System ein hoher hydrostatischer Druck, so wird das Phänomen der Lokalisation begünstigt. Von Kàrmàn [93] konnte 1911 in einem Experiment mit Carrara Marmor zeigen, daß sich dieser bei genügend hohem Druck plastisch verhält und Scherbänder auftreten, s. Abb. 1.3.

Einerseits untersuchen wir Scherzellen, wie sie in der Verfahrenstechnik zur Untersuchung von Schüttgütern angewendet werden. Auf der anderen Seite betrachten wir auch eine Scherzelle auf geophysikalischer Größenskala, nämlich die Scherung zweier tektonischer Platten, wie sie z. B. bei der San Andreas Spalte realisiert ist, wo die Pazifische und die Nordamerikanische Platte angetrieben durch Konvektionsrollen im oberen Erdmantel gegeneinander geschert werden. Unser Modell verursacht in natürli-

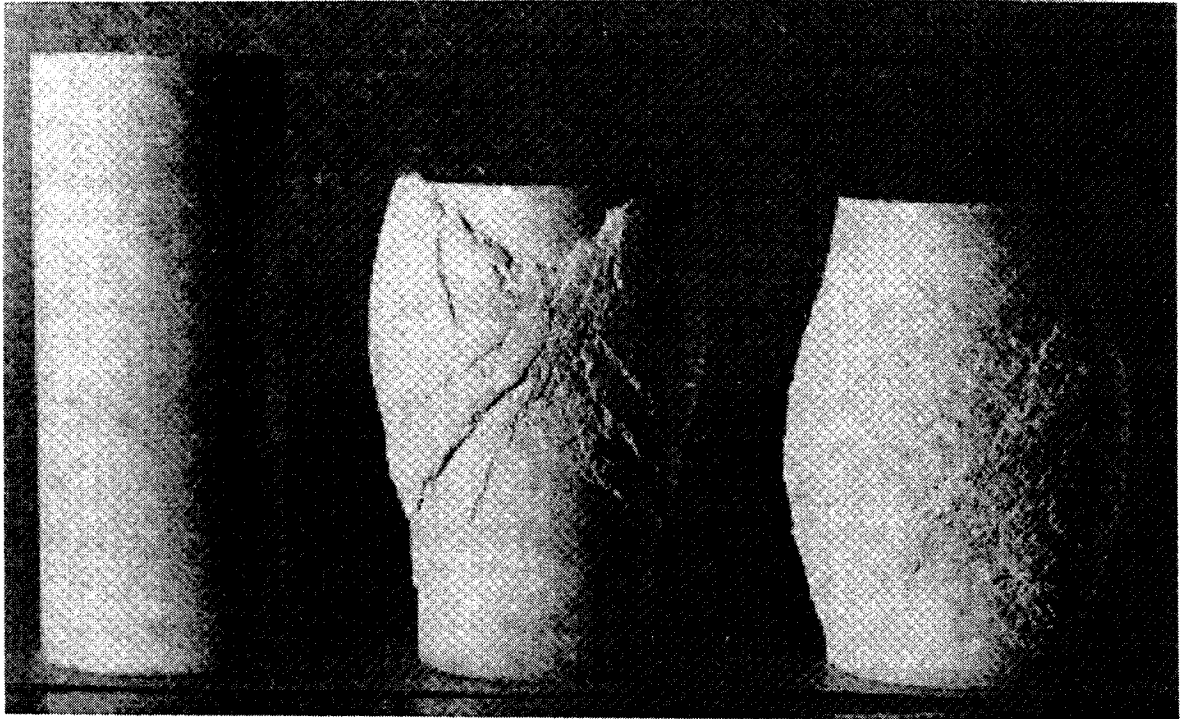


Abbildung 1.3: *Experiment mit Marmor in einer triaxialen Zelle. Bei einem kleinen hydrostatischen Druck von 200 atm verformt sich der Marmorzylinder unter Bildung von Brüchen, während er sich bei einem hohen hydrostatischen Druck von 445 atm duktil verformt. In beiden Fällen ist die Bildung von Scherbändern sehr schön zu erkennen. Entnommen aus Ludman 1982, S. 373.*

cher Weise Erdbeben, die dadurch zustande kommen, daß Zellen eingeklemmt werden, und dann, wenn die Spannung zu groß wird, sich diese ruckartig bewegen. Die Anzahl von Erdbeben einer gewissen Größe folgt dem Gutenberg-Richter Gesetz, wie es aus der Geophysik bekannt ist. Danach ist die Häufigkeitsverteilung der Erdbeben mit der freigesetzten Energie ein Potenzgesetz mit dem Exponenten -1.5.

In unseren Untersuchungen von Scherzellen spielen Scherbänder eine bedeutende Rolle. Dabei handelt es sich um Zonen im Material, in denen sich die Scherung lokalisiert. Der Bereich des Materials außerhalb der Scherbänder bewegt sich dabei im wesentlichen homogen als Ganzes. Diese Art von lokalisierter Deformation findet man in vielen Materialien, wie z. B. in Sand, Tonerde, Böden, Gestein und Metallen. Man findet solche Scherbänder, die oft sehr ästhetisch anmuten, auf vielen Größenskalen. In Abb. 1.1 kann man Scherbänder erkennen, die ein ganzes Gebirge überspannen. Abb. 1.2 zeigt Scherzonen, die sich auf kleinerer Skala gebildet haben und ein Felsmassiv abdecken. Schließlich erkennt man in Abb. 1.3 experimentell, mit Hilfe einer triaxialen Zelle hergestellte Scherbänder in Marmor. Das Phänomen der lokalisierten Zonen ist auch im Zusammenhang mit der Entstehung von Brüchen wichtig, da diese bevorzugt dort ent-

stehen, wo sich eine Scherzone ausgebildet hat. Hier nämlich ist die lokale Deformation im Material am größten und daher wird an dieser Stelle die für das Material geltende Bruchgrenze am ehesten überschritten. Diese Art von durch Lokalisation induzierten Rissen kann man sehr oft an tiefen Fundamenten, tiefen Tunneln und an Ölplattformen beobachten. Viele theoretische, experimentelle und auch numerische Arbeiten haben sich schon mit dem Problem der Scherbänder beschäftigt. Die theoretischen Arbeiten bauen auf der Kontinuumsstheorie auf und betrachten den Übergang zu den Scherbändern als einen Verlust der Eindeutigkeit der Lösungen der Kontinuumsungleichungen (Bifurkation) [48, 149, 172, 174]. Die experimentellen Arbeiten zu den Scherbändern wurden vor allem an unterschiedlichen Arten von Sand durchgeführt und liefern gut reproduzierbare Werte für die Dicke und die Orientierung der Bänder in Abhängigkeit vom hydrostatischen Druck und der Grobkörnigkeit des Sandes [8, 9, 47]. Eine ausgefeilte Technik zur Untersuchung von Sandproben ist z. B. die stereophotogrammetrische Methode, wie sie in der Dissertation von Mokni [116] für die Untersuchung der Scherbänder in bi- und triaxialen Tests Verwendung fand. Aus Experimenten ist bekannt, daß sich Scherbänder nicht spontan bilden, sondern an einem punktförmigen Keim, z. B. einer Unreinheit des Materials, entstehen und von dort aus wachsen [48, 174]. An den rigiden Rändern der Scherzelle können die Bänder reflektiert werden und an elastischen Rändern enden sie. Auch können sich mehrere Scherbänder unabhängig voneinander bilden, die abrupt beginnen und enden [55]. Weiterhin gibt es Scherbänder, die nicht durchgehend sind, sondern aus mehreren Teilstücken bestehen [48]. Eng verbunden mit dem Phänomen der Scherzonen ist die sogenannte Dilatanz, d. h. die volumetrische Ausdehnung des Materials. Desrues [48] fand bei Scherexperimenten mit dichtem Sand, daß die Entstehung von Scherbändern mit einem enormen Anstieg der Dilatanz einhergeht. Nachdem sich diese vollständig gebildet haben, wächst die Dilatanz dann nicht weiter an. Außerdem beobachtete er, daß die Volumenvergrößerung auf die Scherzone beschränkt ist. Solche Experimente mit großen Scherverschiebungen wurden auch in [146] angestellt und die Ergebnisse passen mit denen von Desrues zusammen. In unseren Simulationen in Kapitel 4 bestätigen wir diese Aussage.

Für die Orientierung der Scherbänder gibt es verschiedene theoretische Voraussagen. Dabei spielen zwei Winkel eine Rolle, nämlich der Dilatanzwinkel und der Reibungswinkel. [8, 9, 173]. Jedoch sind die experimentellen Befunde sehr unterschiedlich und unterstützen keine dieser Theorien. Durch diese beiden Winkel sind jedoch der maximale und der minimale Winkel, der für die Orientierung der Scherbänder möglich ist, in etwa eingegrenzt. Einigen Experimenten zufolge verlaufen die Scherbänder entlang der Richtung der Scherbewegung [8, 9, 48, 155, 173]. Wir werden sehen, daß dies auch bei der von uns verwendeten Scherzellengeometrie der Fall ist.

Die experimentellen Befunde zur Breite der Scherbänder [48, 162] besagen, daß diese mit dem hydrostatischen Druck, der auf die Scherzelle wirkt, abnimmt. In dichterem Sand sind die Scherbänder enger als in relativ lockerem Sand. Je höher der hydrostatische Druck ist, um so später werden die Scherbänder gebildet [162]. Unreinheiten haben die Eigenschaft, Kristallisationskeime für Scherbänder darzustellen [48, 174]. Es darf heute als allgemein anerkannt betrachtet werden, daß bei vielen Materialien die

Scherverschiebung sehr stark lokalisiert ist [8, 9, 47, 49, 162, 172, 173, 174].

Unsere Arbeit gliedert sich wie folgt. Zunächst gehen wir im Kapitel 2 auf die simulationstechnischen Aspekte des Computermodells ein. Wir beschreiben dort, wie wir die elastischen Kräfte zwischen den Teilchen berechnen. Es wird sich dabei herausstellen, daß die von uns verwendete Methode, bei der die elastische Kraft proportional zum geometrischen Überlappungsvolumen der Teilchen ist, einem linearen Kraftgesetz beim Kontakt zweier Seiten miteinander und einem konischen Kraftgesetz beim Kontakt einer Ecke mit einer Seite entspricht. Wir beschreiben hier auch die Kräfte, die für die Energiedissipation und die Gleitreibung verantwortlich sind. Ebenso beschreiben wir eine Methode, die es uns erlauben wird, Bruch- und Sintervorgänge zu implementieren. Dabei werden Stäbe durch elastische Stäbe miteinander verbunden. Durch Entfernen von Stäben kann man Bruchphänomene und durch Einsetzen von neuen Stäben Sintervorgänge modellieren.

In Kapitel 3 beschreiben wir schnelle Algorithmen mit deren Hilfe man Voronoi-Zellen generieren kann. Dabei handelt es sich um eine dichte Parkettierung eines Teils der Ebene mit unregelmäßig geformten Polygonen. Wir werden solche Parkettierungen als Anfangskonfigurationen für unsere Scherzellenexperimente benutzen. Im Gegensatz zu regulären Gittern sind solche Zufallsgitter frei von Anisotropien [117, 118]. Die Ergebnisse aus Kapitel 3 werden veröffentlicht in [98].

In Kapitel 4 werden die Ergebnisse von Scherzellenexperimenten vorgestellt. Wir finden die Abhängigkeit der Scherkraft sowohl von der auf die Scherzelle wirkenden Normalkraft als auch von der Schergeschwindigkeit. Wir messen eine Zunahme der Scherkraft mit der Schergeschwindigkeit, die als Scherhärtung bezeichnet wird. Außerdem stellen wir fest, daß die Scherung nicht gleichmäßig in der ganzen Zelle stattfindet, sondern in einem kleinen Bereich lokalisiert ist, was sehr gut mit den experimentellen Befunden über Scherzellen übereinstimmt. Wir messen die Breite dieser Scherzonen und stellen fest, daß diese sowohl von der Schergeschwindigkeit als auch von der Normalkraft abhängt. Je höher die Normalkraft ist, um so dünner ist das Scherband. Auch dieses Ergebnis stimmt mit den oben vorgestellten experimentellen Beobachtungen überein. Die Ergebnisse aus Kapitel 4 werden veröffentlicht in [52].

Im Kapitel 5 beschäftigen wir uns mit der Scherung auf einer geophysikalischen Größenskala, nämlich mit der Bewegung von tektonischen Platten, die sich tangential zueinander bewegen. Ein Beispiel für solche Plattenbewegungen ist die Pazifische Platte, die sich entlang der Nordamerikanischen Platte bewegt. Dabei hat sich die San Andreas Spalte gebildet. Wir beobachten, wie sich in unserem Modell eine sehr stark lokalisierte Zone bildet und messen die Häufigkeitsverteilung von Erdbeben. Wir finden, daß unser Modell das Gutenberg-Richter-Gesetz wiedergibt.

Dann wenden wir uns in Kapitel 6 der ingenieurtechnisch relevanten Frage zu, unter welchen Umständen ein Trichter oder Bunker, der mit Schüttgütern gefüllt wird, verstopft und vollläuft. Wir verwenden dabei zur Untersuchung der verschiedenen Flußregime eines generischen Trichters quadratisch geformte Teilchen. Es zeigt sich, daß es je nach Einfüllrate vier verschiedene Flußbereiche gibt, in denen der Trichter ein unterschiedliches Verhalten zeigt. Ferner untersuchen wir auch einen komplizierter geformten

Trichter mit schrägem Rohransatz, wie er eher den Gegebenheiten in realen industriellen Anwendungen entspricht, und finden durch Analyse der Flußprofile Totwasserbereiche, die nur wenig zum granularen Fluß beitragen.

In Kapitel 7 beschäftigen wir uns mit dem granularen Fluß in Drehtrommeln. Dabei verwenden wir kompliziert aufgebaute Teilchen mit einer inneren Struktur. Wir setzen jeweils vier dreieckige Teilchen mit Hilfe der in Kapitel 2 vorgestellten Stäbe zu einem quadratischen Teilchen zusammen. Energiedissipation führen wir hier dadurch ein, daß die Bewegung der Stäbe gedämpft ist. Bei einer relativ langsamen Umdrehungsgeschwindigkeit der Trommel beobachten wir die Bildung von Lawinen. Weiterhin ist ein Winkel, den die Granulatoberfläche mit der Horizontalen bildet dem experimentell gemessenen sehr nahe.

In Kapitel 8 stellen wir kurz zwei weitere Anwendungsmöglichkeiten vor, bei denen aber nur qualitative Resultate vorliegen. Zum einen handelt es sich dabei um die Simulation von Bruchexperimenten hoher Geschwindigkeit. Wir modellieren dabei einen Betonblock, indem wir ihn aus mit Stäben verbundenen Voronoi-Zellen aufbauen, die bei einer bestimmten Belastung brechen. Es bilden sich realistisch anmutende Bruchmuster. Weiterhin stellen wir Simulationen von Kristallisationsphänomenen mit regulären Fünfecken vor. Es bilden sich sowohl in realen Experimenten auf dem großen Luftkissentisch der Universität Rennes als auch in unseren Simulationen Anordnungen, die einer kristallinen Struktur sehr nahe kommt, die sich dadurch auszeichnet, das sie die maximal mögliche Packungsdichte für regelmäßige Fünfecke hat.

Schließlich fassen wir unsere Ergebnisse zusammen und diskutieren Erweiterungsmöglichkeiten. Im Anhang A sind die wichtigsten Routinen des Programms abgedruckt und kurz beschrieben.

Kapitel 2

Molekulardynamik beliebig geformter Teilchen

In dieser Arbeit soll die Bewegung von komplexen Objekten in der Ebene verfolgt werden. Wir beschränken uns auf konvexe Polygone. Man kann aber auch noch kompliziertere Objekte behandeln, indem man diese, falls sie nicht konvex sind, zunächst in konvexe Unterzellen zerteilt und diese dann durch Polygone approximiert. Für die Berechnung der Bahnkurven und der Rotationen der polygonalen Teilchen verwenden wir die Molekular-Dynamik Methode. Dies ist eine Technik, die es erlaubt, die klassischen Bewegungsgleichungen für die Translations- bzw. Rotationsbewegung von N starren Körpern im Raum zu lösen [3]. Wir behandeln hier Teilchen in der zweidimensionalen Ebene, so daß wir drei Bewegungsgleichungen lösen müssen, nämlich zwei für die Translationsbewegung und eine für die Rotation der Teilchen um ihre eigene Achse. Im dreidimensionalen Raum würden wir mit sechs Bewegungsgleichungen konfrontiert, nämlich drei für die Translation und drei für die verschiedenen Rotationsbewegungen, entsprechend den drei Euler-Winkeln. Unser Modell ist zweidimensional, jedoch ist es an einigen Stellen besser, sich das System als dreidimensional vorzustellen und die Polygone als polygonale Zylinder mit einer Dicke d zu betrachten, um z. B. die Masse der Teilchen aus der geometrischen Form und der Dichte berechnen zu können.

2.1 Integration der Bewegungsgleichungen

Wir müssen für jedes starre Polygon i die Newtonschen Gleichungen lösen:

$$m_i \ddot{\vec{r}}_i = \sum_{j=1}^N \vec{F}_{ij} + \vec{F}_i^{\text{ext}}, \quad (2.1)$$

$$M_i \ddot{\phi}_i = \sum_{j=1}^N L_{ij}, \quad i = 1 \dots N. \quad (2.2)$$

Hierbei ist m_i die Masse des i -ten Teilchens, $\vec{r}_i$ seine Ortskoordinate, ϕ_i seine Winkelkoordinate, $\vec{F}_{ij}$ die vom j -ten auf das i -te Teilchen ausgeübte Kraft und $\vec{F}_i^{\text{ext}}$ die Summe

der externen Kräfte, die auf das i -te Teilchen wirken, z. B. die Gravitationskraft. M_i bezeichnet das Trägheitsmoment des i -ten Teilchens und L_{ij} das vom j -ten auf das i -te Teilchen ausgeübte Drehmoment. Die obere Gleichung beschreibt die Translationsbewegung der Teilchen, die untere die Rotation der Teilchen um ihre Schwerpunkte.

Die Wechselwirkungskräfte zwischen den Teilchen können dabei langreichweitige Kräfte wie z. B. van-der-Waals oder elektrische Kräfte sein oder aber kurzreichweitige Kontaktkräfte, wie z. B. elastische und Reibungskräfte. Da in langreichweitigen Kräften jedes Teilchen mit jedem anderen wechselwirkt, steigt die Rechenzeit in diesem Fall mit dem Quadrat der Teilchenanzahl N , es sei denn man führte eine Abschneidedistanz ein, über die hinaus man Wechselwirkungen vernachlässigt, oder man berücksichtigte die Wechselwirkung mit weiter entfernten Teilchen durch Multipol-Momente. Dann würde man Teilchenwechselwirkungen nicht mehr berücksichtigen, wenn die Entfernung der betrachteten Teilchen größer ist als diese Abschneidedistanz. Man würde dabei einen kleinen Fehler in Kauf nehmen. In der vorliegenden Arbeit werden nur Systeme mit kurzreichweitigen Kontaktkräften betrachtet. Diese umfassen elastische Abstoßungskräfte, Haft- und Gleitreibungskräfte, sowie eine Kraft, die Energiedissipation bewirkt, wenn Teilchen kollidieren. Wir werden diese Kräfte im weiteren Verlauf detailliert beschreiben.

Um Differentialgleichungen wie unsere Bewegungsgleichungen numerisch zu lösen, werden im allgemeinen finite Differenzmethoden herangezogen. Dabei wird die kontinuierliche physikalische Zeit in Zeitintervalle δt diskretisiert und die physikalischen Observablen werden zu diskreten Zeitpunkten t_1, t_2 , usw. berechnet. Die Gleichungen werden dabei Schritt für Schritt gelöst, indem man ausgehend von den physikalischen Größen zum Zeitschritt t_n die Werte zum Zeitschritt t_{n+1} erhält. Je kleiner dabei der Zeitschritt δt ist, um so genauer werden die berechneten Werte sein. Im allgemeinen wird es von bestimmten physikalischen Parametern abhängen, wie groß man δt wählen darf. Erfahrungsgemäß ist der Zeitschritt hinreichend klein gewählt, wenn die kleinste auftretende physikalische Zeitskala durch mindestens etwa 30 Zeitschritte diskretisiert wird. In Molekular-Dynamik-Simulationen ist diese Zeitskala durch die Dauer eines Stoßprozesses gegeben. Neben der Molekular-Dynamik-Methode gibt es andere Verfahren, wie z. B. das ereignisgesteuerte Verfahren (event driven method), bei der die Zeit nicht in gleichmäßige Intervalle unterteilt wird, sondern überprüft wird, zu welchen Zeitpunkten Kollisionen auftreten. Zu diesen Zeiten werden dann die physikalischen Größen neu berechnet. Wenn man den Zeitschritt in Molekular-Dynamik-Simulationen nicht genügend klein wählt, liefern diese physikalisch falsche Werte. Die Energiedissipation ist z. B. nicht groß genug [104, 105, 106, 107, 108, 156]. Es gibt viele verschiedene Algorithmen, die es erlauben, mit Hilfe eines finiten Differenzverfahrens die Bewegungsgleichungen für N Teilchen zu lösen, unter anderem das Verlet Verfahren, das Verfahren nach Berendsen-Gunsteren und das Predictor-Korrektor Verfahren. Beim Verlet-Verfahren handelt es sich um einen Algorithmus zweiter Ordnung, bei dem die Geschwindigkeiten nicht zur Berechnung der neuen Positionen benötigt werden. Viele der möglichen Verfahren sind einander äquivalent, da man sie durch Matrix-Operationen ineinander überführen kann [3]. Das Predictor-Korrektor Verfahren ist dem Verlet-Algorithmus überlegen, insbesondere im Bereich relativ kleiner Zeitschritte [3]. Deswegen verwenden wir in all unseren

Simulationsrechnungen dieses Verfahren, das wir im folgenden genauer beschreiben werden.

Das Predictor-Korrektor Verfahren besteht aus drei Phasen, 1.) dem Predictor-Schritt, in dem neue Werte für die physikalischen Größen vorausgesagt werden, 2.) der Kraftberechnung, wo mit diesen Werte die neuen Kräfte berechnet werden und 3.) dem Korrektor-Schritt, in dem diese Kräfte dazu benutzt werden, die physikalischen Variablen zu korrigieren.

In der ersten Phase berechnet man mit einer Taylor-Entwicklung eine Abschätzung für die Positionen (und Winkel) der Teilchen zum Zeitschritt $(t + \delta t)$ aus den bekannten Werten zum Zeitschritt t :

$$r^P(t + \delta t) = r(t) + \delta t v(t) + \frac{1}{2} \delta t^2 a(t) + \frac{1}{6} \delta t^3 b(t) + \dots \quad (2.3)$$

$$v^P(t + \delta t) = v(t) + \delta t a(t) + \frac{1}{2} \delta t^2 b(t) + \dots \quad (2.4)$$

$$a^P(t + \delta t) = a(t) + \delta t b(t) + \dots \quad (2.5)$$

$$b^P(t + \delta t) = b(t) + \dots \quad (2.6)$$

Hierbei steht der Index P für die Predictor-, also die vorhergesagten Werte, die im zweiten Schritt dann "korrigiert" werden müssen. Es gibt Predictor-Korrektor Verfahren unterschiedlicher Ordnung, je nachdem wieviele Ableitungen der Koordinaten mitberücksichtigt werden. Zunächst wächst bei festgehaltener Größe des Zeitschrittes die Präzision der Berechnung mit der Ordnung des Verfahrens. Bei höheren Ordnungen kann man, falls dies gewünscht wird, eine weitere Steigerung der Präzision dadurch erhalten, daß man während eines Zeitschrittes mehrere Iterationen des Verfahrens durchführt. Wir verwenden in unseren Simulationen das Predictor-Korrektor Verfahren fünfter Ordnung, d. h. wir berücksichtigen alle Ableitungen der Koordinaten bis zur fünften Ordnung.

Bisher ist die Bewegungsgleichung noch gar nicht in den Integrationsalgorithmus eingegangen. Dies geschieht im Korrektor-Schritt, in dem die Koordinaten und ihre Ableitungen durch folgenden Term "korrigiert" werden. Dazu definieren wir zunächst:

$$\delta a(t + \delta t) = a^C(t + \delta t) - a^P(t + \delta t) \quad (2.7)$$

Dabei ist $a^C(t + \delta t)$ die korrekte Beschleunigung zum Zeitschritt $(t + \delta t)$, die man aus den Kräften zwischen den Teilchen erhält. $\delta a(t + \delta t)$ ist also ein Maß für den Fehler, den man im Predictor-Schritt gemacht hat, und wird im Korrektor-Schritt dazu benutzt, diesen zu korrigieren. Die korrigierten Werte $r^C(t + \delta t)$, $v^C(t + \delta t)$, usw. sind dann wesentlich bessere Näherungswerte für die physikalischen Größen als die entsprechenden Werte nach dem Predictor-Schritt:

$$r^C(t + \delta t) = r^P(t + \delta t) + c_0 \delta a(t + \delta t) \quad (2.8)$$

$$v^C(t + \delta t) = v^P(t + \delta t) + c_1 \delta a(t + \delta t) \quad (2.9)$$

$$a^C(t + \delta t) = a^P(t + \delta t) + c_2 \delta a(t + \delta t) \quad (2.10)$$

$$b^C(t + \delta t) = b^P(t + \delta t) + c_3 \delta a(t + \delta t) \quad (2.11)$$

Für eine Bewegungsgleichung der Form

$$\ddot{r} = f(r, \dot{r}), \quad (2.12)$$

mit einem geschwindigkeitsabhängigen Dissipationsterm und das von uns verwendete Verfahren fünfter Ordnung lauten die Korrektur-Koeffizienten [3, 58].

c_0	c_1	c_2	c_3	c_4	c_5
3/16	251/360	1	11/18	1/6	1/60

Die Koeffizienten hängen sowohl von der Ordnung des Verfahrens als auch von der Art der Differentialgleichung ab, also z. B. davon, ob diese geschwindigkeitsabhängig ist oder nicht. Sie müssen so gewählt werden, daß das Verfahren optimale numerische Stabilität und Genauigkeit der Bahnkurven garantiert, vgl. dazu [58, 59].

2.2 Kraftberechnung

Unsere Teilchen sind beliebig geformte, konvexe Polygone, die elastisch miteinander wechselwirken. Für solche Teilchen gibt es keine einfachen Voraussagen, wie die elastische Kraft von der Verformung der Teilchen abhängt. Um diese zu erhalten, müßte man eine Berechnung mit der Methode der finiten Elemente durchführen, in der jedes Teilchen aus vielen kleinen, deformierbaren Dreiecken aufgebaut ist [189]. Mit einer solchen Methode könnte man zwar das Verhalten von zwei Teilchen bei einem Stoß mit großer Genauigkeit untersuchen, wir sind jedoch am Verhalten vieler Teilchen untereinander interessiert, wobei es auf die exakte Simulation der einzelnen Stöße nicht so sehr ankommt, da wir erwarten, daß das Gesamtverhalten des Systems nicht sehr davon abhängt, wie die einzelnen Stöße im Detail ablaufen. Deswegen machen wir eine einfache Annahme über das Wechselwirkungsverhalten zwischen zwei Stößen, das physikalisch plausibel ist. Darüber hinaus zeigt sich, daß sie effizient programmiert werden kann.

In unserem Algorithmus werden die Teilchen als nicht deformierbar betrachtet, d. h. sie behalten immer die gleiche, bei der Programminitialisierung festgelegte Form. Um die elastische Kraft zwischen den Teilchen zu berechnen, betrachten wir den geometrischen Überlapp der beiden Polygone. Nach der Poissonschen Vermutung nehmen wir an, daß diese Kraft dann proportional zum Flächeninhalt dieses Überlappungspolygons ist [182, 183, 184, 185]. Man beachte, daß wir dadurch auf natürliche Weise zwei verschiedene Kraftgesetze bekommen. Wenn zwei Seiten aufeinander treffen, erhält man ein lineares Kraftgesetz, da die Überlappungsfläche linear mit der Überlappungsdistanz wächst. Trifft dagegen eine Ecke auf eine Seite, erhält man ein konisches Kraftgesetz, da dann die Überlappungsfläche quadratisch mit der Überlappungsdistanz wächst.

Um diesen zu berechnen, muß man zunächst die Schnittpunkte aller Seiten des Polygons i mit den Seiten des Polygons j berechnen. Normalerweise wird es nur zwei solche Schnittpunkte geben, in pathologischen Fällen können aber auch Situationen auftreten, wo mehr als zwei Schnittpunkte gefunden werden. Dies kommt zwar selten vor, aber

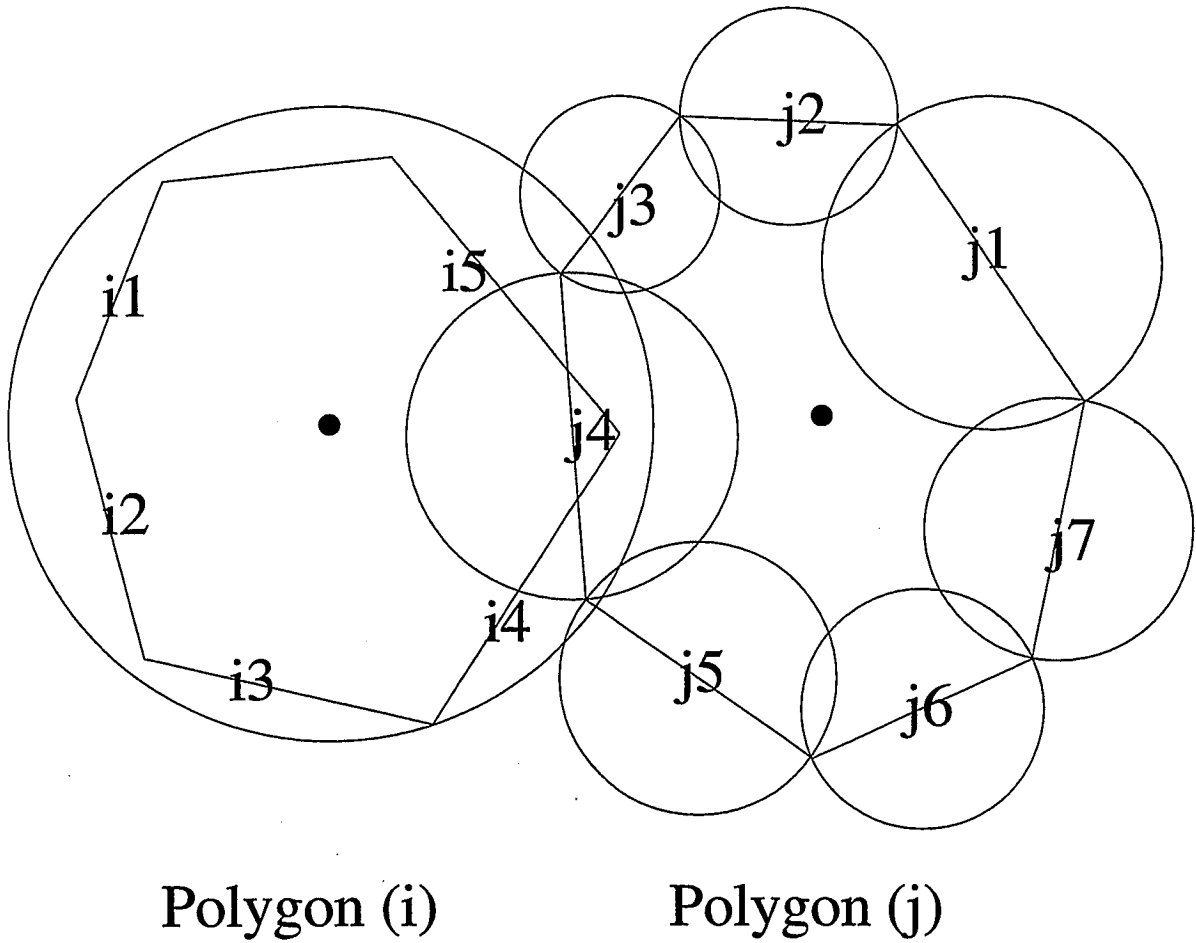


Abbildung 2.1: Bei der Bestimmung der Schnittpunkte kann man Rechenzeit einsparen, indem man überprüft, bei welchen Seitenpaaren ein Schnitt überhaupt möglich ist. Dazu untersucht man den Überlapp zwischen dem Kreis, der das Polygon i völlig enthält und den Kreisen, die die Seiten des Polygons j als Durchmesser enthalten.

trotzdem müssen diese Fälle einer besonderen Behandlung unterzogen werden, wobei aus diesen dann zwei eindeutig bestimmte Punkte ausgewählt werden. Wir werden diesen Reduktionsalgorithmus später genauer beschreiben. Die beiden Schnittpunkte werden zunächst in eine eindeutige Ordnung gebracht. Die beiden Punkte werden mit P_1 und P_2 bezeichnet und zwar so, daß das Vektorprodukt des Verbindungsvektors der Schwerpunkte der beiden Teilchen $C_i C_j$ und des Verbindungsvektors der beiden Schnittpunkte $P_1 P_2$ miteinander positiv ist. Wir benötigen diese Ordnung der Punkte, damit das später zu konstruierende Überlappungspolygon eine eindeutige (bei uns mathematisch positive) Orientierung hat. Das heißt, wir möchten die Ecken des Überlappungspolygons gerne in der Reihenfolge angeordnet haben, die man erhält, wenn man der Umrandung des Überlappungspolygons gegen den Uhrzeigersinn folgt. Die hier beschriebenen Schritte

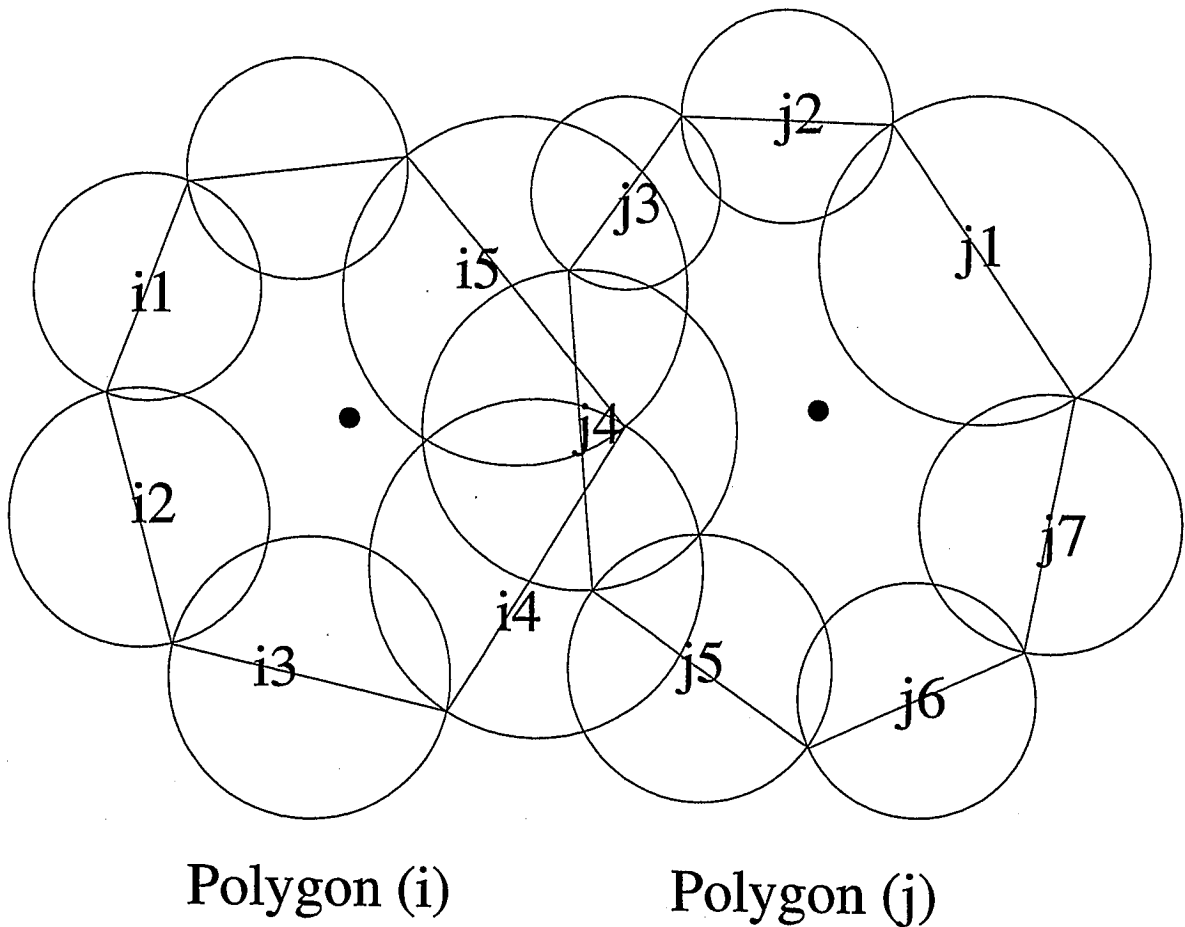


Abbildung 2.2: Eine alternative Methode zur Einsparung von Rechenzeit besteht darin, zu untersuchen, wann die Kreise, die die Seiten des Polygons i als Durchmesser enthalten einen Überlapp mit den Kreisen, die die Seiten des Polygons j als Durchmesser enthalten, haben.

werden im Simulationsprogramm in der Routine **overlap** ausgeführt. Zunächst müssen also die Schnittpunkte aller Seiten des Polygon i mit allen Seiten des Polygons j berechnet werden. Da dies der Teil des Algorithmus ist, der die meiste Rechenzeit benötigt, ist jede Einsparung an dieser Stelle sehr vorteilhaft. Dies ist in der Tat möglich, wie man in Abb. 2.1 erkennt. Man überprüft, ob der Kreis, der das Polygon i komplett enthält, mit den Kreisen überlappt, die die Seiten des Polygons j als Durchmesser enthalten. Gibt es keinen Überlapp, so kann es keinen Schnittpunkt zwischen einer Seite des Polygons i und der überprüften Seite des Polygons j geben und diese Kombination muß im weiteren Verlauf nicht mehr überprüft werden. Indem man diese Vorgehensweise symmetrisch durchführt, kann man die Anzahl der noch zu überprüfenden Seitenpaare sehr stark einschränken. Man beachte, daß diese Methode der Reduzierung besser geeignet ist, als die in Abb. 2.2 dargestellte. Dies hat zwei Ursachen: Zum einen ist der Rechen-

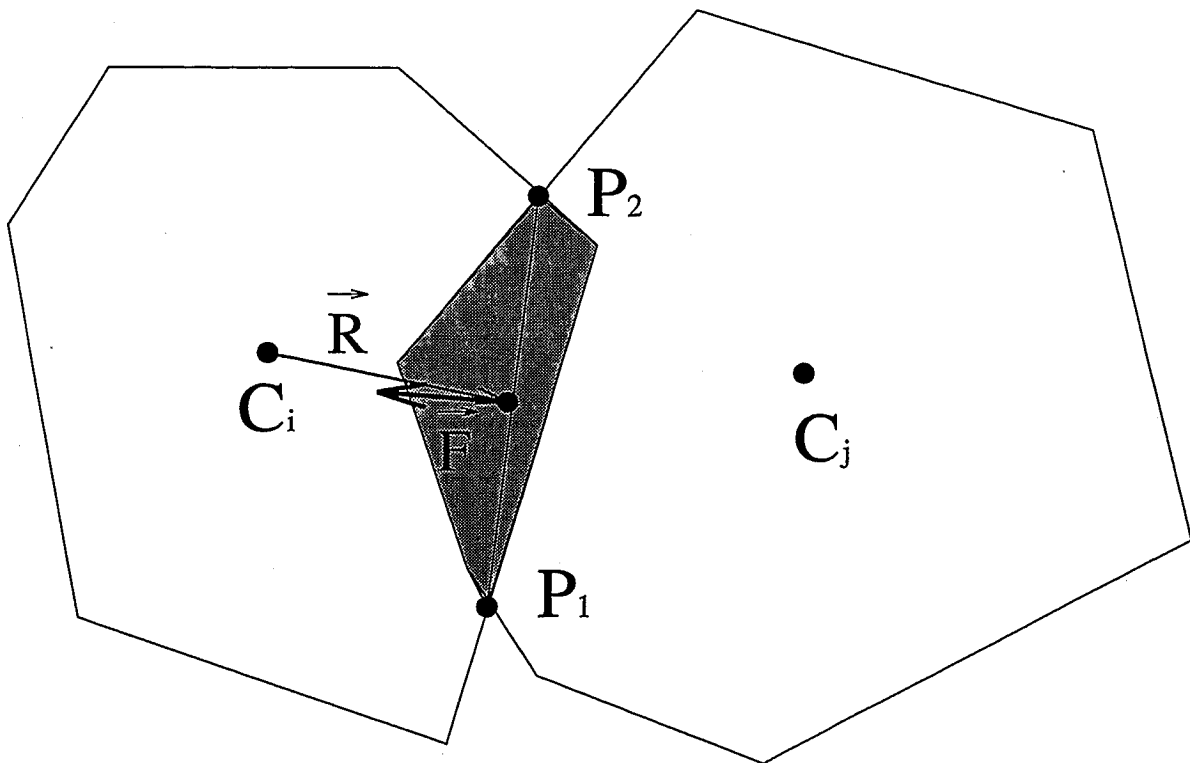


Abbildung 2.3: Zur Bestimmung der elastischen Kraft zwischen zwei Teilchen wird das geometrische Überlappungspolygon herangezogen. C_1 und C_2 sind die Schwerpunkte der beiden Polygone, P_1 und P_2 die beiden Schnittpunkte und $\vec{F}$ ist der Kraftvektor.

zeitaufwand für die in Abb. 2.1 illustrierte Methode proportional zu $(N + M)$, wobei N und M die Anzahl der Seiten der beiden Polygone sind, während der Aufwand der in Abb. 2.2 dargestellten Vorgehensweise proportional zu $(N \cdot M)$ ist. Der zweite Grund, der für die von uns verwendete Methode spricht, liegt darin, daß die kleineren Kreise, die die Seiten als Durchmesser enthalten, stärker gekrümmt sind und weiter in das andere Polygon hineinragen als die großen Kreise, die das Polygon vollständig enthalten. Dadurch werden mehr Paare gefunden, die keinen Schnittpunkt miteinander bilden, und der zweite Algorithmus schränkt die Anzahl der potentiellen Seitenpaare nicht so stark ein wie der erste.

Im weiteren Verlauf wird für jedes der übriggebliebenen Seitenpaare der Schnittpunkt berechnet. Jeder Punkt S_i , der auf einer Seite des Polygons i liegt, gehorcht der folgenden Geradengleichung (entsprechendes gilt für das Polygon j):

$$S_i = E_i + \alpha_i \vec{v}_i, \quad (2.13)$$

$$S_j = E_j + \alpha_j \vec{v}_j. \quad (2.14)$$

Hierbei sind E_i und E_j Eckpunkte der Polygone i und j , $\vec{v}_i$ ist der Vektor, der den Eckpunkt E_i mit seinem im mathematisch positiven Sinne benachbarten Eckpunkt F_i

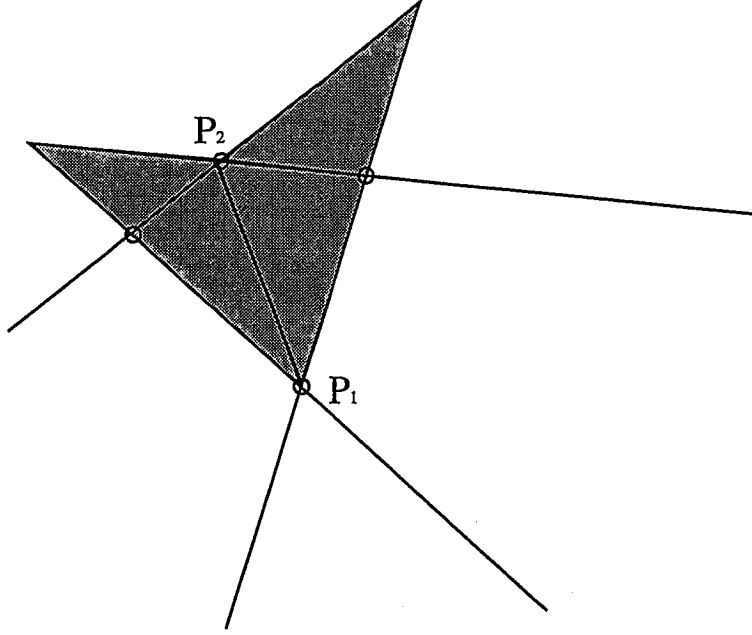


Abbildung 2.4: In pathologischen Situationen, wie der hier gezeigten, kann es vorkommen, daß die beiden Polygone mehr als zwei Schnittpunkte miteinander haben. Dann müssen aus den Schnittpunkten zwei ausgewählt werden. Unsere Auswahlregel besteht darin, daß wir das Punktpaar wählen, für das die Länge der Verbindungslinie maximal ist.

des Polygons i verbindet. Entsprechendes gilt für den Vektor $\vec{v}_j$ und das Polygon j . Die Parameter α_i und α_j liegen zwischen 0 und 1, wobei 0 den Eckpunkten E_i und E_j entspricht und 1 die Eckpunkte F_i und F_j kennzeichnet. Um den Schnittpunkt der beiden Seiten zu berechnen, müssen die beiden Gleichungen gleichgesetzt werden. In Koordinatenform erhält man zwei Gleichungen mit den beiden Unbekannten α_i und α_j :

$$E_{i,x} + \alpha_i v_{i,x} = E_{j,x} + \alpha_j v_{j,x}, \quad (2.15)$$

$$E_{i,y} + \alpha_i v_{i,y} = E_{j,y} + \alpha_j v_{j,y}. \quad (2.16)$$

Auflösen nach α_i und α_j ergibt:

$$\alpha_i = \frac{1}{\Delta} (v_{j,x} d_y - v_{j,y} d_x), \quad (2.17)$$

$$\alpha_j = \frac{1}{\Delta} (v_{i,x} d_y - v_{i,y} d_x). \quad (2.18)$$

Hierbei ist Δ die durch

$$\Delta = \begin{vmatrix} v_{i,x} & v_{j,x} \\ v_{i,y} & v_{j,y} \end{vmatrix} \quad (2.19)$$

gegebene Determinante und d_x und d_y sind die durch folgende Gleichungen bestimmten Differenzen:

$$d_x = E_{i,x} - E_{j,x}, \quad (2.20)$$

$$d_y = E_{i,y} - E_{j,y}. \quad (2.21)$$

Für jeden gefundenen Schnittpunkt wird auch die Information abgespeichert, auf welcher Seite des Polygons i und auf welcher Seite des Polygons j er sich befindet. Nachdem alle Schnittpunkte berechnet worden sind, wird zunächst festgestellt, wieviele Schnittpunkte gefunden worden sind. Ist die Zahl dieser Punkte 0 (kein Kontakt) oder 1 (Berührungspunkt), so ist die Fläche des Überlappungspolygons und somit die elastische Kraft 0. In diesem Falle ist eine weitere Behandlung nicht nötig und die Unteroutine **overlap** kann verlassen werden.

Ist die Anzahl der Schnittpunkte größer als 2, müssen aus diesen zunächst zwei Punkte eindeutig bestimmt werden. Würde man einfach willkürlich zwei Punkte auswählen, könnte es passieren, daß von einem Zeitschritt auf den nächsten ein anderes Paar von Schnittpunkten gefunden würde, was zu sehr großen Schwankungen in der berechneten Kraft führen würde. Solche pathologischen Fälle mit mehr als 2 Schnittpunkten treten insbesondere dann auf, wenn Ecken mit spitzen Winkeln aufeinander treffen und sich gegenseitig durchdringen. Eine einfache Auswahlregel, die wir in unserem Algorithmus anwenden, besteht darin für die weiteren Berechnungen das Punktepaar auszuwählen, für den der Abstand der beiden Punkte am größten ist. Damit sind die beiden Schnittpunkte eindeutig identifiziert.

Um die Ecken des Überlappungspolygons in mathematisch positiver Orientierung zu bestimmen, beginnt man beim Schnittpunkt P_1 und stellt zunächst fest, auf welcher Seite des Polygons i er sich befindet. (Diese Information haben wir ja gespeichert, als der Schnittpunkt gefunden worden ist.) Dann wird festgestellt, auf welcher Seite des Polygons i sich der zweite Schnittpunkt P_2 befindet. Alle Ecken des Polygons i die sich zwischen den beiden Schnittpunkten befinden, werden zu Ecken des Überlappungspolygons. Die nächste Ecke ist dann der zweite Schnittpunkt P_2 .

Nun wird die gleiche Prozedur für das Polygon j durchgeführt: Zunächst wird überprüft, auf welchen Seiten von j sich die beiden Schnittpunkte befinden. Alle Ecken des Polygons j , die sich zwischen dem zweiten Schnittpunkt P_2 und dem ersten Schnittpunkt P_1 befinden werden zu Ecken des Überlappungspolygons. Die letzte Ecke, die mit der ersten Ecke übereinstimmt, ist dann wieder P_1 . Damit ist das Überlappungspolygon bestimmt und es ist sichergestellt, daß seine Eckpunkte in mathematisch positiver Richtung orientiert sind.

Nun kann die Fläche des Überlappungspolygons leicht bestimmt werden. Dazu wird dieses zunächst in Dreiecke unterteilt. Die Fläche jedes dieser Dreiecke ist durch

$$A_{\Delta} = \frac{1}{2} \begin{vmatrix} a_x & b_x \\ a_y & b_y \end{vmatrix} \quad (2.22)$$

gegeben. Hierbei sind a_x , a_y , b_x und b_y die x- und y-Koordinaten von zwei Seitenvektoren des Dreiecks. Aufsummieren aller dieser Teilflächen ergibt die Gesamtfläche des

Polygons. Die elastische Kraft zwischen den beiden Teilchen ist dann gegeben durch:

$$\vec{F}_{el} = \frac{Y \cdot A d}{L_C} \vec{n}. \quad (2.23)$$

Hierbei ist A die Fläche des Überlappungspolygons, Y das Elastizitätsmodul, d die Dicke der Teilchen (in der dritten Dimension) und L_C eine charakteristische Länge. Die charakteristische Länge L_C entspricht der Länge einer Feder im nicht belasteten Zustand. Sie sollte eine typische Größenordnung widerspiegeln und ihre Wahl unterliegt einer gewissen Willkür. Wir definieren L_C als den Durchmesser eines Kreises, der die gleiche Fläche hat, wie das betrachtete Polygon.

Als Angriffspunkt für die Kraft wird der Mittelpunkt der Verbindungslinie der beiden Schnittpunkte P_1 und P_2 gewählt. Dieser wird für die Berechnung der beiden Drehmomente M_1 und M_2 benötigt, die auf die beiden Teilchen einwirken. Um diese zu bestimmen benötigt man die Verbindungsvektoren $\vec{R}_1$ und $\vec{R}_2$ von den Schwerpunkten der beiden Teilchen zum Kraftpunkt. Dann sind die Drehmomente gegeben durch:

$$\vec{M}_1 = \vec{R}_1 \times \vec{F}_{el}, \quad (2.24)$$

$$\vec{M}_2 = \vec{R}_2 \times \vec{F}_{el}. \quad (2.25)$$

Außer der elastischen Wechselwirkungskraft berücksichtigen wir noch eine Kraft, die Energiedissipation verursacht und eine Reibungskraft. Um zu erreichen, daß bei jedem Stoß Energie dissipiert wird, ergänzen wir die Kraftgleichung durch folgende Terme:

$$F_{ij}^N = \frac{Y \cdot A}{L_C} - m_{ij}^{eff} \cdot \gamma_N \cdot v_{rel}^N, \quad (2.26)$$

$$F_{ij}^T = -\min(m_{ij}^{eff} \gamma_T |v_{rel}^T|, \mu |F_{ij}^N|). \quad (2.27)$$

Hierbei ist γ_N ein Dämpfungsparameter, der kontrolliert, wieviel Energie bei einem Zweiteilchenstoß dissipiert wird, v_{rel}^N und v_{rel}^T stellen den Normal-, bzw. Tangentialanteil der Relativgeschwindigkeit zwischen den beiden Teilchen dar und μ ist der Coulomb'sche Reibungskoeffizient. m_{ij}^{eff} ist die wie üblich definierte effektive Masse der beiden Teilchen:

$$m_{ij}^{eff} = \frac{m_i \cdot m_j}{m_i + m_j}. \quad (2.28)$$

Mit unserem Ansatz ist es möglich, den experimentell zugänglichen Restitutionskoeffizienten ϵ zu simulieren. Dieser gibt das Verhältnis der Summe der kinetischen Energien der Teilchen nach einem Stoßprozeß zur Summe vor dem Stoß an. Mithin liegt ϵ immer zwischen 0 und 1, wobei ein Wert von 1 einen völlig elastischen und 0 einen vollkommen unelastischen Stoß bedeutet. Der Zusammenhang zwischen γ_N und ϵ ist für ein lineares Kraftgesetz der folgende:

$$\epsilon = \exp\left(\frac{-\pi \gamma_N}{\sqrt{\omega_0^2 - \gamma_N^2}}\right). \quad (2.29)$$

Hierbei ist ω_0 die elastische Eigenfrequenz eines harmonischen Oszillators mit der Federkonstante k und der effektiven Masse m_{eff} :

$$\omega_0 = \sqrt{\frac{k}{m_{eff}}}. \quad (2.30)$$

Zwischen dem Elastizitätsmodul Y , der Federkonstante k und der charakteristischen Länge L_C besteht für unsere Teilchen der folgende Zusammenhang:

$$k = \frac{Y}{L_C} A_{kont}. \quad (2.31)$$

Dabei ist A_{kont} die Kontaktfläche bei der Wechselwirkung zweier Teilchen. Sie ist gleich der Dicke des Systems in der dritten Dimension multipliziert mit der Länge der Kontaktlinie (Verbindung zwischen P_1 und P_2). Wir sind uns dabei wohl bewußt, daß die Formel 2.29 in unserem Falle nur eine grobe Abschätzung des Restitutionskoeffizienten erlaubt, denn für ein konisches Kraftgesetz, das bei uns immer dann auftritt, wenn eine Ecke auf eine Seite trifft, ist der obige Zusammenhang zwischen ϵ und γ_N nicht korrekt. Mit unserem Ansatz für die tangentielle Kraft F_T wollen wir eine Coulomb-artige Reibungskraft simulieren, d. h. eine Reibungskraft, die unabhängig vom Betrag der Relativgeschwindigkeit ist und proportional zur Normalkraft F_N mit dem Proportionalitätsfaktor μ . Die Richtung dieser Kraft ist immer entgegengesetzt zum Vektor der Relativgeschwindigkeit gerichtet. Dies führt jedoch zu numerischen Problemen, da selbst bei sehr kleinen Relativgeschwindigkeiten schon eine große Reibungskraft wirkt und eine Sprungstelle bei $v_{rel} = 0$ existiert, die zu numerischen Instabilitäten führt. Um dies zu umgehen, lassen wir die Reibungskraft linear mit der Relativgeschwindigkeit ansteigen, bis sie den Wert $\mu \cdot F_N$ erreicht. Bei höheren Relativgeschwindigkeiten wird dann dieser Wert für die Reibungskraft F_T verwendet. Als Proportionalitätsfaktor für diesen linearen Teil des tangentialen Kraftgesetzes wirkt in der Simulation der Parameter γ_T , der selbst keine physikalische Bedeutung hat.

2.3 Bestimmung der Nachbarn

Ein wichtiges Problem bei Molekular-Dynamik Simulation ist die effiziente Bestimmung der Nachbarn, mit denen ein Teilchen wechselwirkt. Der zunächst naheliegende Zugang, die Kraft zwischen allen Teilchenpaaren zu bestimmen, kommt nicht in Frage, da die Rechenzeit dann mit N^2 anwächst, wobei N die Anzahl der Teilchen ist. Ein möglicher Ausweg, besteht darin, sogenannte Verlet-Listen zu benutzen. Diese enthalten die Nachbarn einer Zelle und werden nur nach einer gewissen Anzahl an Zeitschritten, wenn sich die Teilchen um eine kleine Wegstrecke bewegt haben, neu aufgebaut. Durch geschickte Vorgehensweise kann man es hier erreichen, daß der N^2 -Term erst bei sehr großen Systemen eine Rolle spielt.

Wir bevorzugen jedoch einen anderen Weg, der als Linked Cell Algorithmus bekannt ist. Dazu wird das Gebiet, in dem sich die Teilchen bewegen, mit einem zeitlich

unveränderlichen Gitter überdeckt. Wir verwenden ein Quadratgitter, dessen Maschen ungefähr so groß sind wie der mittlere Teilchendurchmesser, nämlich eine Längeneinheit. Für jedes Teilchen wird dann bestimmt, in welcher Zelle dieses Gitters sich das Teilchen befindet. In unserem Einheitsgitter ist dies durch den Fortran-Operator `int()` sehr effizient möglich. Die möglichen Nachbarn eines Teilchens liegen in den Zellen des Quadratgitters, die mit der Zelle, in der sich das Teilchen befindet, benachbart sind. Man muß also nur in der Nachbarschaft der Quadratzelle, in der sich das Teilchen befindet, nach Nachbarn suchen. Dieser Algorithmus hat den Vorteil, daß seine Rechenzeit linear mit der Teilchenzahl N wächst. Die Größe der Nachbarschaft, in der sich Teilchen befinden können, hängt von der Größenverteilung der Teilchen ab, in unserem Fall können sich Nachbarn in Zellen mit einem Abstand von drei Gittereinheiten von der ursprünglichen Zelle befinden, so daß man ein Gebiet von $(3 + 1 + 3)^2 = 49$ Zellen auf Nachbarn zu durchsuchen hat.

2.4 Verbinden der Zellen mit Stäben

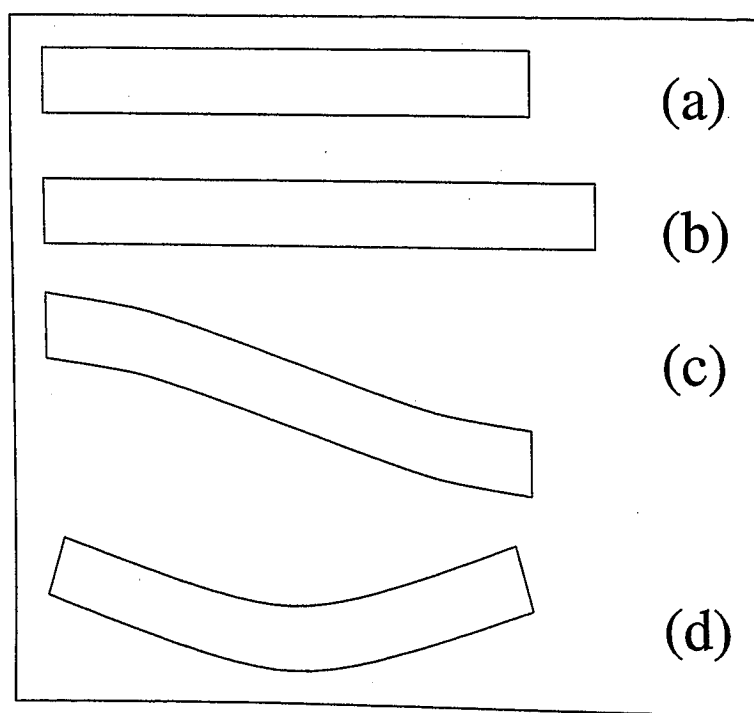


Abbildung 2.5: Die Zellen können durch Stäbe miteinander verbunden werden (a). Diese haben drei Freiheitsgrade: (b) Dehnung oder Stauchung, (c) Scherung und (d) Biegung.

Durch eine Erweiterung des Modells wird es möglich, noch allgemeinere Probleme zu behandeln, wie z. B. solche der Bruchmechanik und dem Sintern von Materialien. Dazu machen wir Gebrauch von Objekten, die in der Elastizitätstheorie als Stäbe bezeichnet

werden. In Kapitel 7 und 8.1 werden wir auf unterschiedliche Art und Weise von dieser Erweiterung Gebrauch machen.

Bei den Stäben handelt es sich um elastische Objekte, die in zwei Dimensionen drei Freiheitsgrade haben: Sie können in longitudinaler Richtung gedehnt oder gestaucht und in transversaler Richtung geschernt werden. Darüberhinaus können sie auch verbogen werden, s. Abb. 2.5. Die rückstellenden elastischen Kräfte hängen in folgender Weise von den Auslenkungen ab, s. z. B. [69, 167]:

$$F_{S,N} = \frac{1}{a}(x_i - x_j) \quad (2.32)$$

$$F_{S,T} = \frac{1}{b + \frac{1}{12}c} \left[y_i - y_j + \frac{1}{2}(z_j - z_i) \right] \quad (2.33)$$

$$M_{S,i} = \frac{b}{c(b + \frac{1}{12}c)}(z_i - z_j) \quad (2.34)$$

$$\frac{1}{2(b + \frac{1}{12}c)} \left[y_i - y_j + \frac{2}{3}z_i + \frac{z_j}{3} \right] \quad (2.35)$$

Hierbei steht $F_{S,N}$ für die vom Stab ausgeübte Kraft in longitudinaler Richtung, $F_{S,T}$ für die tangentielle Komponente und $M_{S,i}$ ist das auf das eine Ende i des Stabes ausgeübte Drehmoment, wobei im Ausdruck für $M_{S,j}$ alle i mit j vertauscht werden müssen. Der Ausdruck $(x_i - x_j)$ stellt die longitudinale Verformung des Stabes dar, entsprechend steht $(y_i - y_j)$ für die Verformung in Scherrichtung. Für z_i gilt: $z_i = l\theta_i$, wobei l die Stablänge im nichtbelasteten Zustand und θ_i die Verbiegung an einem seiner freien Enden ist. Die Parameter a , b und c stellen den Zusammenhang mit den Materialeigenschaften her:

$$a = \frac{l}{YA_S}, \quad b = \frac{l}{GA_S} \quad \text{und} \quad c = \frac{l^3}{YI}. \quad (2.36)$$

Dabei ist A_S die Querschnittsfläche des Stabes, I sein Flächenträgheitsmoment und G sein Schermodul.

Die Stäbe werden an den Schwerpunkten der Teilchen rigide befestigt, so daß eine Relativbewegung zu einer Deformation des Stabes und damit zu einer rückstellenden Kraft führt, die auf die Teilchen übertragen wird. Eine Drehung der Teilchen bewirkt eine Verbiegung der Stäbe. Da bereits die elastische Kraft, die durch das Überlappungspolygon erzeugt wird, dafür sorgt, daß eine rückstellende Kraft existiert, wenn die Polygone sich einander nähern, wird die durch die Normalkomponente des Stabes hervorgerufene Kraft nur dann berücksichtigt, wenn sich die Polygone in normaler Richtung voneinander entfernen, da andererseits die Rückstellkraft doppelt berechnet würde, wenn sich Teilchen in normaler Richtung einander nähern.

Übersteigt die Deformation eines Stabes gewisse Schwellwerte, so bricht er und wird entfernt. Damit ist es möglich, Bruchvorgänge zu simulieren: Man betrachtet einen Block aus durch Stäben miteinander verbundenen Voronoizellen. Die Grenzen der Zellen legen dabei die potentiellen Bruchstellen im Vorhinein fest. Ein gebrochener Stab kennzeichnet dann in der Simulation die kleinste darstellbare Einheit eines Risses. Für unsere

Bruchsimulationen in Kapitel 8.1 haben wir das Bruchkriterium von Tresca und Mises verwendet. Dabei wird angenommen, daß ein Stab in erster Ordnung dann bricht, wenn seine Elongation oder seine Verbiegung zu groß werden. Es ist aber ohne großen Aufwand möglich, beliebige andere Kriterien zu verwenden. Bei Verwendung des Tresca-Mises Kriterium bricht ein Stab, wenn folgende Ungleichung erfüllt ist:

$$\left[\frac{F_{S,N}}{t_F} \right]^2 + \frac{\max(|M_i|, |M_j|)}{t_M} \geq 1. \quad (2.37)$$

Dabei ist $F_{S,N}$ die Kraft in Normalrichtung des Stabes und M_i und M_j sind die auf seine Enden wirkenden Drehmomente; t_F und t_M sind die frei wählbaren Bruchgrenzen. Sie sind ein Maß dafür, wie weit man einen Stab elongieren bzw. komprimieren kann, bzw. um welchen Winkel man ihn verbiegen darf, ohne daß er bricht.

Das Simulationsprogramm erlaubt es darüber hinaus, neue Stäbe zwischen zwei Zellen einzufügen. Damit wird es möglich das Sintern von Granulat zu einem Festkörper zu simulieren. Ein anderes Beispiel für die Anwendung einer solchen Simulation auf geophysikalischer Zeitskala ist das Heilen von Felsmaterial, welches dann stattfindet, wenn Felsgestein für eine Zeit von der Größenordnung von hunderttausend Jahren unter sehr hohem Druck zusammengepreßt wird. Wir haben noch keine Simulationen in diesem Zusammenhang durchgeführt. Das Programm ist dazu aber bereits vorbereitet.

Dazu verwenden wir ein einfaches Modell mit zwei Parametern: Ein neuer Stab wird eingesetzt, wenn der Druck, den zwei Teilchen aufeinander ausüben für eine Zeitspanne T_{sint} einen Schwellwert P_{sint} überschreitet. Wenn der Druck vor Ablauf der Zeit T_{sint} wieder unter P_{sint} fällt, wird diese Zeit auf 0 gesetzt und sollte er anschließend die Schwelle P_{sint} wieder übersteigen, beginnt man mit der Zeitmessung erneut bei 0.

In unserem Algorithmus gibt es, um diese Aufgabe zu bewältigen, virtuelle Stäbe, die initialisiert werden, sobald der Druck zwischen zwei Zellen zum ersten Mal den Schwelldruck P_{sint} überschreitet. Für diese virtuellen Stäbe wird lediglich die Information gespeichert, mit welchen beiden Zellen er verbunden ist und wieviel Zeit seit seiner Initialisierung vergangen ist. Sollte der Druck vor Ablauf der Zeit t_{sint} den Schwellwert P_{sint} unterschreiten, so wird er wieder gelöscht. Ist der Druck jedoch nach Ablauf der Zeit t_{sint} noch größer als P_{sint} so wird der virtuelle Stab zu einem realen Stab. Seine elastischen Parameter a , b und c werden auf geometrische Art bestimmt. Diese Parameter sind vollkommen determiniert, wenn man die Querschnittsfläche A_S des Stabes, sein Flächenträgheitsmoment I_S und sein Elastizitätsmodul Y kennt.

Für das Elastizitätsmodul Y nehmen wir den gleichen Wert an wie für das Material, aus dem die Teilchen bestehen. Die beiden Größen A_S und I_S bestimmen wir dann auf geometrische Art, indem wir für die Länge l des Stabes einfach den Abstand der Schwerpunkte der beiden Zellen voneinander annehmen und für die Breite des Stabes die Kontaktlänge (also die Länge der Verbindungslinie zwischen den beiden Schnittpunkten P_1 und P_2 der Polygone miteinander.) Für das Flächenträgheitsmoment eines rechteckigen Stabes gilt:

$$I_S = \frac{1}{12} d \cdot w^3. \quad (2.38)$$

Dabei ist w die Breite des Stabes und d seine Dicke. (Man beachte, daß hier die Stablänge l nicht eingeht.) Die Dicke des Stabes d entspricht der Dicke unseres Systems in der dritten Dimension.

Eine ähnliche Vorgehensweise wenden wir an, um die Stärke von Stäben zu bestimmen, die gleich am Anfang der Simulation eingesetzt werden sollen. Für die Länge des Stabes nehmen wir wieder den Abstand der beiden Massenzentren der zu verbindenden Zellen. Als Breite des Stabes wählen wir die Projektion der Seite, die die beiden Polygone gemeinsam haben auf eine Linie, die senkrecht zur Verbindungslinie der beiden Massenzentren C_i und C_j steht.

Damit sind die wesentlichen Aspekte unseres Programms und der verwendeten Algorithmen dargestellt. Auszüge aus unserem Programm, nämlich die Unterprogramme zur Berechnung des Überlappungspolygons und der Dissipationskräfte **overlap**, die Routinen zur Berechnung der elastischen Stabkräfte **beamforce** und **beamupdate**, sowie das Hauptprogramm finden sich im Anhang A dieser Abhandlung. Hier findet sich auch ein Beispiel für eine Parameterdatei mit den von außen wählbaren Programmparametern.

Kapitel 3

Voronoi Gitter

3.1 Einleitung

Es gibt sehr viele Anwendungen von Zufallsgittern, z. B. in den Materialwissenschaften [57], in der Modellierung der großräumigen Struktur des Universums [39, 78], in kinetischen Wachstumsmodellen [117, 118], der Modellierung von Sandhaufen [135] und in der Quantenfeld-Theorie [37, 53]. Zufallsgitter können dazu benutzt werden, Systeme zu diskretisieren, ohne dabei - im Gegensatz zu regulären Gittern - Anisotropien einzuführen, d. h. Artefakte, die auf die Regularität des Gitters zurückzuführen sind. Darüberhinaus können Zufallsgitter auch benutzt werden, zelluläre Strukturen wie z. B. granulare Mosaike, biologische Gewebe und Schäume zu beschreiben [97, 99, 119, 144, 187]. Es ist also von Bedeutung, über effiziente Algorithmen zur Konstruktion von Zufallsgittern zu verfügen, insbesondere bei solchen Anwendungen, wo das Gitter während einer Computersimulation ständig neu berechnet werden muß.

Ein oft benutztes Zufallsgitter ist das Delaunay-Gitter [62], welches aus einer Triangulation des Raumes basierend auf N Punkten besteht. Das sog. Voronoi-Gitter ist mathematisch dual zum Delaunay-Gitter und beschreibt eine Aufteilung des Raumes in konvexe Zellen. Es wird in der folgenden Weise beschrieben: Für jeden Punkt P bestimme man den Teil des Raumes, welcher P näher ist als allen anderen Punkten. Diese Prozedur, die Voronoi-Zergliederung des Raumes genannt wird, zerteilt den Raum in konvexe Zellen. Es gibt auch andere Möglichkeiten, den Raum in Zellen aufzuteilen, wie z. B. die Laguerre Partition [79] und kürzlich wurde eine verallgemeinerte Voronoi-Konstruktion dazu benutzt, Seifenschäume zu modellieren [119].

Wir werden statische und dynamische Algorithmen vorstellen, die Delaunay- und Voronoi-Gitter generieren, wobei wir uns auf zwei Dimensionen beschränken wollen. Jedoch können die Verfahren ohne großen Aufwand auf beliebige Dimensionen verallgemeinert werden. Wir werden bestimmen, wie schnell die Algorithmen sind und wie die benötigte Rechenzeit mit der Anzahl der Punkte anwächst. Für ein Zufallsgitter, dem N Punkte zugrunde liegen, bezeichnen wir diese Zeit mit $T(N)$. Messungen werden sowohl für sog. Poisson'sche Zufallsgitter (PZG) [73] als auch für vektorisierbare Zufallsgitter

(VZG) [117, 118] vorgenommen. Das PZG erhält man ausgehend von einer Menge von Punkten, die zufällig und unabhängig voneinander in der Ebene verteilt sind, die dann gemäß der Voronoi-Konstruktion ein Zufallsgitter bestimmen. Diese Methode impliziert, daß das PZG völlig frei von Anisotropien ist [37, 53].

Das vektorisierbare Zufallsgitter ist eine modifizierte Form des PZG. Hier wird die Ebene zunächst in quadratische Zellen aufgeteilt. In jede dieser Zellen wird genau ein Punkt plaziert, wobei die Position innerhalb der Zelle zufallsbestimmt ist. Die Motivation für die Einführung des VZG war, ein vektorisierbares Programm schreiben zu können, wenn man eine Gleichung auf dem Gitter lösen will, z. B. die Laplace Gleichung, die das numerische Wachstum in der diffusionsbeschränkten Anlagerung beschreibt [117, 118]. Darüberhinaus besitzt das VZG für alle praktischen Zwecke keine Anisotropie und ist ausgesprochen nützlich, da vektorisierbare Programme die Rechengeschwindigkeit auf geeigneten Rechnern drastisch erhöhen können.

Zunächst werden die Voronoi- und die Delaunay-Konstruktion in Abschnitt 3.2 detailliert beschrieben. In Abschnitt 3.3 werden diverse Algorithmen vorgestellt. Schließlich wird die Leistungsfähigkeit der Algorithmen in Abschnitt 3.4 diskutiert und in Abschnitt 3.5 eine kurze Zusammenfassung gegeben.

3.2 Voronoi- und Delaunay-Gitter

Die Voronoi-Konstruktion ist für eine gegebene Menge von Punkten folgendermaßen definiert: Für jeden Punkt wird eine assoziierte Zelle bestimmt, die aus dem Teil des Raumes besteht, der diesem Punkt näher liegt als allen anderen Punkten der Menge. Zwei Zellen, die eine Seite gemeinsam haben, bezeichnet man als Nachbarn. Je zwei Punkte, die Zellen definieren, die einander benachbart sind, werden durch eine Verbindungslinie miteinander verbunden. Die Gesamtheit aller Verbindungslinien bildet eine Triangulation des Raumes und wird als Delaunay-Gitter bezeichnet. Das Delaunay-Gitter ist im mathematischen Sinne dual zum Voronoi-Gitter, d. h., die Punkte des Delaunay-Gitters (also die ursprünglichen Zufallspunkte) entsprechen den Zellen des Voronoi-Gitters, die Verbindungslinien des Delaunay-Gitters entsprechen den Seiten des Voronoi-Gitters und die Eckpunkte im Voronoi-Gitter den Dreiecken des Delaunay-Gitters.

Die Triangularisierung der Ebene mit N_P Punkten (Zellen), N_L Verbindungslinien (Seiten) und N_T Dreiecken (Eckpunkten) wird topologisch durch die Euler-Gleichung beschrieben:

$$N_P - N_L + N_T = \chi. \quad (3.1)$$

Hierbei ist χ die Euler Charakteristik der topologischen Fläche, auf der die N_P Zufallspunkte gegeben sind. Sie ist 2 für eine Fläche mit den topologischen Eigenschaften einer Kugel und 0 für einen Torus. Weiterhin gilt für eine Triangularisierung:

$$3N_T = 2N_L. \quad (3.2)$$

Denn jede Dreiecksseite gehört zu zwei Dreiecken und somit wird jede Seite in dem Ausdruck $3N_T$ doppelt gezählt.

Eine äquivalente Methode, das Delaunay-Gitter zu konstruieren, ist die folgende: Zeichne für jedes Tripel von Punkten den eindeutig definierten Kreis, der diese Punkte auf seinem Umfang enthält. Gibt es keine weiteren Punkte im Innern des Kreises, so verbinde die drei Punkte mit drei Linien zu einem Delaunay-Dreieck. Die Mittelpunkte dieser Kreise sind dann die Eckpunkte der Voronoi-Zellen [37, 53].

3.3 Algorithmen

Drei verschiedene Algorithmen für die Generierung von zweidimensionalen Delaunay-Zufallsgittern und den assoziierten Voronoi-Gittern werden vorgestellt. Die Programme sind in den Programmiersprachen C und FORTRAN geschrieben und laufen auf Workstations.

Zunächst beschreiben wir einen Algorithmus für das VZG und dann für das PZG. Anschließend wird ein Algorithmus zur Generierung von beiden Gitterarten vorgestellt. Diesen nennen wir dynamisch, weil er es erlaubt, die Zufallsgitter nach einer kleinen Änderung durch lokale Umordnungen der Verbindungslinien wiederherzustellen.

3.3.1 Vektorisierbare Zufallsgitter

Im Falle des vektorisierbaren Zufallsgitters wird der Teil der Ebene, in dem die Punkte verteilt werden, zunächst mit einem quadratischen Gitter überdeckt, wie in Abb. 3.1 dargestellt ist. Die Seitenlänge einer solchen Zelle ist eine Längeneinheit. Dann wird genau ein Punkt gemäß einer gleichmäßigen Zufallsverteilung in jede dieser Zellen gesetzt.

Jeder Punkt kann nur mit Punkten verbunden sein, die sich in einer von 36 benachbarten Zellen befinden wie in Abb. 3.1(a) dargestellt ist. In Abb. 3.1(b) werden die Wahrscheinlichkeiten tabelliert, mit denen der Punkt im Zentrum mit den Punkten aus dieser benachbarten Zelle verbunden ist [73]. Gehört also der zentrale Punkt zu einem Delaunay-Dreieck, so sind die anderen beiden Punkte dieses Dreiecks in den 36 Nachbarzellen enthalten. Um also alle Punkte eines Dreiecks zu finden, genügt es, in diesem lokal begrenzten Gebiet zu suchen. Eine obere Grenze für den Abstand zweier benachbarter Punkte ist $\sqrt{4^2 + 2^2} = \sqrt{20} \approx 4.472$, wie man aus der Abb. 3.1(a) entnehmen kann [37, 53, 135].

Der VZG Algorithmus

Nachdem die Zufallspunkte gesetzt sind, bestimmt der Algorithmus das erste Dreieck, s. Abb. 3.1(a). Dazu wird ein Punkt A ausgewählt. Für jedes Punktepaaar aus der Nachbarschaft von A wird untersucht, ob der durch diese drei Punkte gegebene Kreis weitere Punkte enthält. Das erste Triplett von Punkten, für das der so definierte Kreis keine weiteren Punkte enthält, wird mit drei Linien zu einem Dreieck verbunden. Diese Verbindungslinien bekommen eine Orientierung im mathematisch positiven Sinne, d. h. gegen den Uhrzeigersinn, s. Abb. 3.1(a).

Jede innere Verbindungslinie in einem Delaunay-Gitter gehört zu zwei Dreiecken. Die Verbindungslinien, die nur zu einem Dreieck gehören sind *Randlinien*, d. h. gehören zum Rand, der das gesamte Delaunay-Gitter umgibt. Jeder Verbindungslinie wird eine binäre Variable **front** zugewiesen, die den Wert 0 hat, wenn die zwei Dreiecke, zu der diese Linie gehört, bereits gefunden sind oder wenn diese Linie eine *Randlinie* ist. Ist nur eine Verbindungslinie gefunden worden, erhält **front** den Wert 1. Diese Linien bilden somit die Suchfront, also die gegenwärtige Ausdehnung des Delaunay-Gitters. Die drei Seiten des ersten Dreiecks AB , BC und CA sind also Frontlinien. Nun werden die folgenden Schritte ausgeführt:

Für jede Frontlinie AB wird geprüft, ob ein Punkt P in der Nachbarschaft von A gefunden werden kann, so daß der Umkreis des Dreiecks APB keine weiteren Punkte enthält, s. Abb. 3.1(a). Ist dies der Fall, so definiert APB ein Delaunay-Dreieck. Dazu wird zunächst geprüft, ob der Punkt P auf der rechten oder der linken Seite der orientierten Verbindungslinie AB liegt, d. h. ob die Determinante der Matrix mit den Spaltenvektoren AB und AP negativ oder positiv ist. Wenn P auf der linken Seite von AB liegt, muß APB nicht in Betracht gezogen werden, da P im Innern der Suchfront liegt und schon behandelt worden sein muß. Alle Punkte P der Nachbarschaft von A werden in dieser Weise untersucht. Läßt sich keiner von diesen mit AB zu einem Delaunay-Dreieck verbinden, muß AB eine *Randlinie* sein und die zugehörige Variable **front** bekommt somit den Wert 0 zugewiesen. Der Algorithmus wird dann mit der Untersuchung der nächsten Frontlinie fortgesetzt. Wird hingegen ein Punkt P gefunden, der mit AB ein gültiges Delaunay-Dreieck APB bildet, so gibt es drei neue Verbindungslinien, nämlich AP , PB und BA , wobei letztere sich von AB nur durch ihre Orientierung unterscheidet. AB ist nun keine Frontlinie mehr und somit wird der Wert ihrer Variablen **front** auf 0 gesetzt. Die Wahrscheinlichkeit, einen verbundenen Punkt P zu finden ist nicht für alle Nachbarschaftspunkte von A gleich groß, sondern ist für die äußeren Punkte der 36 benachbarten Zellen des Quadratgitters wesentlich kleiner. Diese Information kann zur Geschwindigkeits-Optimierung des Algorithmus benutzt werden, indem man in den inneren Zellen mit der Suche beginnt und dann weiter nach außen zu den Zellen mit kleinerer Wahrscheinlichkeit für eine Verbindung fortfährt, s. Abb. 3.1.

Nun muß noch überprüft werden, ob die neuen Verbindungslinien AP und PB schon gefunden worden sind. Falls nicht, sind sie neue Frontlinien und ihrer Variable **front** wird der Wert 1 zugewiesen. Andernfalls wurden sie bereits als Frontlinien abgespeichert und nun wird ihrer Variablen **front** der Wert 0 zugewiesen. Man beachte, daß es nicht notwendig ist, für alle Verbindungslinien zu untersuchen, ob AP und PB bereits gefunden worden sind. In unserem Algorithmus machen wir Gebrauch von einem Datenfeld **connect**, in dem für jeden Punkt alle mit diesem Punkt verbundenen Linien gespeichert sind. Es ist daher ausreichend, zu überprüfen, ob AP und PB in den zu A , bzw. B gehörenden Bereichen von **connect** gespeichert sind. Die Suche nach neuen Linien wird solange es noch Frontlinien gibt fortgesetzt. Gibt es keine Frontlinien mehr, so ist das Delaunay-Gitter fertig.

Dann wird das Voronoi-Gitter folgendermaßen bestimmt: Für jeden Punkt P werden alle Dreiecke, die P als Eckpunkt enthalten, im mathematisch positiven Sinne, also

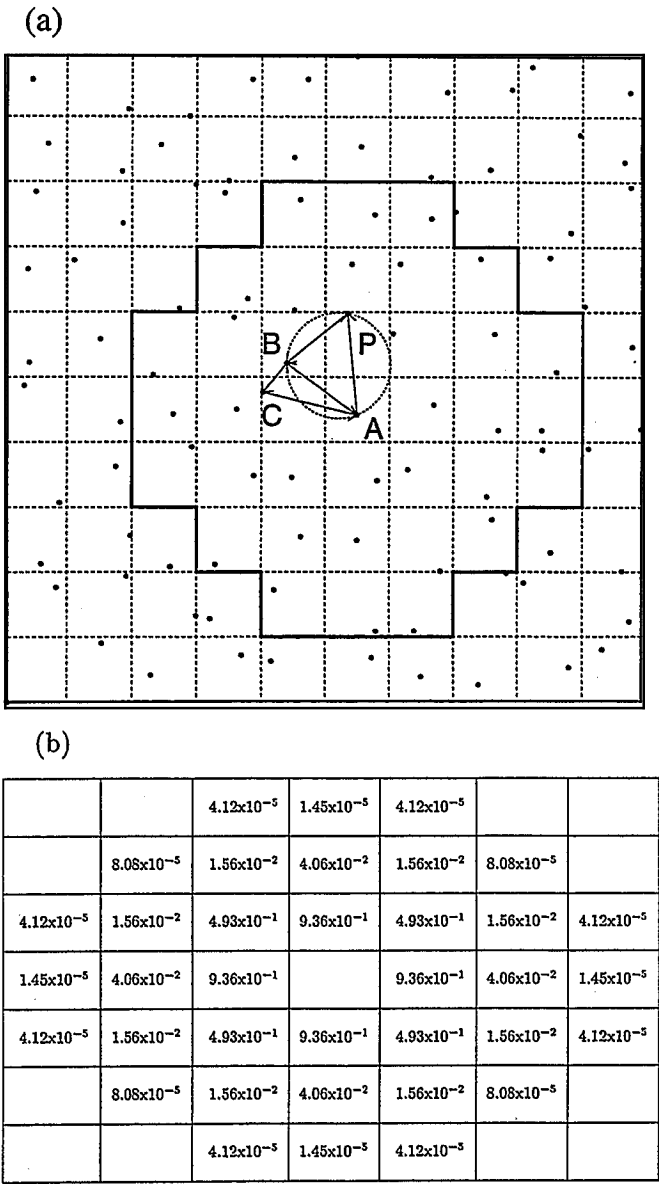


Abbildung 3.1: (a) Die Einteilung des zu diskretisierenden Gebietes in quadratische Gitterzellen, um das VZG zu erhalten. Ebenfalls gezeigt ist das Anfangsdreieck ABC und die aus 36 Zellen bestehende Nachbarschaft des Punktes A, in der sich Nachbarpunkte befinden können. Für die Verbindungslinie AB wird ein Punkt P gesucht, so daß der Umkreis des Dreiecks APB keine weiteren Punkte enthält. Dann ist das Dreieck APB ein gültiges Delaunay-Dreieck. (b) Die Wahrscheinlichkeiten für einen zentral gelegenen Punkt, Nachbarpunkte zu finden. Nur in 36 benachbarten Zellen ist diese von Null verschieden und man sieht, daß die Wahrscheinlichkeit sehr schnell nach außen abnimmt.

gegen den Uhrzeigersinn, geordnet. Die Mittelpunkte ihrer Umkreise werden miteinander verbunden. Das in dieser Weise konstruierte Polygon ist die mit P assoziierte Voronoi-Zelle. In zusammengefaßter Form läßt sich der Algorithmus wie folgt schreiben:

- Setze Zufallspunkte in die Quadrat-Zellen.
- Konstruiere das Startdreieck.
- Konstruiere das Delaunay-Gitter. Führe die folgenden Schritte aus, solange es Frontlinien AB gibt:
 1. Suche Punkt P , der AB zu einem Delaunay-Dreieck ergänzt. Falls dies nicht möglich ist, ist AB eine *Randlinie*.
 2. Prüfe, ob die beiden neuen Linien AP und PB bereits als Frontlinien gespeichert sind. Falls nicht, sind sie neue Frontlinien. Falls ja, sind sie keine Frontlinien mehr.
- Konstruiere das Voronoi-Gitter durch Verbinden der Umkreismittelpunkte aller mit einem Punkt verbundenen Dreiecke.

Die Rechenzeit $T(N)$ zur Konstruktion eines VZG ist proportional zur Anzahl N der Zufallspunkte. Dies ist darauf zurückzuführen, daß die Nachbarn für jeden Punkt nur lokal gesucht werden müssen, nämlich in den 36 benachbarten Quadratzellen.

3.3.2 Das Poisson'sche Zufallsgitter

In diesem Abschnitt beschreiben wir einen Algorithmus zur Generierung von Zufallsgittern, die auf Punkten mit gleichförmig verteilten x- und y-Koordinaten beruhen. Im Grenzfall unendlich großer Gitter entspricht dies einem Poisson-Prozeß [73].

Die Erzeugung eines Zufallsgitters basierend auf einem Poisson-Prozeß hat sehr viele Gemeinsamkeiten mit dem Zugang, der im vorigen Abschnitt beschrieben wurde. Andere Techniken sind jedoch an einigen Stellen notwendig, und zwar hauptsächlich deswegen, weil die Anzahl der Nachbarn im PZG nicht begrenzt ist. Der hier vorgestellte Algorithmus ist detailliert in [135] beschrieben und basiert auf dem in [37, 53] vorgeschlagenen Verfahren.

Der PZG Algorithmus

Während des Initialisierungsprozesses wird ein Anfangsdreieck ausgewählt. Die Verbindungslinien dieses Dreiecks bekommen den Status **aktiv**, ganz ähnlich der Variable **front** im letzten Abschnitt. Dann werden alle aktiven Verbindungslinien in der folgenden Weise behandelt, bis es keine aktiven Verbindungslinien mehr gibt. Um aktive Linien identifizieren zu können, wird eine Variable **treated** benutzt, die für jede neu gefundene Linie mit 0 initialisiert wird. Um das Zufallsgitter zu konstruieren hat man nun folgende Schritte auszuführen:

- Behandle die nächste **aktive** Verbindungslinie AB :
 1. Zeichne den Kreis, der AB als Durchmesser enthält und überprüfe, ob dieser weitere Punkte enthält.
 2. Wenn dies der Fall ist, wähle denjenigen Punkt C aus, für den der Umkreis des Dreiecks ACB den größten Radius hat.
 3. Falls dies nicht der Fall ist, bewege das Zentrum des Kreises um eine kleine Strecke ϵ , die abhängig von der Anzahl der Punkte des Gitters ist, in senkrechter Richtung zur Verbindungslinie AB . Wiederhole dies solange, bis der Kreis Punkte enthält. Wähle dann den Punkt, der zum Kreis mit dem kleinsten Radius führt.
- Speichere den gefundenen Punkt als neuen Eckpunkt und bilde zwei neue Linien durch Verbinden des neuen Punktes mit den Endpunkten der untersuchten Linie.
- Prüfe, ob diese beiden Linien bereits existieren. Falls ja, so sind diese Linien nun nicht mehr aktiv; falls nein, werden sie als neue aktive Linien aufgenommen.
- Die untersuchte Linie ist nun nicht mehr aktiv.

Für jede aktive Verbindungslinie muß man den nächsten Nachbarn gemäß der Delaunay-Konstruktion suchen. Eine Möglichkeit, dies zu tun, ist in Abb. 3.2 dargestellt. Der Kreis, der AB als Durchmesser enthält, wird eingezeichnet. Dann wird überprüft, ob dieser Kreis weitere Punkte in seinem Innern enthält (z. B. C_1). Wenn dies der Fall ist, wähle man unter diesen denjenigen aus, für den das durch AC_1B gegebene Dreieck den maximalen Umkreisradius hat. Ist dies nicht der Fall, bewege man den Mittelpunkt des Kreises um eine kleine Strecke ϵ in der Richtung senkrecht zu AB weiter (z. B. M_2, M_3). Die Größe von ϵ ist abhängig von der Dichte der Zufallspunkte. Dies wird solange fortgesetzt bis ein oder mehrere Punkte im Inneren des Kreises gefunden werden (z. B. C_2, C_3). Dann wird derjenige Punkt ausgewählt, der zum Dreieck mit dem kleinsten Umkreisradius führt. Man sollte bei der Verwendung von Quadratwurzeln und trigonometrischen Funktionen sehr vorsichtig sein, da sie die Rechenzeit drastisch verlängern können.

Datenstrukturen

Die beiden Grundelemente des Zufallsgitters, Punkte und Verbindungslinien, werden als verkettete Listen von speziellen Datenstrukturen organisiert, die die benötigten Informationen enthalten:

```

SITE{
    POINT p;           /* Koordinaten */
    int border;        /* Variable fuer Randpunkte */
    int nNb;           /* Anzahl der Nachbarn */
    NBNODE *pFirstNbNode; /* Adresse der ersten Nachbarecke */
    SITE *pNextSite;    /* Naechste Ecke in der Liste */
    SITE *pNextSlSite;  /* Naechste Ecke in der Untergitterzelle */
};

LINK{
    POINT p1, p2;      /* Koordinaten der Punkte */
    SITE *pSite1,*pSite2; /* Adressen dieser Punkte */
    int orient;        /* Orientierung der Verbindungslinie */
    char border;       /* Variable fuer Randlinie */
    char treated;      /* Variable fuer bereits behandelte Linie */
    LINK *pNextLink;   /* Naechste Verbindungslinie in der Liste */
};

```

Die Liste für die Ecken (**SITE**) wird nur einmal am Anfang initialisiert. Während der Konstruktion des Gitters werden Nachbarn dynamisch an jede Ecke angehängt. Die Adresse des ersten Nachbarn ist in **pFirstNbNode** enthalten. Jeder angehängte Nachbar ist auch wieder eine Struktur, die die Informationen über die Adresse seines Nachbarn und über die Verbindungslinien zu diesem Nachbarn enthält:

```

NBNODE{
    NBNODE *pNextNbNode; /* Naechster Nachbar */
    SITE *pNb;           /* Adresse des Nachbarn */
    LINK *pLink;         /* Adresse der verbindenden Linie */
};

```

Im Gegensatz zur Argumentationsführung in Abschnitt 3.3.1 ist hier nichts über die Region bekannt, wo sich die Nachbarn eines Punktes befinden können. Prinzipiell können diese an jeder beliebigen Stelle in der Ebene liegen. Jedoch ist es nicht akzeptabel diese Suche auf alle Punkte auszudehnen, da dies zu einem Zeitverhalten proportional zu N^2 führen würde. Alternativ könnte man eine baumartige Struktur einführen, die es einem erlaubt, Punkte in der Nachbarschaft aufzuspüren. Eine andere Möglichkeit besteht darin, das Gitter in Untergitter aufzuteilen derart, daß jede Zelle des Untergitters im Durchschnitt etwa die gleiche Anzahl an Punkten enthält. Diese Methode wurde hier verwendet. Der Zeiger **pNextSlSite** enthält die Adresse des nächsten Punktes der betrachteten Unterzelle. Für jede Verbindungslinie muß man also nur eine begrenzte

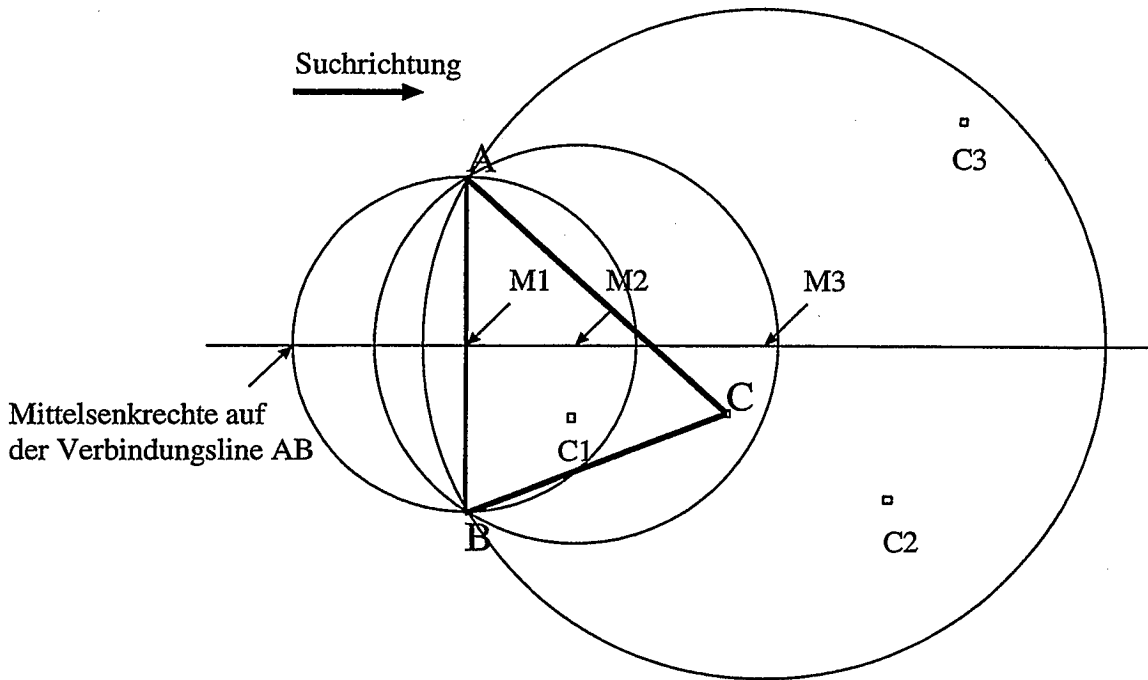


Abbildung 3.2: Suche eines Punktes C , der die Verbindungsline AB zu einem Delaunay-Dreieck ergänzt. M_1 , M_2 und M_3 sind die Mittelpunkte der Kreise und C_1 , C_2 und C_3 Punkte des Gitters.

Anzahl von Punkten untersuchen. Eine Anzahl von etwa 5 bis 10 Punkten pro Unterzelle hat sich als optimal erwiesen. Für jede aktive Verbindungsline müssen die Unterzellen selbst und die 8 nächsten Nachbarn untersucht werden.

Während der Konstruktion des Gitters können Situationen auftreten, wo Nachbarpunkte gefunden werden, die zu Verbindungslinien gehören, die bereits gefunden worden sind und den Status **aktiv** haben. Diese müssen dann als behandelt gekennzeichnet werden, indem die Variable **treated** in der Datenstruktur für die Verbindungslinien (**LINK**) gesetzt wird. Diese werden dann in der verketteten Liste übersprungen.

Das Abbruchkriterium hängt von dem Problem ab, das man untersuchen will. Gibt es keine periodischen Randbedingungen, so muß man eine Grenze angeben, über die hinaus der Algorithmus abbrechen soll und nicht nach weiteren Punkten sucht. Sind die Randbedingungen periodisch, ist eine solche Grenze nicht notwendig, da der Algorithmus automatisch terminiert.

Geschwindigkeit und Speicherbedarf

Die Geschwindigkeit eines Programms hängt von der Art des Algorithmus und der speziellen Realisierung auf dem Computer ab. Das beste Zeitverhalten, das man bekommen kann ist $O(N)$, d. h. die Rechenzeit $T(N)$ wächst linear mit der Anzahl der Punkte an.

Um alle überflüssigen Berechnungen zu vermeiden, wurden die folgenden Maßnahmen ergriffen:

- Während der Konstruktion des Gitters muß man, immer wenn neue Nachbarn und damit zwei neue Verbindungslinien gefunden werden, überprüfen, ob diese sich nicht schon in der Liste der als **aktiv** charakterisierten Verbindungslinien befinden. Um den Einfluß dieser Aufgabe auf das Zeitverhalten in möglichst kleinen Grenzen zu halten, ist jedem Punkt vermöge der Struktur **NBNode** nicht nur sein Nachbar **pNb** bekannt, sondern auch die mit ihm verbundenen Linien **pLink**. So müssen also nur die Linien, die bereits mit dem Punkt verbunden sind, untersucht werden. Die Anzahl dieser Linien ist im Durchschnitt klein.
- Bei der Suche nach potentiellen Nachbarn hat man Kreise zu konstruieren und deren Mittelpunkte zu verschieben. Man muß dafür Sorge tragen, daß die Größe ϵ , die die Länge bezeichnet, um die der Mittelpunkt verschoben wird, von der Dichte der Zufallspunkte abhängt. Diese Strecke ϵ ist dann optimal gewählt, wenn das Produkt aus Punktdichte und Fläche des ersten Kreises von der Größenordnung eins ist.

Mit diesen Vorkehrungen sollte der Algorithmus ein Zeitverhalten von $O(N)$ haben, da für jeden Punkt eine feste Anzahl an Operationen und Speicherallozierungen vorgenommen werden muß.

Diese theoretischen Überlegungen spiegeln jedoch nicht die wirkliche Situation wieder. Dynamische Speicherzuweisungen sind zwar intuitiv und leicht zu programmieren, der Computer wird jedoch gezwungen, immer größer werdende Speicherbereiche zu verwalten, so daß die Zeit für die Speicherallozierung stärker als die Systemgröße wächst.

Es ist deswegen vorteilhaft, den gesamten benötigten Speicher gleich bei der Programminitialisierung anzufordern. Immer wenn neuer Speicher benötigt wird (für neue Nachbarn oder Verbindungslinien), wird er aus diesem Reservoir geholt. Auf diese Art und Weise muß der Computer keine zusätzliche Arbeit für die Speicherallozierung leisten. Die Allokierung des Speichers gleich zu Anfang bringt keine Probleme mit sich, da man ja aus der Euler-Beziehung weiß, daß die durchschnittliche Zahl von Nachbarn 6 ist. Insbesondere für sehr große Gitter ($N \geq 10000$) ist dies sehr vorteilhaft, da die Speicherallozierung dann einen bedeutenden Anteil an der gesamten Rechenzeit einnimmt.

3.3.3 Dynamische Zufallsgitter

Nun wollen wir eine Methode beschreiben, wie man Zufallsgitter dynamisch erzeugen kann, sog. dynamische Zufallsgitter (DZG). Dies macht es möglich, Zufallsgitter in Simulationen dynamisch zu benutzen [97, 98]. Die Grundidee ist es, von einem existierenden Gitter auszugehen (von einem Dreiecksgitter) und dann sukzessive dieses Gitter zu einem Zufallsgitter zu verändern, indem man Punkte verschiebt. Das Gitter erfüllt zu jeder Zeit die Voronoi-Konstruktion und dies gewährleistet, daß auch das Gitter, das man am Schluß erhält, ein Delaunay-Gitter ist.

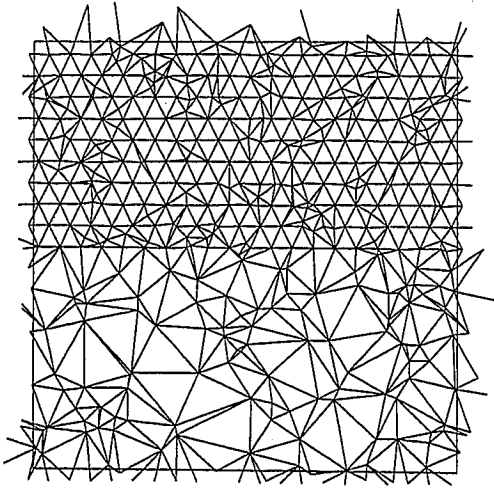
Durch den DZG Algorithmus erhält man ein vektorisierbares Zufallsgitter, indem man die Punkte - einen nach dem anderen - innerhalb der Zelle des Quadratgitters, die zu dem Punkt gehört, verschiebt, s. Abb. 3.1. Diese Vorgehensweise ist in Abb. 3.3(a)-(b) illustriert. Das dynamische Poisson'sche Zufallsgitter erhält man, indem man die Punkte - wieder einer nach dem anderen - beliebig weit innerhalb des ganzen Gitters verschiebt, s. hierzu Abb. 3.3(c)-(d). Das Zeitverhalten für dynamische Zufallsgitter ist komplizierter als bei den anderen beiden Algorithmen. Dies ist auf die Suchphase des Algorithmus zurückzuführen, den wir weiter unten im Detail beschreiben werden. Verglichen mit den anderen beiden Algorithmen ist der für dynamische Zufallsgitter deutlich langsamer.

Der DRL Algorithmus

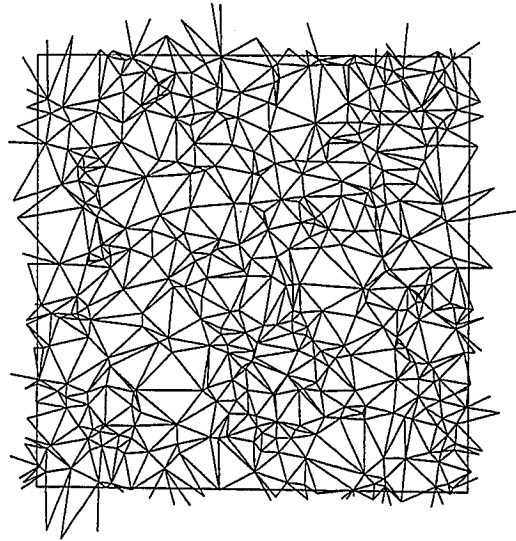
Wir beschreiben nun die grundlegenden Elemente des Algorithmus. Für eine detailliertere Beschreibung, siehe [97, 98]. Wenn ein Punkt langsam von einem Platz zu einem anderen bewegt wird, kann man sich alle Veränderungen im Gitter als "Umklappen" von Verbindungslinien vorstellen. Ein solcher Prozeß ist in Abb. 3.4 illustriert, wo die Verbindungslinie EP im Teil (a) der Abbildung zur Linie DF im Teil (b) umklappt. Die Punkte P und E verlieren dabei jeweils einen Nachbarpunkt, während die Punkte D und F jeweils einen Nachbarn gewinnen. Man kann sich die Funktion unseres Algorithmus so vorstellen, daß ein Punkt langsam durch das Gitter bewegt wird und dabei eine Reihe von solchen "Umklapp"-Vorgängen stattfinden. Ein solcher geschieht immer dann, wenn (i) der Punkt P in den Umkreis eines Dreiecks eintritt (wie bei den Punkten A , B und C in Abb. 3.4(a)), was die Anzahl der Nachbarn um eins erhöht oder wenn (ii) der Punkt P den Umkreis eines Dreiecks verläßt (wie in Abb. 3.4(b) für die Punkte D , E und F dargestellt). Letzteres reduziert die Anzahl der Nachbarn um eins.

Es ist zwar unmöglich, die Anzahl der "Umklapp"-Vorgänge vorauszuberechnen, aber man kann sich vorstellen, daß diese sequentiell vonstatten gehen, wenn man den Punkt auf seiner Bahn zu seiner neuen Position verfolgt. Nun ist es jedoch nicht notwendig, diese Prozesse einen nach dem anderen auszuführen, während man der Trajektorie des Punktes folgt. Wenn ein Punkt bewegt wird, kann man dies auch so verstehen, daß er zunächst entfernt wird, wobei er ein Gitter mit $N - 1$ Punkten hinterläßt. Dann berechnet man die Nachbarschaft dieses Punktes neu. Anschließend setzt man den Punkt auf seine neue Position und es wird bestimmt zu welchem Dreieck er gehört. Da man im Vorhinein nicht weiß zu welchem Dreieck er gehören wird, muß dieses durch eine Suche durch das Gitter bestimmt werden [119]. Schließlich wird der Punkt dann hinzugefügt und das Gitter wird gemäß den beiden oben beschriebenen Grundprozessen korrigiert. In schematisierter Form sieht der DZG Algorithmus wie folgt aus:

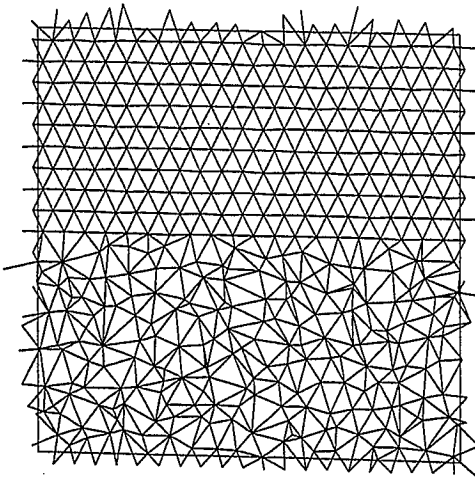
- Generiere ein Dreiecksgitter mit N Punkten und der Topologie eines Torus (periodische Randbedingungen).
- Behandle jeden Punkt des Gitters wie folgt:



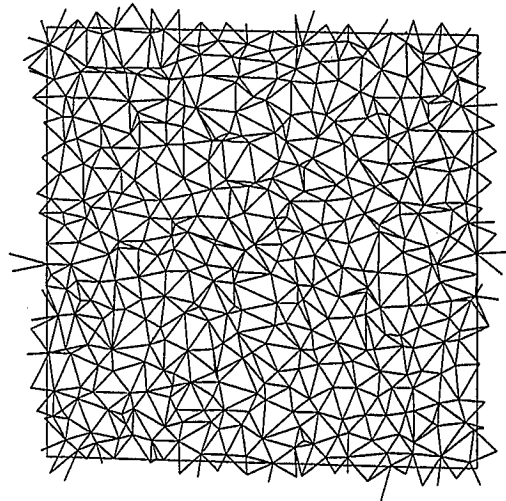
"Halbes" Poisson'sches Zufallsgitter



Poisson'sches Zufallsgitter



"Halbes" vektorisierbares Zufallsgitter



Vektorisierbares Zufallsgitter

Abbildung 3.3: Zufallsgitter, (a) und (b) Poisson'sche Zufallsgitter, (c) und (d) vektorisierbare Zufallsgitter, jeweils mit periodischen Randbedingungen. Die Konstruktion erfolgt gemäß der dynamischen Konstruktionsmethode. Gezeigt wird jeweils ein Bild des Gitters, nachdem die Hälfte der Punkte verschoben worden ist und ein Bild des fertig konstruierten Gitters. Die betrachteten Systeme haben die Größe 20×20 .

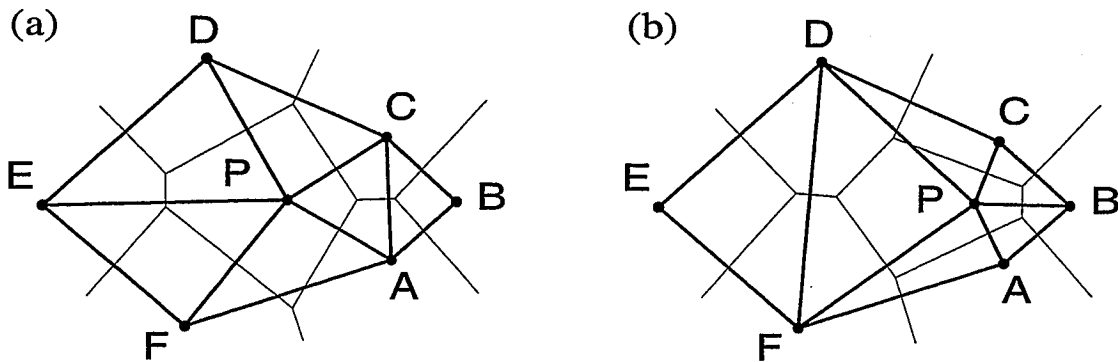


Abbildung 3.4: Teil eines Delaunay'schen Zufallsgitters, (a) bevor und (b) nachdem der Punkt P verschoben worden ist. Es finden zwei "Umklappvorgänge" statt. Der vom Punkt P aus links stattfindende Prozeß reduziert die Zahl der Nachbarn des Punktes E um eins, während der von P aus rechts stattfindende Prozeß die Anzahl der Nachbarn des Punktes B um eins erhöht. Die dick gezeichneten Linien stellen die Verbindungslinien des Delaunay'schen Zufallsgitters dar, während die dünn eingezeichneten das dazu duale Voronoi-Gitter kennzeichnen.

1. Entferne den Punkt P mitsamt seinen Verbindungslinien zu seinen Nachbarn.
2. Korrigiere das nun aus $N - 1$ Punkten bestehende Gitter.
3. Setze den Punkt P an eine neue (zufällig) gewählte Position wieder in das Gitter ein.
4. Durchsuche das Delaunay-Gitter mit $N - 1$ Punkten, um festzustellen, innerhalb welchen Dreiecks der Punkt P nun liegt.
5. Korrigiere das Gitter gemäß den beiden oben beschriebenen elementaren "Umklap"-Vorgängen, s. Abb. 3.4.

Datenstrukturen

Dem Algorithmus liegt eine dynamische Zeigerstruktur zugrunde. Für jeden Eckpunkt ist eine Struktur **SITE** definiert, in der alle mit dieser Ecke assoziierten Informationen gespeichert werden. Unter diesen Informationen befindet sich auch eine Liste **NN** mit den Zeigern auf die nächsten Nachbarn. Für jeden Eckpunkt in dieser Liste gibt es ebenso einen Zeiger, der auf den ursprünglichen Eckpunkt zeigt. Dies hat den Vorteil, daß die mit einer Ecke verbundenen Informationen nur einmal gespeichert werden müssen. Die Eckpunkt-Informationen und die verketteten Listen mit den nächsten Nachbarn sind wie folgt organisiert:

```

SITE {
    POINT p;      /* Koordinaten */
    int    n;      /* Nummer des Punktes */
    int    q;      /* Koordinationszahl */
    NN     *pNN;   /* Zeiger zum ersten Nachbarn */
};

NN {
    SITE    *pSite; /* Zeiger zur Ecke */
    CIRCLE  circle; /* Umkreis */
    int     pos;     /* Position relativ zur Ecke */
    NN      *pNN;    /* Zeiger zum naechsten Nachbarn */
};

```

Die Korrektur des Gitters, nachdem ein "Umklapp"-Vorgang stattgefunden hat, wird durch Umkonfigurieren der Zeiger `pNN` in der Liste der nächsten Nachbarn gewährleistet. Wenn ein neues Dreieck erzeugt oder ein altes gelöscht wird, werden in der Liste der nächsten Nachbarn `NN` die Informationen über die Umkreise korrigiert, bzw. eingetragen [97].

Zeitverhalten des Algorithmus

Die durchschnittliche Rechengeschwindigkeit des Algorithmus kann folgendermaßen geschrieben werden:

$$T(N) = O(N(R(N) + S(N) + A(N))). \quad (3.3)$$

Hierbei kennzeichnet $R(N)$ die durchschnittliche Zeit zum Entfernen eines Punktes, $S(N)$ die mittlere Suchzeit und $A(N)$ die Zeit zum Einfügen eines Punktes. Für ein Dreiecksgitter ist die Koordinationszahl 6, so daß sowohl $R(N)$ als auch $A(N)$ 6 Operationen unter den nächsten Nachbarn beanspruchen. Aus Gleichung 3.2 wissen wir, daß die durchschnittliche Koordinationszahl für Dreiecke 6 ist, so daß sowohl $R(N)$ als auch $A(N)$ im Durchschnitt konstant sind und nicht mehr von N abhängen.

Die Suchzeit $S(N)$ ist proportional zur Entfernung zwischen der alten und der neuen Position des betrachteten Punktes P [97]. Im Falle des VZG befinden sich die neuen Positionen der Punkte innerhalb einer festgelegten Entfernung von den alten Positionen, so daß auch $S(N)$ eine Konstante ist. Wir erwarten daher ein Zeitverhalten, das linear zur Anzahl N der Punkte ist, also $O(N)$. Für das Poisson'sche Zufallsgitter ist die Suchzeit im schlimmsten Falle proportional zur Seitenlänge des Systems, also $\sqrt{N}$, was insgesamt zu einem Zeitverhalten der Art $O(N\sqrt{N})$ führt.

N	100	1000	10000	50000	100000	200000	300000
VZG	0.04	0.18	1.72	8.90	18.3	38.6	60.3
PZG	0.02	0.20	2.05	11.3	23.0	47.3	73.9
DVRL	0.11	0.99	10.15	50.6	101.0	207	308
DPRL	0.14	1.70	29.20	264.8	713.8	1938	3476

Tabelle 3.1: Der obere Teil der Tabelle gibt die Geschwindigkeit zur Konstruktion eines VZG und eines PZG mit Hilfe der entsprechenden Algorithmen an. Der untere Teil enthält die Meßwerte für die dynamischen Varianten dieser Algorithmen.

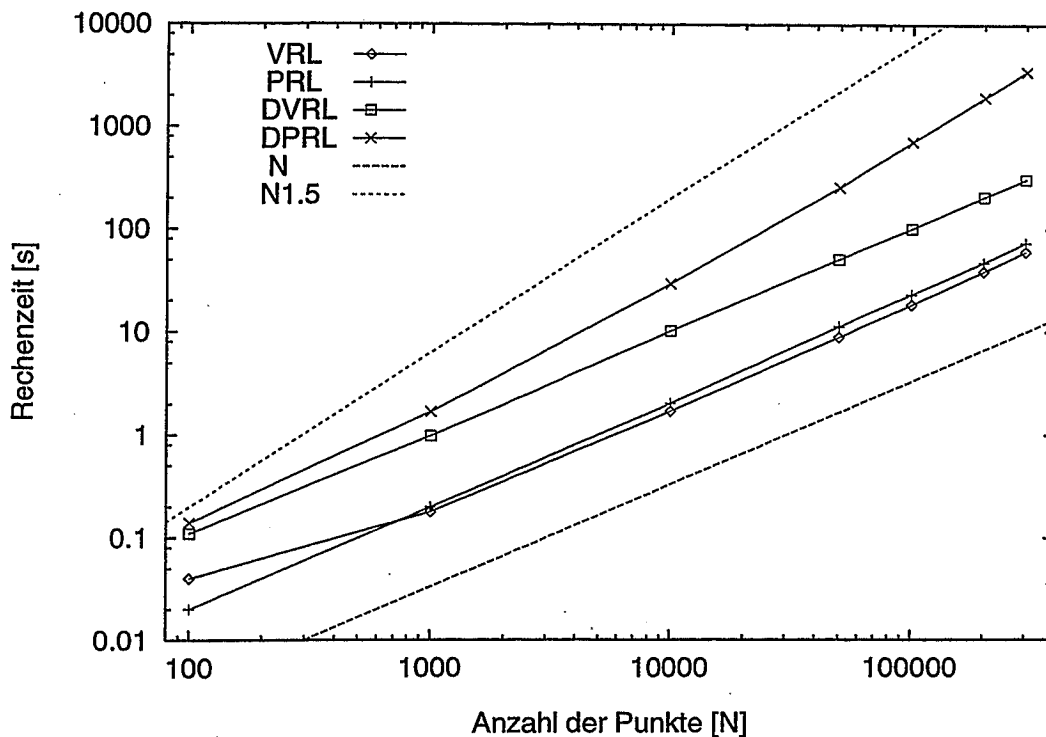


Abbildung 3.5: Rechenzeiten für das Poisson'sche, das vektorisierbare und die beiden dynamischen Zufallsgitter. Die Zeiten sind in Millisekunden angegeben. Zum Vergleich wurden die Geraden eingezeichnet, die einen linearen Anstieg, bzw. einen Anstieg wie $N^{1.5}$ der Rechenzeit gegen die Anzahl N der Punkte angeben.

3.4 Ergebnisse

Alle in diesem Kapitel vorgestellten Rechnungen wurden auf Sun Sparc 10 Workstations ausgeführt. In Tabelle 3.1 und Abb. 3.5 sind die Meßergebnisse für das Geschwindigkeitsverhalten der vier Algorithmen (VZG, PZG, DVZG und DPZG) dargestellt. Bei den statischen Varianten ist ein nahezu linearer Anstieg der Rechenzeit mit der Anzahl N der Punkte festzustellen. Hingegen steigt die Rechenzeit für sehr große Systeme stärker als linear an, da hier fast der gesamte Speicher der Workstation gebraucht wurde. Dann beginnt der Computer, Speicher auf die Festplatte auszulagern ("pagen"). Im Falle des statischen VZG läßt sich das Zeitverhalten in der Form $T(N) \approx 0.20N$ angeben, während das statische PZG durch den Ausdruck $T(N) = 0.24N$ beschrieben wird. Dabei ergibt sich die benötigte Rechenzeit dann in Millisekunden.

Für den dynamischen VZG Algorithmus ergibt sich ebenfalls ein linearer Anstieg der Rechenzeit mit der Anzahl der Gitterpunkte, der mit $T(N) = 1.01N$ angegeben werden kann. Im Falle des dynamischen PZG Algorithmus ist das Verhalten, wie aus den theoretischen Überlegungen zu erwarten war von der Art $O(N\sqrt{N})$. Ein Ausdruck der Form $T(N) = aN(b + \sqrt{N})$, mit $a \approx 150msec$ und $b \approx 65msec$ beschreibt das Verhalten der Kurve gut, s. Abb. 3.5. Für sehr große N nähert sich dieser Ausdruck asymptotisch der vorhergesagten Form $O(N\sqrt{N})$. Wenn man im Vorhinein eine Kenntnis über die Art der Verteilung der Punkte hat, kann es auch beim dynamischen Zufallsgitter von Vorteil sein - ähnlich wie im Falle des statischen PZG - ein Untergitter einzuführen. Aber hier wurde nur der allgemeine Algorithmus für die Konstruktion von dynamischen Zufallsgittern benutzt. Es wäre also noch Raum für weitere Optimierungen.

3.5 Zusammenfassung

Es wurden vier Algorithmen vorgestellt, die verschiedene Arten von Delaunay-Gittern und ihre assoziierten (mathematisch dualen) Voronoi-Gitter konstruieren. Wir haben explizit beschrieben, wie sich solche Algorithmen in den Programmiersprachen C, bzw. FORTRAN implementieren lassen. In den C Programmen wurde dabei extensiver Gebrauch von Zeigern und Strukturen gemacht.

Im Falle des statischen VZG und PZG haben wir gefunden, daß das Zeitverhalten gut als linear zur Anzahl der Punkte beschrieben werden kann. Jedoch steigt die benötigte Rechenzeit stärker an, wenn man große Systeme in Betracht zieht und fast der gesamte Computerspeicher benötigt wird. Es wurden dynamische Varianten des VZG und PZG Algorithmus vorgestellt, mit denen man Gitter dynamisch in Simulationen verändern kann. Dies erkauft man sich mit einem schlechteren Zeitverhalten für die anfängliche Konstruktion des Gitters.

Kapitel 4

Simulation von Scherzellen

4.1 Einleitung

Die Scherung von dichtgepackten granularen Materialien spielt in vielen Anwendungen der Ingenieurwissenschaften, wie der Bodenmechanik und dem Bau von Silos und Trichtern, eine bedeutende Rolle. Ebenso wichtig ist sie in geophysikalischen Anwendungen. In Gebieten, die überwiegend aus Sedimentationsgestein oder Granit gebildet sind, bricht das Material auf und bewegt sich auseinander, wenn es einer großen Scherkraft ausgesetzt ist und wenn der hydrostatische Druck, der das Gestein zusammenpreßt, groß genug ist [10, 131, 132]. Auf wesentlich größerer Längenskala haben wir das Beispiel der tektonischen Platten, die sich gegeneinander bewegen, wobei das Gestein an der Grenze zweier tektonischer Platten zu Gesteinsmehl (Mylonit) zerkleinert wird. Schließlich wird dieses noch weiter in eine lehmartige Substanz zermahlen, die man im Deutschen "Störungsletten" nennt [27, 95]. Auf die Bewegung tektonischer Platten werden wir im nächsten Kapitel, das sich mit der Simulation von Erdbeben beschäftigt, zurückkommen. Ein nahezu zweidimensionales Beispiel für Scherbewegungen kann man beobachten, wenn Packeis aufbricht [153]. Die Bestimmung des Reibungs- und Dilatanzwinkels, wie sie in der klassischen Theorie der Plastizität nach Mohr und Coulomb verwendet werden [149], wird in ingenieurtechnischen Anwendungen oft mit dem Scherapparat nach Jenike durchgeführt [158].

Wegen der einfachen Form des Spannungstensors in diesen Scherzellen und der relativen Leichtigkeit, mit der man die externen Parameter kontrollieren kann, sind sie Gegenstand vieler numerischer [31, 50, 163, 186] und theoretischer Arbeiten [152]. Man mißt, wie die Scherkraft sich mit der Geschwindigkeit verändert. Steigt diese an, so spricht man von Scherhärtung ("shear hardening"), wird sie mit zunehmender Geschwindigkeit kleiner, so nennt man dies Scherschwächung ("shear weakening", "shear thinning"). Bei der Scherung von granularen Medien beobachtet man oft, daß das Volumen dabei anwächst. Dieses Phänomen, das man als Dilatanz bezeichnet, entspricht einer negativen Poissonzahl. Die Dilatanz wird vor allem durch die Normalkraft kontrolliert, die auf die Scherzelle wirkt.

Viele numerische Techniken werden benutzt, um Scherzellen zu simulieren. Zugänge mit Hilfe der Kontinuumsstheorie, wie z. B. bei der FLAC Technik [131, 132], erlauben es, sehr große Systeme zu betrachten, können aber nicht die mikromechanischen Aspekte reproduzieren, die auf die geometrische Form der Teilchen zurückzuführen sind. Eine sehr leistungsstarke Methode, die die Bahnen und Rotationen der Teilchen verfolgt, ist die Diskrete Elemente Methode (DEM), die von Cundall und Strack eingeführt wurde [46]. Sie basiert auf den gleichen numerischen Grundlagen wie die Molekular-Dynamik Methode, die wir in unseren Simulationen verwenden [182, 184, 185].

4.2 Das numerische Experiment

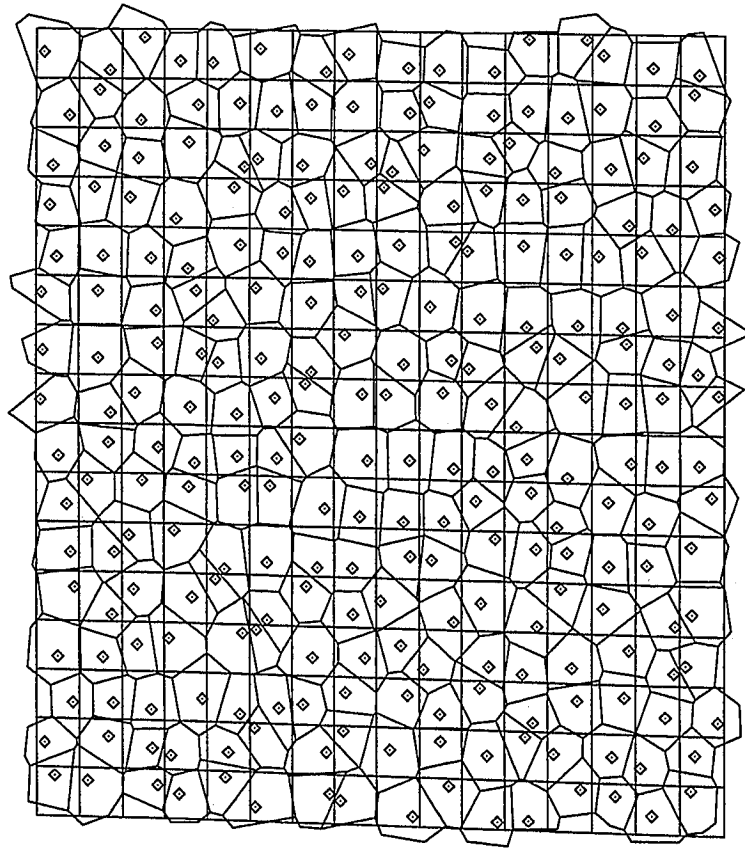


Abbildung 4.1: Ein Voronoi-Gitter von 16 mal 16 Zellen, wie es für unsere Simulationen benutzt wurde. Gezeigt sind auch die Zufallspunkte, die das Gitter definieren, sowie die quadratische Struktur des Gitters, in dem diese Punkte angeordnet sind.

Zur Simulation von Scherzellen verwenden wir die in den Kapiteln 2 und 3 herausgearbeiteten Grundlagen. Eine oft gebrauchte Scherzelle ist der Couette-Apparat, der aus

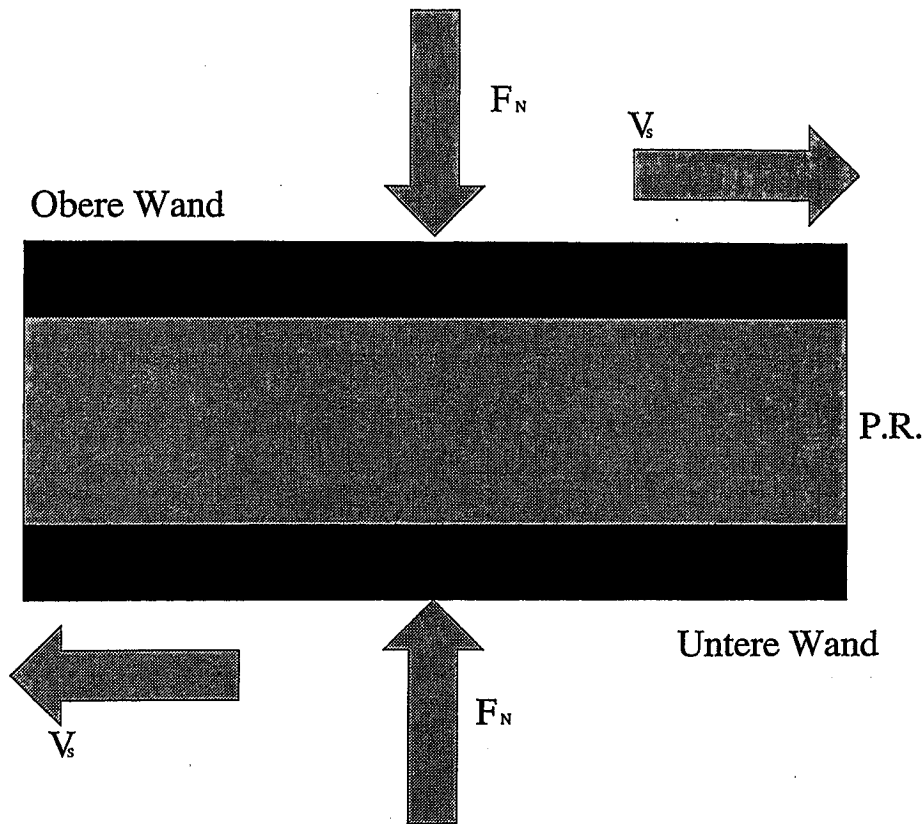


Abbildung 4.2: Skizze der für unsere numerischen Experimente benutzten Scherzelle. Eine Normalkraft F_N wirkt auf die obere und untere Wand des Gerätes. Diese werden mit einer Geschwindigkeit v_s gegeneinander geschert.

einem festen inneren Zylinder und einem drehbaren äußeren Zylinder besteht, der gegen diesen bewegt wird. Das zu untersuchende granulare Material befindet sich zwischen den beiden Zylinderwänden. Ein solches Gerät wurde auch in der ersten systematischen Arbeit im Bereich des schnellen Fließens von granularen Medien von Bagnold benutzt [12, 13]. Die gebräuchlichsten Scherzellen sind die Jenike-Scherzelle und das Ringschergerät. Eine genaue Beschreibung solcher Geräte, sowie Vorschläge zu ihrer Verbesserung findet man in [20, 77, 123, 160].

Später wurden solche Systeme mit sphärischen Teilchen mit Hilfe der Molekulardynamik Methode auch auf dem Computer modelliert [30, 31, 44, 45, 81, 133]. Eine Simulation, die der unseren sehr ähnlich ist, wurde in der Dissertation von Handley [66] beschrieben. Um die Zylindersymmetrie zu simulieren, benutzen wir periodische Randbedingungen in der horizontalen Richtung, d. h. Teilchen, die das System rechts verlassen kommen links wieder hinein und umgekehrt. Wir gehen von einer vollkommen dichten Packung von Voronoi-Polygonen aus, deren Konstruktion wir in Kapitel 2.4 beschrieben haben, s. Abb. 4.1. Die Zellen des oberen und unteren Randes werden als

fest miteinander verbunden betrachtet. Sie werden mit konstanter Geschwindigkeit v_S gegeneinander geschert, wie in Abb. 4.2 dargestellt. Dabei werden sie mit einer Normalkraft F_N gegeneinander gedrückt. Aufgrund dieser Kraft und der Wechselwirkung mit den inneren Granulatkörnern werden sich diese Wände in vertikaler Richtung bewegen. Wir messen die Distanz, um die sie sich insgesamt in der Simulation bewegen, welche als Dilatanz D_Y bezeichnet wird. Die Simulation ist so ausgelegt, daß die Wände sich nicht drehen können, d. h. ihr Abstand voneinander ist an allen Stellen gleich groß.

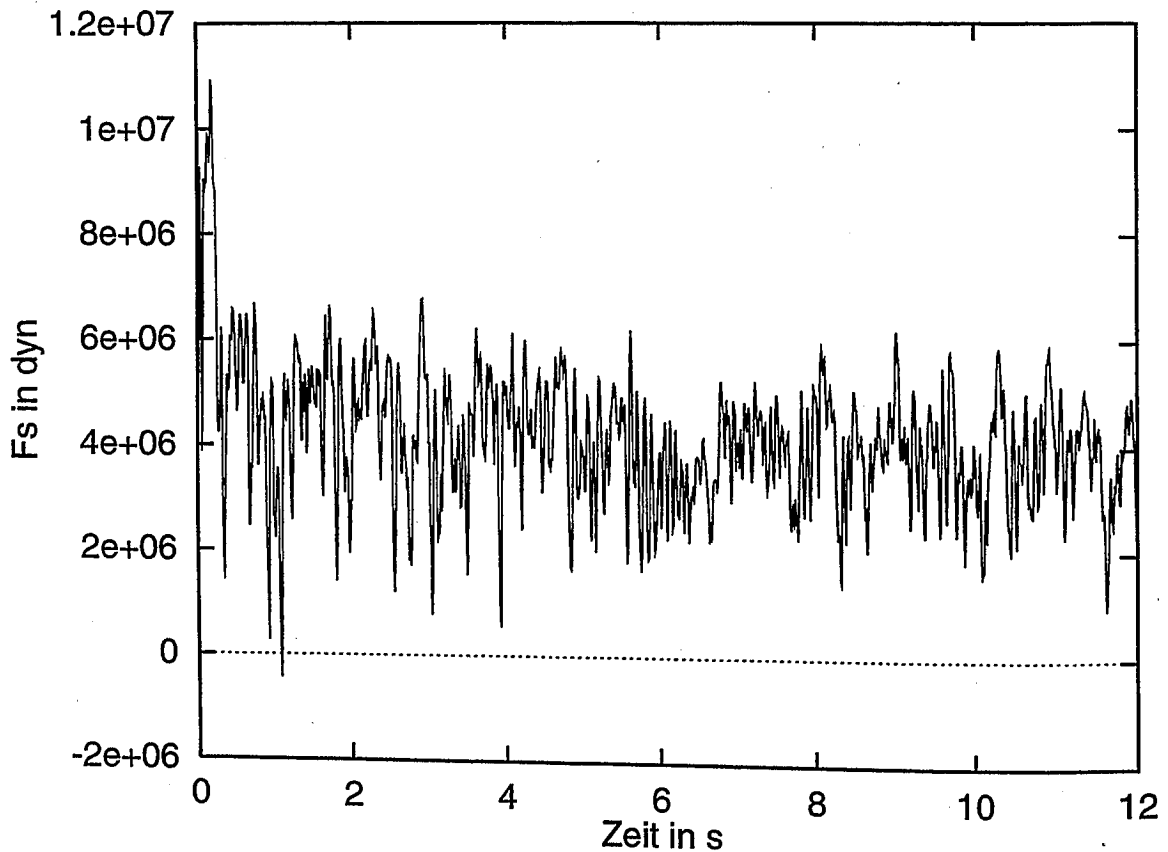


Abbildung 4.3: Scherkraft F_S in dyn gegen Zeit in Sekunden für einen typischen Simulationslauf. Das Signal weist sehr starke Fluktuationen auf. An einer Stelle wird die Scherkraft sogar negativ.

Wir wollen ebenfalls die Scherkraft F_S messen, die nötig ist, um bei gegebener Normalkraft F_N eine gegebene Schergeschwindigkeit v_S aufrecht zu erhalten. Diese wird aufgrund der unregelmäßigen Teilchenformen starke Fluktuationen aufweisen. In Abb. 4.3 ist die Scherkraft F_S für einen typischen Simulationslauf gegen die Zeit aufgetragen. Wie man sieht, sind die Fluktuationen sehr stark und an einer Stelle wird F_S sogar negativ. Ähnlich starke Fluktuationen treten auch im Signal der Dilatanz gegen die Zeit auf. Um dennoch gute Meßwerte mit kleinen Fehlerbalken zu bekommen, führen wir

mehrere Simulationsläufe mit den gleichen physikalischen Parametern aber mit unterschiedlichen Initialisierungswerten für das Zufallsgitter aus. Für einen Parametersatz haben wir dazu jeweils 32 Simulationsläufe mit Systemen ausgeführt, die aus 16 mal 16 Zellen bestehen. Für einen Simulationslauf waren dabei etwa 120000 Zeitschritte nötig, bis sich das System im quasistatischen Bereich befand. Die Simulation lief auf dem Paragon Parallelrechner des ZAM (Zentralinstitut für angewandte Mathematik), der mit 140 Prozessoren vom Typ Intel i860 mit jeweils 32 MB lokalem Speicher ausgestattet ist. Ein Simulationslauf, wie er oben beschrieben ist, lief dann in 4 Stunden auf 32 Knoten dieses Supercomputers.

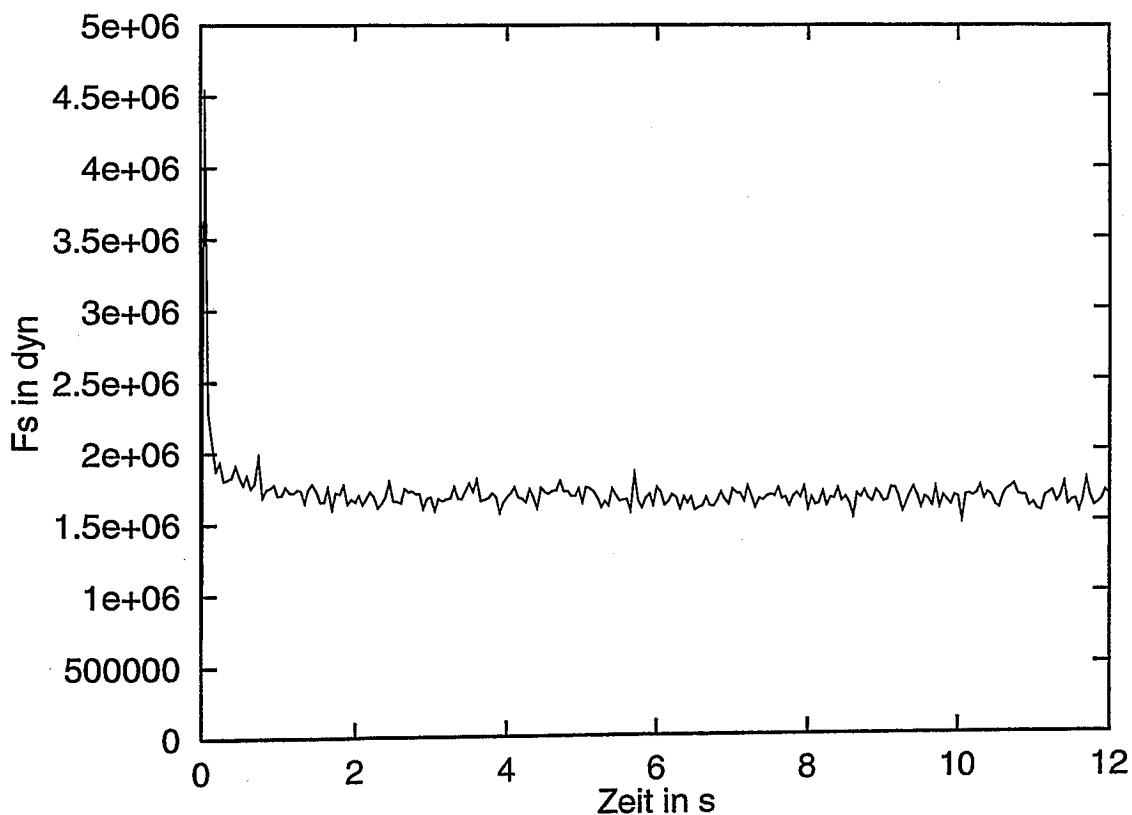


Abbildung 4.4: *Mittelung der Scherkraft über 32 Simulationsläufe mit verschiedenen Zufallsgittern. Die Fluktuationen sind sehr viel kleiner als in Abb. 4.3. Man erkennt, daß zu Beginn der Simulation die Scherkraft zunächst sehr stark ansteigt und dann gegen einen Sättigungswert hin abfällt.*

Die Meßwerte wurden über die 32 Läufe gemittelt, was Ergebnisse mit sehr kleinen Fehlertoleranzen ergab. In Abb. 4.4 ist der Mittelwert der Scherkraft gegen die Zeit für einen typischen Simulationslauf dargestellt. Wie man sieht, sind die Fluktuationen sehr viel geringer, was zu Ergebnissen mit sehr viel kleineren Fehlerbalken führt. Man erkennt nun auch sehr klar das Zeitverhalten der Scherkraft, welches man in Abb. 4.3

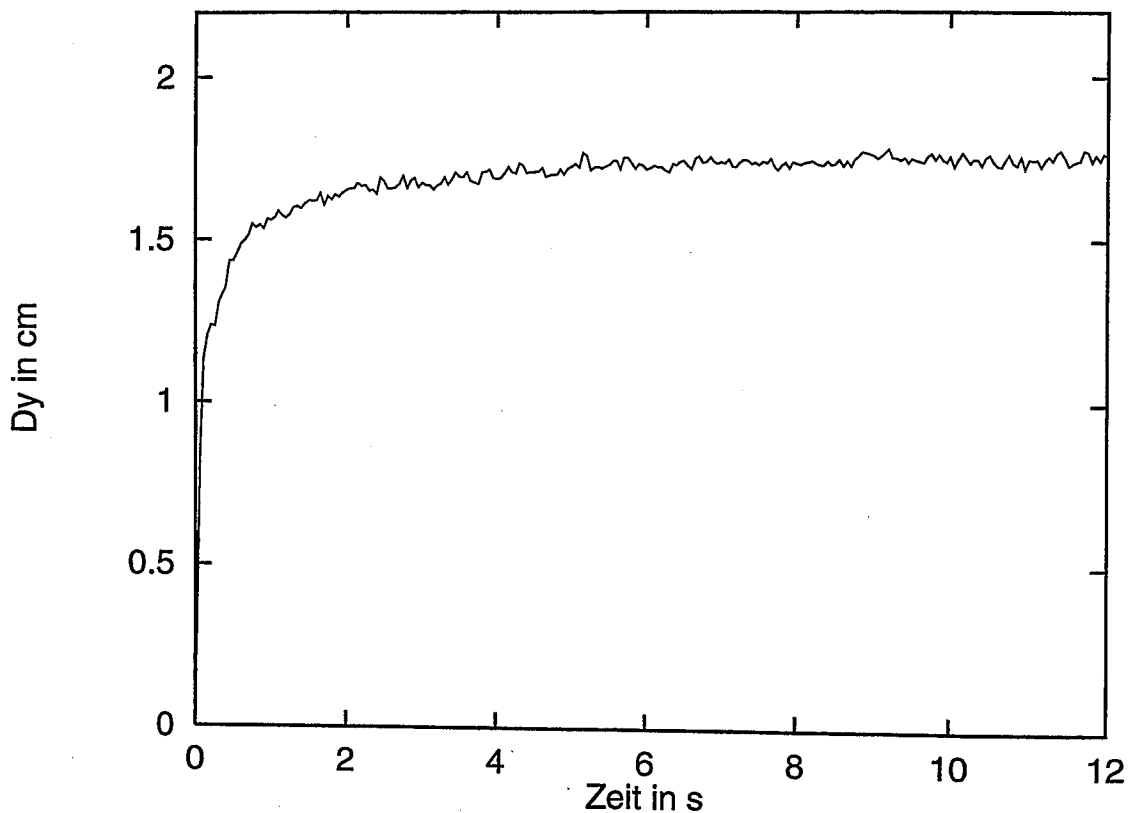


Abbildung 4.5: Dilatanz gegen Zeit als Mittelung über 32 verschiedenen Simulationsläufe. Zu Beginn wächst die Dilatanz sehr stark an und geht dann asymptotisch gegen einen Sättigungswert ($F_N = 10N$, $v_s = 20\text{cm/s}$).

nur erahnen kann. Am Anfang steigt die Scherkraft sprunghaft von Null auf einen hohen Wert an, um dann ebenfalls sehr schnell wieder abzufallen und allmählich gegen einen Sättigungswert zu streben. Um die Scherkraft F_s für einen gegebenen Satz physikalischer Parameter zu bestimmen, bilden wir den Mittelwert über die letzten 20000 Zeitschritte, wo sich das System im saturierten Zustand befindet. Betrachtet man die Abb. 4.5, wo der Mittelwert der Dilatanz D_Y über 32 Simulationsläufe für den gleichen Parametersatz wie in Abb. 4.4 dargestellt ist, so erkennt man, daß auch diese zu Beginn der Simulation ein sehr abruptes Verhalten zeigt. Sie steigt zunächst sehr stark an und strebt dann asymptotisch gegen einen konstanten Wert.

4.3 Scherkraft und Dilatanz

Um das abrupte Verhalten in den beiden Kurven Abb. 4.4 und Abb. 4.5 zu verstehen, ziehen wir die Abb. 4.7 und 4.8 heran. Hier sehen wir die Entwicklung einer Scherzelle zu vier aufeinander folgenden Zeitschritten. Der obere Teil von Abb. 4.7 zeigt das System

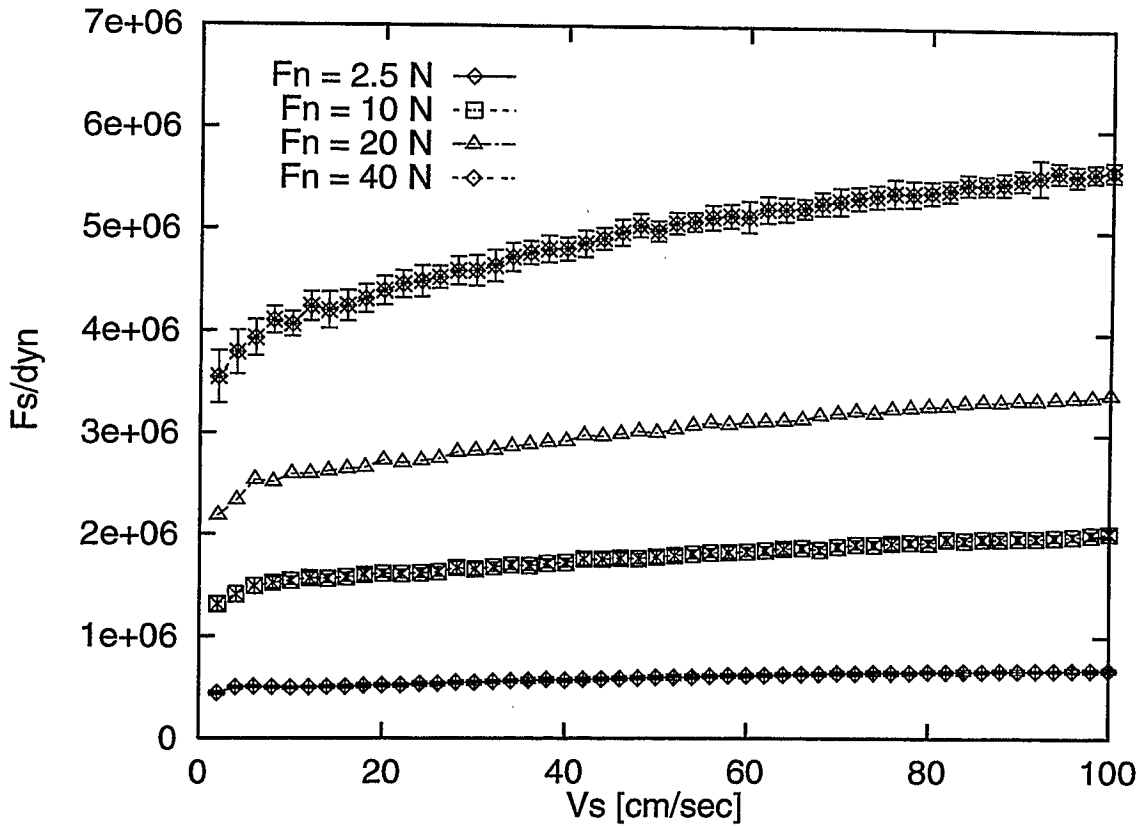


Abbildung 4.6: Scherkraft F_s gegen Schergeschwindigkeit v_s für vier verschiedene Normalkräfte F_N .

zu Beginn der Simulation. Die Teilchen sind dicht gepackt, und die Zellen haben sich noch nicht um ihre eigene Achse gedreht, was durch die schwarzen Pfeile angedeutet wird. (Pfeilrichtung nach rechts bedeutet keine Rotation, Pfeilrichtung nach oben würde z. B. eine Rotation um 90 Grad gegen den Uhrzeigersinn bedeuten). Im nächsten Bild ist der Zustand des Systems nach 0.1 s dargestellt. Man sieht, daß sich einige Lücken zwischen den Zellen gebildet haben, und demnach eine Ausdehnung stattgefunden hat. Im nächsten Bild zum Zeitpunkt 0.2 s und noch besser im letzten Bild der Sequenz zum Zeitpunkt 0.4 s erkennt man, daß Teilchen im mittleren Bereich der Scherzelle rotieren, während Teilchen nahe den Wänden sich nicht drehen. Es bildet sich also eine lokalisierte Scherzone. Der Verlauf dieser Scherzone parallel zu den Wänden ist durch die von uns verwendete Geometrie der Scherzelle vorgegeben.

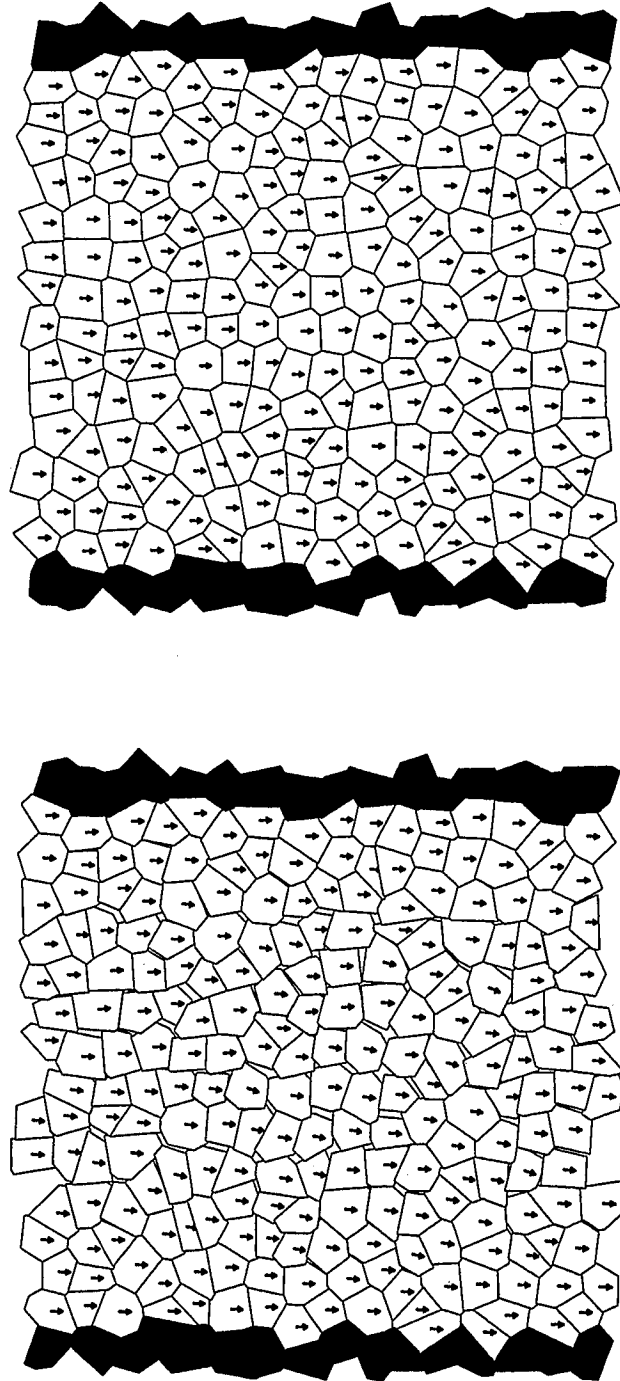


Abbildung 4.7: In diesem und im folgenden Bild werden vier Zeitschritte in der Entwicklung eines Scherexperimentes gezeigt, und zwar nach 0 s (links oben), 0.1 s (links unten), 0.2 s (rechts oben) und 0.4 s (rechts unten).

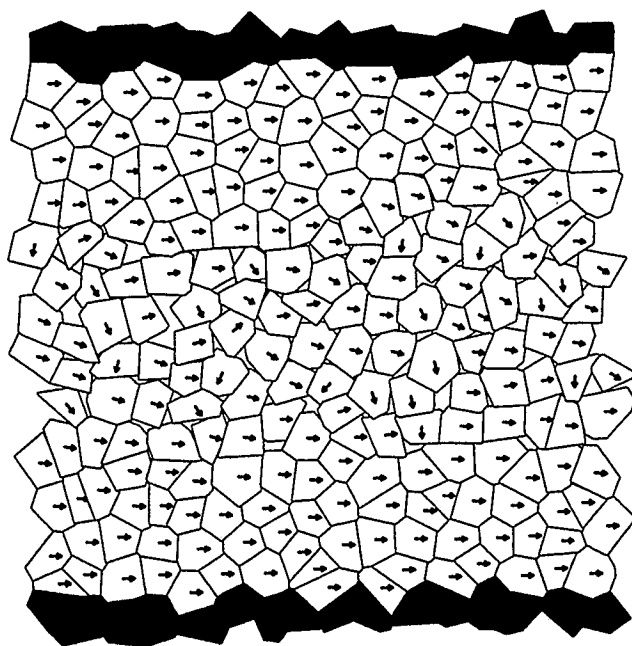
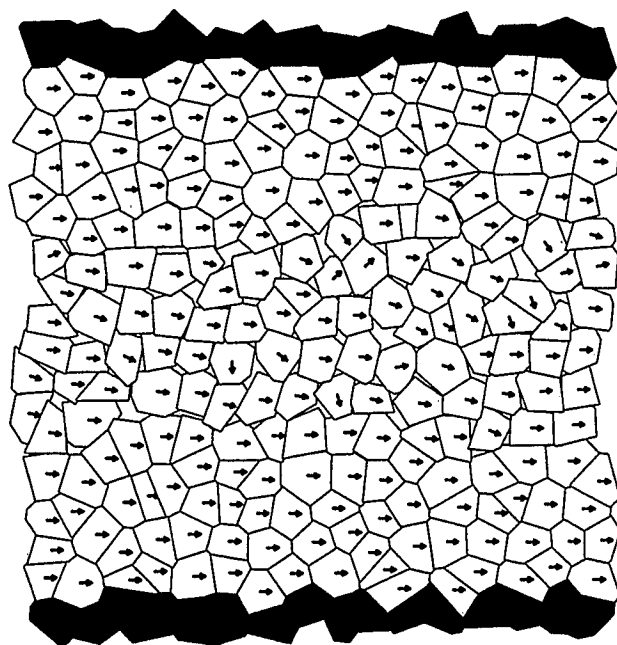


Abbildung 4.8: Die kleinen Pfeile zeigen an, um welchen Winkel sich die Zellen seit Beginn der Simulation gedreht haben. Ein nach rechts zeigender Pfeil bedeutet keine Drehung.

Wir können nun verstehen, wie es zu dem abrupten Verhalten der Scherkraft F_S und der Dilatanz D_Y zu Beginn der Simulation kommt. Zunächst sind die Teilchen dicht gepackt. Sobald man beginnt, das System zu scheren kann die dichte Packung nicht mehr aufrechterhalten werden. Die Teilchen haben nur die Möglichkeit auszuweichen, indem sich das System vertikal ausdehnt. Dazu ist eine sehr große Kraft aufzubringen, da sich die Teilchen gegenseitig im Weg sind. Ist eine gewisse Ausdehnung erreicht, können Teilchen in der Mitte des Systems sich drehen und die Ausdehnung des System geschieht wesentlich langsamer bis sie schließlich völlig zum Erliegen kommt. In unseren Simulationen haben wir stets beobachtet, daß die Scherkraft F_S mit der Zeit abnimmt, und es ergab sich stets ein Verhalten wie in Abb. 4.4 dargestellt.

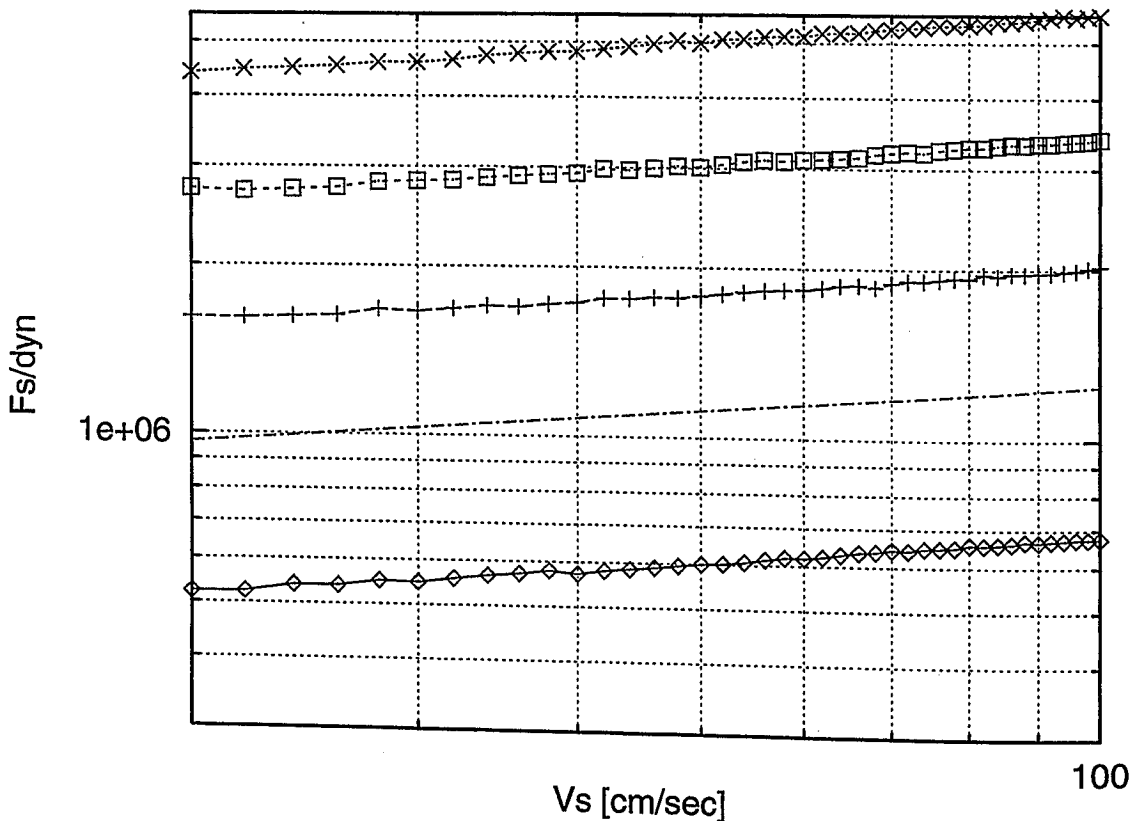


Abbildung 4.9: Scherkraft F_S gegen Schergeschwindigkeit v_S für vier verschiedene Normalkräfte F_N in doppelt-logarithmischer Darstellung. Zum Vergleich ist eine Kurve dargestellt, die einem Potenzgesetz mit dem Exponenten 0.16 entspricht.

Wir haben für vier verschiedene Normalkräfte F_N die Abhängigkeit der Scherkraft F_S von der Schergeschwindigkeit v_S gemessen. Das Resultat ist in Abb. 4.6 dargestellt. In all unseren Experimenten haben wir ein Elastizitätsmodul von 1MPa verwendet. Für die Coulomb Reibung μ wählten wir 0.5, γ_T war 100 s^{-1} , γ_N war ebenfalls 100 s^{-1} , was einem Restitutionskoeffizienten von $\epsilon \approx 0.8$ entspricht. In allen Fällen beobachten

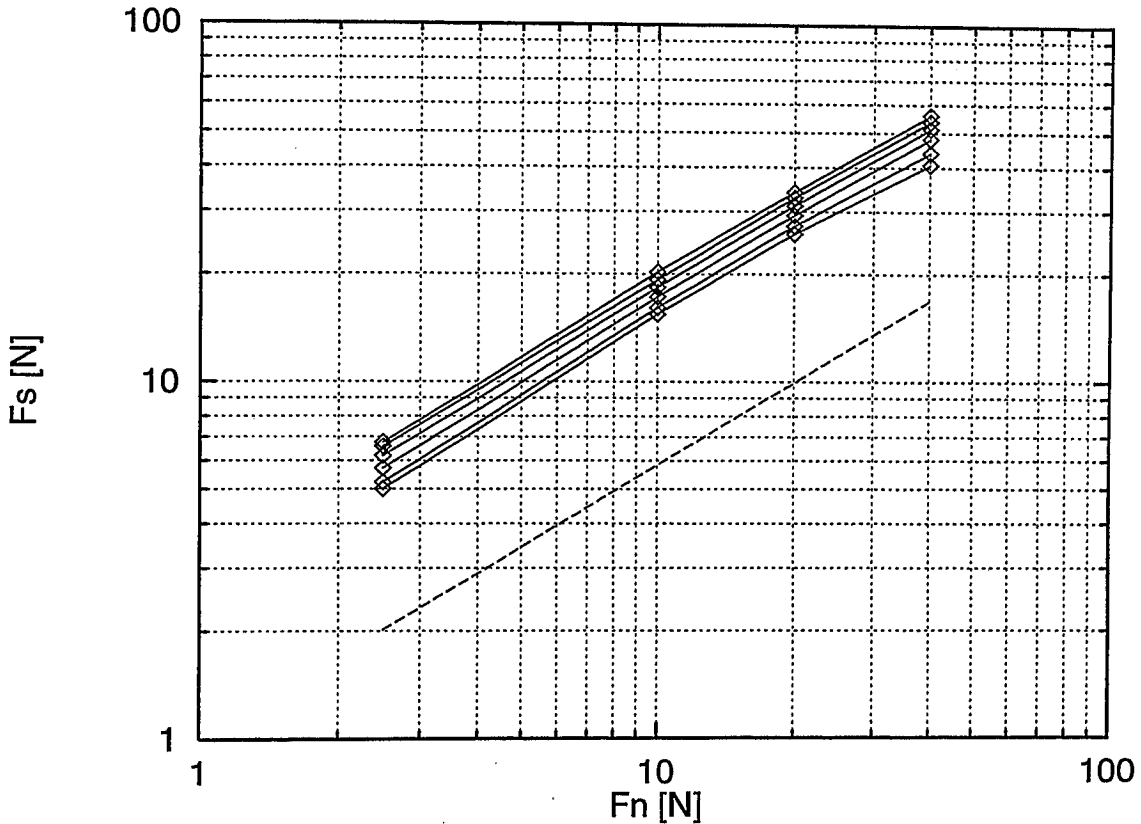


Abbildung 4.10: Scherkraft F_S gegen Normalkraft F_N für sechs verschiedene Schergeschwindigkeiten v_S (2, 4, 6, ..., 12 cm/s) in doppelt logarithmischer Darstellung. Eine Kurve, die einem Potenzgesetz mit dem Exponenten 0.76 entspricht, ist zum Vergleich ebenfalls dargestellt.

wir, daß die Scherkraft F_S mit der Schergeschwindigkeit v_S zunimmt, also Scherhärtung. Das Verhalten dieses Anstiegs ist sehr unterschiedlich für kleine Schergeschwindigkeiten bis etwa 15 cm/s und große Schergeschwindigkeiten über diesem Wert. Für den Teil der Kurve oberhalb von 15 cm/s kann das Verhalten der Kurve an ein Potenzgesetz angepaßt werden. Der Exponent dieses Potenzgesetzes ist für alle vier Kurven gleich und beträgt etwa (0.16 ± 0.01) . In Abb. 4.9 ist die Abhängigkeit der Scherkraft F_S von der Schergeschwindigkeit noch einmal, diesmal doppelt logarithmisch dargestellt. Zum Vergleich ist eine Kurve eingezeichnet, die ein Potenzgesetz mit dem Exponenten 0.16 darstellt. Man sieht, daß die Kurven parallel sind.

In Abb. 4.10 ist die Abhängigkeit der Scherkraft F_S von der Normalkraft F_N für sechs verschiedene Geschwindigkeiten v_S doppelt logarithmisch dargestellt. Die Kurven können sehr gut an ein Potenzgesetz mit dem Exponenten 0.76 angepaßt werden, wie man erkennt, wenn man diese mit der ebenfalls eingezeichneten Kurve vergleicht, die einem solchen Potenzgesetz entspricht. Würde das Coulomb-Gesetz gelten, wäre also die

Scherkraft F_S proportional der Normalkraft F_N , dann wäre der Exponent 1. Eine mögliche Erklärung für diesen schwächeren Anstieg werden wir später im Zusammenhang mit den Eigenschaften der Scherbänder vorschlagen.

Insgesamt hängt also die Scherkraft F_S folgendermaßen von der Normalkraft F_N und der Schergeschwindigkeit v_S ab:

$$F_S \propto v_S^\alpha F_N^\beta. \quad (4.1)$$

Für unsere Messungen, bei denen alle anderen Parameter konstant gehalten wurden, betragen diese Exponenten $\alpha = 0.16$ und $\beta = 0.76$.

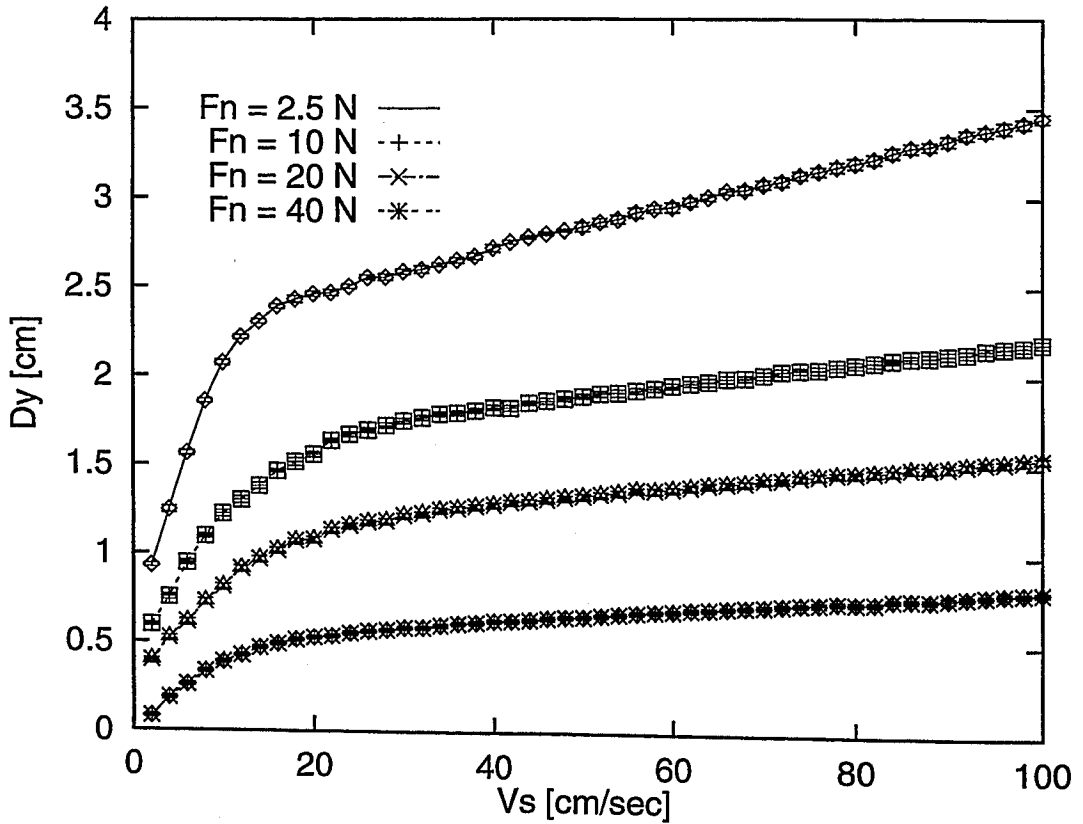


Abbildung 4.11: Dilatanz D_Y gegen Schergeschwindigkeit v_S für vier verschiedene Normalkräfte F_N .

Wir betrachten nun die Abhängigkeit der Dilatanz D_Y von der Schergeschwindigkeit v_S . Diese ist in Abb. 4.11 für die von uns untersuchten vier Normalkräfte F_N dargestellt. In allen vier Kurven steigt die Dilatanz D_Y streng monoton mit der Schergeschwindigkeit an. Auch hier sind wie in Abb. 4.6 zwei Bereiche mit unterschiedlichem Verhalten erkennbar und die Grenze zwischen diesen beiden Bereichen liegt auch hier wieder bei etwa 20 cm/s . In beiden Bereichen ist der Anstieg der Dilatanz mit der Schergeschwindigkeit linear. Im Bereich kleiner Schergeschwindigkeiten ist die Steigung der Geraden jedoch wesentlich größer als im Bereich hoher Schergeschwindigkeiten. In beiden Regimen sind

sowohl die Dilatanz als auch ihre Ableitung um so höher, je kleiner die Normalkraft F_N ist. Dies ist auch zu erwarten, da eine höhere Normalkraft dazu führt, daß sich das System nicht so leicht ausdehnen kann. Die Abhängigkeit der Ableitung der Dilatanz $\frac{dD_Y}{dv_S}$ von der Normalkraft F_N kann für den Bereich höherer Werte für die Schergeschwindigkeit an ein Potenzgesetz mit dem Exponenten -0.5 angepaßt werden, wie man der Abb. 4.12 entnehmen kann. Dort ist zum Vergleich auch eine Kurve aufgetragen, die einem Potenzgesetz mit dem Exponenten -0.5 entspricht. Die Steigung der Dilatanz mit der Schergeschwindigkeit ist also im Bereich höherer Schergeschwindigkeiten umgekehrt proportional zur Quadratwurzel der Normalkraft,

$$\frac{dD_Y}{dv_S} \propto \frac{1}{\sqrt{F_N}}. \quad (4.2)$$

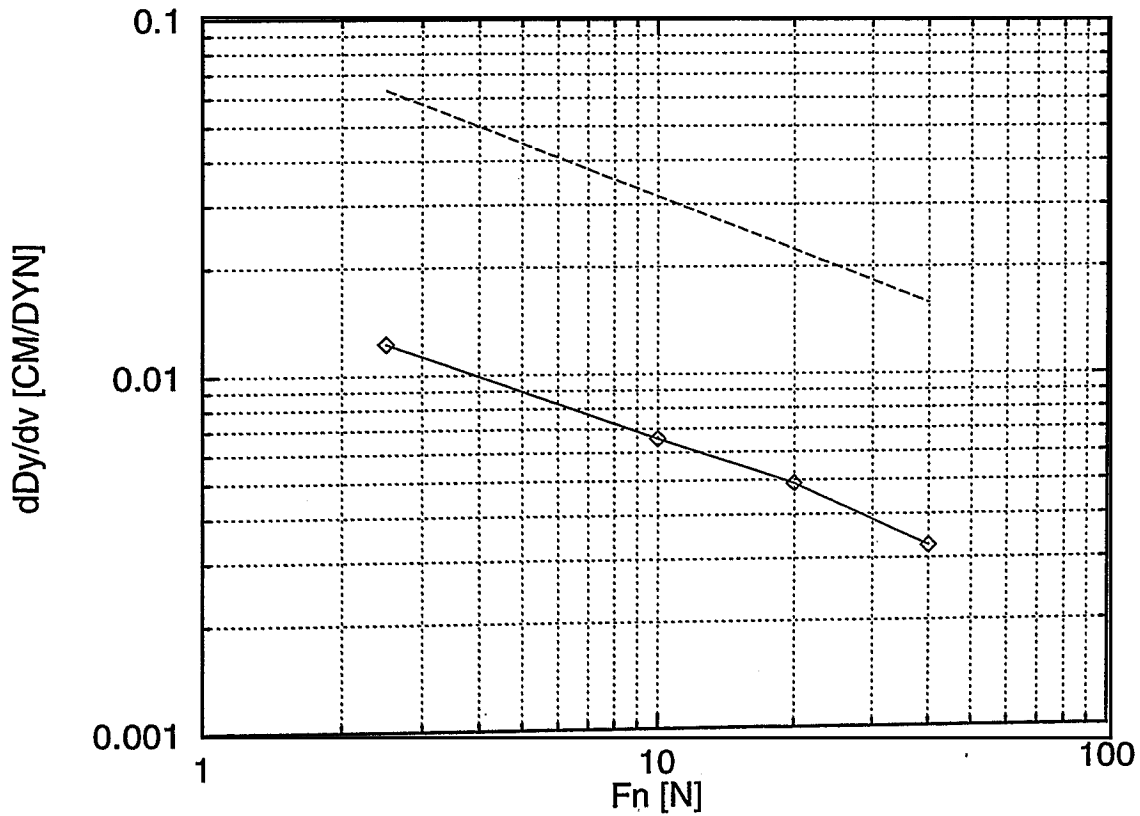


Abbildung 4.12: Die Steigung der Dilatanz D_Y mit der Schergeschwindigkeit v_S in doppelt-logarithmischer Darstellung. Diese Kurve kann gut an ein Gesetz der Form $F_N^{-1/2}$ angepaßt werden, wie man erkennt, wenn man sie mit der ebenfalls eingezeichneten gestrichelten Kurve vergleicht, die diesem Gesetz gehorcht.

Wir haben ebenfalls den Dilatanzwinkel untersucht. Darunter versteht man den Steigungswinkel des linearen Anteils in der Kurve der Dilatanz des Systems gegen die Sche-

runge ϵ . Anschaulich betrachtet ist dies der Winkel, den die Bahnkurve eines festgehaltenen Punktes auf der oberen Wand der Scherzelle mit der Horizontalen bildet. Zunächst haben wir überprüft, ob dieser Winkel eindeutig definiert ist. Für eine festgehaltene Normalkraft F_N wurden dazu für alle Schergeschwindigkeiten v_S die Kurven für die Dilatanz gegen die Scherung des Systems in ein Diagramm eingezeichnet, s. Abb. 4.13. Bei diesen Kurven kann man drei verschiedene Regime (I), (II) und (III) unterscheiden. Im ersten Teil der Kurve (I) ist die Dilatanz negativ, weil zu Beginn der Simulation das System ein wenig komprimiert wird, da die Normalkraft F_N auf das Innere der Zelle einzuwirken beginnt. Im mittleren Teil (II) der Kurve ist das Verhalten linear. In diesem Teil ist der Dilatanzwinkel zu messen. Im Teil (III) der Kurve saturiert die Dilatanz gegen einen festen Wert, der von der Schergeschwindigkeit v_S abhängt. In den Teilen (I) und (II) fallen alle Kurven zusammen, so daß der Dilatanzwinkel also eindeutig bestimmt ist.

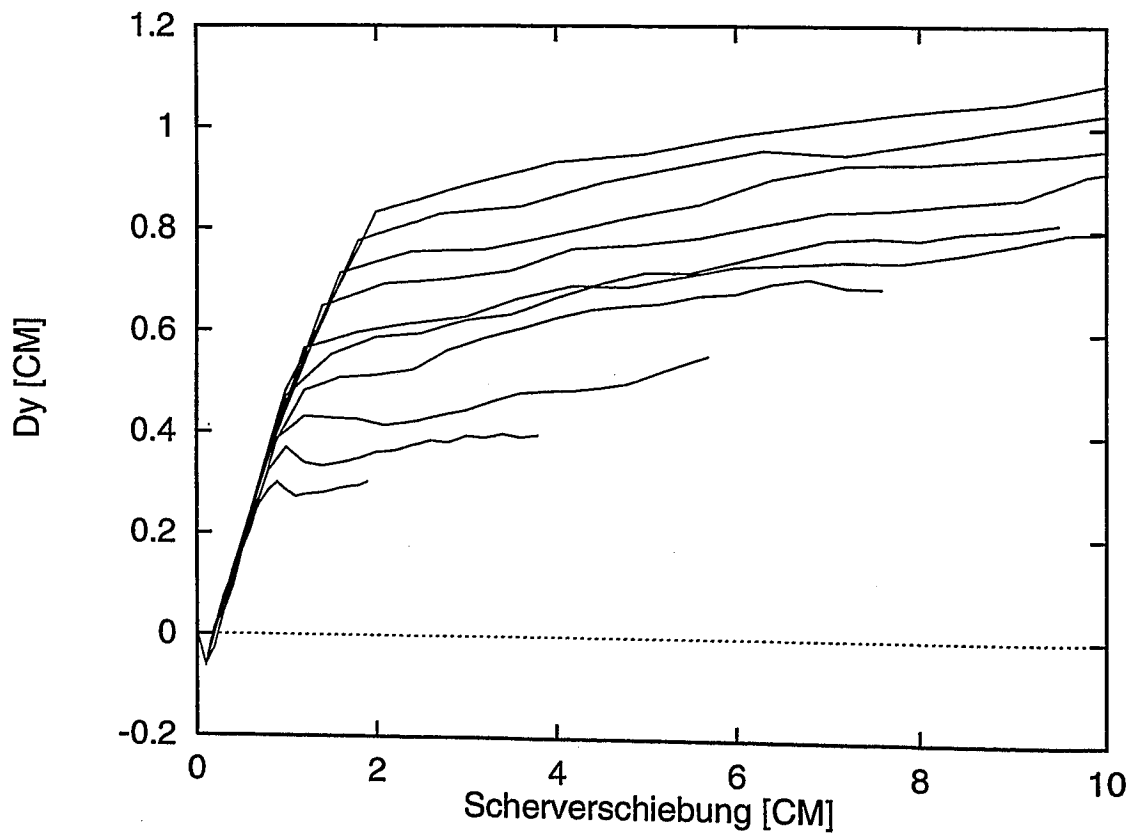


Abbildung 4.13: Die Kurven für die Dilatanz D_Y gegen die Scherverschiebung ϵ für verschiedene Schergeschwindigkeiten v_S . Der negative Bereich (I) und der lineare Bereich (II) der Kurven fallen zusammen, so daß der Dilatanzwinkel eindeutig bestimmt ist.

In Abb. 4.14 ist die Dilatanz gegen die Scherung für die kleinste von uns gemessene Schergeschwindigkeit von 1 cm/s für die vier verschiedenen Normalkräfte F_N aufgetragen. Wie man sieht, bleibt die Dilatanz für die größte der untersuchten Scherkräfte von

40 N sehr lange negativ. Für die kleinste Normalkraft von 2.5 N ist der negative Bereich (I) der Kurve hier nicht auflösbar. Wir bestimmten den Dilatanzwinkel für alle vier Normalkräfte und erhielten folgende Werte:

F_N/N	Druck/MPa	Steigung	Winkel/Grad
2.5	0.016	0.8545	40.5
10	0.062	0.6164	31.6
20	0.125	0.4783	25.5
40	0.250	0.0237	1.1

Die Daten zeigen eine signifikante Abhängigkeit von der Normalkraft F_N . Dies wurde auch in den Computersimulationen von Handley [66] gefunden, jedoch dort für wesentlich härtere Materialien. Im Vergleich zu Experimenten mit Scherzellen sind unsere Werte jedoch sehr groß. Dies erklären wir damit, daß wir zu Beginn der Simulation ein vollkommen dichtgepacktes System vorliegen haben. Bei realen granularen Materialien ist dies nicht der Fall. Es wäre interessant, die Simulationen auch für solche, weniger dicht gepackten Systeme durchzuführen. Dies könnte man z. B. erreichen, indem man an einigen Stellen im Voronoi-Gitter Zellen entfernt. Eine andere Methode bestünde darin, die Zellen um einen gewissen Betrag zu kontrahieren. Wir erwarten, daß bei solchen Messungen dann kleinere Werte für die Dilatanzwinkel gefunden werden.

4.4 Scherbänder

Um lokalisierte Rotationen von Teilchen zu untersuchen, unterteilen wir die Scherzelle in horizontale Streifen konstanter Dicke, die parallel zur oberen und unteren Wand orientiert sind. Es wird überprüft, welche Zellen in welche Streifen gehören. Dann wird für jeden Streifen die durchschnittliche Rotation und die durchschnittliche Schergeschwindigkeit der darin enthaltenen Teilchen bestimmt. In Abb. 4.18 ist für eine Normalkraft F_N von 10 N und eine Schergeschwindigkeit von 10 cm/s die durchschnittliche Schergeschwindigkeit der Zellen gegen die normierte Tiefe d der Scherzelle aufgetragen. Man erkennt am abrupten Übergang der Schergeschwindigkeit von -10 auf +10 cm/s sehr deutlich die lokalisierte Zone.

In Abb. 4.15 ist für die Normalkraft F_N von 10 N und vier verschiedene Schergeschwindigkeiten die Gesamtrotation in rad gegen die normalisierte Tiefe d innerhalb der Scherzelle aufgetragen. Für die beiden kleinsten Schergeschwindigkeiten von 2 und 10 cm/s stellt man fest, daß es ein Maximum für die Rotation in der Mitte der Zelle gibt und daß diese zu den Wänden hin auf 0 abfällt. Es gibt somit im mittleren Bereich eine lokalisierte Zone, wo die Zellen rotieren. Dies ist auch in der Sequenz der Zeitentwicklung eines Scherexperiments in Abb. 4.7 und Abb. 4.8 sehr schön zu erkennen. In Abb. 4.15 haben wir wieder den Mittelwert über 32 Simulationsläufe verwendet.

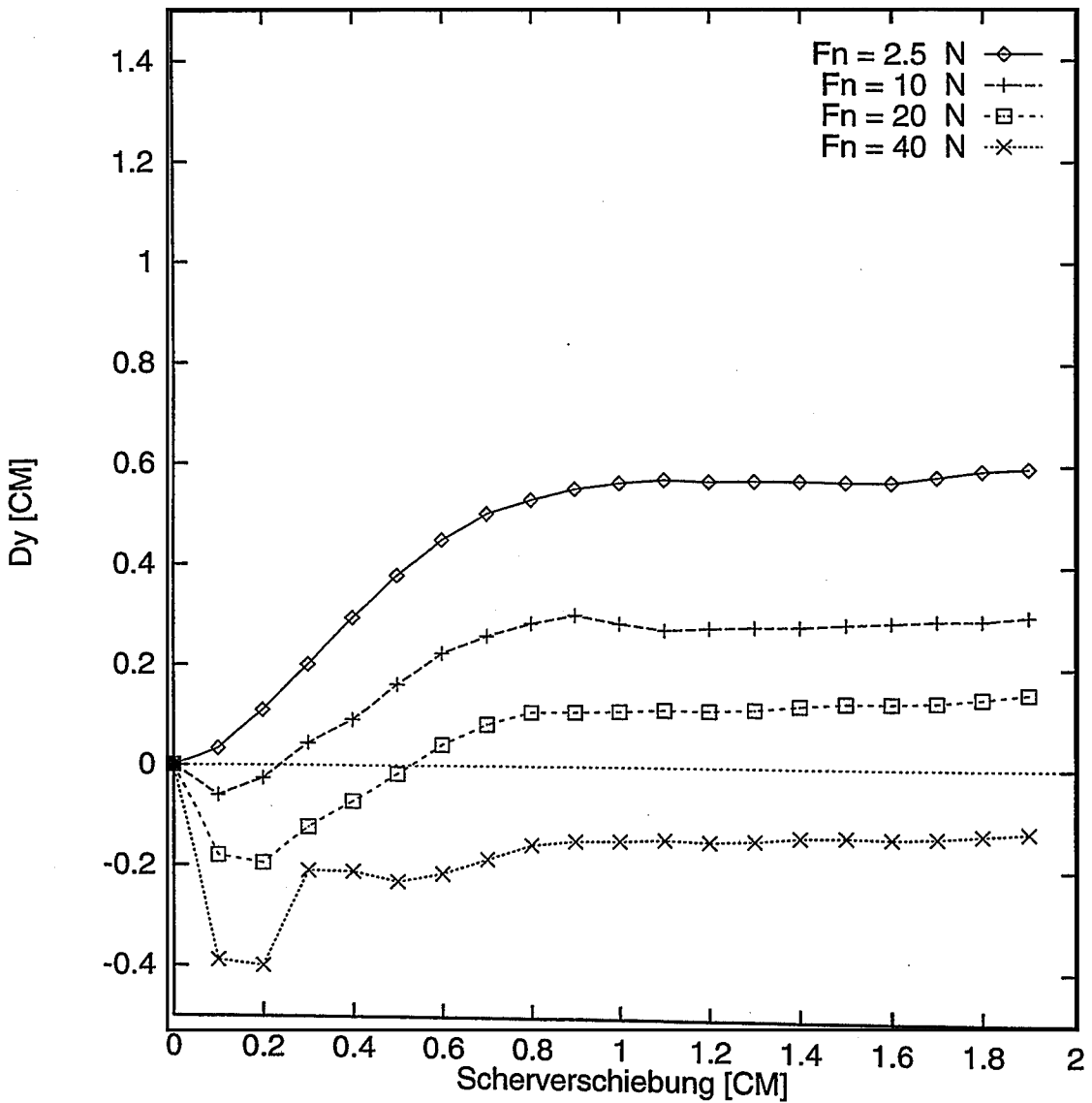


Abbildung 4.14: Dilatanz D_Y in cm gegen Scherverschiebung ϵ in cm für die vier untersuchten Normalkräfte und eine Schergeschwindigkeit von 2 cm/s. Man unterscheidet drei Bereiche mit unterschiedlichem Verhalten. Im Bereich (I) wird die Dilatanz aufgrund der Kompression zunächst negativ. Dem Bereich (II) mit dem linearen Verhalten ist der Dilatanzwinkel zu entnehmen und im Bereich (III) saturiert die Dilatanz gegen einen festen Wert.

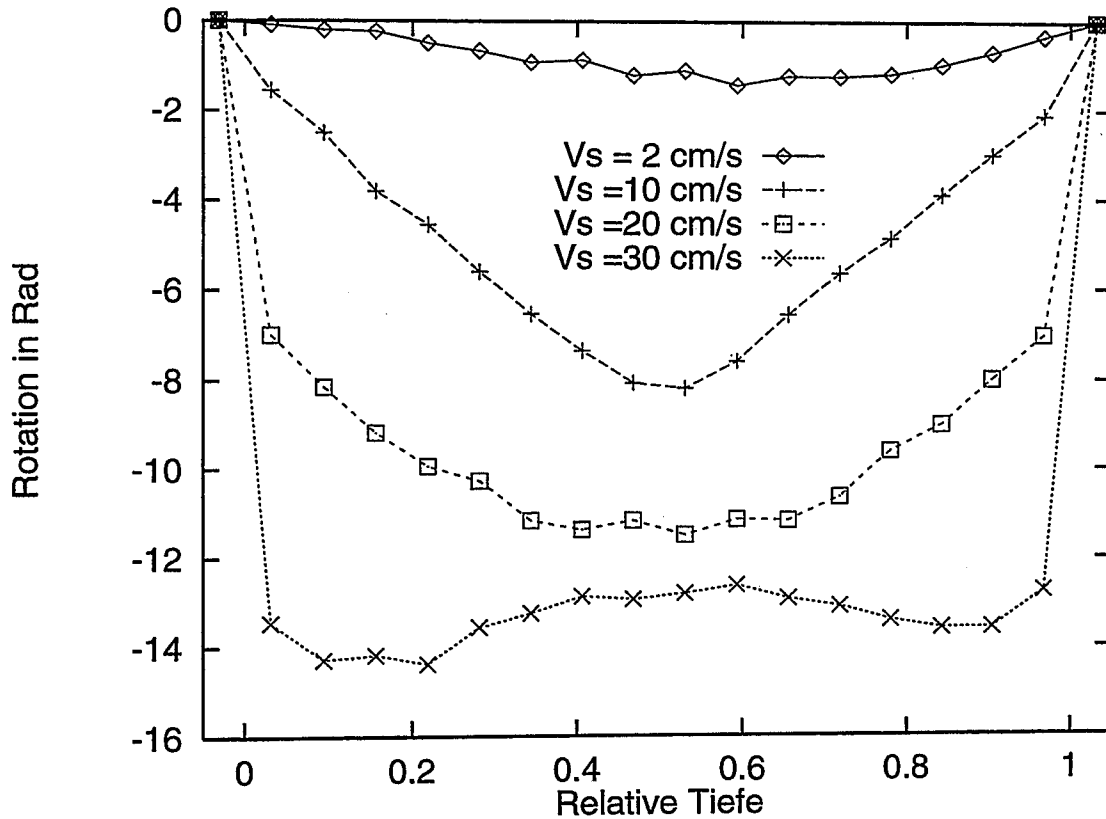


Abbildung 4.15: Die Gesamtrotation von Teilchen in Streifen konstanter Dicke, die parallel zur oberen und unteren Wand orientiert sind für eine Normalkraft F_N von 10 N und für vier verschiedene Schergeschwindigkeiten v_s . Man erkennt, daß es für kleine Schergeschwindigkeiten Lokalisation von Rotationen gibt, während diese bei großen Schergeschwindigkeiten langsam verschwindet.

Dadurch wird das Phänomen der Lokalisation etwas ausgeschmied, da das Scherband nicht immer genau symmetrisch in der Mitte der Zelle positioniert ist. In der Sequenz in Abb. 4.7 und Abb. 4.8 sieht man, daß die Zellen, die sich nicht im Scherband befinden, gar nicht rotieren. Die Zellen nahe der oberen und der unteren Wand bilden jeweils einen festen Block, so daß man aus der Lokalisation in Bezug auf Rotationen auch auf eine Lokalisation in Bezug auf lokale Scherungen schließen kann. Scherbewegung findet daher nur in der lokalisierten Scherzone statt.

Wie man in der Abb. 4.15 erkennen kann, findet Teilchenrotation bei höheren Schergeschwindigkeiten v_s von 20 cm/s und 30 cm/s auch in der Nähe der oberen und unteren Wände statt und das Phänomen der Lokalisation verschwindet langsam. Bei der Schergeschwindigkeit von 30 cm/s hat es sich sogar etwas umgekehrt und die Rotation der Zellen im Zentrum ist etwas kleiner als in der Nähe der Wände. In Abb. 4.16 ist die Scherbandbreite W für die vier untersuchten Normalkräfte F_N gegen die Schergeschwindigkeit v_s aufgetragen. Die Scherbandbreite W definierten wir dabei als die Breite des

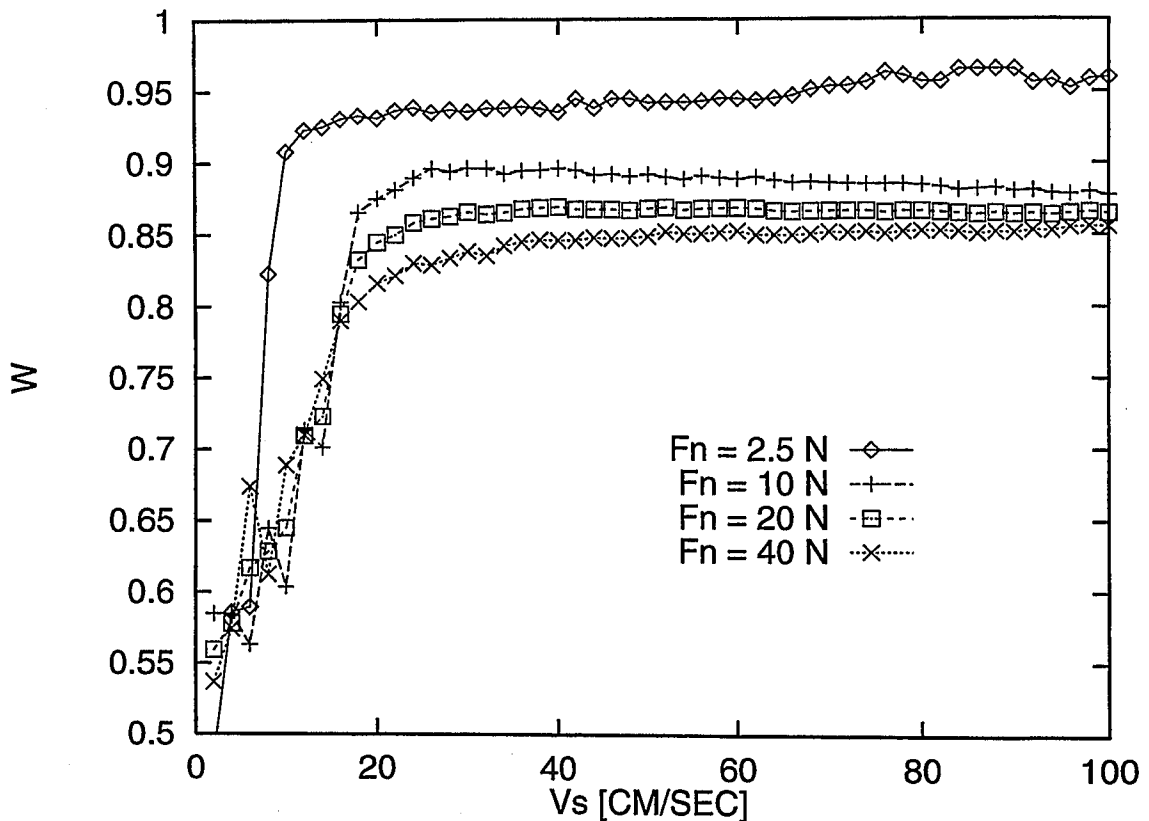


Abbildung 4.16: Die relative Breite W der Scherbänder ist für die vier untersuchten Normalkräfte F_N gegen die Schergeschwindigkeit v_s aufgetragen. Es gibt einen Übergang bei einer Schergeschwindigkeit von etwa 15 cm/s, oberhalb derer das Lokalisationsphänomen fast verschwindet.

Bereiches, in dem die lokale Rotation der Zellen höher ist als der halbe Betrag der lokalen Rotation im Zentrum der Scherzelle, wo diese maximal ist. Man sieht, daß diese Breite bei einem Wert von etwa 15 cm/s sprunghaft ansteigt und dann fast die gesamte Scherzelle überspannt. Der Sättigungswert der Scherbandbreite ist für die höchste Normalkraft F_N am kleinsten und für die kleinste Normalkraft am höchsten.

Wenn man die Abhängigkeit der Dilatanz D_Y von der Schergeschwindigkeit v_s mit der Abhängigkeit der Scherbandweite W von v_s vergleicht, stellt man fest, daß der abrupte Übergang von einer hohen Steigung der Dilatanz mit der Schergeschwindigkeit zu einer geringeren Steigung etwa bei der selben Schergeschwindigkeit stattfindet, wo die Scherzone sich abrupt über fast das gesamte System ausbreitet. Dies kann man verstehen, wenn man bedenkt, daß die Teilchen aufgrund ihrer nichtsphärischen Form das System dilatieren müssen, um rotieren zu können. Deswegen muß der Anstieg der Dilatanz mit der Schergeschwindigkeit groß sein, bis nahezu alle Zellen an der Rotation teilnehmen. Bei weiterer Steigerung der Schergeschwindigkeit ist der Anstieg der

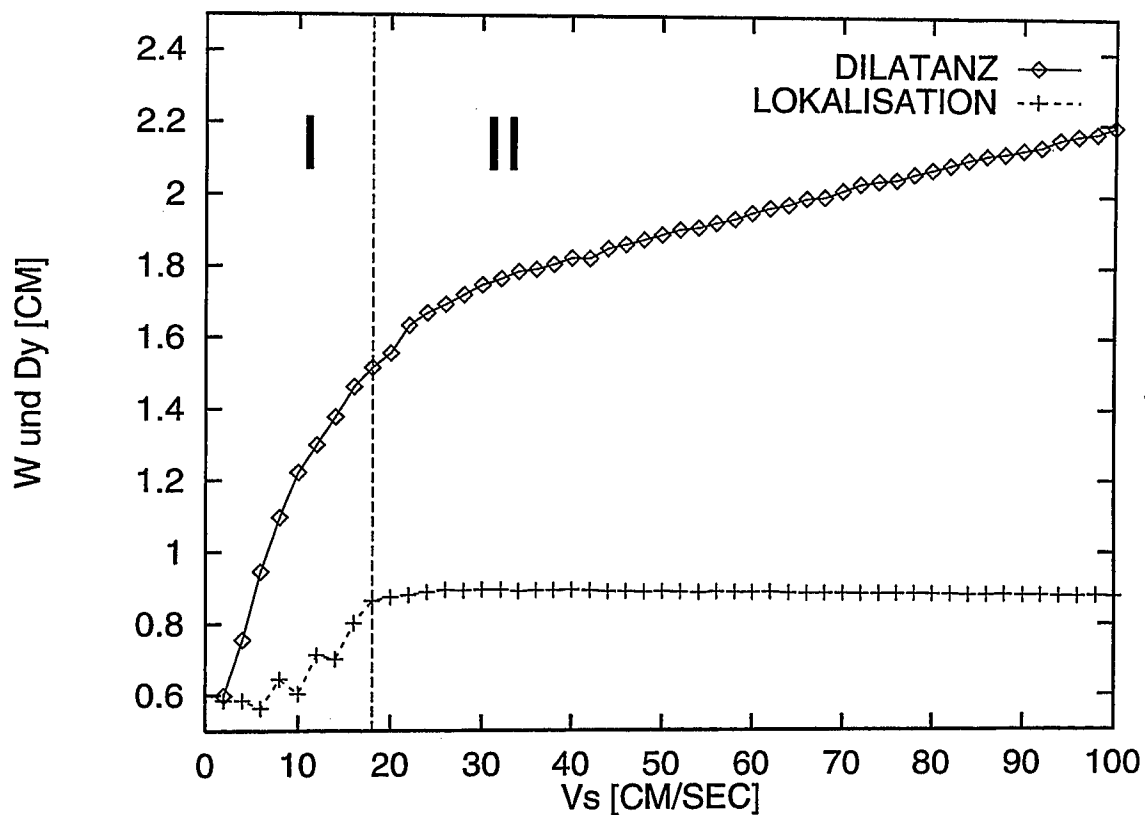


Abbildung 4.17: In dieser Kurve wird das Verhalten der Dilatanz gegen die Schergeschwindigkeit verglichen mit dem Verhalten der Breite des Scherbandes gegen die Schergeschwindigkeit. Die starke Veränderung in der Steigung der Dilatanzkurve findet etwa an der Stelle statt, wo das Scherband sich über das gesamte System ausbreitet.

Dilatanz dann weitaus geringer.

In [131, 132] wurde mit Hilfe einer kontinuumsmechanischen Simulation auf Basis der FLAC Technik gefunden, daß sich Scherbänder spontan bilden, sobald ein System gesichert wird. Sehr große Systeme wurden untersucht und generell wurde eine große Anzahl von Scherbändern gefunden. Die mikroskopischen Aspekte eines aus unregelmäßig geformten Teilchen bestehenden Systems können aber von diesem Modell nicht richtig wiedergegeben werden. Unser Modell ist in gewisser Weise komplementär zu diesem, da zwar nur kleine Systeme untersucht werden können, dafür aber die geometrisch bedingte Wechselwirkung zwischen den Teilchen besser wiedergegeben wird.

In Abb. 4.16 erkennt man, daß die Lokalisierung der Scherzonen auch für große Schergeschwindigkeiten nicht vollkommen verschwindet, sondern in Abhängigkeit von der Normalkraft F_N gegen einen konstanten Wert konvergiert. Je höher die Normalkraft F_N ist, desto kleiner ist die Breite der Scherzone. Für die höchste von uns betrachtete Normalkraft von 40N ist die Breite der Scherzone immerhin auf etwa 0.84 zurückge-

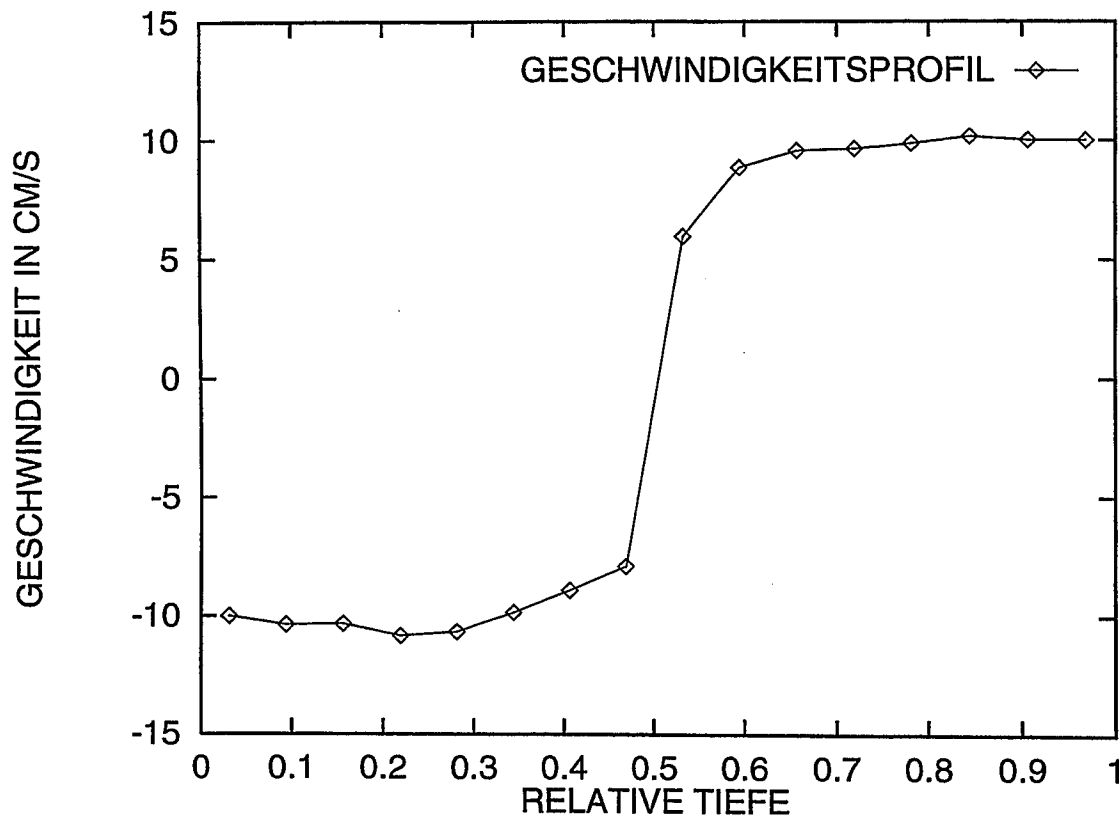


Abbildung 4.18: Diese Kurve zeigt das Profil der Schergeschwindigkeiten gegen die normierte Tiefe in der Scherzelle, 0 kennzeichnet den unteren, 1 den oberen Rand. Man erkennt hier sehr deutlich die lokalisierte Scherzone am abrupten Übergang der Schergeschwindigkeit von -10 auf +10 cm/s. Die Normalkraft in dieser Simulation beträgt 10N.

gangen. Von Karman [93] fand 1911 bei Scherexperimenten mit Felsgestein unter sehr hohem Druck, daß dieses plastisch wird und anfängt zu fließen. Dabei bildete sich eine lokalisierte Scherzone. Unsere Beobachtung, daß die Lokalisierung mit zunehmender Normalkraft größer wird, deutet in die gleiche Richtung und wird auch von experimentellen Arbeiten mit Sand gestützt [48, 155, 162, 174]. Wir kommen auf dieses interessante Phänomen im nächsten Kapitel noch einmal zurück.

4.5 Zusammenfassung

Wir haben in numerischen Scherzellenexperimenten untersucht, wie die Dilatanz und die Scherkraft von der Schergeschwindigkeit und der Normalkraft abhängen. Ebenso haben wir den Dilatanzwinkel gemessen. Die von uns gemessenen Dilatanzwinkel sind verglichen mit experimentellen Werten zu groß, was auf die dichte Packung unserer Teilchen

skzuführen ist. Die Scherzellen in unseren Simulationen bildeten spontan lokalisierherbänder aus. Diese wurden auch experimentell [116], in anderen Simulationen [132] und in theoretischen Arbeiten gefunden [172, 173, 174]. Eine interessante iterung der Simulation bestünde darin, das Modell auf drei Dimensionen zu verallinern. Wir werden auf die damit verbundenen Schwierigkeiten im Kapitel 9 noch einmal eingehen.

Kapitel 5

Simulation von Erdbeben

5.1 Einleitung

Wir wenden unser Modell in diesem Kapitel auf ein geophysikalisches Thema an, nämlich auf die Bewegung tektonischer Platten. Die Lithosphäre, die äußerste etwa 100 km dicke Schicht des Erdmantels, besteht aus einer Anzahl von etwa 20 ineinander passenden Bruchstücken, den tektonischen Platten. Diese bewegen sich sehr langsam, mit Geschwindigkeiten in der Größenordnung von Zentimetern pro Jahr, relativ zueinander. Der Antrieb dieser Bewegung wird im Erdinnern vermutet, welches aus drei Schichten besteht, dem flüssigen Erdkern, dem festen inneren Mantel und dem flüssigen äußeren Erdmantel. An der Grenzfläche zwischen dem inneren und dem äußeren Erdmantel wird bedingt durch den radioaktiven Zerfall von Materialien ständig Wärmeenergie frei, die durch Konvektion nach außen transportiert wird. Die hierbei wirkenden Konvektionsrollen, deren Geschwindigkeit ebenfalls im Bereich von einigen Zentimetern bis Dezimetern pro Jahr liegt, bewegen die darüber liegenden tektonischen Platten.

Es gibt drei Möglichkeiten dafür, wie die Grenze zwischen zwei tektonischen Platten aussehen kann. Die erste besteht darin, daß sie sich senkrecht zu ihrer Begrenzungslinie vertikal voneinander wegbewegen, wie das bei den mittelozeanischen Rücken der Fall ist. Dabei wird Material aus dem Erdinnern nach außen transportiert. Die zweite Möglichkeit besteht darin, daß sich zwei Platten aufeinander zu bewegen oder daß die eine Platte auf die andere aufläuft. Ein Beispiel bildet die Indische Platte, die in die Asiatische Platte eindringt. Sie schiebt sich dabei unter diese, wobei Material langsam in das Erdinnere absinkt. Sie hat dabei in einem Zeitraum, der mit etwa 100 Millionen Jahren angesetzt werden kann, das Himalaya-Gebirge aufgefaltet [127, 178, 179]. Sobald die Relativgeschwindigkeit zwischen den beiden Platten zum Erliegen kommt, wird dieses langsam wieder absinken, da der durch das Gebirge verursachte Gravitationsdruck nicht mehr ausgeglichen wird.

Unser Modell ist nicht in der Lage, diese beiden ersten möglichen Relativbewegungen tektonischer Platten zu simulieren, da es die dritte Dimension nicht berücksichtigt und Prozesse, bei denen Material aus dem Erdinnern aufsteigt oder in das Erdinnere

absinkt, nicht wiedergegeben werden können. Jedoch ist es möglich, die dritte mögliche Grenzfläche zwischen zwei tektonischen Platten zu modellieren, bei der diese eine tangentielle relative Geschwindigkeit zueinander haben. Ein berühmtes Beispiel für diesen Fall ist die San-Andreas-Verwerfung, wo die Pazifische und die Nordamerikanische Platte sich scherend gegeneinander bewegen, die Pazifische in nordwestlicher und die nordamerikanische in südöstlicher Richtung. Wir interpretieren dies als eine Scherzelle auf geophysikalischer Skala und können direkt das Modell aus dem letzten Kapitel darauf anwenden.

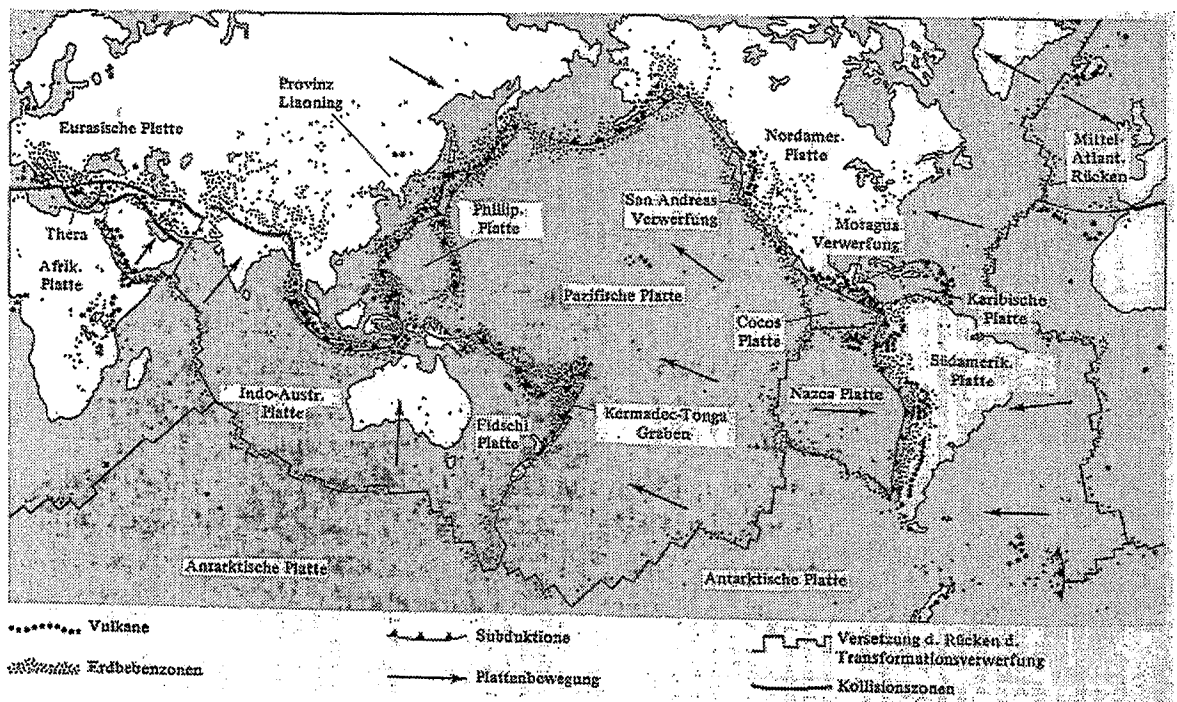


Abbildung 5.1: Weltkarte, die den Zusammenhang zwischen größeren tektonischen Platten, Erdbeben (der jüngsten Gegenwart) und Vulkanen aufzeigt. Epizentren sind durch kleine, Vulkane durch große Punkte angedeutet. Entnommen aus Bolt 1984.

Die Bewegung der tektonischen Platten hängt eng mit der Entstehung der Erdbeben zusammen, was eindrucksvoll bestätigt wird, wenn man einen Blick auf eine Karte wirft, in der die Berührungslinien der tektonischen Platten und die Orte, an denen die stärksten Erdbeben, die in einem gewissen Zeitraum stattgefunden haben, aufgetragen sind. Diese sind fast ausschließlich entlang dieser Grenzen lokalisiert. Der Grund hierfür ist darin zu suchen, daß die tektonischen Platten trotz ihrer immensen lateralen Ausdehnung und der Tatsache daß sie relativ dünn sind, als starre Platten fungieren und sich kohärent als Ganzes bewegen. An den Rändern, die sehr rauh sind, verhaken sie sich ineinander. Dadurch wird langsam eine Spannung aufgebaut, die bei Erreichen einer bestimmten Schwelle eine ruckartige Bewegung verursacht, die sich als Erdbeben

manifestiert. Die Entstehung von Erdbeben ist eng verwandt mit der Schallabstrahlung bei Bruchvorgängen in Metallen [171].

5.2 Das Modell

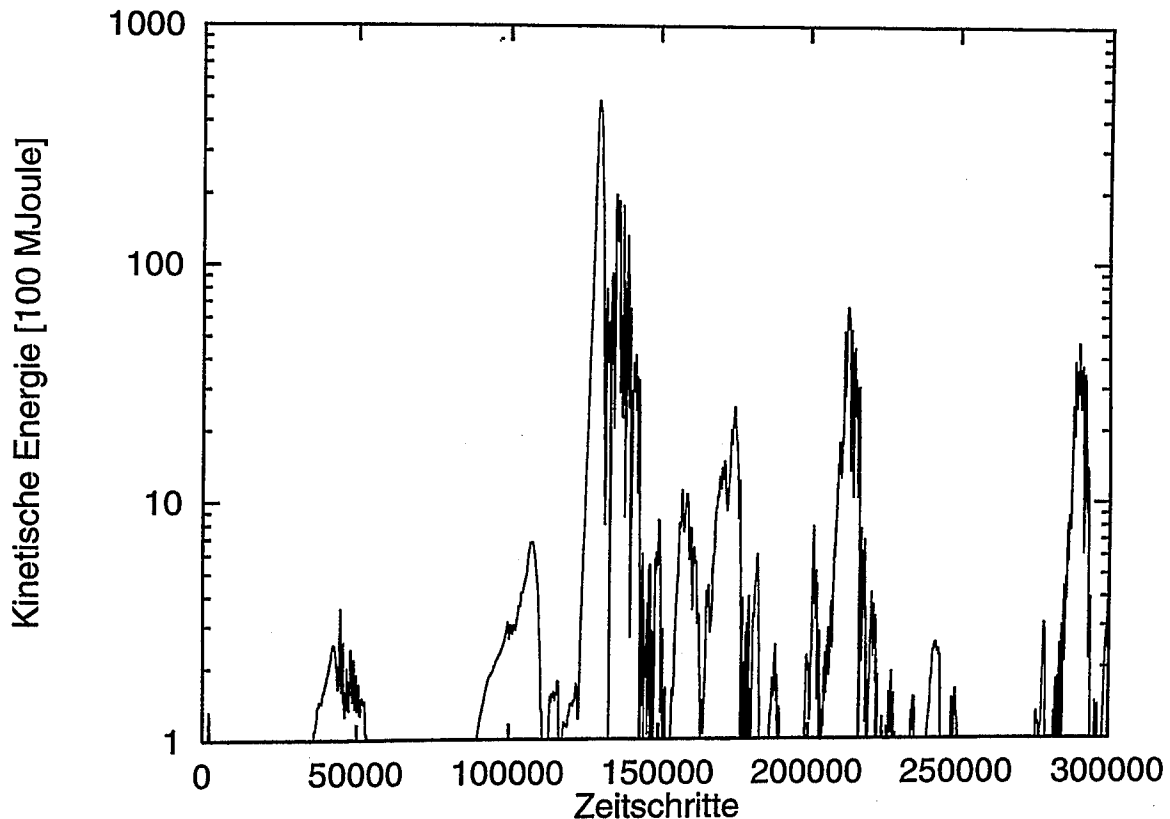


Abbildung 5.2: *Kinetische Energie einer einzelnen Zelle logarithmisch gegen die Zeit aufgetragen. Wir interpretieren die bei einem Erdbeben freigesetzte Energie als die Energiedifferenz zwischen einem lokalen Minimum in der Energie und dem darauffolgenden Maximum.*

Unser Modell zur Simulation von Scherzellen ist sehr gut geeignet, den dritten Typ der Grenzfläche zwischen zwei tektonischen Platten zu modellieren, nämlich die tangentielle Scherbewegung, wie sie z. B. der San-Andreas-Verwerfung zugrunde liegt. Die Scherzelle stellt dabei einen Ausschnitt der Berührungszone zweier tektonischer Platten dar. Dabei werden die Ränder der Zelle durch die beiden aneinandergrenzenden Platten geschert. Die Polygone im inneren der Scherzelle stellen das Felsmaterial dar, das sich in der Grenzregion befindet. Wir betrachten Systeme aus 32 mal 32 und 64 mal 64 Zellen, wobei diese Zellen hier die Größenordnung von einem Kilometer haben (die Größe der

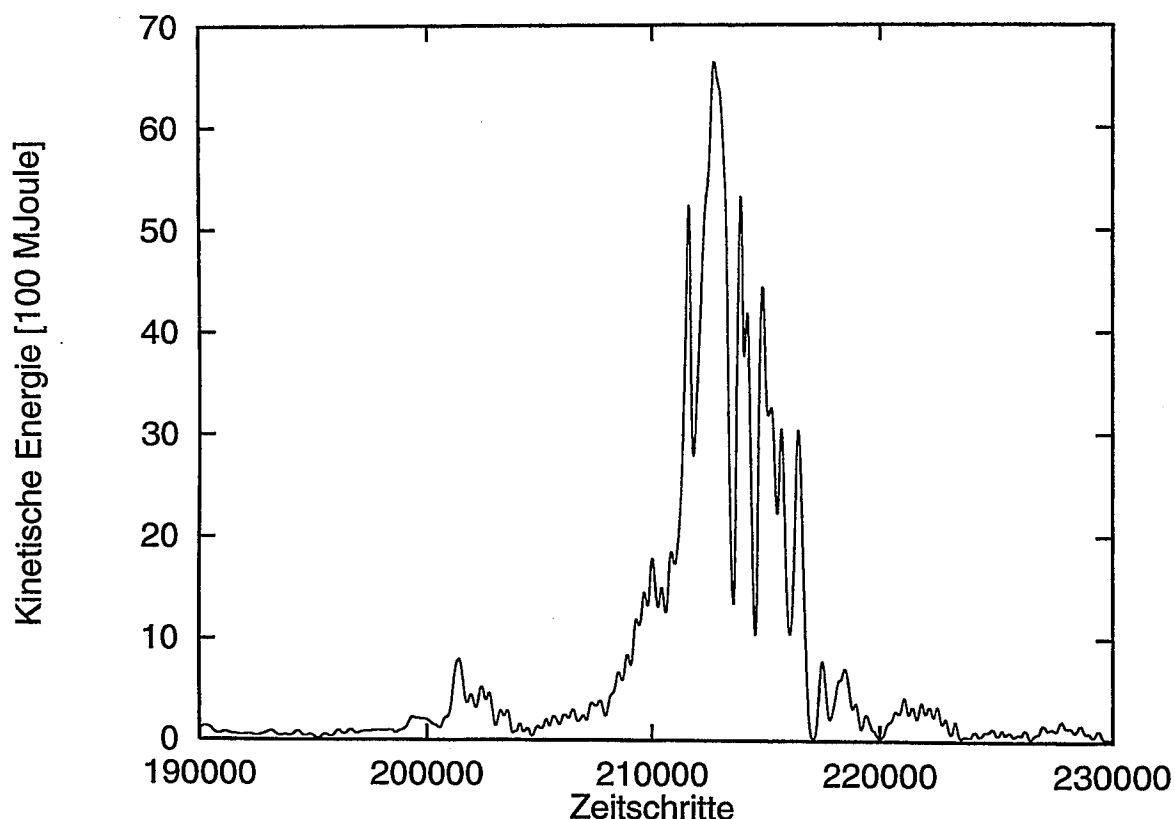


Abbildung 5.3: Ein Ausschnitt aus der Abbildung 5.2, diesmal in linearer Darstellung. Bei allen Erdbeben sieht man ein reiches Spektrum an Nachbeben. Das Erdbeben nach etwa 215000 Zeitschritten kündigt sich vorher durch kleinere Vorbeben an.

Zellen des zugrunde liegenden Quadratgitters ist 1 km). Wir betrachten also eine quadratische Fläche mit einer Ausdehnung von 320 bzw. 640 km. Im Unterschied zu den Schersimulationen im letzten Kapitel sollen die beiden Ränder, an denen das System gesichert wird, jetzt nicht vertikal beweglich sein, d. h. die Dilatanz ist null. Dies trägt der Tatsache Rechnung, das sich das geophysikalische System nicht ausdehnen kann. Abgesehen von dieser kleinen Änderung entspricht das Modell vollkommen dem im letzten Kapitel benutzten. Wir wählten folgende Simulationsparameter:

Parameter	Symbol	Einheiten	Wert
Dichte	ρ	$\frac{g}{cm^3}$	5
Young Modul	Y	$\frac{dyn}{cm^2}$	10^{11}
Dissipation	γ_N	s^{-1}	0.1
Gleitreibung	γ_T	s^{-1}	0.1
Haftreibung	μ	1	0.5
Zeitschritt	dt	s	0.005

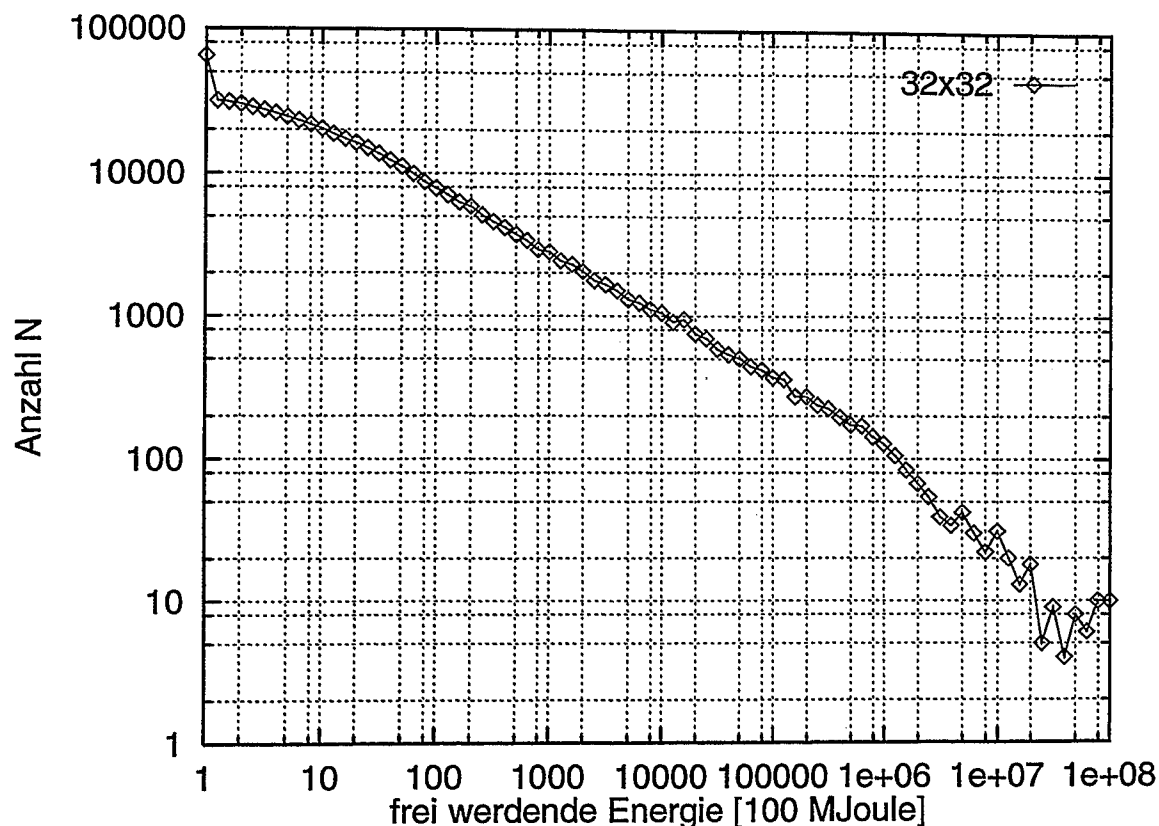


Abbildung 5.4: Die Häufigkeitsverteilung von Erdbeben verschiedener Energie in einer doppelt logarithmischen Darstellung. Die Kurve folgt über fast fünf Größenordnungen einem Potenzgesetz mit einem Exponenten von etwa (-1.4) .

Wir führten insgesamt $2 \cdot 10^6$ Zeitschritte durch, wobei die Messungen in den letzten 10^6 Zeitschritten durchgeführt wurden, nachdem sich ein quasistatischer Zustand eingestellt hatte. Als Erdbeben wollen wir ruckartige Verschiebungen von Zellen verstehen, jedes lokale Energieminimum definiert also den Beginn und das darauffolgende lokale Energiemaximum das Ende eines Erdbebens. Die Energiedifferenz zwischen diesen beiden Werten interpretieren wir als die durch das Erdbeben freigesetzte Energie. Diese ist ein Maß für die Stärke des Bebens. Der Dissipationsparameter γ_N betrug $0.1 \frac{1}{s}$, was bedeutet, daß die kinetische Energie nach einem Erdbeben nach etwa 10 s auf einen Bruchteil von $1/e$ gesunken ist. Der Wert liegt etwa im Bereich der bei Erdbeben gemessenen Werte. Das von uns gewählte Elastizitätsmodul von 10 GPa liegt etwa im Bereich von Gestein und stellt damit einen realistischen Wert dar. Unsere Schergeschwindigkeit beträgt 100 cm/s und ist damit im Vergleich zur Schergeschwindigkeit bei tektonischen Platten um ungefähr neun Größenordnungen zu hoch. Der Zeitschritt bei unseren Simulationen beträgt 0.005 s, die $2 \cdot 10^6$ Simulationsschritte entsprechen somit einer Zeitdauer von etwa 10000 s oder knapp drei Stunden. Leider lassen sich Simulationen mit reali-

stischen Schergeschwindigkeiten und über längere Zeitintervalle nicht ausführen, da die Anzahl der benötigten Zeitschritte die Möglichkeiten heutiger Computer bei weitem übersteigt.

5.3 Ergebnisse

Wir untersuchen mit unserem Modell die Verteilung der Häufigkeit von Erdbeben gegen ihre Stärke. Dabei wird die Stärke eines Erdbebens normalerweise durch ihre Magnitude gemessen. Gutenberg und Richter [63] fanden durch Analyse vieler Erdbeben für diese Verteilung das nach ihnen benannte Gesetz, daß der Zehnerlogarithmus der Häufigkeit von Erdbeben einer bestimmten Stärke umgekehrt proportional zu ihrer Magnitude abfällt:

$$N \propto 10^{-Mb}, \quad b \approx 1. \quad (5.1)$$

M ist die Magnitude des Erdbebens, die ursprünglich durch die Größe des Zeigeraus-
schlags der Seismographen definiert wurde und heute durch das Moment der bei einem
Erdbeben bewegten Masse gegeben ist. Der Vorfaktor dieser Verteilung hängt von der
betrachteten Erdbebenregion ab, ebenso ist b etwas abhängig von der beobachteten Re-
gion. Die bei einem Erdbeben freigesetzte Energie ist proportional zu $10^{1.5M}$ [157], so
daß eine doppelt logarithmische Darstellung der Anzahl von Erdbeben gegen die dabei
freigesetzte Energie eine Gerade mit der Steigung (-1.5) ergeben sollte. Diese Vertei-
lung wird in unserer Simulation gemessen. In Abb. 5.2 ist für eine einzelne Zelle der
Logarithmus der kinetischen Energie gegen die Zeit aufgetragen. Abb. 5.3 enthält einen
Ausschnitt dieser Darstellung in linearer Darstellungsweise. Wir interpretieren die bei
einem Erdbeben freigesetzte Energie als den Betrag an potentieller Energie, der in ki-
netische Energie umgewandelt wird. In den beiden Abb. 5.2 und 5.3 entspricht dies der
Differenz zwischen einem lokalen Energieminimum und dem darauffolgenden Maximum.
Um die Erdbebenstatistik unseres Modells zu erhalten, messen wir diese Energiediffe-
renzen für alle Zellen und tragen sie in ein logarithmisch unterteiltes Histogramm ein.
Wir beginnen mit der Messung nach 10^6 Zeitschritten, nachdem die Simulation in einem
stationären Zustand ist. Wir führen zwei Simulationen durch, die erste mit 32 mal 32
Zellen und die zweite mit 64 mal 64 Zellen. Für die $2 \cdot 10^6$ durchzuführenden Zeitschritte
wurden auf einer Sun Sparc 10 Workstation fünf bzw. 20 Tage benötigt. In Abb. 5.4
ist die Erdbebenstatistik für das System mit 32 mal 32 Zellen aufgetragen. Im mittlere-
ren Teil ergibt sich für viereinhalb Größenordnungen ein linearer Teil mit der Steigung
0.41. Da ein logarithmisches Histogramm verwendet wurde, ist von diesem Wert 1 zu
subtrahieren, so daß der Exponent unseres Potenzgesetzes 1.41 beträgt. Für den Simu-
lationslauf mit 64 mal 64 Zellen erhalten wir ebenfalls ein Potenzgesetz, hier ist der
Exponent 1.61. Unsere Exponenten sind äquivalent zu einem b -Wert im Gutenberg-
Richter Gesetz, die 0.94 bzw. 1.07 betragen, was in sehr guter Übereinstimmung mit
den geologischen Beobachtungen steht [157, 21], wonach dieser Wert etwa 1 beträgt. Es
gibt zwei mögliche Gründe dafür, daß die Exponenten für die beiden Simulationsläufe

nicht übereinstimmen: Einerseits kann es sich um einen Effekt handeln, der durch Randeffekte zustande kommt. Der zweite mögliche Grund dafür, daß der Simulationslauf mit 64 mal 64 Zellen einen größeren Exponenten liefert, könnte darin liegen, daß in beiden Fällen die gleiche Schergeschwindigkeit von 100 cm/s verwendet wird, was bedeutet, daß beim kleineren System die Energiezufuhr pro Zelle größer ist. Die Frage, ob der von uns gemessene b -Wert universell ist, kann nur dadurch beantwortet werden, daß viele Simulationsläufe über verschiedene physikalische Parameter und mit unterschiedlichen Systemgrößen durchgeführt werden.

Wenn wir einen Blick auf die Abb. 5.2 werfen, erkennen wir, daß große Erdbeben kleinere Erdbeben nach sich ziehen und in einigen Fällen auch kleinere Erdbeben unmittelbar zeitlich vor ihnen stattfinden. Um die Situation genauer zu beleuchten, schauen wir uns das große Erdbeben nach etwa 210000 Zeitschritten in einer Vergrößerung in linearer Darstellung an. In Abb. 5.3 erkennt man sehr deutlich ein sehr reiches Spektrum an Nachbeben und einige Vorbeben. Wir bemerken jedoch, daß die Vor- und Nachbeben zwar oft vorhanden sind, aber ihre Stärke nicht immer mit der Stärke des Hauptbebens korreliert ist, wie man in Abb. 5.2 bei dem großen Erdbeben nach etwa 140000 Zeitschritten erkennt.

Eine weitere interessante Beobachtung tritt zutage, wenn man die Abb. 5.5 und 5.6 betrachtet. Die Abb. 5.5 stellt unser aus 64 mal 64 Zellen bestehendes System am Schluß der Simulation, also nach $2 \cdot 10^6$ Zeitschritten dar. Wie man sieht, hat sich eine stark lokalisierte Bruchzone gebildet, die das gesamte System durchzieht und eine Breite von durchschnittlich zwei bis drei Zellen aufweist. Die grau schattierten Zellen haben sich um mehr als drei Grad absolut gedreht. Die beiden einzelnen grau schattierten Zellen etwas unterhalb der Bruchzone, verursachen gerade ein Erdbeben und haben eine starke Auslenkung. Die Beobachtung, daß sich eine einzelne relativ gerade Bruchzone bildet, ist in perfekter Übereinstimmung mit den Beobachtungen bei der San-Andreas-Spalte. In Abb. 5.6 ist ein Ausschnitt aus Abb. 5.5 dargestellt, der die Bruchzone in vergrößerter Form enthält. Man erkennt, daß sich einerseits Lücken gebildet haben und andererseits relativ starke Überlappungen zwischen den Polygonen bestehen, die andeuten, daß diese im Bereich der Scherzone unter sehr großen Spannungen stehen.

Zum Schluß wollen wir noch einmal auf eine Beobachtung zurückkommen, die im letzten Kapitel hervorgehoben worden ist. Die Breite der dort untersuchten Scherbänder im saturierten Bereich, also für den Bereich hoher Schergeschwindigkeiten, war um so kleiner, je höher der auf die Scherzelle wirkende Normaldruck war. Unsere Beobachtung, daß die Scherzonenbreite sehr gering ist, wenn man dem System nicht erlaubt, zu dilatieren, fügt sich sehr schön mit der Tatsache zusammen, daß man sehr hohe Normaldrücke braucht, um ein System daran zu hindern zu dilatieren. Beide Beobachtungen passen sehr gut zu den experimentellen Untersuchungen, die von Kårmån 1911 angestellt hat [93]. Von Karman fand heraus, daß Gestein, das man unter sehr hohem Normaldruck schert, eine lokale Scherzone bildet und anfängt, plastisch zu fließen.

5.4 Ausblick

Unsere zweidimensionale Simulation einer Scherzone zwischen zwei tektonischen Platten liefert eine Erdbebenstatistik mit einem Potenzgesetz über mehr als vier Größenordnungen. Der von uns ermittelte b -Wert des Gutenberg-Richter Gesetzes von 0.94 bzw. 1.07 liegt sehr nahe am Wert, der in Erdbebenzonen beobachtet wurde und 1 beträgt. Die beiden Simulationsläufe lieferten nicht exakt den gleichen Wert. Um herauszufinden, ob es eine Abhängigkeit des b -Wertes von den Simulationsparametern gibt, sollten noch mehr Simulationsläufe mit anderen Parametern durchgeführt werden. Es wäre auch interessant und im Rahmen des Modells machbar, zu überprüfen, ob das Omori-Gesetz gilt, welches für den Zusammenhang zwischen den Zeitdifferenzen zweier aufeinanderfolgender Erdbeben und deren Anzahl ein Potenzgesetz voraussagt.

Es ist auch interessant, das Modell auf drei Dimensionen zu erweitern. Die Simulation von Scherzonen wäre dann realistischer, da die tektonischen Platten eine große Dicke aufweisen und es daher besser ist, diese durch mehrere übereinanderliegende Schichten von Zellen darzustellen.

Mit einem dreidimensionalen Modell wäre es auch möglich, zu simulieren, wie eine tektonische Platte in eine andere hineingeschoben wird und diese hochdrückt, wie es z. B. bei der Indischen Platte und der Asiatischen Platte der Fall ist. Jedoch ist ein solches Modell sehr aufwendig. Wir werden in Kapitel 9 noch darauf zurückkommen.

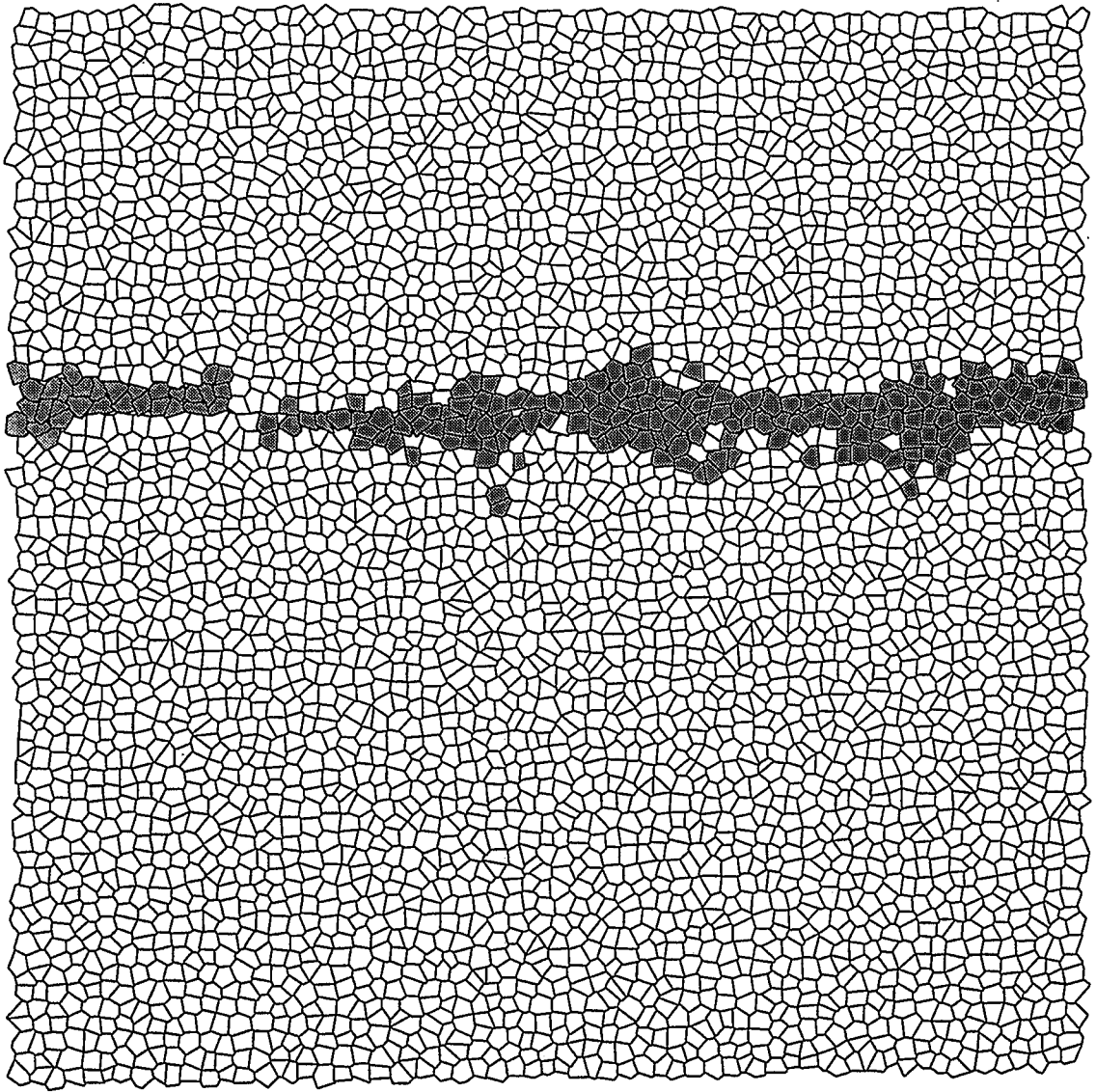


Abbildung 5.5: Zustand eines aus 64 mal 64 Zellen bestehenden Gebietes nach zwei Millionen Zeitschritten. Es hat sich nur eine, stark lokalisierte Scherzone gebildet, deren durchschnittliche Breite etwa 2 bis 3 Zellen entspricht. Die grau schattierten Zellen haben sich um mehr als drei Grad im oder gegen den Uhrzeigersinn gedreht.

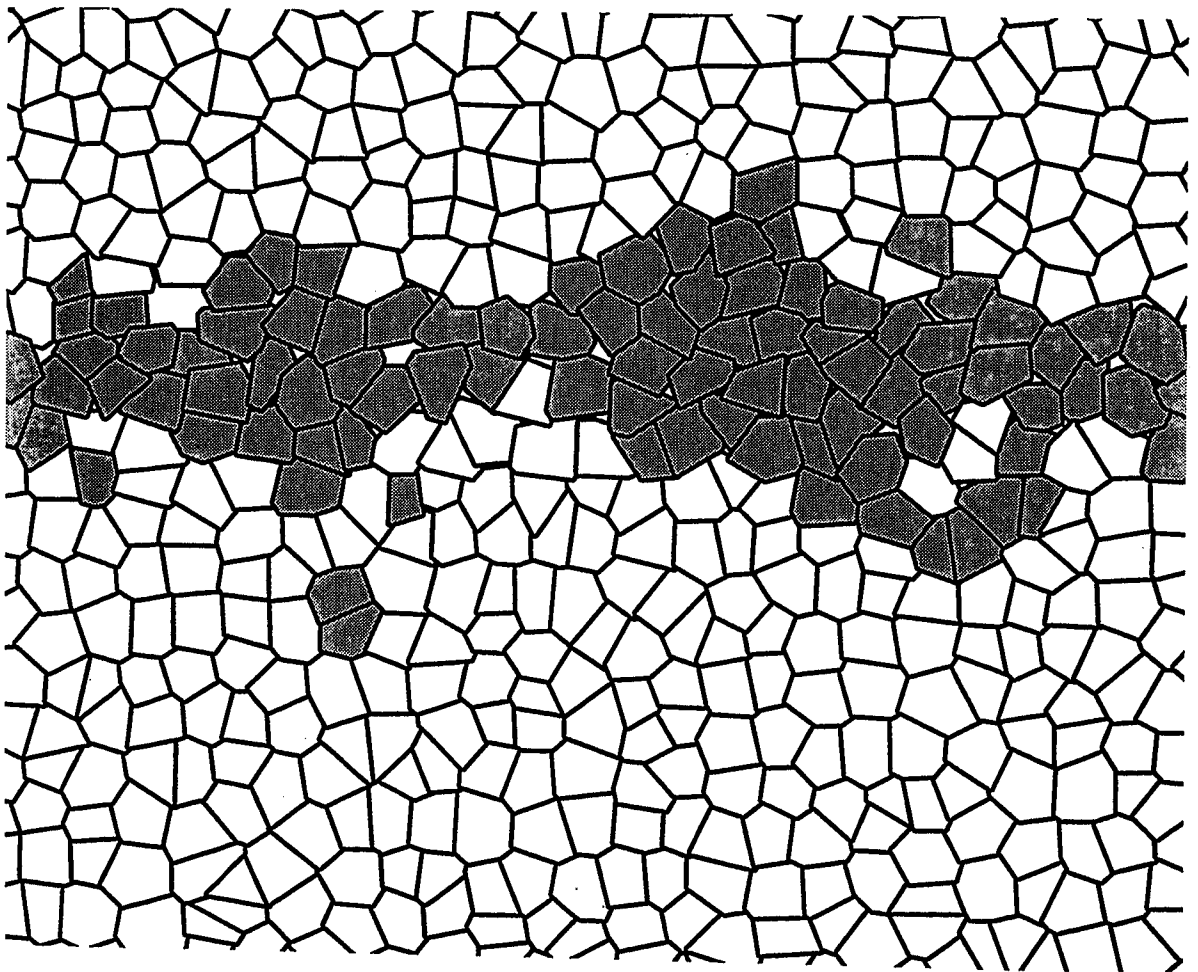


Abbildung 5.6: Diese Abbildung stellt den mittleren Teil der Abb. 5.5 vergrößert dar. Einerseits erkennt man deutlich Lücken zwischen den Teilchen und andererseits werden relativ große Überlappflächen zwischen den Polygonen sichtbar, die darauf hindeuten, daß das Material in der Scherzone unter sehr großen Spannungen steht.

Kapitel 6

Der granulare Fluß durch Trichter

6.1 Einleitung

Der Fluß von Schüttgütern durch Trichter und Rohre ist ein von vielen Seiten untersuchtes Problem, das traditionell mit Hilfe der Kontinuumsmechanik behandelt wird [158]. Dabei werden grobkörnige Materialien für gewöhnlich als nicht kompressibel und feinkörniges Schüttgut als kompressibel betrachtet. Die in solchen Studien betrachteten Trichter sind im allgemeinen so konstruiert, daß sie von oben mit einem konstanten Massenfluß gespeist werden und Schüttgut von unten weggenommen wird.

Eine wichtige Anwendung von Trichtern in der Industrie besteht in der Kollimierung von granularen Medien. Bei der Herstellung von synthetischen Granulaten etwa kann Schüttgut von einer Zerkleinerungsmühle mit Hilfe eines Trichters gesammelt werden, so daß es dem nächsten Produktionsschritt zugeführt werden kann. Der Trichter hat hierbei physikalisch betrachtet die Funktion, die Impulse der Teilchen in eine Richtung zu lenken. In solchen Anwendungen ist es nicht wünschenswert, daß der Trichter zu hoch mit Teilchen gefüllt ist, da kleine Schwankungen im Teilchenzufluß dann dazu führen können, daß es zum Überlaufen kommt. Dann muß der Produktionsprozeß unterbrochen werden, was erhebliche finanzielle Kosten verursachen kann.

In vielen Anwendungen führen äußere Zwänge dazu, daß die Ausflußöffnung des Trichters sehr klein ist und der Durchmesser dieser Öffnung nur wenigen Teilchendurchmessern entspricht. Solche Fälle können technisch dann notwendig werden, wenn das Schüttgut entweder für den nächsten Produktionsschritt sehr stark kollimiert werden muß oder wenn es Platzprobleme in der Produktionshalle gibt. In der Tat sind Fälle, wo die Ausflußöffnung einen Durchmesser hat, der nur 5-10 mal so groß ist wie der Teilchendurchmesser nicht selten. In solchen Anwendungen kann die körnige Struktur des Schüttgutes nicht mehr vernachlässigt werden, und ein Zugang mit Hilfe der Kontinuumsmechanik ist nicht angemessen.

Wir vermeiden diese Probleme hier, indem wir eine Computersimulation benutzen, die die granulare Natur der Teilchen explizit in Betracht zieht [3, 46, 163]. Das dabei benutzte Simulationsmodell wurde ausführlich in Kapitel 2 beschrieben. Man beachte

in diesem Zusammenhang, daß für jedes einzelne Korn die genaue Bahnkurve, seine Rotation, Haft- und Gleitreibung, sowie Dissipationseffekte berechnet wurden.

Hier beschäftigen wir uns hauptsächlich mit der zweiten der oben angeführten Anwendungsmöglichkeiten eines Trichters, nämlich der Kollimation von Schüttgut. Um die in Bezug auf Computerzeit ohnehin aufwendigen Berechnungen so einfach wie möglich zu halten, betrachten wir ein rotationssymmetrisches System, so daß die wesentlichen physikalischen Vorgänge mit einer zweidimensionalen Simulation eingefangen werden können. Jedoch sind wir uns dabei wohl bewußt, daß wir möglicherweise Effekte vernachlässigen, die durch die nicht berücksichtigten drei Bewegungsmöglichkeiten hervorgerufen werden können. Zunächst gehen wir im nächsten Abschnitt kurz auf die speziellen, in Kapitel 2 nicht behandelten modellartigen Aspekte ein. Dann stellen wir in Abschnitt 6.3 die verschiedenen von der Einfüllrate abhängigen Flußregime vor und analysieren in Abschnitt 6.4 im Detail, wie das Flußfeld für einen in der Praxis der Verfahrenstechnik eher üblichen Trichter mit schräg ansetzendem Ausflußrohr aussieht. Schließlich wird in Abschnitt 6.5 besprochen, wie sich das Modell noch verbessern läßt und wo es noch Schwachstellen hat.

6.2 Das numerische Modell

Wir betrachten die Granulatkörner als irregulär geformte Polygone und verfolgen ihre Flugbahnen und Drehbewegungen mit Hilfe der Molekulardynamik-Methode, wie sie in Kapitel 2 explizit beschrieben worden ist. Die Trichterwände stellen wir als Teilchen mit einer sehr großen Masse dar. Es wird dafür Sorge getragen, daß die Wandteilchen sich nicht bewegen, indem ihre Beschleunigungen nach der Kraftberechnung explizit auf Null zurückgesetzt werden. Zusätzlich zu den in Kapitel 2 vorgestellten Techniken gibt es hier noch Routinen, die dafür sorgen, daß der Trichter von oben mit einer gleichmäßigen Einfüllrate mit Teilchen beschickt wird und daß die Teilchen, die aus der unteren Öffnung des Trichters austreten, aus der Simulation entfernt werden. Ferner muß man die Gravitation berücksichtigen, was dadurch geschieht, daß man zur y-Koordinate der Beschleunigung die Erdgravitation addiert.

Für die Simulation der Haftreibung benutzen wir hier ein Modell, das erstmals von Lee [100] und Lee und Herrmann [101] vorgeschlagen wurde. Hier werden kleine Federn benutzt, die am Kollisionspunkt zwischen den beiden beteiligten Teilchen eingesetzt werden und eine Kraft erzeugen, die der Relativbewegung der beiden Teilchen entgegenwirkt. Von dieser Kraft wird nur der tangentielle Anteil verwendet, der den gewünschten Effekt hat, daß die Teilchen sich gegenseitig abbremsen. Der normale Anteil dieser Kraft wird verworfen, da dieser einen nicht erwünschten adhäsiven Effekt hat. Die Federlänge L_F ergibt sich zu

$$L_F(t) = \int_{t_0}^t v_{rel}^T dt. \quad (6.1)$$

Dabei bezeichnet t_0 den Beginn des Zusammenstoßes oder den Zeitpunkt des letzten

Federbruches. Die Federkraft ist nach dem Hooke'schen Gesetz

$$F_H = k_H \cdot L_F. \quad (6.2)$$

Wenn die Feder eine gewisse Auslenkung L_B überschreitet, bricht sie und wird entfernt. Falls die Partikel dann noch Kontakt zueinander haben, wird eine neue Feder eingesetzt. Für die gesamte tangentiale Kraft gilt dann, in Erweiterung der Gleichung aus Kapitel 2 :

$$F_T = -\min(m_r \gamma_N v_{rel}^N + F_H, \mu F_N). \quad (6.3)$$

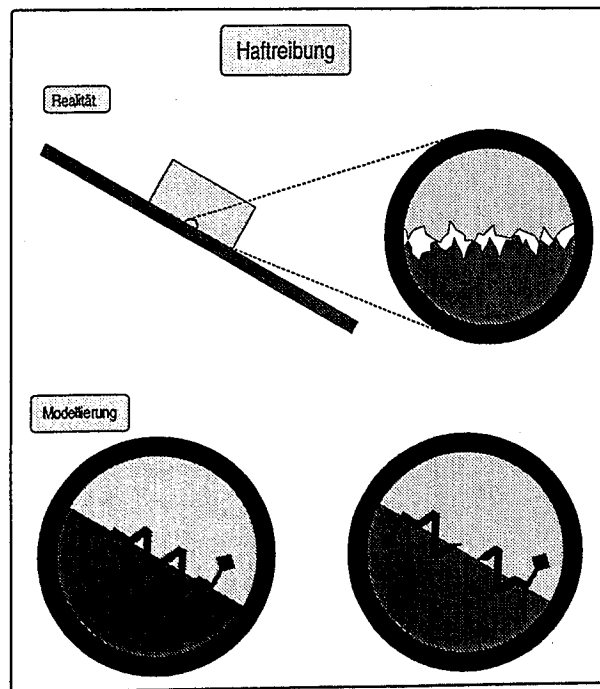


Abbildung 6.1: Die Haftreibung realer Oberflächen ist auf deren Rauigkeit zurückzuführen (oben). In unserem Modell wird die Haftreibung durch kleine Federn realisiert, die bei zu hoher Ausdehnung brechen (unten).

6.3 Die granularen Flußregime

Die Geometriedaten des hier untersuchten Trichters sind der Abb. 6.9 zu entnehmen. Um die Größenverhältnisse zwischen dem Trichter und den Teilchen zu illustrieren, ist auch ein individuelles Teilchen eingezeichnet. Die hier untersuchten Teilchen haben eine quadratische Form und sind monodispers, d. h. sie sind alle gleich groß. Es ist zwar wie im bisherigen Verlauf der Darstellung schon mehrfach zum Ausdruck gekommen ist, möglich fast beliebig geformte Teilchen zu simulieren (solange diese durch konvexe

Parameter	Symbol	Einheiten	Wert
Gravitation	g	$\frac{cm}{s^2}$	1000
Dichte	ρ	$\frac{g}{cm^3}$	1
Young Modul	Y	$\frac{dyn}{cm^2}$	10^7
Dissipation	γ_N	s^{-1}	500
Gleitreibung	γ_T	s^{-1}	500
Federhärte	k_F	$\frac{dyn}{cm}$	10^5
Federlänge	L_B	cm	0.05
Haftreibung	μ	1	2.0
Zeitschritt	dt	s	$5 \cdot 10^{-5}$

Tabelle 6.1: Tabelle der Simulationparameter im Programm. Alle Werte sind in cgs-Einheiten angegeben. Die genaue Bedeutung der Parameter ist im Text sowie in Kapitel 2 erklärt.

Polygone darstellbar sind), jedoch wollen wir uns hier auf einen möglichst einfachen nicht sphärischen Fall beschränken. Wir bemerken, daß die gefundenen quantitativen Ergebnisse zwar nur auf den untersuchten Trichter zutreffen, die qualitativen Aussagen aber eine sehr generelle Aussagekraft haben.

In Tabelle 6.1 sind die benutzten Parameterwerte dargestellt. Das gewählte Elastizitätsmodul entspricht einem synthetischen Granulat mittlerer Härte. Unsere Dissipationskonstante γ_N entspricht einem Restitutionskoeffizienten von etwa $\epsilon = 0.9$, was einem Energieverlust von 10% bei jedem Teilchenstoß entspricht. Dies ist ein realistischer Wert für synthetisches Granulat [158]. Um Verstopfungen von Trichtern reproduzieren zu können, ist es essentiell, die Haftreibung von Partikeln untereinander und mit den Wänden mitzuberechnen. Die in den Simulationen gewählte Kombination der Parameter μ und k_f entspricht einem Haftreibungskoeffizienten, der etwas über eins liegt, vgl. Tab. 6.1. Mit anderen Worten bedeutet das, daß die von uns verwendeten Teilchen eine relativ raue Oberfläche haben.

Zunächst bestimmten wir die Ausflußrate des Trichters als Funktion der Einfüllrate. Das Resultat dieser Messung ist in Abb. 6.4 dargestellt. Die Datenpunkte stellen jeweils einen Simulationslauf dar. Die Fehlerbalken ergeben sich aus den zeitlichen Fluktuationen der Meßwerte. Die gestrichelte horizontale Linie gibt die Ausflußrate an, wenn der Trichter verstopft ist. Wie man sieht, liegt diese Rate beträchtlich unter der maximal erreichbaren Ausflußrate bei einem kontinuierlichen granularen Fluß. In diesem Diagramm können wir vier verschiedene Flußregime unterscheiden.

Das Regime A ist der Bereich des kontinuierlichen granularen Flusses. Soll der Trichter zur Kollimation von granularen Medien gebraucht werden, so hat dieses Regime zwei sehr wünschenswerte Vorzüge. Erstens ist die Ausflußrate stetig und exakt gleich groß wie die Einfüllrate. Zweitens ist der Trichter im Regime A sehr unempfindlich gegen große Schwankungen in der Einfüllrate. Selbst wenn diese so groß sind, daß es vorüber-

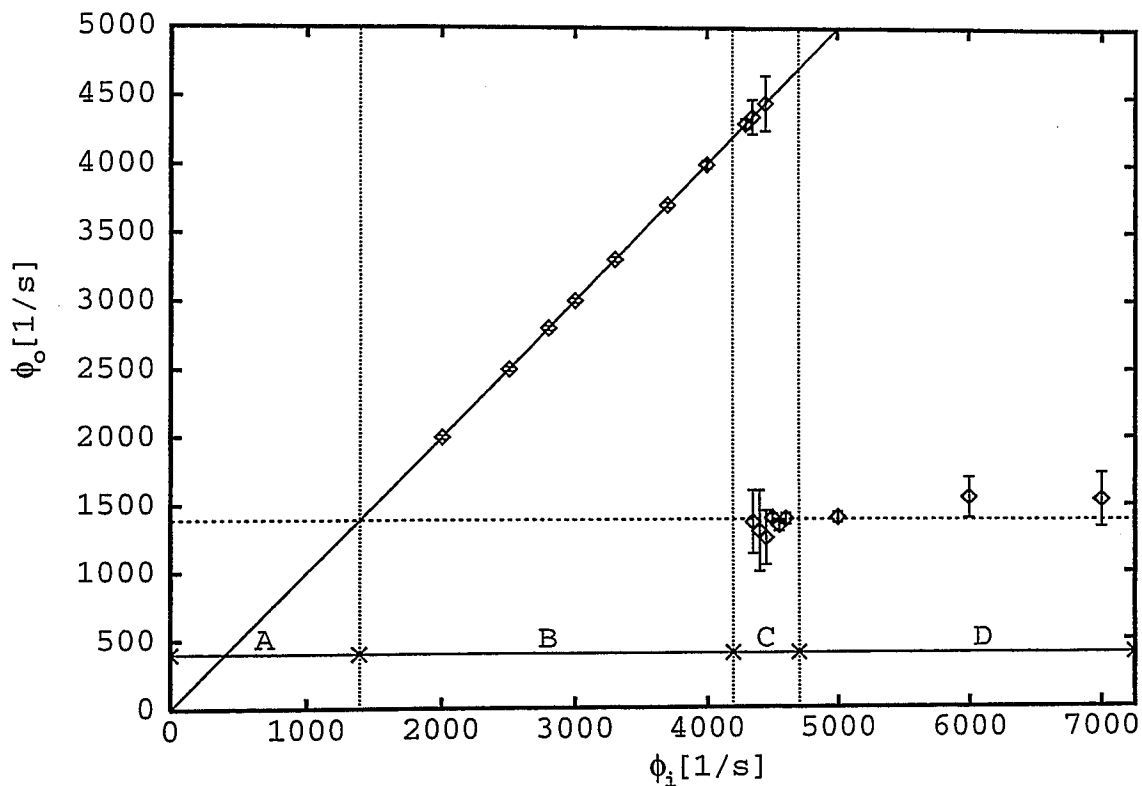


Abbildung 6.2: Darstellung der Ausflußrate gegen die Einfüllrate des Teilchenflusses im Trichter. Die Simulationsdaten sind die Mittelwerte aus jeweils 50 Simulationsläufen. Die durchgezogene Linie markiert das Regime des kontinuierlichen granularen Flusses. Die horizontale gestrichelte Linie gibt die Ausflußrate für verstopfte Trichter an. Die vertikalen gestrichelten Linien geben die Grenzen zwischen den vier Flußregimen an.

gehend zur Verstopfung kommt, kehrt er sofort wieder zu seinem stetigen Operationsmodus zurück, sobald die Einfüllrate wieder im ursprünglichen Bereich liegt. In anderen Worten, der Trichter sollte im Regime A niemals verstopfen.

Das Flußregime D bezeichnen wir als den Bereich des unstetigen granularen Flusses. Hier ist der Trichter immer verstopft und die Ausflußrate ist konstant und kleiner als die Einfüllrate, so daß ein Trichter in diesem Operationsmodus schließlich überlaufen wird.

Das Regime B ist ein Bereich stetigen granularen Flusses, unterscheidet sich aber von A dadurch, daß der Trichter hier nicht mehr unempfindlich gegen große Fluktuationen in der Einfüllrate ist. Tritt nach einer größeren Fluktuation der Einfüllrate eine Verstopfung des Trichters auf, so läuft dieser voll, da dann die Ausflußrate kleiner ist als die Einfüllrate. Kleinere Fluktuationen übersteht der Trichter jedoch auch im Regime B noch ohne Schaden.

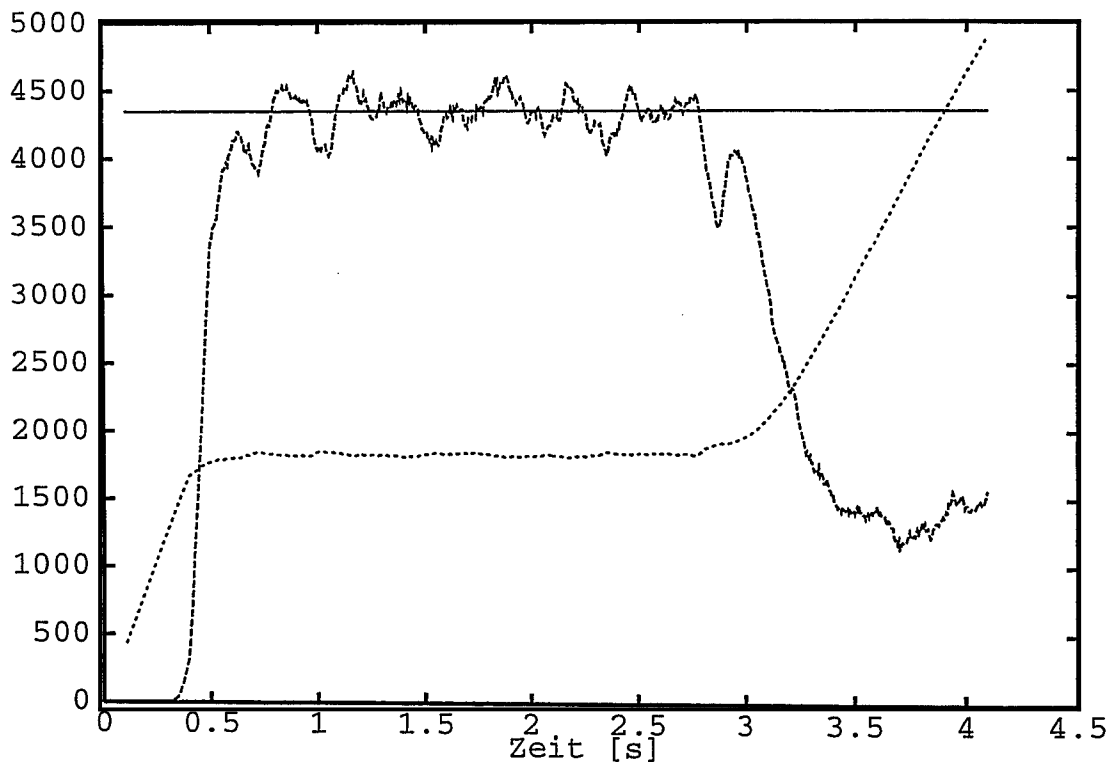


Abbildung 6.3: Darstellung der Einfüllrate und der Ausflußrate (jeweils in Teilchenzahl pro Sekunde), sowie der Gesamtzahl der Teilchen, die sich im Trichter befinden, aufgetragen gegen die Zeit. Die durchgezogene Linie stellt die Einfüllrate dar, die hier konstant ist. Die lang-gestrichelte Linie zeigt die Ausflußrate, während die kurz-gestrichelte Linie die Teilchenzahl im Trichter kennzeichnet.

Im Regime C ist das Flußverhalten des Trichters unstetig. Für jede beliebige Einfüllrate im Regime C, auch wenn diese stetig ist und keine Fluktuationen aufweist, wird der Trichter für eine gewisse Zeit gut funktionieren, aber schließlich verstopfen und überlaufen. Ein Beispiel für solch einen Fall ist in der Bildfolge Abb. 6.10(I) bis Abb. 6.11(IV) dargestellt. Hier sieht man, daß das Ausflußverhalten für eine relativ lange Zeit konstant ist und dann plötzlich unstetig wird und der Trichter verstopft.

Im Regime C bezeichnen wir die Zeit, die es im Mittel dauert, bis der Trichter verstopft, als die "mittlere Zeit bis zum Verstopfen" (MZV). In Abb. 6.4 ist die mittlere Zeit bis zum Verstopfen gegen die Einfüllrate für das Regime C dargestellt. Jeder Datenpunkt ist hier eine Mittelung über 50 Simulationsläufe, welche sich durch die Anfangsgeschwindigkeiten und -winkel der einzufüllenden Teilchen unterscheiden. Die Fehlerbalken ergeben sich aus der Standardabweichung der Meßwerte der verschiedenen Simulationsläufe. Dieses Diagramm zeigt sehr deutlich, daß die Kurve, die dieses Zeit-

verhalten ausdrückt, an der Grenze zwischen Regime B und Regime C divergiert. Das

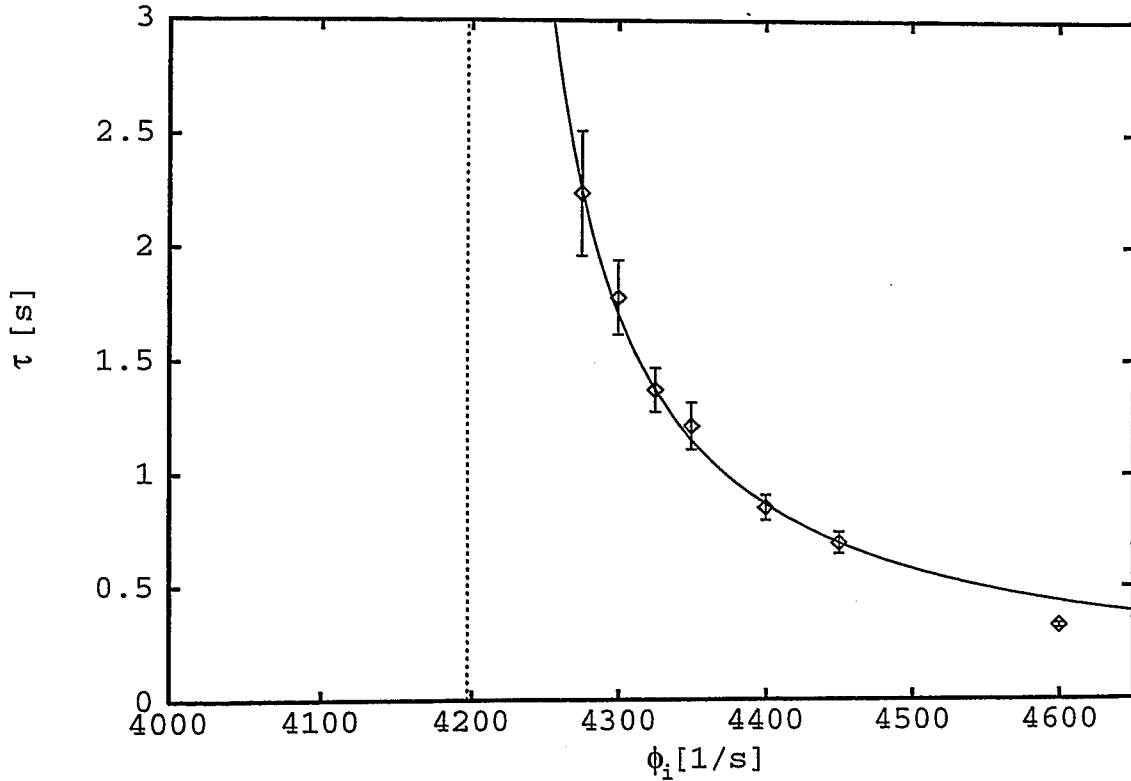


Abbildung 6.4: Darstellung der mittleren Zeit bis zum Verstopfen (MZV) als Funktion der Einfüllrate im Flußregime C. Die durchgezogene Kurve stellt die im Text angegebene Gleichung dar, an die das kritische Verhalten angepaßt werden kann. Die gestrichelte Linie markiert die Einfüllrate, bei der die MZV unendlich würde. Jeder Datenpunkt ist eine Mittelung über 50 Simulationsläufe.

kritische Verhalten in diesem Flußregime kann an folgende Gleichung angepaßt werden:

$$\tau_C = \frac{A}{|\Phi_i - \Phi_c|^\beta}. \quad (6.4)$$

Die durchgezogene Kurve in Abb. 6.4 stellt eine solche Kurve dar, mit den Parametern: $A = 176 \pm 10$ und $\Phi_c = 4197 \pm 10$. Die Werte passen gut für ein β der Größe eins. Es ist bemerkenswert, daß diese Verhaltensweise typisch ist für einen Phasenübergang zweiter Ordnung. Deswegen nennen wir das Flußverhalten im Regime C in der Sprache der statistischen Physik "kritisches Flußverhalten".

Um ein tieferes Verständnis vom eigentlichen Mechanismus, der zur Verstopfung des Trichters führt, zu erlangen, haben wir unsere Simulationsdaten visualisiert und Filme hergestellt (mit Hilfe der Visualisierungssoftware AVS). Szenen eines solchen Films,

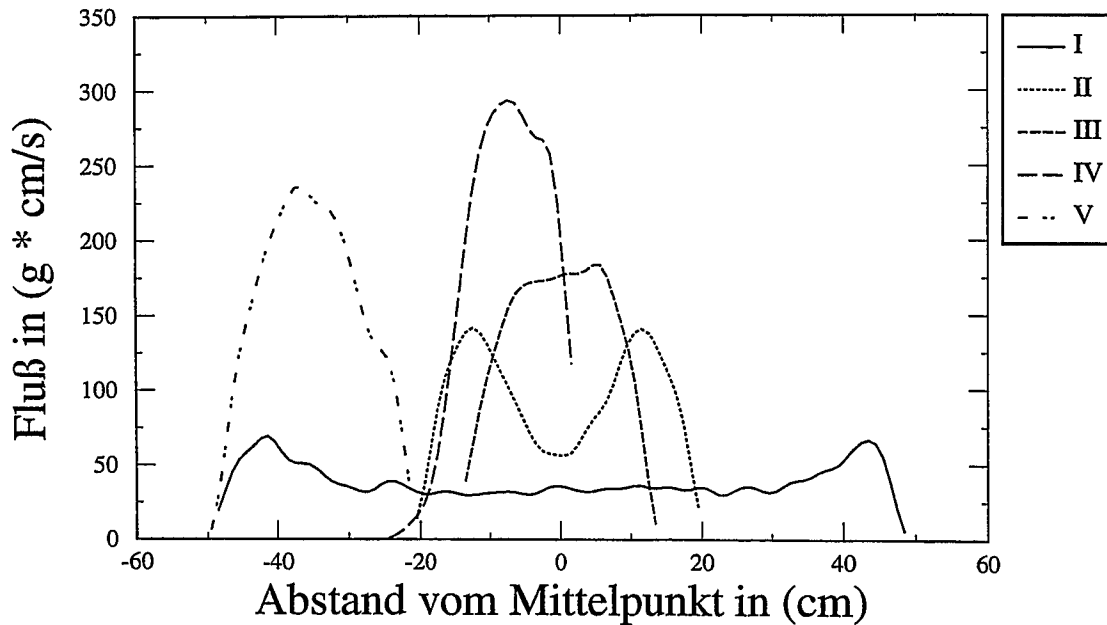


Abbildung 6.5: Die Flußprofile für die fünf Schnitte aus Abb. 6.12. Die Kurven wurden zusätzlich geglättet.

in dem ein Trichter dargestellt wird, der in Regime C arbeitet und verstopft, sind in Abb. 6.10 und Abb. 6.11 (I) bis (IV) dargestellt. Die Graustufen kodieren hier die kinetische Energie der Partikel (Translationsenergie plus Rotationsenergie): helle Graustufen entsprechen einer kleinen kinetischen Energie, dunkle Graustufen stellen hohe kinetische Energien dar. In Abb. 6.10(I) haben die meisten Partikel eine hohe Geschwindigkeit, bis auf diejenigen, die Kontakt mit den Wänden haben. Im nächsten Bild Abb. 6.10(II) sieht man, daß die kinetische Energie der Teilchen an den Trichterwänden verloren geht. Denn die Teilchen nahe den Wänden haben kleinere kinetische Energien als die im mittleren Bereich des Trichters. Dies wird im nächsten Bild 6.11(III) der Sequenz noch deutlicher. Das letzte Bild 6.11(IV) zeigt, wie sich die Verstopfung im Trichter nach oben hin fortpflanzt. Aus der gleichen Simulation wie diese Bildsequenz entstammt die Abb. 6.3, wo die Einfüllrate, die Ausflußrate und die Gesamtzahl der Teilchen im Trichter gegen die Zeit aufgetragen sind. Hier ist deutlich erkennbar, daß der Trichter eine relativ lange Zeit einwandfrei arbeitet und dann plötzlich verstopft.

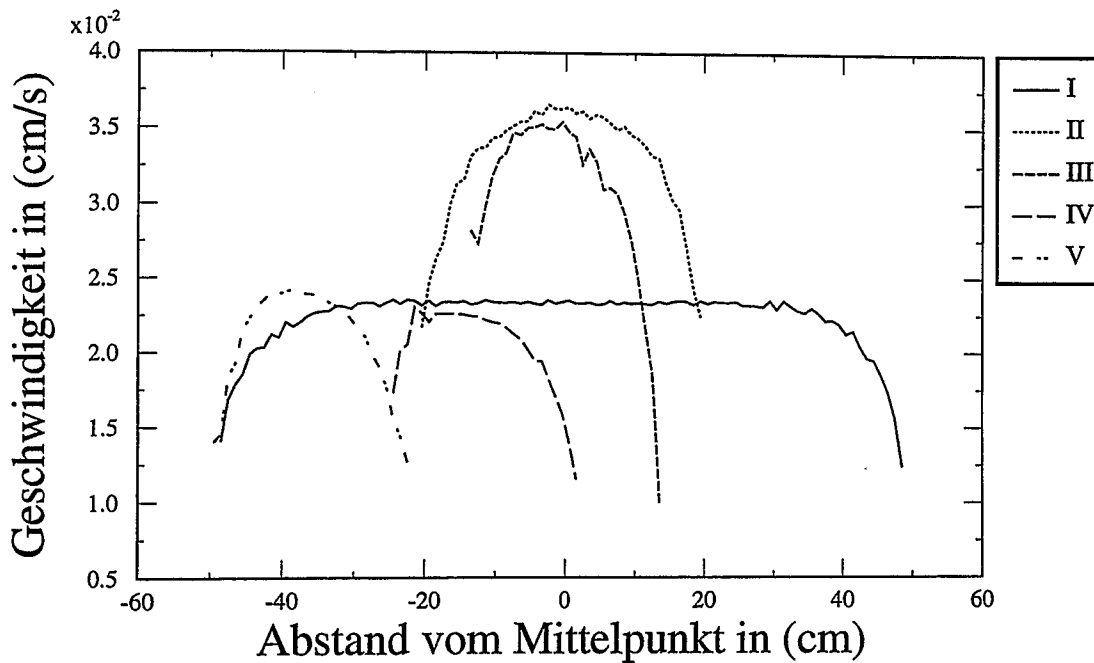


Abbildung 6.6: Die Absolutbeträge der Geschwindigkeiten der fünf Schnitte aus Abb. 6.12 gegen den Abstand vom Trichtermittelpunkt aufgetragen.

6.4 Flußprofile

Zur Bestimmung des mittleren Flußverhaltens im Trichter kann man die aus der Hydrodynamik bekannten Flußprofile aufstellen. Allerdings muß dazu, im Gegensatz zur Hydrodynamik inkompressibler Medien, außer der mittleren Geschwindigkeit eines Teilchens auch deren Produkt mit der zeitlich gemittelten Teilchendichte ausgewertet werden. Wir untersuchen hier einen Trichter mit einer komplizierteren Geometrie mit nicht gerade ansetzendem Ausflußrohr, der im Gegensatz zum im vorigen Abschnitt behandelten Trichter eher typisch ist für reale verfahrenstechnische Anwendungen. Da wir hier eher ein in der Praxis realisiertes System untersuchen wollen, benutzen wir auch für die Teilchen nun nicht mehr einfache quadratische Teilchen sondern Rechtecke mit einer polydispersen Größenverteilung ($x = (3.5 \pm 2.5) \text{ cm}$ und $y = (1.5 \pm 1) \text{ cm}$).

In Abb. 6.12 ist das Geschwindigkeits- und das Flußfeld für das Gesamtsystem dargestellt. Abb. 6.5 zeigt die Flußprofile für die fünf in Abb. 6.12 eingezeichneten Schnitte. Die Abbildungen 6.6, 6.7 und 6.8 zeigen jeweils die Geschwindigkeitsprofile für den Absolutbetrag, die vertikale und die horizontale Komponente.

Es ist bemerkenswert, daß die Flüsse zumindest im oberen Teil des Trichters in der Mitte kleiner sind als in den Regionen nahe am Rand, obwohl die Teilchengeschwindig-

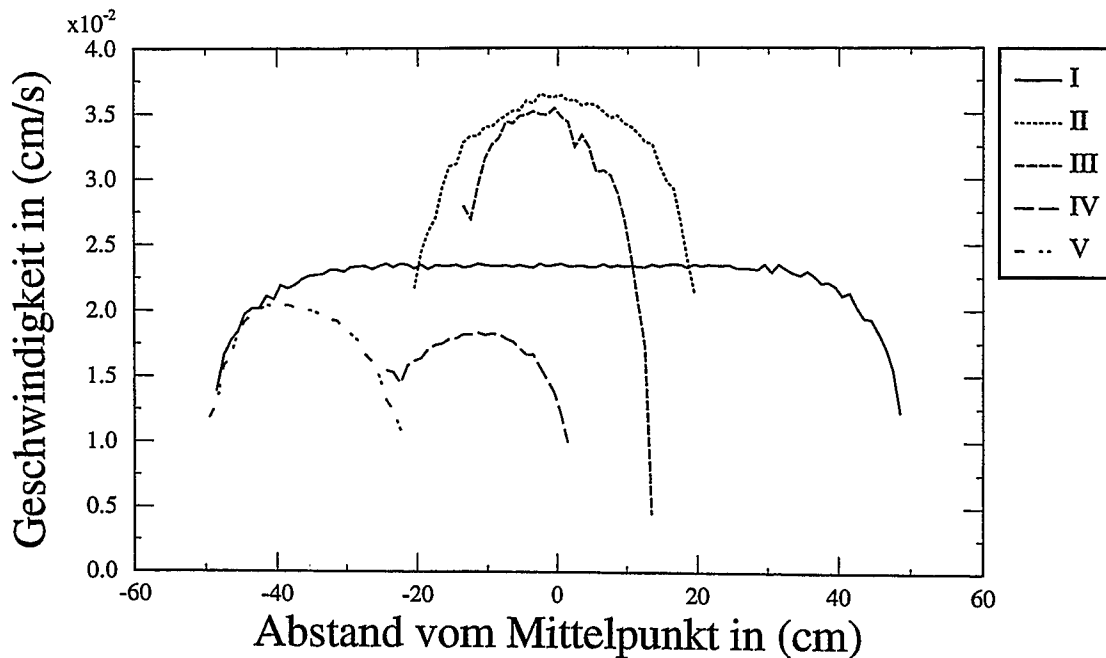


Abbildung 6.7: Die Vertikalkomponenten der Geschwindigkeiten der fünf Schnitte aus Abb. 6.12 gegen den Abstand vom Trichtermittelpunkt aufgetragen.

keiten zu den Rändern hin abnehmen. Der Grund hierfür liegt in einer Art “Kondensation” der Teilchen am Rand. Ein Teilchen trifft mit hoher Geschwindigkeit auf den Rand und dissipiert dort einen Teil seiner kinetischen Energie. Aufgrund seiner niedrigeren Geschwindigkeit nach der Kollision entfernt es sich horizontal langsamer von der Wand, in vertikaler Richtung wird es durch die Erdanziehung beschleunigt. Da aber die Wände zusammenlaufen, verbleibt es in der Nähe der Wand.

Ein anderes Phänomen ist im Rohr direkt hinter dem Trichteransatz sichtbar. Obwohl das Geschwindigkeitsfeld dort relativ konstant ist, zeigt das Flußfeld dort eine starke Asymmetrie. Für den nicht verstopfenden Fall trägt dieser *Totwasserbereich* links unter dem Ansatz so gut wie nichts zum Massentransport bei.

6.5 Diskussion

Wir haben den granularen Fluß von nicht-sphärischen Teilchen durch einen kollimierenden Trichter mit Hilfe einer Computersimulation untersucht. Wir konnten qualitativ vier verschiedenen Flußregime unterscheiden, von denen eines ein kritisches Verhalten im Sinne der statistischen Physik aufweist. Darüberhinaus haben wir mittels Visualisierungstechniken ein klares Bild darüber erhalten, auf welche Art und Weise ein Trichter

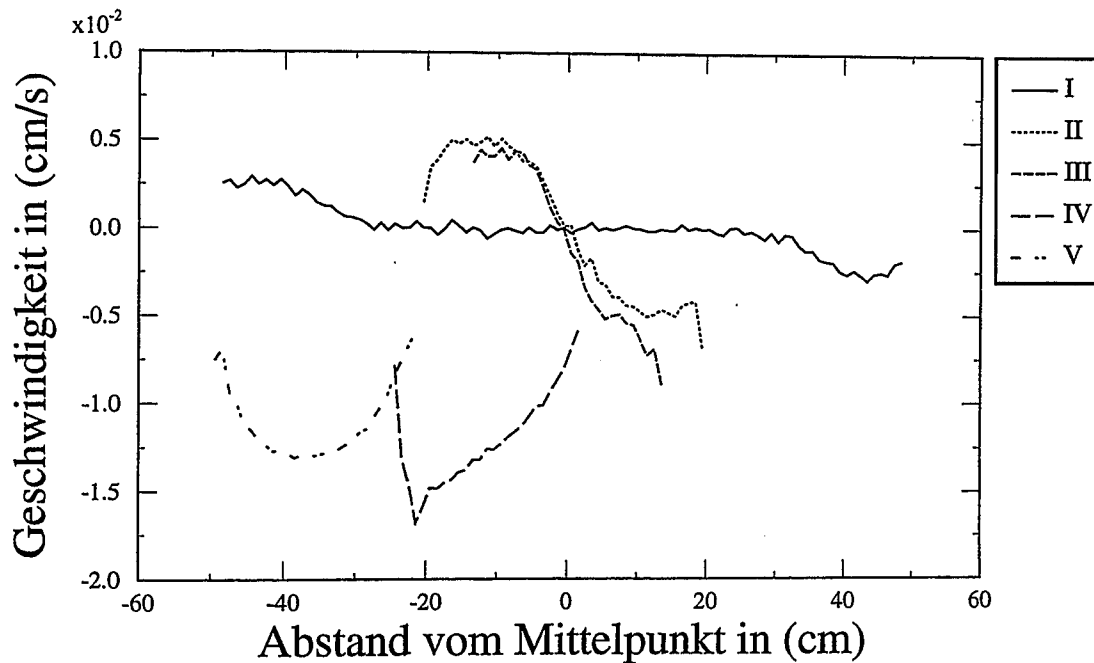


Abbildung 6.8: Die Horizontalkomponenten der Geschwindigkeiten der fünf Schnitte aus Abb. 6.12 gegen den Abstand vom Trichtermittelpunkt aufgetragen.

verstopft. Die Ursache für diese Verstopfung liegt darin, daß Teilchen Energie an die Wände verlieren. Zusätzlich haben wir einen Trichter sowie Teilchen mit komplizierterer Geometrie untersucht, um das Flußfeld eines Trichters zu untersuchen, wie er in der verfahrenstechnischen Praxis typisch ist.

Das kritische Flußregime ist sehr allgemeiner Natur und unabhängig von der Geometrie des Trichters und der Teilchenform, obschon andere Details des Regimes C, wie seine Breite und seine Begrenzungen von den Trichter- und Teilchencharakteristiken abhängen mögen. Für den Exponenten β in der Gleichung, die den Phasenübergang zweiter Ordnung beschreibt, haben wir einfach eins eingesetzt. Man könnte allgemeiner β als freien Parameter behandeln und dann durch die Meßdaten bestimmen, welcher Wert sich für β ergibt. Würde sich dann immer wieder der gleiche Wert für β ergeben, unabhängig von der speziellen Form des Trichters und der Teilchen, dann wäre β ein universeller Parameter und man müßte das Phänomen des kritischen Flusses als eine immanente Erscheinung betrachten, die in der Natur des granularen Flusses in Trichtern liegt. Die Daten reichen jedoch noch nicht aus, um zu entscheiden, ob β universell ist oder nicht. Es liegt auf der Hand, daß dies eine sehr Rechenzeit intensive Aufgabe ist, da man Simulationsläufe für viele verschiedene Trichter- und Teilchenformen durchführen müßte.

Unser Computermode'll erlaubt es über diese qualitativen Aussagen hinaus, für eine spezielle Trichter- und Teilchengemetrie auch quantitative Aussagen zu machen. Die vorliegenden Ergebnisse wurden in zwei Dimensionen berechnet, da der von uns betrachtete Trichter eine Rotationssymmetrie bezüglich der zentralen Achse aufweist. Im Prinzip ist es möglich, eine volle dreidimensionale Computersimulation durchzuführen. Jedoch wäre eine solche Simulation sehr rechenzeitintensiv und würde große Parallelrechner erfordern. Auf schnellen Workstations ist es derzeit möglich, einige tausend Teilchen in zwei Dimensionen zu berechnen (ein Simulationslauf, der eine Echtzeitdauer von etwa 10 Sekunden berechnet, benötigt dabei etwa einen Tag Rechenzeit auf einer Sun Sparc 10 Workstation). Mit der gleichen Workstation kann man etwa hundert Teilchen in drei Dimensionen berechnen. Zusätzlich muß man berücksichtigen, daß man in drei Dimensionen für Systeme mit gleicher Seitenlänge wie in zwei Dimensionen viel mehr Teilchen braucht, um den Raum auszufüllen.

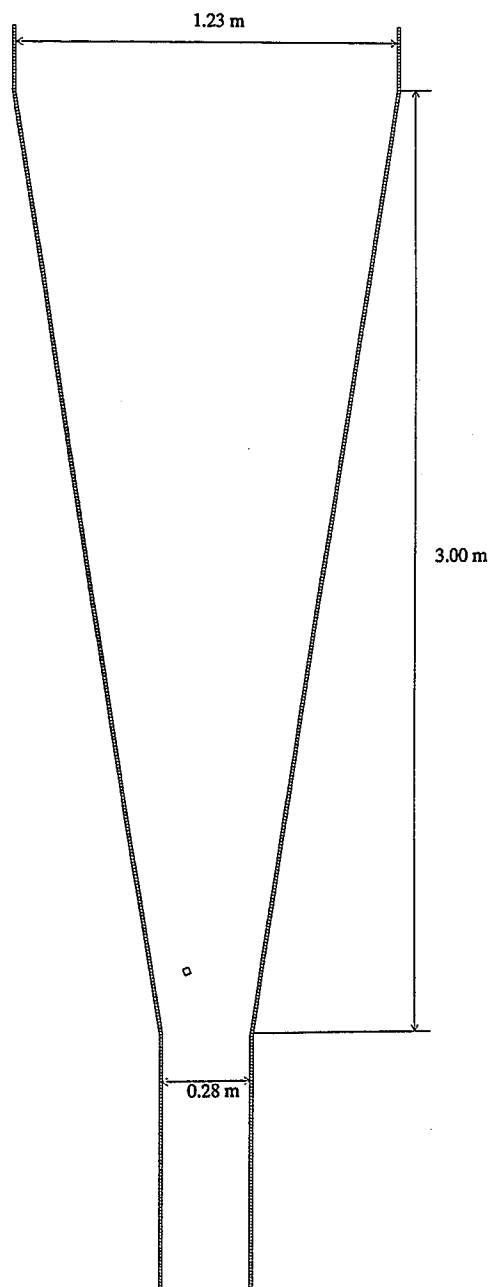
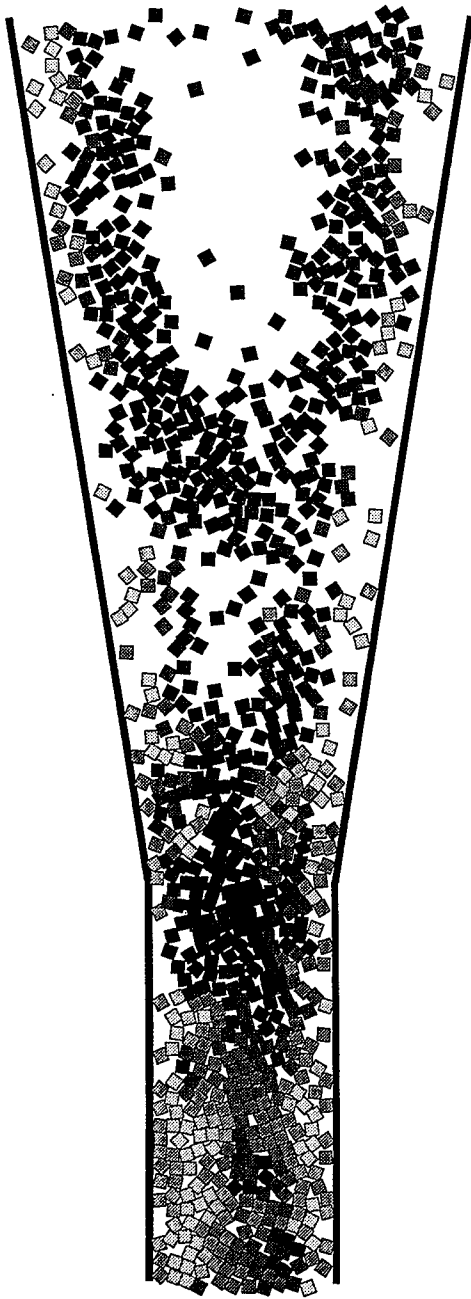
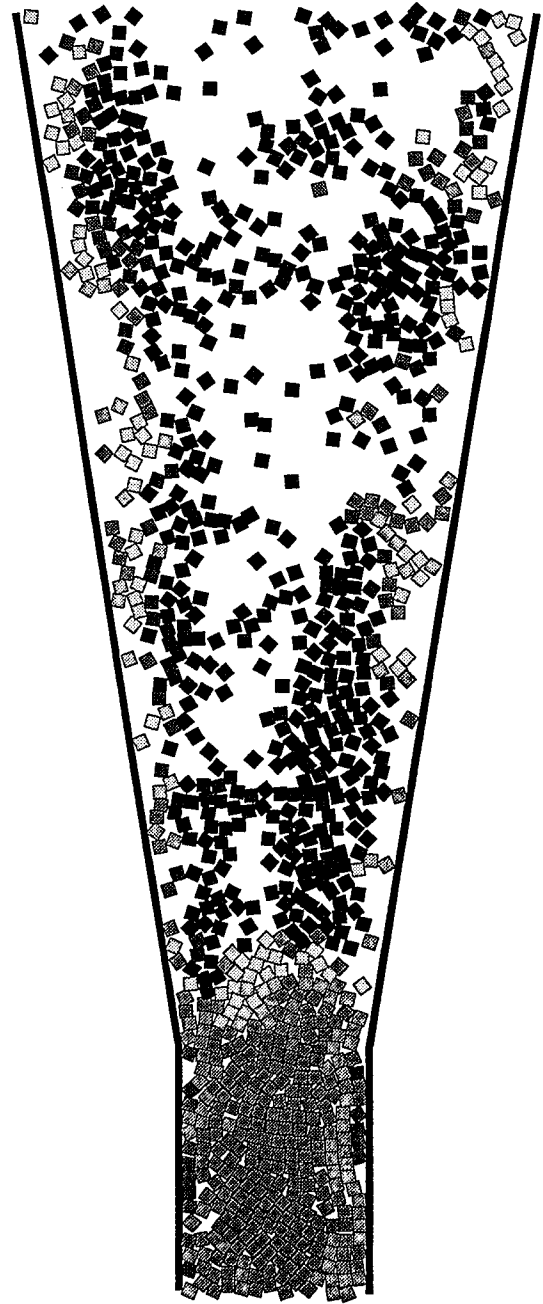


Abbildung 6.9: Schematische Darstellung der Trichter-Geometrie, die wir für die Untersuchung der Flußregime verwendet haben. Die Höhe des Trichters beträgt 150 Einheiten, die obere Einfüllöffnung hat einen Durchmesser von 61,5 Einheiten und der Ausflußdurchmesser beträgt 14 Einheiten. Die Einheiten sind so normiert, daß ein Teilchen eine Seitenlänge von einer Einheit hat, hier 2cm. Zur Illustration der Größenverhältnisse ist zusätzlich ein frei fallendes Teilchen eingezeichnet.



(I)



(II)

Abbildung 6.10: Hier und in der folgenden Abbildung werden vier Sequenzen einer Simulation gezeigt, in der es zur Verstopfung eines Trichters kommt. Abb. 6.10(I) ist das früheste Bild der Sequenz. Abb. 6.11(IV) das späteste.



Abbildung 6.11: Die Graustufen repräsentieren die kinetische Energie der Teilchen, d. h. Translations- plus Rotationsenergie. Helle Töne entsprechen einer kleinen, dunkle Töne einer großen kinetischen Energie.

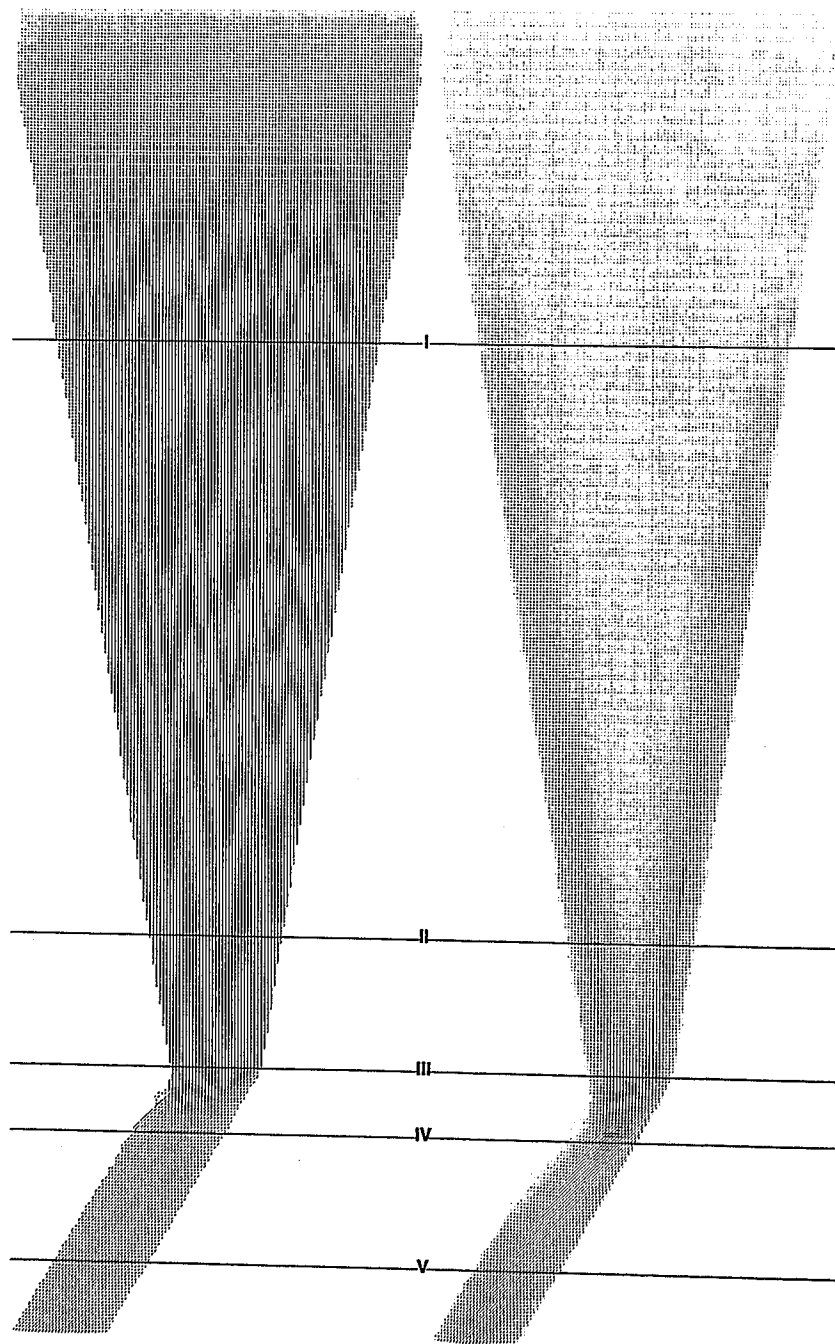


Abbildung 6.12: Das linke Bild zeigt das Geschwindigkeitsfeld für die mittlere Teilchengeschwindigkeit an. Das rechte Bild zeigt das Feld für das Produkt aus mittlerer Teilchengeschwindigkeit und mittlerer Massendichte an. Die Linien geben Betrag und Richtung der jeweiligen Größe an. Die waagerechten Striche zeigen an, wo die Geschwindigkeits- (Abb. 6.6, 6.7 und 6.8) und Flußprofile (Abb. 6.5) bestimmt wurden.

Kapitel 7

Schüttgut in einer sich drehenden Trommel

7.1 Einleitung

Der Fluß von Granulaten in rotierenden Trommeln oder Zylindern enthüllt eine Fülle interessanter Effekte und ist sowohl in vielen experimentellen Arbeiten als auch in numerischen Untersuchungen behandelt worden. Darüberhinaus ist die genaue Kenntnis des dynamischen Verhaltens von granularen Medien in rotierenden Trommeln auch von großer praktischer Bedeutung, da viele wichtige technische Anwendungen darauf beruhen, wie z. B. das Mischen und Zerkleinern von Granulaten [145, 148]. Nakagawa verwendete in einem Experiment die nuklear-magnetische Resonanz Spektroskopie (NMR) für eine rotierende Trommel, die Senfkörner enthielt [122]. Diese enthalten Öl, dessen Wasserstoffatome für das Verfahren notwendig sind. Mit dieser Methode war es auf sehr elegante Weise möglich, die Geschwindigkeit und die Konzentration der Partikel in der gesamten Trommel zu bestimmen. In [137, 188] wurde experimentell ein hystereseartiges Verhalten in rotierenden Trommeln beobachtet. Je nachdem ob man die Rotationsgeschwindigkeit der Trommel erhöht oder verringert, gibt es zwei unterschiedliche kritische Winkel, bei denen die Bewegung der Granulatkörner an der Oberfläche von einem nichtstationären Fluß, der durch das Auftreten von Lawinen gekennzeichnet ist, zu einem stationären Fluß übergeht. Wenn man die Rotationsgeschwindigkeit langsam erhöht ist dieser kritische Winkel höher als im Falle, daß man diese von hohen Werten langsam verringert. Für die Beziehung zwischen dem Oberflächenfluß, dem kritischen Winkel θ_C und dem tatsächlichen Winkel θ , den das Granulat an der Oberfläche mit der Horizontalen bildet, wurde folgendes Potenzgesetz gefunden:

$$J \propto (\theta - \theta_C)^m, \quad (7.1)$$

wobei für m der Wert 0.5 gefunden wurde. In [145] wurde ein sehr interessantes und anspruchsvolles Experiment benutzt, um die Kollisionsraten zwischen Partikeln zu bestimmen, wobei gefunden wurde, daß diese zum Massenschwerpunkt des Granulates

hin ansteigt. Hierbei wurden kleine Meßapparaturen benutzt, die die gleiche Größe wie die untersuchten Partikel hatten und mit denen Druckdifferenzen gemessen werden konnten. Ebenso enthielten sie kleine integrierte Schaltkreise, die in der Lage waren, Daten zu speichern oder aber kleine Sender, die die Daten an einen externen Computer übertrugen.

Von den bisher verwendeten numerischen Methoden sind zwei hervorzuheben. Die erste Methode beruht auf einem 1992 von Jullien, Meakin und Pawlowitch eingeführten Verfahren zur Simulation von geschüttelten granularen Medien [88]. Ein Simulationsschritt besteht darin, daß für jedes Teilchen, das "lokale" Energieminimum bestimmt wird. Dabei wird in der geometrischen Nachbarschaft die Position für das Teilchen bestimmt, bei der sein Massenschwerpunkt möglichst niedrig ist. Sobald es dieses gefunden hat, wird es in diesem Schritt nicht mehr bewegt und bleibt dort stecken. Diese Methode ist zwar sehr schnell, hat aber den Nachteil, daß die Elastizität und die Trägheit der Teilchen nicht mitberücksichtigt werden und nur Partikel mit einem Restitutionskoeffizient von 0 behandelt werden können, also vollkommen unelastische Teilchen. Baumann et al. fanden 1993 mit einem Modell, das auf dieser Methode aufbaut, ein sehr interessantes, auch experimentell gefundenes Phänomen: Zwei Sorten sphärischer Teilchen unterschiedlicher Größe, die zu Beginn der Simulation vollkommen gemischt sind, entmischen sich nach nur ein oder zwei Trommelumdrehungen. Dabei versammeln sich die kleineren Teilchen nahe des Zentrums der Trommel.

Die zweite Methode, die in numerischen Arbeiten, wie auch in der hier vorliegenden, verwendet wurde, ist die Molekular-Dynamik Methode. Ristow benutzte eine Simulation mit sphärischen Partikeln zweier verschiedener Größen und fand ebenfalls teilweise Entmischung eines ursprünglich vollständig gemischten Systems [142]. Pöschel und Buchholz verwendeten 1993 ein Modell, in dem Partikel aus fünf durch Federn miteinander verbundenen sphärischen Teilchen aufgebaut wurden [129]. Sie fanden ein Potenzgesetz, allerdings mit einem Exponenten von 0.8 statt 0.5 wie in den Experimenten beobachtet. Walton et al. benutzten eine dreidimensionale Molekular-Dynamik Simulation mit komplizierten Teilchen, die aus bis zu acht Kugeln zusammengesetzt waren [185]. Sie untersuchten die Übergänge vom nicht stationären zum stationären granularen Fluß und vom stationären Fluß zum Zustand, in dem die Trommel so schnell rotiert, daß das Granulat auf die Oberfläche herabregnet.

In den hier durchgeführten Simulationen wird unser in Kapitel 2 vorgestelltes Modell mit leichten Abänderungen dazu benutzt, den granularen Fluß in rotierenden Trommeln zu untersuchen. Dazu setzen wir vier dreieckige Partikel mit Hilfe von Stäben zu einem komplexen Teilchen zusammen. Die Konstruktion aus vier Dreiecken ist einer Konstruktion aus nur zwei Dreiecken gegenüber zu bevorzugen, da sie eine isotropere Struktur der Teilchen garantiert.

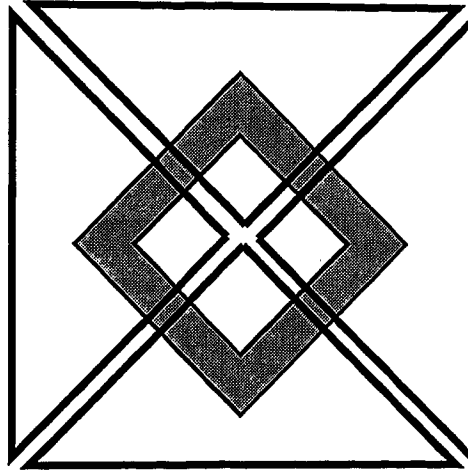


Abbildung 7.1: Die komplexen, quadratisch geformten Partikel werden aus dreieckigen Teilchen zusammengesetzt, die durch gedämpfte Stäbe miteinander verbunden sind.

7.2 Das Modell

In Abb. 7.1 ist ein Teilchen, wie es in unserer zweidimensionalen Simulation benutzt wird, dargestellt. Vier gleichschenklige Dreiecke sind mittels Stäben der Ruhelänge L_0 zu einem komplexen, quadratisch geformten Teilchen zusammengesetzt. Die Stäbe sind an den Massenschwerpunkten befestigt und stehen senkrecht auf den Grenzflächen zwischen den konstituierenden Dreiecken. In den hier durchgeführten Simulationen können die Stäbe nicht brechen. Ferner werden auch keine neuen Stäbe während der Simulation eingesetzt.

Abweichend von dem in Kapitel 2 gesagten wechselwirken die Dreiecke, die demselben komplexen Teilchen angehören, nur durch die Stäbe miteinander, also nicht durch ihre Überlappungsflächen. Weiterhin lassen wir zwischen Dreiecken, die zu verschiedenen Komplexen gehören, ausschließlich nicht-dissipative elastische Kräfte wirken. Die Glgn. (2.26) und (2.27) gelten, mit $\gamma_N = 0$ und $\gamma_T = 0$. Die elastische Kraft zwischen Dreiecken, die zu verschiedenen Komplexen gehören, werden also wieder mit Hilfe des geometrischen Überlappungspolygons berechnet. Zur Einführung von Dissipation nehmen wir an, daß Energie durch interne Schwingungen und interne Reibung innerhalb des Teilchens verloren geht. Wir modellieren diese Annahme dadurch, daß wir für die Stäbe dissipative Bewegungsgleichungen einführen. Die Dämpfung ist charakterisiert durch die Konstanten γ_L und γ_B für Verbiegungen und longitudinale Bewegungen:

$$F_{S,N} = \frac{1}{a}(x_i - x_j) - \gamma_L m_{eff}(\dot{x}_i - \dot{x}_j),$$

$$F_{S,T} = \frac{1}{b + \frac{1}{12}c} \left[y_i - y_j + \frac{1}{2}(z_j - z_i) \right],$$

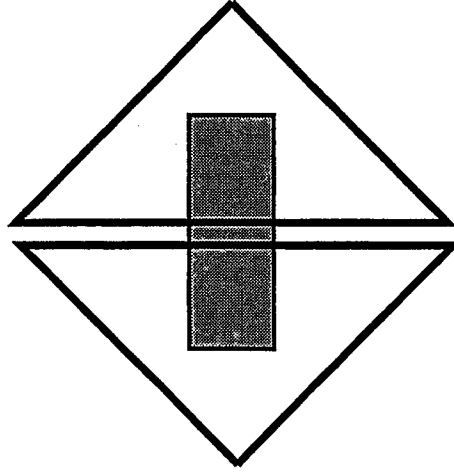


Abbildung 7.2: Die Wände sind aus Teilchen aufgebaut, die aus zwei miteinander verbundenen Dreiecken bestehen. Davon ist das eine fest in der Wand verankert.

$$\begin{aligned}
 M_{S,i} &= \frac{b}{c(b + \frac{1}{12}c)}(z_i - z_j) + \\
 &\quad \frac{1}{2(b + \frac{1}{12}c)} \left[y_i - y_j + \frac{2}{3}z_i + \frac{z_j}{3} \right] - \gamma_B m_j L_0^2 \dot{\theta}_i, \\
 M_{S,j} &= \frac{b}{c(b + \frac{1}{12}c)}(z_j - z_i) + \\
 &\quad \frac{1}{2(b + \frac{1}{12}c)} \left[y_j - y_i + \frac{2}{3}z_j + \frac{z_i}{3} \right] - \gamma_B m_i L_0^2 \dot{\theta}_j.
 \end{aligned}$$

Dabei sind θ_i und θ_j die Winkel, um die die Stäbe an ihren beiden Enden verbogen sind, m_i und m_j sind die Massen des i -ten und j -ten Teilchens und L_0 ist die Länge des Stabes im unbelasteten Zustand. Die übrigen Bezeichnungen sind im Kapitel 2 erklärt. Mit dieser Methode wollen wir das Verhalten von realen granularen Materialien nachahmen, bei denen die Partikel bei Stößen in Schwingung versetzt werden und dabei durch interne Reibungsprozesse kinetische Energie in Wärme umwandeln. Wir sind uns dabei wohl bewußt, daß dies nicht die einzige Möglichkeit der Energiedissipation ist, eine weitere besteht darin, daß Energie durch die Erzeugung von Mikrorissen verloren geht.

Die Zylinderwände werden ebenfalls aus komplexen Teilchen aufgebaut, und zwar aus zwei durch einen Stab verbundenen Dreiecken, wie in Abb. 7.2 dargestellt. Das eine der beiden Dreiecke ist fest in der Wand verankert, während das zweite sich entlang der Freiheitsgrade des Stabes bewegen kann und somit Energiedissipation auch an den Wänden möglich ist. Die von uns benutzte Trommel hat einen Durchmesser von 100 cm. Die Teilchen bestehen aus gleichschenkligen Dreiecken, deren Schenkel Längen haben, die im Intervall [0.8 cm, 1.2 cm] gleichverteilt sind.

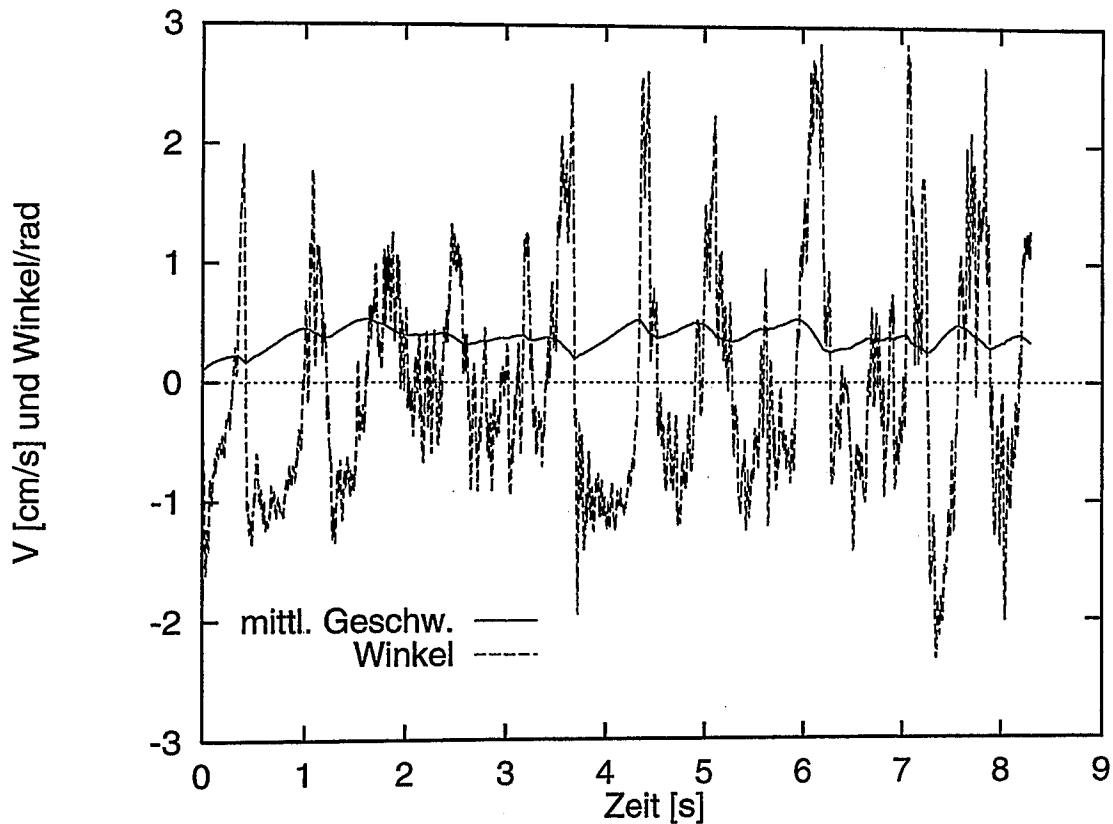


Abbildung 7.3: Mittlere Geschwindigkeit der Teilchen an der Oberfläche der Trommel gegen die Zeit, sowie der Winkel, den die Oberfläche mit der Horizontalen bildet, gegen die Zeit aufgetragen. Die Rotationsfrequenz der Trommel bei dieser Simulation beträgt 0.1 s^{-1} .

7.3 Ergebnisse

In Abb. 7.3 ist der Winkel, den die Oberfläche des Granulates mit der Horizontalen bildet, gegen die Zeit aufgetragen. Zusätzlich ist der über die Teilchen an der Oberfläche gemittelte granulare Fluß in cm/s eingezeichnet. Man erkennt, daß immer dann, wenn der Oberflächenwinkel wächst, der Oberflächenfluß gering ist. Bei hohem Fluß der Teilchen an der Oberfläche wird der Winkel kleiner. Der Oberflächenfluß unterliegt sehr hohen Fluktuationen. Bei der betrachteten Simulation mit einer Rotationsfrequenz der Trommel von 0.1 s^{-1} befinden wir uns also im Bereich des nicht-stationären granularen Flusses, der durch das Auftreten von Lawinen unterschiedlicher Größe charakterisiert ist.



Abbildung 7.4: Vier Zeitschritte der Entstehung einer Lawine. Die Graustufen stellen die Geschwindigkeiten der Teilchen dar. Die Rotationsfrequenz der Trommel beträgt 0.1 s^{-1} . Die Zeiten sind in den Grafiken angegeben.

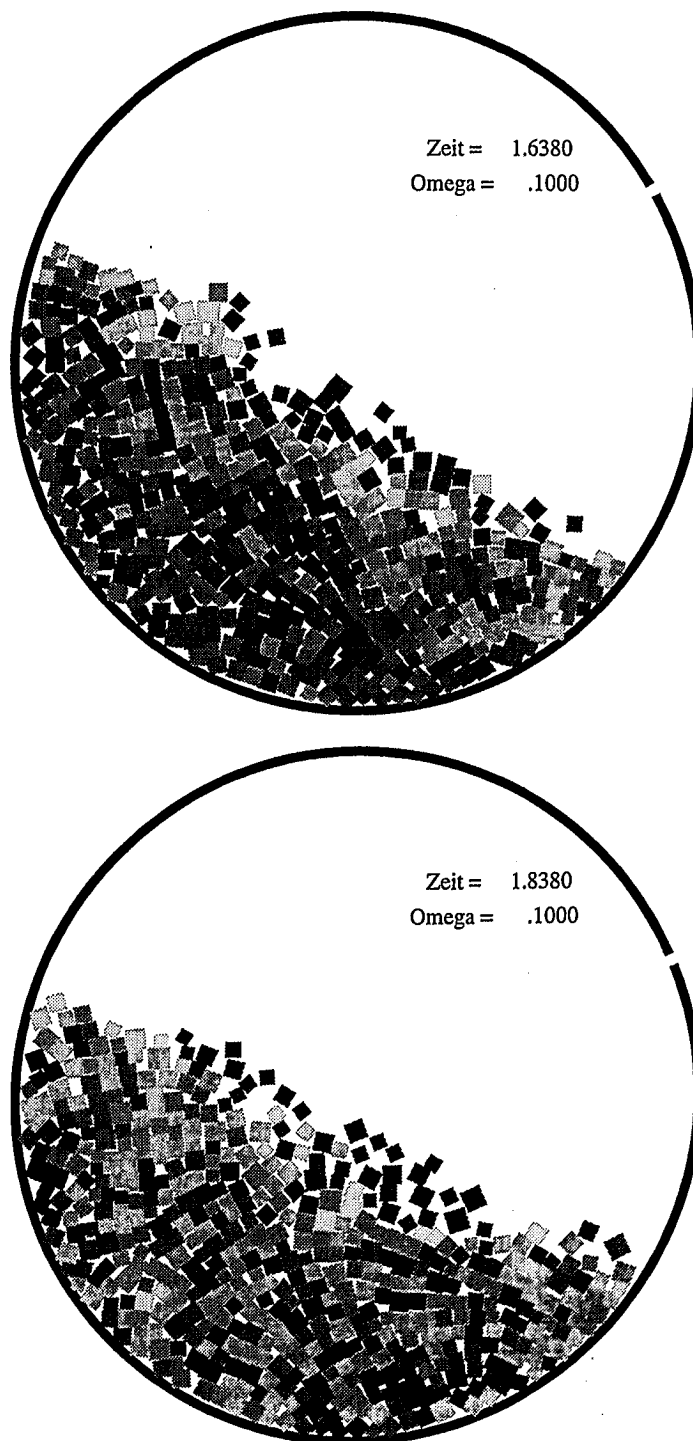


Abbildung 7.5: *Vier Zeitschritte der Entstehung einer Lawine (Fortsetzung).*

In den Abb. 7.4 und 7.5 sind vier Zeitschritte in der Entwicklung einer solchen Lawine dargestellt. Die Graustufen markieren die Geschwindigkeit der Teilchen. Dunkle Graustufen markieren Teilchen, die gerade eine hohe kinetische Energie haben und sich daher gerade schnell bewegen, während helle Graustufen Teilchen darstellen, die eine geringe kinetische Energie aufweisen. Im ersten Bild der Sequenz (Abb. 7.4 oben) erkennt man, daß sich links an der Oberfläche des Granulates ein Überhang gebildet hat, der eine Lawine auslöst. In den nächsten Bildern der Sequenz erkennt man, wie sich diese entlädt und einige Partikel entlang der Oberfläche hinabrutschen, wobei der Winkel, den die Oberfläche mit der Horizontalen bildet, verringert wird.

Der mittlere Winkel, den die Oberfläche des Granulates mit der Horizontalen bildet, beträgt bei unserer Simulation mit einer Winkelgeschwindigkeit von 0.1 s^{-1} etwa (23 ± 6) Grad. Dieser Wert liegt höher als bei bisher durchgeführten Simulationen, bei denen sphärische Partikel oder aus sphärischen Partikeln zusammengesetzte komplexe Partikel verwendet wurden [25]. Unser Wert liegt näher an typischerweise experimentell beobachteten Winkeln [137]. Wir führen diesen realistischeren Wert auf die von uns verwendeten komplexen Partikel zurück, die die sterisch bedingten Eigenschaften wirklicher Granulatkörner besser einfangen als sphärische Teilchen. Sphärische Partikel würden in einer nicht rotierenden Trommel eine Oberfläche bilden, die parallel zur Horizontalen ist, da sie an einander entlang hinabrollen können. Bei unseren polygonalen Teilchen ist dies nicht möglich, was die wirklichen Eigenschaften realer Granulate wesentlich besser widerspiegelt.

Kapitel 8

Weitere Anwendungen

In diesem Kapitel wollen wir abschließend auf zwei weitere Anwendungsmöglichkeiten eingehen. Dabei handelt es sich zunächst um die Simulation von Bruchvorgängen, die in sehr kurzer Zeit ablaufen, wie etwa dem Zertrümmern eines Betonblocks mit einem Hammer. Dabei kommen die in Kapitel 2 eingeführten Stäbe zum Einsatz. Anschließend gehen wir noch kurz auf Simulationen mit regelmäßigen Pentagonen ein. Es zeigt sich, daß unter bestimmten Bedingungen regelmäßige Kristallstrukturen gebildet werden, bei denen die dichteste mögliche Packung von regelmäßigen Pentagonen erreicht wird. Wir vergleichen dies mit experimentellen Befunden, die am größten europäischen Luftkissen-tisch an der Universität Rennes gemacht worden sind, wo ebenfalls diese Kristallstruktur beobachtet wurde. Die Untersuchungen in diesem Kapitel sind qualitativer Natur und sollen andeuten, welche anderen Möglichkeiten unser Simulationsprogramm bietet.

8.1 Bruchexperimente

Wir modellieren einen Block aus festem Material, indem wir eine rechteckige Fläche von Voronoi-Polygonen durch Stäbe miteinander verbinden. Die Stäbe sind, wie in Kapitel 2 ausführlich beschrieben, elastische Objekte mit drei Freiheitsgraden: Elongation, Scherung und Biegung. Bei einer Elongation oder einer Biegung, die einen gewissen Grenzwert t_L bzw. t_B überschreitet, bricht der Stab und wird aus der Simulation entfernt. Die beiden Zellen, die vorher durch diesen verbunden waren, verspüren nun nur noch ihre gegenseitige elastische Abstoßungskraft und Dämpfung. Ein gebrochener Stab repräsentiert die in unserem Modell kleinste darstellbare Risseinheit. Unser Modell hat gegenüber vielen anderen Methoden zur Simulation von Bruchvorgängen den Vorteil, daß es nicht auf einem Gitter aufbaut, dessen Struktur sich in den Bruchmustern wieder spiegelt. Aufgrund der verwendeten irregulär geformten Polygone ist unser Modell intrinsisch isotrop.

Wir führen dem System nun Energie zu, indem wir einer Zelle in der Mitte des unteren Randes des Blocks eine hohe Geschwindigkeit in Richtung des Innern des Blocks geben. Man kann sich etwa vorstellen, daß man ihn dort mit einem Hammer trifft. Wir

verfolgen die zeitliche Entwicklung des Systems für zwei verschiedene Geschwindigkeiten der angeregten Zelle ($V_Z = 300 \text{ m/s}$ und $V_Z = 400 \text{ m/s}$) und dem folgenden Satz von Simulationsparametern:

Parameter	Symbol	Einheiten	Wert
Dichte	ρ	$\frac{\text{g}}{\text{cm}^3}$	5
Young Modul	Y	$\frac{\text{dyn}}{\text{cm}^2}$	10^{10}
Dissipation	γ_N	s^{-1}	3000
Gleitreibung	γ_T	s^{-1}	100
Haftreibung	μ	1	0.5
Zeitschritt	dt	s	$5 \cdot 10^{-7}$
Schwellwert	t_B	1	0.03
Schwellwert	t_L	Grad	2

Drei Zeitschritte der Entwicklung dieser zwei Systeme sind in den Abb. 8.1 und 8.2 gezeigt, und zwar nach 0.0001 s, 0.001 s und 0.005 s. Nach dieser Zeit hat sich das Rißmuster vollständig ausgebildet. Die Farben kodieren die kinetische Energie der Teilchen, blaue Farbtöne entsprechen geringen, rote Farbtöne hohen kinetischen Energien. Im ersten Bild der beiden Sequenzen sieht man sehr schön, wie sich die Schallwelle ausbreitet, die erzeugt wird, wenn der Block angeregt wird. Im zweiten Bild hat sich die Rißstruktur schon fast vollständig ausgebildet. Im weiteren Verlauf bilden sich dann nur noch wenige neue Risse und man kann verfolgen, wie die entstandenen Trümmer absplintern und sich vom Block wegbewegen. Die Risse bilden verzweigte, für das Brechen von Festkörpern sehr realistisch erscheinende Strukturen [114]. Man erkennt sehr deutlich das unterschiedliche Verhalten in den beiden Simulationen. In der Sequenz, in der die Zelle im mittleren Teil des unteren Randes mit 300 m/s bewegt wird, bildet sich zwar eine Rißstruktur aus, aber der Block bleibt insgesamt versehrt. In der zweiten Simulation, wo diese Geschwindigkeit 400 m/s beträgt und die kinetische Energie dieser Zelle nahezu doppelt so groß ist, bildet sich ein durchgehender Riß und Bruchstücke splintern an der gegenüber liegenden Seite des Blocks ab.

Da für eine Bruchsimulation dieser Art nur etwa 10000 Zeitschritte nötig sind, kann man sehr leicht größere System auf einer Workstation berechnen. Systeme mit 32000 Zellen lassen sich an einem Tag berechnen. Der begrenzende Faktor ist hier eher der Speicherplatzbedarf, der bei etwa 2.5 KB pro Zelle liegt. Dieser hohe Wert ergibt sich aus der Tatsache, daß man für jedes Polygon die Koordinaten aller seiner Eckpunkte abspeichern muß. Außerdem müssen für jeden Stab die elastischen Parameter, sowie Informationen über seine Deformation gespeichert werden.

Die vorliegenden Sequenzen geben nur einen vorläufigen, qualitativen Einblick in die Möglichkeiten des Simulationsprogramms im Bereich der Bruchmechanik. Es muß noch Arbeit darin investiert werden, die Parameter an experimentelle Werte anzupassen. Trotzdem sind die vorliegenden Ergebnisse mit den realistischen Rißstrukturen sehr vielversprechend.

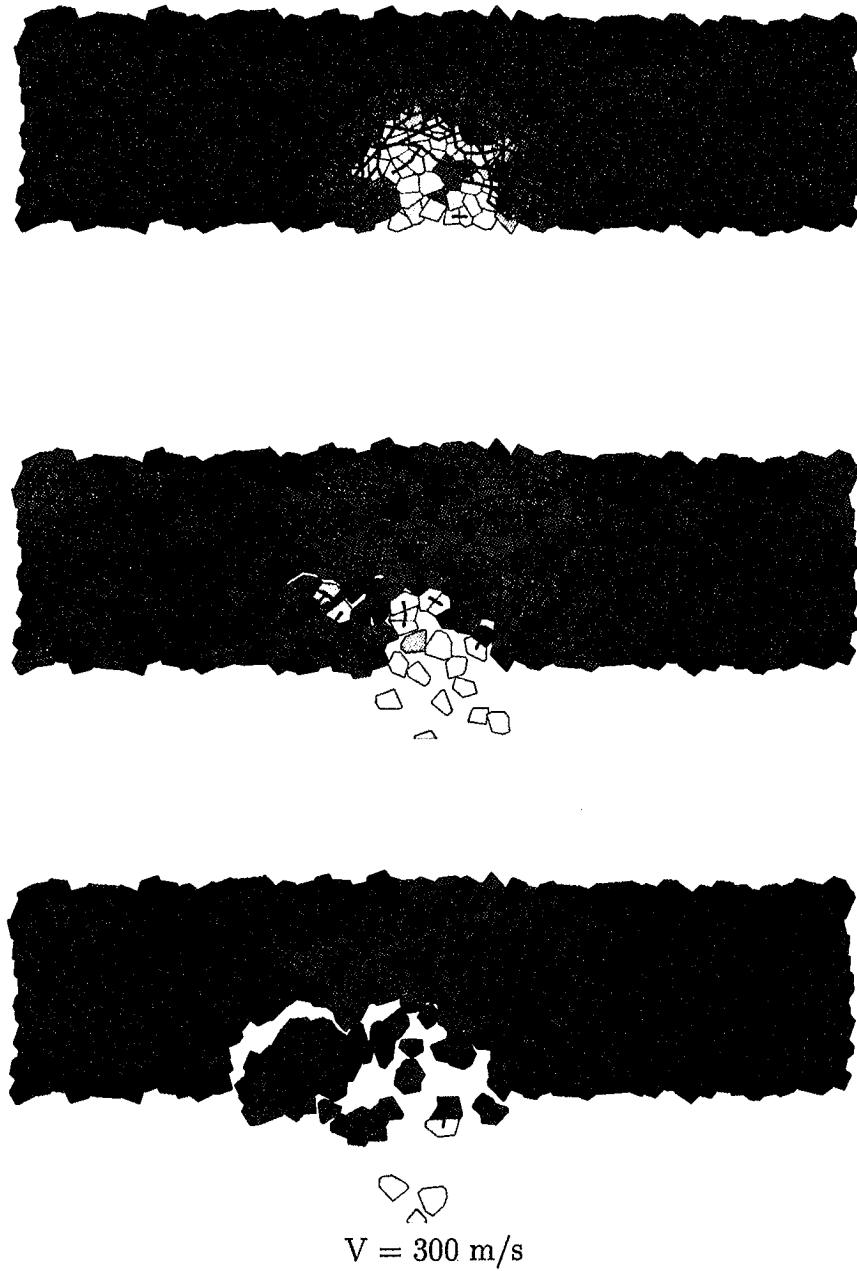


Abbildung 8.1: Bruchexperimente mit einem Betonblock, der aus durch elastische Stäbe miteinander verbundenen Zellen besteht, die bei zu starker Auslenkung brechen. Einem Partikel in der Mitte der unteren Seite wird eine Geschwindigkeit von 300 m/s gegeben, die in den Block hineingerichtet ist. Es werden drei Zeitschritte der Simulation gezeigt: oben nach $t=0.0001 \text{ s}$, Mitte nach $t=0.001 \text{ s}$ und unten nach $t=0.005 \text{ s}$.

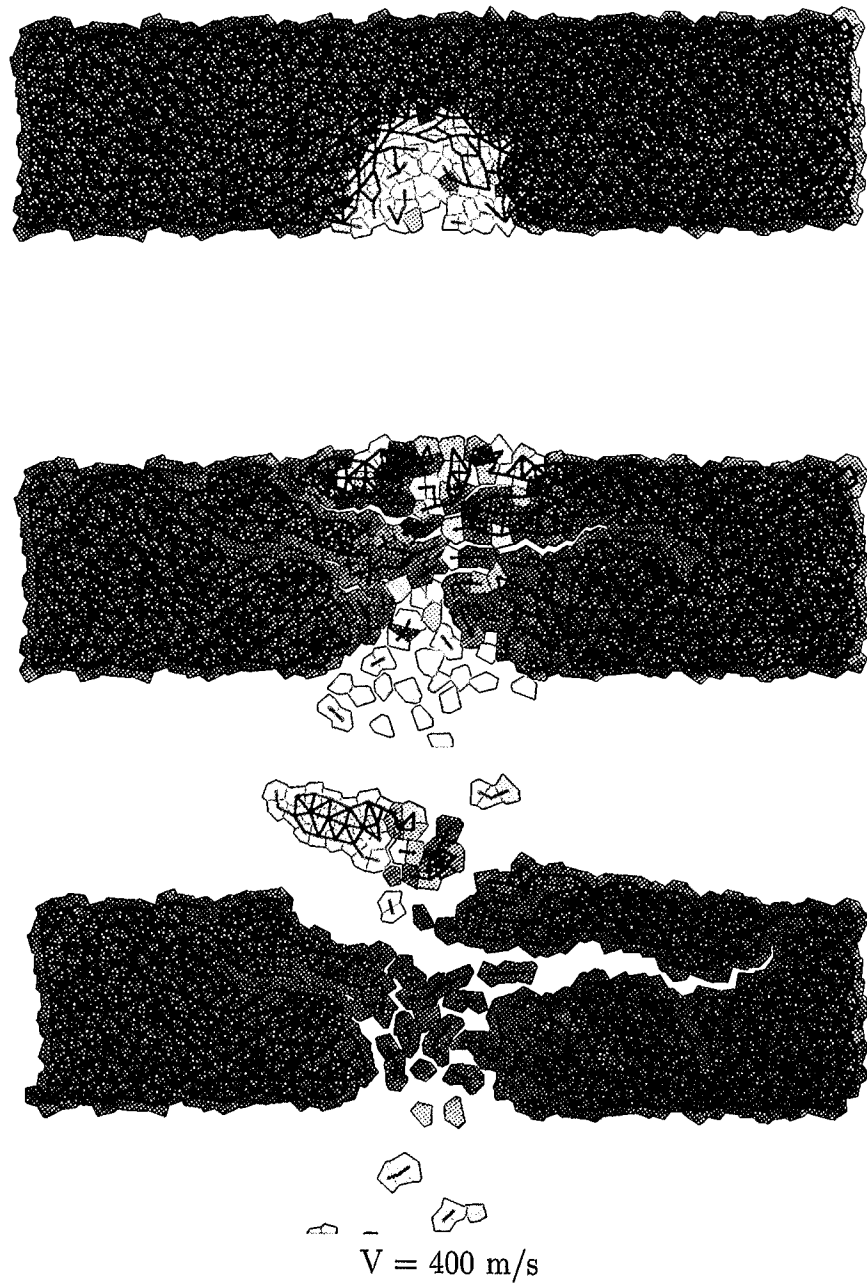


Abbildung 8.2: Das gleiche Experiment wie in Abb. 8.1, diesmal jedoch mit einer Geschwindigkeit von 400 m/s für die inwärts gerichtete Zelle. Auch hier wird die Entwicklung des Systems nach $t = 0.0001 \text{ s}$, $t = 0.001 \text{ s}$ und $t = 0.005 \text{ s}$ gezeigt. Wie man sieht, wird bei diesem Experiment der Block viel stärker zerstört und ein Riß durchdringt den gesamten Block.

8.2 Kristalle aus Pentagonen

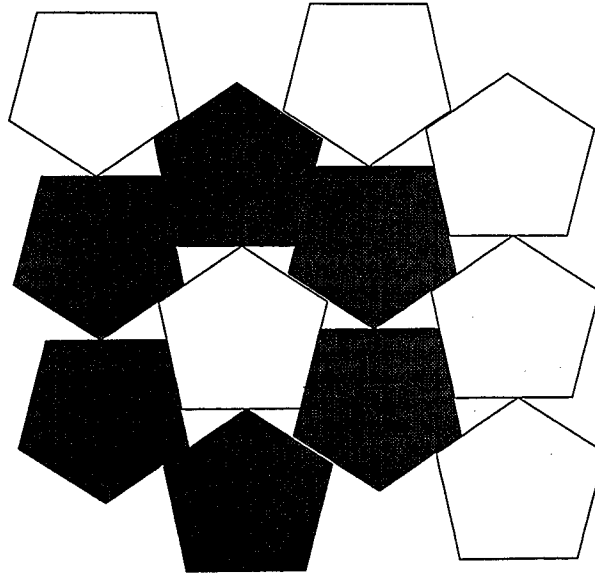


Abbildung 8.3: Die dichteste Packung, die mit regelmäßigen Pentagonen erreicht werden kann. Jedes Fünfeck berührt sechs andere Pentagonen. Es hat sechs Nachbarn (grau schattiert). Dabei hat es mit vier benachbarten Zellen Seiten-Seiten-Kontakte und mit zwei Seiten-Ecken-Kontakte.

Wir zeigen in diesem Abschnitt sowohl experimentell als auch numerisch mit Hilfe unseres Simulationsprogramms, wie sich bei regulären Pentagonen aus einer ungeordneten Anfangskonfiguration Bereiche hoher Ordnung bilden. Diese bestehen aus einer geordneten Struktur, wie sie in Abb. 8.3 dargestellt ist. Jedes der Pentagonen hat dabei sechs Nachbarn, wobei es insgesamt vier Seiten-Seiten-Kontakte und zwei Berührungen einer Seite mit einer Ecke gibt. Die bei dieser kristallinen Anordnung entstehende Packungsdichte ist 0.92. Dies ist das Maximum, welches mit regulären Pentagonen erreicht werden kann.

Experimentell wurde diese Situation auf dem großen Luftkissentisch der Universität von Rennes [102] untersucht. Dabei wurden kleine aus Polysterol bestehende Pentagonen verwendet, die etwa 1 cm groß waren. Aufgrund einer schwachen abstoßenden Kraft der Pentagonen untereinander, die dadurch zustande kommt, daß die Luft die Partikel an den Seiten umströmt, gibt es nur selten wirkliche Kontakte zwischen den Teilchen. Die Reibung der Teilchen untereinander ist dadurch vernachlässigbar klein. Der Luftstrom kann über ein Stellrad, welches die elektrische Spannung der Pumpe regelt, kontrolliert werden. Bei kleiner werdendem Luftstrom bleibt das Strömungsfeld des Tisches nicht homogen, sondern es besteht ein Gradient, der die Teilchen in Richtung der Mitte des Tisches treibt. Je weiter man also die Spannung drosselt, um so stärker wird die räumliche Anordnung der Pentagonen kompaktifiziert.

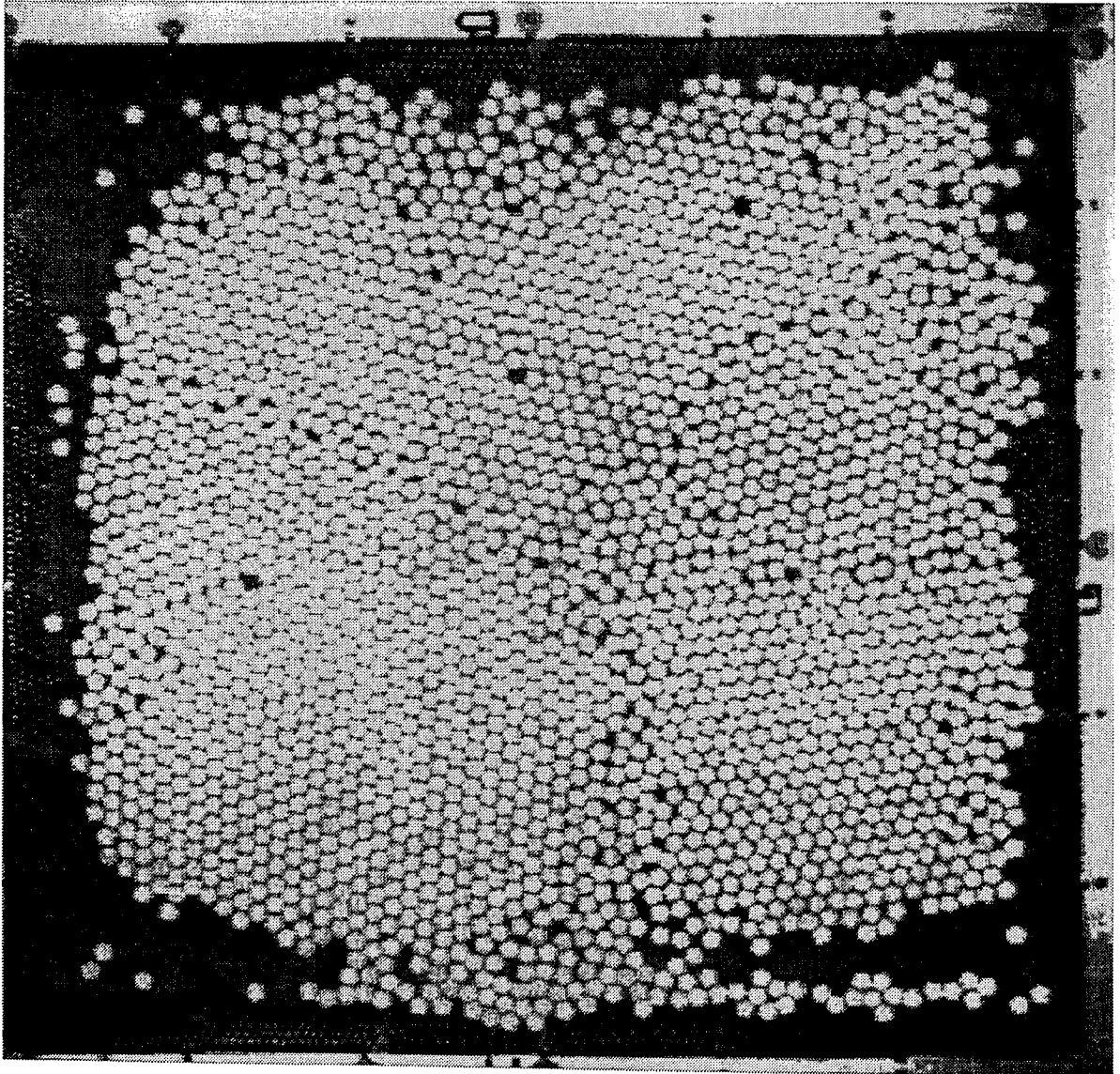


Abbildung 8.4: *Ein Experiment mit Pentagonen auf dem großen Luftkissentisch in Rennes. Es bilden sich Domänen, in denen sich Strukturen ähnlich der dichtesten Packung bilden. Wir nennen diese Kristalle.*

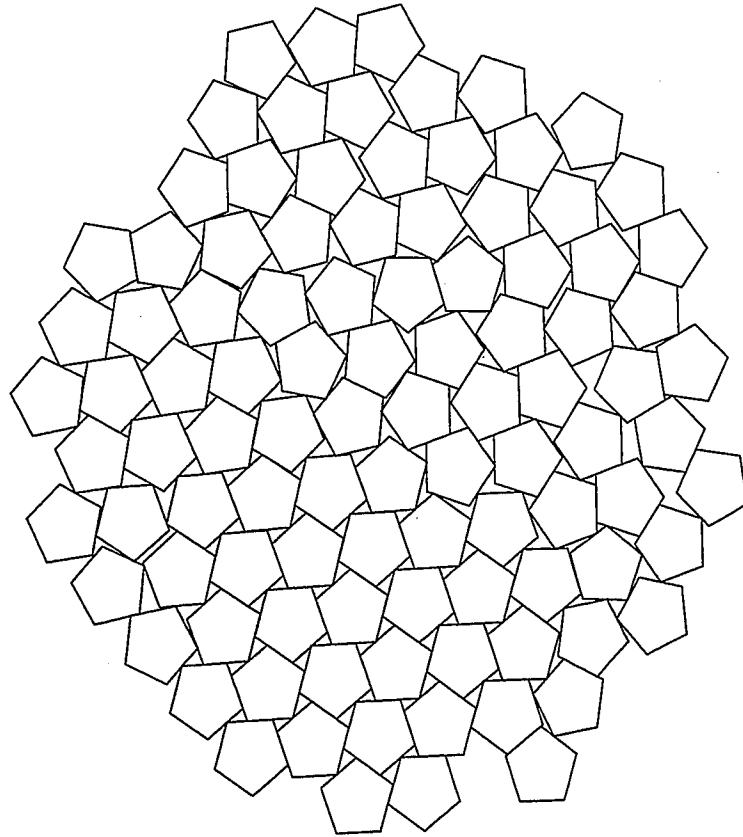


Abbildung 8.5: Die Struktur, die sich mit unserem Simulationsprogramm bildet, wenn man eine auf das Zentrum der Pentagone gerichtete Zentralkraft verwendet. Sie ist der auf dem Luftkissentisch beobachteten sehr ähnlich.

Wenn man die elektrische Spannung sehr langsam im Verlauf von etwa zwei Stunden reduziert, beobachtet man, daß sich eine kristalline Struktur wie in Abb. 8.3 bildet. Diese ist nicht ideal, sondern besteht aus mehreren kristallinen Bereichen, die voneinander getrennt sind. Außerdem gibt es Störungen in Form von Fehlstellen. Abb. 8.4 zeigt das Ergebnis eines solchen Experimentes. Wir führen dies darauf zurück, daß das System dem Zustand geringster potentieller Energie zustrebt, der gleichbedeutend ist mit der dichtest möglichen Packung. Um diesen zu erreichen, muß die Spannung sehr langsam reduziert werden, damit sich die Ordnung über das System ausbreiten kann. Wird die Spannung zu schnell reduziert, wird das System gewissermaßen abgeschreckt und die Unordnung wird eingefroren.

Wir waren in der Lage, den gleichen Effekt auch mit unserem Simulationsmodell zu finden. Wir benutzten reguläre Pentagone als Partikel und führten eine konstante, zur Mitte des Systems gerichtete Kraft ein. Die Teilchen hatten ein Elastizitätsmodul von 1 MPa und eine Normaldissipation von $\gamma_N = 100s^{-1}$. Wir wählten $\gamma_T = 0$, um den

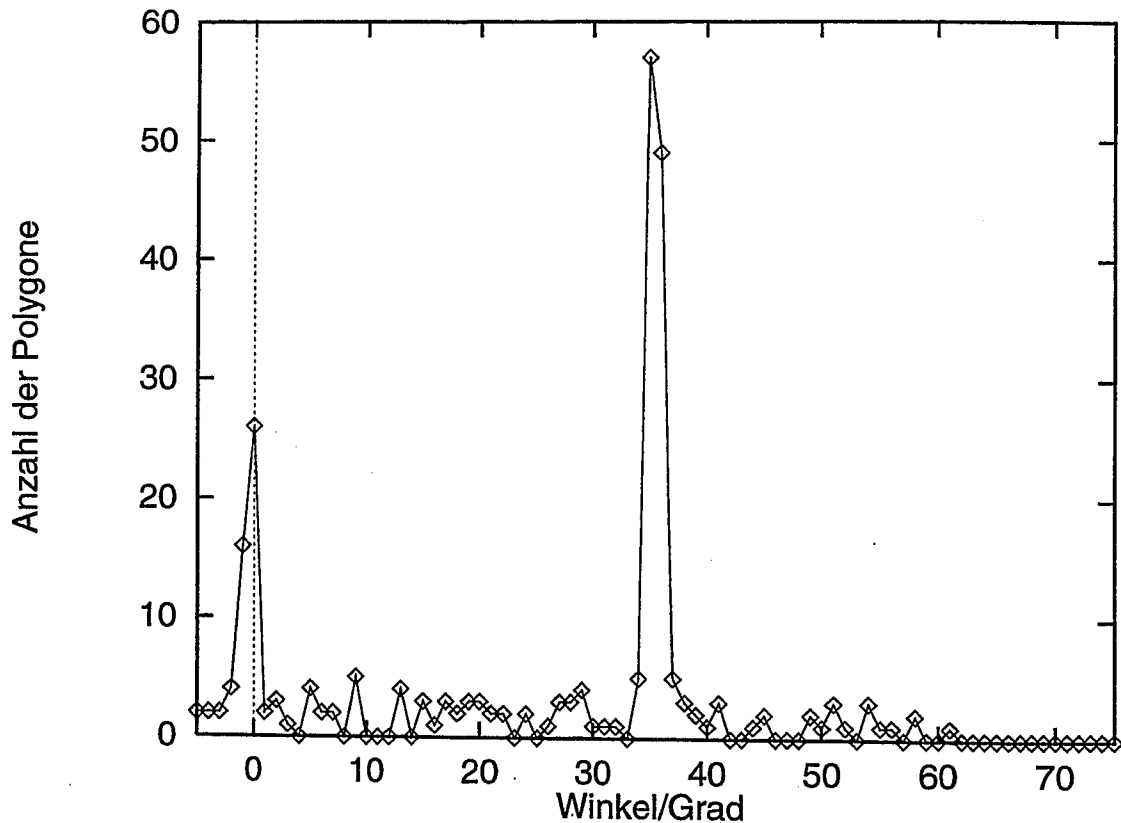


Abbildung 8.6: Die Anzahl von Kontakten gegen die relative Orientierung der beiden Pentagone. Man erkennt deutlich zwei Peaks bei 0 Grad und 36 Grad, die darauf hindeuten, daß die Struktur dem Kristall sehr ähnlich ist.

reibungslosen Kontakt zwischen den Teilchen zu simulieren. Es ergaben sich aus anfangs ungeordneten Konfigurationen Teilchenpackungen wie die in Abb. 8.5 dargestellte. Diese ist der Kristallstruktur mit der höchsten Packungsdichte sehr ähnlich, was man erkennt, wenn man ein Histogramm mit den relativen Winkeln zwischen benachbarten Teilchen aufstellt, s. Abb. 8.6. Deutlich erkennbar sind die beiden scharfen Maxima bei 0 Grad und 36 Grad. Dies sind die beiden Winkel, die einem Seiten-Seiten-Kontakt, bzw. einem Seiten-Ecke-Kontakt entsprechen.

Kapitel 9

Schlußbetrachtungen und Ausblick

In dieser Arbeit haben wir ein Simulationsprogramm vorgestellt, mit dem es möglich ist, die Bahnkurven und Rotationen von Partikeln zu berechnen, die die Gestalt von konvexen Polygonen haben. Wir berücksichtigten dabei eine elastische Wechselwirkungskraft, sowie Dissipation und Gleitreibung. Der wesentliche Vorteil dieses Modells gegenüber Molekulardynamik-Simulationen mit sphärischen Teilchen ist die Berücksichtigung von sterischen Effekten, die auf die geometrische Form der Teilchen zurückzuführen ist. Ein markantes Beispiel ist in der rotierenden Trommel der Winkel, den die Oberfläche des Granulates mit der Horizontalen bildet. Mit unserem Modell erhalten wir einen Wert, der im Bereich der experimentellen Befunde [137] liegt. Modelle, die sphärische Teilchen verwenden, liefern zu kleine Werte [129].

Zunächst präsentierten wir effiziente Algorithmen zum Erstellen von Voronoi-Zellen, die als Anfangskonfigurationen für Simulationen dicht gepackter Systeme sehr geeignet sind, da sie unregelmäßig geformte Polygone erzeugen, die die Ebene in isotroper Weise dicht parkettieren.

Mit solchen Konfigurationen führten wir Scherzellenexperimente durch, bei denen die Schergeschwindigkeit und die Normalkraft variiert wurden. Wir fanden dabei Scherhärting, d. h. ein Ansteigen der Scherkraft mit der Schergeschwindigkeit. Die Untersuchung der lokalen Rotation der Zellen führte uns zu der Beobachtung, daß sich zwei Regime in der Scherzelle ausbilden, das erste, in dem die Zellen rotieren und welches wir als Scherband bezeichnen, und das andere, in dem die Zellen weder rotieren noch sich relativ zueinander bewegen. Die Breite der Scherbänder ist sowohl von der Schergeschwindigkeit als auch von der Normalkraft abhängig. Es existiert ein Schwellwert für die Schergeschwindigkeit, oberhalb dessen die Scherbänder bis auf einen kleinen Restbereich in der Nähe der Wände die gesamte Scherzelle überspannen. An diesem Schwellwert ändert sich das Verhalten der Abhängigkeit der Scherkraft von der Schergeschwindigkeit. Unterhalb dieses Wertes steigt die Scherkraft wesentlich stärker mit der Schergeschwindigkeit an als über diesem Wert. Das gleiche Verhalten kann auch bei der Dilatanz, also bei der Ausdehnung des Systems, beobachtet werden. Die Begründung dieses Verhaltens sehen wir darin, daß die Verbreiterung der Scherzone aufgrund der sterischen Effekte durch die Teilchengometrie nur möglich ist, wenn das System dilatiert. Dilatation verursacht

ein zusätzliches Ansteigen der Scherkraft, da die Normalkraft der Dilatationsbewegung entgegenwirkt. Der Sättigungswert der Scherzonenbreite nimmt mit zunehmender Normalkraft ab, was in Übereinstimmung mit den experimentellen Untersuchungen von von Karman aus dem Jahre 1911 ist [93]. Danach beginnt Gestein plastisch zu fließen, wenn es unter sehr hoher Normalkraft gesichert wird.

Wir simulierten mit unserem Modell die Scherbewegung von zwei tektonischen Platten gegeneinander. Eine solche kommt z. B. im Westen Nordamerikas an der San-Andreas-Verwerfung vor, wo die Pazifische und die Nordamerikanische Platte aneinander entlang gleiten. Ein anderes Beispiel findet sich in den neuseeländischen Alpen. Dabei erlaubten wir keine Dilatation, wodurch die geophysikalischen Randbedingungen besser eingefangen werden. Wir fanden, daß sich eine räumlich sehr begrenzte Scherzone bildet, entlang derer sich das System bewegt. Obwohl die Zellen frei beweglich sind, bilden sich also zwei Blöcke und eine relative Verschiebung findet nur an der Scherzone statt. Dies ist in sehr guter Übereinstimmung mit den Beobachtungen an der San-Andreas-Verwerfung, wo die Scherverschiebungen ebenfalls in einem räumlich stark lokalisierten Bereich stattfinden. Für die richtige Vorhersage unseres Modells ist die durch die unregelmäßige Gestalt der Polygone bedingte sterische Reibung essentiell. Weiterhin untersuchten wir die Häufigkeitsverteilung von Erdbeben unterschiedlicher Stärke und fanden in Übereinstimmung mit dem Gutenberg-Richter Gesetz ein Verhalten, das einem Potenzgesetz gehorcht. Der Exponent dieses Gesetzes, der sogenannte b -Wert, wird von unserem Modell korrekt wiedergegeben.

Wir untersuchten eine Anwendung, die in der Verfahrenstechnik von großer Bedeutung ist, nämlich das Verhalten von granularen Medien in einem kollimierenden Trichter. Wir fanden vier Flußregime, wobei zwei durch einen Phasenübergang im Sinne der statistischen Physik voneinander getrennt waren. In einem Bereich verstopft der Trichter immer und die durchschnittliche Zeit bis zum Verstopfen wächst an, wenn man die Einfüllrate verringert. An der Phasengrenze wird diese schließlich unendlich. Auch für den Fluß von Schüttgütern durch Trichter ist die geomterische Form der Teilchen bedeutsam. Die Verstopfungen kommen nämlich durch Brückenbildungen zustande, die bei unseren eckigen Teilchen durch die Möglichkeit zu verkanten und bei Rotation ineinander zu verkeilen stark begünstigt werden.

Eine weitere sowohl ingenieurtechnisch als auch theoretisch bedeutsame Anwendung stellt die Bewegung von Schüttgut in rotierenden Trommeln dar. Dort existieren zwei phänomenologisch unterschiedliche Flußregime. Im Bereich kleiner Rotationsgeschwindigkeiten ist der granulare Fluß an der Oberfläche nicht stationär und gekennzeichnet durch das Auftreten von Lawinen. Für höhere Geschwindigkeiten wird der Fluß dagegen stationär. In unseren Simulationen lassen sich an der Oberfläche stattfindende Lawinen klar erkennen. Unser Modell gibt den Winkel, den das Granulat mit der Oberfläche bildet, in guter Übereinstimmung mit experimentellen Werten wieder. Dies ist wieder auf die Berücksichtigung von sterischen durch die Teilchenform verursachten Eigenschaften zurückzuführen.

Schließlich gaben wir noch zwei Anwendungsbeispiele an, um die vielfältigen Einsatzmöglichkeiten des Simulationsmodells anzudeuten. Zum einen handelte es sich da-

bei um die Simulation von schnellen Bruchvorgängen. Dabei stellten wir den Festkörper durch mit Stäben verbundene Voronoi-Zellen dar, die bei einer bestimmten Auslenkung brechen. Die sich ergebenden Risse sind durch eine verzweigte, sehr realistisch anmutende Struktur gekennzeichnet [114]. Der große Vorteil gegenüber anderen Bruchsimulationen, die auf einem Gitter aufbauen, ist das völlige Fehlen von Anisotropieeffekten in den Ergebnissen. Das zweite kurz angesprochene Beispiel ist die Simulation von kleinen aus Polyesterol bestehenden Fünfecken, die auf dem großen Lufttisch der Universität Rennes verwendet wurden. Sowohl in unseren Simulationen als auch im Experiment bilden sich bei Vorhandensein einer Zentralkraft eine kristalline Struktur, die die höchste für reguläre Pentagone mögliche Packungsdichte von 0.92 aufweist.

Das Modell konnte also durch Anwendung auf eine Reihe theoretischer und praktisch wichtiger Probleme auf seine Verwendbarkeit hin geprüft werden. Es wird gerade für die Bearbeitung einiger Projekte für das Institut Français du Pétrole in Solaize vorbereitet. Dabei handelt es sich um die Simulation von biaxialen Scherzellen und um die Modellierung eines chemischen Reaktors. Weiterhin kann man mit dem Programm Kompaktifizierungsvorgänge untersuchen, indem man davon Gebrauch macht, daß Stäbe eingefügt werden können. Dadurch kann man kohäsive Kräfte zwischen den Teilchen berücksichtigen. Eine interessante geophysikalische Anwendung, die ebenfalls darauf beruht, daß Stäbe eingesetzt werden können, ist die Simulation einer Scherzone ausgehend von einem Festkörper, den man durch mit Stäben verbundene Voronoi-Zellen darstellt. Beginnt man, den Block zu scheren, werden zunächst Stäbe brechen. Nach einiger Zeit werden aber an einigen Stellen neue Stäbe eingesetzt, wenn Teilchen stark zusammengepreßt werden. Es ist zu erwarten, daß sich ein Gleichgewichtszustand bildet, bei dem die pro Zeiteinheit gebrochenen Stäbe gleich der pro Zeiteinheit eingefügten Stäbe ist. Mit einem solchen Ansatz wäre es möglich, Rißnetzwerke zu untersuchen. Trotz seiner Vielseitigkeit gibt es noch eine Reihe von Erweiterungsmöglichkeiten und Schwächen.

Das Modell betrachtet die Teilchen als nicht deformierbar und ist in dieser Hinsicht Modellen unterlegen, in denen die Teilchen als deformierbar betrachtet werden. Auf der anderen Seite können durch diese Vereinfachung wesentlich mehr Teilchen simuliert werden.

Das Programm berechnet nur Teilchen in zwei Dimensionen und ein naheliegender Gedanke wäre, das Modell auf drei Dimensionen zu erweitern. Die Teilchen würden dann zu Polyedern und hätten insgesamt sechs Freiheitsgrade, drei für die Translation und drei für die Rotationen, entsprechend den drei Eulerschen Winkeln. Die benötigte Rechenzeit pro Teilchen ist aber in drei Dimensionen um etwa eine Größenordnung höher als in zwei Dimensionen, weil aus den Überlappungspolygonen nun Überlappungspolyeder werden. Um diese zu bestimmen, muß man zunächst die Schnittlinien aller Flächen des einen mit dem anderen Polyeder bestimmen. Dann sind die jeweils auf einer Seitenfläche des Schnittpolyeders liegenden Linien zum Schnitt zu bringen, um seine Eckpunkte zu berechnen. Aus diesen Informationen kann dann das Volumen des Polyeders berechnet werden. Wir schätzen, daß diese Routine um etwa eine Größenordnung langsamer ist als die entsprechende in zwei Dimensionen. Zudem muß man berücksichtigen, daß in drei Dimensionen mehr Teilchen benötigt werden, um ein gegebenes Volumen zu füllen als in

zwei Dimensionen und daß man mehr Teilchen benötigen wird, um Effekte, die durch die Systemgröße bedingt werden, ausschließen zu können. Es gibt jedoch keine prinzipiellen Schwierigkeiten, und man kann erwarten, daß in einigen Jahren Computer in der Klasse der Workstations schnell genug sein werden, um solche Probleme zu bearbeiten.

Eine weitere Möglichkeit, das Programm zu erweitern, besteht darin zwei Phasen zu berücksichtigen. Neben den Teilchen wird eine Flüssigkeit simuliert, die mit diesen wechselwirkt. Man kann dann Phänomene wie z. B. Sedimentation mit einem solchen Simulationsprogramm untersuchen. An einem solchen Programm wird im P.M.M.H. an der E.S.P.C.I., Paris gearbeitet und erste Ergebnisse liegen bereits vor [89]. Dabei wurden bisher sphärische Teilchen in zwei Dimensionen an ein Flüssigkeitsfeld angekoppelt. Es ist sowohl geplant, das Programm in zwei Dimensionen mit dem hier vorgestellten zu kombinieren, so daß man polygonale Teilchen zusammen mit einer Flüssigkeit simulieren kann. Das Programm soll auch auf drei Dimensionen erweitert werden, wobei dabei jedoch zunächst nur sphärische Teilchen in einer Flüssigkeit behandelt werden sollen. Die letzte Ausbaustufe wäre dann eine vollständige dreidimensionale Simulation beliebig geformter Teilchen in einer Flüssigkeit. Die Programmierung eines solchen Simulationswerkzeuges stellt eine große Herausforderung dar. Das Programm wäre eine große Hilfe sowohl bei vielen ingenieurtechnischen Anwendungen als auch bei wissenschaftlichen Fragestellungen, z. B. im Bereich der Geophysik oder der Ozeanographie.

Die in dieser Arbeit behandelten Themen lassen erahnen, welche Möglichkeiten Molekulardynamik-Simulationen bieten. Die Rechner werden von Jahr zu Jahr schneller und ihr Hauptspeicher verdoppelt sich alle zwei bis drei Jahre. Wir möchten daher diese Arbeit mit der optimistischen Prognose schließen, daß die oben besprochenen Erweiterungsmöglichkeiten in naher Zukunft realisiert werden können.

Anhang A

Das Simulationsprogramm

Wir wollen hier die wichtigsten Unterprogramme des Simulationsprogramms vorstellen. Wir sehen von einem vollständigen Programmlisting ab, da dies den Rahmen dieser Abhandlung sprengen würde. (Das Programm besteht aus etwa 3500 Zeilen Fortran Code, was ein Listing von etwa 70 Seiten ergibt.) Statt dessen beschränken wir uns auf die wichtigsten Routinen, die zur Berechnung der verschiedenen Kräfte, die die Partikel aufeinander ausüben, notwendig sind.

Abschnitt A.1 enthält das vom Programm zu Beginn einzulesende File mit den physikalischen Parametern und den Geometriedaten, die zum Erzeugen der Grafik notwendig sind. Die Bedeutung der Parameter ist der dritten Spalte in diesem File zu entnehmen.

Dann folgt in Abschnitt A.2 das Hauptprogramm, von welchem aus die Unterprogramme aufgerufen werden. Zunächst werden hier die Voronoi-Zellen in der Routine **vorogen** erzeugt. Anschließend werden in der Routine **voroinit** die für die Simulation wesentlichen dynamischen Größen der Partikel, wie Schwerpunkt, Masse und Trägheitsmoment berechnet. Dann wird die Routine zur Bestimmung der Nachbarschaftsrelationen **neighbours** aufgerufen. In **beaminit** werden die Stäbe, die die Partikel miteinander verbinden können, erzeugt. Das Unterprogramm **ioconfig** dient zum Herausschreiben (bzw. Einlesen) des Programmstatus nach einer bestimmten Anzahl von Zeitschritten. Die Routine **voroupdate** berechnet die Koordinaten der Eckpunkte der Zellen aus der Winkelkoordinate und den Geometriedaten der Zellen. Danach beginnt die eigentliche Simulation mit dem Unterprogramm **predictor**. Nach dem Scheren der oberen gegen die untere Wand und der Neuberechnung der Ecken in **voroupdate** und der Bestimmung der Nachbarschaftsrelationen in **neighbours** werden die geometrischen Überlappe der Zellen in **overlap** berechnet, um die elastischen und dissipativen Kräfte zu berechnen. Dann werden im Unterprogramm **beamforce** die durch die Stäbe erzeugten Normal- und Scherkräfte sowie die durch die Verbiegungen der Stäbe erzeugten Drehmomente berechnet. Die Routine **measure** schreibt die zu messenden physikalischen Observablen heraus. Im Unterprogramm **boundary** werden je nachdem, ob eine Simulation mit vertikal beweglichen oder vertikal fixierten Wänden gefahren wird, die Randbedingungen entsprechend angepaßt. Nachdem die Grafik in **graphics** erzeugt worden ist, wird die **corrector** Routine aufgerufen und anschließend werden in **beamupdate** die

lokalen Verzerrungen der Stäbe neu berechnet. Damit ist ein Zyklus beendet und das Programm setzt, nachdem in **checkend** überprüft worden ist, ob die gewünschte Anzahl an Simulationsschritten durchgeführt worden ist, mit dem nächsten **predictor**-Schritt fort.

In Abschnitt A.3 folgt die in Kapitel 2 ausführlich beschriebene Routine zur Berechnung des geometrischen Überlapps. Hier wird auch kontrolliert, wann der Druck zwischen zwei Partikeln so groß ist, daß ein neuer Stab eingesetzt werden muß. Dann folgt in Abschnitt A.4 die Routine, die die durch die Stäbe verursachten Kräfte und Momente berechnet. Die hier verwendeten Formeln für die elastischen Stäbe sind ebenfalls in Kapitel 2 erklärt worden. Schließlich wird in Abschnitt A.5 die Routine vorgestellt, die dafür sorgt, daß die lokalen Verformungen der Stäbe richtig neu berechnet werden. Hier wird auch überprüft, ob diese Verformungen, die von uns festgelegten Schwellwerte überschreiten. Falls dies der Fall ist, bricht der Stab und wird entfernt.

Bei Simulationsläufen, in denen keine Stäbe verwendet werden sollen, wie unseren Scherzellen- und Erdbebensimulationen, werden die entsprechenden Aufrufe der Unter-routinen im Hauptprogramm einfach ausdokumentiert.

A.1 Das Parameterfile

10d0	vshear	Shear velocity
100d0	gamman	dissipation
100d0	gammata	static friction
0.5d0	mue	parameter of static friction
1d8	emudul	Youngs modulus
2d-5	dt	time step
5.0d0	rho	density
0d0	gammaf	friction with floor
0	simul	0 = with/ 1 = without dilatancy
5d5	fyext	normal force on system
8	isize	x - size of system
8	jsize	y - size of system
0.8d0	aa	irregularity parameter of cells
1	seed	seed of random number generator
100	grphup	updates of graphics output
1	graph	which graphics? 0:nothing, 1:x11, 2:postscript
-2.0	grphx0	lower left x-value of graphics region (REALS!!!)
10.0	grphx1	upper right x-value of graphics region
-2.0	grphy0	lower left y-value of graphics region
10.0	grphy1	upper right y-value of graphics region
0.005d0	showen	display variable for energy representation
1	showbm	display variable for beams: 1:show beams
0	showrt	display variable for rotation: 1:show rotation
1	showtx	display variable for text: 1 show text
10000	iocfup	updates of checkpointing
0.03	threlo	breaking threshold for elongation (ratio of elongation)
4.00	thrang	breaking threshold for flexion (in degrees)
1d8	thrheal	healing pressure threshold
0.1d0	thrtim	healing time threshold

A.2 Das Hauptprogramm

```

c      *****
c      *
c      *              SHEAR
c      *
c      *****

c      include "/usr/include/cm/cmmd_fort.h"

      PROGRAM SHEAR

      implicit none

c      *) read voronoi cells
c
c      *) init (calculate mass, momentum of inertia and center of mass)
c
c --> *) calculate time step dt
c |
c | *) PREDICTOR step (new positions)
c |
c | *) Update the positions of the Voronoi cells
c |
c | *) Update the neighbour list.
c |
c | *) calculate forces ( beam forces
c |                      repulsive and shear forces of area overlap )
c |
c | *) CORRECTOR step ( forces, corrected positions )
c |
c | *) check thresholds ( brekaking and healing ), add and remove beams
c |
c |__ *) check if another step has to be done

      integer          MAXI, MAXJ, NC, ITERAT, I, J, CO, C1,
$                      ISIZE, JSIZE, numedg, bild
      parameter (MAXI = 64, MAXJ = 64 )
c      Attention!!! Please use same values in subroutine vorogen.
      parameter (NC = MAXI * MAXJ)
      parameter (numedg = 12)
      double precision Time, DT, Vshear, MR(NC), MR2(NC), MI(NC), M(NC),
$      x(nc) , vx(nc) , ax(nc) ,
$      y(nc) , vy(nc) , ay(nc) ,
$      f(nc) , vf(nc) , af(nc) ,
$      xp(nc), vxp(nc), axp(nc),
$      yp(nc), vyp(nc), ayp(nc),
$      fp(nc), vfp(nc), afp(nc),
F      XOP(NC), X1P(NC), X2P(NC), X3P(NC), X4P(NC), X5P(NC),
F      YOP(NC), Y1P(NC), Y2P(NC), Y3P(NC), Y4P(NC), Y5P(NC),
F      FOP(NC), F1P(NC), F2P(NC), F3P(NC), F4P(NC), F5P(NC),
F      FXT0, FYT0, FFT0, FFT1, KINEN, ekin(nc),
F      Leng(0:numedg,nc),
F      VorX(0:numedg,nc), VorY(0:numedg,nc), fi0(0:numedg,nc),
F      GAMMAN, EMODUL, GAMMAT, MUE, gammaf,
F      invchl(NC), rho, aa, mbot, mtop, fyext, rad(0:numedg,nc),
$      rad2(0:numedg,nc), rot(-maxj:2*maxj), dist0, impuls, oldimp
      double precision
$      oldtim, bealen(numedg,nc),
$      alpha(numedg,nc), beam1(numedg,nc), beam2(numedg,nc),
$      beam3(numedg,nc),
$      beaelo(numedg,nc), beath0(numedg,nc), beath1(numedg,nc),

```

```

$    beashr(numedg,nc),
$    xold(nc), yold(nc), threlo, thrang,
$    shown, thrheal, beatim(numedg,nc), thrtim
integer
$    VorN(NC), num(-maxj:2*maxj), beacon(numedg,nc),
F    POSI(NC), POSJ(NC), bnum(nc),
F    NumOrd(0:maxi-1,-maxj:2*maxj, 0:3),
F    NumVIJ(0:maxi-1,-maxj:2*maxj),
F    NumNb(NC), NbLIST(20, nc), simul, seed, grphup, iocfup,
$    graph, showbm, showrt, showtx
logical
$    STFLAG, bexist(numedg,nc)
real
$    grphx0, grphx1, grphy0, grphy1

c    integer mynode
CHARACTER*3    NAME(140)
CHARACTER*8    fname, cname

DATA NAME/
$    '000','001','002','003','004','005','006','007','008','009',
$    '010','011','012','013','014','015','016','017','018','019',
$    '020','021','022','023','024','025','026','027','028','029',
$    '030','031','032','033','034','035','036','037','038','039',
$    '040','041','042','043','044','045','046','047','048','049',
$    '050','051','052','053','054','055','056','057','058','059',
$    '060','061','062','063','064','065','066','067','068','069',
$    '070','071','072','073','074','075','076','077','078','079',
$    '080','081','082','083','084','085','086','087','088','089',
$    '090','091','092','093','094','095','096','097','098','099',
$    '100','101','102','103','104','105','106','107','108','109',
$    '110','111','112','113','114','115','116','117','118','119',
$    '120','121','122','123','124','125','126','127','128','129',
$    '130','131','132','133','134','135','136','137','138','139'/

C    *****
C    Are we on a workstation?
C    I = 1
C    Are we on a paragon?
C    I = MYNODE() + 1
C    Are we on a CM5?
C    I = cmmd_self_address() + 1

fname    = 'data.'//name(i)
cname    = 'conf.'//name(i)

C    *****
C    READ CONFIGURATION

open(unit=10,file= fname)
read (10,*) VSHEAR
read (10,*) GAMMAN
read (10,*) GAMMAT
read (10,*) MUE
read (10,*) EMODUL
read (10,*) DT
read (10,*) RHO
read (10,*) gammaf
read (10,*) simul
read (10,*) fyext
read (10,*) ISIZE
read (10,*) JSIZE
read (10,*) AA

```

```

read (10,*) SEED
read (10,*) grphup
read (10,*) graph
read (10,*) grphx0
read (10,*) grphx1
read (10,*) grphy0
read (10,*) grphy1
read (10,*) showne
read (10,*) showbm
read (10,*) showrt
read (10,*) showtx
read (10,*) iocfup
read (10,*) threlo
read (10,*) thrang
read (10,*) thrheal
read (10,*) thrtim
thrang = thrang * (2d0*3.141592653589d0)/(360d0)
close(10)

CCCCCCCCC Initialisation
Time = 0d0
ITERAT = 0
STFLAG = .FALSE.
bild = 0

call VOROGEN (
$   isize, jsize, aa, vorx, vory, vorn, seed, nc, numedg)
call VOROINIT (
F   X, Y, F, XP, YP, FP, MI, MR, MR2, M, NC, isize, jsize,
F   VorX, VorY, VorN, Leng, invchl, rho, mtop, mbot,
$   rad, rad2, numedg, fi0)
dist0 = y(isize*jsize) - y(1)

call NEIGHBOURS (
F   XP, YP, NC, POSI, POSJ, NumVIJ, ISIZE, JSIZE, maxi, maxj,
F   NumOrd, NumNb, NbLIST, MR, MR2, numedg )

if (time.eq.0d0) then
  do c0=1, isize*jsize
    bnum(c0)=0
    do i=1, numnb(c0)
      c1 = nblist(i,c0)
      call beaminit(
f        vorx, vory, vorn, xp, yp, c0, c1, nc, isize, jsize,
f        rad, rad2, mr, mr2, emodul, beacon, bnum,
$        bealen, beaelo, beath0, beath1, alpha, beashr,
$        beam1, beam2, beam3, rho, numedg, bexist)
    enddo
  enddo
endif

call ioconfig(
$   x, y, f, vx, vy, vf, ax, ay, af,
$   xp, yp, fp, vxp, vyp, vfp,
$   time, iterat, 'r', isize, jsize, nc, cname,
$   bnum, beaelo, beath0, beath1, alpha, beashr, numedg,
$   beam1, beam2, beam3, bealen, bexist, beacon, beatim)
call voroupdate(
F   XP, YP, FP, VorX, VorY, VorN, NC, Leng, fi0,
$   ISIZE, jsize, numedg)

call NEIGHBOURS (
F   XP, YP, NC, POSI, POSJ, NumVIJ, ISIZE, JSIZE, maxi, maxj,
F   NumOrd, NumNb, NbLIST, MR, MR2, numedg )

```

```

      if (graph.ne.0) call graphinit(1, isize, jsize, graph,
$      grphx0, grphx1, grphy0, grphy1 )

C      *****

1      continue

      call PREDICTOR (
F      X , VX , AX ,
F      Y , VY , AY ,
F      F , VF , AF ,
F      XP, VXP, AXP,
F      YP, VYP, AYP,
F      FP, VFP, AFP,
F      XOP, X1P, X2P, X3P, X4P, X5P,
F      YOP, Y1P, Y2P, Y3P, Y4P, Y5P,
F      FOP, F1P, F2P, F3P, F4P, F5P,
F      DT, ISIZE, JSIZE, NC, numedg )

CCCCCCCCCCCC Shear upper and lower wall.

      do i=1, isize
        xp(i) = xp(i) - dt*vshear
        xOp(i) = xp(i)
      enddo
      do i=(isize*jsize)-isize+1, (isize*jsize)
        xp(i) = xp(i) + dt*vshear
        xOp(i) = xp(i)
      enddo

      call VOROUUPDATE (
F      XP, YP, FP, VorX, VorY, VorN, NC, Leng, fi0,
$      ISIZE, jsize, numedg)

      if (mod(iterat,10).eq.0) then
        call NEIGHBOURS (
F      XP, YP, NC, POSI, POSJ, NumVIJ, ISIZE, JSIZE, maxi, maxj,
F      NumOrd, NumNB, NbLIST, MR, MR2, numedg )
      endif

CCCCCCCCCCCC FORCE CALCULATION

      do I=1, (isize*jsize)
        AX(I) = 0d0
        AY(I) = 0d0
        AF(I) = 0d0
      enddo

      do I=1, (isize*jsize)
        do J=1, NumNB(I)
          C0=I
          C1=NbList(J,I)
          if (c1.gt.c0) then
            goto 99
          endif
          call OVERLAP(
f      vorx, vory, vorn, xp, yp, c0, c1, nc, isize, jsize,
F      FXI0, FYI0, FFI0, FFI1,
f      invchl, vxp, vyp, vfp, m, rad, rad2,
F      mr, mr2, EMODUL, GAMMAN, GAMMAT, MUE, numedg,
$      bnum, beashr, beath0, beath1, bealen, alpha, beacon,
$      beam1, beam2, beam3, thrheal, beaelo, rho,
$      bexist, beatim, dt, thrtim)

```

```

        AX(C0) = AX(C0) + FXTO
        AY(C0) = AY(C0) + FYTO
        AF(C0) = AF(C0) + FFTO
        AX(C1) = AX(C1) - FXTO
        AY(C1) = AY(C1) - FYTO
        AF(C1) = AF(C1) + FFT1
99      CONTINUE
      enddo
100    continue
    enddo

    call beamforce (
$      nc, bnum, beacon, x, y, f, xp, yp, fp, bealen,
$      vxp, vvp, vfp, ax, ay, af, alpha, beam1, beam2, beam3, isize,
$      numedg, jsize, bexist)

    call measure (
$      isize, jsize, nc, iterat, time, ax, m, dt, y, f, posj, maxj,
$      num, rot, dist0, fname, impuls, oldimp, oldtim )
    call boundary(
$      ax, ay, af, m, nc, isize, jsize, simul, fyext,
$      mbot, mtop)

CCCCC Accelerations
do I=1, (ISIZE*JSIZE)
  AX(I) = AX(I) / M(I)
  AY(I) = AY(I) / M(I)
  AF(I) = AF(I) / MI(I)
enddo

c Dissipation caused by friction with the floor:
do i=isize+1, (isize*jsize)-isize
  ax(i) = ax(i) - gammaf * vxp(i)
  ay(i) = ay(i) - gammaf * vvp(i)
  af(i) = af(i) - gammaf * vfp(i)
enddo

if ( (mod(ITERAT,grphup).eq.0) .and. (graph.ne.0) ) then
  call graphics (
F      XP, YP, FP, VorX, VorY, VorN, NC,
$      KINEN, DT, TIME, ISIZE, JSIZE,
$      iterat, ekin, ax,
$      bnum, beath0, beath1, beacon, beaelo, beashr,
$      graph,shown, showbm, showrt, showtx,
$      grphx0, grphx1, grphy0, grphy1, numedg, bnum, nc, bexist)
endif

do i=1, isize*jsize
  xold(i) = x(i)
  yold(i) = y(i)
enddo

call CORRECTOR(
F      X , VX , AX ,
F      Y , VY , AY ,
F      F , VF , AF ,
F      XP, VXP, AXP,
F      YP, VYP, AYP,
F      FP, VFP, AFP,
F      XOP, X1P, X2P, X3P, X4P, X5P,
F      YOP, Y1P, Y2P, Y3P, Y4P, Y5P,
F      FOP, F1P, F2P, F3P, F4P, F5P,
F      ISIZE, JSIZE, NC, DT )

```

```

      call beamupdate (
$      nc, numedg, bnum, beacon, xold, yold, x, y, f,
$      alpha, beaelo, beath0, beath1, beashr, bealen, isize,
$      threlo, thrang, jsize, beam1, beam2, beam3, bexist, beatim)

      if ( mod(iterat,iocfup).eq.0 ) then
        call ioconfig(
$          x, y, f, vx, vy, vf, ax, ay, af,
$          xp, yp, fp, vxp, vvp, vfp,
$          time, iterat, 'w', isize, jsize, nc, cname,
$          bnum, beaelo, beath0, beath1, alpha, beashr, numedg,
$          beam1, beam2, beam3, bealen, bexist, beacon, beatim)
      endif

      call ENERGY(NC, VX, VY, VF, M, MI, KINEN, ekin, isize, jsize )
      call CHECKEND ( ITERAT, STFLAG, Time, DT, NC, ISIZE )
      if (STFLAG) then
        goto 10000
      endif
      goto 1

10000 CONTINUE
      if (graph.ne.0)
$      call graphinit(
$      0, isize, jsize, grphx0, grphx1, grphy0, grphy1 )

      STOP
      END

```

A.3 Berechnung des Überlappungspolygons

```

c *****
c *
c * overlap:      this subroutine determines all intersections
c *              of two voronoi cells and computes the
c *              overlap polygon.
c *              the area of this polynom, the contact line
c *              and the direction of force are also determined.
c *
c *****

c check for each link of cell a if it intersects with a link of
c cell b.
c first check if the yboxes defined by the two points of the links
c intersect. if not the links do not intersect.
c for overlapping boxes test if the links really intersect.

      subroutine overlap (
f      vorx, vory, vorn, xp, yp, c0, c1, nc, isize, jsize,
f      fxt0, fyt0, fft0, fft1,
f      invchl, vxp, vvp, vfp, m, rad, rad2,
f      mr, mr2, emodul, gamman, gammat, mue, numedg,
$      bnum, beashr, beath0, beath1, bealen, alpha, beacon,
$      beam1, beam2, beam3, thrheal, beaelo, rho,
$      bexist, beatim, dt, thrtim )

      implicit none

      integer
$      i0, i1, j0, j1, isn, isl(0:1, 32),
f      con(0:1,0:16), cp0, cp1, i, opn, vc, edge,
f      lvorn(0:1), vorn(nc), isize, jsize, j,
$      nc, c0, c1, numedg, bnum(nc), beacon(numedg,nc), beam

      double precision
$      px0, px1, vecx0, vecx1,
f      py0, py1, vecy0, vecy1,
f      det, pa0, pa1, area, swap, sx, sy, ai,
f      isx(0:32), isy(0:32),
f      fx, fy,
f      x1, x0, y1, y0, opx(18), opy(18),
f      fxt0, fyt0, fft0, fft1,
f      radx0, rady0, radx1, rady1, fnorm,
f      lvorx(0:1,0:16), lvory(0:1,0:16),
f      vorx(0:numedg, nc), vory(0:numedg, nc),
f      difpy, difpx,
f      x(0:1), y(0:1),
f      meff, emodul, gamman, GAMMAT,
f      vrelx, vrelx, mue, vreln, vrelt, fn, ft,
f      oldlen, leng,
$      xp(nc), yp(nc),
$      flag0(0:16), flag1(0:16), invchl(nc), tmpc, mi,
$      vxp(nc), vvp(nc), vfp(nc), m(nc), xx, yy, wid, tmpa, dt

      double precision
$      rad(0:numedg,nc), rad2(0:numedg,nc),
$      mr(nc), mr2(nc), midx, midy, cont, thrheal, press,
$      beam1(numedg,nc), beam2(numedg,nc), beam3(numedg,nc),
$      beashr(numedg,nc), beath0(numedg,nc), beath1(numedg,nc),
$      bealen(numedg,nc), alpha(numedg,nc), beaelo(numedg,nc),
$      rho, tmpb, beatim(numedg,nc), thrtim

      logical
$      bflag, bexist(numedg,nc)

```

```

C   Are the two guys connected with a beam?
    bflag = .false.
    do i=1, bnum(c0)
      if (beacon(i,c0).eq.c1) then
        bflag = .true.
        beam = i
        goto 1
      endif
    enddo
1   continue

c   initialisation
    isn = 0
    x(0) = xp(c0)
    y(0) = yp(c0)
    x(1) = xp(c1)
    y(1) = yp(c1)

    do i=0, numedg
      lvorx(0,i) = vorx(i,c0)
      lvery(0,i) = vary(i,c0)
      lvorx(1,i) = vorx(i,c1)
      lvery(1,i) = vary(i,c1)
    enddo

    lvorn(0) = vorn(c0)
    lvorn(1) = vorn(c1)

c   remember periodic boundary conditions.

    if (x(0)-x(1) .gt. 3d0) then
      x(1) = x(1) + dble(isize)
      do i=0, numedg
        lvorx(1,i) = vorx(i,c1) + dble(isize)
      enddo
    elseif (x(1)-x(0) .gt. 3d0) then
      x(0) = x(0) + dble(isize)
      do i=0, numedg
        lvorx(0,i) = vorx(i,c0) + dble(isize)
      enddo
    endif

CCCCCCCC Reduce the number of possible intersections

    do i=0, lvorn(0) - 1
      midx = 0.5d0 * (lvorx(0,i) + lvorx(0,i+1) )
      midy = 0.5d0 * (lvery(0,i) + lvery(0,i+1) )
      flag0(i) = (midx-x(1)) * (midx-x(1))
$      + (midy-y(1)) * (midy-y(1))
$      - rad2(i,c0) - mr2(c1) - 2d0*rad(i,c0)*mr(c1)
    enddo

    do i=0, lvorn(1) - 1
      midx = 0.5d0 * (lvorx(1,i) + lvorx(1,i+1) )
      midy = 0.5d0 * (lvery(1,i) + lvery(1,i+1) )
      flag1(i) = (midx-x(0)) * (midx-x(0))
$      + (midy-y(0)) * (midy-y(0))
$      - rad2(i,c1) - mr2(c0) - 2d0*rad(i,c1)*mr(c0)
    enddo

```

CCCCCCCC Determination of all intersection points between the two cells.

```

do i0=0, lvorn(0)-1
  if (flag0(i0).gt.0d0) goto15
  do i1=0, lvorn(1)-1
    if (flag1(i1).gt.0d0) goto10
    j0 = i0 + 1
    j1 = i1 + 1

c      determine the intersection between the both sites
c      |x|      |px0|      |vecx0|
c      |y| = |py0| + pa0 * |vecy0| , 0 <= pa0 <= 1
c      and
c      |x|      |px1|      |vecx1|
c      |y| = |py1| + pa1 * |vecy1| , 0 <= pa1 <= 1
c      if the intersection parameters are not between 0 and 1 the
c      sites do not overlap.

c      if (.not.ints(i0,i1)) then
c        goto 10
c      endif

px0 = lvorx(0, i0)
py0 = lvory(0, i0)

px1 = lvorx(1, i1)
py1 = lvory(1, i1)

vecx0 = lvorx(0, j0) - lvorx(0, i0)
vecy0 = lvory(0, j0) - lvory(0, i0)

vecx1 = lvorx(1, j1) - lvorx(1, i1)
vecy1 = lvory(1, j1) - lvory(1, i1)

det = (vecx0*vecy1 - vecy0*vecx1)
if (abs(det).eq.0d0) goto 10
det = 1d0 / det

difpy = py0-py1
difpx = px0-px1

pa0 = (vecx1*difpy - vecy1*difpx) * det
if ((pa0.le.0d0) .or. (pa0.gt.1d0)) then
  goto 10
endif

pa1 = (vecx0*difpy - vecy0*difpx) * det
if ((pa1.lt.0d0) .or. (pa1.ge.1d0)) then
  goto 10
endif

c      save new intersection point.
isn = isn + 1
isx(isn) = px0 + pa0*vecx0
isy(isn) = py0 + pa0*vecy0
c      link intersection point to sites of voronoi cells.
isl(0,isn) = i0
isl(1,isn) = i1
c      the connection number of the two sites is incremented.

con(0,i0) = con(0,i0) + 1
con(1,i1) = con(1,i1) + 1

```

```

10      continue
      end do
15      continue
      end do

20      continue

c      *****
c      computation of overlap polygon.
c      *****

      if ( isn .lt. 2 ) then
          fxt0 = 0d0
          fyt0 = 0d0
          fft0 = 0d0
          fft1 = 0d0
          goto 1000
      endif

c      *****
c      check which pair of contact points is the correct one.
c      *****

      oldlen = 0d0
      do i=1, isn
          do j=i+1, isn
              leng = (isx(i)-isx(j)) * (isx(i)-isx(j))
              + (isy(i)-isy(j)) * (isy(i)-isy(j))
$          if (leng.gt.oldlen) then
              oldlen = leng
              cp0 = i
              cp1 = j
          endif
          enddo
      enddo
      if ( oldlen.lt.1d-20 ) then
          fxt0 = 0d0
          fyt0 = 0d0
          fft0 = 0d0
          fft1 = 0d0
          goto 1000
      endif

c      put the two contact points into the right order. this means:
c      c0c1 has a counterclockwise orientation relatively to v0.
      x0 = x(1) - x(0)
      y0 = y(1) - y(0)
      x1 = (isx(cp1)-isx(cp0))
      y1 = (isy(cp1)-isy(cp0))
      det = x0*y1 - x1*y0
      if (det .lt. 0d0) then
          swap = cp0
          cp0 = cp1
          cp1 = swap
      endif

c      now only two contact points are left: they have the indices
c      cp0 and cp1. finally the overlap polygon has to be determined.

```

```

      opn      = 1
      opx(1)   = isx(cp0)
      opy(1)   = isy(cp0)

c      *****
c      determination of all sites of the overlap polygon.
c      *****

      vc = 0
      edge = isl(vc, cp0)
100  continue
      if ((isl(vc, cp1) .eq. edge) .and. (vc .eq. 0)) then
          opn = opn + 1
          vc = vc + 1
          edge = isl(vc, cp1)
          opx(opn) = isx(cp1)
          opy(opn) = isy(cp1)
      else
          opn = opn + 1
          edge = edge + 1
          if (edge .eq. lvorn(vc) ) then
              edge = 0
          endif
          opx(opn) = lvorn(vc, edge)
          opy(opn) = lvorn(vc, edge)
      endif
      if ( (vc .eq. 1) .and. (edge .eq. isl(1, cp0)) ) then
          goto 200
      else
          goto 100
      endif
200  continue

c      *****
c      * computation of the overlap area and the force point.
c      *****

      area = 0.0d0
      do i=2, opn-1
          x0 = (opx(i) - opx(1))
          y0 = (opy(i) - opy(1))
          x1 = (opx(i+1) - opx(1))
          y1 = (opy(i+1) - opy(1))
          ai = x0*y1 - x1*y0
          sx = sx + ai * (x0 + x1)
          sy = sy + ai * (y0 + y1)
          area = area + ai
      enddo

      sx = 0.5d0 * (isx(cp0)+isx(cp1))
      sy = 0.5d0 * (isy(cp0)+isy(cp1))
      fx = - ( isy(cp1) - isy(cp0) )
      fy = ( isx(cp1) - isx(cp0) )
      cont = sqrt(fx*fx + fy*fy)
      fnorm = cont
      fx = fx / fnorm
      fy = fy / fnorm
      area = 0.5d0 * area * (invchl(c0) + invchl(c1))

```

```

c *****
c * computation of friction forces
c *****

radx0 = (sx-x(0))
rady0 = (sy-y(0))

radx1 = (sx-x(1))
rady1 = (sy-y(1))

meff = (m(c0)*m(c1)) / (m(c0)+m(c1) )
vrelx = (vxp(c1)-vxp(c0)) + (-rady1*vfp(c1) + rady0*vfp(c0))
vrelx = (vyp(c1)-vyp(c0)) + ( radx1*vfp(c1) - radx0*vfp(c0))

vreln = - ( vrelx*fx + vrelx*fy )
vrelt = - (-vrelx*fy + vrelx*fx )

fn = emodul * area - gamman * meff * vreln
ft = - GAMMAT * meff * vrelt

if (abs(ft).gt.(mue*abs(fn))) then
    ft = sign( (mue*abs(fn)) , ft )
endif

c *****
c * computation of the two components of the elastic force:
c * repulsion force and shear force.
c *****

fxt0 = fn * fx - ft * fy
fyt0 = fn * fy + ft * fx
fft0 = ( radx0 * fyt0 - rady0 * fxt0)
fft1 = (- radx1 * fyt0 + rady1 * fxt0)

C *****
C * Inserting new beams.
C *****

press = area * emodul * fnorm

if (.not.bflag) then
C not yet a beam, pressure doesn't exceed thrheal
    if (press.lt.thrheal) then
        goto 1000
C not yet a beam, but pressure exceeds thrheal
    elseif (bnum(c0).lt.numedg) then
        bnum(c0) = bnum(c0)+1
        beacon(bnum(c0), c0) = c1
        bexist(bnum(c0), c0) = .false.
        beatim(bnum(c0), c0) = 0d0
        goto 1000
    endif
else
C Real beam exists
    if (bexist(beam,c0)) then
        goto 1000
    else

```

```

C      Ghost beam exists, pressure exceeds thrheal
      if (press.ge.thrheal) then
        beatim(beam, c0) = beatim(beam, c0) + dt
        if (beatim(beam, c0).gt.thrtim) then
          bexist(beam, c0) = .true.
          beacon(beam, c0) = c1
          beashr(beam, c0) = 0d0
          beaelo(beam, c0) = 0d0
          beath0(beam, c0) = 0d0
          beath1(beam, c0) = 0d0
          alpha (beam, c0) = 0d0
          xx = x(1) - x(0)
          yy = y(1) - y(0)
          bealen(beam, c0) = sqrt( xx*xx + yy*yy )
          leng = bealen(beam, c0)
          wid = cont
          area = wid * 1d0
          mi = wid**3 * 1d0 / 12d0 * ( rho * area * leng )
          tmpa = leng / ( emodul * area )
          tmpb = 2d0 * tmpa
          tmpc = leng**3 / ( emodul * mi )
          beam1(beam, c0) = 1d0 / tmpa
          beam2(beam, c0) = 1d0/2d0 / (tmpb + 1d0/12d0 * tmpc)
          beam3(beam, c0) = tmpb / (tmpc*(tmpb+1d0/12d0*tmpc))
        endif
        goto 1000
      else
C      Ghost beam exists, pressure too low: remove it
        beacon(beam, c0) = beacon(bnum(c0), c0)
        beashr(beam, c0) = beashr(bnum(c0), c0)
        beaelo(beam, c0) = beaelo(bnum(c0), c0)
        beath0(beam, c0) = beath0(bnum(c0), c0)
        beath1(beam, c0) = beath1(bnum(c0), c0)
        alpha (beam, c0) = alpha (bnum(c0), c0)
        bealen(beam, c0) = bealen(bnum(c0), c0)
        beam1 (beam, c0) = beam1 (bnum(c0), c0)
        beam2 (beam, c0) = beam2 (bnum(c0), c0)
        beam3 (beam, c0) = beam3 (bnum(c0), c0)
        beatim(beam, c0) = beatim(bnum(c0), c0)
        bexist(beam, c0) = bexist(bnum(c0), c0)
        bnum (c0) = bnum (c0) - 1
        goto 1000
      endif
    endif
  endif
1000 continue

  return
end

```

A.4 Berechnung der Stabkräfte

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   BEAMFORCE :
C   Calculation of the forces of the beams on the cells.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

```

```

subroutine BEAMFORCE(
$   nc, bnum, beacon, x, y, f, xp, yp, fp, bealen,
$   vxp, vyp, vfp, ax, ay, af, alpha, beam1, beam2, beam3, isize,
$   numedg, jsize, bexist)

implicit none

integer          numedg, c0, c1
integer          beacon(numedg,nc), nc, bnum(nc), i, isize, jsize
double precision

$   px0, py0, px1, py1, qx0, qy0, qx1, qy1, le1, le2,
$   b1x, b1y, b2x, b2y, skp, vkp, alphap, vrx, vry, valph,
$   vth0, vth1, tmpelo, tmpth0, tmpth1, fn0, fn1, m0, m1,
$   x(nc), y(nc), xp(nc), yp(nc), fp(nc), ax(nc), ay(nc), af(nc),
$   vxp(nc), vyp(nc), vfp(nc), bealen(numedg,nc),
$   fi0, fi1, alpha(numedg,nc), beam2(numedg,nc),
$   beam3(numedg,nc), ft0, ft1, tmpshr, beam1(numedg,nc),
$   gamma1, gamma2, f(nc)

logical
$   bexist(numedg,nc)

do c0=1, isize*jsize
  do i=1, bnum(c0)
    c1 = beacon(i,c0)
    if ( (c1.gt.c0) .or. (.not.bexist(i,c0)) ) goto 99
    px0 = x(c0)
    py0 = y(c0)
    px1 = x(c1)
    py1 = y(c1)
    qx0 = xp(c0)
    qy0 = yp(c0)
    qx1 = xp(c1)
    qy1 = yp(c1)
    fi0 = fp(c0)
    fi1 = fp(c1)

    if (px0-px1 .gt. 3d0) then
      px1 = px1 + dble(isize)
    elseif (px1-px0 .gt. 3d0) then
      px0 = px0 + dble(isize)
    endif

    if (qx0-qx1 .gt. 3d0) then
      qx1 = qx1 + dble(isize)
    elseif (qx1-qx0 .gt. 3d0) then
      qx0 = qx0 + dble(isize)
    endif

    le1 = sqrt( (px1-px0)*(px1-px0) + (py1-py0)*(py1-py0) )
    b1x = 1d0 / le1 * (px1-px0)
    b1y = 1d0 / le1 * (py1-py0)
    le2 = sqrt( (qx1-qx0)*(qx1-qx0) + (qy1-qy0)*(qy1-qy0) )
    b2x = 1d0 / le2 * (qx1-qx0)
    b2y = 1d0 / le2 * (qy1-qy0)
  
```

```

    skip = b1x*b2x + b1y*b2y
    vkp = b1x*b2y - b1y*b2x
    alphap = alpha(i,c0) + atan(vkp/skp)
    vrx = vxp(c1) - vxp(c0)
    vry = vyp(c1) - vyp(c0)
    valph = b2x * vry - b2y * vrx
    vth0 = vfp(c0) - valph
    vth1 = vfp(c1) - valph

    tmpelo = (le2 - bealen(i,c0))
    tmpth0 = fi0 - alphap
    tmpth1 = fi1 - alphap
    tmpshr = - (tmpth1 + tmpth0) * bealen(i,c0) / 2d0

    if (tmpelo.ge.0d0) then
        fn0 = beam1(i,c0) * tmpelo
    else
        fn0 = 0d0
    endif
    fn1 = - fn0
    ft0 = 3d0 * beam2(i,c0) * tmpshr
    ft1 = - ft0
    m0 = beam3(i,c0) * ( (4d0/3d0)*tmpth0 - (1d0/3d0)*tmpth1 )
$   + beam2(i,c0) * ( tmpth0 - tmpth1 )
$   m1 = beam3(i,c0) * ( (4d0/3d0)*tmpth1 - (1d0/3d0)*tmpth0 )
$   + beam2(i,c0) * ( tmpth1 - tmpth0 )

    vrx = vxp(c1) - vxp(c0)
    vry = vyp(c1) - vyp(c0)

    ax(c0) = ax(c0) + b2x * fn0 + gamma1 * vrx - b2y * ft0
    ay(c0) = ay(c0) + b2y * fn0 + gamma1 * vry + b2x * ft0
    af(c0) = af(c0) - m0 - gamma2 * vth0
    ax(c1) = ax(c1) - b2x * fn0 - gamma1 * vrx + b2y * ft0
    ay(c1) = ay(c1) - b2y * fn0 - gamma1 * vry - b2x * ft0
    af(c1) = af(c1) - m1 - gamma2 * vth1

99    continue
      enddo
    enddo

    return
end

```

A.5 Bewegung der Stäbe

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   BEAMUPDATE :
C   Update of the beam variables
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      subroutine BEAMUPDATE(
$      nc, numedg, bnum, beacon, xold, yold, x, y, f,
$      alpha, beaslo, beath0, beath1, beashr, bealen, isize,
$      threlo, thrang, jsize, beam1, beam2, beam3, bexist, beatim)

      implicit none

      integer          numedg, nc, c0, c1, j
      integer          beacon(numedg,nc), nc, bnum(nc), i, isize, jsize
      double precision px0, py0, px1, py1, qx0, qy0, qx1, qy1, le1, le2,
$      xold(nc), yold(nc), x(nc), y(nc), f(nc), alpha(numedg,nc),
$      b1x, b1y, b2x, b2y, beaslo(numedg,nc), beath0(numedg,nc),
$      beath1(numedg,nc), fi0, fi1, bealen(numedg,nc), skp, vkp,
$      beashr(numedg,nc), threlo, thrang, tmp,
$      beam1(numedg,nc), beam2(numedg,nc), beam3(numedg,nc),
$      beatim(numedg,nc)
      logical
$      bexist(numedg,nc)

      do c0=1, isize*jsize
        do i=1, bnum(c0)
          c1 = beacon(i,c0)
          if ( (c1.gt.c0) .or. (.not.bexist(i,c0)) ) goto 99
          px0 = xold(c0)
          py0 = yold(c0)
          px1 = xold(c1)
          py1 = yold(c1)
          qx0 = x(c0)
          qy0 = y(c0)
          qx1 = x(c1)
          qy1 = y(c1)
          fi0 = f(c0)
          fi1 = f(c1)

          if (px0-px1 .gt. 3d0) then
            px1 = px1 + dble(isize)
          elseif (px1-px0 .gt. 3d0) then
            px0 = px0 + dble(isize)
          endif

          if (qx0-qx1 .gt. 3d0) then
            qx1 = qx1 + dble(isize)
          elseif (qx1-qx0 .gt. 3d0) then
            qx0 = qx0 + dble(isize)
          endif

          le1 = sqrt( (px1-px0)*(px1-px0) + (py1-py0)*(py1-py0) )
          b1x = 1d0 / le1 * (px1-px0)
          b1y = 1d0 / le1 * (py1-py0)
          le2 = sqrt( (qx1-qx0)*(qx1-qx0) + (qy1-qy0)*(qy1-qy0) )
          b2x = 1d0 / le2 * (qx1-qx0)
          b2y = 1d0 / le2 * (qy1-qy0)
          skp = b1x*b2x + b1y*b2y

```

```

vkp = b1x*b2y - b1y*b2x
alpha(i,c0) = alpha(i,c0) + atan(vkp/skp)

beaelo(i,c0) = le2 - bealen(i,c0)
beath0(i,c0) = fi0 - alpha(i,c0)
beath1(i,c0) = fi1 - alpha(i,c0)
beashr(i,c0) = (beath0(i,c0) - beath1(i,c0) ) * le2

```

C Characteristics for breaking (Tresca and von Mises).

```

if (.not.bexist(i,c0)) goto 99
tmp = max( abs(beath0(i,c0)), abs(beath1(i,c0)) )
if ( ((beaelo(i,c0)/(bealen(i,c0)*threlo)) *
$   (beaelo(i,c0)/(bealen(i,c0)*threlo)) + abs(tmp/thrang))
$   .ge. 1d0 ) then
    j = bnum(c0)
    bnum(c0) = j - 1
    beaelo(i,c0) = beaelo(j,c0)
    beashr(i,c0) = beashr(j,c0)
    beath0(i,c0) = beath0(j,c0)
    beath1(i,c0) = beath1(j,c0)
    beam1 (i,c0) = beam1 (j,c0)
    beam2 (i,c0) = beam2 (j,c0)
    beam3 (i,c0) = beam3 (j,c0)
    alpha (i,c0) = alpha (j,c0)
    beacon(i,c0) = beacon(j,c0)
    bexist(i,c0) = bexist(j,c0)
    bealen(i,c0) = bealen(j,c0)
    beatim(i,c0) = beatim(j,c0)
endif

99      continue
      enddo
      enddo

      return
      end

```

Literaturverzeichnis

- [1] G. R. Adams and A. J. Jaeger, Journal of the South African Institute of Mining und Metallurgy, **6**, 204, 1980.
- [2] G. R. Adams, A. J. Jaeger und C. Roering, 22nd U. S. Symposium on Rock Mechanics, Massachusetts Institute of Technology, pp 213-218.
- [3] M. P. Allen und D. J. Tildesley, *Computer Simulations of Liquids*. Clarendon Press, Oxford, 1987.
- [4] G. Ahmadi und M. Shahinpoor, Powder Technology, **35**, 241, 1983.
- [5] G. Ahmadi und D. Ma, Int. J. Bulk Solids Storage in Silos, **2**, 8, 1986.
- [6] A. Alsini, I. Vardoulakis und A. Drescher, Géotechnique, **42(1)**, 395, 1992.
- [7] Z. H. Gu, P. C. Arnold und A. G. McLean, Powder. Tech., **71**, 157, 1992.
- [8] J. R. F. Arthur, T. Dunstan, Q. A. J. L. Al-Ani und A. Assadi, Géotechnique, **27**, 53, 1977.
- [9] J. R. F. Arthur und T. Dunstan, IUTAM Conf. Def. Fail. Gran. Media, Delft (Netherlands), 1982.
- [10] B. K. Atkinson, *Fracture Mechanics of Rock*, Academic Press, London, 1987.
- [11] J.-P. Avouac und P. Taponnier, C. R. Acad. Sci. Paris, **315II**, 1791, 1992.
- [12] R. A. Bagnold, Proc. Roy. Soc. A, **225**, 49, 1954.
- [13] R. A. Bagnold, Proc. Roy. Soc. A, **295**, 219, 1966.
- [14] P. Bak, C. Tang und K. Wiesenfeld, Phys. Rev. Lett., **59**, 381, 1987.
- [15] P. Benefice und J. P. Boehler, Proc. IUTAM conf. on Optical Methods in Mechanics of Solids, 1979.
- [16] C. C. Barton und P. R. LaPointe, Mémoire XX, Geol. Soc. Amer.
- [17] G. Baumann, I. Jánosi und D. E. Wolf, preprint, subm. to EPL, 1994.
- [18] Y. Ben-Zion und J. R. Rice, *Slip patterns und earthquake populations along different classes of faults in elastic solids*, subm. to Journal of Geophysical Research, 1994.
- [19] D. Bideau und A. Hansen, eds., *Disorder und Granular Media*, Elsevier Sci. Publ., Amsterdam, 1993.

- [20] A. W. Bishop, *Géotechnique*, **21**, 1971.
- [21] B. A. Bolt, *Erdbeben*. Springer-Verlag, Berlin 1984.
- [22] P. B. Bowden und S. Raha, *Phil. Mag.*, **22**, 163, 1970.
- [23] P. J. Bransby und P. M. Blair-Fish, *Géotechnique*, **25(2)**, 384, 1975.
- [24] R. L. Brown und J. C. Richard, *Principle of Powder Mechanics*, Pergamon Press, Oxford, 1966
- [25] V. Buchholtz und T. Pöschel, *Physica A*, **202**, 390, 1994.
- [26] V. Buchholz, T. Poeschel und H.-J. Tillemans, HLRZ Preprint 20/1994.
- [27] R. Burridge und L. Knopoff, *Bull. Seismol. Soc. Am.*, **57**, 341, 1967.
- [28] R. Butterfield, R. Harkness und K. Z. Undrawes, *Géotechnique*, **20(3)**, 308, 1970.
- [29] C. S. Campbell, *Proc. US-Japan Seminar on the Micromechanics of Granular material*, Sendai - Zao, Japan, 1987.
- [30] C. S. Campbell, submitted to *J. Fluid. Mech.*, 1992.
- [31] C. S. Campbell und C. E. Brennen, *J. Fluid. Mech.*, **151**, 167, 1985.
- [32] C. S. Campbell und A. Gong, *J. Fluid. Mech.*, **164**, 107, 1986.
- [33] J. M. Carlson, *Phys. Rev. A*, **44**, 10, 1991.
- [34] J. M. Carlson und J. S. Langer, *Phys. Rev. A*, **40**, 11, 1989.
- [35] J. M. Carlson, J. S. Langer, B. E. Shaw und C. Tang, *Phys. Rev. A*, **44(2)**, 884, 1991.
- [36] S. Chapman und T. Cowling, *The mathematical theory of nonuniform gases*, Cambridge University Press, Cambridge, 1970.
- [37] N. H. Christ, R. Friedberg und T. D. Lee, *Nucl. Phys. B*, **202**, 89, 1982.
- [38] S. Cilberto und C. Laroche, *subm. to Journal de Physique*, August 1993.
- [39] P. Coles, *Nature*, **346**, 446, 1990.
- [40] J. L. Colliat-Dangus, J. Desrues und P. Foray, *Advanced Triaxial Testing for Soil und Rocks - ASTM STP 977*, 1988.
- [41] P. A. Cundall und O. D. L. Strack, *Géotechnique* **29**, 47, 1979.
- [42] P. A. Cundall, *UDEC - A Generalized Distinct Element Method Program for Modeling Jointed Rock*, P. Cundall Associates, Report PCAR-1-80, European Research Office, U. S. Army, Contract DAJA37-79-C-0548, March.
- [43] P. A. Cundall, *Int. J. Rock Mech., Min. Sci. & Geomech. Abstr.*, **25**, 107, 1988.
- [44] P. A. Cundall, *Ingenieur-Archiv*, **59**, 148, 1989.
- [45] P. A. Cundall, J. T. Jenkins und I. Ishibashi, *Powders und Grains*, 1989.

- [46] P. A. Cundall und O. D. L. Strack, *Géotechnique*, **29**, 47, 1979.
- [47] J. Desrues und B. Duthilleul, *Journal de Mécanique Théorique et Appliqué*, **3(1)**, 79, 1984.
- [48] J. Desrues, *La localization de la déformation dans les matériaux granulaires. PhD thesis.*, Inst. National Polytech. de Grenoble, Grenoble, 1993.
- [49] J. Desrues, J. Lanier und P. Stutz, *Engineering fracture mechanics*, **21**, 909, 1985.
- [50] F. Donzé und S.-A. Magnier, Preprint, 1994.
- [51] D. Fincham und D. M. Heyes, *Dynamical Process in Condensed Matter – Advances in Chemical Physics*, Vol. LXIII, ed. M. W. Evans, John Wiley & Sons, New York, 1985.
- [52] W. Form, G. A. Kohring, S. Melin, H. Puhl und H.-J. Tillemans, submitted to *Comp. Meth. Appl. Mech.*, 1994.
- [53] R. Friedberg und H.-C. Ren, *Nucl. Phys. B*, **235**[FS311], 310, 1984.
- [54] M. Friedman und J. M. Logan, *Geol. Soc. America Bull.*, **84**, 1465, 1973.
- [55] D. Gapais, G. Fiquet und P. R. Cobbold, *Slip system domains, 3 New insights in fault Kinematics from plane-strain sandbox experiments*, Experiments and Numerical Modeling of Continental Deformation. Tectonophysics, 1991.
- [56] J. A. C. Gallas und S. Sokołowski, *J. Mod. Phys. B*, **7**, 2037, 1993.
- [57] G. Gavoille, J. Hubsch und S. Koutani, *J. Magn. Mat.*, **102**, 283, 1991.
- [58] C. W. Gear, Technical Report ANL 7126, Argonne National Laboratory, 1966.
- [59] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, Englewood Cliffs, New York, 1971.
- [60] W. Goldsmith, *Impact The Theory und physical behaviour of colliding solids*, Edward Arnold (publishers), London, 1964.
- [61] H. Grad, *Comm. Pure und Appl. Math.*, **2**, 331, 1949.
- [62] P. J. Grenn und R. Sibson, *The Computer Journal*, **21**, 168, 1977.
- [63] B. Gutenberg und C. F. Richter, *Ann. Geofis.*, **9**, 1, 1956.
- [64] P. K. Haff, *J. Fluid Mech.*, **134**, 401, 1983.
- [65] P. K. Haff und B. T. Werner, *Powder Techn.*, **48**, 239, 1986.
- [66] M. F. Handley, *An Investigation into the Constitutive Behaviour of Brittle Granular Media by Numerical Experiments, PhD thesis*, Faculty of the graduated school of the university of Minnesota, 1993.
- [67] T. Hashida, H. Oghikubo, H. Takahashi und T. Shoji, *International Journal of Fracture*, **59**, 227, 1993.

- [68] H. Heard, Geol. Soc. America Mem., **79**, 201, 1960.
- [69] H. J. Herrmann, A. Hansen und S. Roux, Phys. Rev. B., **39**, 637, 1989.
- [70] H. Hertz, J. f. reine u. angewandte Math., **92**, 156, 1882.
- [71] F. Heslot, T. Baumberger, B. Perrin, B. Caroli und C. Caroli, *Creep, Stick-Slip and Dry Friction Dynamics: Experiments and Heuristic Model*, Preprint 1994.
- [72] A. Hettler und I. Vardoulakis, Géotechnique, **34(2)**, 183, 1984.
- [73] A. L. Hinde und R. E. Miles, J. Statist. Comput. Simul., **10**, 205, 1980.
- [74] C. Hogue und D. E. Newlund, submitted to Powder Technology, 1993.
- [75] M. A. Hopkins und Michel Y. Louge, Phys. Fluids A, **3(1)**, 47, 1990.
- [76] K. Hui, P. K. Haff, J. E. Ungar und R. Jackson, J. Fluid Mech., **145**, 223, 1984.
- [77] M. J. Hvorslev, Proc. Intl. Conf. Soil. Mech. Found. Engng., Cambridge Mass. **2**, 125, 1936.
- [78] S. Ikeuchi und E. L. Turner, Notes Royal Astron. Soc., **250**, 519, 1991.
- [79] H. Imai, M. Iri und K. Murota, Siam. J. Comput., **14**, 93, 1985.
- [80] H. M. Jaeger, Chu-Heng Liu, S. R. Nagel und T. A. Witten, Europhys. Lett., **11**, 619, 1990.
- [81] J. T. Jenkins, P. A. Cundall und I. Ishibashi, Powders und Grains, 1989.
- [82] J. T. Jenkins und F. Mancini, J. Fluid Mech., **17**, 53, 1987.
- [83] J. T. Jenkins und M. W. Richman, Arch. Rat. Mech. Anal., **87**, 355, 1985.
- [84] J. T. Jenkins und M. W. Richman, Phys. Fluids., **28**, 3485, 1985.
- [85] J. T. Jenkins und M. W. Richman, J. Fluid Mech., **171**, 53, 1986.
- [86] J. T. Jenkins und S. B. Savage, J. Fluid. Mech., **130**, 187, 1983.
- [87] H. J. Jensen, K. Christensen und H. C. Fogedby, Phys. Rev. B, **40(10)**, 7425, 1989.
- [88] R. Jullien, P. Meakin und A. Pawlovitch, Phys. Rev. Lett., **69**, 640, 1992.
- [89] W. Kalthoff, G. Ristow, S. Schwarzer und H. J. Herrmann, *Numerical Predictions for Sedimenting Non-Brownian Particles: Hydrodynamic Diffusion and Velocity Fluctuations*, HLRZ Preprint 1994.
- [90] K. Kanatani, Int. J. Eng. Sci., **17**, 419, 1979.
- [91] K. Kanatani, Comm. Theor. und Appl. Mech., **27**, 571, 1979.
- [92] K. Kanatani, Comm. Theor. und Appl. Mech., **28**, 485, 1980.
- [93] T. von Kàrmàn, Z. Ver. Dt. Ing., **55**, 1749, 1911.
- [94] W. M. Kirkpatrick und D. J. Belshaw, Géotechnique, **18(3)**, 336, 1968.

- [95] L. Knopoff, J. A. Landoni und M. S. Abinante, *Phys. Rev. A*, **46**, 12, 1992.
- [96] L. D. Landau und E. M. Lifschitz, *Theoretische Physik, Band VII, Elastizitätstheorie*, Akademie Verlag, Berlin, 1989.
- [97] K. B. Lauritsen und C. Moukarzel, in preparation, 1993.
- [98] K. B. Lauritsen, H. Puhl und H.-J. Tillemans, submitted to *International Journal of Modern Physics C*, 1993.
- [99] G. Le Caër und J. S. Ho, *J. Phys.*, **A23**, 3279, 1990.
- [100] J. Lee, *Heap Formation in Granular Media*, HLRZ Preprint 101/92, 1992.
- [101] J. Lee und H. J. Herrmann, *Journal of Physics*, **A26**, 373, 1993.
- [102] J. Lemaitre, A. Gervois, H. Peerhossaini, D. Bideau und J. P. Troadec, *J. Phys.*, **23**, 1396, 1990.
- [103] G. Löffelmann, *Theoretische und experimentelle Untersuchungen zur Schüttgut-Wand-Wechselwirkung und zum Mischen und Entmischen von Granulaten*, Dissertation, Universität Fridericiana Karlsruhe, 1989.
- [104] S. Luding, E. Clément, A. Blumen, J. Rajchenbach und J. Duran, *Int. J. of Mod. Phys.*, **B7**, 1807, 1993.
- [105] S. Luding, *Models and Simulations of Granular Materials*, PhD thesis, Albert Ludwigs Universität, Freiburg i. Br., 1994.
- [106] S. Luding, E. Clément, A. Blumen, J. Rajchenbach und J. Duran, *Anomalous Energy Dissipation in Vibrated Granular Media: the Detachment Effect*, eingereicht bei *Phys. Rev. B*, 1994.
- [107] S. Luding, H. J. Herrmann und A. Blumen, *Scaling Behavior of 2-Dimensional Arrays of Beads under External Vibrations*, eingereicht bei *Phys. Rev. E*, 1994.
- [108] S. Luding, E. Clément, A. Blumen, J. Rajchenbach und J. Duran, *The Onset of Convection in Molecular Dynamics Simulations of Grains*, eingereicht bei *Phys. Rev. E*, Rapid Communications, 1994.
- [109] Allan Ludman und Nicholas K. Koch, *Physical Geology*, McGraw-Hill Book Company, New York, 1982.
- [110] C. K. K. Lun, S. B. Savage, D. J. Jeffrey und N. Chepurniy, *J. Fluid. Mech.*, **140**, 223, 1984.
- [111] C. K. K. Lun und S. B. Savage, *J. Fluid Mech.*, **54**, 47, 1987.
- [112] G. C. Barker, A. Mehta und M. J. Grimson., *Phys. Rev. Lett.*, **70**, 2194, 1993.
- [113] R. L. Michalowski, *Géotechnique*, **40(3)**, 389, 1990.
- [114] van Mier, persönliche Mitteilung.
- [115] G. Mitra, *Geol. Soc. Amer. Bull.*, **90**, 935, 1979.

- [116] M. Mokni, *Relations entre déformations en masse et déformations localisées dans les matériaux granulaires. PhD thesis*, UJF - INPG, Grenoble, 1992.
- [117] C. Moukarzel, *Physica A*, **190**, 13, 1992.
- [118] C. Moukarzel und H. J. Herrmann, *J. Stat. Phys.*, **68**, 911, 1992.
- [119] C. Moukarzel, *HLRZ Preprint 4/93*, 1993.
- [120] H.-B. Mühlhaus und I. Vardoulakis, *Géotechnique*, **37(3)**, 271, 1987.
- [121] A. Nadai, *Theory of flow und fracture of solids.*, McGraw-Hill, New York, 1950.
- [122] M. Nakagawa, *Preprint*, 1993.
- [123] J. Novosad, *Collection Czech. Chem. Commun.* **29**, 2697, 1964.
- [124] S. Ogawa, *Proc. U. S. - Japan Seminar on Continuum - Mechanical und Statistical Approaches in the Mechanics of Granular Materials*, 208, 1978.
- [125] S. Ogawa, A. Umemaru und N. Oshima, *J. Appl. Math. Phys.*, **31**, 483, 1980.
- [126] M. S. Paterson, *Experimental Rock Deformation. The Brittle Field*, Springer Verlag, New York, 1978.
- [127] G. Peltzer und P. Taponnier, *Journal of Geophysical Research*, **93**, 15085, 1988.
- [128] S. L. Pepke und J. M. Carlson, *Predictability of self-organized systems*, *Preprint* 1993.
- [129] T. Pöschel und V. Buchholtz, *Phys. Rev. Lett.*, **71**, 3963, 1993.
- [130] T. Pöschel und H. J. Herrmann, *Physica*, **A198**, 441, 1993
- [131] A. N. B. Poliakov und H.-J. Herrmann, *Self-organized criticality in plastic shear bands. Geophysical Research Letters (im Druck)*, 1994.
- [132] A. N. B. Poliakov, H.-J. Herrmann, Y. Podladchikov und S. Roux, *Fractal plastic shear bands. Fractals (im Druck)*, 1994.
- [133] A. V. Potapov, C. S. Campbell und M. A. Hopkins, *submitted to Comp. Meth. Appl. Mech.*, 1992.
- [134] A. V. Potapov, M. A. Hopkins und C. S. Campbell, *submitted to Comp. Meth. Appl. Mech.*, 1992.
- [135] H. Puhl, *Physica*, **A197**, 14, 1993.
- [136] *Friction und Wear of Materials*, John Wiley und Sons, New York - London - Sidney, 1965.
- [137] J. Rajchenbach, *Phys. Rev. Lett.*, **65**, 2221, 1990.
- [138] J. G. Ramsay, *J. Struct. Geol.*, **2**, 83, 1980.
- [139] S. Raynaud, D. Fabre, F. Mazerolle et al., *Tectonophysics*, **159**, 149, 1989.

- [140] M. W. Richman und C. S. Chou, J. Appl. Math. Phys., **39**, 885, 1988.
- [141] K. Rietma, *The Dynamics of Fine Powders*, Elsevier, London - New York, 1991.
- [142] G. H. Ristow, Preprint, subm. to EPL, 1994.
- [143] G. H. Ristow, F. Cantelaube-Lebec und D. Bideau, Preprint, 1994.
- [144] N. Rivier, Phil. Mag., **B52**, 795, 1985.
- [145] L. Rolf, Zement-Kalk-Gips, **46**, 117, 1993.
- [146] K. H. Roscoe, A. N. Schofield und C. P. Wroth, Géotechnique, **8**, 22, 1958.
- [147] K. H. Roscoe, Géotechnique, **20**, 129, 1970.
- [148] B. Rothkegel, *Örtliche Verteilung der Stoßenergien ≥ 22 mJ und die zugeordneten Bewegungszustände von Modellmahlkörpern in einer Modellkugelmühle*, Verlag Shaker, Aachen, 1992.
- [149] J. W. Rudnicki und J. R. Rice, J. Mech. Phys. Solids, **23**, 371, 1975.
- [150] S. B. Savage, J. Fluid. Mech., **92**, 53, 1979.
- [151] S. B. Savage und D. J. Jeffrey, J. Fluid. Mech., **110**, 225, 1981.
- [152] S. B. Savage, J. Fluid. Mech., **241**, 109, 1992.
- [153] S. B. Savage, *Marginal ice zone dynamics modelled by computer simulations involving floe collisions*. Report for the contract 91-1547/5471, prepared for the IME of the National Research Council Canada, 1992.
- [154] M. Sayed und S. B. Savage, J. Appl. Math. Phys., **34**, 84, 1983.
- [155] G. Scarpelli und D. M. Wood, Proc. IUTAM Conf. Def. Fail. Gran. Media, 1982.
- [156] J. Schäfer und D. E. Wolf *Bistability in granular flow along corrugated walls*, HLRZ Preprint, 1994.
- [157] C. H. Scholz, *The mechanics of Earthquakes und Faulting*, Cambridge University Press, New York, 1990.
- [158] J. Schwedes, *Fließverhalten von Schüttgütern in Bunkern.*, Verlag Chemie GmbH, Weinheim, 1968.
- [159] A. Sornette, P. Davy und D. Sornette, J. Geophys. Res., **98**, 121, 1993.
- [160] R. Stadler, *Schnelles Fließen von trockenen und feuchten Schüttgütern*, Dissertation, Universität Fridericiana Karlsruhe, 1986.
- [161] S. Tan und T. Fwa, Géotechnical testing journal GTJODJ, **14(3)**, 257, 1991.
- [162] F. Tatsuoka, T. Nakamura, C. C. Huang und K. Tani, Soils und Foundations, **30(1)**, 35, 1990.
- [163] P. A. Thompson und G. S. Grest, Rhys. Rev. Lett., **67**, 13, 1991.

- [164] H.-J. Tillemans und H. J. Herrmann, *Simulating deformations of granular solids under shear*, HLRZ Report 31/1994.
- [165] H.-J. Tillemans und A. Aharony, *Simulating Earthquakes using a Cellular Automaton*, HLRZ Report 50/1994.
- [166] D. Tillard, *Etude de la rupture dans les géomatériaux cohésifs. Application à la marne de beaucaire. PhD thesis.*, Université Joseph Fourier, Grenoble I, 1992.
- [167] S. P. Timoshenko und J. M. Gere, *Mechanics of Materials*, van Nordstrand Reinhold Company, New York, 1972.
- [168] Triantafyllidis und C. Aifantis, *Journal of Elasticity*, **16**, 225, 1986.
- [169] D. L. Turcotte, *Fractals and chaos in geology and geophysics.*, Cambridge Univ. Press, U. K., 1992.
- [170] F. J. Turner und L. E. Weiss, *Structural analysis of metamorphic tectonites.*, McGraw-Hill, New York, 1963.
- [171] F. Tzschichholz und H. J. Herrmann, *Simulation of Pressure Fluctuations and Acoustic Emission in Hydraulic Fracturing*, eingereicht zur Veröffentlichung in *Phys. Rev. E*, 1994.
- [172] I. Vardoulakis, *Acta Mechanica*, **32**, 35, 1979.
- [173] I. Vardoulakis, *Int. J. Num. Anal. Meth. Geom.*, **4**, 103, 1980.
- [174] I. Vardoulakis und B. Graf, *IUTAM Conf. Def. Fail. Gran. Media*, Delft (Netherlands), 1982.
- [175] P. A. Vermeer und R. de Borst, *Heron*, **29**, 1, 1984.
- [176] M. de Sousa Vieira, *Chaos in a spring-block model for earthquakes*, HLRZ preprint 1993.
- [177] M. de Sousa Vieira und H. J. Herrmann, *Self-similarity of friction laws*, HLRZ preprint 1993.
- [178] J. P. Vilotte und M. Daignières, *Journal of Geophysical Research*, **87(B13)**, 10709, 1982.
- [179] J. P. Vilotte, M. Daignières, R. Madariaga und O. C. Zienkiewicz, *Physics of the Earth and Planetary Interiors*, **36**, 236, 1984.
- [180] H. J. Vinegar, J. A. de Waal und S. L. Wellington, *Int. Rock Mech. Min. Sci. und Geomech. Abstr.*, **28(5)**, 441, 1991.
- [181] W. M. Visscher und M. Bolsterli, *Nature*, **239**, 504, 1972.
- [182] O. R. Walton, *Explicit Particle Dynamics Model for Granular Materials, Numerical Methods in Geomechanics*, Edmonton 1982, Z. Eisenstein, ed., A. A. Balkema, Rotterdam, **3**, 1261, 1982.

- [183] O. R. Walton, D. M. Maddix, T. R. Butkovich und F. E. Heuzé, Recent Advances in Mechanics of Structured Continua, **117**, 65, 1991.
- [184] O. R. Walton, Mech. of Materials, **16**, 239, 1993.
- [185] O. R. Walton und R. L. Braun, Joint DOE/NSF Workshop on flow of particulates and fluids, 1993.
- [186] D. G. Wang und Charles S. Campbell, J. Fluid. Mech., **224**, 527, 1992.
- [187] D. Wearie und N. Rivier, Contemp. Phys., **25**, 59, 1984.
- [188] G. R. Woodle und J. M. Munro, Powder Techn., **76**, 241, 1993.
- [189] O. C. Zienkiewicz, *The Finite Element Method*, McGraw-Hill, New York, 851 Seiten, 1973.

Danksagung

Wie bei jeder größeren Veröffentlichung sind am Entstehen meiner Dissertation sehr viele Menschen direkt oder indirekt beteiligt, ohne deren freundliche Unterstützung diese Arbeit nicht zustande gekommen wäre. In erster Reihe danke ich meinem Betreuer Dr. Hans-Jürgen Herrmann, der durch seine stets aufmunternde und anspornende Art mit dazu beigetragen hat, daß diese Arbeit in der kurzen Zeit von zwei Jahren fertig geworden ist. Meinem Doktorvater, Prof. Dietrich Stauffer, danke ich für viele gute Ratschläge, die sich nicht nur auf die Physik beschränkten, sondern, insbesondere während unserer gemeinsamen Dienstreise nach Israel, auch die sonstigen Dinge im Leben betrafen. Meinem langzeitigen "Room Mate" Dr. Stefan Schwarzer danke ich, daß er meine dauernden Fragen geduldig und stets freundlich bleibend ertragen hat und daß er sie zu meiner großen Verwirrung auch noch beantworten konnte. Herrn Prof. Dr. Klaus Kehr danke ich dafür, daß er die Aufgabe des Koreferenten übernommen hat und für sein Verständnis dafür, daß die Zeit am Schluß meiner Dissertation etwas knapp wurde.

Meinen Mitstreitern Harald Puhl, Wolfgang Vermöhlen, Stephan Melin und Dr. Peter Ossadnik danke ich für viele anregende Diskussionen und dafür, daß sie mich insbesondere während meiner Anfangszeit in die Geheimwissenschaft des Unix-Betriebssystems eingeweiht haben. Für viele anregende Gespräche und für viele gute Kaffeeseminare danke ich der Crew und den Gästen des HLRZ: Prof. Dr. Jason Gallas, Prof. Dietrich Wolf, Dr. Nobuyasu Ito, Dr. Gerald Ristow, Michele Zamparelli, Wolfgang Kalthoff, Dr. Cristian Moukarzel, Kent Baekegaard Lauritsen, Dr. Thorsten Pöschel, Dr. Gregory Kohring, Hans-Georg Mattutis, Prof. Dr. Stefan Sokolowski, Dr. Maria Sousa de Viera, Dr. Jysoo Lee, Dr. Michael Leibig, Jan Hemmingson, Yann Limon Duparcmeur und Dr. Renaud Delannay.

Nicht zu vergessen sind die Mitarbeiter und Mitarbeiterinnen des Zentralinstitutes für angewandte Mathematik (ZAM). Insbesondere vielen Dank an Renate Knecht, Katja Faelski, Dr. Jutta Ehrhardt und Frau Docter. Dr. Martin Frick und Dr. Malcolm Brown danke ich für die Betreuung bei der Benutzung des CM5 Supercomputers der Gesellschaft für Mathematik und Datenverarbeitung.

Nicht zuletzt danke ich meinen Eltern, Hartwin und Katharina Tillemans, sowie meiner Schwester Martina und meinem Bruder Axel für ihre verständisvolle Unterstützung. Unserer Sekretärin Helga Bongartz danke ich für die vielen gemeinsamen Kaffeeschwätzchen (ohne Physik) und Martina Kamps und Ansgar Backhaus für die Hilfe im Umgang mit den Arbeitsplatzrechnern des HLRZ.

Meinen Freunden, in erster Reihe Axel Bargende, Klaus Holthausen, Michael Putsch und Andreas Preuß danke ich für ihr Verständnis, daß ich sie in den letzten zwei Jahren so stark vernachlässigt habe. Ich gelobe Besserung.

Jül-3038
Februar 1995
ISSN 0944-2952