

**Ein Simulationssystem für granulare Aufschüttungen aus Teilchen  
variabler Form**

**Dissertation**

zur Erlangung des akademischen Grades

doctor rerum naturalium

(Dr. rer. nat)

genehmigt durch die Fakultät für Naturwissenschaften  
der Otto-von-Guericke-Universität Magdeburg

von Dipl. Phys. Alexander Schinner  
geb. am 3.12.1968 in Wasserburg am Inn

Gutachter: Prof. Dr. Klaus Kassner  
Prof. Dr.-Ing. Jürgen Tomas  
Prof. Dr. Dietrich Wolf

Eingereicht am: 25.10.2000  
Verteidigung am: 10.7.2001



# Danksagung

Eine Dissertation kann man nicht fertigstellen, ohne daß viele Leute direkt oder indirekt daran beteiligt sind.

Meinem Betreuer Prof. Klaus Kassner danke ich für die vielfältige Unterstützung meiner Arbeit, die Diskussionen mit ihm und für die Geduld, die er hatte, als ich `gc2d` entwickelte. Seine Ideen und Ratschläge haben mir sehr viel geholfen. Sein kritisches Auge hat so manchem Rohentwurf einer Veröffentlichung oder eines Vortrages erst zur endgültigen Form verholfen.

Die Zusammenarbeit mit Dr. Hans-Georg Matuttis begann, als er mich bei einem Frühstück in Regensburg auf die Idee brachte, doch über Granulare Medien zu arbeiten. Und dank Email ist die Entfernung zwischen Magdeburg und Tokio sehr klein geworden.

Im Laufe der letzten Jahre haben viele Leute das Arbeitsumfeld in unserer Arbeitsgruppe mitgestaltet und damit zur Form dieser Arbeit beigetragen. Jens "MPI" Kappey und ich haben laufend versucht, unsere jeweiligen Programme noch "effizienter" zu gestalten. Peter "Maple" Kohlert hat das Manuskript gelesen, seine Schlagfertigkeit war in Diskussionen stets eine Herausforderung. Beiden danke ich für unzählige Gespräche über Physik, Computer, den Sinn des Lebens und den Rest des Universums. Thomas Fischaleck bestand darauf, daß ich "Sandhaufen middle" und Dorothea Erndt war als Sekräterin bei vielen Formularen unentbehrlich.

Heiko Bauke war der erste, der mit dem Quellcode von `gc2d` zu arbeiten hatte. Er hat die alte Grafikbibliothek YGL durch GTK ersetzt und die Bibliothek `gsi.pm` aufgebaut. Außerdem hat er mir TeXnische Tricks beigebracht.

Dr. Stephan Mertens hat mit mir so manchen Abend zugebracht, entweder bei der Administration unseres Clusters oder in den Freien Kammerspielen. Bei den Diskussionen beim Mittagessen habe ich viel gelernt. Zusammen mit Prof. Klaus Kassner haben wir TINA, einen Beowulf-Cluster, geplant und beantragt.

Dr. Severin Thummerer danke ich für unzählige Gespräche, das Lesen des Manuskripts und vieles mehr.

Meiner Mutter danke ich für all die Unterstützung und Hilfe während des Studiums und der Promotion.

**Herzlichen Dank!**



# Inhaltsverzeichnis

Einleitung . . . . .	1
<b>1 Granulare Medien</b>	<b>3</b>
1.1 Grundlegende Eigenschaften granularer Materie . . . . .	3
1.1.1 Einführung . . . . .	3
1.2 Granulare Aufschüttungen . . . . .	6
1.2.1 Böschungswinkel . . . . .	6
1.2.2 Lawinenbildung . . . . .	9
1.2.3 Größen–Segregation . . . . .	11
1.3 Druckverteilung unter granularen Aufschüttungen . . . . .	11
1.4 Ausblick auf die Dynamik von Granulaten . . . . .	18
<b>2 Simulationenmethoden</b>	<b>21</b>
2.1 Verschiedene Methoden zur Simulation von Granulaten . . . . .	21
2.1.1 Abstrahierende Methoden . . . . .	22
2.1.2 Diskrete-Elemente Methode . . . . .	22
2.2 Physikalische Grundlagen der Simulation . . . . .	26
2.2.1 Gestalt und Eigenschaften der Partikel . . . . .	26
2.2.2 Kraftberechnung bei der Kollision . . . . .	27
2.2.3 Änderung der Teilchengröße . . . . .	34
2.3 Berechnung physikalischer Daten aus den Simulationsergebnissen . . . . .	35
2.3.1 Simulationsdaten . . . . .	35
2.3.2 Berechnung weiterer Größen . . . . .	36
2.3.3 Berechnung der lokalen gemittelten Dichte . . . . .	38
2.3.4 Berechnung von Spannungstensenoren aus den Kräften . . . . .	39
2.4 Methoden zur effizienten Simulation . . . . .	41
2.4.1 Grundlegende Konzepte . . . . .	41
2.4.2 Bounding–Boxes und Sortierverfahren . . . . .	42
2.4.3 Bestimmung der Partikelabstände . . . . .	45
2.4.4 Parallelisierung . . . . .	48
2.4.5 Anpassung an die Algorithmen . . . . .	52
2.5 Test der Simulationen . . . . .	53
2.5.1 Reibung . . . . .	54
2.5.2 Stoß von zwei und mehr Teilchen . . . . .	56
2.5.3 Brachystochrone . . . . .	64
2.5.4 Galton–Brett . . . . .	65
2.5.5 Maxwellscher Dämon . . . . .	65

<b>3</b>	<b>Druckverteilung in Schüttgutsäulen</b>	<b>69</b>
3.1	Grundlagen	69
3.2	Erweiterung auf verallgemeinerte Siloformen	71
3.3	Stufenförmiges Silo	72
3.4	Silo mit schrägen Wänden	74
3.5	Parabelförmiges Silo	75
<b>4</b>	<b>Granulare Aufschüttungen</b>	<b>79</b>
4.1	Geordnete Sandhaufen	79
4.2	Sandhaufen	84
4.2.1	Geschichtete Systeme	84
4.2.2	Geschüttete Systeme	88
4.3	Beeinflussung eines Sandhaufens	94
4.3.1	Vibration	94
4.3.2	Externe Kontrolle des Druckminimums	95
4.3.3	Elastische Konstanten	96
4.4	Eigenschaften kohäsiver Aufschüttungen	101
<b>5</b>	<b>Zusammenfassung und Ausblick</b>	<b>109</b>
<b>A</b>	<b>Namenskonventionen bei Dateinamen</b>	<b>111</b>
<b>B</b>	<b>Format der Eingabedateien</b>	<b>113</b>
B.1	Eingabedatei zur Simulationsbeschreibung ( <i>gsi</i> )	113
B.1.1	Einführung	113
B.1.2	Das Objekt GENERAL	114
B.1.3	Das Objekt DISPLAY	118
B.1.4	Das Objekt PARTICLE	119
B.1.5	Das Objekt WALL	123
B.1.6	Das Objekt GENERATE_BLOCK	125
B.1.7	Das Objekt DROPSOURCE	128
B.1.8	Beispiel für eine Eingabedatei	130
B.2	Eingabedateien für Kollisionen ( <i>gci</i> )	132
<b>C</b>	<b>Format der Ausgabedateien</b>	<b>133</b>
C.1	Ausgabedateien für die Systembeschreibung ( <i>gso</i> ) und die Kollisionen ( <i>gco</i> )	133
C.2	Beschreibung der Teilchengeometrie ( <i>ggf</i> )	134
C.3	Beschreibung der Kräfte ( <i>gff</i> )	134
C.4	Position und Geschwindigkeit ( <i>gpf</i> )	135
C.5	Zeitabhängige Größen ( <i>gtf</i> )	135
C.6	Trajektorien ( <i>gpl</i> )	136
<b>D</b>	<b>Ein Präprozessor für die Eingabedateien</b>	<b>137</b>
D.1	Das Perl-Interface	137
D.2	Makrobibliotheken	140
D.3	Kreise	140
D.4	Ellipsen	141
D.5	Rechtecke	142
D.6	Reguläre Haufen	142
D.7	Rotierende Trommeln	143
D.8	Einlesen von <i>XFig</i> -Dateien	144

---

<b>E</b>	<b>Interna von gc2d</b>	<b>145</b>
E.1	Kommandozeilenoptionen von gc2d . . . . .	145
E.2	Konfiguration bei der Kompilierung . . . . .	145
E.3	Graphische Übersicht der Funktionsaufrufe . . . . .	146
E.4	Strukturen . . . . .	146
<b>F</b>	<b>Gear Predictor-Corrector</b>	<b>151</b>



Ein Sandkorn wurde in ein Haus geweht.  
Es liegt dort in einer dunklen Ecke hinter einem Schrankbein,  
unerwischt vom Putzlappen der Hausfrau.  
Denkt nicht, daß es so still daliegt, wie es euch erscheint;  
in ihm rasen Atome, umkreisen Elektronen die Atomkerne.  
Wehte der Wind das Sandkorn in den Winkel,  
damit ihm die Gelegenheit wurde,  
Erkenntnisse zu sammeln?

Erwin Strittmatter *Der Wundertäter*



# Einleitung

Granulare Materialien, deren prominentester Vertreter Sand ist, sind in sehr vielen Forschungsbereichen von Bedeutung. Ihr besonderen Eigenschaften machen sie sowohl für die industrielle Anwendung bedeutsam, als auch als Arbeitsgebiet in der Grundlagenforschung. Die vorliegende Arbeit befaßt sich mit der numerischen Untersuchung von Granulaten.

Die Größenskala typischer granularer Teilchen beginnt im  $\mu\text{m}$ -Bereich für feine Stäube. Die obere Grenze liegt etwa im Bereich von einigen Kilometern Teilchendurchmesser für die Felsbrocken in den Ringen des Saturn [1]. Obwohl man im allgemeinen bei Granulaten immer an große Zahlen von Partikeln denkt, so ist dies nicht notwendig, um interessante Systeme zu erhalten. Schon die Untersuchung der Dynamik eines einzelnen Teilchens in Kontakt mit einer Wand [2–5] kann interessante Erkenntnisse liefern. Einige Dutzend Murmeln in einer Petrischale zeigen schon neuartige Verhaltensweisen wie die sog. Reptationsbewegung [5–8]. Zur Untersuchung komplexerer Systeme mit Hilfe einer Computersimulation ist es notwendig, hinreichend große Teilchenzahlen berechnen zu können. Hier ist es dann sinnvoll, durch die Implementierung optimierter Algorithmen die vorhandene Rechenzeit besser zu nutzen. Von besonderer Bedeutung ist auch die Form der Teilchen in einem granularen Medium. Runde, glatte Dragees in einer Pharmafirma haben sicher andere Eigenschaften als die rauen Erzbrocken im Bergbau. Deswegen wurde in dem für diese Arbeit erstellten Simulationsprogramm auch die Gestalt der Teilchen mit berücksichtigt. Ein noch komplizierteres Verhalten kann entstehen, wenn Granulate noch mit Flüssigkeiten [9–12] oder Gasen [13] wechselwirken. Ebenso hat die Form der Wechselwirkung zwischen zwei Partikeln einen Einfluß auf das Verhalten des Gesamtsystems. Kommt zur Kraft aufgrund von Kollisionen noch eine weitere anziehende Kraft [14] hinzu, kann man den Einfluß von Kohäsion untersuchen.

Die Arbeit gliedert sich folgendermaßen: zuerst wird im Kapitel 1 auf diejenigen grundlegenden Eigenschaften von Granulaten eingegangen, die für die untersuchten Systeme von Bedeutung sind. Im darauf folgenden Kapitel 2 werden die verwendeten Simulationsmethoden vorgestellt und erläutert. Dies umfaßt die physikalische Modellierung der Kontakte, die Auswertung der Simulationsergebnisse, den Test der Numerik und die verwendeten Strategien, um die Simulation in akzeptabler Zeit durchzuführen. Die beiden folgenden Abschnitte befassen sich dann mit der Untersuchung zweier verschiedener Systeme. Kapitel 3 beschäftigt sich mit Schüttgutsäulen; hier wird eine Erweiterung der klassischen Theorie von Janssen vorgestellt und mit Simulationen verglichen. Im Kapitel 4 werden dann Simulationsergebnisse beschrieben, die einen Einblick in Sandhaufen ermöglichen. Es werden für die Theorien interessante Größen untersucht. Ebenso wird versucht, den Blick auf eine bisher nicht berücksichtigte Größe, die lokale Dichte, zu lenken. Abschließend werden die Ergebnisse der Arbeit nochmals zusammengefaßt und im Hinblick auf zukünftige Entwicklungen bewertet.

Das im Rahmen der vorliegenden Arbeit entstandene Programmsystem `gc2d` wurde so entwickelt, daß verschiedenste granulare Systeme untersucht werden können. Dieses Simulationssystem besteht aus drei Komponenten für die Beschreibung des zu simulierenden Systems, die Simulation selbst und die Auswertung der Simulationsergebnisse.

Die Vorgabe des zu berechnenden Systems über eine Beschreibungssprache soll es erleichtern, die

vielfältigen Möglichkeiten effektiv zu nutzen. Die eigentliche Simulation selbst geschieht durch ein C-Programm mit dem Namen `gc2d`<sup>1</sup>, es gab auch dem Gesamtsystem seinen Namen. Die Auswertung der Simulationsergebnisse mit **Matlab**, einem Programm für die numerische Berechnung und graphischer Darstellung, erfolgt durch selbstentwickelte Routinen. Dies, zusammen mit den verwendeten komplexen Algorithmen, führte zu einem sehr großen Umfang von mehr als 15.000 Zeilen C-Code, 3.000 Zeilen Perl-Skripten und über 5000 Zeilen **Matlab**-Routinen. Um die Nutzung des Systems auch für andere Nutzer zu ermöglichen, dient der Anhang als Programmdokumentation. Es werden in den Anhängen B und C die Formate der Ein- und Ausgabedateien beschrieben, sowie im Anhang D das Interface zur Skriptsprache Perl. Im Anhang E.3 wird ein graphischer Überblick über die verwendeten Funktionen gegeben. Die weiteren Anhänge beschreiben die verschiedenen Optionen bei der Kompilierung und beim Start des Programms sowie andere programminterne Details. Auf der beigelegten CD sind verschiedene Filme, die einen Eindruck von der vielfältigen Nutzungsmöglichkeit von `gc2d` geben. Ebenso findet sich dort eine lauffähige Version des Programms.

---

<sup>1</sup>Der Name `gc2d` ist lediglich eine Abkürzung für “**G**ranulate / **C**-Programm / **2** Dimensionen”.

# Kapitel 1

## Granulare Medien

### 1.1 Grundlegende Eigenschaften granularer Materie

#### 1.1.1 Einführung

In diesem Abschnitt soll ein Überblick über die Vielfalt der Physik granularer Systeme gegeben werden. Detaillierte Beschreibungen der Grundlagen für die in dieser Arbeit behandelten granularen Aufschüttungen folgen dann im Abschnitt 1.2.

Typische Vertreter für diese Stoffklasse sind beispielsweise Getreide, Sand, Kies, Tabletten oder Pellets. Aber auch die Ringe des Saturns [1] oder Eisschollen auf dem Meer [15, 16] können als granulares Material betrachtet werden. Die Verschiedenartigkeit der Materialien läßt erahnen, daß man bei der Untersuchung von Granulaten eine Vielzahl von Arbeitsbereichen zu berücksichtigen hat.

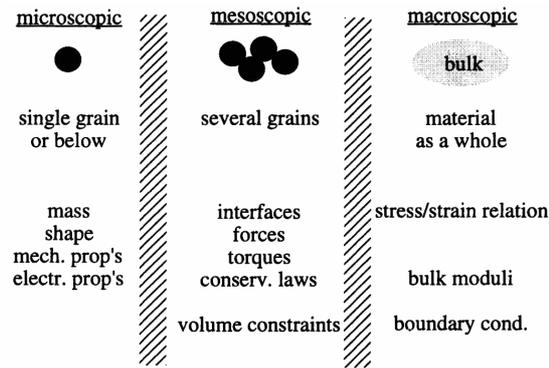
Bezeichnung	Korngrößenbereich in mm
Blöcke	über 200
Steine	über 63 bis 200
Kies	über 2 bis 63
Sand	über 0.06 bis 2.0
Schluff	über 0.002 bis 0.06
Tonkorn	unter 0.002

**Tabelle 1.1:** Korngrößenverteilung von Sand nach DIN 4022/1. Für Kies, Sand und Schluff gibt es jeweils noch Einteilungen in grob, mittel und fein.

Je nach Größe haben verschiedene Kräfte einen Einfluß auf die Partikel.

Bei sehr kleinen Partikeln (Staub, Feinkorn) ist der Einfluß von anziehenden oder abstoßenden Wechselwirkungen aufgrund von Ladungen noch von Bedeutung. Auch das die Partikel umgebende Medium muß berücksichtigt werden. Eine einfache Abschätzung, wie schnell ein derartig kleines Teilchen in Luft fällt, zeigt das Problem. Nimmt man für das Partikel die Kugelform mit  $r = 1\mu\text{m}$  und der Dichte  $\rho = 1000\frac{\text{kg}}{\text{m}^3}$  an, so gilt mit der Stokes-Reibung  $ma = -mg - cv = -\frac{4}{3}r^3\rho g + 6\pi\eta rv$ . Als quasistationäre Lösung für hinreichend lange Zeiten  $\frac{m}{c} \gg 10^{-5}\text{s}$  erhält man also  $v = \frac{mg}{c} \approx 10^{-3}\frac{\text{m}}{\text{s}}$ .

Bei etwas größeren Partikeln (typischerweise  $< 1\text{mm}$ ) kann der Einfluß der Luft in bestimmten Systemen vernachlässigt werden. Dies gilt natürlich nicht, wenn ein Luftstrom über das Granulat streicht; dabei kann es zur Bildung Dünen und Rippeln kommen.



**Abbildung 1.1:** Die physikalische Beschreibung eines granularen Systems geschieht auf mehreren Ebenen. Bei der Simulation mit `gc2d` werden Eigenschaften der mikroskopischen Ebene (Masse, Form, etc.) vorgegeben. Die Simulation liefert als Ergebnisse Daten im mikroskopischen Bereich (z.B. Kräfte). Durch Mittelung über kleine/große Volumina, erhält man dann meso-/makroskopische Größen. Durch die in Abschnitt 2.3 beschriebene Auswertung kann man dann beispielsweise die Spannungstensoren bestimmen. (aus [22])

Die Bagnold-Zahl [17]

$$Ba = \sqrt{\lambda} \frac{\rho_{ku} d^2 \frac{\partial u}{\partial y}}{\eta_{fl}} \quad (1.1)$$

gibt Auskunft darüber, ob man das Granulat als reine Flüssigkeit mit einer korrigierten Viskosität betrachten kann. Dabei beschreibt  $\lambda$  die Konzentration der Teilchen in einer Flüssigkeit mit der dynamischen Viskosität  $\eta_{fl}$ .  $\frac{\partial u}{\partial y}$  ist die vorgegebene Scherrate,  $\rho_{ku}$  die Dichte und  $d$  der Durchmesser der Partikel. Für Bagnoldzahlen kleiner als 40 dominiert die Fluidviskosität, im Bereich größer als 450 dominiert die Kollision der Körner. Teilchen dieser Größe können aber noch durch elektrostatische Kräfte beeinflusst werden. Untersuchungen von Scheffler [18] befassen sich mit den dabei entstehenden Fragestellungen.

Bei Teilchen im Bereich von Millimetern bis hin zu vielen Metern können langreichweitige Wechselwirkungen im allgemeinen vernachlässigt<sup>1</sup> werden. Dies ist der Bereich, für den die in dieser Arbeit vorgestellte Simulation konzipiert wurde. Die Teilchen wechselwirken nur durch einen Stoß, zusätzliche Kräfte, wie Adhäsion aufgrund von Luftfeuchtigkeit oder Ölbeimengungen [19–21] können berücksichtigt werden.

Auch wenn man Granulate zuerst mit großen Partikelzahlen assoziiert, so ist auch die Untersuchung der Wechselwirkung einzelner Partikel miteinander oder mit dem Untergrund wichtig. Auch Systeme mit wenigen Teilchen zeigen interessante Phänomene. In Abbildung 1.1 unterteilt Jäger die Forschungsbereiche in eine mikroskopische, eine mesoskopische und eine makroskopische Ebene. Die Übergang von der mikroskopischen auf die mesoskopische Ebene ist Aufgabe vieler Untersuchungen. An erster Stelle steht hierbei ein detailliertes Verständnis der Reibung [23] und der Stoßvorgänge. Dabei wird das Gleiten oder Rollen eines Partikels auf einer Unterlage betrachtet. Dippel et al. [2, 3, 24–26] haben die Bewegung einer einzelnen Kugel auf einer rauhen, schiefen Ebene untersucht. Es wurde gezeigt, daß es dabei verschiedene Formen der Bewegung (Abbremsen bis

<sup>1</sup>Entscheidend ist dabei eigentlich das Verhältnis der Gravitationskraft  $F_G = mg$  und der zusätzlich wirkenden Kraft. In meinem Besitz befindliche Styroporkugeln mit ca. 4 mm Durchmesser zeigen ein deutlich von elektrischen Ladungen und der Luft dominiertes Verhalten.

zum Stillstand, konstante mittlere Geschwindigkeit oder beschleunigte Bewegung) gibt, die von der Reibung und der Energiedissipation beim Stoß abhängen. Untersuchungen zur sogenannten stick-slip Bewegung beim Ziehen eines Blocks über eine Oberfläche wurden von Elmer [27, 28] durchgeführt. Ein einzelnes Teilchen in einer Trommel zeigt ein Schwingungsverhalten, mit dem sich verschiedene Kraftgesetze untersuchen lassen. Dabei zeigt sich, daß zur Beschreibung dieses einfachen Falls das Coulombsche Reibungsgesetz nicht ausreichend ist [5, 29]. Scherer et al. haben die Bewegung von wenigen Kugeln in einer geschwenkten Schale untersucht [6–8, 29]. Hierbei zeigt sich, daß sich eine Umkehr der Bewegungsrichtung abhängig vom Füllgrad des Behälters ergibt. Man kann dabei den Übergang von der Bewegung einzelner Teilchen zur Bewegung beobachten.

H. Jäger bezeichnet die beiden Extremfälle eines schnellen Scherflusses und der kompakten ruhenden Teilchenanordnung als “alive state” und “dead state”. Ein System kann durch externe Kräfte angeregt werden. Dissipation und Reibung entziehen ihm Energie und bringen es zur Ruhe. Das Problem ist, daß der Übergang als eine Funktion der Dichte durch einen großen Bereich metastabiler Systeme erfolgt. Hier ist die Dichte der Partikel groß genug, um durch gegenseitige Blockade den Übergang zu verlangsamen. Als Folge kann das System sehr leicht “einfrieren”, also in einem metastabilen Bereich hängenbleiben. Das Verhalten des Systems wird irreversibel und hysteretisch, abhängig davon ab, wie stark das System “gekühlt” oder “geheizt” wird. Jäger bezeichnet diesen Zustand als glasartig [22].

Bei der technischen/physikalischen Bedeutung von Granulaten gibt es ein breites Spektrum an Möglichkeiten. Die Geowissenschaften beschäftigen sich schon seit Tausenden von Jahren mit Granulaten; der Ausspruch “...auf Sand gebaut” findet sich schon in der Bibel. Aussagen über die Tragfähigkeit des Untergrundes ermöglichen es erst, Hochhäuser und Brücken zu konstruieren, die auch standfest sind. Auch der Transport von Granulaten ist von großer technischer Bedeutung. Die Fördertechnik beschäftigt sich mit den dabei auftretenden Problemen. Die Physik begann sich erst deutlich später für diese Materialien zu interessieren, wie Arbeiten von Coulomb [30] über die Reibung, Faraday [31] über Sand auf einer schwingenden Platte und Reynolds [32] über die Dilatanz zeigen. Reynolds beispielsweise beschrieb die Dilatanz, also die Volumenzunahme eines Granulates bei Scherung. In den letzten Jahren hat das Interesse dann sehr stark zugenommen. Dies hängt sicher mit der zunehmenden Leistungsfähigkeit der Computer zusammen, aber auch mit der Vereinfachung der Bildanalyse, so daß granulare Systeme störungsfrei untersucht werden können.

Die Forschung an granularen Medien berührt auch andere Forschungsbereiche. Die Ähnlichkeit von fließendem Sand und dem Autoverkehr ist schon seit einigen Jahren Thema einer Konferenzreihe unter dem Namen “Traffic and granular Flow” [33, 34].

Bak, Tang, Wiesenfeld führten ein Modell [35, 36] eines Sandhaufens als ein Paradigma für die selbstorganisierte Kritikalität ein (siehe Abschnitt 1.2.2). Frette [37] untersuchte die Lawinenbildung für Reis in einer Hele-Shaw Zelle, also einem quasi zweidimensionalen System, und fand dieses Verhalten.

All dies zeigt, daß Granulate aufgrund ihrer Vielfalt ein anspruchsvolles, vielseitiges Forschungsgebiet sind. Da es unmöglich ist, die ganze Vielfalt der Systeme gleichzeitig bearbeiten zu wollen, muß die Art des Systems festgelegt werden, das beschrieben werden soll. In der vorliegende Arbeit werden Systeme mit den folgenden Eigenschaften behandelt:

1. zweidimensionale Systeme
2. keine langreichweitigen Wechselwirkungen.
3. Reibung, Kohäsion werden berücksichtigt.
4. Rauhe und glatte Partikel sollen unterschieden werden.

5. Die Partikel sind elastisch, nicht plastisch, und können nicht brechen.
6. beliebige Größenverteilungen

Aufgrund dieser Vorgaben können die vom Simulationsprogramm zu erfüllenden Forderungen angegeben werden:

1. Kräfte wirken nur bei Kontakten. Berühren sich 2 Teilchen nicht, ist die Kraft zwischen ihnen Null.
2. Die Partikel werden durch Polygone beschrieben. Eine große Zahl von Ecken nähert ein glattes Teilchen beliebig genau an.
3. Eine hinreichend große Zahl an Partikeln muß berechnet werden, um physikalisch relevante Systeme simulieren zu können.

Die zweite und dritte Forderung zusammen führen zu einem Problem mit der Rechenzeit. Typische Simulationen in dieser Arbeit berechnen zwischen 3000 und 10000 Teilchen bei etwa  $10^6$  bis  $10^7$  Zeitschritten. Dabei kann man sich die speziellen geometrische Eigenschaften (siehe Abschnitt 2.4) der Partikel zunutze machen, um schnelle, effiziente Algorithmen zu entwickeln und zur Anwendung zu bringen.

## 1.2 Granulare Aufschüttungen

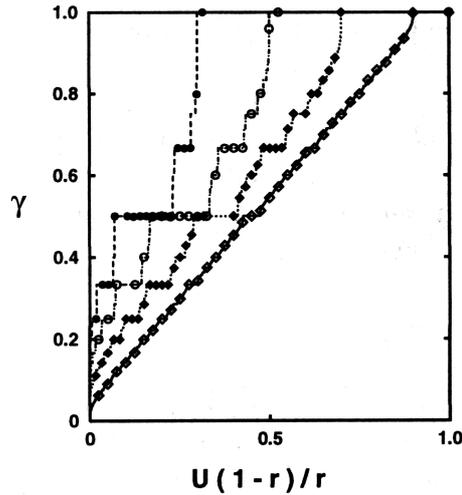
### 1.2.1 Böschungswinkel

Schüttet man ein gewöhnliches, trockenes Granulat auf eine Ebene, so entsteht ein Haufen. Diese Tatsache ist als solche eigentlich schon bemerkenswert. Eine Flüssigkeit würde zu einem dünnen Film zerfließen, begrenzt nur durch seine Oberflächenspannung. Ein Festkörper würde seine Form bis auf elastische und plastische Verformungen beibehalten.

Die Oberfläche eines Sandhaufens scheint auf Skalen, die größer als der typische Teilchendurchmesser sind, eine glatte Kurve zu sein. Der Winkel, den die Oberfläche eines Sandhaufens mit dem Untergrund einschließt, heißt "Böschungswinkel" oder "angle of repose". Dieser Winkel ist eine typische Größe des verwendeten Materials. Am einfachsten ist dann die Form eines Sandhaufens zu bestimmen, der auf einer Platte mit einer bestimmten Form aufgeschüttet wird. Dabei entsteht dann der Körper, der unter der Voraussetzung, daß die Oberfläche den Böschungswinkel hat, maximales Volumen hat. Auf einer runden Scheibe ist dies beispielsweise ein Kegel und auf einer quadratischen Grundfläche eine vierseitige Pyramide.

Nach Herrmann [38] gibt es noch keine Theorie, mit der man den Böschungswinkel für eine bestimmte Granulatsorte aus dessen mikroskopischen Eigenschaften wie Größe, Form, Oberflächenrauigkeit, Restitutionskoeffizient und Elastizitätsmodul berechnen kann. In derselben Veröffentlichung wird ein Modell vorgestellt, welches den Böschungswinkel für ein vereinfachtes System bestimmen kann. Es wird angenommen, daß die Oberfläche des Sandhaufens aus einzelnen Stufen besteht. Springt nun ein Teilchen mit der Energie  $e_i$  eine Stufe hinunter, hat es anschließend die Energie

$$e'_i = (e_i + \Delta h)r \quad , \quad (1.2)$$



**Abbildung 1.2:** Numerisch bestimmte Werte des Böschungswinkels  $\gamma$  in Abhängigkeit der Funktion  $U \frac{1-r}{r}$  für  $r = 0.3$  ( $\bullet$ ),  $0.5$  ( $\circ$ ),  $0.7$  ( $\blacklozenge$ ) und  $0.9$  ( $\diamond$ ). Dabei ist  $r$  der Restitutionskoeffizienten und  $U$  die Höhe der Stufen (aus [38]).

wobei  $r$  der Restitutionskoeffizient ist und  $\Delta h$  der Energiegewinn aufgrund der Höhe der Stufe. In einem realen Sandhaufen kann ein herabrollendes Korn in einem lokalen Energieminimum liegenbleiben. Dies wird durch eine Energiebarriere  $U$  beschrieben. Wenn also  $e'_i < U$  ist, bleibt das Teilchen liegen und erhöht stattdessen die Stufe. Die beiden Parameter  $r$  und  $U$  fassen dann die mikroskopischen Parameter zusammen, wobei  $r$  den Stoß zweier Teilchen charakterisiert und  $U$  das Anhaften an der Oberfläche des Haufens.

Numerische Untersuchungen [38,39] ergeben das in Abbildung 1.2 dargestellte Verhalten. Die Stufen sind Artefakte des Modells, da auf einem quadratischen Gitter simuliert wurde. Der allgemeine Trend ist aber, wenn entweder bei jedem Sprung viel Energie dissipiert wird (kleines  $r$ ) oder die lokalen Minima sehr groß sind (großes  $U$ ), dann steigt der Böschungswinkel. Das genannte Modell läßt sich auch analytisch berechnen, wobei man annimmt, daß sich der Haufen in einem stationären Zustand befindet. Dies stellt gewisse Bedingungen an die Energien der einzelnen Stufen, die dann nicht mehr wachsen oder schrumpfen dürfen. Eine genaue Ableitung findet sich in [38,40]. Interessant ist, daß sich dieses Modell auf ein Ising-Spinalgas mit langreichweitiger Wechselwirkung abbilden läßt. Die Hamiltonfunktion ist dann:

$$\mathcal{H} = \sum_i U s_i = \frac{1}{2} \sum_{i,j} J(|i-j|) (s_i + 1)(s_j + 1), \quad (1.3)$$

wobei  $s_i = \pm 1$  und die Wechselwirkung

$$J(x) = (1 - r^L) r^x \quad (1.4)$$

ist.

Einer Stufe entspricht in diesem Modell ein Spin mit  $s_i = +1$ .  $e_i$  ist die Wechselwirkungsenergie des Spins. Der Böschungswinkel  $\theta$  entspricht der Magnetisierung der Spinkette und  $U$  dem externen Magnetfeld  $H$ . Dieses Modell wurde von Bak und Bruinsma gelöst und bestätigt die Ergebnisse der Numerik.

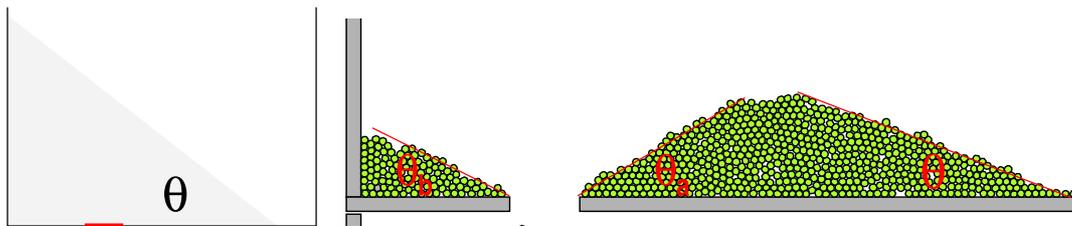
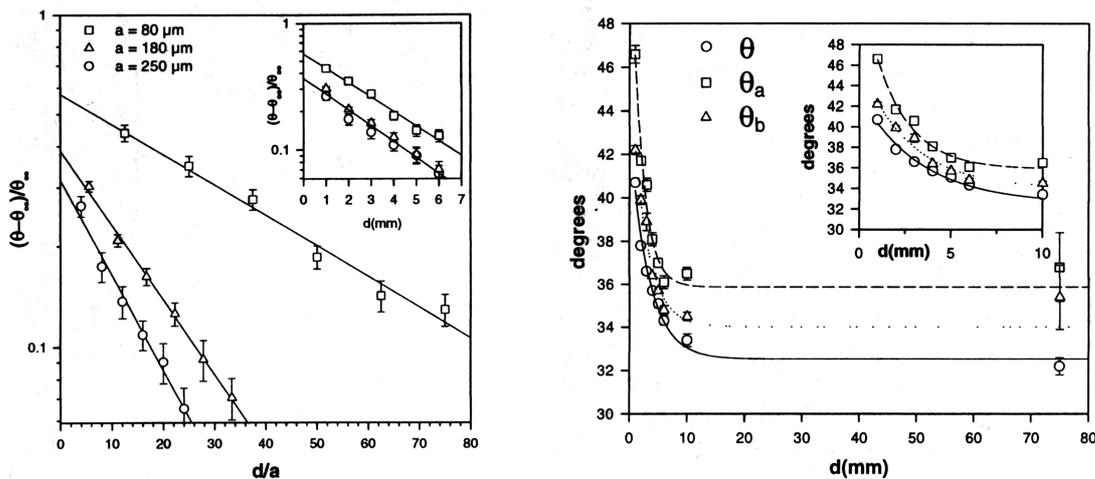


Abbildung 1.3: Geometrie zur Bestimmung der Böschungswinkel  $\theta$ ,  $\theta_a$  und  $\theta_b$



(a) Abhängigkeit des Böschungswinkels  $\theta_a$  von der Dicke der Hele-Shaw Zelle  $d$  und dem Partikeldurchmesser  $a$

(b) Abhängigkeit der Böschungswinkel  $\theta$ ,  $\theta_a$  und  $\theta_b$  von der Zellenbreite  $d$

Abbildung 1.4: Experiment von Grasselli in einer Hele-Shaw Zelle (aus [38])

Die Angabe eines Böschungswinkels ist aber nicht ausreichend, um echte Sandaufschüttungen zu beschreiben. Dazu muß man das Verhalten eines echten Sandhaufens ansehen. Man schüttet Material auf und die Neigung der Oberfläche nimmt zu. Irgendwann ist der Winkel  $\theta_s$  erreicht und die Oberfläche setzt sich in Bewegung und kommt erst bei einem zweiten Winkel  $\theta_d$  zum Stehen. Schon Bagnold zeigte dies [17] und bestimmte  $\Delta\theta = \theta_s - \theta_d \approx 1^\circ - 2^\circ$ ;  $\theta_s$  heißt der statische (static angle of repose) und  $\theta_d$  der dynamische Böschungswinkel (dynamic angle of repose).

Man muß berücksichtigen, daß der Haufen eine ausreichende Größe haben muß. Sonst ist der durch die Winkel  $\theta_s$  und  $\theta_d$  eingeschlossene Bereich so eng, daß dort keine Teilchen Platz finden. Dann kann sich keine große Lawine ausbilden und die Art der Lawinenbildung ändert sich [41]. Eine Basis von ungefähr 30 Partikeln kann als typische Mindestgröße angenommen werden.

Die Experimente von Grasselli [38, 42] sind aus zwei Gründen interessant. Erstens wurden die Versuche in einer vertikalen Hele-Shaw Zelle durchgeführt, deren Dicke variiert wurde. Damit kann der Einfluß der Wände beim Übergang von einem quasi<sup>2</sup>-zweidimensionalen zu einem dreidimensionalen Experiment untersucht werden.

In Abbildung 1.4(a) sieht man, daß der Böschungswinkel mit zunehmender Dicke der Zelle abnimmt.

<sup>2</sup>Die Reibung mit den Wänden hat Einfluß auf das Verhalten des Systems.

Der Einfluß der Wände ist bei Systemen, deren Dicke ungefähr dem Teilchendurchmesser entspricht, sehr groß. Deswegen ist ein Vergleich dieser Systeme mit echt zweidimensionalen Simulationen schwierig. Die Reibung zwischen den Partikeln und der Glasplatte wird bei den Simulationen umgangen. Die Experimente zeigen außerdem einen Zusammenhang zwischen den kritischen Winkeln und der inneren Struktur beziehungsweise der Entstehungsgeschichte des Haufens. Das Granulat wurde in die Hele-Shaw Zelle am Rand eingefüllt, so daß nur ein "halber" Sandhaufen entstand. Dann wurde ein kleines Loch an der Unterseite der Zelle geöffnet, so daß das Material langsam, ohne Lawinen, ausfließen konnte. Wie in Abbildung 1.3 dargestellt, entstanden damit zwei weitere Böschungen mit den Winkeln  $\theta_a$  und  $\theta_b$ . Dabei ist aus Abbildung 1.4(b) ersichtlich, daß durchgängig  $\theta < \theta_b < \theta_a$  gilt. Ein Erklärungsansatz könnte in der Strukturierung des Materials durch den vorhergehenden Aufschüttungsprozeß gesucht werden.

Ein weiterer kritischer Winkel  $\theta_D$  tritt auf, wenn man das Granulat am Fließen hält. Dies wird bewerkstelligt, indem es in einer horizontalen Trommel langsam um die Längsachse gedreht wird (z.B. [43–48]).

Im allgemeinen gibt es also mehrere kritische Winkel, die zusätzlich noch durch Kohäsion [49], Dichte des Systems [50] und andere Parameter beeinflusst werden können.

### 1.2.2 Lawinenbildung

Eng verbunden mit dem Phänomen des Böschungswinkels ist die Bewegung von Partikeln auf der Oberfläche eines Granulates. Dabei treten typischerweise zwei Bewegungsmuster auf: entweder ein einzelnes Teilchen bewegt sich über die Oberfläche des Sandhaufens oder aber das bewegte Teilchen löst eine Lawine aus.

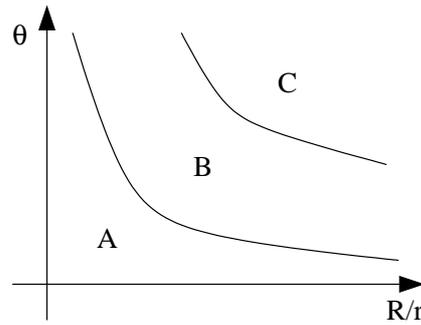
Bei der Bewegung eines einzelnen Partikels über eine raue Fläche gibt es nach Experimenten [51, 52] und theoretischen Arbeiten [2, 3, 24–26] drei Bereiche:

1. Das Teilchen verliert bei jeder Kollision Energie und bleibt irgendwann liegen.
2. Das Teilchen erreicht einen stationären Zustand: im Mittel verliert es durch die Kollisionen genau soviel Energie, wie es durch seine Abwärtsbewegung gewinnt.
3. Die Zeit zwischen den Stößen wird immer länger, die Sprünge des Teilchens werden immer größer. Dabei nimmt die Geschwindigkeit linear zu.

Ein schematisches Phasendiagramm ist in Abbildung 1.5 dargestellt. Ein erweitertes System wurde von Quartier et al. [53] untersucht. Dabei wird der Untergrund durch Kugellager ersetzt, so daß die "Beweglichkeit" einer Sandoberfläche, das Abrollen der Körner, genauer repräsentiert wird. Diese Systeme als Modell für die Oberfläche eines Sandhaufens haben dort ihre Grenzen, wo das sich bewegende Teilchen genug Energie hat, um weitere Teilchen aus der Oberfläche herauszulösen.

Wenn es eine kollektive Bewegung von vielen Sandteilchen gibt, spricht man von einer Lawine. Da diese in einem relativ großen Bereich das Material umlagern, können sie die innere Struktur eines Sandhaufens verändern. Interessant ist dabei, wie tief eine Lawine wirkt und ob lawinenartige Umordnungsvorgänge auch im Inneren eines Sandhaufens stattfinden können [54].

Bei den Experimenten mit Lawinen in dünnen Sandschichten konnte Aguirre [55, 56] zeigen, daß die Größe des Böschungswinkels für Systeme, die tiefer als etwa 10 Lagen sind, nicht mehr beeinflusst wird. Auch andere Daten zeigen, daß dies eine typische Größe der Lawinen in dem untersuchten System ist.



**Abbildung 1.5:** Phasendiagramm der Bewegung eines Teilchens auf der schiefen Ebene. Auf der Abszisse wird das Verhältnis der Kugelgröße  $R$  zur Oberflächenrauigkeit  $r$  der Ebene aufgetragen, auf der Ordinate die Neigung  $\theta$  der Ebene. Bereich A: Das Teilchen kommt zur Ruhe. Bereich B: Das Teilchen bewegt sich mit einer konstanten mittleren Geschwindigkeit. Bereich C: Das Teilchen wird beschleunigt. (nach [3])

Daerr [57–59] untersuchte die Ausbreitung einer Lawine in einer dünnen Granulatschicht auf einer rauen Unterlage. Dabei zeigt sich, daß die Lawine ab einem bestimmten Neigungswinkel zusätzlich auch bergaufwärts laufen kann. Außerdem wurde in dieser Arbeit die Form der Front untersucht.

Allgemeinere theoretische Modelle versuchen die zeitliche Entwicklung der Oberfläche eines Sandhaufens  $h(x, t)$  unter verschiedenen Annahmen [60, 61] zu berechnen. Das wohl bekannteste Modell von Bak, Tang und Wiesenfeld [35, 36] diskretisiert zusätzlich die Ortsvariablen. Man hat einen zellulären Automaten mit den Regeln:

$$z(x, y) \rightarrow z(x, y) - 4, \quad (1.5)$$

$$z(x \pm 1, y) \rightarrow z(x \pm 1, y) + 1, \quad (1.6)$$

$$z(x, y \pm 1) \rightarrow z(x, y \pm 1) + 1, \quad (1.7)$$

falls  $z(x, y)$  größer als ein kritischer Wert  $K$  ist. Der Wert  $z$  entspricht dabei der lokalen Steigung eines Sandhaufens. Für die Ränder des Systems gilt  $z = 0$ , was bedeutet, daß die “Teilchen” dort “herabfallen”. Das System wird mit einer zufälligen Anfangsbedingung  $z \gg K$  initialisiert und entwickelt sich dann in einen stationären Zustand; für alle  $z(x, y)$  gilt  $z < K$ . Um die Dynamik zu untersuchen, stört man das System an zufälligen Positionen  $(x, y)$ . Die dabei entstehenden Lawinen können auf allen Größenskalen auftreten ( $\frac{1}{f}$ -Rauschen). Dieses Modell ist das Paradigma der “Self-Organized Criticality”. Ein ähnlicher zellulärer Automat von Nishimori [62, 63] beschreibt sehr anschaulich die Bildung der verschiedenartigen Dünen und ihre Wanderung in einer Wüste. Diese Modelle sind für die Ziele dieser Arbeit nur sehr schwer anwendbar, da sie keine Aussage über die innere Struktur eines Sandhaufens treffen. Da keine Kräfte im Inneren des Haufens beschrieben werden, ist es nicht sinnvoll, Aussagen über Spannungstensoren oder Druckverteilungen zu treffen, ohne vorher weitere Annahmen zu machen.

Interessant ist in diesem Zusammenhang noch eine Arbeit von Alonso und Herrmann [64], in der sie die Form des Fußes eines Sandhaufens beschreiben. Lawinen, die auf einem Sandhaufen abgehen, werden ja am Ende des Abhangs vom Untergrund aufgehalten. Die Oberfläche trifft nicht exakt unter dem Winkel  $\theta$  den Untergrund, sondern hat einen Ausläufer. Aus Überlegungen zum Fluß der Teilchen  $\Phi(h)$  und der Forderung, daß der Haufen während des Wachstums immer stationär sein soll, folgt als Profil  $x(h)$

$$x(h) = \frac{h_m - h}{\gamma} + l_e \ln \frac{h_m}{h} . \quad (1.8)$$

Dabei ist  $h_m$  die Höhe des Haufens an der Spitze,  $\gamma$  der Reibungskoeffizient und  $l_e$  die typische horizontale Entfernung, bevor ein Teilchen auf der Oberfläche liegenbleibt. Zur Steigung aufgrund des Böschungswinkels kommt also noch ein logarithmischer Korrekturterm hinzu.

### 1.2.3 Größen–Segregation

Bei der Handhabung einer Mischung von unterschiedlichen granularen Materialien (Größe, Reibung, Rauigkeit) gibt es einen Effekt, der die Ergebnisse stark beeinflussen kann. Die verschiedenen Partikel des Granulates entmischen sich.

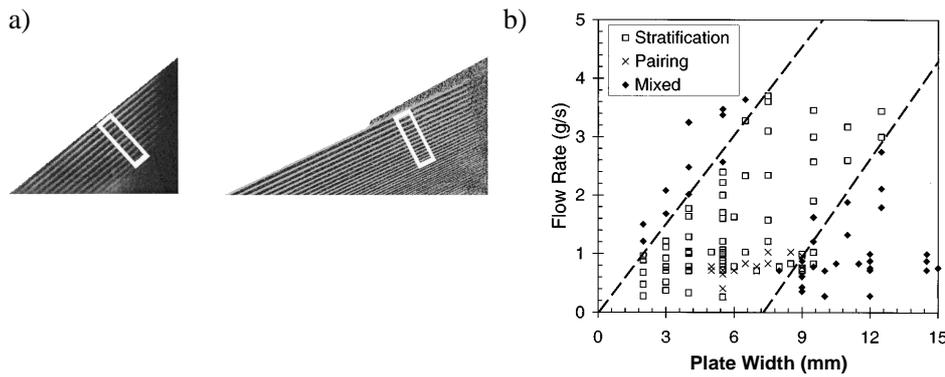
Das klassische Beispiel für die Entmischung ist die Frage “Warum liegen im Müsli die Nüsse immer oben?” und führt zum bekanntesten Problem, dem Brazil-Nut-Effekt. Wenn man ein Granulat, das aus vielen kleinen Partikel und einigen größeren besteht, in einem Behälter vibriert, so ist es möglich, daß die großen Partikel nach oben wandern, selbst wenn sie eine höhere Dichte als die kleinen Partikel haben. Beispielsweise untersuchten Jäger et al. [22, 65] mit NMR-Mesungen den Aufstieg von Kaffeebohnen in Mohnsamen und konnten Geschwindigkeitsprofile im Inneren des Systems bestimmen. K. Aoki [66] zeigte sehr anschaulich, daß in diesen Systemen Konvektionsrollen auftreten. Diese sollen in bestimmten Systemen für den Transport von Teilchen an die Oberfläche verantwortlich sein. Duran [67] untersuchte die Wechselwirkung zweier Partikel in einer derartigen Versuchsanordnung. Wenn im Abschnitt 4 ein Sandhaufen durch Vibration verändert wird, so muß man die Amplitude der Vibration so klein wählen, daß dieser Effekt nicht auftritt.

Als weiteren Effekt gibt es die Entmischung, wenn man einen mit einem Granulat gefüllten Zylinder um seine Längsachse dreht. Dabei sind zwei Formen zu unterscheiden, die axiale Segregation und die radiale. Bei der radialen Segregation entsteht ein Kern, in dem sich dann eine Sorte des Materials ansammelt. Bei der axialen Segregation entstehen Streifen, in denen eine Sorte des Materials gehäuft auftritt. Ristow untersuchte mit Simulationen das Geschwindigkeitsprofil [43, 68] in einer derartigen Trommel, bestimmte mit Hilfe von NMR die Geschwindigkeit, mit der sich der Kern ausbreitet [69] und zeigte den Einfluß der Röhrendenden auf das System [43]. Ein ähnlicher Effekt kann auch auftreten, wenn man ein Material in einen Behälter füllt. Károlyi et al. zeigten mit der Simulation eines zellulären Automaten, daß sich eine bidisperse Mischung von Teilchen beim Einfüllen in ein Silo entmischen kann, wenn man die Teilchen nicht mittig, sondern seitlich zuführt [70].

Der wohl optisch reizvollste Effekt der Entmischung ist die Stratifikation [44, 72–76]. Ein großflächiges Streifenmuster entsteht, wenn sich eine Mischung zweier unterschiedlicher Partikel lawinenartig hangabwärts bewegt und dabei von einer sich hangaufwärts bewegenden Schockfront abgebremst wird. Da die im Abschnitt 4 simulierten Haufen mit der gleichen Methode aufgeschüttet werden wie in den Experimenten, mußte beachtet werden, ob es zur Streifenbildung kam. Köppe zeigt [71] ein Phasendiagramm, das die Bereiche beschreibt, in denen Streifenbildung beobachtet werden kann.

## 1.3 Druckverteilung unter granularen Aufschüttungen

Die Druckverteilung auf dem Untergrund eines Sandhaufens ist nicht eindeutig bestimmt. Typischerweise zeigen kegelförmige Sandhaufen ein deutliches lokales Minimum im Druck direkt unter der Spitze (siehe 1.7). Der Druckabfall kann bis zu 50% des Maximaldrucks betragen [77], hängt aber sehr stark von der Art des verwendeten Materials [78] und dessen Größenverteilung [79] ab. Es



**Abbildung 1.6:** a) Bilder der Stratifikation bei Sandhaufen. b) Phasendiagramm für das Auftreten der Stratifikation in Abhängigkeit vom Plattenabstand und der Flussrate. (aus [71])

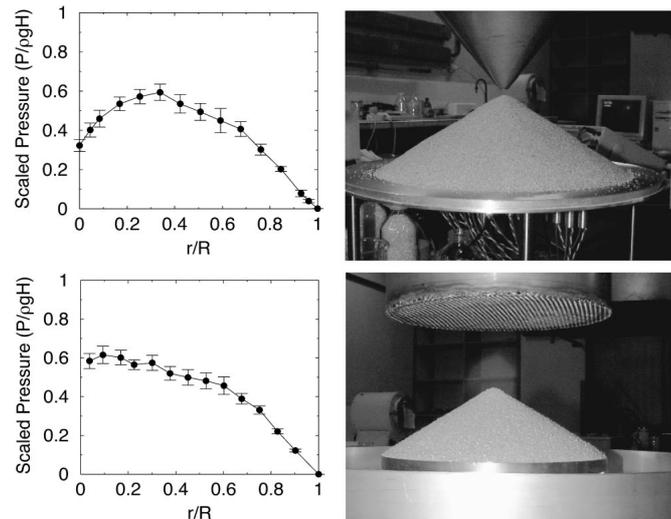
gibt aber auch Materialien, deren Druckminima sehr ähnlich sind, zum Beispiel Seesand und eine bestimmte Sorte Dünger [80, 81]. Die Form des Sandhaufens spielt eine große Rolle, keilförmige Sandhaufen (Dünen) haben bisher in den Experimenten keinen [82, 83] oder einen sehr kleinen [77] sog. *Dip* gezeigt. Dieser Unterschied kann durch eine Mittelung entlang der Düne erklärt werden [84]. Die Effekte, die aufgrund der Eigenschaften der einzelnen Körner wie Form, Reibung, Oberflächenrauigkeit und anderen entsteht, sind dagegen sehr viel komplexer.

Die Schwierigkeiten bei der Beschreibung von Granulaten resultieren aber nicht nur aus den Materialeigenschaften der einzelnen Partikel, sondern auch aus den entstehenden Strukturen. Der Magdeburger Dom ist aus Sandsteinblöcken errichtet. In einem Haufen Sandsteinblöcke haben zwar die Partikel die gleichen Eigenschaften wie die erste gotische Kathedrale nördlich der Alpen, aber die Eigenschaften als Ganzes sind völlig unterschiedlich (natürlich ganz abgesehen von den ästhetischen Aspekten dieses Vergleiches). Im folgenden sollen einige einfache Modellvorstellungen für den inneren Aufbau eines Sandhaufens und die daraus resultierenden Druckverteilungen gezeigt werden. Entsprechende Simulationen zu einigen dieser Beispiele finden sich im Abschnitt 4.1.

Abbildung 1.8(a) zeigt das einfachste Modell, dessen Druckverteilung auch dem entspricht, was man intuitiv erwartet. Einzelne Steine werden übereinandergeschichtet, der Druck wird senkrecht nach unten abgeleitet. Solche Druckverteilungen erhält man bei Sandhaufen, die schichtweise aufgeschüttet wurden (siehe Abschnitt 4.2.1) und als Modelle für Dämme dienen [83].

Das zweite (Abbildung 1.8b) Modell ordnet die Partikel auf einem Gitter so an, daß die Partikel ihr Eigengewicht auf die beiden unteren Nachbarn gleichmäßig verteilen. Die Partikel am Boden tragen dann alle jeweils gleichviel Last, der Druck am Boden ist konstant. Wichtig ist, daß der Untergrund dann auch die Kräfte in Tangentialrichtung aufnehmen muß. Wenn man den Winkel des Haufens verkleinert, indem man einzelne Teilchen an der Oberfläche entfernt (siehe 1.8c), umfaßt der konstante Bereich zwar nicht mehr den ganzen Haufen, bleibt aber vorhanden. Eine genauere mathematische Behandlung dieses Modells findet sich bei Hong [86]. Molekulardynamische Simulationen [87, 88] zeigten, daß diese analytischen Resultate mit der Simulation, abgesehen von einer Abrundung des Druckverlaufes durch die endliche Elastizität, sehr gut übereinstimmen.

Das dritte Modell (Abbildung 1.8d) setzt den Sandhaufen aus Stäben zusammen, um einen sogenannten "falschen Bogen" zu bilden. Die Kräfte werden dann nur entlang der Stäbe weitergeleitet und in der Mitte ist der Druck gleich Null. Die Vorstellung, daß Granulate einzelne Stäbe bilden, erscheint auf den ersten Blick etwas eigentümlich. Die verschiedenen Bilder der Kraftnetzwerke in dieser Arbeit zeigen aber, daß die Kräfte sich tatsächlich auf Pfaden fortsetzen, die den Stäben ähnlich sehen. Dreht man die Stäbe so, daß sie einen anderen Winkel haben als der Böschungswinkel, so entsteht

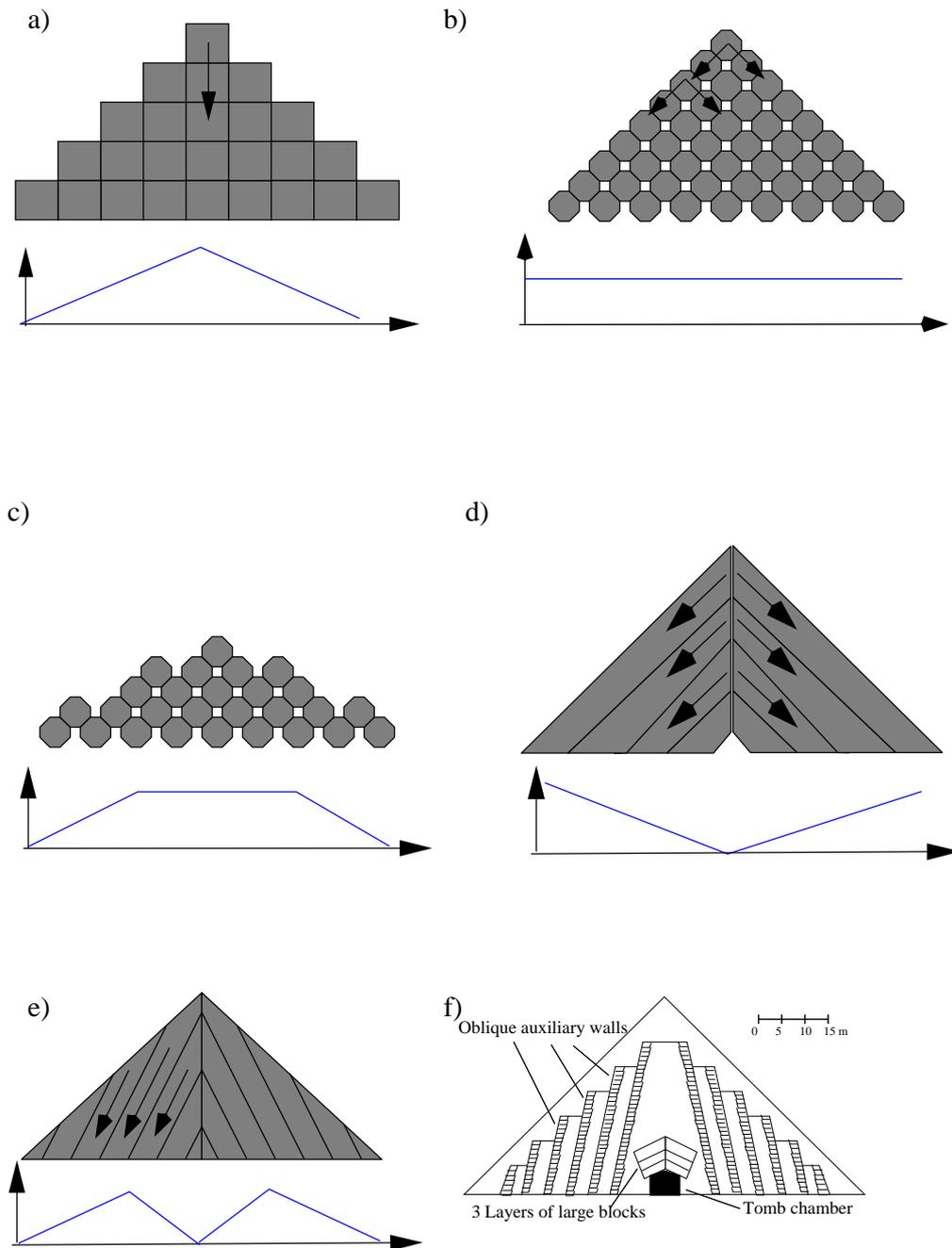


**Abbildung 1.7:** Experimentelle Bestimmung der Druckverteilung eines (oben) von einer punktförmigen Quelle aus aufgeschütteten Sandhaufens (unten) unter einem geschichteten Sandhaufen. (aus [85])

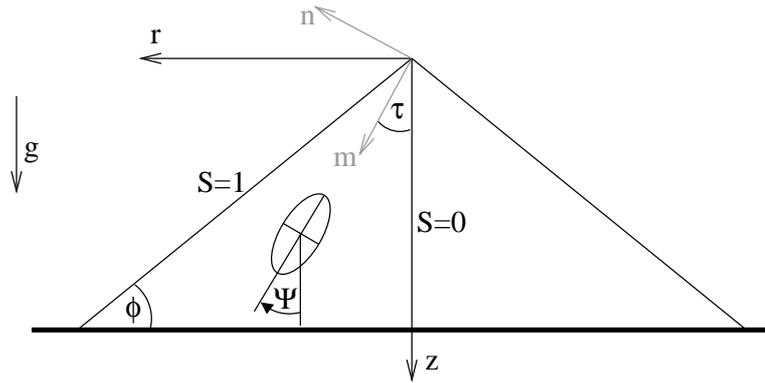
eine Druckverteilung mit einem Dip (Abbildung 1.8e). Ein ähnliches Bild zitieren auch Wittmer et al [89], die auf Edwards und Oakeshott [90] verweisen. Eigentlich ist die Art und Weise, mit dieser Anordnung Druck zur Seite abzuleiten, schon deutlich älter. Die alten Ägypter haben mit dieser Methode bereits ihre Pyramiden (Abbildung 1.8f) gebaut, wobei sie im Inneren zur Verstärkung versteckte Wände einsetzen, die die Last von der in der Mitte befindlichen Grabkammer weggleiten; die falschen Bögen wurden auch über der Grabkammer selbst eingesetzt. Pyramiden hatten eine ähnliche Form wie Sandhaufen, auch neigen manche Pyramiden, wie beispielsweise diejenige in Medun, gelegentlich zur Lawinenbildung. Trotzdem sind diese Bauwerke mit einem Böschungswinkel von  $56^\circ$  und ihrer regelmäßigen inneren Struktur (vergl. Abschnitt 4.1) keine geeigneten Modelle für Sandhaufen.

Durch eine Linearkombination dieser Modelle lässt sich eine fast beliebige Druckverteilung realisieren. Kommt dann noch die zufällige Positionierung der Teilchen und der Kräfte hinzu, so existiert ein riesiger Parameterraum, der die a-priori Auswahl eines Modells sehr schwer macht. Erst das Wissen über die innere Struktur von Sandhaufen mit einem Dip in der Druckverteilung erleichtert dabei die Auswahl.

Auf der Tagung “Physics of Dry Granular Media” 1997 in Cargèse gab es eine intensive Diskussion über die Modellierung von granularen Medien und den zugehörigen Randwertproblemen. Im Zentrum stand dabei ein Ansatz von Wittmer, Cates und Claudin [89, 91–94], der das lokale Druckminimum zu erklären versucht. Dabei werden Annahmen über die Spannungen im Inneren eines Sandhaufens gemacht, die anschaulich motiviert werden können. Diese Annahmen sind aber experimentell nur sehr schwer [95] zu verifizieren. Bisher gibt es nur wenige experimentelle Methoden, um Kraftnetzwerke zu visualisieren. Man kann die Spannungen durch den photoelastischen Effekt [96, 97] sichtbar machen oder mit der Methode von Tsoungui [98] arbeiten, bei der die Kontaktfläche gemessen wird. Diese Methoden bieten aber noch nicht die Möglichkeit, sehr große Systeme mit unterschiedlichen Teilchen quantitativ zu untersuchen. Dies motivierte zu den Untersuchungen an Sandhaufen durch Simulationen.



**Abbildung 1.8:** "Einfache" Modelle, um das Innere eines Sandhaufens zu beschreiben. f) zeigt schematisch das Innere einer ägyptischen Pyramide.



**Abbildung 1.9:** Koordinatensystem nach Wittmer [89]

Um die Simulationsergebnisse im Abschnitt 4 werten zu können, sollen hier kurz die Theorien von Wittmer et al. vorgestellt werden. In ihren Veröffentlichungen [89] werden vier verschiedene Lösungsansätze für ein zweidimensionales System untersucht.

Da das Modell in den Originalarbeiten auf 3 Dimensionen erweitert wird, werden Zylinderkoordinaten verwendet (siehe Abbildung 1.9). Der Ursprung liegt in der Spitze des Haufens,  $z$  zeigt senkrecht nach unten. Die Dichte  $\rho$  wird als konstant angenommen und gleich 1 gesetzt. Im mechanischen Gleichgewicht gilt dann

$$\partial_r \sigma_{rr} + \partial_z \sigma_{rz} = 0 \quad (1.9)$$

$$\partial_r \sigma_{rz} + \partial_z \sigma_{zz} = g. \quad (1.10)$$

Man hat also zwei Gleichungen für drei Unbekannte. In einem elastischen Festkörper hat man als dritte, konstitutive Gleichung das Hookesche Gesetz, welches Spannung und Verformung zueinander in Beziehung setzt. Bei einem Granulat scheint eine sinnvolle Definition einer Verformung aber nicht immer möglich. Vielmehr wird versucht, das System aufgrund der Reibung an den Kontakten zu verstehen. Die Reibung gibt aber nur einen Maximalwert für die Scherung an, die ein einzelner Kontakt tragen kann.

Wittmer et al. nehmen nun an, daß auf einer Längenskala größer als die Einzelteilchen ein Zusammenhang zwischen der *mittleren* Reibung und den Normalkräften besteht. Dadurch rechtfertigt sich die Annahme, daß die fehlende konstitutive Gleichung einen Zusammenhang zwischen den einzelnen Komponenten des Spannungstensors darstellt. Folgende Anforderungen werden dabei gestellt:

1. Die Gleichung ist *lokal*.
2. Die lokale Anordnung und die Kontakte der Partikel soll wiedergespiegelt werden.
3. Die konstitutiven Gleichung sollen die Entstehungsgeschichte des Materials berücksichtigen.
4. Ab dem Zeitpunkt, an dem ein Bereich überdeckt ist, ist die Entstehungsgeschichte zu einem Abschluß gekommen und die konstitutiven Gleichung können sich nicht mehr ändern. Wittmer et al. sprechen hier von einem *perfekten Gedächtnis*.

Zu bemerken ist hier, daß Wittmer den Begriff der konstitutiven Gleichung etwas anders benutzt, als normalerweise üblich. Die vorgeschlagenen konstitutiven Gleichungen sind nicht nur bloss von den Materialeigenschaften abhängig, sondern gelten nur für das Ergebnis eines bestimmten Prozesses.

Für die weitere Rechnung wird noch eine SkalenvARIABLE  $S = \frac{r}{c}z = r \tan \varphi z$  benötigt, sie ist gleich 0 im Zentrum des Haufens und 1 an der Oberfläche; dabei ist  $\varphi$  der Böschungswinkel des Haufens.

Aus der Annahme, daß an der Oberfläche gerade das Mohrsche Stabilitätskriterium gilt<sup>3</sup> (IFSRandbedingung), kann man dann folgern, daß die Komponenten des Spannungstensors an der Oberfläche gleich Null sind:

$$\sigma_{rr}(S = 1) = \sigma_{zz}(S = 1) = \sigma_{rz}(S = 1) = 0 \quad (1.11)$$

Die Stabilitätsbedingung legt gleichzeitig auch noch das Verhältnis der Komponenten des Spannungstensors in der Nähe der Oberfläche fest

$$\lim_{S \rightarrow 1} \frac{\sigma_{rz}(S)}{\sigma_{rr}(S)} = \tan \varphi \quad (1.12)$$

$$\lim_{S \rightarrow 1} \frac{\sigma_{rr}(S)}{\sigma_{zz}(S)} = \frac{1}{1 + 2 \tan^2 \varphi} = \eta_0 \quad (1.13)$$

und ergibt dann an bzw. nahe der Oberfläche als Winkel der Hauptachsen des Spannungsellipsoides  $\Psi = (\frac{\pi}{2} - \varphi)/2$ .

Im nächsten Schritt wird eine Skalenanalyse durchgeführt. Dabei nimmt man an, daß es keine ausgezeichnete Längenskala im System gibt; die Größenordnung der Teilchen soll klein genug sein, um nicht von Bedeutung zu sein. Für Gleichung 1.10 wird eine Lösung der Form

$$\sigma_{ij} = g z s_{ij}(S) \quad (1.14)$$

gesucht. Diese RSF-Skalierung (Radial Stress Field) zusammen mit der Forderung nach der Lokalität ergibt als allgemeinsten Ansatz

$$\frac{\sigma_{rr}}{\sigma_{rz}} = C\left(\frac{\sigma_{rz}}{\sigma_{zz}}, S\right). \quad (1.15)$$

Die Annahme des perfekten Gedächtnisses des Sandhaufens führt dazu, daß die Funktion  $C$  nicht von  $S$  abhängen darf, also

$$\frac{\sigma_{rr}}{\sigma_{rz}} = C\left(\frac{\sigma_{rz}}{\sigma_{zz}}\right) = C(U) \text{ mit } U(S) = \frac{\sigma_{rz}}{\sigma_{zz}} \quad (1.16)$$

Für ein derartiges  $C(U)$  gilt, ist der Winkel  $\Psi$  der Hauptachse des Spannungstensors gegen die Senkrechte der Oberfläche immer gleich

$$\Psi(1) = \psi \equiv (\pi - 2\varphi)/4 \quad . \quad (1.17)$$

Nun zu den einzelnen Modellen. Das erste Modell IFE (*Incipient Failure Everywhere*) nimmt an, daß die Bedingung, daß das Material an der Stabilitätsgrenze ist, nicht nur an der Oberfläche gilt, sondern im ganzen Haufen erfüllt ist. Es gilt dann

$$C(U) = \frac{1}{\cos \varphi} \left( (\sin^2 \varphi + 1) \pm 2 \sin \varphi \sqrt{1 - (U \cot \varphi)^2} \right) \quad (1.18)$$

Dieses Modell zeigt für  $\sigma_{zz}$  kein lokales Minimum.

Das BCC-Modell (*Bouchaud-Cates-Claudin*) nimmt an, daß die Diagonalelemente des Spannungstensors zueinander proportional sind.

<sup>3</sup>Dies bedeutet, daß die Oberfläche gerade nicht am Abrutschen ist.

$$C(U) = \eta \quad (1.19)$$

Aus der IFS-Annahme folgt außerdem  $\eta = \eta_0$ .  $\sigma_{zz}$  hat hier im Inneren ein ebenes Plateau, aber keinen Dip.

Das FPA-Modell (*Fixed Principal Axis*) postuliert, daß die Orientierung  $\Psi$  der Hauptachsen im Inneren des Sandhaufens immer gleich ist. Aus der IFS-Annahme folgt dann  $\Psi = \psi = 1 - 2U \tan \varphi$  und damit

$$C(U) = 1 - 2U \tan \varphi \quad (1.20)$$

Aus dem Böschungswinkel als einzigem Materialparameter ergibt sich dann ein Druckverlauf, der einen deutlichen Dip zeigt.

Die FPA-Annahme ist ein Spezialfall des OSL-Modells (*Oriented Stress Linearity*), das wiederum eine Verallgemeinerung des BCC-Modells darstellt. Anstatt im Koordinatensystem  $(r, z)$  eine Proportionalität der Diagonalelemente zu fordern, wird nur angenommen, daß dies in einem um den Winkel  $\tau$  gegen die Vertikale gedrehten Koordinatensystem  $(m, n)$  gilt.

$$\sigma_{nn} = K \sigma_{mm} \quad (1.21)$$

Damit folgt dann

$$C(U) = \eta + \mu U \quad (1.22)$$

wobei dann

$$\eta = \frac{K - \tan^2 \tau}{1 - K \tan^2 \tau} \quad \text{und} \quad \mu = \frac{2(K + 1) \tan^2 \tau}{1 - K \tan^2 \tau} \quad (1.23)$$

gilt. Der Spezialfall  $\eta = \eta_0$  und  $\mu = 0$  entspricht dem BCC-Modell,  $\eta = 1$  und  $\mu = -2 \tan \varphi$  ergibt das FPA Modell.

Wittmer et al. stellen dann die Gleichung 1.10 in der Form

$$(\partial_z - c_1 \partial_r)(\partial_z - c_2 \partial_r)\sigma_{ij} = 0 \quad (1.24)$$

mit  $c_{1,2} = \frac{1}{2}(\mu \pm \sqrt{\mu^2 + 4\eta})$  dar.

Für die Lösungen unterscheidet man den inneren  $S \leq \frac{c_1}{c}$  und einen äußeren Bereich  $\frac{c_1}{c} < S$ . Mit der Abkürzung

$$s^* = \frac{c}{c^2 - \mu c - \eta} \quad (1.25)$$

gilt dann

$$\sigma_{zz} = s^*(1 - S)(c - \mu) \quad (1.26)$$

$$\sigma_{rr} = s^*(1 - S)\eta c \quad (1.27)$$

$$\sigma_{rz} = s^*(1 - S)\eta \quad (1.28)$$

im Inneren und

$$\sigma_{zz} = s^* \frac{c - c_1}{c_1} (c_1 - \mu S) \quad (1.29)$$

$$\sigma_{rr} = s^* \frac{c - c_1}{c_1} \eta c_1 \quad (1.30)$$

$$\sigma_{rz} = s^* \frac{c - c_1}{c_1} \eta S \quad (1.31)$$

im Außenbereich.

Im Abschnitt 4 sollen die Winkel der Hauptachsen des Spannungstensors mit diesen Ergebnissen verglichen werden; dazu werden die Eigenvektoren von  $\sigma_{ij}$  bestimmt.

Im Außenbereich haben die Eigenvektoren von  $\sigma_{ij}$  die Form

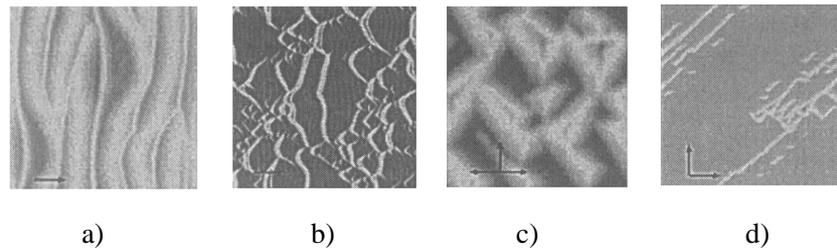
$$e_1 = \begin{pmatrix} 1 \\ f_1(\varphi, \mu) \end{pmatrix} \quad \text{und} \quad e_2 = \begin{pmatrix} f_2(\varphi, \mu) \\ 1 \end{pmatrix} \quad (1.32)$$

und sind somit unabhängig von  $S$ . Daraus kann man folgern, daß der Winkel der Hauptachsen konstant ist, zusammen mit der IFS-Bedingung ergibt sich  $\Psi(\frac{c_1}{c} < S < 1) = (\frac{\pi}{2} - \varphi)/2$ .

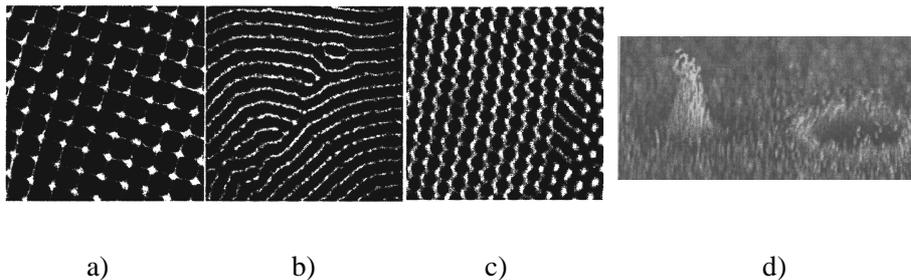
Im Innenbereich ist  $\Psi$  nur beim FPA-Modell konstant. Bei den anderen Modellen ist der Winkel abhängig von  $S$ .

## 1.4 Ausblick auf die Dynamik von Granulaten

Die bisherigen Abschnitte beziehen sich auf diejenigen Bereiche der granularen Medien, die wichtig für die vorliegende Arbeit sind. Bei den beschriebenen Systemen handelt es sich um statische Systeme, zum Teil erzeugt durch langsames Fließen der Teilchen. Dies ist aber nur ein Teil der vielfältigen Fragestellungen, die es bei der Forschung zu Granulaten gibt. In der Dynamik von Granulaten gibt



**Abbildung 1.10:** Verschiedene Dünenformen, simuliert mit dem zellulären Automaten von Nishimori. a) Wanderdünen b) Sicheldünen c) seif Dünen und c) sternförmige Dünen (aus [63] )



**Abbildung 1.11:** Muster auf der Oberfläche eines vibrierenden Granulates. a) Quadrate b) Streifen c) Hexagone d) Oszillonen (aus [100])

es noch vielfältige Probleme, bei denen Fluktuationen, Intermittenz und Instabilitäten eine wichtige Rollen spielen. Um den Formenreichtum granularer System zu zeigen, sollen einige Beispiele vorgestellt erwähnt werden.

Das sicherlich bekannteste Phänomen aus dem Bereich der Granulate ist die Bildung von Dünen und Rippeln. Der Grundlagen zu diesem Forschungsgebiet wurden von Bagnold in seinem Buch [99] “The Physics of Blow Sand and Desert Dunes” gelegt. Die Entstehung der vielfältigen Formen und ihre zeitliche Entwicklung birgt noch eine Vielzahl an unverstandenen Problemen.

Zur Musterbildung kann es kommen, wenn man eine dünne Schicht aus Kugeln vibriert. Ein anfänglich flaches Bett aus Bronzekugeln zeigt dann Strukturen wie Faraday-Instabilität oder Rayleigh-Taylor, die ähnlich jenen von anderen strukturbildenden Systemen sind. Dabei können auch lokalisierte Strukturen entstehen, die sogenannten Oszillonen (siehe Abbildung 1.11).

Die Untersuchung stark verdünnter Granulate bzw. granularer Gase ist ein weiteres fruchtbares Forschungsgebiet. Es treten verschiedene interessante Phänomene auf, wie beispielsweise die Bildung von Clustern, Strukturbildung und Schockwellen. Ein kleines Beispiel für den Effekt der Clusterbildung findet sich im Abschnitt 2.5.5.

Eine Schilderung aller Effekte würde den Rahmen dieser Arbeit sprengen. Eine Einführung bietet der Band der Sommerschule “Physics of Dry Granular Media” [101], die Konferenzbände [23, 33, 34], das Buch von Duran [102] und speziell zu granularen Gasen [103]. Ein Teil der Effekte kann bereits jetzt mit dem Programmsystem `gc2d` untersucht werden. Im folgenden konzentriert sich die vorliegende Arbeit auf granulare Aufschüttungen aus Teilchen variabler Form.



## Kapitel 2

# Simulationsmethoden

Mit der Entwicklung leistungsstarker Computer etabliert sich die Simulation als drittes Standbein der Physik. Abbildung 2.1 zeigt, wie Allen und Tildesley [104] die Rolle der Computersimulation in der Physik sehen. Durch den Vergleich von Experiment und Simulation läßt sich das der Simulation zugrundeliegende Modell bestätigen. Dann kann man aus der Simulation weitere Erkenntnisse ziehen, die dabei helfen, eine Theorie zu entwickeln und zu testen. Komplexe Simulationen wie das in dieser Arbeit vorgestellte System sind zusätzlich im Grenzgebiet zur Numerik/Informatik angesiedelt, da das Laufzeitverhalten die Größe der untersuchbaren Systeme begrenzen kann.

### 2.1 Verschiedene Methoden zur Simulation von Granulaten

Es sollen verschiedene Methoden vorgestellt werden, mit denen man ein System granularer Medien oder bestimmte Teilaspekte davon simulieren kann. Zuerst sollen bei einigen Methoden die Grundideen kurz beschrieben werden. Außerdem wird diskutiert, warum sie nicht zur Anwendung kamen. Im nächsten Abschnitt wird schließlich die verwendete *Diskrete-Elemente*-Methode erklärt.

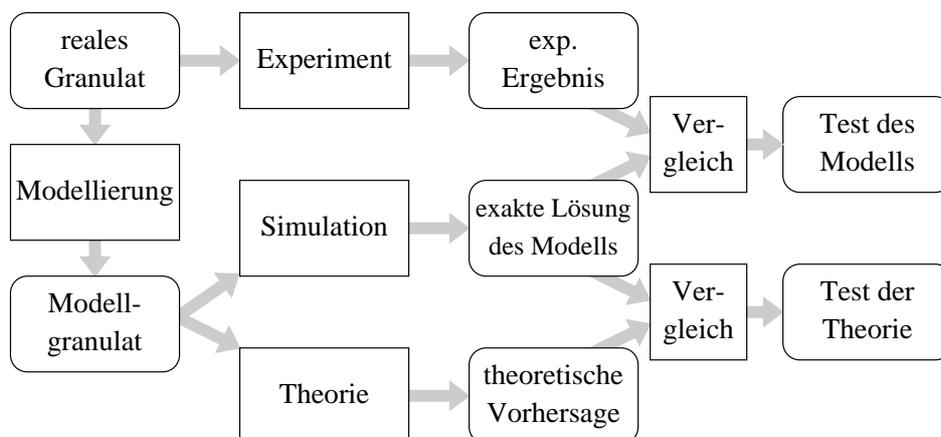


Abbildung 2.1: Die Verbindung von Experiment, Theorie und Simulation nach [104]

### 2.1.1 Abstrahierende Methoden

Eine Möglichkeit bei der Simulation ist, die Teilchenebene nicht zu berücksichtigen, sondern das Granulat als Kontinuum zu beschreiben. Die besonderen Eigenschaften des Granulates werden durch geeignete Materialgesetze berücksichtigt. Beispielsweise ist in der Geotechnik die Finite-Elemente-Methode ein bewährtes Hilfsmittel für quasi-statische Systeme. Auch der schnelle Fluß eines Granulates kann damit untersucht werden. Auf weitere Details der kontinuumsmechanischen Modellierung festkörper- oder flüssigkeitsähnlicher Systeme soll nicht weiter eingegangen werden, da immer Stoffgesetze zur Beschreibung benötigt werden. Gerade diese sollen aber durch die Simulation untersuchbar gemacht werden.

Bei der Beschreibung der Phänomenologie eines granularen Systems sind zelluläre Automaten nützliche Werkzeuge. Dabei wird die kontinuierliche Natur des Raums durch auf einem Gitter angeordnete gleichartige Zellen und die Zeit durch feste Zeitintervalle abstrahiert. Diese diskrete Struktur ist eine der wichtigsten charakteristischen Eigenschaften. Jede Zelle kann nur in bestimmten Zuständen sein<sup>1</sup>, im einfachsten Fall nur, ob sich ein Sandkorn an dieser Position befindet oder nicht. Die Entwicklung einer Zelle hängt dann nur noch von den Zuständen der Nachbarzellen<sup>2</sup> ab. Die Wahl der Regeln für diese Zustandsänderungen ist das zentrale Element der Simulation. Einer der besonderen Vorzüge der zellularen Automaten ist die Geschwindigkeit, mit der die Simulationen berechnet werden können. Die im allgemeinen sehr einfachen Regeln müssen in einer großen Matrix nur lokal angewandt werden. Die Simulation der Entstehung und Bewegung von Sicheldünen [62, 63] ist eines der schönsten Beispiele für die Simulation von Granulaten. Allerdings vereinfachen zelluläre Automaten die Kraftberechnung übermäßig stark, so daß die Berechnung von Kraftnetzwerken und Spannungsverteilungen nicht mehr sinnvoll erscheint.

Ein Modell, das die geometrischen Beschränkungen bei der Bewegung sehr stark betont, ist die von Baumann [46, 47] verwendete "Bottom-to-Top-Restructuring" Methode. Diese Methode wurde hauptsächlich zur Untersuchung von Größensegregation in verschiedenen Systemen (Aufschüttung, rotierende Trommel) verwendet. Dabei bewegt sich ein Teilchen auf der Oberfläche so lange entlang der größten Steigung hangabwärts, bis es in einem lokalen Minimum festsetzt; die Berechnung von Kräften erfolgt nicht. Die Kräfte werden nur noch abstrakt durch die Geometrie des Systems berücksichtigt.

Eine beeindruckende Teilchenzahl bei der Simulation von Gasen erreicht man durch die Nutzung von "Direct Simulation Monte Carlo" [105]. Müller [106–108] erweiterte diese Methode auf granulare Gase. Dabei wird die Dynamik der einzelnen Partikel sehr stark idealisiert. Diese Methode kommt ähnlich wie die ereignisgesteuerte Molekulardynamik ohne Kraftberechnungen aus und eignet sich gleichzeitig hervorragend zur Parallelisierung. Dabei werden Bewegung und Kollision der Teilchen vollständig entkoppelt. Durch die Zusammenschaltung zweier Cray T3E (Stuttgart und Pittsburgh) konnten 1,759,165,695 Teilchen simuliert werden.

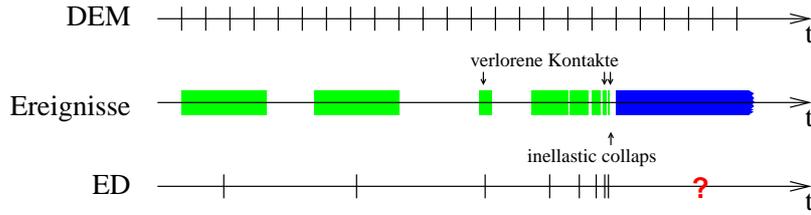
### 2.1.2 Diskrete-Elemente Methode

Zur Untersuchung von Gasen und Flüssigkeiten wurden Simulationsmethoden entwickelt, die auf der Untersuchung der individuellen Bewegung der einzelnen Teilchen beruhen [104]. Diese Methode heißt *Molecular dynamics* (MD). Dabei berechnet man die Trajektorie jedes Moleküles oder Atoms der Simulation. Erstmals auf die Untersuchung von Granulaten wurde diese Methode in den 70er Jahren von Cundall und Strack [109] angewandt. Natürlich berechnet man nicht mehr die Bewegung

---

<sup>1</sup>Manche Simulationen gehen inzwischen von diskreten Werten für die Zustände zu Gleitkommazahlen in den einzelnen Zellen über.

<sup>2</sup>Welche Zellen als benachbart betrachtet werden, hängt von der Art der Simulation ab.



**Abbildung 2.2:** Duran [102] stellt in einem ähnlichen Diagramm die Ereignisse als einzelne Zeitpunkte dar. Bei Simulationen, die ein Ergebnis mit ruhenden Teilchen liefern, können Kontakte aber auch sehr lange dauern. Erläuterungen zum “inelastischen Kollaps” werden im Abschnitt 2.1.2.1, zu den “verlorenen Kontakten” in Abschnitt 2.1.2.2 gegeben.

einzelner Atome, sondern die kleinste Einheit ist ein Sandkorn. Für diese Art der Simulation hat sich der Name “discrete element method” (DEM) eingebürgert. Prinzipiell gibt es keinen Unterschied der Methoden, aber bei Simulationen für Gase und Flüssigkeiten hat man meistens langreichweitige Wechselwirkungen, bei Granulaten typischerweise nur kurzreichweitige, doch auch hier gibt es Ausnahmen, wenn man elektrisch geladene Teilchen untersucht [18]. Ausgehend von den Newtonschen Bewegungsgleichungen muß man für jedes Polygon der Simulation

$$m_i \ddot{\mathbf{r}}_i = \mathbf{F}_i + \sum_{j=1}^n \mathbf{F}_{ij} \quad (2.1)$$

$$I_i \ddot{\varphi}_i = L_i + \sum_{j=1}^n L_{ij} \quad (2.2)$$

lösen. Dabei sind  $\mathbf{r}_i$  und  $\varphi_i$  Ort und Winkel des Polygons. Als Teilcheneigenschaften hat man deren Masse  $m_i$  und das Trägheitsmoment  $I_i$ .  $F_i$  ist die Summe aller extern wirkenden Kräfte, meistens nur die Gravitation. Für diese Arbeit wird angenommen, daß es keine externen Drehmomente  $L_i$  gibt. Daraus ergeben sich aber keine Einschränkungen, so daß man bei Bedarf auch  $L_i \neq 0$  setzen kann.  $\mathbf{F}_{ij}$  ist die Kraft, die auf das  $i$ -te Teilchen vom  $j$ -ten Teilchen ausgeübt wird. Es gilt  $\mathbf{F}_{ij} = -\mathbf{F}_{ji}$ . Drei- oder Mehrteilchenkräfte werden vernachlässigt:  $\mathbf{F}_{ij\dots k} \equiv 0$ .

Durch diese Vereinfachungen folgt aus den Gleichungen 2.1 und 2.2 dann

$$m_i \ddot{\mathbf{r}}_i = m_i g + \sum_{j=1}^n \mathbf{F}_{ij} \quad (2.3)$$

$$I_i \ddot{\varphi}_i = \sum_{j=1}^n L_{ij} \quad (2.4)$$

### 2.1.2.1 Ereignisgesteuerte Dynamik

Zu Erklärung der ereignisgesteuerten Dynamik (ED, event driven) geht man sinnvollerweise von einem sehr stark verdünnten granularen System aus (siehe z.B. Abbildung 2.42(b)). Dabei bewegen sich die Partikel zwischen zwei Kollisionen für relativ lange Zeit, vom Blickpunkt der zeitschrittgesteuerten Molekulardynamik aus gesehen, nur beeinflusst von der Gravitation, und dies auf einfachen,

analytisch zugänglichen Trajektorien. Man nimmt nun an, daß die Zeit für die Kollision zweier Partikel unendlich kurz ist. Dies bedeutet, daß man die Kugeln als unendlich hart annimmt. Die Kollision wird durch einen Kollisionsoperator beschrieben, der aus den Geschwindigkeiten und der Position die Geschwindigkeiten nach dem Stoß berechnet.

Der Algorithmus ist dann sehr einfach und elegant. Wir starten zum Zeitpunkt  $t$ :

1. Berechne den Zeitpunkt  $t_c$  der nächste Kollision eines Teilchenpaares
2. Berechne die neue Position aller Teilchen für das Zeitintervall  $\Delta t = t_c - t$

$$\mathbf{r}_i(t_c) = \frac{1}{2}\mathbf{g}\Delta t^2 + \mathbf{v}_i(t)\Delta t + \mathbf{r}(t) \quad (2.5)$$

$$\mathbf{v}_i(t_c) = \mathbf{g}\Delta t + \mathbf{v}_i(t) \quad (2.6)$$

3. Berechne die Kollision von Teilchen  $i$  und  $j$

$$\mathbf{v}'_{i,j} = \mathbf{F}(\mathbf{r}_i, \mathbf{r}_j, \mathbf{v}_i, \mathbf{v}_j) \quad (2.7)$$

4. setze  $t = t_c$
5. Springe zu 1.

Die Art, wie die Berechnung der Kollision erfolgt, bestimmt die physikalischen Eigenschaften. Auf Details soll nicht genauer eingegangen werden, sie finden sich unter anderem in [3, 102, 110] und in vielen Artikeln in [23, 33, 34, 101].

Der große Vorteil von ED ist, wie man im Beispiel in Abbildung 2.2 sieht, daß man mit sehr großen Schritten die Bereiche überspringt, in denen sich nichts ereignet. Damit ist es dann möglich, sehr große Partikelzahlen und/oder sehr lange Zeiten zu simulieren.

Die Anpassung der Zeitschritte an die stattfindenden Ereignisse führt allerdings auch dazu, daß es umso mehr Kontakte pro Zeitraum gibt, je dichter das System ist; der Algorithmus wird langsamer. Der Fall des “inelastic collapse” ist der schlimmstmögliche Fall. McNamara et al. zeigten, daß es möglich ist, bei bestimmten Konstellationen in einem granularen Gas [111–114] in einem festen Zeitraum unendlich viele Kollisionen zu haben. Der Algorithmus kann dann über diesen Zeitpunkt nicht hinausrechnen. Erst durch zusätzliche Erkennungsalgorithmen von Luding [113] kann man das System über diese Grenze hinwegsetzen. Die sinkende Effizienz bei steigender Dichte bleibt aber erhalten.

Die Annahme, daß die Kontaktzeit unendlich kurz ist, bedeutet ein weitere Grenze für diese Methode. Es ist nicht möglich, ruhende Kontakte zu haben. Systeme, in denen ein Klotz auf einer Unterlage wie Abschnitt 2.5.1 ruht, sind nicht simulierbar.

Neben diesen methodenbedingten Nachteilen gibt es auch noch ein numerisches Problem, wenn ED auf Polygone angewandt werden soll. Für zwei Kugeln  $i$  und  $j$  mit den Radien  $R_i$  und  $R_j$  berechnet man über [3]

$$|\mathbf{r}_i(t + t_{ij}) - \mathbf{r}_j(t + t_{ij})| - (R_i + R_j) = 0 \quad (2.8)$$

den Zeitpunkt  $t_{ij}$  der Kollision. Wie im Abschnitt 2.4.3 gezeigt wird, ist die Berechnung des Abstands zweier Polygone aber äußerst aufwendig: für beliebig geformte Teilchen würde dies zu einer deutlichen Steigerung der Rechenzeit führen.

ED ist damit nur dann die Methode der Wahl, wenn man verdünnte System, z.B. vibrierte Schichten [100, 115] und granulare Gase untersuchen möchte.

### 2.1.2.2 Zeitschrittgesteuerte Methode

Im Rahmen dieser Arbeit wurde die zeitschrittgesteuerte Methode (“time driven”, TD) verwendet. Dabei löst man die Gleichungen 2.3 und 2.4 mit einem Differentialgleichungslöser in festen Zeitabschnitten  $\Delta t$ . Für  $gc2d$  wurde das Predictor-Corrector Verfahren nach Gear [104, 116, 117] verwendet. Eine Beschreibung dieses Differentialgleichungslösers findet sich im Anhang F. Er bietet hohe Stabilität und hat vor allem den Vorteil, daß die Energie gut erhalten wird [118]. Dies heißt, daß der Löser nicht durch numerische Fehler Energie zuführt. Gleichzeitig erlaubt das Verfahren relativ große Zeitschritte, was der Geschwindigkeit der Simulation zugute kommt.

Anders als beim ED ist jetzt die Kontaktzeit nicht gleich Null, sondern endlich. Der Differentialgleichungslöser benötigt die aufgrund der Kontakte wirkenden Kräfte. Diese Kraftberechnung (siehe Abschnitt 2.2.2) bestimmt wie bei der ED die physikalische Relevanz des Systems.

Die Wahl des Zeitschrittes ist von entscheidender Wichtigkeit. Um die Programmeffizienz zu steigern, sollte der Zeitschritt natürlich möglichst groß sein. Andererseits benötigt der DGL-Löser circa 10 Schritte, um eine Kollision korrekt zu beschreiben.

Die Obergrenze wird wie folgt abgeschätzt: da die Kontaktkräfte wie ein harmonischer Oszillator modelliert sind, kann man aus der Eigenfrequenz  $\omega_0 = \sqrt{\frac{k}{m}}$  die Zeitdauer einer halben Periode berechnen und damit den Zeitschritt abschätzen:

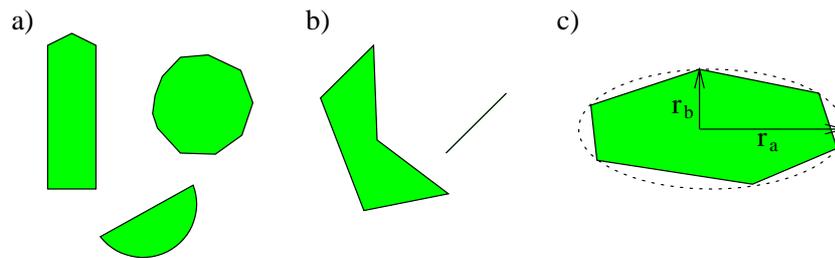
$$\Delta t \leq \frac{1}{10} \cdot \sqrt{\frac{m}{Y}} \pi \quad (2.9)$$

Dabei wird als Masse  $m$  die des leichtesten Partikels angenommen. Diese Abschätzung ist nur für rundliche Teilchen brauchbar. Werden lange, dünne Teilchen simuliert, berechnet man besser für jedes Teilchen  $i$  den maximalen Radius  $r_i$  des Kreises, der sich in das Polygon um den Schwerpunkt einbeschreiben läßt. Mit dem kleinsten Radius  $r_{min}$  wird die Obergrenze von  $\Delta T$  mit  $m = r_{min}^2 \pi \rho$  abgeschätzt. Da die Kontaktzeit auch von der Teilchenform abhängt, sind dies nur Obergrenzen. Zur Kontrolle muß die Simulation mit kürzeren Zeitschritten durchgeführt werden, da nur so überprüfbar ist, ob das System stabil arbeitet. Aus der festen Wahl der Zeitschritte resultieren aber auch einige typische Probleme:

Wenn der Zeitschritt zu groß gewählt wird, kann es sein, daß Kontakte, die kürzer als  $\Delta t$  dauern, übersehen werden (siehe Abbildung 2.2). Dabei gibt es zwei Extremfälle: die erste Möglichkeit ist ein streifender Kontakt, wie beispielsweise bei den Simulationen in Abschnitt 2.5.2. Wenn dieser Kontakt dann nur so kurz dauert, daß er übersehen würde, ist der Fehler, den man dabei macht, nicht sehr groß. Bei den Untersuchungen in Abschnitt 2.5.2 konnte dieser Bereich gar nicht gefunden werden.

Die zweite, sehr viel problematischere Situation liegt vor, wenn ein Teilchen so schnell wird, daß es mitten in ein anderes Teilchen hineinspringt. Dadurch steigt die Gesamtenergie des Systems dramatisch an, es explodiert regelrecht. Das bedeutet aber, daß ein derartiger Fehler im Endzustand des Systems deutlich sichtbar ist. Die Gefahr, daß der Fehler unentdeckt bleibt, ist gering.

Die zeitschrittgesteuerte Diskrete-Elemente Methode ist sehr gut geeignet, um statische und dynamische Systeme hoher Dichte zu untersuchen. Das Verfahren zeichnet sich durch große Zuverlässigkeit und Robustheit aus. Aus diesem Grund wurde es zur Untersuchung der in dieser Arbeit vorgestellten Systeme verwendet.



**Abbildung 2.3:** a) Zulässige Teilchenformen b) Unzulässige Teilchenformen, die Polygone sind nicht konvex oder nur Strecken c) Bestimmung der zufällig generierten Teilchen durch Platzierung der Ecken auf einer Ellipse

## 2.2 Physikalische Grundlagen der Simulation

### 2.2.1 Gestalt und Eigenschaften der Partikel

Der Versuch, einen Haufen aus Glasmurmeln zu bauen, wird zwangsläufig scheitern, da Teile des Haufens abrollen. Um in der zweidimensionalen Simulation realistische Haufen zu erzeugen, gibt es typischerweise zwei Alternativen. Entweder, man stellt die Teilchen als Kreise dar und schränkt den Rotationsfreiheitsgrad künstlich ein oder man versucht, rauhe Teilchen zu simulieren [48, 119–121].

Für die vorliegende Arbeit wurde die zweite Möglichkeit gewählt und als Form der zu simulierenden Teilchen Polygone verwendet. Dabei ergeben sich einige Herausforderungen, die bei der Simulation von Scheiben nicht auftreten: die Modellierung des Kraftgesetzes wird schwieriger und der Rechenaufwand steigt deutlich an. Werden geeignete Kraftgesetze (Abschnitt 2.2.2) und effiziente Algorithmen (Abschnitt 2.4) verwendet, bekommt man ein zuverlässiges Werkzeug, um granulare Systeme zu untersuchen.

Für die simulierbaren Polygone gibt es nur zwei Beschränkungen:

1. Das Polygon hat mindestens 3 Ecken.
2. Das Polygon ist konvex.

Die erste Forderung ist naheliegend, wenn die Art der Kraftberechnung berücksichtigt wird. Wie in Abschnitt 2.2.2 erläutert wird, ist die Abstoßung bei Kontakt proportional zu der Fläche des Überlapps zweier Polygone. Strecken oder Punkte als Teilchen sind dann nicht sinnvoll, da die Überlappfläche immer gleich Null wäre. Damit würde auf diese Partikel keine Kraft aufgrund von Kollisionen wirken. Die zweite Forderung ist eine Einschränkung des Algorithmus zur Abstandsbestimmung, gewährleistet aber außerdem, daß es zwischen zwei Teilchen nur einen Kontaktbereich gibt.

Bei vielen in dieser Arbeit vorgestellten Systemen werden große Mengen von Teilchen simuliert, die sich zwar ähnlich sein sollen, aber doch Unterschiede in Größe und Form haben müssen, um Ordnungseffekte zu umgehen. Diese Partikel werden während des Programmlaufs erzeugt, indem man per Zufall Ecken auf einer Ellipse plaziert. Wenn also in den weiteren Abschnitten Angaben zu den Radien eines Teilchens gemacht werden, beziehen sich diese immer auf die Ellipse, in die das Teilchen eingepaßt wurde. Detaillierte Angaben, wie die Form, Eckenzahl und Größenverteilung vorgegeben und bestimmt werden, findet sich im Anhang B.1.6.

Für die Simulation müssen für jedes Polygon drei Größen berechnet werden:

1. Masse
2. Schwerpunkt
3. Trägheitstensor

Es wird angenommen, daß die Polygone eine homogene Massenverteilung haben, die Dichte ist  $\varrho$ . Zur Berechnung der Masse wird das Polygon in  $N$  einzelne Dreiecke  $i$  zerlegt, deren Fläche  $f_i$  einfach zu bestimmen ist.

Dann gilt

$$m_{ges} = \int d^2r \varrho(r) = \varrho \sum_{i=1}^N f_i \quad (2.10)$$

Gleiches gilt auch für die Berechnung des Schwerpunktes. Man zerlegt das Polygon wieder in geeignete Dreiecke, für die die Masse  $m_i$  und der Schwerpunkt  $\mathbf{x}_i$  bestimmt wird. Es gilt dann

$$\mathbf{x} = \frac{\int d^2r \mathbf{r}}{A} = \frac{\sum_{i=1}^N \mathbf{x}_i f_i}{\sum_{i=1}^N f_i}. \quad (2.11)$$

Danach wird das Polygon so verschoben, daß sein Schwerpunkt bei (0,0) liegt und der Trägheitstensor wird bestimmt. Dann wird das Polygon ein weiteres mal in Dreiecke zerlegt, diesmal so, daß von allen Dreiecke  $i$  eine Ecke auf dem Ursprung liegt. Dann kann das Trägheitsmoment mit Hilfe des Steinerschen Satzes für die Rotation um diese Ecke berechnet und somit das Gesamtträgheitsmoment bestimmt werden [118].

$$I_{ges} = \sum_{i=1}^N I_i \quad (2.12)$$

Die Berechnung dieser Größen wird für jedes Partikel nur einmal bei dessen Erzeugung durchgeführt (siehe Abbildung E.1). Zu diesem Zeitpunkt werden noch viele andere Initialisierungen (Listenverwaltung, Voronoiregionen, Bounding-Boxes) für die effektive Repräsentation eines Teilchens durchgeführt. Diese sind aber für die physikalischen Eigenschaften nicht von Bedeutung. Einen kurzen Überblick bietet der Anhang E, ansonsten sind diese Schritte ausführlich im Quellcode von `gc2d` dokumentiert.

### 2.2.2 Kraftberechnung bei der Kollision

Die Kraft, die auf polygonale Partikel wirkt, zu modellieren, ist die Herausforderung bei der Simulation. Folgende Kräfte<sup>3</sup> wurden berücksichtigt:

- Gravitation
- Abstossung bei Kontakt
- Reibung

---

<sup>3</sup>Mit `gc2d` ist es auch möglich, Federn an ein einzelnes Partikel anzuhängen. Da diese Kräfte nicht zur physikalischen Modellierung der Kollision gehören, werden sie hier nicht beschrieben (siehe Anhang B).

- Kohäsion

Da von jedem Korn  $i$  die Masse  $m_i$  bekannt ist, erfolgt die Berechnung der Gravitationskraft einfach über

$$\mathbf{F}_g = \begin{pmatrix} 0 \\ m_i g \end{pmatrix}. \quad (2.13)$$

Der Wert von  $g$  kann vorgegeben werden. Solange nichts anderes erwähnt wird, ist  $g = 9.81 \text{ms}^{-2}$  verwendet worden.

Eigentlich müßte man bei der Kollision von weichen Teilchen deren Deformation berechnen. Möglich wäre dies zum Beispiel über eine Finite-Elemente-Methode. Aus Rechenzeitgründen ist dies aber mit heutigen Computern nicht praktikabel. Stattdessen erlaubt man den Teilchen während der Simulation zu überlappen. Nach dem von Hertz [122] hergeleiteten Kraftgesetz ist bei Kugeln die Kraft  $F_{ij}$  von der Eindringtiefe  $h = d - r_1 - r_2$  abhängig:

$$F_{ij} \propto h^{\frac{3}{2}} \quad (2.14)$$

Die Richtung der Kräfte ist dann parallel zur Verbindungslinie der Schwerpunkte. In gleicher Weise wird dieses Kraftgesetz auch für die Simulation von Scheiben in zweidimensionalen Systemen verwendet [3, 123]. Für Polygone muss man diesen Ansatz erweitern. Hier beschreibt die Eindringtiefe die Kollision sicher nicht mehr vollständig, sondern die Kraft hängt auch von der Geometrie der beteiligten Teilchen ab. Außerdem ist die Krafrichtung nicht mehr aus der Lage der Schwerpunkte bestimmbar, wie sich am Beispiel zweier langer Stäbe zeigt, die sich an ihren Enden berühren.

Man nimmt nun an, daß der deformierte Bereich bei der Kollision zweier Partikel  $i$  und  $j$  dem Überlappbereich entspricht. Dann kann man mit der Poissonschen Vermutung [123], daß die Kraft proportional zur verformten Fläche ist, folgern, daß die Normalkraft auch proportional zur Fläche  $A_{ij}$  des Überlappgebietes sein muß. Da die Überlappfläche  $A$  und Eindringtiefe  $h$  bei Scheiben über  $A \propto d^{\frac{3}{2}}$  zusammenhängen, gilt für diesen Fall weiterhin  $F_{ij} \propto h^{\frac{3}{2}}$ .

In Abbildung 2.4 sind die beteiligten Vektoren und Punkte dargestellt. Als Angriffspunkt  $s_{ij}$  der Kräfte wurde die Mitte der Strecke  $\overline{c_1 c_2}$  gewählt. Es ist auch möglich, stattdessen den Schwerpunkt der Fläche  $A$  zu verwenden, dies führt aber zu stärkeren Oszillationen bei verschiedenen Geometrien. Im Falle von Kugeln sind beide Punkte identisch. Normalkräfte wirken senkrecht zur Verbindungslinie, Tangentialkräfte parallel zur ihr.

Bevor die Berechnung genauer beschrieben wird, sollen noch einige praktische Größen definiert werden.

Charakteristische Länge:

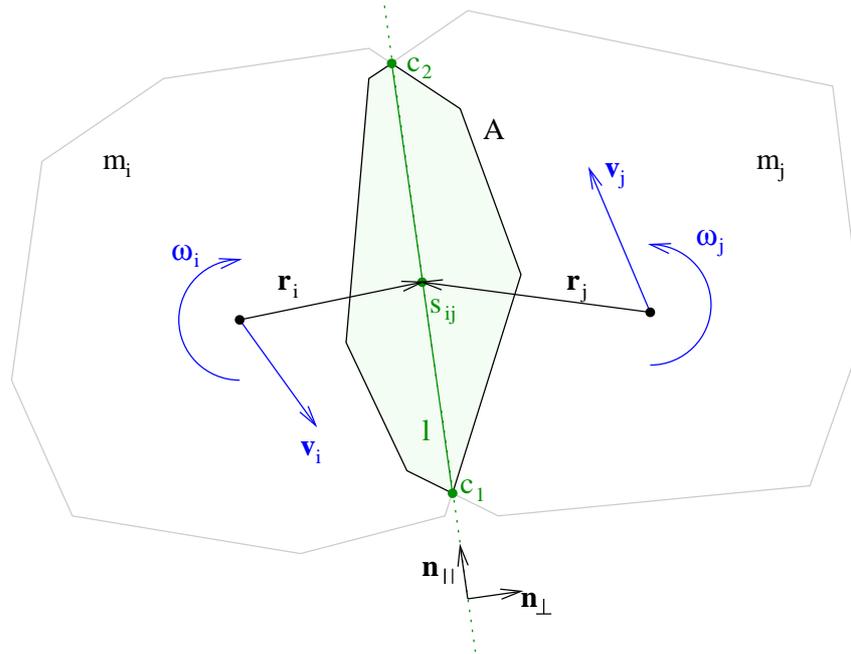
$$l = \frac{r_i \cdot r_j}{r_i + r_j} \quad (2.15)$$

Reduzierte Masse:

$$m_{\perp} = \frac{m_i \cdot m_j}{m_i + m_j} \quad (2.16)$$

Zur Berechnung der Tangentialkräfte notwendige Masse unter Berücksichtigung der Trägheitsmomente  $I_i$  und  $I_j$ :

$$m_{\parallel} = \frac{1}{\frac{1}{m_i} + \frac{1}{m_j} + \frac{r_i^2}{I_i} + \frac{r_j^2}{I_j}} \quad (2.17)$$



**Abbildung 2.4:** Die Normalen- und Tangentialrichtung bei der Kollision wird durch die Kontaktlinie (grün) bestimmt. Die Vektoren  $r_i$  und  $r_j$  deuten vom Schwerpunkt des jeweiligen Teilchens zur Mitte der Kontaktlinie.

Tangentialgeschwindigkeit am Punkt  $s_{ij}$

$$v_{\parallel} = (v_1 + v_2 + (r_1 \times \omega_1) + (r_2 \times \omega_2))n_{\parallel} \quad (2.18)$$

Effektive Eindringtiefe

$$h_{eff} = \frac{A}{l} \quad (2.19)$$

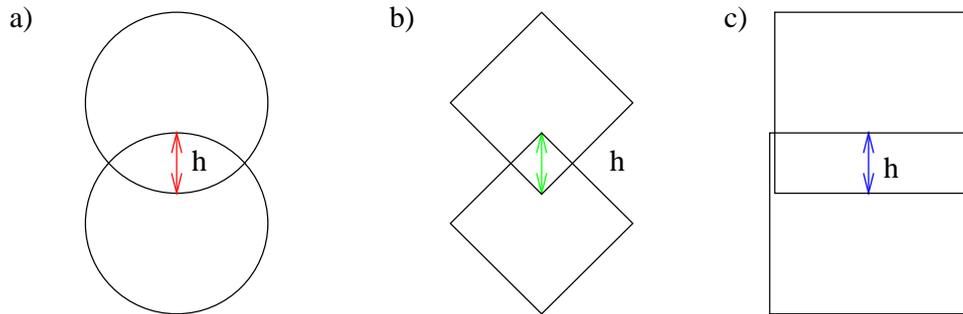
Die effektive Eindringtiefe setzt dabei die Überlappfläche und die charakteristische Länge zueinander ins Verhältnis.

Die Normalkraft, also die Kraft in Richtung von  $n_{\perp}$ , wird bestimmt durch:

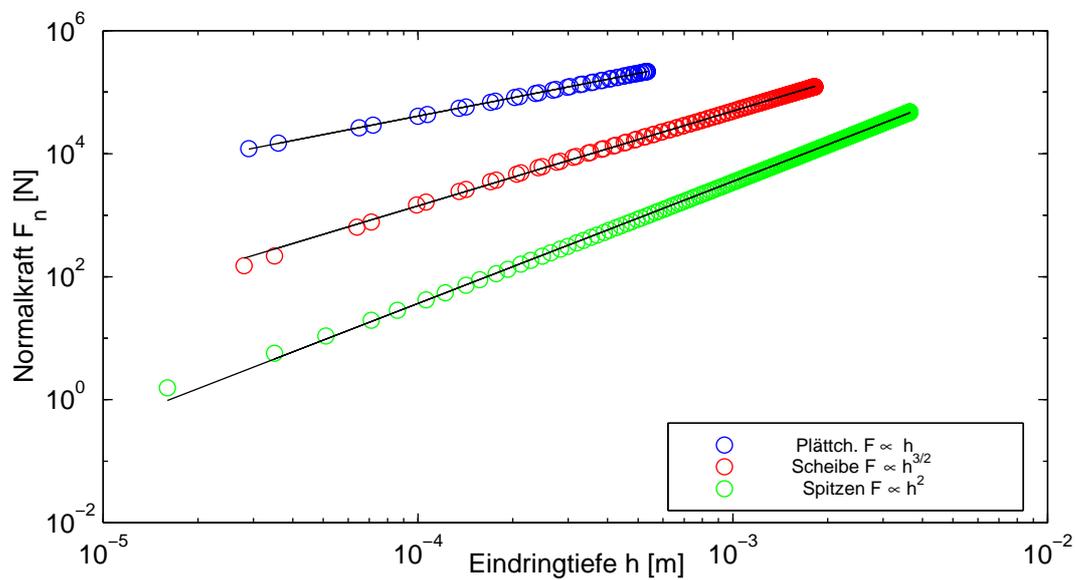
$$F_{\perp} = E \cdot h_{eff} = E \cdot \frac{A}{l} \quad (2.20)$$

Dabei ist  $E$  der Elastizitätsmodul (Young-Modul), in zwei Dimensionen hat es die Einheit N/m. Diese Modellierung der Kraft orientiert sich am Hookschen Gesetz. Die Definition der effektiven Eindringtiefe führt zu folgender Überlegung unter der Annahme des Überlapps  $A$  zweier Partikel: je kleiner die beteiligten Teilchen sind, umso kleiner ist  $l$ , entsprechend ist die Normalkraft dann größer. Dies modelliert die Tatsache, daß es leichter wird, einen Körper um ein festes Volumen zu komprimieren, je größer er ist.

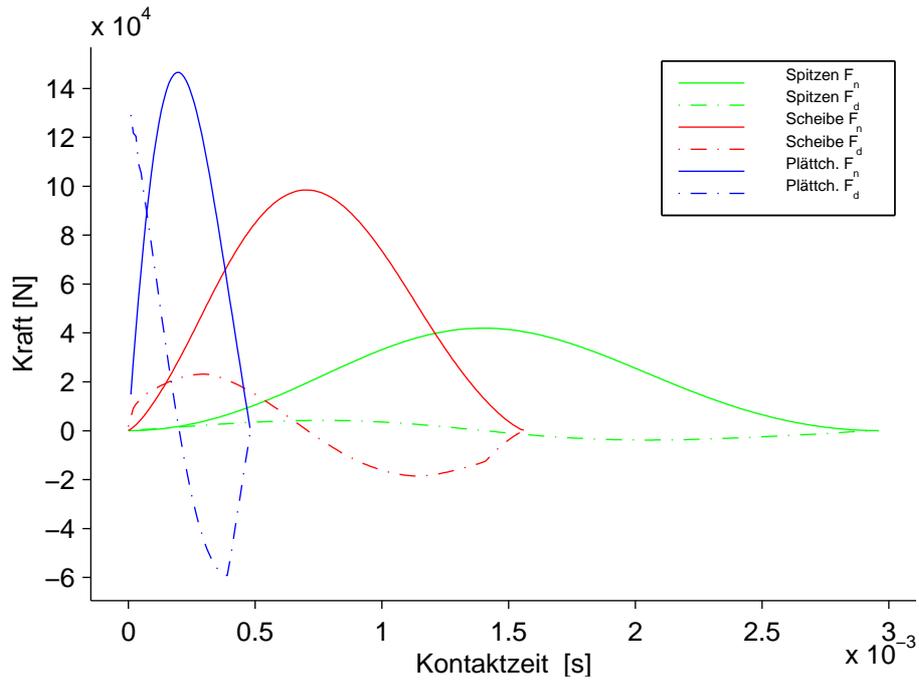
Beim zentralen Stoß zweier Partikel ist  $l$  praktisch konstant und die Kraft nimmt mit zunehmender Überlappfläche zu. Für die in Abbildung 2.5 dargestellten Geometrien wurde nun der Kraftverlauf für eine Kollision zweier identischer Teilchen simuliert. In Abbildung 2.6 ist die Abhängigkeit der Normalkraft von der Eindringtiefe  $h$  dargestellt. Für die Kollision zweier Kugeln erhält man  $F_n \propto$



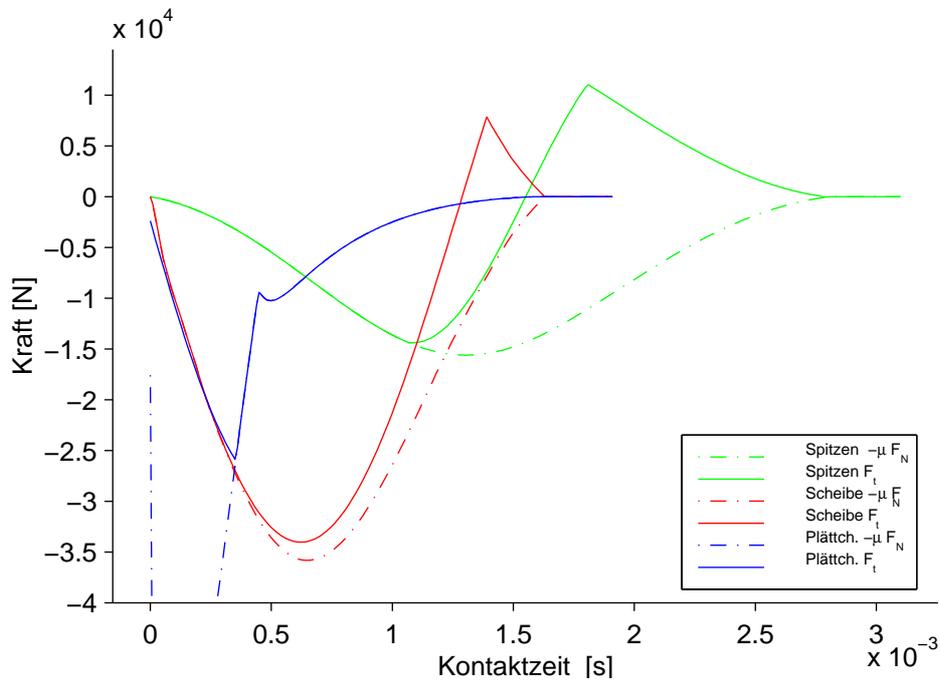
**Abbildung 2.5:** Bestimmung der Eindringtiefe bei der Kollision zweier Scheiben (rot), zweier Spitzen (grün) und zweier Plättchen (blau). Die Plättchen wurden leicht versetzt dargestellt, um die Darstellung zu verbessern. Bei den Simulationen zu den Abbildung 2.6 und 2.7 gab es diese Versetzung nicht.



**Abbildung 2.6:** Der Zusammenhang zwischen der Eindringtiefe und der Normalkraft bei der Kollision zweier Spitzen, zweier Scheiben oder zweier Plättchen.



**Abbildung 2.7:** Der zeitliche Verlauf der Normalkraft und Normalendämpfung bei der Kollision zweier Spitzen, zweier Scheiben oder zweier Plättchen.



**Abbildung 2.8:** Der zeitliche Verlauf der Tangentialkraft beim schrägen Auftreffen ( $45^\circ$ ) einer Spitze, einer Scheibe oder eines Plättchens auf eine Wand.

$h^{\frac{3}{2}}$ . Dies ist die Hertzsche Pressung [109]; die Kollision zweier Stäbe entspricht dem Hookschen Gesetz  $F_n \propto h$ . Aus dem Überlapp zweier  $45^\circ$ -Spitzen folgt für diesen Fall  $F_n \propto h^2$ .

Während des Stoßes wird, außer im voll elastischen Fall, Energie dissipiert. Im ersten Schritt wird die Dämpfung der Dämpfungskraft  $D_\perp$  des harmonischen Oszillators nachempfunden.

$$D_\perp^* = \gamma \cdot \sqrt{Em_\perp} \dot{h}_{eff} = \gamma \cdot \sqrt{Em_\perp} l \frac{\Delta A}{\Delta t} \quad (2.21)$$

Dabei wird wieder angenommen, daß  $l$  während der Kollision konstant ist.  $\gamma$  ist die Dämpfungskonstante, wobei  $\gamma = 0$  dem elastischen Fall entspricht. Bei der Kollision kann nun der Fall auftreten, daß die Dämpfung betragsmäßig größer wird als die Normalkraft. Dies kann auftreten, wenn sich die Teilchen wieder voneinander entfernen, da dann die Normalkraft überkompensiert und damit negativ werden kann. Dies würde kurzzeitig zu einer attraktiven Wechselwirkung führen. Deswegen erfolgt die Berechnung der eigentlichen Dämpfung durch

$$D_\perp = \begin{cases} D_\perp^* & \text{bei Annäherung} \\ \max[D_\perp^* ; -F_\perp] & \text{bei Separation} \end{cases} \quad (2.22)$$

der Partikel. Ein typischer Verlauf der Normalkraft und der Dämpfung ist in Abbildung 2.7 gezeigt. Simuliert wurde wieder die Situation aus Abbildung 2.5, diesmal mit  $\gamma = 0.3$ . Dargestellt ist immer der Kraftverlauf ab dem Beginn der Kollision. Man sieht, daß die Kontaktzeit  $t_{coll}$  von der Geometrie abhängig ist. Bei der Kurve für die Kollision der Quadrate wird das Abschneiden von  $D_\perp$  im Zeitraum  $[4 \cdot 10^{-4}; 5 \cdot 10^{-4}]$  sichtbar.  $D_\perp$  macht zur Zeit  $4 \cdot 10^{-4}$ s einen Knick. Ab dort ist dann  $D_\perp = -F_\perp$  und die Dämpfung wird damit betragsmäßig nicht größer als die Normalkraft.

Die Tangentialkraft und die Reibung sind entsprechend den Modellen von Cundall und Strack [109] gestaltet. Die einfachste Vorstellung ist, daß in dem Moment, in dem sich die beiden Teilchen berühren, eine Feder am Kontaktpunkt angebracht wird, die bei der weiteren Bewegung gespannt ist und somit das Teilchen bremst. Die Reibungskraft gibt aber nur eine Maximalkraft an, die auf das System wirkt, ohne daß es beschleunigt wird. Übersteigt also die Kraft der Feder die Reibungskraft, so wird sie nicht weiter gedehnt, sondern mitgezogen.

Zu Beginn der Kollision ist die Tangentialkraft  $F_\parallel(0) = 0$ . Die Kraft, die diese Feder zum Zeitpunkt  $t_c$  der Kollision ausübt, ist dann durch

$$F_\parallel^*(t_c + \Delta t) = \min \left[ F_\parallel(t_c)^* + v_\parallel \Delta t \cdot \frac{2}{7} E ; F_r(t) \right] \quad (2.23)$$

gegeben. Der Faktor  $\frac{2}{7}$  ist angepasst an eine Hertzsche Pressung ohne Rotation [124]. Im allgemeinen wurde die Coulombsche Reibung

$$F_r(t) = \mu F_\perp(t) \quad (2.24)$$

verwendet, aber auch die Modellierung eines geschwindigkeitsabhängigen Reibungsgesetzes nach Rabinowicz [125] mit

$$F_r = \mu v^{-\frac{1}{10}} F_\perp(t) \quad (2.25)$$

oder andere Formen sind möglich.

Zu betonen ist, daß die verwendeten Reibungsgesetze nicht rein viskos sind, also auch für eine Relativgeschwindigkeit  $v_\parallel = 0$  eine Reibungskraft  $F_r > 0$  existiert.

Warum nun kann man nicht einfach mit dem Ansatz  $F_{\parallel} = \mu F_{\perp}$  und die Krafrichtung entgegengesetzt zu  $v_{\parallel}$  arbeiten? Im Bereich der Oszillation sehr kleiner Geschwindigkeiten um 0 würde die Reibung bei gleichem Betrag dauernd ihr Vorzeichen wechseln; die Unstetigkeit des Kraftverlaufs führt zu numerischen Problemen.

Um dies zu umgehen, wird normalerweise ein viskoser Reibungsterm [3, 102, 126] verwendet, für kleine Geschwindigkeiten gilt dann  $F_r = \mu v_{\parallel} F_{\perp}(t)$ .

Dieses Modell führt bei polygonalen Teilchen dazu, daß ein rechteckiger Block auf einer schiefen Ebene nie zum Stillstand kommt oder, wenn er gestoppt wird, nicht liegen bleibt. Erst der Trick mit dem ‐langsamen Einschalten‐ der Reibung gewährleistet, daß ein Block auf einer schiefen Ebene zur Ruhe kommen kann (siehe Abschnitt 2.5.1). Um Tangentialschwingungen zu dämpfen, wurde ein viskoser Dämpfungsterm  $D_{\parallel}$  analog zu Gleichung 2.21 verwendet:

$$D_{\parallel} = v_{\parallel} \sqrt{\frac{2}{7} E m_{\parallel}} \quad (2.26)$$

Da auch die Summe  $F_{\parallel}^* + D_{\parallel}$  immer kleiner als die Reibungskraft bleiben soll, ist die vollständige Darstellung der Tangentialkraft  $F_{\parallel}$  bei Coulombscher Reibung dann

$$F_{\parallel}(t_c + \Delta t) = \pm \min \left[ |F_{\parallel}(t_c) + v_{\parallel} \Delta t \frac{2}{7} E + v_{\parallel} \sqrt{\frac{2}{7} E m_{\parallel}}| ; |\mu F_{\perp}(t_c)| \right] \quad (2.27)$$

wobei sich das Vorzeichen nach dem des ersten Terms richtet.

In Abbildung 2.8 wird die Entwicklung der Normal- und der Tangentialkraft exemplarisch für drei Körper (Scheibe, Plättchen mit der Kante und Dreieck mit der Spitze) bei schrägem Auftreffen auf den Boden dargestellt. Für den Reibungskoeffizienten gilt  $\mu = 0.3$ , für die Dämpfung  $\gamma = 0.5$ . Bei allen drei Körpern sieht man das Ansteigen des Betrages der Tangentialkraft mit zunehmender Dauer des Kontaktes bis zu einem jeweiligen Extremwert. Diese Minima bei den drei Kurven der Tangentialkraft sind jeweils der Zeitpunkt, bei dem sich das Vorzeichen der Normalengeschwindigkeit ändert, die Teilchen sich also wieder voneinander wegbewegen. Beim Quadrat sieht man ab dem Zeitpunkt  $t_c = 5 \cdot 10^{-3}$  eine Änderung des Kurvenverlaufs. Ab dann rutscht das Teilchen nicht mehr mit der Kante über den Untergrund, sondern kippt über die vordere Ecke ab. Gleichzeitig wird die Tangentialkraft durch die Coulombsche Reibung dominiert. Bei Dreieck und Kugel ist das Verhalten etwas anders. Zu Beginn des Kontaktes dominiert die Coulombsche Reibung. Dann, bedingt durch die Rotation, nimmt die Tangentialkraft ab und wechselt sogar das Vorzeichen. Erst gegen Ende dominiert der Term  $\mu F_{\perp}$  wieder  $F_{\parallel}$ .

Eine einfache Modellvorstellung für die Kohäsion ist, daß die Kraft proportional zur Kontaktfläche ist. In unserem zweidimensionalen Modell entspricht die Größe der Kontaktfläche der Länge  $l$  der Strecke  $c_1 c_2$  (siehe Abbildung 2.4).

Die Kohäsionskraft ist dann

$$F_{\text{coh}} = k_{\text{coh}} \cdot |l|, \quad (2.28)$$

wobei der Parameter  $k_{\text{coh}}$  die Stärke der Kohäsion bestimmt und die Einheit [N/m] hat. Im Gegensatz zu dem von Kun [127] vorgeschlagenen Modell, bei dem die attraktiven Kräfte zwischen den Schwerpunkten direkt wirken, greifen sie hier am Kontaktpunkt an und sind nicht konstant. Diese Definition der Kohäsion läßt sich sehr einfach für die vorgestellte Simulation von Polygonen realisieren, da die Geometrie des Kontaktes auch schon für die Berechnung von Normal- und Tangentialkraft notwendig ist. Der numerische Aufwand reduziert sich dadurch auf eine Addition und eine Multiplikation pro Kontakt.

Das aufgrund eines Kontaktes wirkende Drehmoment um den Schwerpunkt für Teilchen  $i$  ist

$$M = \mathbf{r}_i \times \mathbf{F} . \quad (2.29)$$

### 2.2.3 Änderung der Teilchengröße

Es kann aus verschiedenen Gründen interessant sein, die Größe des Teilchens während der Simulation zu verändern:

- Ausdehnung aufgrund von globalen Temperaturänderungen
- Materialabtrag durch Reibung
- Volumenänderung durch Austrocknung des Granulates

Dabei werden keine Formänderungen berücksichtigt, sondern eine Skalierung des Teilchendurchmessers durchgeführt. Diese wird bei der Berechnung der Position der  $j$ -ten Ecke des  $i$ -ten Teilchens berücksichtigt.  $x_{ij}, y_{ij}$  sind die Koordinaten der Ecken im Schwerpunktsystem des Teilchens,  $x_i, y_i$  dessen Schwerpunkt und  $\varphi_i$  dessen Winkel:

$$x'_{ij} = s_i \cdot (x_{ij} \cos \varphi + y_{ij} \sin \varphi) + x_i \quad (2.30)$$

$$y'_{ij} = s_i \cdot (-x_{ij} \sin \varphi + y_{ij} \cos \varphi) + y_i . \quad (2.31)$$

$s_i$  ist der Skalierungsfaktor des einzelnen Teilchens.  $s_i = 1$  gibt die Originalteilchengröße,  $0 < s_i < 1$  bedeutet eine Verkleinerung der Teilchen,  $1 < s_i$  eine Vergrößerung.

Um die Einflüsse einer globalen Temperaturänderung zu simulieren, setzt man mit

$$s_1 = s_2 = \dots = s_m = s(t) \quad (2.32)$$

alle Skalierungsfaktoren auf den selben Wert.  $s > 1$  entspricht dann einem Aufheizen des Systems in Bezug auf  $s = 1$  und  $s < 1$  ein Abkühlen. Damit sind Simulationen zu Systemen, wie sie Geminard [128, 129] untersucht, durchführbar. Dabei wird die Kompaktion eines Granulates gemessen, welches periodisch erwärmt und wieder abgekühlt wird. Die Veränderung der Größe der Partikel führt dann zu einer Volumenveränderung des Gesamtsystems. Diese Möglichkeiten, das System "aufzuheizen", ist in gc2d implementiert. Wie diese Möglichkeiten genutzt werden, ist im Anhang B.1.2.3 beschrieben. In der gleichen Art und Weise kann man auch das Schrumpfen des Granulates durch Austrocknung simulieren.

Eine anderes System, bei dem die Größenänderung der Partikel berücksichtigt werden muß, ist in Erweiterung von [130] der Materialabtrag von Eisenbahnschotter bei Belastung. Bei der Simulation bietet es sich an, die einzelnen Skalierungsfaktoren in Abhängigkeit der wirkenden Kräfte zu verändern. Ein mögliches Modell wäre, daß man den Teilchendurchmesser um  $\epsilon$  reduziert, wenn eine gewisse Kraft  $F_{max}$ , die auf das Teilchen wirken darf, überschritten wird:

$$s_i(t + \Delta t) = \begin{cases} s_i(t + \Delta t) & F_i < F_{max} \\ \max(s_{min}; (1 - \epsilon)s_i(t)) & F_i \geq F_{max} \end{cases} . \quad (2.33)$$

Kleiner als  $s_{min}$  dürfen die Teilchen dabei nicht werden, um Konflikte mit dem Zeitschritt des Differentialgleichungslösers auszuschließen. `gc2d` ist für diese Erweiterungen vorbereitet (siehe Anhang E.2), in dieser Arbeit gilt aber grundsätzlich<sup>4</sup>  $s_i = 1$ .

## 2.3 Berechnung physikalischer Daten aus den Simulationsergebnissen

“*The aim of computing is enlightenment, not numbers*” [131] könnte man als Motto über diesen Abschnitt setzen. Ein Simulationslauf von `gc2d` kann mehrere Gigabyte an Daten liefern. Aus diesen Rohdaten müssen dann erst die aus physikalischer Sicht interessanten Daten gewonnen werden, sei es durch geeignete Mittelung oder durch komplexere Berechnungen. Die dazu verwendeten Methoden werden im folgenden Abschnitt vorgestellt.

### 2.3.1 Simulationsdaten

Zuerst soll beschrieben werden, welche Daten zur Verfügung stehen. Dabei ist es unwichtig, wie diese Daten tatsächlich gespeichert werden. Eine genaue Beschreibung dieses technischen Aspektes findet sich in Anhang C.1. Die Daten, die man aus der Simulation für ein einzelnes Teilchen  $i$  erhält, lassen sich in zwei Untergruppen einteilen:

1. zeitunabhängige Daten
2. Daten für einen bestimmten Zeitpunkt  $t$

Zu den zeitunabhängigen Daten gehören in erster Linie Materialparameter wie Dichte, Elastizität und Reibungskoeffizient. Bei einer Simulation ist es zwar einfach möglich, diese Werte während eines Laufs zu verändern, in der vorliegenden Arbeit wurde von dieser Möglichkeit aber kein Gebrauch gemacht<sup>5</sup>. Ebenfalls zu den zeitunabhängigen Daten gehört die Masse eines Teilchens, die sich aus dessen Geometrie berechnen läßt. Hier muß man aber berücksichtigen, daß bei Simulationen in denen die Ausdehnung eines Granulates aufgrund externer Erwärmung beschrieben wird, sich die Fläche und Dichte des Teilchens sehr wohl ändern. Im weiteren wird, wenn nichts anderes erwähnt ist, mit konstanter Teilchenfläche gerechnet. Zeitunabhängig ist auch die Form der Partikel, also ist für ein Teilchen  $i$  auch die Zahl der Ecken  $n_i$  konstant. Bekannt ist für den Fall, daß der Schwerpunkt  $\mathbf{x}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$  im Ursprung liegt und die Rotation  $\varphi = 0$  gleich Null ist, die Position der  $n$ -ten Ecken

$$\mathbf{x}_i^n = \begin{pmatrix} x_i^n \\ y_i^n \end{pmatrix} \quad \text{mit } n = 1 \dots n_i. \quad (2.34)$$

Zeitabhängige Größen eines Partikels sind seine Position  $\mathbf{x}(t)$ , sein Rotationswinkel  $\varphi(t)$ , sowie die entsprechenden Geschwindigkeiten  $\dot{\mathbf{x}}(t)$ ,  $\dot{\varphi}(t)$  und Beschleunigungen  $\ddot{\mathbf{x}}(t)$ ,  $\ddot{\varphi}(t)$ . Die Lage der Ecken zum Zeitpunkt  $t$  berechnet sich dann für Teilchen  $i$  mit dem Schwerpunkt  $\mathbf{x}_i(t)$  und der Rotation  $\varphi_i(t)$  mittels

<sup>4</sup>Um Rechenzeit zu sparen, ist durch entsprechende Kompilierung (siehe Anhang E.2) sichergestellt, daß die Multiplikation mit 1 garnicht erst ausgeführt wird

<sup>5</sup>Ein Ausnahme bildet der Film `mid9f.mpg` auf der beigelegten CDROM. Hier wurde während der Simulation die Reibung schlagartig auf Null reduziert, was zur Fluidisierung eines Sandhaufens führt.

$$\mathbf{E}_i^n(t) = \begin{pmatrix} x_i^n \cos \varphi_i(t) + y_i^n \sin \varphi_i(t) \\ -x_i^n \sin \varphi_i(t) + y_i^n \cos \varphi_i(t) \end{pmatrix} + \begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix}. \quad (2.35)$$

Neben den Daten für einzelne Teilchen sind auch die kompletten Daten der Kräfte zu einem bestimmten Zeitpunkt bekannt. Für jede Kraft  $i$  sind

1. der Ansatzpunkt der Kraft  $\mathbf{x}_i^f$
2. die Größe der Kraft  $\mathbf{F}_i$

bekannt, ebenso, auf welche Teilchen die Kräfte wirken.

### 2.3.2 Berechnung weiterer Größen

In den Abschnitten 3 und 4 wird es nötig werden, den Druck auf den Boden bzw. die Seitenwand des Systems zu berechnen. Dazu werden aus allen Kräften des Systems zuerst die Kräfte  $\mathbf{F}_i$  an den Punkten  $\mathbf{x}_i^f$  ausgesucht, die auf die zu untersuchende Wand wirken. Die Seitenwand bzw. der Boden habe die Länge  $l$ . Der mittlere Druck auf einen Boden ergibt sich dann aus

$$p_{Boden} = \frac{1}{l} \sum_i \mathbf{n}_y \mathbf{F}_i, \quad (2.36)$$

der auf eine Seitenwand aus

$$p_{Wand} = \frac{1}{l} \sum_i \mathbf{n}_x \mathbf{F}_i. \quad (2.37)$$

Interessant ist aber nicht nur der mittlere Druck, sondern auch die örtliche Verteilung  $P(x)$ . Um dies zu erreichen, bestimmt man den Druck in einem geeigneten Messintervall der Breite  $2b$ .

Dazu definieren wir eine Testfunktion  $\Theta(x, x')$  mit

$$\Theta(x) = \begin{cases} 1 & x - b \leq x' \leq x + b \\ 0 & \text{sonst} \end{cases}. \quad (2.38)$$

Angewandt auf die Angriffspunkte der Kräfte ergibt  $\mathbf{F}_i \Theta(x, \mathbf{n}_x \mathbf{x}_i^f)$  alle Kräfte, die innerhalb eines bestimmten Intervalls um  $x$  angreifen.

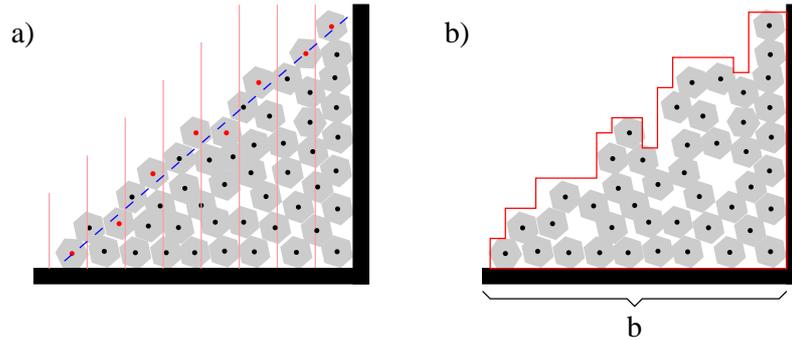
Der lokale Druck auf einen Boden ist

$$p_{Boden}(x) = \frac{1}{2b} \sum_i \mathbf{n}_y \mathbf{F}_i \Theta(x, \mathbf{n}_y \mathbf{x}_i^f), \quad (2.39)$$

der auf eine Wand

$$p_{Wand}(y) = \frac{1}{2b} \sum_i \mathbf{n}_x \mathbf{F}_i \Theta(y, \mathbf{n}_x \mathbf{x}_i^f). \quad (2.40)$$

Bei der Bestimmung des Drucks gilt es, einen geeigneten Wert für die Größe des Meßintervalls zu finden. Ist das Intervall zu klein, wird über zu wenige Kräfte gemittelt und in der Druckkurve sind



**Abbildung 2.9:** Die Bestimmung des Böschungswinkels kann a) über einen Fit an die obersten Partikel in einzelnen Intervallen oder b) über die Fläche des Systems erfolgen.

die einzelnen Kräfte zu erkennen. Ist das Meßintervall zu groß, so werden Details wie das lokale Druckminimum bei einem Sandhaufen weggemittelt. Die Erfahrung hat gezeigt, daß 10-20 Kontakte pro Intervall ein guter Kompromiß sind. Es muß aber immer durch Variation der Intervallbreite überprüft werden, ob vermeintliche Effekte nicht nur Artefakte sind. Anders als im realen Experiment ist es auch möglich, die Meßbereiche überlappen zu lassen, so daß eine bessere örtliche Auflösung möglich ist.

Aus der Simulation sind alle wirkenden Kräfte bekannt und es ist einfach, entsprechende Wahrscheinlichkeitsverteilungen zu bestimmen. Man berechnet die Richtungen  $\alpha_i$  der Kräfte  $\mathbf{F}_i$  über

$$\tan \alpha_i = \frac{F_i^y}{F_i^x} \quad (2.41)$$

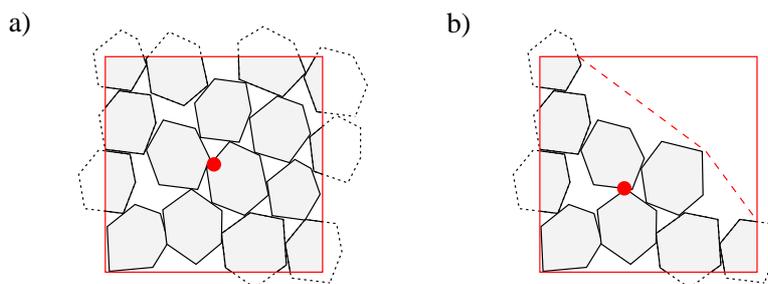
und bestimmt dann deren Wahrscheinlichkeiten  $P(\alpha)$ . Diese Daten werden sinnvollerweise in einem Polarkoordinatensystem aufgetragen (siehe z.B. Abbildung 4.2(e) oder 4.5(e)). Hier ist sehr leicht zu erkennen, ob das Kraftnetzwerk anisotrop ist. Da an jedem Kontakt zwei entgegengesetzte Kräfte angreifen<sup>6</sup>, gilt  $P(\alpha) = P(\alpha \pm \pi)$ . Die Verteilung der Größe der Kräfte  $P(F / < F >)$  ist auch direkt berechenbar.

Da bekannt ist, auf welches Teilchen  $m$  jede der Kräfte wirkt, kann für ein bestimmtes Teilchen abgezählt werden, wieviele Kontakte es hat. Diese Zahl heißt Koordinationszahl  $K$ , ihre Verteilung  $P(K)$  ist wiederum direkt berechenbar.

Zur Bestimmung des Böschungswinkels wurden zwei Methoden verwendet:

Im Fall eines Haufens, der auf einem Untergrund aufgeschüttet wurde, ist die Oberfläche hinreichend glatt. Ein Beispiel ist der Haufen in Abbildung 4.17. Von jedem Teilchen  $i$  ist dessen Schwerpunkt bekannt. Dann wird für Intervalle an den Orten  $x_i$  (ähnlich wie Gleichung 2.38) der jeweils höchstgelegene Schwerpunkt  $h(x_i)$  bestimmt. An  $h(x)$  wird eine Gerade angefitet, über deren Steigung der Böschungswinkel  $\Phi$  berechnet wird. Eigentlich sollte bei der Bestimmung der Oberfläche die höchstliegende Ecke gesucht werden, da bei der Berechnung mit Hilfe des Schwerpunktes die Oberfläche  $h_{Ecken}(x) = h_{Schwerpunkt}(x) + \epsilon$  zu tief angesetzt wird. Da aber nur die Steigung interessant ist, ist  $\epsilon$  unwichtig. Die Berechnung über die Schwerpunkte ist im allgemeinen um den Faktor 10-20 schneller. Problematisch könnte diese Vereinfachung erst werden, wenn man einen Haufen aus sehr langen Stäben aufschütten würde.

<sup>6</sup>Gravitationskräfte seien hier unberücksichtigt.



**Abbildung 2.10:** Die Testfläche (rot) zur Berechnung des lokalen Füllgrads. Grau eingezeichnete Flächen werden bei der Berechnung der lokalen Dichte berücksichtigt und der gemittelte Füllgrad wird dann für den rot markierten Schwerpunkt der Testfläche angegeben. b) Die Testfläche schneidet die freie Oberfläche des Granulates; die Fläche wird entsprechend verkleinert (gestrichelte rote Linie).

Wenn, wie im Abschnitt 4.4 die Oberflächen allerdings sehr zerklüftet sind, ist eine andere Methode, analog der in [49, 132] verwendeten, günstiger. Allerdings muß hierzu die Breite  $b$  der Basis der Aufschüttung bekannt sein. Aus der Fläche  $a$ , die das Granulat einnimmt, ergibt sich der Böschungswinkel über

$$\tan \varphi = 2 \cdot \frac{a}{b^2} . \quad (2.42)$$

Der Vorteil diese Methode ist, daß sie integralen Charakter hat und eine Bestimmung der Oberfläche nicht notwendig ist. Dadurch umgeht man die Probleme mit stark zerklüfteten Granulathaufen bei sehr starker Kohäsion.

Eher der Visualisierung als der Auswertung dienend ist die Darstellung des Systems selbst bzw. des Kraftnetzwerkes; “eigenartige” Simulationen lassen sich so frühzeitig erkennen.

Bei der Darstellung des Systems selbst bekamen die Teilchen immer eine Farbe entsprechend ihrer internen Teilchennummer zugewiesen. Da diese Nummer im allgemeinen aufsteigend vergeben wird, haben bei den Sandhaufen Teilchen gleichen “Alters” ähnliche Farben. Damit kann man die “Geschichte” des Systems erkennen. So ist zum Beispiel an dem Haufen in Abbildung 4.16 zu erkennen, daß wohl im Laufe der Entstehung dieses Haufens ein Teil des Haufens zur Seite gerollt/gerutscht ist und sich darüberliegendes Material wie ein Keil dazwischengeschoben hat. Der Grund dafür ist, daß die Teilchen sehr glatt waren und exakt dieselbe Größe haben.

Radji (siehe Titelseite von [23]) verbindet zur Darstellung der Kraftnetzwerke die Schwerpunkte zweier sich berührender Teilchen. Die Breite der Linie ist dann proportional zur Kraft. Hat man aber nichtrunde Teilchen, so wird im allgemeinen der Aufpunkt der Kraft nicht auf dieser Linie liegen. Deswegen wurde stattdessen für jeden Kontakt je eine Linie vom Schwerpunkt zum Aufpunkt der Kraft gezeichnet. Auch hier ist die Dicke und Farbigekeit proportional zum Betrag der Kraft. Durch die Darstellung (z.B. Abbildung 4.20(a)) lassen sich sehr einfach Pfade darstellen, entlang derer die Kräfte wirken.

### 2.3.3 Berechnung der lokalen gemittelten Dichte

Betrachtet man einen Ausschnitt aus einer granularen Aufschüttung, so ist zu sehen, daß die einzelnen Körner den vorhandenen Raum nicht wie eine Flüssigkeit oder ein Gas ausfüllen, sondern

daß große Zwischenräume entstehen. In zwei Dimensionen beschreibt die Packungsdichte, wieviel Prozent der Fläche vom Festkörper bedeckt werden. Die Packungsdichte ist nicht nur von Form und Größenverteilung der Teilchen abhängig, sondern auch von der Entstehungsgeschichte des Systems. Dementsprechend ist es interessant, diesen Wert auch örtlich aufgelöst zu bestimmen.

Dazu wird eine Testfläche  $A$  über das System gelegt und für alle Teilchen  $i$  der Simulation berechnet, welcher Anteil  $\alpha_i$  der Teilchenfläche  $a_i$  innerhalb der Testfläche liegt. Dabei gilt

$$\alpha_i = \begin{cases} 1 & \text{Teilchen } i \text{ vollständig in } A \\ 0 & \text{Teilchen } i \text{ außerhalb von } A \\ 0 < \alpha_i < 1 & \text{sonst} \end{cases} . \quad (2.43)$$

Der einfachste Ansatz für die lokale relative Dichte ist dann

$$\varrho'_{rel} = \frac{\sum_i \alpha_i a_i}{A} . \quad (2.44)$$

Für den Fall, daß wie in Abbildung 2.10a) die Testfläche komplett von Granulat umgeben ist, ist diese Methode zuverlässig. Problematisch wird es aber, wenn die Testfläche am Rand des Systems liegt, also nicht komplett ausgefüllt wird. In Abbildung 2.10b) würde dann die lokale Dichte extrem unterschätzt werden. Deswegen wird in einem zweiten Schritt die Testfläche verkleinert. Dazu wird die Menge  $E_1$  der innerhalb von  $A$  liegenden Ecken derjenigen Teilchen bestimmt, für die  $\alpha_i > 0$  gilt. Dann berechnet man die Menge der Schnittpunkte  $E_2$  der Teilchen mit dem Rand. Von den Punkten  $E_1 \cup E_2$  wird dann die konvexe Hülle und deren Fläche  $A_k$  bestimmt.

Die verbesserte Definition für die lokale Dichte lautet dann

$$\varrho_{rel} = \frac{\sum_i \alpha_i a_i}{A_k} . \quad (2.45)$$

Im Inneren des Systems sind Gl. 2.44 und Gl. 2.45 gleichwertig. Am Rand unterschätzt die Variante mit der konvexen Hülle die Dichte aber deutlich weniger.

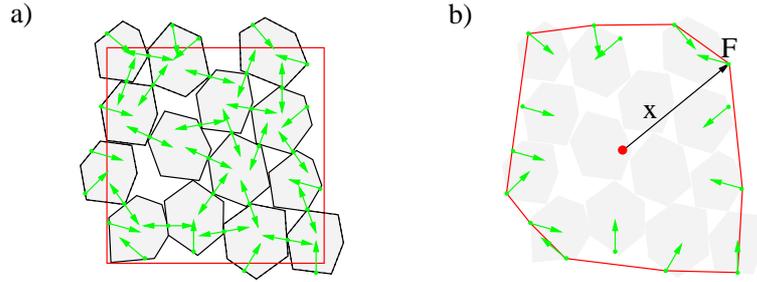
### 2.3.4 Berechnung von Spannungstensoren aus den Kräften

Zur Bestimmung der Spannungstensoren im Granulat stehen die Kräfte und deren Angriffspunkte zur Verfügung. Zuerst soll gezeigt werden, wie daraus der mittlere Spannungstensor in einem einzelnen Partikel berechnet wird, anschließend wird die Rechnung auf größere Gebiete mit mehreren Teilchen erweitert; es gilt immer  $i, j, k, l \in \{x, y\}$ . Im Gleichgewichtszustand gilt für einen deformierten Körper

$$\frac{\partial \sigma_{il}}{\partial x_l} = 0 . \quad (2.46)$$

Zur Berechnung des mittleren Spannungstensors wird Gleichung 2.46 mit  $x_j$  multipliziert und dann über das Volumen  $V^p$  des Teilchens integriert [122].

$$\int \frac{\partial \sigma_{il}}{\partial x_l} x_j dV^p = \int \frac{\partial (\sigma_{il} x_j)}{\partial x_l} dV^p - \int \underbrace{\sigma_{il} \frac{\partial x_j}{\partial x_l}}_{=\delta_{jl}} dV^p = 0 \quad (2.47)$$



**Abbildung 2.11:** a) Alle Teilchen, deren Schwerpunkt innerhalb einer Testfläche liegt, werden zur Spannungsberechnung herangezogen. b) Kräfte (grün) die im Inneren wirken, tragen zur Spannungsberechnung nicht bei. Die Testfläche wird den Aufpunkten  $x$  der Kräfte angepaßt.

Mit dem Gausschen Integralsatz wird das Volumenintegral in ein Oberflächenintegral umgeschrieben.

$$\oint \sigma_{il} x_j df_l = \int \sigma_{ij} dV^p \quad (2.48)$$

Da an der Oberfläche des Partikels nur einzelne Kräfte angreifen, kann die linke Seite der Gleichung als Summe über alle angreifenden Kräfte geschrieben werden.

$$\langle \sigma_{ij}^p \rangle = \frac{1}{V^p} \sum_F x_j F_i = \frac{1}{V^p} \sum_F x_i F_j \quad (2.49)$$

Die Symmetrie des Spannungstensors erlaubt die Vertauschung der Indizes bei  $x$  und  $F$ . Die Spannungstensenoren in einem Teilchen werden jetzt über ein größeres Volumen  $V$  gemittelt. Da der die Partikel umgebende Leerraum spannungsfrei ist, gilt

$$\langle \sigma_{ij} \rangle = \frac{1}{V} \sum_p V^p \langle \sigma_{ij}^p \rangle = \frac{1}{V} \sum_p \sum_F x_j F_i . \quad (2.50)$$

Die Doppelsumme kann nun durch Zerlegung der Kontaktvektoren und anschließende Summation über alle Kontakte vereinfacht werden [123]. Man kann aber auch anschaulich (siehe Abbildung 2.11 argumentieren. Zuerst wird der Ursprung des Koordinatensystems o.B.d.A. in den Schwerpunkt der Testfläche gesetzt. Da im Inneren an einem Kontakt zwei entgegengesetzt gerichtete und gleich große Kräfte wirken, tragen diese zur mittleren Spannung nichts bei. Es bleiben also nur die Kräfte an der Oberfläche des Testkörpers übrig. Über diese wird nun summiert:

$$\langle \sigma_{ij} \rangle = \frac{1}{V} \sum x_i F_j \quad (2.51)$$

Cundall und Strack [109] haben in ähnlicher Weise den Spannungstensor in einem Granulat berechnet. Um zu gewährleisten, daß  $\langle \sigma_{ij} \rangle$  auch bei der Simulation symmetrisch ist, verwendet man bei der Auswertung

$$\langle \sigma_{ij} \rangle = \frac{1}{2V} \sum (x_i F_j + x_j F_i) . \quad (2.52)$$

Bei der Auswertung wird dann zuerst für ein beliebiges Rechteck, das ungefähr die gewünschte Größe hat, bestimmt, welche Partikel innerhalb dieser Fläche liegen. Für die Kontakte dieser Teilchen wird nun die Summe aus Gleichung 2.52 bestimmt. Es muß jetzt noch die Größe der Testfläche festgelegt werden. Die Kräfte sollen eigentlich an der Oberfläche angreifen. Aus der Fläche des Rechtecks erhält man zu niedrige Werte, da die Kräfte meist außerhalb der Testfläche angreifen. Kommt man an die Oberfläche des Systems (ähnlich Abbildung 2.10b), ist die Fläche  $V$  deutlich zu klein.

Zu betonen ist, daß damit kein eigentlicher Fehler macht wird. Die Berechnung der Spannungstensor für diese Fläche ist korrekt. Das Problem ist vielmehr, daß über mehrere Teilchen gemittelt wird und somit die Testfläche möglichst genau den Teilchen entsprechen soll. Deswegen wird als Fläche  $V$  dann diejenige der konvexen Hülle um die Aufpunkte der Kraft genommen. Dies ist vernünftig, da dann alle Kräfte an oder in der Testfläche angreifen. Eine weitere Möglichkeit wäre, die konvexe Hülle um die Teilchen wie bei der Dichtebestimmung zu benutzen. Die Unterschiede zwischen beiden Flächen sind im allgemeinen marginal, lediglich ist diese Methode bei der Auswertung deutlich rechenzeitaufwendiger.

Bei der Größe der Testfläche ist man, wie bei der Druckbestimmung, wieder auf Erfahrungswerte angewiesen; mit 100-200 Teilchen pro Testfläche erhält man im allgemeinen sehr gute Ergebnisse. Vorteilhaft für die Auswertung ist natürlich, daß die Testflächen sich überlappen dürfen, so erreicht man eine hohe räumliche Auflösung.

Durch eine Hauptachsentransformation kann man dann  $\langle \sigma_{ij} \rangle$  auf Diagonalgestalt bringen. Daraus erhält man das Spannungsellipsoid praktisch ohne zusätzliche Arbeit. Damit kann man den Winkel der Hauptachsen des Spannungsellipsoiden<sup>7</sup>  $\Psi$  (siehe Abschnitt 1.3) berechnen.

## 2.4 Methoden zur effizienten Simulation

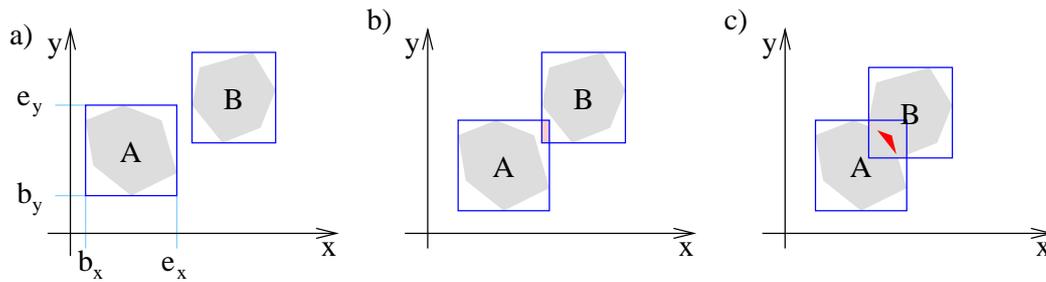
### 2.4.1 Grundlegende Konzepte

In den Abschnitt 2.1 und 2.2 wurde erklärt, welchen Regeln eine DEM-Simulation für polygonale Partikel zu folgen hat. Dabei blieb allerdings der Zeitaufwand unerwähnt, der für eine derartige Rechnung notwendig ist. Entscheidend für eine aussagefähige Simulation ist eine so große Anzahl an Partikeln, daß die Bestimmung physikalisch interessanter Größen durch die Mittelung über hinreichend große Bereiche möglich ist.

Selbstverständlich sollte eine aufwendige Gleitkommaberechnungen, wenn das Ergebnis mehrfach benötigt wird, nur einmal berechnet werden. Eines von unzähligen Beispielen ist die Berechnung der reduzierten Masse zweier Teilchen bei der Kollision (Gleichung 2.16). Diese Größe wird nur beim ersten Auftreten der Kollision berechnet und in den weiteren Zeitschritten wiederverwendet.

Die Entscheidung, welche Algorithmen verwendet werden, legt schon im Vorfeld das grundlegende Laufzeitverhalten der Simulation fest. Effiziente Algorithmen können hier die Möglichkeiten bieten, sehr große physikalische Systeme zu bewältigen. Zuerst wird eine worst-case-Abschätzung versucht, um zu überprüfen, ob sich der Arbeits- und Zeitaufwand für komplexere Algorithmen lohnt. Zu simulieren seien 1000 Teilchen mit je 10 Kanten. Würden alle 500000 möglichen Kollisionen einzeln überprüft, wobei jeweils  $10^2 = 100$  Strecken auf einen Schnittpunkt zu prüfen sind, hätte man  $5 \cdot 10^7$  Streckenschnittpunkte zu berechnen. Eine einzelne Berechnung dauert ca.  $5 \cdot 10^{-6}$  Sekunden. Insgesamt würde dann die Berechnung eines einzelnen Zeitschritts ca. 250 Sekunden dauern. Typische Simulationen in der vorliegenden Arbeit haben etwas  $10^6$  Zeitschritte, die Simulation würde also etwa 8 Jahre dauern. Dies überschreitet aber deutlich den Zeitrahmen einer Doktorarbeit.

<sup>7</sup>In zwei Dimensionen der Spannungsellipse.



**Abbildung 2.12:** Dargestellt ist die verwendete Form von Bounding-Boxes. Um eine Bounding-Box vollständig zu beschreiben, genügen die Werte  $b_x$ ,  $b_y$ ,  $e_x$  und  $e_y$ .

a) Die Bounding-Boxes berühren sich nicht, so daß die darin enthaltenen Teilchen auch nicht kollidieren können. b),c) Berühren sich die Bounding-Boxes, ist keine Aussage über die Kollision der enthaltenen Partikel mehr möglich. Die weitere Kontaktberechnung muß durch den closest-feature-Algorithmus übernommen werden.

Betrachtet man das Problem nun genauer, so sieht man, daß für die meisten Kräfte zwischen Teilchen  $i$  und  $j$  aber  $F_{ij} = 0$  gilt. Bei typischen Koordinationszahlen von 4 (siehe Abschnitt 3 und 4) muß man für  $N$  Teilchen dann  $N^2$  Kraftvektoren bestimmen, davon sind aber nur  $4N$  ungleich Null. Ziel eines verbesserten Verfahrens muß es also sein, möglichst viele Kräfte, für die  $F_{ij} = 0$  gilt, sehr schnell oder, noch besser, garnicht zu berechnen. Verschiedene typische Eigenschaften der beabsichtigten Simulationen kann man sich dabei zunutze machen:

1. Es gibt keine langreichweitigen Wechselwirkungen. Sind zwei Partikel soweit voneinander entfernt, daß sich zwei einfache, die einzelnen Partikel umgebende, Körper nicht berühren, so gilt  $F_{ij} = 0$ . Dies macht man sich bei der Verwendung von Bounding-Boxes zunutze.
2. Die Position der Schwerpunkte der Teilchen ändert sich vom Zeitpunkt  $t$  zum nächsten  $t + \Delta t$  nur sehr geringfügig. Mit sehr großer Wahrscheinlichkeit gilt dann

$$F_{ij}^{t+1} = 0 \quad \text{wenn} \quad F_{ij}^t = 0 \quad \text{und} \\ F_{ij}^{t+1} \neq 0 \quad \text{wenn} \quad F_{ij}^t \neq 0.$$

Man kann durch das im Folgenden vorgestellte Sortierverfahren diejenigen Einträge identifizieren, für die die obige Aussage nicht gilt. Alle anderen Einträge müssen nicht verändert<sup>8</sup> werden und können aus dem letzten Zeitschritt übernommen werden.

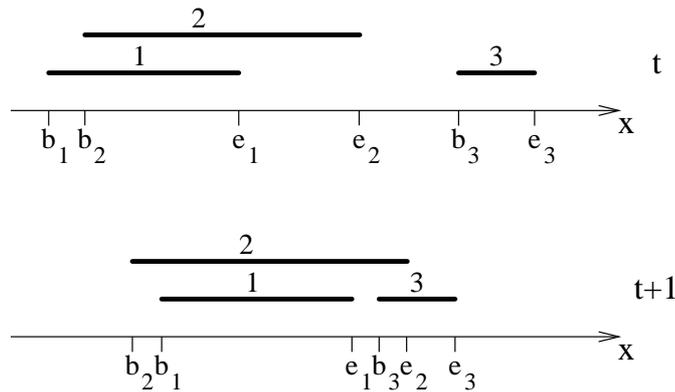
3. Die Lage zweier Partikel zueinander ändert sich auch nur sehr geringfügig. Auf der Basis dieser Eigenschaft gibt es ein schnelles Verfahren, daß den Abstand zweier Partikel zueinander bestimmen kann.

Zusammenfassend kann man sagen, daß versucht wurde, alte Information, soweit möglich, nur anzupassen oder als optimale Anfangsbedingung zu verwenden, anstatt diese zu jedem Zeitschritt neu zu berechnen.

## 2.4.2 Bounding-Boxes und Sortierverfahren

Es gibt verschiedene Verfahren, um die Partikel in der Nachbarschaft eines Teilchens zu bestimmen. Nur mit diesen Teilchen aus der näheren Umgebung wird dann die aufwendige Kontaktprüfung

<sup>8</sup>Daß sich der Betrag von  $F_{ij}$  ändert, ist hier noch nicht ausschlaggebend.



**Abbildung 2.13:** Dargestellt ist ein einfaches Beispiel von drei Bounding-Boxen zu zwei aufeinanderfolgenden Zeitschritten. Trotz der Änderung der Position der Anfangs- und Endkoordinaten ist die Reihenfolge weitestgehend gleich geblieben. Nur  $e_2$  und  $b_3$  haben den Platz getauscht, daran erkennt man einen neu hinzukommenden Kontakt.

durchgeführt. Zum Beispiel ist es möglich, über das zu simulierende Gebiet ein Gitter zu legen und nur mit Teilchen in den benachbarten Maschen zu vergleichen. Dies hat aber Nachteile, wenn Teilchen unterschiedlicher Größe verwendet werden. Ein Vergleich dieses Verfahrens mit dem in der vorliegenden Arbeit verwendeten findet sich in [133].

Bei der Verwendung von Bounding-Boxen wird das zu prüfende Element mit einer geeigneten einfachen geometrischen Form umgeben. Dann wird geprüft, ob sich die Bounding-Boxen von zwei Teilchen berühren. Ist dies nicht der Fall, dann können sich auch die entsprechenden Partikel nicht berühren. Überschneiden sich die Bounding-Boxes, kann nicht entschieden werden, ob sich die Teilchen tatsächlich berühren oder nur sehr nahe sind. Eine weitere Kontaktprüfung ist notwendig.

Für das zugrunde liegende Sort-and-Sweep Verfahren (siehe [118, 133, 134]) haben die Bounding-Boxen die Form eines Rechtecks, dessen Seiten parallel zu den Koordinatenachsen sind. Die Werte  $b_x$ ,  $b_y$ ,  $e_x$  und  $e_y$  reichen aus, um sie zu beschreiben (siehe Abbildung 2.12). Diese Methode basiert auf einem Sortierverfahren, dem Insertion-Sort [118, 135]. Ziel ist es, eine Liste<sup>9</sup>  $K_t^B$ , die für den Zeitpunkt alle Kollisionen von Bounding-Boxen enthält, zu erstellen. Je nach verwendetem Sortierverfahren kann der Zeitaufwand bei unsortierten Listen auf  $T \propto N \log N$  reduziert werden. Beim Insertion-Sort erreicht man für unsortierte Listen nur  $T \propto N^2$ , für sortierte Listen aber  $T \propto N$ . Allerdings wird dieses Verfahren nur noch zur Initialisierung des Systems verwendet.

Anstatt nämlich die Liste in jedem Zeitschritt neu zu erstellen, wurde eine Methode entwickelt, um die Liste nur noch zu korrigieren. Zuerst soll dies am eindimensionalen Fall erläutert werden, danach folgt die Verallgemeinerung auf zwei oder mehr Dimensionen.

Im eindimensionalen Fall sind die Bounding-Boxen nur Intervalle auf der X-Achse. In Abbildung 2.13 sieht man drei Intervalle. Die Liste der Kollisionen enthält nur das Paar (1, 2). Anfang  $b$  und Ende  $e$  der Bounding-Boxen sind auf der X-Achse markiert. Aus dem vorhergehenden Schritt hat man eine vorsortierte Liste dieser Punkte:  $b_1, b_2, e_1, e_2, b_3, e_3$ .

Generell gilt, wenn

1.  $b_m \leq b_n \leq e_m \leq e_n$  oder
2.  $b_m \leq b_n \leq e_n \leq e_m$  oder

<sup>9</sup>Es ist wichtig, daß dies eine Liste ist. Eine Matrix, die die Information enthält, ob zwei Teilchen kollidieren oder nicht, müßte wieder in  $N^2$  Schritten ausgewertet werden, würde den Vorteil des Verfahrens also zunichte machen.

$t$	$t + 1$	Änderungen in $K_t^B$
$b_m b_n \Rightarrow b_n b_m$		keine Änderung
$e_m e_n \Rightarrow e_n e_m$		keine Änderung
$b_m e_n \Rightarrow e_n b_m$		Kollision aus $K_t^B$ entfernen
$e_m b_n \Rightarrow b_n e_m$		Kollision zu $K_t^B$ hinzufügen

**Tabelle 2.1:** Diese Tabelle zeigt die Regeln, nach denen die Liste der Kollision  $K_t^B$  bei einer Vertauschung zweier Werte mit Insertion-Sort angepaßt werden muß. Werden zum Beispiel, zwei “Anfänge” vertauscht, so bleibt die Liste unverändert.

3.  $b_n \leq b_m \leq e_m \leq e_n$  oder

4.  $b_n \leq b_m \leq e_n \leq e_m$ ,

dann überschneiden sich die Bounding-Boxen. Nun wird angenommen, die Teilchen haben sich bewegt, d.h. die Bounding-Boxen haben sich verschoben. In Bild. 2.13 ist diese Situation dargestellt. Die Werte für  $b_1, e_1, \dots$  haben sich alle geändert, aber ihre Reihenfolge ist fast unverändert. Dies ist eine Folge der kurzen Zeitschritte bei einer DEM-Simulation, die ja eine Kollision genau nachverfolgen muß.

Wie oben erwähnt, enthält die Sortierreihenfolge der Teilchen die vollständige Information über die Bounding-Box-Kontakte. Es wurde nun ein Schema entwickelt, mit dem die Neusortierung der alten Liste mit den Änderungen in der Liste der Kollisionen verknüpft werden kann.

Dieses Verfahren beruht wieder auf dem bereits erwähnten Sortierverfahren Insertion-Sort. Dieses Verfahren, obwohl im allgemeinen eines der ineffizienteren Verfahren, hat seine Stärken, wenn es eine Liste sortieren soll, die schon fast perfekt sortiert ist. Es werden dann nur noch lokale Korrekturen durchgeführt. Der Zeitbedarf dafür liegt in der Größenordnung von  $N$ . Jeder dieser Schritte bedeutet das Vertauschen von zwei Werten. Dies bedeutet, daß sich nur der Überlappstatus eines einzigen Paares ändern kann. Die Regeln für einen Austausch finden sich in Tabelle 2.1. Es kann bei Insertion-Sort vorkommen, daß ein Wert mehrfach mit seinen Nachbarn vertauscht wird. Dann muß für jeden einzelnen dieser Schritte die Liste korrigiert werden. Das Verfahren bleibt dabei korrekt, lediglich die Effizienz sinkt. Dies passiert allerdings in realen Simulationen eher selten.

Ein Schritt für diesen Algorithmus an Hand des Beispiels in Abbildung 2.13 findet sich in Tabelle 2.2. Der Zeitbedarf ist im allgemeinen Fall proportional der Zahl der beteiligten Bounding-Boxen.

Die Erweiterung dieses Verfahrens auf den zweidimensionalen Fall ist offensichtlich. Die Projektion jeder Bounding-Box auf die Koordinatenachsen ergibt Grenzen  $b_x, b_y, e_x, e_y$ . Zwei Bounding-Boxen berühren sich dann und nur dann, wenn sich die Intervalle auf beiden Achsen berühren.

Wiederum kann eine Kollision nur verschwinden oder neu auftauchen, wenn zwei Grenzen auf einer der Koordinatenachsen vertauscht werden. Müssen nun zwei Grenzen vertauscht werden, spielt der Zustand der Intervalle auf der anderen Koordinatenachse eine entscheidende Rolle. Wenn diese korrespondierenden Intervalle sich nicht überlappen, kann keine Kollision entstehen. Nur wenn dies nicht der Fall ist, kann der Austausch zweier Grenzen überhaupt einen Einfluß auf die Liste  $K_t^B$  haben. Die genauen Regeln sind in Tabelle 2.3 dargestellt.

Wenn man zwei “Anfänge” oder “Enden” vertauscht, bleibt alles beim Alten; der Zustand der Intervalle auf der anderen Koordinatenachse ist unwichtig. Wenn man  $b_1 e_2 \Rightarrow e_2 b_1$  ausführt, kann man eine eventuell vorhandene Kollision einfach entfernen. Auch hier ist der Zustand auf der anderen Achse nicht von Bedeutung.

Schritt	sortierte Liste	akt. Element	unsortierte Liste	Vertauschungen	Anderungen in $K_t^B$
1	<u><math>b_1</math></u>	$b_2$	$e_1 e_2 b_3 e_3$	$b_1 \leftrightarrow b_2$	keine
2	$b_2$ <u><math>b_1</math></u>	$e_1$	$e_2 b_3 e_3$	-	-
3	$b_2 b_1$ <u><math>e_1</math></u>	$e_2$	$b_3 e_3$	-	-
4	$b_2 b_1$ <u><math>e_1 e_2</math></u>	$b_3$	$e_3$	$e_2 \leftrightarrow b_3$	(2;3) hinzufügen
5	$b_2 b_1 e_1$ <u><math>b_3 e_2</math></u>	$e_3$		-	-
	$b_2 b_1 e_1 b_3 e_2 e_3$				

**Tabelle 2.2:** Diese Tabelle zeigt einen vollständigen Lauf für das Beispiel in Abbildung 2.13. Das Konzept von Insertion-Sort ist einfach: es gibt eine sortierte und eine unsortierte Liste. Ein Element der unsortierten Liste wird ausgewählt und mit den Elementen der sortierten Liste verglichen (Elemente, die verglichen werden, sind unterstrichen dargestellt). Die Vergleiche beginnen mit dem rechten Wert; ist das aktuelle Element größer, dann ist der korrekte Platz schon gefunden. Ist es kleiner, dann werden die Position der beiden Werte vertauscht und die Liste der Kollisionen gemäß Tabelle angepasst. Dies wird wiederholt, bis das aktuelle Element seine korrekte Position gefunden hat.

$t$	$t + 1$	kein Überlapp auf der anderen Koordinatenachse	Überlapp auf der anderen Koordinatenachse
$b_m b_n \Rightarrow b_n b_m$		keine Änderung	keine Änderung
$e_m e_n \Rightarrow e_n e_m$		keine Änderung	keine Änderung
$b_m e_n \Rightarrow e_n b_m$		keine Änderung	Kollision entfernen
$e_m b_n \Rightarrow b_n e_m$		keine Änderung	Kollision hinzufügen

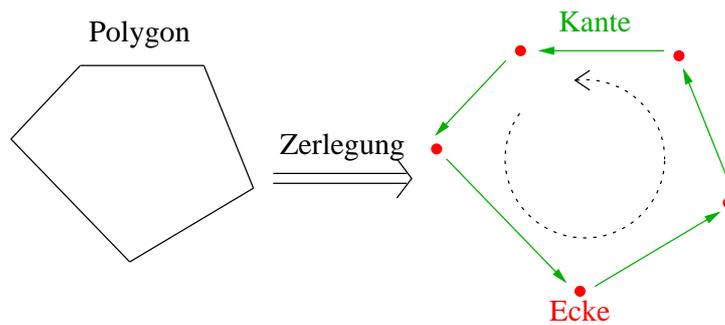
**Tabelle 2.3:** Diese Tabelle zeigt die Regeln, nach denen die Liste der Kollision  $K_t^B$  bei einer Vertauschung zweier Werte mit Insertion-Sort im zweidimensionalen Fall angepasst werden muß. Werden zum Beispiel, zwei "Anfänge" vertauscht, so bleibt die Liste unverändert, egal ob die Intervalle auf der anderen Koordinatenachse überlappen oder nicht.

Nur im letzteren Fall muß man die Intervalle der anderen Achse testen. Gibt es dort einen Überlapp, hat man eine neue Kollision gefunden und muß diese in  $K_t^B$  aufnehmen.

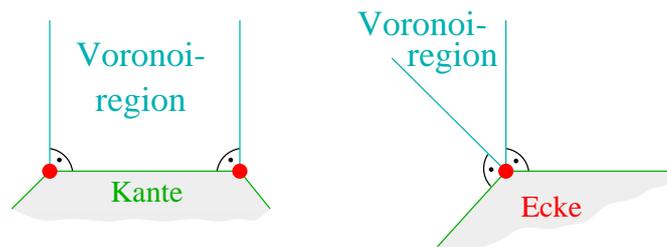
Der Zeitbedarf ist absolut gesehen doppelt so groß wie im eindimensionalen Fall, bleibt aber weiterhin proportional zur Zahl der verwendeten Teilchen. Eine Diskussion eines worst-case Szenarios findet sich in [136].

### 2.4.3 Bestimmung der Partikelabstände

Der *Closest-Feature Algorithmus*, der hier kurz vorgestellt wird, berechnet den Abstand zweier Polygone in konstanter Zeit. Dabei wird versucht, diejenigen Elemente der konvexen Polygone über die Zeit hinweg zu verfolgen, die sich am nächsten liegen. Mit Hilfe lokaler Tests kann dies bewerkstelligt werden. Es sind also nur wenige Ecken und Kanten zu berücksichtigen. Die ganze Komplexität der Partikel ist nicht von Bedeutung; es ist egal, wie viele Kanten die beteiligten Polygone haben. Die Methode wurde ursprünglich von Lin [137] für Anwendungen in der virtuellen Realität und in der Bewegungsplanung bei Robotern eingesetzt. Der Algorithmus wurde angepaßt und optimiert für



**Abbildung 2.14:** Die Zerlegung eines Polygons in einen Ring aus Ecken und Kanten. Kanten haben dabei eine Orientierung; blickt man entlang dieser Richtung, liegt das Polygon links der Kante.



**Abbildung 2.15:** Voronoiregion einer Kante und einer Ecke

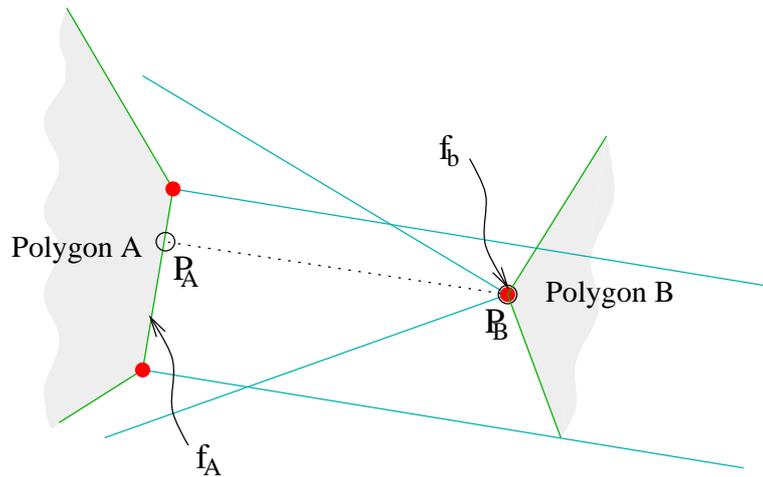
die Simulation zweidimensionaler Granulate.

Von grundlegender Wichtigkeit ist die Frage, wie das Polygon dargestellt werden soll. Hier gibt es verschiedene Möglichkeiten: man kann eine einfache Liste von Eckpunkten verwenden oder eine Liste von Strecken, die die Kanten beschreibt. Für das hier zu behandelnde Problem ist es am sinnvollsten, das Teilchen durch seine Bestandteile (Features) zu beschreiben. Dies sind Ecken und Kanten. Diese Elemente werden ihrer Position gemäß miteinander verknüpft, so daß man einen Ring aus sich abwechselnden Ecken und Kanten erhält (Abbildung 2.14). Eine Ecke darf aber nicht mit einem Punkt gleichgesetzt werden, ebensowenig eine Kante mit einer Strecke. Obwohl natürlich die Koordinaten eines Punktes Bestandteil einer Ecke sind, so ist diese doch eine weitaus komplexere Struktur, die mehr Information enthält. Gleiches gilt entsprechend für eine Kante; diese hat eine Orientierung. Es ist festgelegt, daß bei einer Kante auf der Seite “links” ist, auf der das Polygon liegt.

Das Konzept der “Voronoregionen” ist für den Algorithmus von Bedeutung. Das Voronoidiagramm [135] ist eine Zerlegung der Ebene in Bereiche, wobei jeder eine Menge von Punkten ist, die näher am Punkt  $p_i$  als an den Punkten  $p_{j \neq i}$  liegen. Man kann also mit einem Voronoidiagramm die Frage “Welcher Punkt  $p_i$  liegt am nächsten zu einem beliebigen Punkt  $q$ ?” beantworten.

Erweitert man dieses Konzept von Punkten  $p_i$  auf Ecken und Kanten eines Polygons, so kann man dann die Frage “Welches Polygonbestandteil  $f$  liegt am nächsten zu einem beliebigen (außerhalb des Polygons liegenden) Punkt  $q$ ?” beantworten. Die einem Polygonbestandteil zugehörige Voronoiregion ist dann die Menge aller Punkte, die näher zu diesem Element liegen als zu jedem anderen. Die Region wird begrenzt durch das jeweilige Element und zwei Halbgeraden (siehe Abbildung 2.15).

Für den Closest-Feature Algorithmus gehört also zu den einzelnen Bestandteilen des Polygons auch die entsprechende Voronoi-Region. Jeder Polygonbestandteil enthält damit die Koordinaten von einem (Ecke) oder zwei (Kante) Punkten, Verbindungen zu den Vorgängern und Nachfolgern innerhalb des Rings und die Beschreibung der Grenzen der Voronoiregion. Eine graphische Darstellung dieser



**Abbildung 2.16:** Die Punkte  $P_A$  und  $P_B$  mit dem kürzesten Abstand der getesteten Polygonbestandteile (Ecke und Kante) liegen jeweils in der Voronoiregion des anderen Elements. Der Abstand dieser Punkte ist also der Abstand der Polygone.

Datenstrukturen findet sich im Anhang E. Mit Hilfe der Voronoiregionen wird im folgenden dann der Abstand zweier Polygone bestimmt.

Gegeben seien die beiden Polygone  $A$  und  $B$ . Der Abstand von  $A$  und  $B$  ist die kürzeste Distanz

$$d_{AB} = \inf_{P_A \in A, P_B \in B} |P_A - P_B|$$

Dabei sucht man vor allem das Paar an Polygonbestandteilen  $f_A, f_B$ , auf dem die Punkte liegen, die den kürzesten Abstand bestimmen. Dies sind diejenigen, für die  $P_A \in f_A$  und  $P_B \in f_B$  gilt. Sie geben dem Algorithmus (closest features) seinen Namen [138–140].

Dies führt zu folgender Aussage:

Gegeben seien zwei beliebige Polygonbestandteile  $f_A$  und  $f_B$  der Polygone  $A$  und  $B$ . Die Punkte  $P_A$  und  $P_B$  bestimmen die Distanz  $d_{AB}$  von  $f_A$  und  $f_B$ . Wenn  $P_A$  innerhalb der Voronoiregion von  $f_B$  liegt und  $P_B$  innerhalb der Voronoiregion von  $f_A$ , dann ist  $d_{AB}$  der Abstand der beiden Polygone (zur Veranschaulichung siehe Abbildung 2.16).

Ob also  $f_A$  und  $f_B$  mit den zugehörigen Voronoiregionen  $V_A$  and  $V_B$  den kürzesten Abstand der Polygone festlegen, kann mit zwei Schritten festgestellt werden. Zuerst sucht man die Punkte, die den Abstand der zwei Bestandteile festlegen. Für alle drei Möglichkeiten<sup>10</sup> (Ecke-Ecke, Kante-Kante, Ecke-Kante) sind dies einfache geometrische Rechnungen. Anschließend prüft man die Gültigkeit der obigen Aussage. Ist sie zutreffend, hat man den Abstand der beiden Polygone bestimmt und hat dabei nur Information über die beiden Bestandteile  $f_A$  und  $f_b$  benutzt. Dieser Test ist also unabhängig von der Komplexität der beteiligten Polygone.

Die Prüfung, ob zwei Polygonelemente den Abstand festlegen, ist aber nur ein Teil der Arbeit. Die Frage, wie man diese Elemente findet, ist genauso wichtig. Was passiert, wenn die Überprüfung, ob ein Punkt  $P_A$  in der Voronoiregion  $V_B$  liegt, ergibt, daß der Punkt außerhalb ist? Ist der Punkt außerhalb des Polygons<sup>11</sup>, so muß nach der Definition der Voronoiregionen das benachbarte Poly-

<sup>10</sup>Hier muß man einige Spezialfälle [118, 139] berücksichtigen, z.B. parallele Kanten oder eine Ecke liegt auf einer Kante.

<sup>11</sup>Liegt der Punkt links der Kante und gleichzeitig innerhalb der nach links verlängerten Grenzen der Voronoiregionen, so ist dies der “worst-case”; während einer Simulation kommt dieser Fall außer bei der Initialisierung praktisch nicht vor.

gonelement  $f'_B$  näher an  $P_A$  liegen als  $f_B$ . Dann wird  $f_B$  durch  $f'_B$  ersetzt und die Überprüfung wiederholt. Ein Beweis für die Konvergenz dieses Verfahrens in drei Dimensionen findet sich in [137]. Der Beweis ist auch für zwei Dimensionen gültig, wenn man zylindrische Polyeder konstruiert.

Der gesamte Algorithmus, um den Abstand zweier Polygone zu bestimmen ist also:

Angenommen, man hat die Bestandteile  $f_A$  and  $f_B$  auf den Polygonen  $A$  and  $B$

1. Bestimme die Punkte  $P_A$  and  $P_B$  mit dem kürzesten Abstand der Elemente  $f_A$  and  $f_B$ .
2. Berechne die Grenzen der zu  $f_A$  gehörigen Voronoiregion  $V_A$ .
3. Prüfe, ob  $P_B$  im Inneren von  $V_A$  liegt. Wenn nicht, ersetze  $f_A$  durch das zur verletzen Grenze gehörige  $f'_A$  und springe zu 1.
4. Berechne die Grenzen der zu  $f_B$  gehörigen Voronoiregion  $V_B$ .
5. Prüfe, ob  $P_A$  im Inneren von  $V_B$  liegt. Wenn nicht, ersetze  $f_B$  durch das zur verletzen Grenze gehörige  $f'_B$  und springe zu 1.
6. Berechne den Abstand  $d$  von  $P_A$  und  $P_B$
7. Wenn  $d = 0$ , dann schneiden sich  $f_A$  and  $f_B$ . Der erste Schnittpunkt für den Überlapp der Polygone ist gefunden.

Ein Beispiel für einen möglichen Ablauf des Algorithmus ist in Abbildung. 2.17 dargestellt.

Der Zeitbedarf dieses Algorithmus ist proportional zu  $\mathcal{O}(n \log n)$  wobei  $n$  die Zahl der Polygonbestandteile ist. An diesem Punkt wird nun die Tatsache wichtig, daß ein physikalisches System simuliert werden soll. Um die Differentialgleichung 2.2 zu lösen, ist es notwendig, daß sich die Partikel innerhalb eines Zeitschrittes nur sehr wenig bewegen und drehen. Damit ist die Annahme gerechtfertigt, daß, wenn zwei Elemente  $f_A$  and  $f_B$  zum Zeitpunkt  $t$  den kürzesten Abstand haben, sie dieses auch sehr wahrscheinlich auch zum Zeitpunkt  $t + \Delta t$  tun. Trifft dies nicht zu, sind  $f_A$  and  $f_B$  immer noch sehr gute Anfangswerte für den Algorithmus; meistens muß man den Algorithmus nur ein zweites mal durchlaufen. In [118] wird ein Beispiel mit 2 Polyedern mit jeweils 2762 Elementen gezeigt. Bei 100000 Zeitschritten muß in 2.2% der Fälle ein zweiter, in 0.1% der Fälle ein dritter Durchlauf gemacht werden. Die Laufzeit dieses Algorithmus in einer physikalischen Simulation ist also im wesentlichen konstant. Ein Beispiel für die Laufzeiten sieht man in Abbildung 2.18.

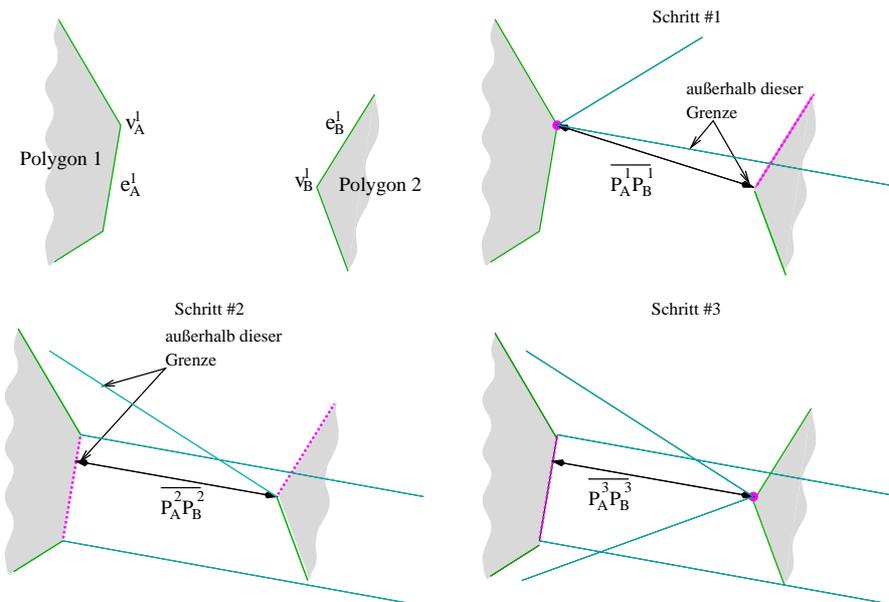
#### 2.4.4 Parallelisierung

In den Abschnitten 2.4.2 und 2.4.3 wurde gezeigt, wie man das Laufzeitverhalten der Simulation verbessern kann, so daß die Untersuchung größerer Teilchenzahlen möglich ist. Diese Methoden verfolgen das Ziel, die vorhandene Rechenleistung besser zu nutzen. Um eine weitere Leistungssteigerung zu erreichen, wird versucht, das Problem auf mehrere CPUs zu verteilen. Eine genaue Darstellung, wie die Parallelisierung erfolgt, findet sich in [133, 136], eine Einführung in das Methode bietet [141]. Hier sollen nur nochmals kurz die grundlegenden Schritte erläutert werden.

Der am häufigsten angewandte Ansatz, eine DEM-Simulation zu parallelisieren, ist es, den Raum aufzuteilen und die Teilgebiete von einzelnen Rechnern bearbeiten zu lassen [107, 108, 123, 142].

Werden dabei Informationen benötigt, die eine andere CPU vorhält, so müssen diese mit Hilfe eines geeigneten Protokolls [143–147] über das Netzwerk ausgetauscht werden<sup>12</sup>. Die Kommunikation

<sup>12</sup>Im allgemeinen bedient man sich einer Bibliothek, die geeignete Kommunikationsroutinen bereitstellt. Genannt seien hier das *Message Passing Interface (MPI)* oder die *Parallel Virtual Machine (PVM)*.



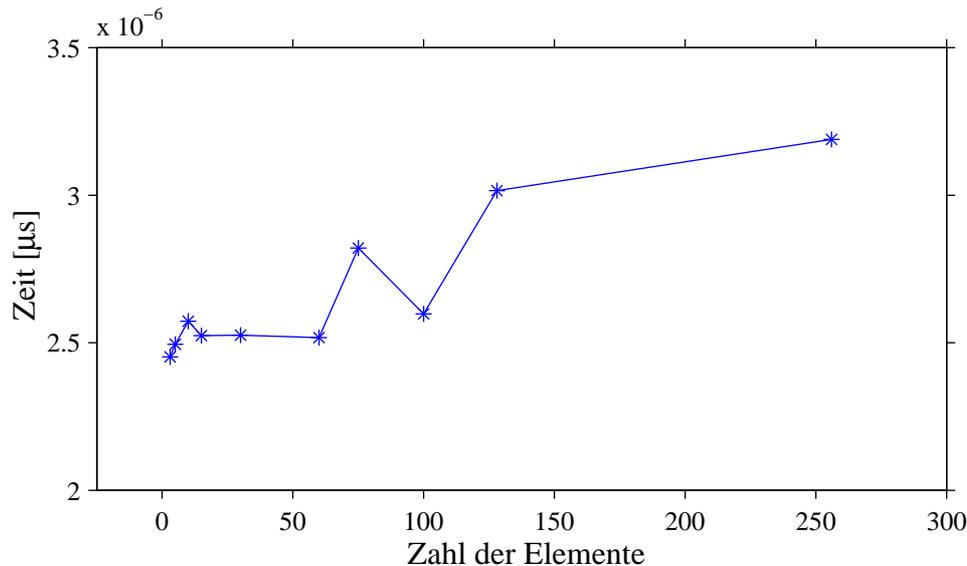
**Abbildung 2.17:** Ein Beispiel der Berechnung des kürzesten Abstands mit Hilfe des closest-feature-

Algorithmus. Zu Beginn habe man die Ecke  $v_A^1$  und die Kante  $e_B^1$ .

Die Punkte  $P_A^1$  und  $P_B^1$  werden berechnet. Dann wird geprüft, ob  $P_B^1$  innerhalb der Voronoiregion  $V_A$  liegt. Die Begrenzung zwischen  $v_A^1$  und  $e_A^2$  wird verletzt, der Durchlauf wird abgebrochen und  $v_A^1$  durch  $e_A^1$  ersetzt.

Der zweite Schritt wird mit  $e_A^1$   $e_B^1$  durchgeführt. Wieder werden die Punkte  $P_A^2$  und  $P_B^2$  berechnet.  $P_B^2$  liegt auch innerhalb von der Voronoiregion von  $e_A^1$ , aber die Begrenzung zu  $e_B^1$  wird verletzt.

Der dritte Schritt nutzt entsprechend  $e_A^1$  und  $v_B^1$ . Nach der Berechnung von  $P_A^3$  and  $P_B^3$  stellt man fest, daß beide Punkte jeweils innerhalb der entsprechenden Voronoiregion liegen,  $P_A^3$  and  $P_B^3$  geben den Abstand der Polygone vor.



**Abbildung 2.18:** Dargestellt ist die Laufzeit (auf eine 200 MHz Sun Ultra2 mit Ultrasparc Prozessoren) des Closest-Feature-Algorithmus bei verschiedenen Eckenzahlen der beteiligten Polygone. Der leichte Anstieg der Laufzeit kommt daher, daß durch das verwendete Testsetup bei höheren Eckenzahlen häufiger (Faktor 100) der Abstand erst mit der zweiten Iteration gefunden wird.

wird aber auch sehr schnell zum “bottle-neck”, es kann vorkommen, daß die Kommunikation mehr Zeit als die eigentliche Rechenarbeit benötigt [148].

Dagegen sind Parallelrechner, bei denen sich mehrere CPUs einen gemeinsamen Speicher teilen, in der Leistungsklasse der Workstations immer häufiger anzutreffen. Threads sind dort ein geeignetes Werkzeug, um ein paralleles Programm auf derartigen Rechner zu implementieren. Explizite Kommunikation ist hier nicht notwendig, alle CPUs arbeiten mit denselben Daten. Entsprechende Vorarbeit bei den verwendeten Algorithmen und Datenstrukturen vorausgesetzt, kann mit Hilfe des adäquaten Programmiermodells (*Threads*) eine einfache und effiziente Umsetzung erfolgen.

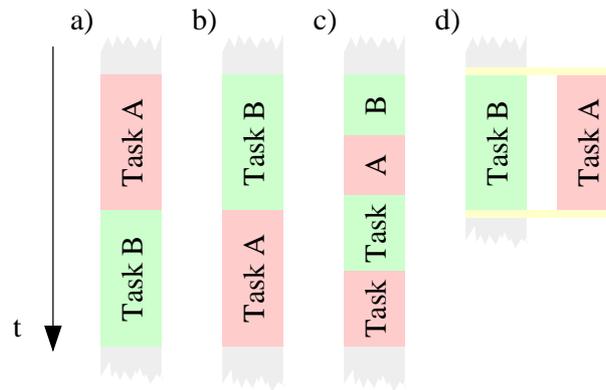
Im folgenden sollen nur die wichtigsten Grundkonzepte erläutert werden. Damit können dann die Forderungen an die eingesetzten Algorithmen formuliert werden, so daß diese sich mit Threads kombinieren lassen.

Kleiman [141] definiert einen Thread folgendermaßen: *A thread of control, or more simply a thread, is an independent sequence of execution of program code inside a UNIX process. [...] The threads share all of the process’s memory.*

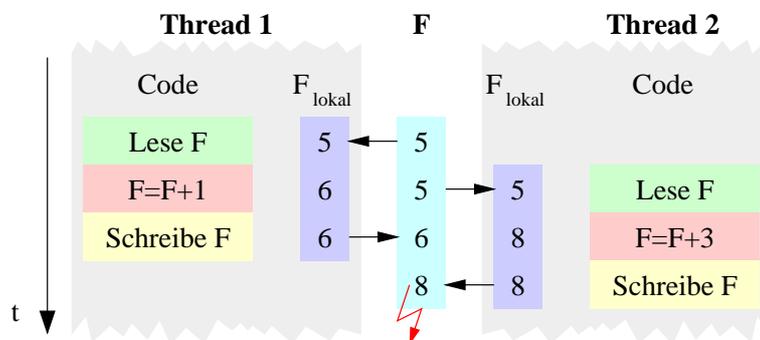
Die Grundidee dabei ist einfach: Man stelle sich ein Programm als ein langes Band von Befehlen vor. Es gibt dann eine logische Einheit, die sich an diesem Band entlangarbeitet und die Befehle ausführt. Dabei werden dann die Daten im Speicher gemäß den Programmanweisungen manipuliert<sup>13</sup>. Um nun parallel zu arbeiten, kann man mehrere dieser Einheiten gleichzeitig auf diesem Band arbeiten lassen, wobei sich die zu manipulierenden Daten im selben Speicher befinden. Diese einzelnen Pfade der Programmausführung nennt man dann Thread.

Beim Programmieren ist das Arbeiten mit Threads sehr einfach. Man ruft, vereinfacht gesagt, eine Funktion auf, wartet aber nicht auf deren Beendigung. Stattdessen wird sie, parallel zum Hauptpro-

<sup>13</sup>Dabei ist diese Einheit ein logisches Konstrukt aus einer CPU, einem Programm-Counter und einem Stack. Unter UNIX spricht man dann von einem Prozeß.



**Abbildung 2.19:** Wenn die Reihenfolge bei der Ausführung zweier Aufgaben in beliebiger Reihenfolge a), b) und im Wechsel c) erfolgen kann, so ist es möglich, sie auch gleichzeitig auszuführen.

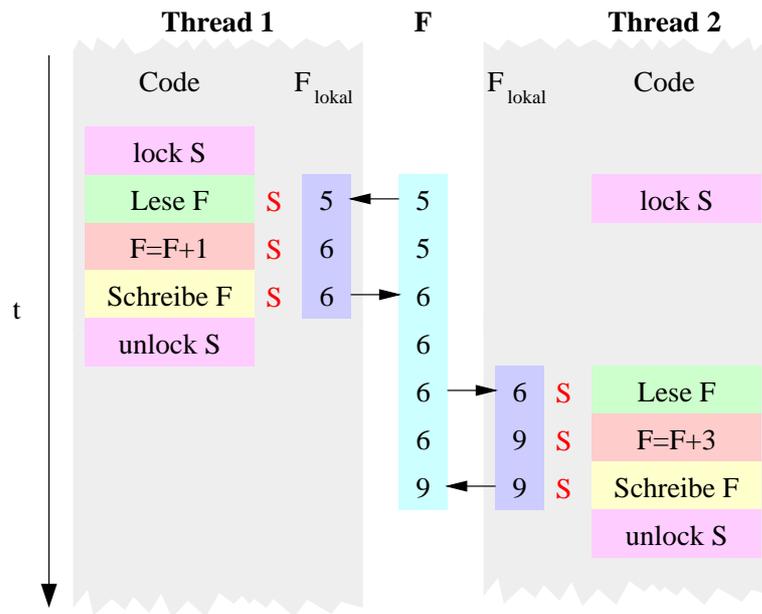


**Abbildung 2.20:** Ein Problem tritt beim gemeinsamen Zugriff auf Daten auf, da Daten nicht direkt im Speicher manipuliert werden, sondern gelesen, verarbeitet und wieder geschrieben werden.

gramm, im Hintergrund ausgeführt. Entscheidend ist beim Entwickeln, daß die einzelnen Routinen voneinander unabhängig sind. Wenn es nämlich unwichtig ist, in welcher Reihenfolge die Aufgaben erledigt werden (siehe Abbildung 2.19 a, b) und die Aufgaben sogar abwechselnd bearbeitet werden können (c), dann ist es auch möglich, sie gleichzeitig zu bearbeiten. Die Funktion selbst greift dabei auf die gleichen Daten wie das Hauptprogramm zurück und kann diese auch ändern. Dieser letzte Punkt bedarf gesonderter Erläuterungen, da bei nicht korrekter Regelung beim Zugriff auf gemeinsame Daten das Programmergebnis zufällig wird.

Das Problem taucht auf, wenn verschiedene Threads gleichzeitig denselben Speicher verändern wollen. Nehmen wir an, wir haben zwei Threads. Beide sollen auf eine Variable  $F$  einen festen Wert addieren. Wie man nun in Abbildung 2.20 sieht, ist es möglich, die Zugriffe der Threads zeitlich so zu verzahnen, daß das Ergebnis des einen Threads durch das des anderen überschrieben wird. Das Ergebnis, mit dem weitergerechnet wird, ist 8 und nicht 9.

Offensichtlich muß man dafür Sorge tragen, daß immer nur ein Thread Daten verändern darf. Hierzu verwendet man sogenannte Semaphore. Ein Semaphore ist damit eine Art Berechtigungsschein. Bevor Daten verändert werden sollen, wird diese Berechtigung angefordert, danach zurückgegeben. Fordert man ein Semaphore an, obwohl das Semaphore im Moment nicht verfügbar ist, wird der Thread solange angehalten, bis jenes wieder freigegeben wurde. In der Sprache der Threads heißen diese Zugriffe auf ein Semaphore **lock** und **unlock**, die Semaphore selbst werden als **mutex** bezeichnet.



**Abbildung 2.21:** Durch die Nutzung des Semaphors  $S$ , daß ein Thread vom Betriebssystem zugeteilt bekommt, kann gewährleistet werden, daß nur ein Thread Zugriff auf bestimmte Daten hat.

Die korrekte Ausführung des obigen Beispiels ist in Abbildung 2.21 dargestellt. Thread 1 sichert sich durch die Anforderung des Semaphors  $S$  den alleinigen Zugriff auf  $F$ . Thread 2 muß solange warten, bis Thread 1 fertig ist und den Zugriff auf  $F$  wieder freigibt.

Wieviele Semaphore man nutzt, und welche Zugriffe abgesichert werden liegt allein beim Entwickler. Ein Semaphor schützt nämlich aktiv keine bestimmte Variable, sondern ist nur ein Hilfsmittel. Ein Thread kann jederzeit einen Semaphor ignorieren und auf die zu schützende Variable zugreifen. Entsprechende Fehler sind sehr schwer zu entdecken, da sie nicht deterministisch auftreten. Sorgfältige Planung ist hier notwendig, um die Integrität von Daten zu gewährleisten. Da Zugriffe auf ein Semaphor Zeit in Anspruch nehmen, sollte man sie nur verwenden, wenn dies notwendig ist.

## 2.4.5 Anpassung an die Algorithmen

Die Hauptschleife des Programmes gliedert sich in mehrere Arbeitsschritte, wobei jeder Arbeitsschritt eines oder mehrere Ergebnisse des vorhergehenden Schritts benötigt.

In der Tabelle 2.4 sind diese einzelnen Arbeitsschritte aufgeführt. Gleichzeitig sind die Datenbereiche angegeben, auf die lesend oder schreibend zugegriffen wird. Folgende Bereiche sind relevant:

$POS(i)$	Daten des Predictor-Corrector, die Position des Teilchens, Lage der Ecken des $i$ -te Teilchens
$BB(i, XY)$	Daten der Bounding-Box, jeweils für die X- oder Y-Achse und das $i$ -te Teilchen
$KOL(i, j)$	Daten für eine (mögliche) Kollision zwischen dem $i$ -ten und $j$ -ten Teilchens $KOL(i, j) \equiv KOL(j, i)$
$F(i)$	Daten der Kraftberechnung für das $i$ -te Teilchen

Aus der Tabelle kann man dann schließen, wie die Arbeit auf Threads aufgeteilt werden kann.

Die Schritte 1-4 können für jedes Teilchen unabhängig von allen anderen durchgeführt werden. Man

	Arbeitsschritt	lesen	schreiben
1	Predictor	$POS(i)$	$POS(i)$
2	div. Vorberechnungen	$POS(i)$	$POS(i)$
3	Position der Ecken	$POS(i)$	$POS(i)$
4	Lage der BB.	$POS(i)$	$BB(i, X) BB(i, Y)$
5	Sortieren X-Achse	$BB(i, X) POS(i) POS(?)$	$BB(i, X) BB(? , X) KOL(i, j)$
6	Sortieren Y-Achse	$BB(i, Y) POS(i) POS(?)$	$BB(i, Y) BB(? , Y) KOL(i, j)$
7	Abstandsberechnung	$KOL(i, j) POS(i) POS(j)$	$KOL(i, j)$
8	Kraftberechnung	$KOL(i, j) POS(i) POS(j)$	$F(i) F(j)$
9	Corrector	$F(i)$	$POS(i)$

**Tabelle 2.4:** Alle Schritte und Zugriffe sind immer für das  $i$ -ten Teilchen oder die Kombination aus  $i$ -ten und  $j$ -ten Teilchen angegeben.

kann also die Zahl der Threads an der Zahl der verfügbaren Prozessoren orientieren und jeden davon einen Teil der Partikel bearbeiten lassen.

Da Schritt 6 keine Ergebnisse von Schritt 5 benötigt und vice versa, können diese beiden Schritte gleichzeitig abgearbeitet werden; es bietet sich an 2 Threads zu verwenden. Stehen mehr Prozessoren zur Verfügung, sollte es möglich sein, die Sortierarbeit pro Achse auch auf mehrere Threads zu verteilen, solange die "Nahtstellen" der Teillisten dann noch nachsortiert werden. Da es vorkommen kann, dass mehrere Threads nach  $KOL(i, j)$  schreiben, müssen diese Zugriffe durch Semaphore abgesichert werden.

In Schritt 7 und 8 müssen alle gefundenen Kollisionen  $KOL(i, j)$  weiterbearbeitet werden. Da die Berechnung pro Kollision nicht von den Ergebnissen einer anderen Kollision abhängt, können die Daten  $KOL(i, j)$  auf beliebige Threads verteilt werden. Nur der mögliche gleichzeitige Zugriff auf  $F(i)$  muß mit Semaphoren gesichert werden.

Im neunten Schritt ist die Situation den Schritten 1-4 sehr ähnlich. Die Berechnungen betreffen nur ein Teilchen. Die Arbeit kann wieder auf beliebig viele Threads aufgeteilt werden.

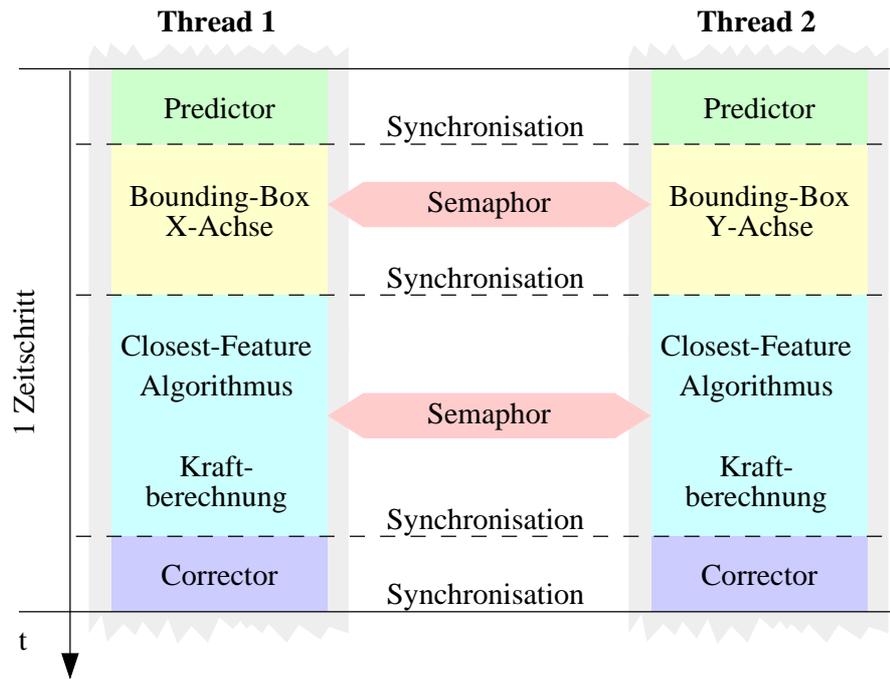
## 2.5 Test der Simulationen

Das GIGO<sup>14</sup>-Prinzip [149] bei der Entwicklung komplexer Programme macht es erforderlich, die Software intensiv zu testen. Dabei sind verschiedene Sorten von Fehlern zu unterscheiden:

- syntaktische Fehler
- semantische Fehler
- logische Fehler
- Fehler des Modells

Typische Syntaxfehler sind falsch geschriebene Befehlswoorte oder fehlende Klammern. Ein fertig kompiliertes Programm ist immer syntaktisch korrekt, wenn der Compiler fehlerfrei arbeitet. Semantische Fehler können nur teilweise von Compilern erkannt werden. Zwei Fehler sind dabei geradezu

<sup>14</sup>Garbage in - Garbage out



**Abbildung 2.22:** Die grundlegende Struktur von `gc2d` bei der Nutzung von zwei Threads. Die Kombination von Predictor-Corrector, Sortieren und closest-feature-Algorithmus ist dargestellt, ebenso die notwendigen Semaphoren und die Zeitpunkte der Synchronisation. Eine Kommunikation im Sinne des Austausches von Variablen ist nicht notwendig.

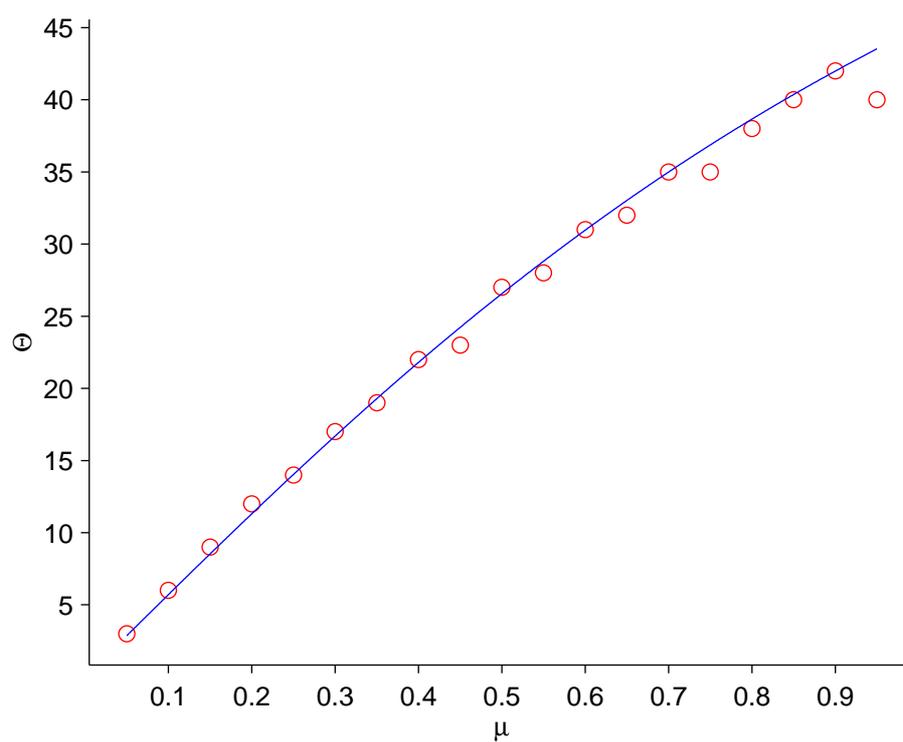
“klassisch”: die Division durch Null und ein Zugriff auf ungültige Speicherbereiche. Der erste Fehler führt zum Abbruch eines laufenden Programmes, der zweite kann unentdeckt bleiben und zu falschen Ergebnissen führen. Ein logischer Fehler zeigt sich nicht durch Fehlermeldungen, sondern produziert im allgemeinen falsche Resultate oder ein unerwartetes Verhalten. Ein typischer logische Fehler bei `gc2d` wäre ein Problem mit der Listenverwaltung oder ein Synchronisationsproblem der Threads. Durch ausführliche Tests während der gesamten Entwicklungsphase von `gc2d` wurde versucht, diese Fehler zu entdecken und zu beseitigen. Durch vielfältige Sicherheitsabfragen wurde dafür Sorge getragen, daß keine falschen Ergebnisse berechnet werden. Auf diese Tests soll nicht weiter eingegangen werden.

Viel wichtiger für eine Simulation ist, ob das zugrundeliegende Modell zuverlässig ist. Bevor in den Abschnitten 3 und 4 die Ergebnisse großer Simulationen mit den Experimenten verglichen werden, sollen hier noch einige einfache Simulationen gezeigt werden, deren Ergebnis analytisch bekannt ist.

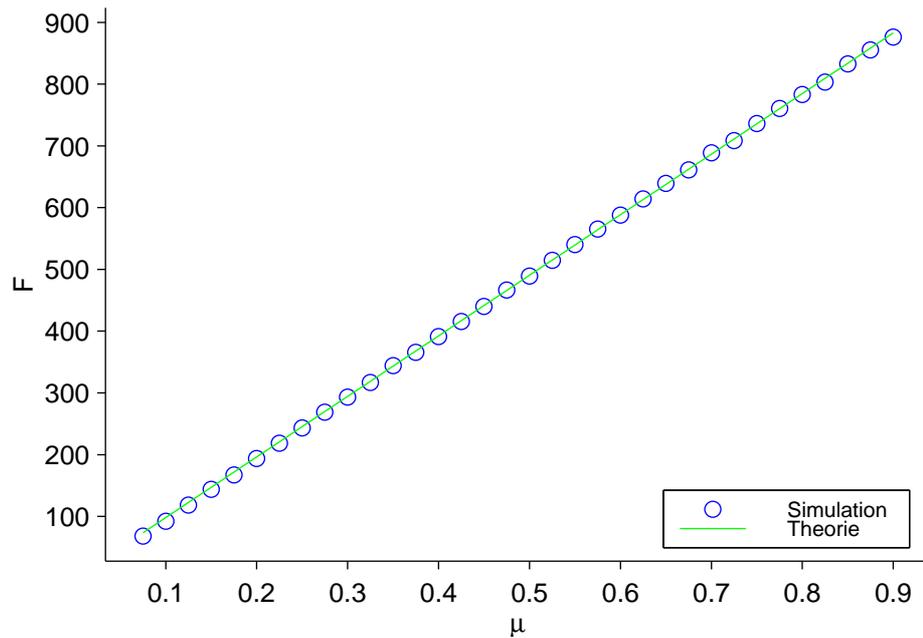
### 2.5.1 Reibung

Hier wird das Verhalten eines Blockes auf einer schiefen Ebene untersucht. Wenn die schiefe Ebene den Winkel  $\Theta$  mit der Waagerechten einschließt, wirkt auf den Körper die Hangabtriebskraft  $F_h = mg \sin \Theta$  und die Normalkraft  $F_n = mg \cos \Theta$ . Der Körper fängt dann an zu rutschen, wenn die Reibung  $F_r = \mu F_n$  kleiner ist als  $F_h$ . Es wurden in der Simulation ein flacher Block auf verschiedene geneigte Ebenen gesetzt und ermittelt, ab welchem Winkel er zu rutschen beginnt. Es muß dabei

$$\mu = \tan \Theta \quad (2.53)$$



**Abbildung 2.23:** Die durchgezogene Linie ist  $\tan \Theta$ , die Punkte geben die Neigung der Ebene an, ab der in der Simulation der Block zu gleiten beginnt. Dabei wurden die Ebenen in  $1^\circ$ -Schritten geneigt.



**Abbildung 2.24:** Dargestellt ist die notwendige Kraft, um einen ruhenden Block der Masse  $100g$  zu beschleunigen.

gelten. In Abbildung 2.23 erkennt man, daß die Simulation sehr gut mit der Theorie übereinstimmt.

Ein weiterer Test für die Reibungsgesetze ist, auf einen Block, der auf einem glatten Untergrund ruht, eine immer größer werdende Kraft wirken zu lassen. Solange diese Kraft kleiner ist als  $\mu \cdot F_n$  darf der Block sich nicht bewegen; dies gilt unabhängig von der Auflagefläche. Dies wurde simuliert, wobei für jeden Wert des Reibungskoeffizienten 18 verschiedene Teilchenformen gleicher Masse  $m = 100g$  simuliert wurden.

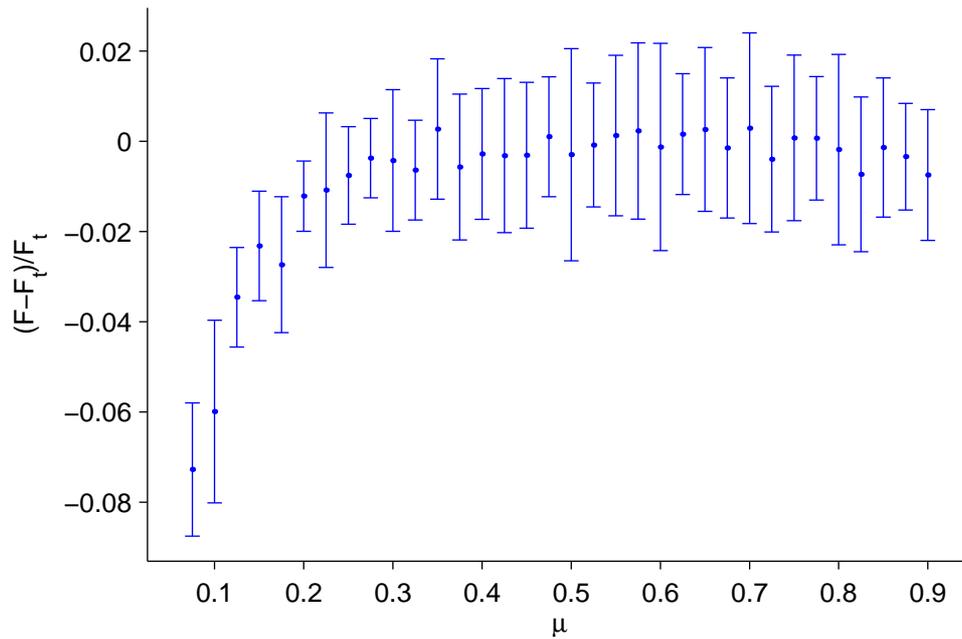
In Abbildung 2.24 ist die Kraft aufgetragen, die notwendig war, um den Block zu beschleunigen. Der lineare Anstieg wird sauber reproduziert. In Abbildung 2.25 ist die prozentuale Abweichung der tatsächlichen Kraft vom theoretischen Wert dargestellt. Bei kleinen Werten von  $\mu$  ist der Fehler am größten. Dies hängt aber nicht mit der Modellierung selbst zusammen, sondern vielmehr damit, daß es schwierig ist, den exakten Zeitpunkt des Bewegungsbeginns zu bestimmen.

Betrachtet man aber die Abhängigkeit von der Auflagefläche (Abbildung 2.26), stellt man eine leichte Abhängigkeit der benötigten Kraft von der Auflage fest. Dies liegt nicht an den Kraftgesetzen; diese machen keinen Unterschied bei der Auflagefläche. Beim theoretischen Modell wird angenommen, daß der Körper weder rotiert, noch sich senkrecht zur Oberfläche bewegt. Tatsächlich übt die an der Vorderkante des Teilchens befestigte Feder aber immer auch ein kleines Drehmoment aus. Dieser Effekt ist bei den Partikeln mit der kleinsten Auflagefläche, also den hohen Teilchen, besonders groß.

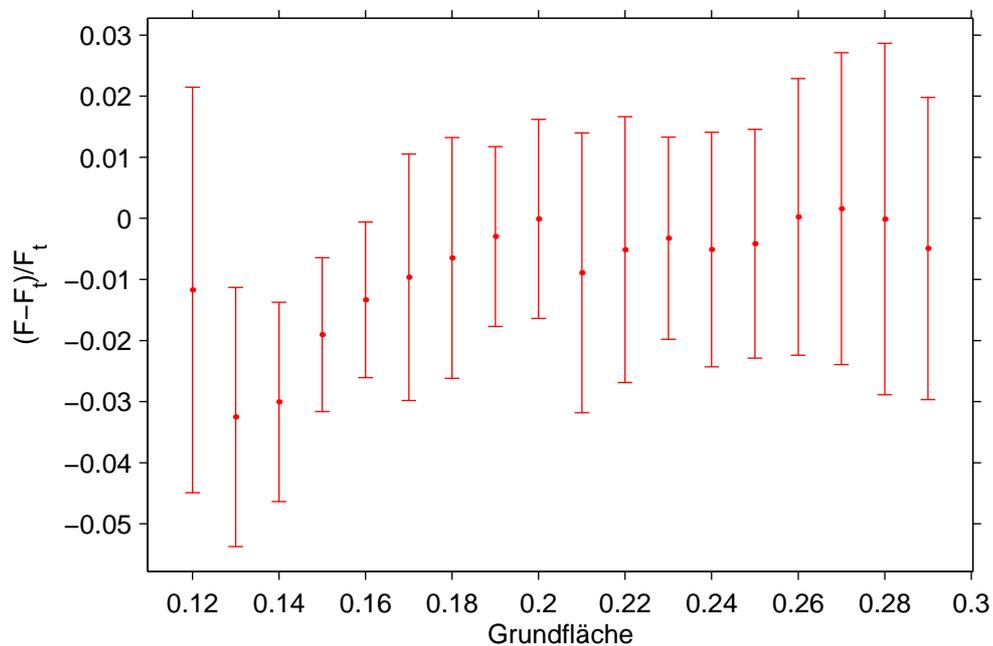
## 2.5.2 Stoß von zwei und mehr Teilchen

Die einfachste Art, die Kollision von Partikeln zu untersuchen, ist es, Stoßexperimente durchzuführen. Diese Untersuchungen können auch mit `gC2d` durchgeführt werden.

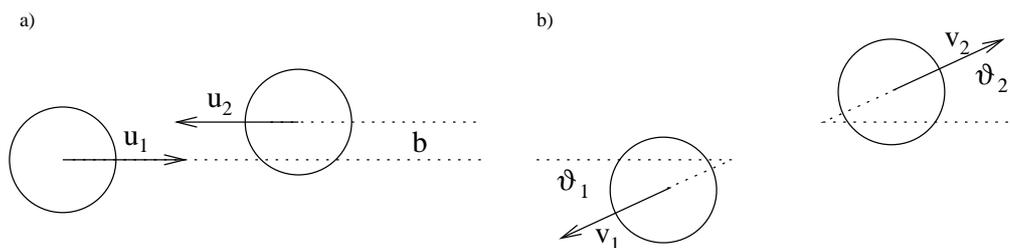
Da der elastische Stoß untersucht werden soll, ist die Dämpfungskonstante  $\gamma = 0$ . Für die Reibung



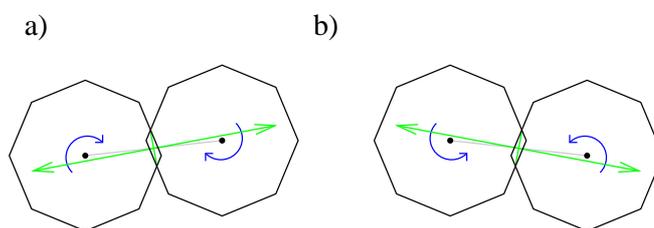
**Abbildung 2.25:** Die relative Abweichung vom theoretischen Wert ist im Bereich sehr kleiner Reibung am größten. Dies liegt aber an der Schwierigkeit, den Beginn der Bewegung festzustellen. Gemittelt wurde jeweils über verschiedene Auflageflächen des Blocks, die Fehlerbalken entsprechen der Standardabweichung.



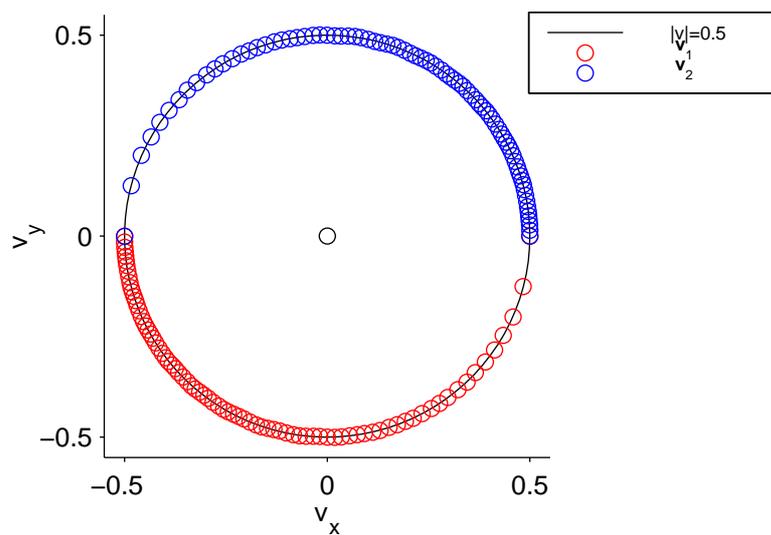
**Abbildung 2.26:** Hier wurde die Abweichung der zur Beschleunigung notwendigen Kraft für verschiedene Auflageflächen bestimmt. Gemittelt wurde dabei über Werte von  $\mu$  im Bereich 0.05 bis 0.95, die Fehlerbalken entsprechen der Standardabweichung.



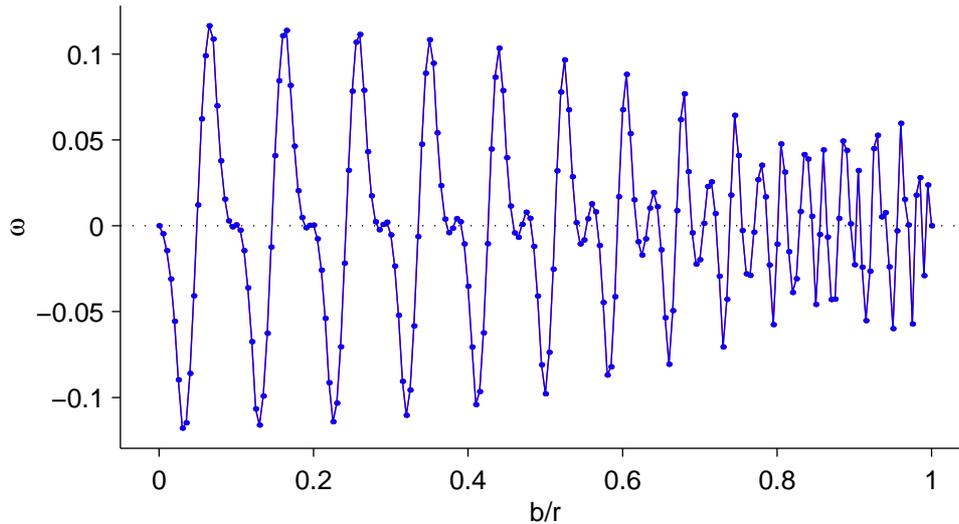
**Abbildung 2.27:** Zwei Scheiben a) vor und b) nach dem Stoß



**Abbildung 2.28:** Je nach Lage der Ecken können verschiedene Drehmomente beim Stoß entstehen. Grün eingezeichnet sind die Kräfte und die Strecke  $\overline{c_1 c_2}$  (siehe Abbildung 2.4)



**Abbildung 2.29:** Geschwindigkeitsvektoren nach dem Stoß.



**Abbildung 2.30:** Winkelgeschwindigkeit zweier 64-Ecke nach dem Stoß in Abhängigkeit des Stoßparameters  $\frac{b}{r}$ . Aufgrund der Symmetrie der Ausgangssituation können die Partikel nur gleichen Drehsinn haben.

gilt ebenso  $\mu = 0$ ,  $G$  ist ebenfalls 0. Die beiden Vielecke haben gleiche Maße und zu Beginn die Geschwindigkeiten  $u_1 = \begin{pmatrix} 0.5 \\ 0 \end{pmatrix}$  und  $u_2 = \begin{pmatrix} -0.5 \\ 0 \end{pmatrix}$ . Nach dem Stoß werden die Geschwindigkeiten  $v_1$  und  $v_2$  bestimmt (siehe Abbildung 2.27).

In Abbildung 2.29 sind für verschiedene Werte von  $\frac{b}{r}$  die Geschwindigkeitsvektoren  $v_1$  (rot) und  $v_2$  (blau) aufgetragen. Entsprechend der Energie- und Impulserhaltung liegen die Endpunkte der Vektoren auf einem Kreis mit dem Radius 0.5. Der Streuwinkel ist gegeben [150] durch

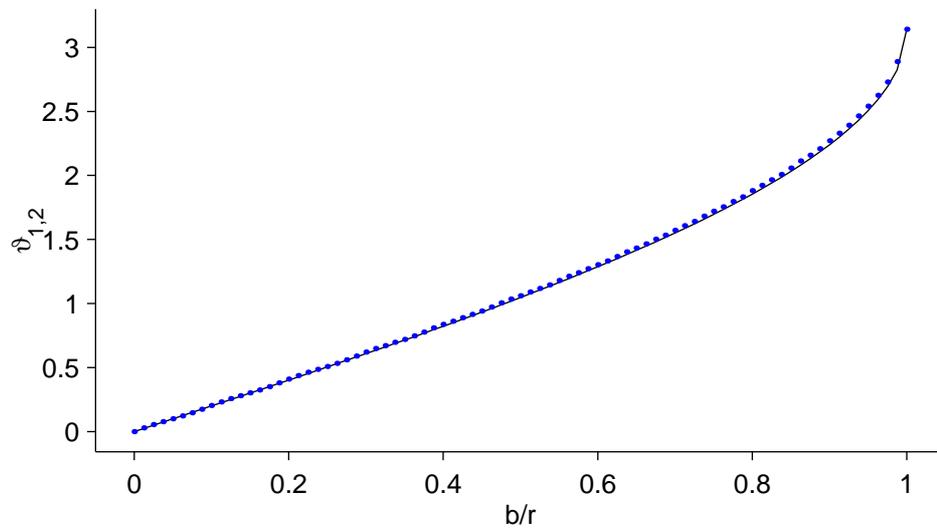
$$\vartheta = 2 \arcsin \frac{b}{r}. \quad (2.54)$$

In Abbildung 2.31 wird dieser Wert mit der Simulation des 64-Ecks verglichen. Die Übereinstimmung von Theorie und Simulation ist auch für den Fall, daß sich die Teilchen nur noch streifen, sehr gut.

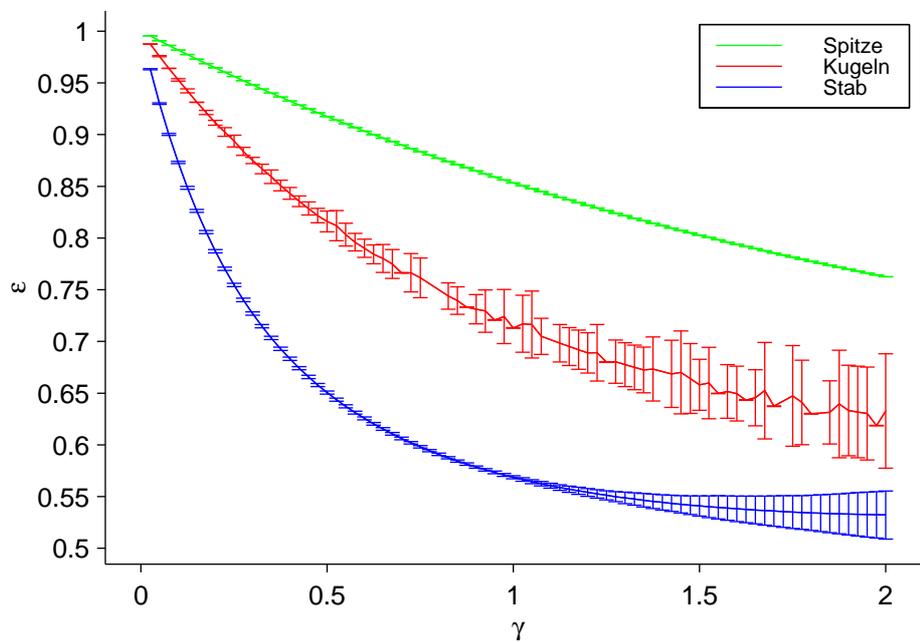
In Abbildung 2.30 ist die Winkelgeschwindigkeit der Partikel nach dem Stoß dargestellt. Wenn die Reibung gleich Null ist, haben Scheiben keine Möglichkeit, bei einem Stoß das Rotieren zu beginnen. Ganz anders ist dies bei Polygonen. Hier können aufgrund der Tatsache, daß die Teilchen nicht glatt sind, sondern Ecken und damit eine gewisse Rauigkeit haben, beim Stoß auch Drehmomente entstehen (siehe Abbildung 2.28). Dies bedeutet, daß durch die Polygonalgestalt der Teilchen eine Oberflächenrauigkeit repräsentiert werden kann, die einen deutlichen Einfluß auf die Simulation hat.

Für den Fall der inelastischen Kollision kann der Energieverlust bei einer Kollision bestimmt werden. Dieser wird durch den Restitutionskoeffizienten bestimmt. Wie in Abbildung 2.32 zu erkennen ist, hängt der Energieverlust nicht nur von der Dämpfung ab, sondern auch von der Geometrie der Kontakte.

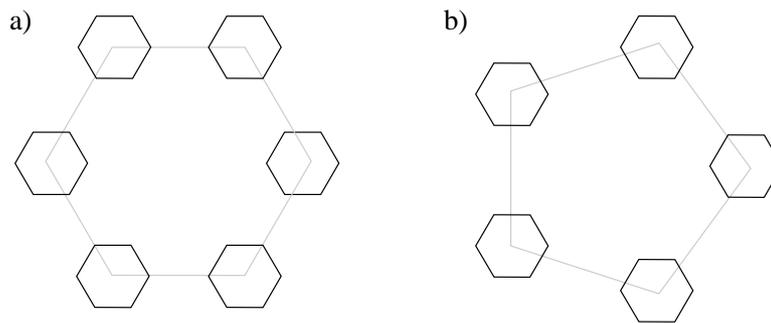
In der nächsten Simulation wurde überprüft, wie sich das System bei der gleichzeitigen Kollision von mehreren Teilchen verhält. Dazu werden  $N$  Teilchen auf die Ecken eines regelmäßigen  $N$ -Ecks gesetzt. Der Betrag der Geschwindigkeiten aller Teilchen ist gleich, die Richtung ist zum gemeinsamen



**Abbildung 2.31:** Der Streuwinkel  $\vartheta$  in Abhängigkeit vom Stoßparameter  $\frac{b}{r}$ . Die theoretische Verlauf  $\vartheta = 2 \arcsin \frac{b}{r}$  ist als Linie eingezeichnet.



**Abbildung 2.32:** Die Abhängigkeit des Restitutionskoeffizienten  $\epsilon$  von der Dämpfung und der Form der Partikel. Gemittelt wurde über verschiedene Stoßparameter  $b$ , die Fehlerbalken entsprechend der Standardabweichung.



**Abbildung 2.33:** a) Ist die Teilchenzahl ein Teiler der Eckenzahl, so sind die Kontakte alle gleich. Im Fall b) trifft dies nicht zu und die Ausgangssituation ist nicht mehr rotationssymmetrisch

Schwerpunkt hin. Theoretisch sollten die Teilchen gleichzeitig als Ring kollidieren und dann wieder sternförmig mit gleicher Geschwindigkeit auseinanderfliegen. Dies ist weniger ein Test für die verwendeten Kraftgesetze. Diese können das Verhalten auf jeden Fall beschreiben. Vielmehr könnte die Numerik oder der Differentialgleichungslöser zu Fehlern führen, wenn für ein Teilchen die Stöße der beiden Nachbarn nicht gleich behandelt werden und dieser Fehler aufsummiert wird.

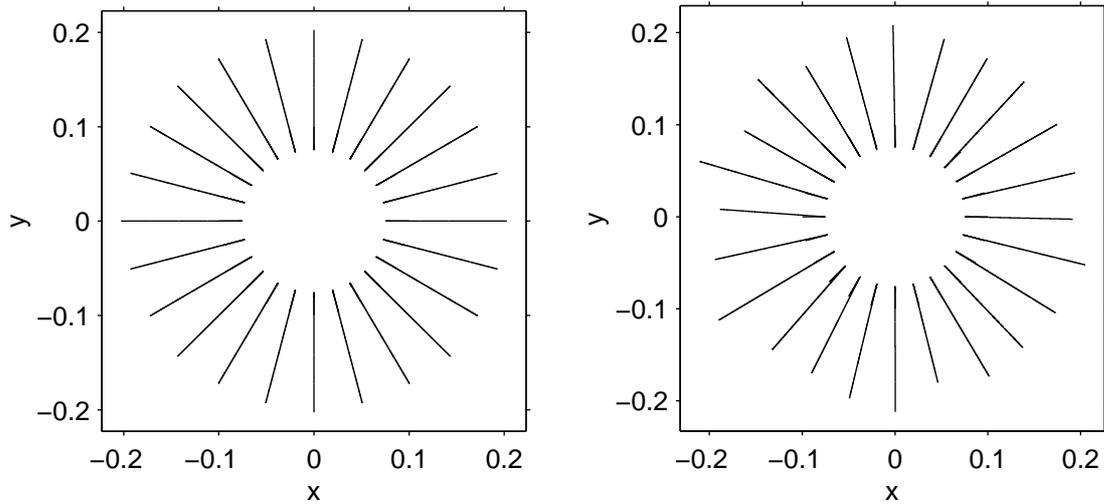
Die Trajektorien für den Fall von 24 Teilchen sind in Abbildung 2.34(a), dargestellt. Die Abweichung der Geschwindigkeiten nach dem Stoß sieht man in Abbildung 2.35(a), die Abweichung vom Winkel in 2.35(b). Gemittelt wurde jeweils über alle Teilchen. Die Fehler sind bei der Geschwindigkeit selbst im schlechtesten Fall unter 1%, schwanken aber beträchtlich. Insbesondere für 2,4,8,16 Ecken sind die Fehlerbalken so klein, daß sie fast nicht mehr dargestellt werden können. Hier stellt sich heraus, daß wieder die Teilchengestalt der Grund für dieses Verhalten ist. Die verwendeten Teilchen waren 64-Ecke, die besonders sauberen Programmläufe hatten immer Eckenzahlen, die Teiler von 64 sind. In Abbildung 2.33 sieht man das Problem. Da die die Partikel so erzeugt wurden, daß eine Symmetrieachse parallel zur X-Achse war, sind die einzelnen Kontakte nicht gleich wenn Eckenzahl und Teilchenzahl teilerfremd sind. Damit finden die Kollisionen sind nicht mehr gleichzeitig statt und das simulierte System weicht in einem entscheidenden Punkt von der Theorie ab, die annimmt, daß alle Kollisionen gleichzeitig stattfinden und identisch sind. Dies ist aber kein Problem der Kraftgesetze, sondern eine Folge des Übergangs von runden zu eckigen Teilchen.

Wie nun ist der Einfluß der Form im Vergleich zu möglichen Schwankungen der Geschwindigkeit, wenn man das Experiment [151] in der Realität durchführen würde? Dazu wurden noch zwei weitere Fälle simuliert. Einmal schwankte die Anfangsgeschwindigkeit um 0.05%, beim zweiten Lauf um 0.5%.

Für den ersten Lauf sind die Daten in Abbildung 2.36 dargestellt. Die Fehler sind deutlich größer und es gibt auch keine Sonderstellung der Kollisionen von 4, 8,16 Teilchen.

Für den zweiten Lauf sind in Abbildung 2.34(b) die Trajektorien für 24 Teilchen dargestellt. Es ist zu erkennen, daß die (scheinbare) Symmetrie der Anfangsbedingung nicht mehr vorhanden ist. Die Fehler bei der Geschwindigkeit und bei den Winkeln ist um eine Zehnerpotenz größer und liegt in der Größenordnung der Startgeschwindigkeit. Ein vergleichbares Experiment müßte mit extremer Sorgfalt die Anfangsbedingungen realisieren, wenn der Einfluß der Teilchenform nachgewiesen werden soll.

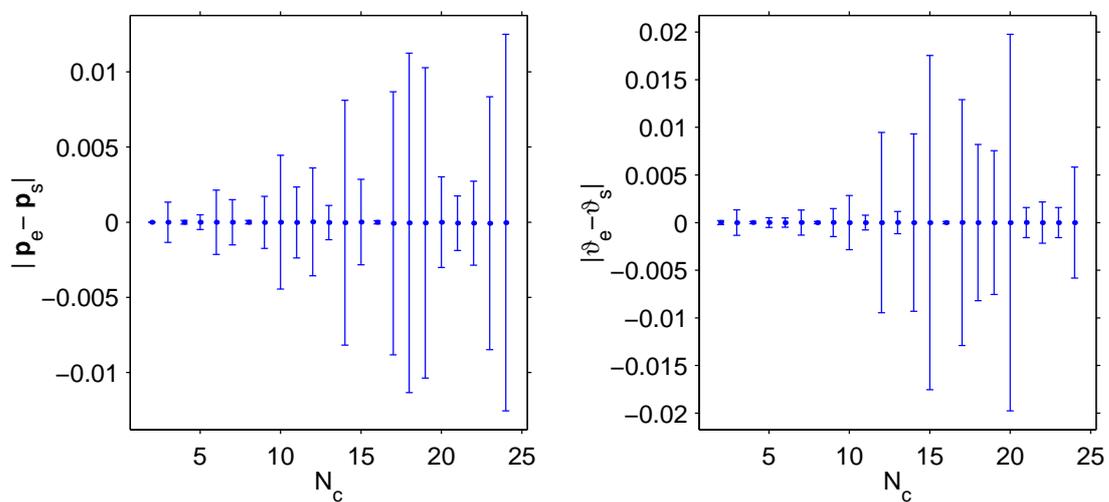
Zusammenfassend kann man sagen, daß die Kraftgesetze die verschiedenen Stöße gut reproduzieren. Die Abweichungen aufgrund der Polygondarstellung der Teilchen sind nachvollziehbar und vernünftig.



(a) Gleichzeitige Kollision

(b) Zeitversetzte Kollisionen

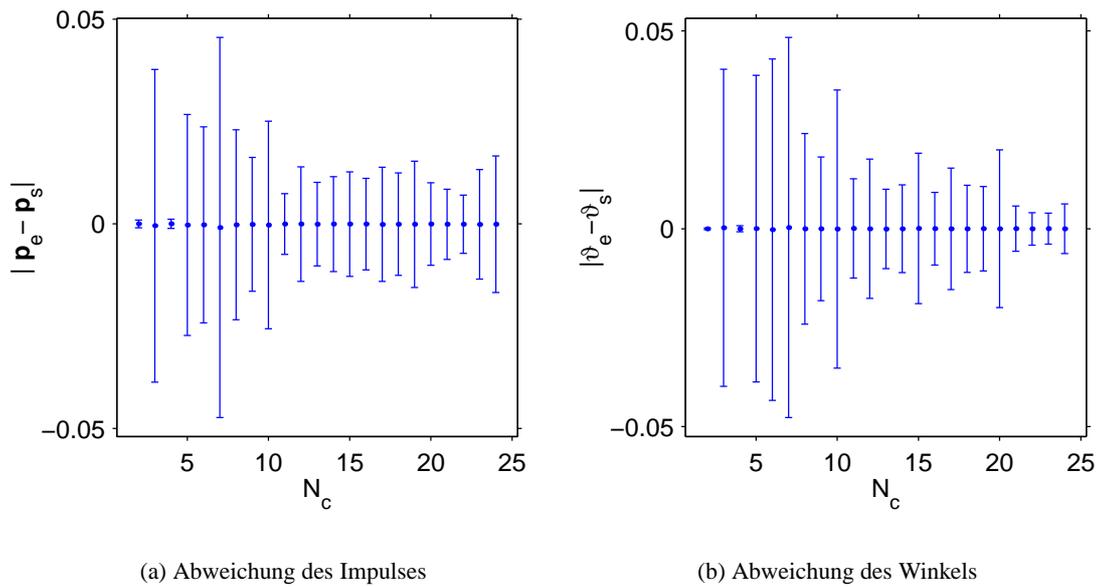
**Abbildung 2.34:** Trajektorien von 24 Teilchen, die auf einem Kreis mit dem Radius 0.1 in Richtung Ursprung fliegen. Der innere Kreis ist der Umkehrpunkt bei der Kollision, der äußerste Punkt jeder Trajektorie gibt die Position am Ende der Simulation an.



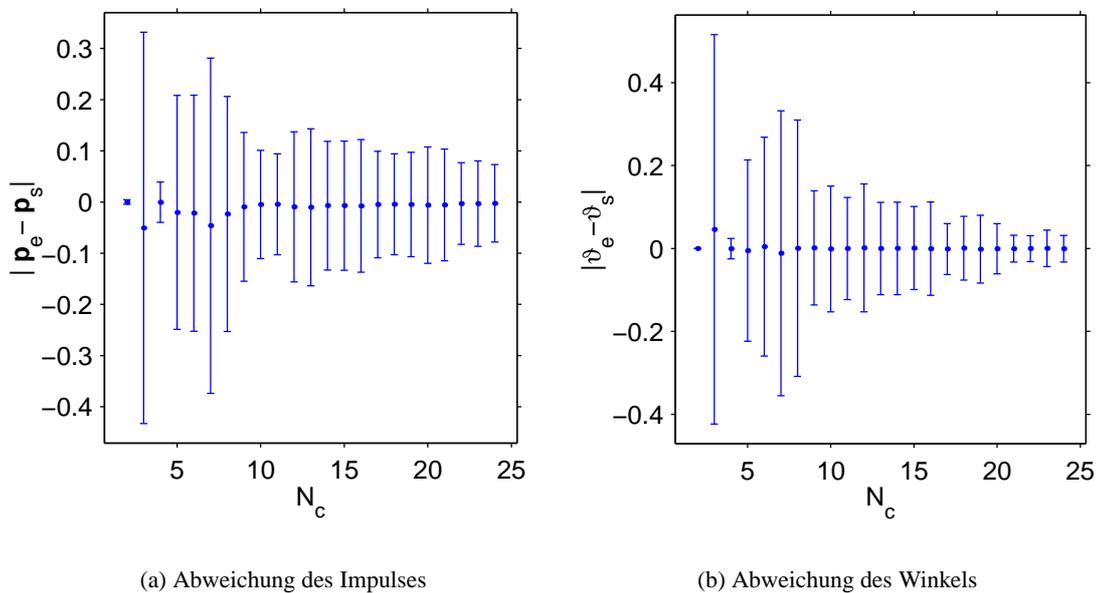
(a) Abweichung des Impulses

(b) Abweichung des Winkels

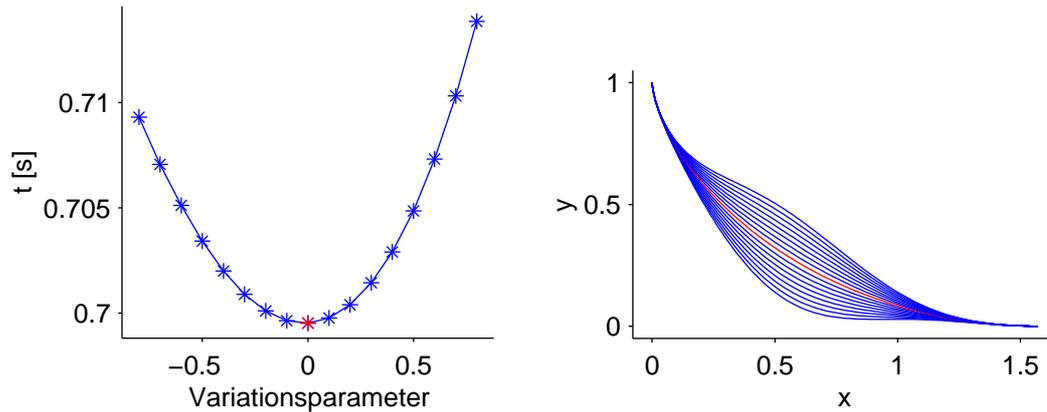
**Abbildung 2.35:** Abweichungen bei der gleichzeitigen Kollision von Kugeln mit gleicher Anfangsgeschwindigkeit. Gemittelt wurde über alle Partikel, die Fehlerbalken entsprechen der Standardabweichung.



**Abbildung 2.36:** Abweichungen bei der Kollision von Kugeln mit zufällig um 0.05% variierenden Anfangsgeschwindigkeiten. Gemittelt wurde über alle Partikel, die Fehlerbalken entsprechen der Standardabweichung.



**Abbildung 2.37:** Abweichungen bei der Kollision von Kugeln mit zufällig um 0.5% variierenden Anfangsgeschwindigkeiten. Gemittelt wurde über alle Partikel, die Fehlerbalken entsprechen der Standardabweichung.



(a) Dargestellt ist die Zeit, die der Gleiter bis zum Ende der Bahn benötigt.

(b) Form der Bahn

**Abbildung 2.38:** Rot eingezeichnet ist die Simulation mit  $\epsilon = 0$ , also der Brachystochrone.

### 2.5.3 Brachystochrone

“Die Gerade ist die kürzeste Verbindung zwischen zwei Punkten”, aber nicht immer die schnellste, muß man hinzufügen. Eine Rutsche verbinde zwei Punkte  $(x_1, y_1)$  und  $(x_2, y_2)$ , es gilt  $y_0 = y_2 - y_1 > 0$  und  $x_0 = x_2 - x_1 > 0$  und die Gravitation ist parallel zur Y-Achse. Die Frage ist, welche Form die Rutsche haben muß, damit ein auf ihr reibungsfrei rutschen der Punkt möglichst schnell von einem Punkt zum anderen gelangt. Dieses Problem geht zurück auf die Gebr. Bernoulli (1696). Die Lösungskurve heißt Brachystochrone. Die allgemeine Lösung [152] ist eine Zyklode mit der Parameterdarstellung

$$x = \frac{c^2}{4g}(\Theta - \sin \Theta) \quad , \quad y = y_0 - \frac{c^2}{4g}(1 - \cos \Theta) \quad , \quad (2.55)$$

wobei sich das  $\Theta$  der schnellsten Bahn aus

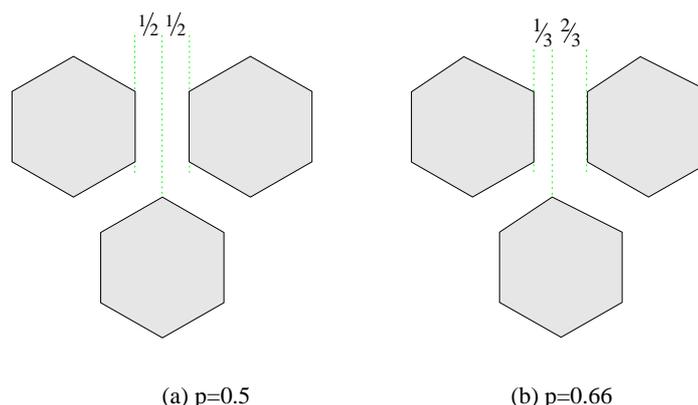
$$\frac{x_0}{y_0} = \frac{\Theta - \sin \Theta}{1 - \cos \Theta} \quad (2.56)$$

bestimmt. Für die Simulation [153] wurde  $\Theta = \pi$  gewählt und damit gilt  $\frac{x_0}{y_0} = \frac{\pi}{2}$ . Die Rutschbahn wurde aus vielen Wänden zusammengesetzt, sie ist also eigentlich ein Polygonzug. Das rutschende Partikel hat 64 Ecken, ist also hinreichend rund. Der Wert für die Reibung  $\mu$  wurde auf Null gesetzt.

Die Form (siehe Abbildung 2.38(b)) der Rutschbahn wurde durch

$$x = \frac{c^2}{4g}(\Theta - \sin \Theta) \quad , \quad y = (y_0 - \frac{c^2}{4g}(1 - \cos \Theta)) + (1 + \epsilon \sin^2(2x)) \quad (2.57)$$

vorgegeben.  $\epsilon = 0$  ist die ungestörte Lösung,  $\epsilon > 0$  überhöht den Mittelteil der Bahn und  $\epsilon < 0$  senkt ihn ab. In Abbildung 2.38(a) ist dann die Form der Bahnen und die Zeit, die das Teilchen bis zum Ziel braucht, dargestellt.



**Abbildung 2.39:** Detailansicht der “Nägel” des Galtonbretts. Durch die Verformung der oberen Spitze wird eine Asymmetrie der Verteilung erreicht.

### 2.5.4 Galton–Brett

Die vorhergehenden Tests haben immer das Verhalten der Simulation bei einzelnen Kollisionen eines oder mehrerer Teilchen betrachtet. Sehr viel schwieriger ist es, ein System zu finden, das zwar möglichst zufällige Kollisionen umfaßt, dessen genauer Endzustand aber leicht zugänglich ist. Gleichzeitig soll die Simulation auch anschaulich sein und die Korrektheit der Simulation soll mit dem bloßen Auge erkennbar sein.

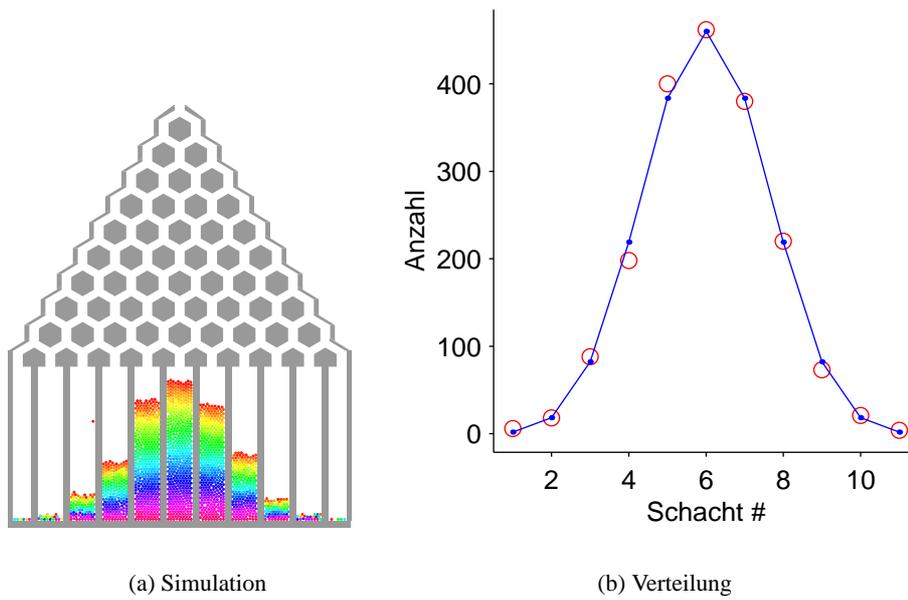
Ein schönes System, das all diese Eigenschaften hat, ist das Galton-Brett. Eigentlich dient das Galton-Brett der Veranschaulichung der Binomialverteilung. Die Binomialverteilung beschreibt ein Experiment mit genau zwei möglichen Ergebnissen  $A$  und (*nicht*  $A$ ), welche die Wahrscheinlichkeit  $p$  und  $(1 - p)$  haben. Wird dieses Experiment  $n$ -mal unter gleichen Bedingungen durchgeführt, dann wird die Wahrscheinlichkeit, daß das Ergebnis  $A$   $k$ -mal vorkommt, durch die Binomialverteilung:

$$b_{n,p}(k) = \binom{n}{k} p^k (1 - p)^{n-k} \quad (2.58)$$

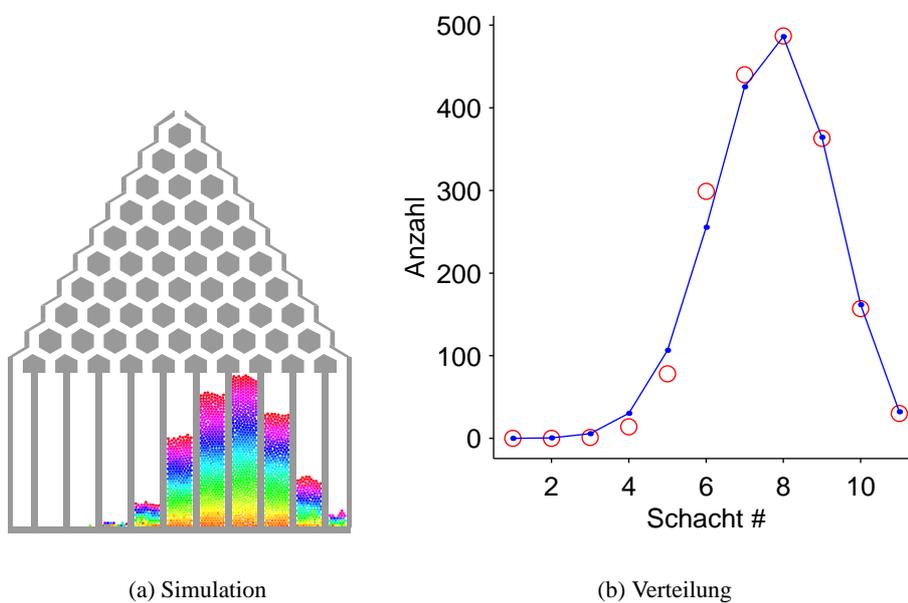
beschrieben. In der Simulation gibt es 10 “Nagelreihen”, also ist  $n = 10$ . Indem man einen Nagel nicht unterhalb der Mitte zweier Nägel positioniert, sondern versetzt dazu, kann  $p$  variiert werden. In der Simulation wurde dazu die Spitze der Sechsecke verschoben, so daß die Kugeln nicht mehr gleichmäßig nach links und recht abgelenkt werden, sondern es eine bevorzugte Seite gibt. Simuliert wurde  $p = \frac{1}{2}$  (Abbildung 2.39(a)) und  $p = \frac{2}{3}$  (Abbildung 2.39(b)). Jede Simulation wurde mit ca. 2300 Kugeln durchgeführt, die Endsituation sieht man in Abbildung 2.40(a) und 2.41(a). Dann wurden die Kugeln in jedem Schacht ausgezählt und aufgetragen, gleichzeitig wurde mit Gleichung 2.58 die theoretische Verteilung berechnet. Dargestellt sind die Ergebnisse in Abbildung 2.40(b) und 2.41(b), die Werte stimmen gut überein, auch die Asymmetrie der Verteilung bei  $p = \frac{2}{3}$  ist vorhanden. Mit `gc2d` kann man also ein Galtonbrett realitätsnah simulieren.

### 2.5.5 Maxwellscher Dämon

Es gehört zu den Standardaufgaben einer Vorlesung über Statistik, daß man ausrechnet, wie wahrscheinlich es ist, daß sich in einem Würfel, der eine Trennwand mit einem kleinen Loch enthält,  $n\%$  mehr Gasmoleküle in der einen als in der anderen Hälfte befinden. Typische Ergebnisse werden



**Abbildung 2.40:** Simulation des Galton-Brettes mit  $p = \frac{1}{2}$ . a) Der Endzustand der Simulation b) Zahl der Kugeln in einem Schacht (blau) und der theoretische Wert (rot).

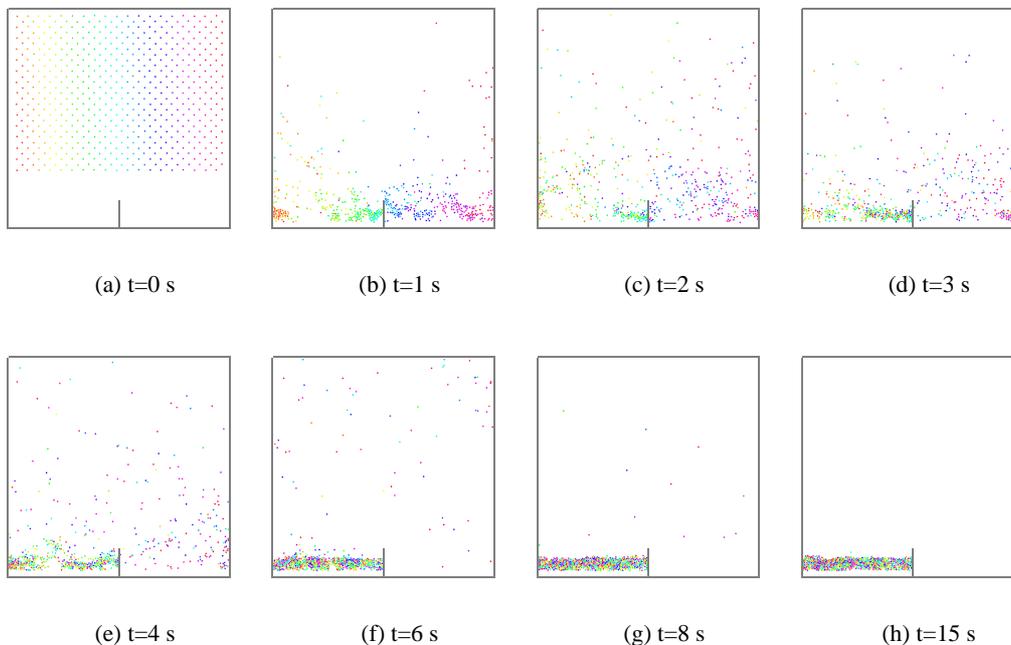


**Abbildung 2.41:** Simulation des Galton-Brettes mit  $p = \frac{2}{3}$ . a) Der Endzustand der Simulation b) Zahl der Kugeln in einem Schacht (blau) und der theoretische Wert (rot).

dann immer mit dem Alter des Universums verglichen. In diesem Zusammenhang wird dann auch der Maxwellsche Dämon vorgestellt. Macht man das gleiche Experiment stattdessen mit Granulaten, so findet man Erstaunliches; man kann dem Maxwellschen Dämon bei der Arbeit zusehen.

Einem Granulat wird in einem Kasten durch Vibration laufend Energie zugeführt, um den Energieverlust durch Kollisionen auszugleichen. Am Boden des Systems gibt es eine kleine Trennwand. Bei entsprechender Amplitude und Frequenz<sup>15</sup> der Vibration wird binnen kurzer Zeit die Symmetrie gebrochen und das Granulat sammelt sich auf einer Seite der Trennwand an. Eggers hat dieses Phänomen genauer untersucht [154]. Hier sei nur ein einzelner Lauf von `gc2d` beispielhaft vorgestellt.

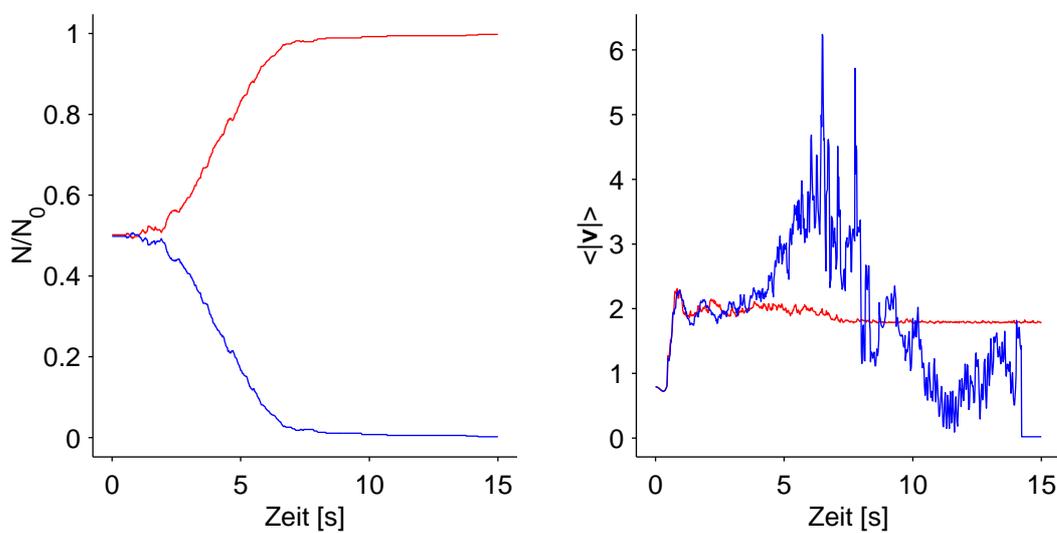
Anschaulich wird das Verhalten des Systems, wenn man die zeitliche Entwicklung (siehe Abbildung 2.42) untersucht<sup>16</sup>. Ein Cluster von Teilchen kann bei der Kollision mit einem einzelnen Teilchen dessen Energie sehr leicht dissipieren. Sobald sich also in einer Hälfte des Behälters ein hinreichend großer Cluster gebildet hat, wird dieser weitere Teilchen einsammeln, aber durch die Vibration des Behälters nicht zerstört. Gleichzeitig wird in der anderen Hälfte des Systems das granulare Gas verdünnt, die mittlere Geschwindigkeit steigt an. Im Endzustand haben sich alle Partikel auf einer Seite in einer Schicht angesammelt, die durch die Vibration nicht mehr angeregt wird.



**Abbildung 2.42:** Zeitliche Entwicklung von Partikeln in einem vibrierenden Kasten mit einer Trennwand.

<sup>15</sup>Aus der Amplitude und der Frequenz der Vibration kann die maximale Beschleunigung  $\Gamma$  der Teilchen abgeleitet werden. Es wurden keine systematischen Studien für verschiedene Werte  $\Gamma$  durchgeführt. Vielmehr zeigten mit den gewählten Parametern für die Frequenz  $f \approx 50 - 150\text{Hz}$  und der Amplitude von wenigen Teilchendurchmessern alle Simulationen das beschriebene Phänomen.

<sup>16</sup>Auf der beigelegten CDROM befindet sich der Film `gas.mpeg`, der einen kompletten Simulationslauf zeigt.



(a) Zeitliche Entwicklung der Teilchenzahl in der linken (rot) und rechten (blau) Seite des Kastens für die Simulation in Abbildung 2.42.

(b) Zeitliche Entwicklung der mittleren Geschwindigkeit in der linken (rot) und rechten (blau) Seite des Kastens für die Simulation in Abbildung 2.42.

**Abbildung 2.43:** Zeitliche Entwicklungen in einer vibrierenden Box mit einer Trennwand.

## Kapitel 3

# Druckverteilung in Schüttgutsäulen

### 3.1 Grundlagen

Im Gebiet der mechanischen Verfahrenstechnik gibt es einen interessanten Effekt, der sehr anschaulich das nichtintuitive Verhalten von Granulaten belegt. Dabei wird granulares Material in eine Röhre gefüllt und der Druck auf Untergrund bzw. auf die Seitenwände bestimmt. Eine Skizze dieser Anordnung ist in Abbildung 3.1 gezeigt.

Würde sich der Sand wie eine Flüssigkeit verhalten, wäre zu erwarten, daß der Druck linear von  $z$  abhängt  $P_{hydro}(z) \propto z$ . Im Gegensatz dazu zeigt sich allerdings, daß bei Granulaten der Druck ab einer gewissen Tiefe nicht mehr weiter zunimmt, sondern einen Sättigungswert<sup>1</sup> erreicht.

Zur Modellierung gibt es einen bekannten Ansatz aus dem vorletzten Jahrhundert von Janssen [155], der dieses Verhalten mit einem Kontinuumsmodell beschreibt. Da dieses Modell im folgenden erweitert werden soll, wird hier der klassische Lösungsweg und die entsprechenden Annahmen für den zweidimensionalen Fall skizziert.

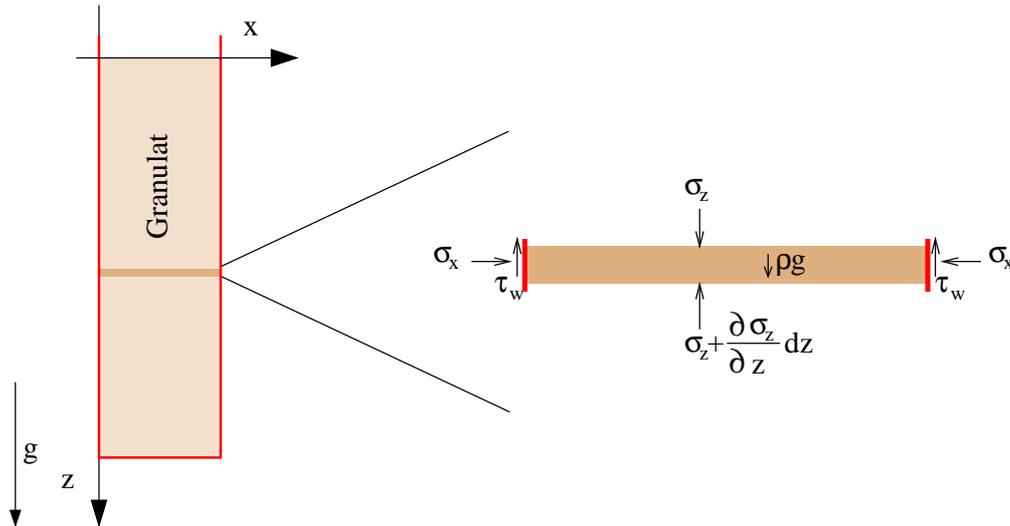
Hier soll das Problem in zwei Dimensionen behandelt werden. Nach Janssen betrachtet man eine dünne Scheibe mit der Dicke  $dz$  und der Breite  $2r$ . Für die Vertikalspannung  $\sigma_{zz}$  wird angenommen, daß sie über die ganze Breite der Scheibe konstant ist. Weiterhin wird angenommen, daß  $\sigma_{xz} = \sigma_{zx} = 0$  ist. Damit sind  $\sigma_{xx}$  und  $\sigma_{zz}$  die Hauptspannungen und die durch die Reibung an der Wand verursachte Spannung  $\tau_w$  verhält sich wie eine Volumenkraftdichte [123]. Die Gleichgewichtsbedingung lautet dann

$$\frac{d\sigma_{zz}}{dz} + \frac{\tau_w}{r} - \rho g = 0 \quad . \quad (3.1)$$

Mit der Coulombschen Reibung  $\tau_w = \mu_j \sigma_{xx}$  und der Annahme, daß  $\sigma_{xx}$  und  $\sigma_{zz}$  in einem festen Verhältnis

$$k = \frac{\sigma_{xx}}{\sigma_{zz}} \quad (3.2)$$

<sup>1</sup>Dies ist der Grund, warum Silowände eine konstante Dicke aufweisen, während Staudämme nach unten hin immer breiter werden.



**Abbildung 3.1:** Das Koordinatensystem für die Berechnung des Spannungstensors in einer Schüttgutsäule

stehen, ergibt sich für das zweidimensionale System die Differentialgleichung

$$\frac{d\sigma_{zz}}{dz} + \sigma_{zz} \frac{\mu_j k}{r} - \rho g = 0 \quad , \quad (3.3)$$

deren Lösung mit der Anfangsbedingung  $\sigma_{zz}(0) = 0$

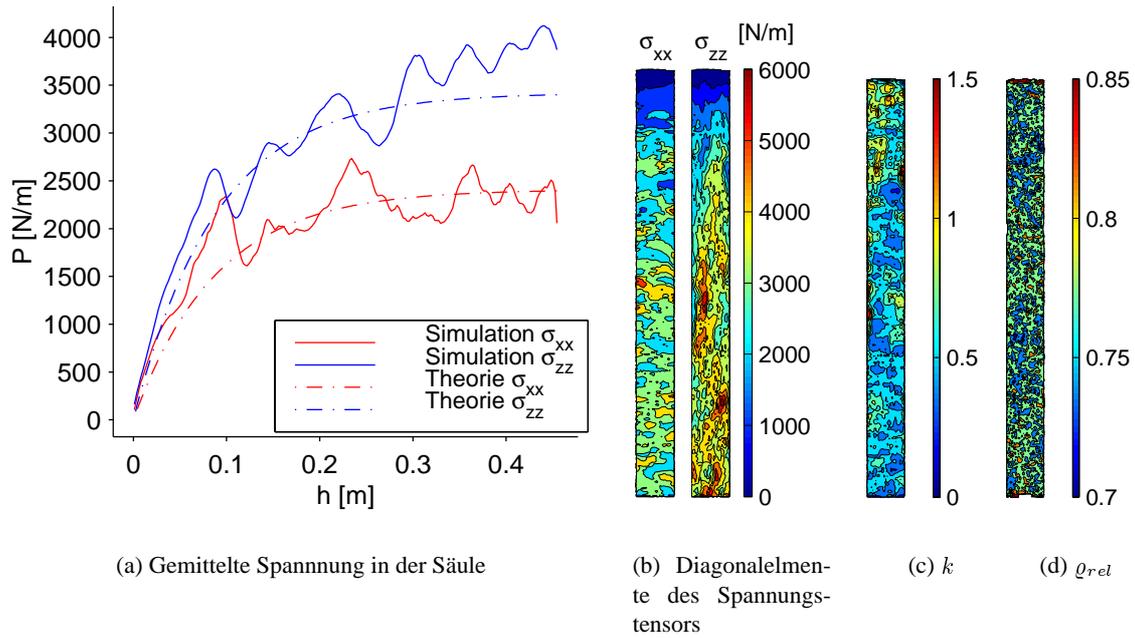
$$\sigma_{zz} = \frac{\rho g r}{k \mu_j} \left( 1 - e^{-\frac{k \mu_j}{r} z} \right) \quad \text{bzw.} \quad \sigma_{xx} = \frac{\rho g r}{\mu_j} \left( 1 - e^{-\frac{k \mu_j}{r} z} \right) \quad (3.4)$$

ist.

Dabei gibt  $\rho g r \mu_j^{-1}$  den Grenzwert an, gegen den der Druck  $\sigma_{xx}$  auf die Wand asymptotisch strebt. Zu beachten ist, daß der Reibungskoeffizient  $\mu_j$  nicht identisch sein muß mit dem Reibungskoeffizienten  $\mu$ , der zwischen zwei Teilchen wirkt [123]. Der Grund hierfür ist, daß  $\mu_j$  auch von der Oberfläche der Wand abhängt (rauh, uneben, etc). Deswegen wird  $\mu_j$  als Fitparameter verwendet. In Abbildung 3.7(a) ist der Druckverlauf für verschiedene Werte von  $\mu_j$  gezeigt, der Radius der Röhre ist 10 cm. Für kleine Werte von  $z$  nimmt der Druck vergleichbar dem hydrostatischen Fall mit  $\rho g z$  zu. Je größer die Reibung ist, desto eher geht das System dann in asymptotische Bereiche konstanten Drucks über.

Ein Beispiel für eine Simulation ist in Abbildung 3.2 dargestellt. Simuliert wurde ein Rohr mit einem Durchmesser und einer Höhe von etwa 40 cm. Es wurden 10000 Teilchen mit einem Radius von  $1 \text{ mm} \pm 15\%$ , einem Reibungskoeffizienten von  $\mu = 0.5$ , einer Eckenzahl zwischen 8 und 10 und einen Elastizitätsmodul von  $10^7 \frac{\text{N}}{\text{m}^2}$  simuliert. Gefüllt wurde das Silo durch Einwurf der Teilchen am oberen Ende des Rohres. Dabei darf die Zuflußrate nicht zu groß sein, da sich sonst Pfropfen<sup>2</sup> in der Schüttgutsäule bilden können. Dabei ist das Granulat zwischen den Wänden eingeklemmt,  $\sigma_{xx}$  kann beliebig stark fluktuieren. Der Druckverlauf an den Wänden (Abbildung 3.2(a)) zeigt für  $\sigma_{xx}$  und  $\sigma_{zz}$ , daß auch in der Simulation ein Bereich der Sättigung erreicht wird. Durch die Simulation ist es nun möglich, die Annahmen der Janssenschen Lösung im Inneren des Granulates zu überprüfen. Die

<sup>2</sup>Die Pfropfen können in der Fördertechnik aber auch erwünscht sein, wie Gröger [123] für den Schlauchbandförderer zeigte.



**Abbildung 3.2:** Silo mit senkrechten Wänden,  $\mu = 0.4$

Spannungen in Vertikal- und Horizontalrichtung ( $\sigma_{xx}$  und  $\sigma_{zz}$ ) sind in Abbildung 3.2(b) ortsaufgelöst dargestellt. Man sieht, daß die Annahme, daß  $\sigma_{xx}$  über den Querschnitt konstant sei, relativ gut erfüllt ist. Bei  $\sigma_{zz}$  zeigen sich hingegen größere Schwankungen, die von ihrer Form auf die Ausbildung von Kraftpfaden schließen lassen. Berechnet man das Verhältnis  $k$  (siehe auch Abbildung 3.2(c)), dann ergibt sich der Mittelwert  $k = 0.64 \pm 0.19$ . Die Dichte  $\rho$  in Gl.3.4 gibt die mittlere Dichte des Granulates an, nicht die eines einzelnen Partikels. Dies wird durch das Produkt der Teilchendichte und der Raumauffüllung der Teilchen in der Simulation berechnet:

$$\rho = \rho_{rel} \cdot 5000 \frac{\text{kg}}{\text{m}^3} \quad (3.5)$$

Dabei ist  $\rho_{rel}$  (siehe Abbildung 3.2(d)) im gesamten System homogen, eine höhere Dichte durch die Kompression der unteren Bereiche oder Fluktuationen durch den Einfüllvorgang sind nicht zu erkennen. Gemittelt ergibt sich  $\rho_{rel} = 0.78 \pm 0.03$ . Mit diesen, von der Simulation vorgegebenen Werten, wird  $\mu_j$  in Gleichung 3.4 an die Daten der Simulation angefitet. Das Ergebnis eines least-square Fits ergibt  $\mu_j = 0.33$ . Damit stimmen die Druckverläufe für  $\sigma_{xx}$  und  $\sigma_{zz}$  sehr gut mit der Theorie überein.

### 3.2 Erweiterung auf verallgemeinerte Siloformen

Gleichung 3.4 wird nun von einem Rohr mit festem Radius auf eines mit veränderlichem Radius  $r(z)$  erweitert. Für die Elemente des Spannungstensors werden dieselben Annahmen wie in Abschnitt 3.1 gemacht. Für  $\sigma_{zz}$  wird angenommen, daß es über die ganze Breite konstant ist, und es gelte  $\sigma_{xz} = \sigma_{zx} = 0$ . Die Gleichgewichtsbedingung aus Gleichung 3.1 wird in der Form

$$\frac{d\sigma_{zz}}{dz} + \frac{\tau_w}{r(z)} - \rho g = 0 \quad (3.6)$$

geschrieben.

Um die Wandschubspannung mit der Coulombschen Reibung zu berechnen, benötigt man die x-Komponente der Normalspannung an der Wand. Mit den Komponenten des Normalenvektors auf die Wand

$$n_x = \frac{1}{\sqrt{1 + \left(\frac{d}{dz}r(z)\right)^2}} \quad \text{und} \quad n_z = -\frac{\frac{d}{dz}r(z)}{\sqrt{1 + \left(\frac{d}{dz}r(z)\right)^2}} \quad , \quad (3.7)$$

ist

$$\tau_w = \mu_j n_x \begin{pmatrix} n_x \\ n_z \end{pmatrix} \begin{pmatrix} \sigma_{xx} & 0 \\ 0 & \sigma_{zz} \end{pmatrix} \begin{pmatrix} n_x \\ n_z \end{pmatrix} \quad (3.8)$$

$$= \mu_j n_x (\sigma_{xx} n_x^2 + \sigma_{zz} n_z^2) \quad . \quad (3.9)$$

Bei einer schrägen Wand hat man noch einen direkten Betrag der Normalspannung zur Kraftkomponenten in z-Richtung. Dieser Beitrag ist proportional zu  $n_z$ , für fast senkrechte Wände ist er klein und wird im weiteren vernachlässigt.

Zusammen mit Gleichung 3.2 und 3.7 ist die zu lösende Differentialgleichung

$$\frac{d}{dz}\sigma_{zz} + \frac{\mu_j \sigma_{zz} (r'(z)^2 + k)}{r(z) (1 + r'(z)^2)^{\frac{3}{2}}} - \rho g = 0 \quad . \quad (3.10)$$

Als Lösung erhält man

$$\sigma_{zz}(z) = e^{-\mu_j \xi(z)} \rho g \int_0^z e^{\mu_j \xi(z')} dz' \quad (3.11)$$

mit

$$\xi(z) = \int_0^z \frac{r'(u)^2 + k}{r(u) (1 + r'(u)^2)^{\frac{3}{2}}} du \quad . \quad (3.12)$$

Die klassische Lösung von Janssen ergibt sich hieraus dann mit  $r(z) = r_0$ . Anzumerken ist noch, daß sich für  $\xi(z)$  und  $\xi(z) + c$  mit  $c \neq 0$  dieselbe Lösung  $\sigma_{zz}(z)$  ergibt:

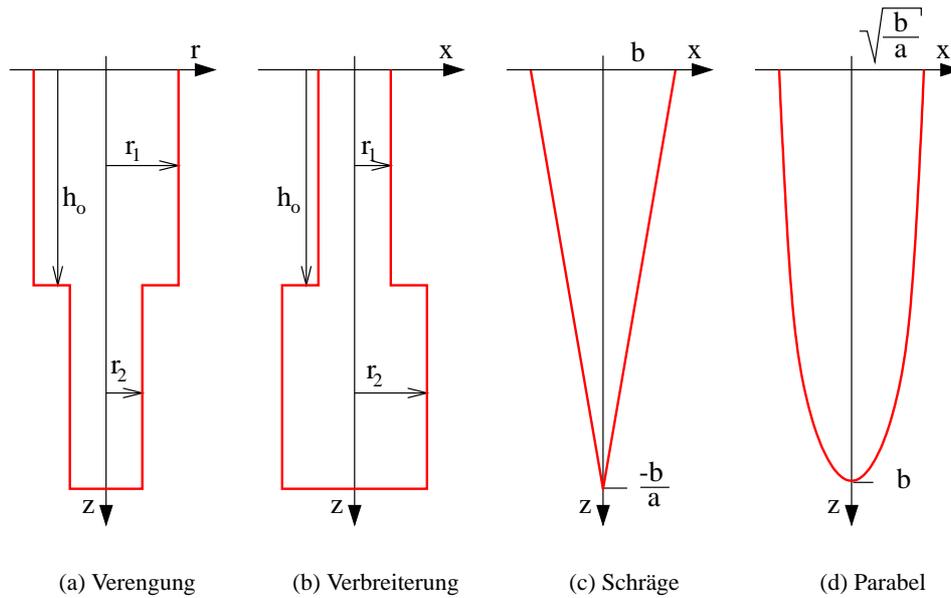
$$e^{-\mu_j(\xi(z)+c)} \rho g \int_0^z e^{\mu_j(\xi(z')+c)} dz' = e^{-\mu_j \xi(z)} \rho g \int_0^z e^{\mu_j \xi(z')} dz' \quad (3.13)$$

Es kann nun für verschiedene Rohrformen jeweils eine Lösung bestimmt werden, die sich dann in den folgenden Abschnitten mit Simulationsergebnissen vergleichen läßt.

### 3.3 Stufenförmiges Silo

Die erste zu untersuchende Rohrform ist ein Silo mit einer sprunghaften Änderung des Rohrdurchmessers, aber senkrechten Wänden. In Abbildung 3.3(a) und 3.3(b) sind die entsprechenden Rohre dargestellt. Die allgemeine Gleichung für diese Siloform lautet dann

$$r(z) = r_1 \Theta(h_0 - z) + r_2 (1 - \Theta(h_0 - z)) \quad , \quad (3.14)$$



**Abbildung 3.3:** Verschiedene untersuchte Siloformen

wobei für den Fall der Verengung  $r_1 > r_2$  gilt und für die Verbreiterung  $r_1 < r_2$ . Der Sonderfall  $r_1 = r_2$  ergibt die klassische Lösung. Im Fall  $r_1 < r_2$  ist eine wichtige Annahme nicht erfüllt: am Punkt  $z = h_0$  ist Voraussetzung, daß  $\sigma_{xx}$  und  $\sigma_{zz}$  über die ganze Breite konstant sind, nicht mehr erfüllt. Deswegen beschränkt sich die Gültigkeit der nachfolgenden Rechnung auf  $r_1 > r_2$ .

Als Lösung für die Spannung  $\sigma_{xx}$  aus Gleichung 3.11 erhält man

$$\sigma_{zz} = -\frac{\rho g}{\mu_j k} \left( r_1 e^{\frac{\mu_j k (h_0 - z)}{r_1}} \Theta(z - h_0) - \Theta(z - h_0) r_1 + r_2 e^{-\frac{\mu_j k (h_0 r_1 - h_0 r_2 + r_2 z)}{r_1 r_2}} \Theta(z - h_0) - r_2 e^{\frac{\mu_j k (h_0 - z)}{r_1}} \Theta(z - h_0) - r_2 + \Theta(z - h_0) r_2 + r_2 e^{-\frac{\mu_j k z}{r_2}} - r_2 e^{-\frac{\mu_j k z}{r_2}} \Theta(z - h_0) \right),$$

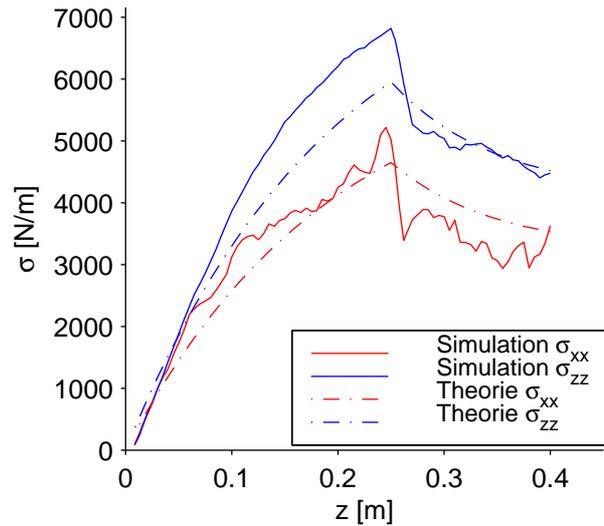
was umgeschrieben werden kann in

$$\sigma_{xx} = \begin{cases} \frac{\rho g r_1}{\mu_j} \left( 1 - e^{-\frac{\mu_j k z}{r_1}} \right) & z \leq h_0 \\ \frac{\rho g}{\mu_j} \left( (r_1 - r_2) e^{\frac{\mu_j k (h_0 - z)}{r_2}} + r_2 - e^{\frac{\mu_j k (h_0 r_1 - r_1 z - h_0 r_2)}{r_1 r_2}} r_1 \right) & h_0 < z \end{cases}. \quad (3.15)$$

Damit entspricht der obere Teil des Silos der Lösung für ein Rohr mit senkrechten Wänden, der untere Teil ist die Lösung von Gleichung 3.3 mit der Anfangsbedingung  $\sigma_{zz}(h_0) = \sigma'$ , wobei  $\sigma'$  durch die Lösung des oberen Teils des Rohres an der Stelle  $z = h_0$  vorgegeben wird. Beispiele für Druckverläufe sind in Abbildung 3.7(b) gezeigt, wobei die Stufe bei  $z = 50$  mm ist und die Radien  $r_1 = 100$  mm und  $r_2 = 50$  mm sind. Es kann die für eine ruhende Flüssigkeit paradoxe Situation entstehen, daß der Druck in einem Rohr weiter oben größer ist als am Fuß der Röhre.

Mit gc2d wurde ein entsprechendes Silo simuliert. Die Daten der Rohrgeometrie sind:

- $r_1 = 40$  mm



**Abbildung 3.4:** Der Druckverlauf in einem Rohr mit einer un stetigen Änderung des Radius.

- $r_2 = 20\text{mm}$
- $h_0 = 250\text{mm}$

Die verwendeten Teilchen haben dieselbe Form wie die Teilchen in Abschnitt 3.1. Der Reibungskoeffizient ist  $\mu = 0,4$ . Es wurden die Daten von zwei Simulationen mit je 10000 Teilchen gemittelt. Aus den Kräften der Simulation wird in Abbildung 3.4 der Verlauf der Wanddrücke berechnet. Deutlich zu erkennen ist die durch die Stufe entstandene Druckänderung. Aus den Daten der Spannungssensoren wird der Mittelwert für  $k$  berechnet; es ergibt sich  $k = 0,64 \pm 0,19$ . Die Position beeinflusst den lokalen Wert von  $k$  nicht. Dieser ist unabhängig davon, ob er im engen oder im weiten Teil des Rohres bestimmt wird. Zur Bestimmung von  $\mu_j$  wurde Gleichung 3.15 an die vorhandenen Daten angefitet. Das Ergebnis eines least-square Fits ergibt  $\mu_j = 0,33$ . Die Verläufe von  $\sigma_{xx}$  und  $\sigma_{zz}$  stimmen also gut mit den theoretischen Kurven überein.

### 3.4 Silo mit schrägen Wänden

Nun soll ein Silo berechnet werden, dessen Wände nicht senkrecht, sondern schräg sind. Der Radius wird durch

$$r(z) = az + b \quad (3.16)$$

beschrieben. Dabei muß immer  $a \leq 0$  gelten, so daß das Silo nach unten immer enger wird. Die zu lösenden Differentialgleichung ist

$$\frac{d}{dz}\sigma_{zz} + \frac{\mu_j \sigma_{zz} (b^2 + k)}{(a + zb) (1 + b^2)^{3/2}} - \rho g = 0. \quad (3.17)$$

Außerdem beschränken wir uns auf den Bereich  $z \leq -\frac{b}{a}$ , da negative Radien nicht sinnvoll sind. Mit der Abkürzung  $\alpha = (1 + a^2)^{\frac{3}{2}}$  erhält man als Funktion für den Wanddruck dann

$$\sigma_{zz} = \frac{\alpha \rho g}{\mu_j b^2 + \mu_j k + \alpha a} \left( b + za - (b + za)^{-\frac{\mu_j (b^2+k)}{\alpha a}} b^{\frac{\mu_j b^2 + \mu_j k + \alpha a}{\alpha a}} \right). \quad (3.18)$$

Dann gilt in der am unteren Ende des Silos befindlichen Spitze  $\sigma_{xx}(-\frac{b}{a}) = 0$ . Die Beispiele für die Druckverläufe in Abbildung 3.7(b) sind mit  $a = 1000$  und  $b = 100$  mm gerechnet. Allerdings wird in der Simulation die Kontinuumsnäherung im unteren Bereich nicht mehr gültig sein, sobald der Radius kleiner als einige Teilchendurchmesser wird. Ein Rückgang von  $\sigma_{xx}$  und  $\sigma_{zz}$  auf Null wird nur sehr schwer zu beobachten sein. Interessant ist, daß es bei dieser Form des Silos zu einem Druckmaximum an der flachen Wand kommt. Gewöhnlich würde man diese Druckspitzen an Stellen erwarten, an denen sich bestimmte Eigenschaften der Wand ändern (Radius, Reibung durch Verunreinigungen etc.)

Die Parameter für die Simulation sind

- $a = -0.1087$
- $b = 65$  mm.

Für die Simulation wurde die Siloform leicht modifiziert. Da man nicht mehr über ausreichend viele Teilchen mitteln kann, ist ein Silodurchmesser kleiner als 10 Teilchen ist nicht sinnvoll. Deswegen wurde die Spitze des Silos abgeschnitten. In Abbildung 3.5 werden die Daten zweier Simulationen mit unterschiedlichen Reibungskoeffizienten gezeigt. Die verwendeten Teilchen haben einen Radius von 1.5 mm, alle anderen Materialparameter entsprechen den Simulationen aus den vorhergehenden Abschnitten. Bei beiden Simulationen weist  $\sigma_{xx}$  deutlich geringere Fluktuationen als  $\sigma_{zz}$  auf. Für  $\mu = 0.4$  (Abbildung 3.5(a)) ist  $k = 0.77 \pm 0.24$ ; bei  $\mu = 0.7$  (Abbildung 3.5(b)) ist  $k = 0.76 \pm 0.12$ . Durch das Anfitzen der allgemeinen Lösung (Gl. 3.18) erhält man  $\mu_j = 0.3$  bzw  $\mu_j = 0.37$ . In beiden Fällen nehmen die Fluktuationen der Drücke nach unten hin zu. Für  $z > 0.35$  sind weniger als 25 Teilchen nebeneinander im Rohr, die Bestimmung der Spannungen wird unzuverlässig. Die Theorie beschreibt aber in beiden Fällen den Anstieg und die Position der Maxima von  $\sigma_{xx}$  und  $\sigma_{zz}$  gut.

### 3.5 Parabelförmiges Silo

Es soll nun ein Silo mit einem parabolischen Profil untersucht werden. Die Wand wird durch

$$r(z) = \sqrt{\frac{b-z}{a}} \quad (3.19)$$

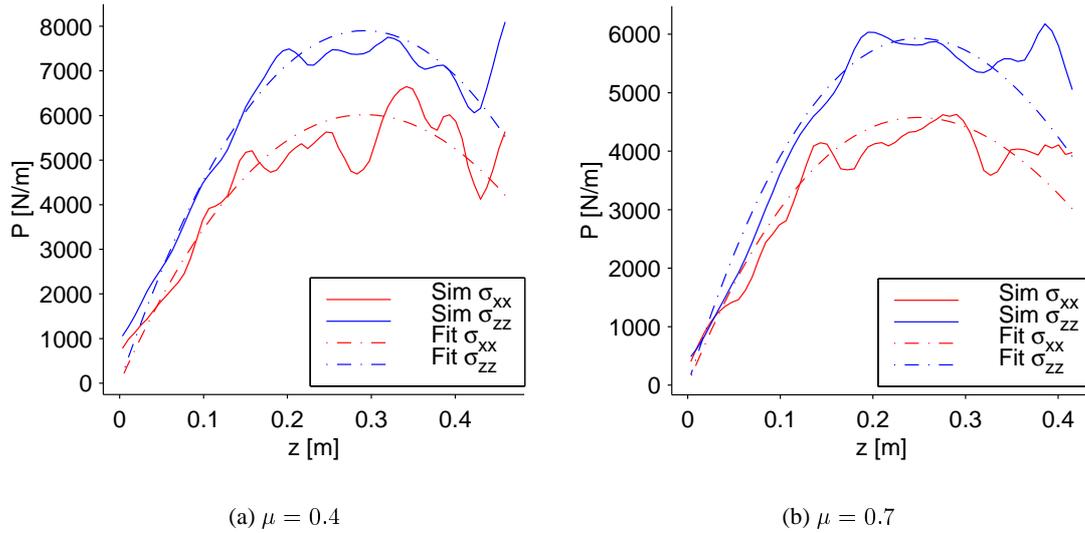
festgelegt, damit hat das Silo die Tiefe  $b$  und an der Oberfläche den Radius  $\sqrt{\frac{b}{a}}$ .

Nach dem Einsetzen dieser Gleichung in 3.11 erhält man

$$\xi(z) = \frac{1 - 4kab + 4kaz - 2k}{\sqrt{4ab - 4az + 1}} \quad (3.20)$$

Im Bereich sehr kleiner Werte von  $z \ll b$ , also nahe an der Oberfläche des Granulates, kann  $\xi(z)$  in

$$\Xi(z) = \frac{(1 - 4kab - 2k)}{\sqrt{4ab + 1}} + 2 \frac{a(1 + 4kab)z}{(4ab + 1)^{3/2}} \quad (3.21)$$



**Abbildung 3.5:** Druckverläufe bei der Simulation eines Silos mit schrägen Wänden. Dargestellt sind  $\sigma_{xx}$  und  $\sigma_{zz}$  sowie die Kurven aus Gleichung 3.18.

entwickelt werden. Nach Gleichung 3.13 hat der konstante Term keine Bedeutung für die Lösung, wird also weggelassen.

Mit

$$r_P = \frac{(1 + 4ab)^{3/2} k}{2a(1 + 4kab)k} = \sqrt{\frac{b}{a} + \frac{1}{4a^2}} \left(1 + \frac{k-1}{1 + 4kab}\right) \quad (3.22)$$

kann man  $\Xi(z)$  kurz als

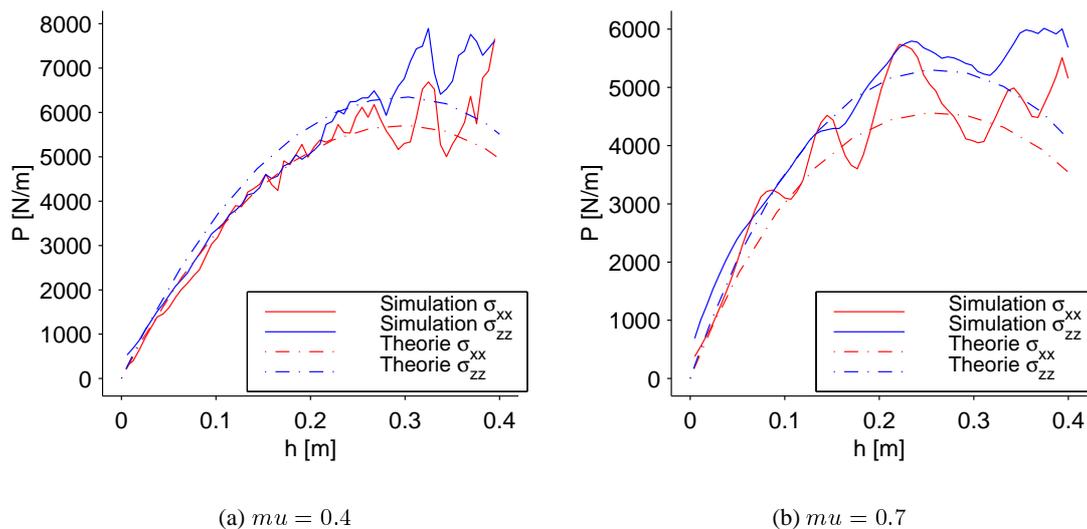
$$\Xi(z) = \frac{k}{r_P} \cdot z \quad (3.23)$$

schreiben. Die Lösung ist dann

$$\sigma_{xx} = \frac{\rho g r_P}{\mu_j k} \left(1 - e^{-\frac{\mu_j k z}{r_P}}\right) \quad (3.24)$$

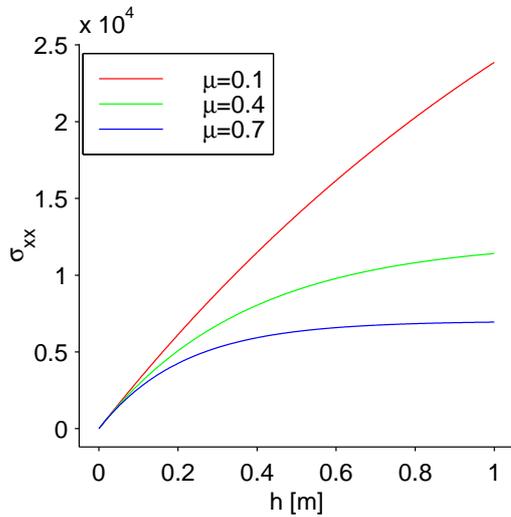
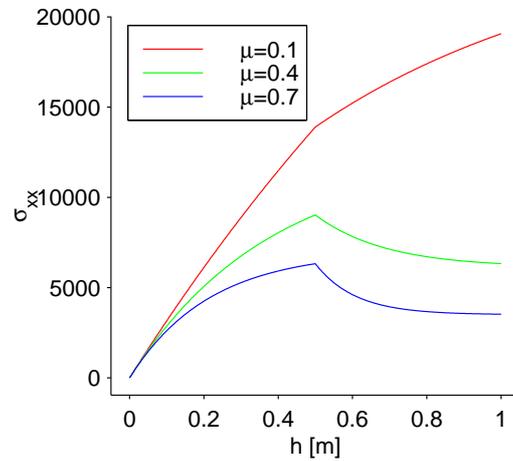
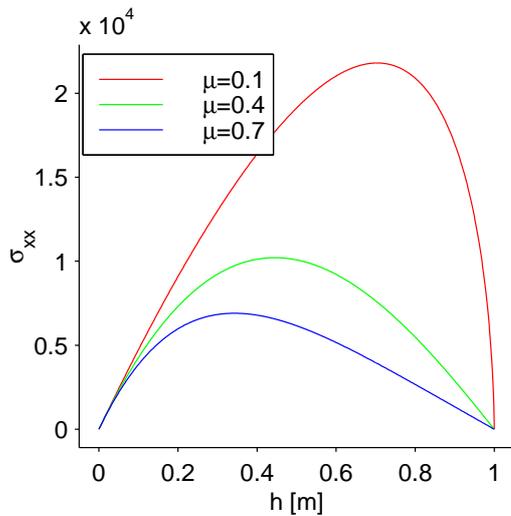
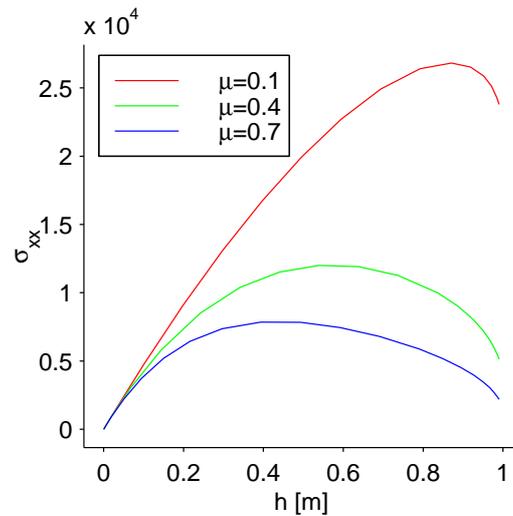
Dies bedeutet, daß man den oberen Teil des parabelförmigen Silos durch eines mit senkrechten Wänden und dem Radius  $r_P$  annähern kann. Wenn ein hohes schmales Silo angenommen wird, also  $b \gg \frac{1}{a}$  und  $k \approx 1$ , liegt  $r_P$  sehr nahe an  $r_P = \sqrt{\frac{b}{a}}$ , also dem tatsächlichen Radius an der Oberfläche. Interessant ist auch, wie sich die Spannung  $\sigma_{zz}$  am Boden des Silos verhält. Mit  $\xi(b) = e^{(1-2k)} > 0$  und  $\int_0^b e^{\mu_j \xi(z)} > 0$  folgt, daß  $\sigma_{zz}$  am Boden des Silos ungleich 0 ist (siehe Abbildung 3.7(d)). Allerdings ist, wie bei dem Beispiel mit den schrägen Wänden, einerseits die Untersuchung dieses Bereichs sehr schwierig, andererseits ist die Annahme eines Kontinuums unzutreffend. Bei der Simulation wurde dieselbe Teilchensorte mit einem Radius von 1.5 mm verwendet, alle anderen Materialparameter entsprechen den Simulationen aus den vorhergehenden Abschnitten. Die Parameter der Parabel entsprechend Gl. 3.19 sind

1.  $a=100\frac{1}{m}$
2.  $b=0.46m$ .



**Abbildung 3.6:** Druckverläufe bei der Simulation eines parabelförmigen Silos. Dargestellt sind  $\sigma_{xx}$  und  $\sigma_{zz}$  sowie die Kurven aus Gleichung 3.24

Die Abbildung 3.6 zeigt zwei Simulationen mit unterschiedlichen Werten des Reibungskoeffizienten. Für  $\mu = 0.4$  erhält man  $k = 0.86 \pm 0.14$  und für  $\mu = 0.7$  ist  $k = 0.89 \pm 0.11$ . Für die Darstellung der Theorie (Gl. 3.24) wurde wiederum  $\mu_j$  an die Daten der Simulation angepaßt. Dargestellt sind  $\mu_j = 0.3$  (Abbildung 3.6(a)) und  $\mu_j = 0.47$  (Abbildung 3.6(b)). Der Abfall der Spannungen ist nur ansatzweise zu erkennen, da der Bereich zu nahe am Scheitel der Parabel ist. Der Anstieg und das Maximum der Kurven werden gut wiedergegeben.

(a) senkrechte Wände;  $r = 0.1\text{m}$ .(b) Verengung;  $r_1 = 0.1\text{m}$ ,  $r_2 = \frac{1}{2}r_1$ ,  $h_0 = 0.5\text{m}$ (c) Schräge Wände;  $a = -0.1$ ,  $b = 0.1\text{m}$ (d) Parabelförmige Wände;  $a = 100 \frac{1}{\text{m}}$ ,  $b = 1\text{m}$ .

**Abbildung 3.7:** Theoretische Kurven für  $\sigma_{xx}$  bei verschiedenen Siloformen. Es sind jeweils die Kurven für  $\mu_j = 0.1$ ,  $\mu_j = 0.4$  und  $\mu_j = 0.7$  dargestellt. Die anderen Parameter sind  $k = \frac{2}{3}$  und  $\rho = 5000\text{kgm}^{-2}$ . Die Höhe ist bei allen Silos 1m, die maximale Breite immer 0.1m.

## Kapitel 4

# Granulare Aufschüttungen

### 4.1 Geordnete Sandhaufen

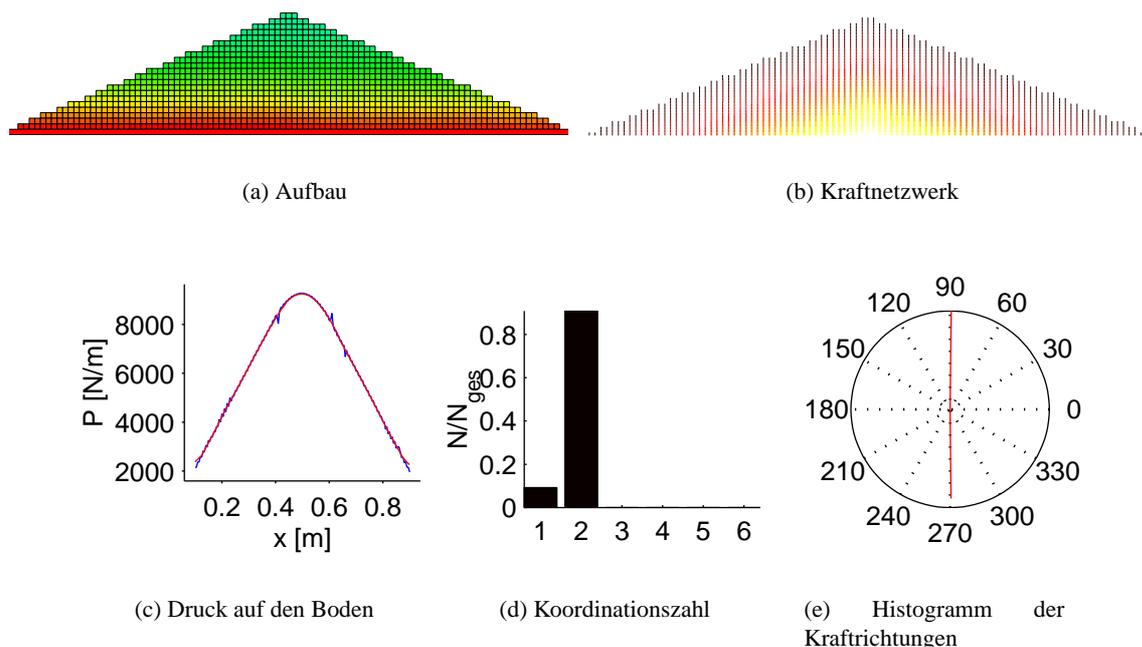
Bei der Frage nach der Ursache für das lokale Druckminimum bei Aufschüttungen wurden auch Systeme vergleichbar denen in Abbildung 1.8 untersucht, bei denen die Teilchen geordnet sind. Das Modell von Hong [86] beschreibt Kugeln, die auf einem hexagonalen Gitter angeordnet sind. Die Methode von Oron [156, 157] bestimmt die exakte Kraftverteilung für ein gegebenes Kontaktnetzwerk. Der Vorteil dieser Herangehensweise ist, daß sich die Kräfte analytisch bestimmen lassen. Bei beiden Autoren sind die Haufen aber nicht freistehend, also nicht nur durch Reibung am Untergrund stabilisiert. Der Grund dafür ist, daß die Reibung nicht berücksichtigt wird. Stattdessen wird der Haufen entweder links und rechts durch ein fixiertes Teilchen stabilisiert, oder aber der Untergrund besteht aus einer ganzen Reihe unbeweglicher Teilchen.

Für einige Systeme, die im Inneren stark geordnet sind, werden nun die mit  $gc2d$  simulierten Daten dargestellt. Dabei wurde immer Reibung, Elastizität etc. mit berücksichtigt, um realistische Ergebnisse zu erhalten. Wie bereits in Abbildung 1.8 dargestellt, kann sich trotz äußerlich gleicher Form die innere Struktur sehr stark auf die einzelnen Größen wie Spannungstensoren und Druck auswirken.

Zuerst soll das wohl einfachste Modell untersucht werden: einzelne Säulen aus quadratischen Partikeln. Diese Simulation produziert, was zu erwarten ist. Das Kraftnetzwerk (Abbildung 4.2(b)) zeigt, daß zwischen den einzelnen Säulen kein Kontakt besteht. Entsprechend ist die Koordinationszahl (Abbildung 4.2(d)) bei fast allen Teilchen gleich 2, nur bei Teilchen an der Oberfläche ist sie 1. Ähnliches zeigt sich auch im Histogramm der Kraftrichtungen (Abbildung 4.2(e)). Die Kräfte wirken senkrecht auf den Untergrund. Der Druck (Abbildung 4.2(c)) auf den Untergrund hängt damit nur von der jeweiligen Säulenhöhe ab und steigt linear an. Die kleinen Spikes sind hier, wie auch in den



**Abbildung 4.1:** Bei regelmäßigen Partikelanordnungen kann es vorkommen, daß der Abstand der Kräfte (schwarz) auf den Boden (grau) und die Breite des Meßintervalls (rot) nicht harmonisieren. Dies führt zu den Spikes bei den Kurven für den Druckverlauf (z.B. Abbildung 4.2(c))



**Abbildung 4.2:** Ein Haufen aufgebaut aus säulenartig gestapelten Rechtecken (vgl. Abbildung 1.8a)

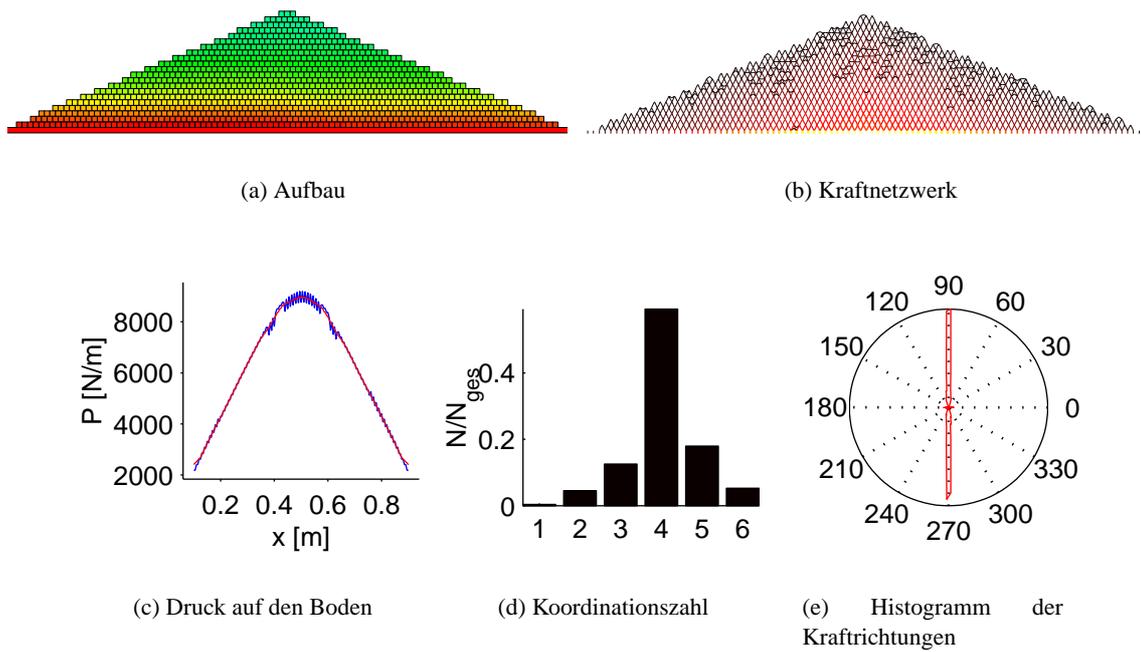
weiteren Graphen in diesem Abschnitt, auf das Verhältnis von Testfläche (vgl. Gleichung 2.39) und Teilchenbreite zurückzuführen (siehe Abbildung 4.1). Deswegen wird in allen Abbildungen dieses Abschnittes, in denen der Druck auf den Boden dargestellt wird, zusätzlich der gleitende Durchschnitt eingezeichnet. Die abgerundete Spitze entsteht durch die Mittelung über mehrere Säulen.

Scheiben auf einem Rechteckgitter anzuordnen, ergibt ein sehr instabiles Gebilde. Auch wenn sich die verwendeten Polygone in der Simulation entsprechend plazieren ließen, wäre ein anderes Verhalten als bei den Quadraten ist nicht zu erwarten.

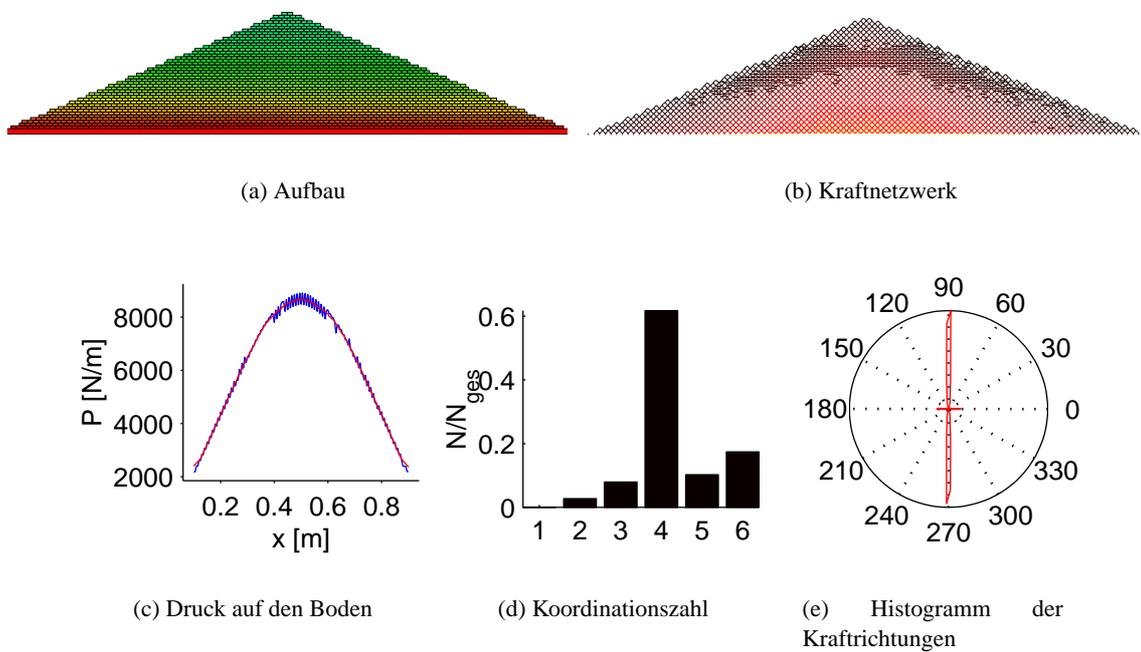
Für die nächsten Simulationen wurden Quadrate Abbildung 4.3 bzw. Abbildung 4.4 (Breite zu Höhe = 2:1) auf einem Dreiecksgitter angeordnet. Obwohl das System mit den flachen Rechtecken doppelt so viele Schichten besitzt, sind die Eigenschaften doch sehr ähnlich. Die Histogramme der Krafrichtungen<sup>1</sup> (Abbildungen 4.3(e)) und 4.4(e)) zeigen, daß fast alle Kräfte senkrecht zum Untergrund gerichtet sind. Allerdings sind auch einige Kräfte parallel zum Untergrund vorhanden. Auf ein Partikel drücken von oben immer zwei Teilchen. Entsprechend der Druckverteilung (4.3(c) und 4.4(c)) ist der Druck auf der Innenseite etwas größer, auf das Teilchen wirkt ein Drehmoment. Wird dadurch das Teilchen stark genug verdreht, bekommt es Kontakt zu seinen Nachbarn in derselben Ebene. Diese Kontakte sieht man dann auch bei der Koordinationszahl (4.3(d) und 4.4(d)), ca. 40% der Teilchen haben Kontakt mit 5 oder 6 Nachbarn. Die Druckverteilungen auf den Untergrund entsprechen dem Fall der Säulen.

Statt Quadrate werden beim nächsten System Scheiben auf einem Dreiecksgitter aufgeschichtet. Diese Anordnung ist nur für kleinere Systeme und flache Böschungswinkel stabil. Sind die Scherkräfte in den Flanken zu groß, fängt das System an zu rutschen oder zu rollen. Die Simulation aus Abbildung 4.5(a) ist gerade noch stabil, lediglich der Fuß des Haufens hat sich schon etwas verschoben,

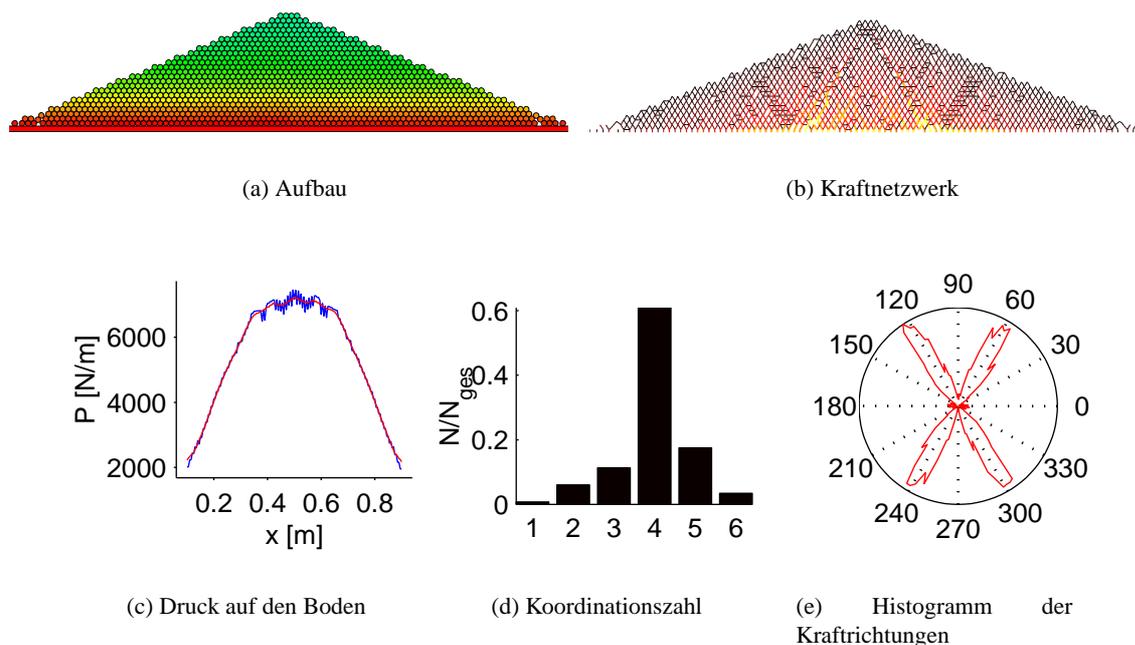
<sup>1</sup>Die Kraftnetzwerke geben entsprechend ihrer Definition in Abschnitt 2.3.2 keine Information über die Krafrichtungen wieder. Besonders deutlich ist dies in den Abbildungen 4.3(b) und 4.4(b) zu sehen. Bei runden Partikeln weisen die Kräfte häufig in Richtung des Schwerpunkts, hier ist die Übereinstimmung der Richtungen dann besser.



**Abbildung 4.3:** Ein Haufen, bei dem Quadrate auf einem Dreiecksgitter plaziert wurden.



**Abbildung 4.4:** Ein Haufen bei dem flache Rechtecke auf einem Dreiecksgitter plaziert wurden.



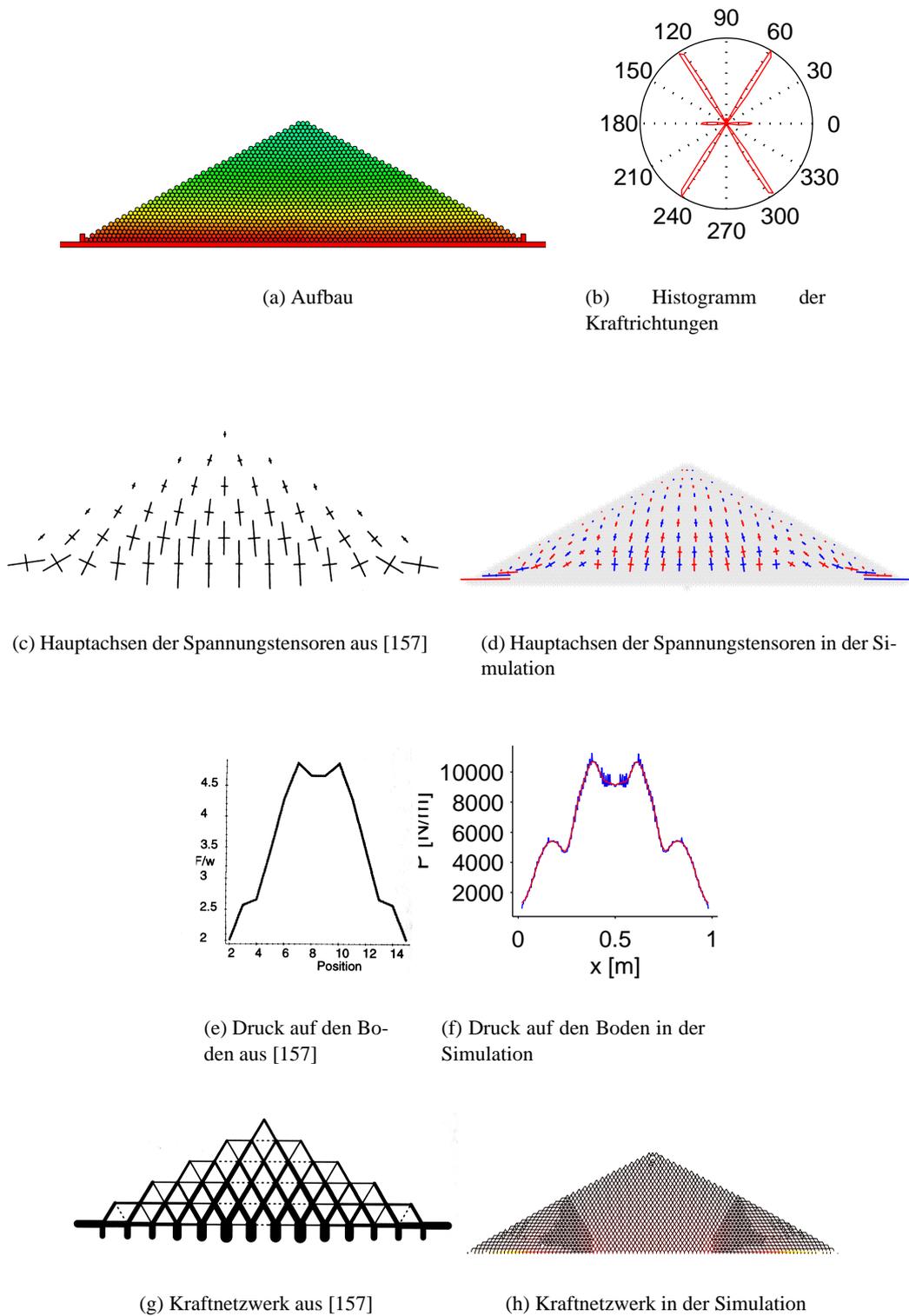
**Abbildung 4.5:** Ein Haufen aus hexagonal gestapelten Kugeln (vgl. Abbildung 1.8c)

wie man auch am Kraftnetzwerk (4.5(b)) und dem Haufen selbst sieht. Dies bedeutet aber auch, daß sich ganze Bereiche des Haufens verschoben haben. Dies wird durch die breite Winkelverteilung der Krafrichtungen (4.5(e)) deutlich. Der Druck (4.5(c)) auf den Untergrund weist einen breiten flachen Bereich auf, der dem System von Hong entspricht. Auffällig ist, daß die mittlere Koordinationszahl (4.5(d)) wieder nahe bei 4 ist, obwohl die Kugeln 6 Nachbarn hat. Am Kraftnetzwerk und der Winkelverteilung sieht man, daß die Nachbarn zur linken und zur rechten eines Teilchens nicht am Kontaktnetzwerk beteiligt sind.

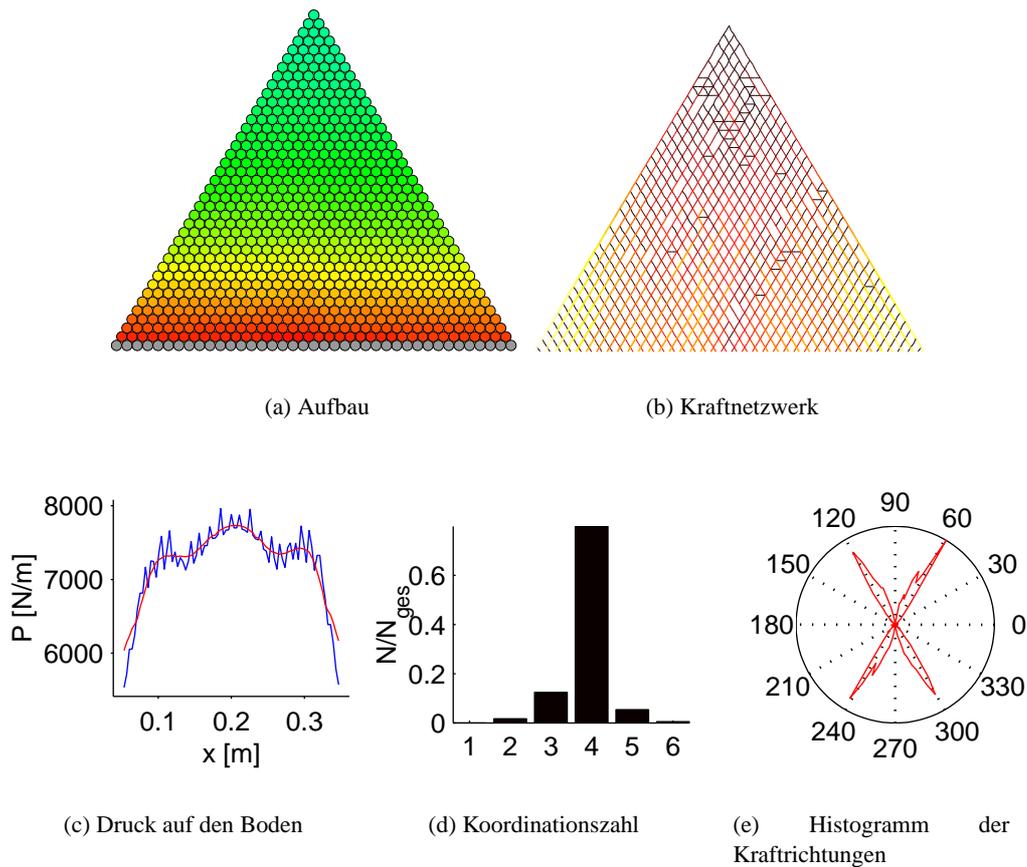
Das Auseinanderrollen des Haufens kann verhindert werden, indem links und rechts am Fuß ein Stück unbewegliche Wand plziert wird. Das System entspricht dem, welches Oron [157] als Anordnung c) beschreibt.

Mit Hilfe von `gc2d` ist es möglich, größere Setups als jene von Oron (11 Teilchen als Basis) zu berechnen. Der hier vorgestellte Haufen hat eine Basis von 113 Teilchen. Im Vergleich der Ergebnisse zeigen sich die selben charakteristischen Merkmale. Die Druckverteilung mit dem lokalen Minimum ist ähnlich. Die Flanken des mit `gc2d` simulierten Systems zeigen links und rechts je ein Nebenminimum, bei Oron sind hier nur Schultern zu erkennen. Dies ist vermutlich ein Teilchenzahleffekt, da bei dem kleineren System dieser Bereich nur zwei Teilchen, im großen System hingegen circa dreizehn Teilchen breit ist. Die Spannungstensoren zeigen das gleiche Bild, ebenso die Kraftnetzwerke. In beiden Simulationen sind die Kräfte im Bereich des Fußes in der untersten Schicht parallel zum Untergrund und steigen nach außen hin an. Dies bedeutet aber auch, daß dieses System ohne die künstliche Einengung durch die Wände nicht stabil wäre.

Eine weitere Möglichkeit, eine Kugelpackung ohne Reibung zu stabilisieren, besteht darin, den Untergrund sehr rau zu gestalten. Im Fall der Systeme von Hong bedeutet dies, daß der Untergrund aus fixierten Kugeln besteht. Nach Hong läßt sich der Druck durch einfaches Abzählen bestimmen. Der Druck müßte über die ganze Breite des Haufens konstant sein. Ansatzweise ist dies in Abbildung 4.6(e) zu sehen, allerdings kann sich, anders als im theoretischen Modell mit harten Kugeln,



**Abbildung 4.6:** Ein Haufen aus hexagonal gestapelten Kugeln, zusätzlich wird das System durch Wände links und rechts der Basis gestützt.



**Abbildung 4.7:** Ein Haufen aus hexagonal gestapelten Kugeln auf einem Untergrund aus ortsfesten Kugeln (grau).

der Schwerpunkt der Teilchen verlagern. Die dadurch entstehenden Veränderungen der Krafrichtung und des Kraftnetzwerkes führen zu den Abweichungen.

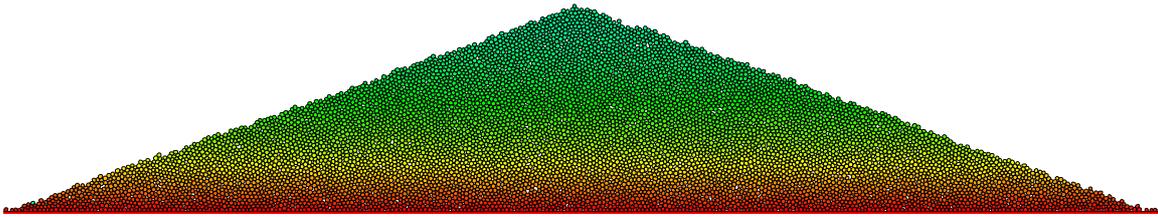
Bei allen Haufen, die auf Gittern aufgesetzt wurden, ist die Dichte im Inneren homogen, ein Kern mit abweichender Dichte bildet sich nicht. Das System, welches ein lokales Minimum zeigt, ist aber im Hinblick auf das Kraftnetzwerk (maximale Kräfte am Fuß) nicht typisch für experimentelle Systeme. Offensichtlich besitzen diese Systeme keine ausreichende Komplexität, um einen experimentellen Dip zu reproduzieren.

## 4.2 Sandhaufen

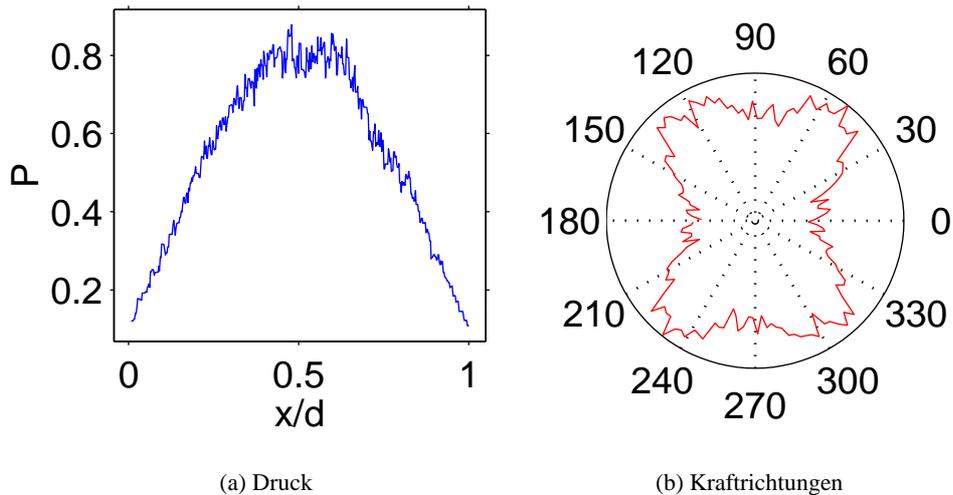
### 4.2.1 Geschichtete Systeme

In verschiedenen Experimenten wurde wiederholt gezeigt, daß ein kegelförmiger Haufen, der schichtweise aufgebaut wurde, kein Minimum im Druck auf den Untergrund zeigt. Bei zweidimensionalen Systemen, die nach dem gleichen Prinzip erzeugt wurden, ist ein ähnliches Ergebnis zu erwarten.

Technisch wurde dies so realisiert, daß die Startposition der einzelnen Teilchen auf einer Linie ausgewürfelt wird. Die Länge dieser Linienquelle gibt dann die Basis des Haufens vor. Im Verlauf der Simulation wurde die Strecke dann immer weiter verkürzt, bis sie die Länge Null hat. Dadurch entsteht



**Abbildung 4.8:** Bild eines schichtweise aufgeschütteten Sandhaufens.

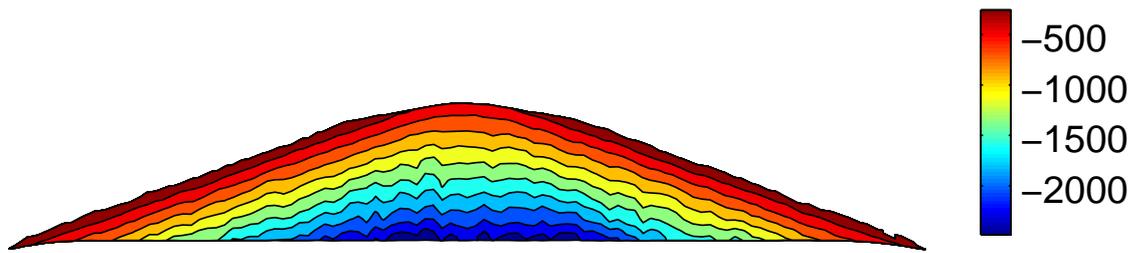


**Abbildung 4.9:** Druck auf den Untergrund und Histogramm der Krafrichtungen bei einem schichtweise aufgeschütteten Haufen.

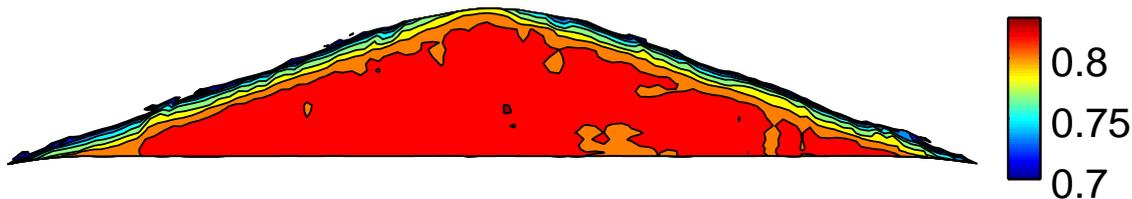
ein dreieckiger Haufen, dessen Flankenwinkel von der Geschwindigkeit der Streckenverkürzung beeinflusst wird. Für die Simulation wurden Polygone mit 12 Ecken verwendet, der Radius der Teilchen beträgt  $0.8\text{mm} \pm 20\%$ , der Reibungskoeffizient ist  $\mu = 0.6$ , der Youngmodul  $Y = 10^7 \frac{\text{N}}{\text{m}}$ . Kohäsionskräfte wurden nicht benutzt, da trockene Sandhaufen simuliert werden sollen, bei denen auch im Experiment versucht wird, Kohäsion zu vermeiden. Der Haufen hat eine Breite von  $d = 50\text{cm}$ , eine Höhe  $h = 8\text{cm}$  und einen Winkel von  $21^\circ$ .

Abbildung 4.8 zeigt den Haufen am Ende der Simulation. Die Farbe der Teilchen entspricht in dieser Darstellung ihrem ‘Alter’, also dem Zeitpunkt, zu dem sie in der Simulation eingesetzt wurden. Man sieht deutlich den schichtweisen Aufbau. Aufgrund des Winkels von  $21^\circ$ , der unter dem Böschungswinkel eines derartigen Systems liegt, haben sich am Rand keine Lawinen gebildet, die zu Umordnungen hätten führen können.

Der Druck auf den Untergrund wird über Gleichung 2.39 berechnet, wobei ein einzelner Testbereich im Durchschnitt 17.1 Teilchen enthält. In Abbildung 4.9(a) ist der Druck dargestellt, wobei hier wie im folgenden der Druck mit  $\rho gh$  skaliert wurde. Für die Darstellung der Spannungstensoren werden diese aber weiterhin nicht skaliert dargestellt. Die x-Achse wird mit  $\frac{1}{d}$  skaliert und so verschoben, daß sie von 0 (linke Seite des Haufens) bis 1 (rechte Seite des Haufens) geht. Es ist zu erkennen, daß dieser Haufen in der Mitte einen leicht abgeflachten Bereich hat, ein echtes Minimum ist nicht vorhanden. Die minimale Absenkung des Drucks in der Mitte ist kleiner als die Fluktuationen. Diese



**Abbildung 4.10:** Dargestellt ist  $\sigma_{zz}(x, y)$  in einem geschichteten Haufen, die Einheit ist  $\frac{N}{m}$ .



**Abbildung 4.11:** Dichte eines geschichteten Haufens

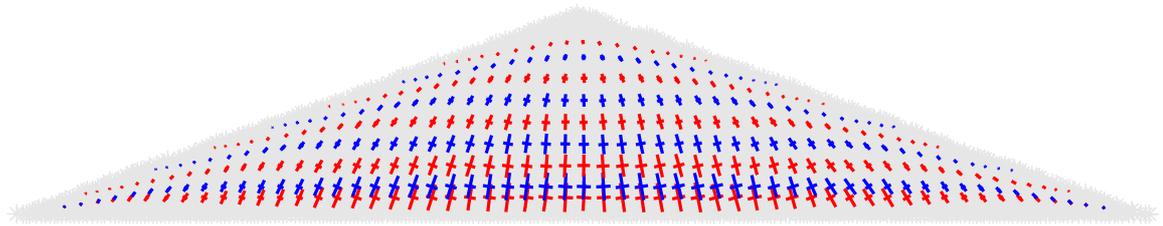
Druckverteilung entspricht somit den Experimenten.

Nun soll untersucht werden, welche Eigenschaften das Innere eines derartigen Systems hat. Das Histogramm der Krafrichtungen (Abbildung 4.9(b)) zeigt eine leichte Ausrichtung der Kräfte in Richtung von etwa  $230^\circ$  und  $310^\circ$ . Dies ist aus der Konstruktionsgeschichte des Systems zu verstehen. Jedes Teilchen sucht sich auf der immer waagrechten, aber sehr rauhen Oberfläche des Systems seinen Platz. Da die Teilchen eine breite Größenverteilung haben, ist ein Ordnen auf einem hexagonalen Gitter nicht zu erwarten. Jedes Teilchen hat im Durchschnitt mit 3.6 anderen Teilchen einen Kontakt.

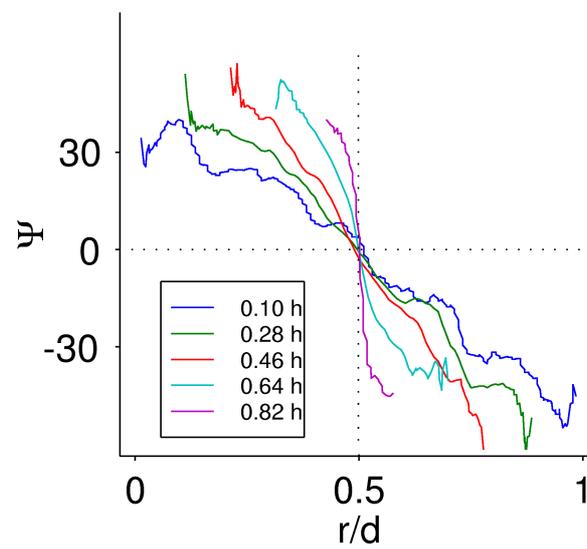
Die Experimente haben bisher nur gezeigt, daß es am Boden des Systems kein Druckminimum gibt. Vorstellbar wäre, daß sich im Inneren des Systems trotzdem ein Bereich verminderten Drucks ausbildet. Diese Überlegung wird durch Abbildung 4.10 widerlegt. Hier ist die Komponente  $\sigma_{zz}$  des Spannungstensors aufgetragen. Dargestellt sind die Isobaren des Systems. Im Inneren dieses, wie aller anderer untersuchten Systeme ist kein Druckminimum zu erkennen.

Die aus Gleichung 2.45 berechnete lokale Dichte wurde mit der Dichte eines Teilchens skaliert  $\rho_{rel} = \frac{\rho}{\rho_{Teilchen}}$ . Dies ist gleichzeitig die Raumauffüllung des Systems. In Abbildung 4.11 ist  $\rho_{rel}$  farbkonturiert dargestellt. Im Inneren des Haufens ist  $\rho_{rel}$  konstant, die Abnahme der Dichte in einem schmalen Randbereich liegt an der Art der Berechnung. In diesem Bereich schneidet die Testfläche, innerhalb der die Dichte bestimmt wird, die Oberfläche des Sandhaufens. Obwohl die Testfläche dann angepaßt wird, wird eine geringere Dichte berechnet (siehe Abbildung 2.10 b). Der Wert von  $\rho_{rel}$  liegt im Inneren etwa bei 0.81.

In Abbildung 4.12 sind für einige Spannungstensenoren die Hauptachsen dargestellt. Da die Testflächen an die Kontour des Sandhaufens angepasst werden (siehe Abschnitt 2.3.4 und Abbildung 2.11), sind die Schnittpunkte der Spannungstensorhauptachsen nicht auf einer geraden Linie, sondern weichen nahe der Oberfläche des Haufens von ihr ab. Für die Spannungstensenoren wurde jeweils der Winkel  $\Psi$  berechnet, der die Neigung des Spannungstensors angibt. Für einige Höhen wurden dann diese Winkel in Abhängigkeit von der X-Position in Abbildung 4.13 dargestellt. Was schon in dem Bild der Spannungstensenoren selbst erkennbar ist, wird hier noch einmal deutlich. Die Spannungstensenoren in



**Abbildung 4.12:** Spannungstensenoren im Inneren des Sandhaufens. Es wurde nicht alle berechneten Spannungstensenoren dargestellt, sondern in der Vertikalen jeder dritte und in der Horizontalen jeder vierte Spannungstensor. Die Farben sollen nur die Unterscheidbarkeit verbessern.



**Abbildung 4.13:** Winkel der Hauptachsen des Spannungstensors in verschiedenen Höhen eines geschichteten Haufens.

einer Hälfte des Haufens sind nicht parallel. In der Mittel steht die größere der Hauptachsen senkrecht. In diesem System gilt, wie erwartet, die FPA-Annahme nicht. Dies wäre nach Wittmer et al. eine Konsequenz einer für die Erzeugung eines Druckminimums ungünstige Entstehungsgeschichte des Sandhaufens und würde der Theorie in keiner Weise widersprechen.

#### 4.2.2 Geschüttete Systeme

Ein Beispiel für einen experimentellen Aufbau, der einen Sandhaufen mit Dip zeigt, ist Abbildung 1.7. Das Granulat fließt durch einen Trichter, so daß der Auslaß als Punktquelle für Teilchen interpretiert werden kann. Dieser Aufbau wurde für die Simulationen in gleicher Weise in zwei Dimensionen realisiert. Der Untergrund ist dabei immer eben und hat in Bezug auf Reibung, Elastizität etc. die gleichen Eigenschaften wie die simulierten Partikel. Die Ergebnisse dieser Simulationen sollen im folgenden vorgestellt werden.

Verschiedenen Experimente [78,81] haben gezeigt, daß die Tiefe des Druckminimums von der Form und Oberfläche der verwendeten Teilchen abhängt. Die verschiedenen Veröffentlichungen zusammengefaßt ergeben folgendes Bild: sind die Partikel alle etwa gleich groß, gibt es im allgemeinen keinen Dip. Ausnahmen hiervon zeigen sich, wenn

1. die Teilchen sehr rauh sind (Seesand)
2. die Teilchen so klein werden, daß kohäsive Kräfte zu berücksichtigen sind.

Der Einfluß der Kohäsion auf die Entstehung des Dips bei Simulationen wird in [158] diskutiert. Im folgenden soll der Einfluß der Oberflächenrauigkeit der Partikel beschrieben werden. Die Abbildungen 4.14(a) und 4.15(a) zeigen die Druckverteilung bei der Simulation von zwei Systemen mit unterschiedlichen Teilchenarten. Bei beiden Systemen wurden folgende Simulationsparameter verwendet:

1.  $\Delta t = 2 \cdot 10^{-6} \text{ s}$
2.  $Y = 1 \cdot 10^7 \frac{\text{N}}{\text{m}}$
3.  $\gamma = 0.75$
4.  $\mu = 0.6$
5. Teilchenradius 1.5mm

Der Unterschied zwischen den Simulationen ist die Zahl der Ecken. Die Simulationen aus Abbildung 4.14(a) haben 7 Ecken, die aus Abbildung 4.15(a) 13 Ecken. Die Ecken werden in beiden Fällen zufällig auf einer Ellipse plziert (siehe **RAND1** in Tabelle B.1.6). Die Simulation mit den Siebenecken zeigt ein Druckminimum, die mit den "glatteren" 13-Ecken nicht mehr. Bei Systemen mit gleich großen, runden Teilchen ordnen sich im allgemeinen die Partikel während des Aufschüttvorgangs auf einem hexagonalen Gitter an. Die Druckverteilung zeigt nach Hong [86] keinen Dip. Bei der Simulation mit Polygonen anstatt Scheiben werden die Teilchen durch ihre Oberflächenrauigkeit am Rollen gehindert. Damit wird die gitterartige Anordnung gestört. Bei den rauhen Teilchen (mit wenigen Ecken) sieht man auch am Histogramm der Kraftrichtungen (Abbildung 4.14(b)), daß zwar die Richtungen  $60^\circ$  und  $120^\circ$  häufiger sind, Anzeichen für ein Gitter gibt es jedoch nicht. Bei den glatten Partikeln (mit vielen Ecken) hingegen sind die Kräfte fast ausschließlich auf die Richtungen  $60^\circ$  und  $120^\circ$  beschränkt, ein deutliches Indiz für eine hexagonale Anordnung. Wenn die

verwendeten Polygone einer Scheibe zu ähnlich werden, also sehr viele Ecken haben, nimmt die Bedeutung der Scherkräfte zu. Die Reibung reicht nicht mehr aus, das System zu stabilisieren, da ganze Teile des Haufens zur Seite rollen können. Das Ergebnis einer Simulation mit Teilchen, die 20 Ecken haben, wird in Abbildung 4.16 gezeigt. Der Bereich **2** hat wie ein Keil den Bereich **3** nach links weggeschoben. In gleicher Weise hat sich Bereich **1** gebildet. Die rechte Seite des Haufens wurde nicht durch Scherung beeinflusst, so daß sich der von Alonso [64] beschriebene Fuß gebildet hat (vgl. Gleichung 1.8). Die von Herrmann [38] beschriebenen Stufen auf der Oberfläche des Sandhaufens sind vorhanden.

Die selbstständige Anordnung der Teilchen auf einem Gitter tritt nicht mehr auf, wenn die Teilchen unterschiedliche Größen haben. In Abbildung 4.17 wird ein Haufen gezeigt, bei dem zwei Teilchensorten verwendet werden. Die kleinen Teilchen haben einen Radius von 1.15 mm und 8 Ecken die großen den Radius 3.28 mm und 24 Ecken und entsprechen damit den Experimenten von Aoki [159]. Es ist weder die Bildung einer gitterartigen Struktur zu erkennen, noch kommt es in diesem Beispiel zur Stratifikation.

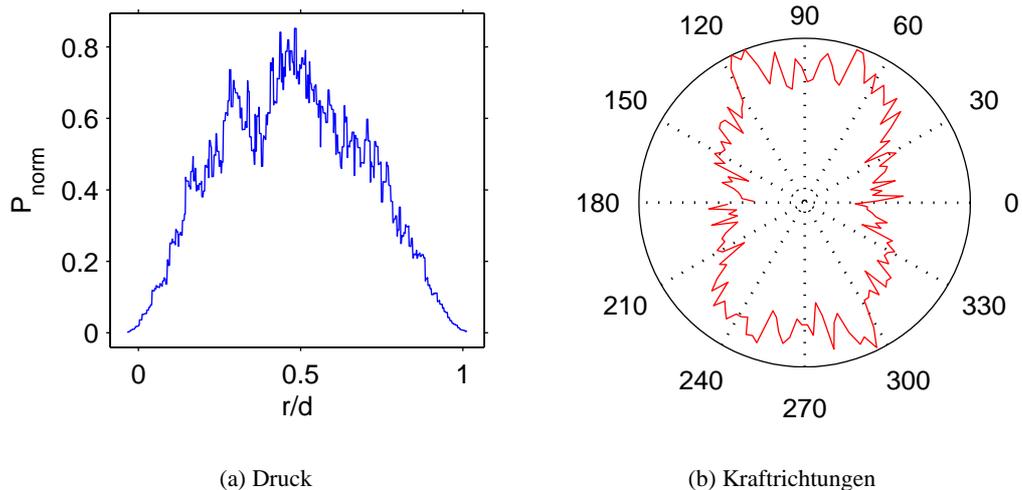
Die weiteren Simulationen wurden mit Polygonen gerechnet, deren Größe in einem Intervall gleichverteilt ist<sup>2</sup>. Die Rauigkeit der Teilchen bleibt auch bei den polydispersen Systemen wichtig. Ein Beispiel ist die Änderung des Böschungswinkels mit der Eckenzahl. Dazu wurden Teilchen mit einer Variation des Durchmessers von 20% aufgeschüttet und der Böschungswinkel bestimmt. Die Winkel der beiden Seiten der Haufen wurden gemittelt und sind in Abbildung 4.18 dargestellt. Mit zunehmender Rauigkeit der Teilchen können die Flanken immer steiler werden. Dies stimmt mit den Beobachtungen der Experimente überein.

Zur Untersuchung der Eigenschaften eines Haufens mit einem Druckminimum unter der Spitze wurden Haufen mit identischen Parametern aufgesetzt:

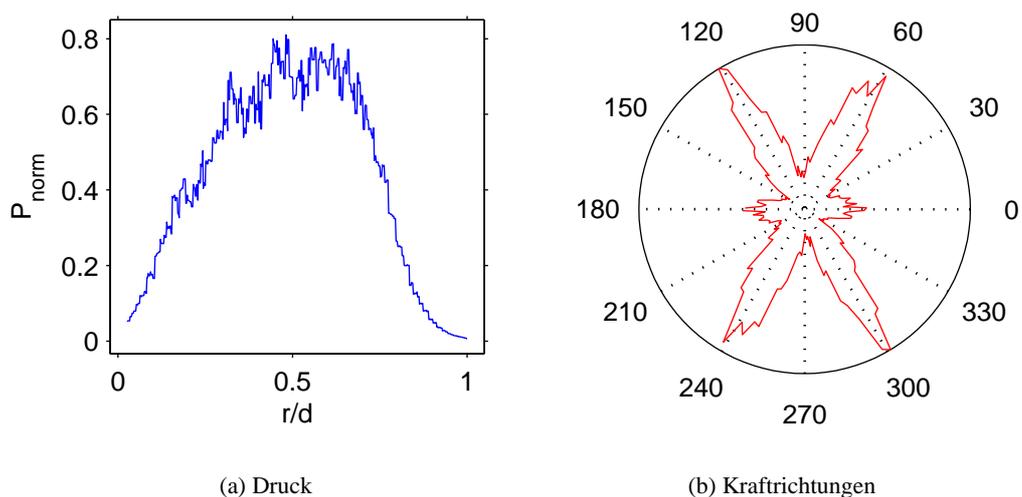
1.  $\Delta t = 2 \cdot 10^{-6} \text{ s}$
2.  $Y = 1 \cdot 10^7 \frac{\text{N}}{\text{m}}$
3.  $\gamma = 0.75$
4.  $\mu = 0.6$
5. Teilchenradius  $1.75 \text{ mm} \pm 25\%$
6. ca. 2500 Teilchen mit 7 Ecken

Die einzelnen Simulationen unterscheiden sich nur durch die Initialisierung des Zufallszahlengenerators für die Teilchengenerierung. Dies bedeutet, daß die Polygone der einzelnen Simulationen unterschiedliche Positionen der Ecken haben, aber im Mittel gleich aussehen. Eines der neun zur Mittelung verwendeten Systeme ist in Abbildung 4.19 dargestellt. Im oberen Teil ist der zum Aufschütten verwendete Trichter erkennbar. Der gemessene Böschungswinkel ist  $26.8^\circ \pm 2.0^\circ$ , die Breite des Haufens  $(0.46 \pm 0.012) \text{ m}$  und die Höhe bis zur Spitze ist  $(0.10 \pm 2 \cdot 10^{-3}) \text{ m}$ . Da die Haufen leicht abgerundet sind, kann der Böschungswinkel nicht aus der Höhe und der Breite direkt berechnet werden. Die Farbe der Teilchen gibt den Zeitpunkt an, an dem die Partikel generiert wurden. An der Anordnung der Farben ist zu erkennen, daß die älteren Bereiche des Haufens von den neueren Teilchen bedeckt werden. Ein ähnliches Bild ergab sich auch für die anderen Simulationen. Inneren des Sandhaufens ausgeht. Da der tatsächliche Böschungswinkel unter dem durch die Reibung gegebenen Wert von  $\arctan 0.65 = 33^\circ$  liegt, kann die IFS-Randbedingung von Wittmer [93] nicht erfüllt sein.

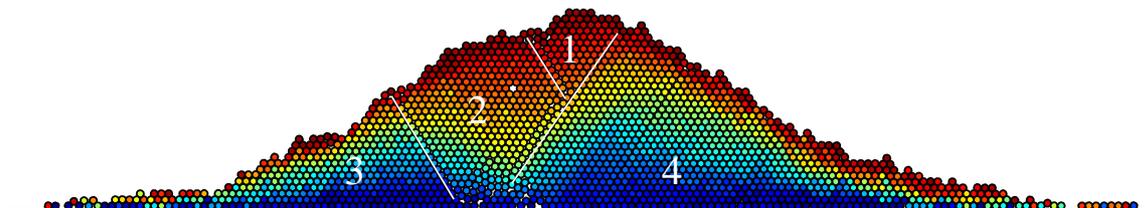
<sup>2</sup>Details zur Bestimmung der Form werden im Anhang B.1.6 beschrieben.



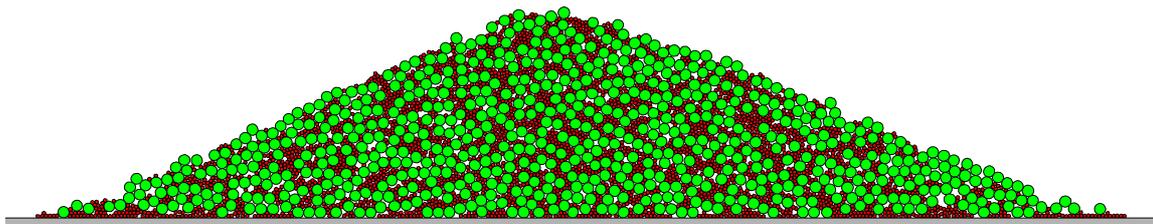
**Abbildung 4.14:** Druck auf den Untergrund und Histogramm der Krafrichtungen bei einem Haufen mit monodisperser Größenverteilung und 7 Ecken je Teilchen.



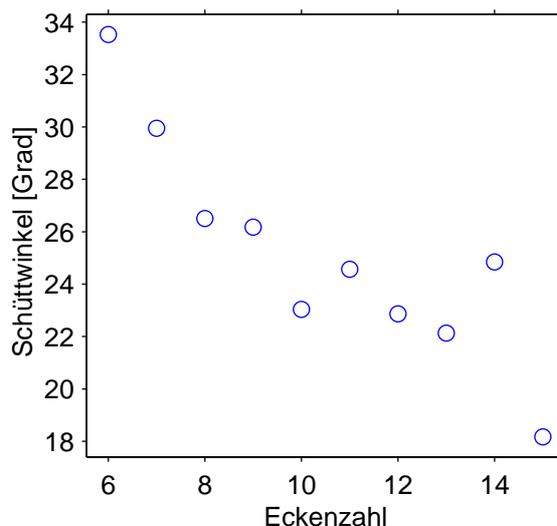
**Abbildung 4.15:** Druck auf den Untergrund und Histogramm der Krafrichtungen bei einem Haufen mit monodisperser Größenverteilung und 13 Ecken je Teilchen.



**Abbildung 4.16:** Bild eines Haufens mit runden, monodispersen Teilchen. Die Form des Haufens wird durch die Entstehung von Verschneidungen bestimmt (weiße Linien).



**Abbildung 4.17:** Bild eines Sandhaufens, bestehend aus einer bidispersen Mischung. Die Partikel sind ihrer Größe entsprechend eingefärbt.



**Abbildung 4.18:** Die Abhängigkeit des Böschungswinkels von der Eckenzahl. Die Größenverteilung der Teilchendurchmesser beträgt 20%.

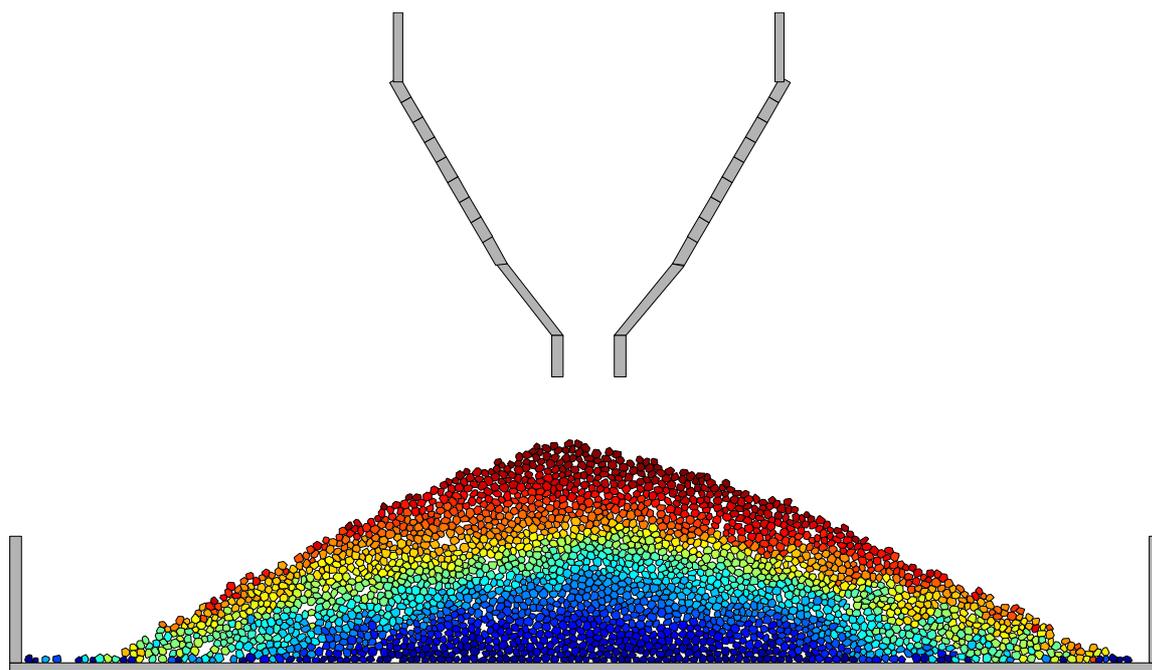
Ein typisches Kennzeichen für granulare Medien ist die Bildung von Pfaden, entlang derer Kräfte wirken. Für das Beispielsystem aus Abbildung 4.19 ist das Kraftnetzwerk in Abbildung 4.20(a) dargestellt. Die Breite der Linien und ihre Farbe ist proportional zum Betrag der jeweiligen Kraft. Man sieht links und rechts der Mitte am Boden zwei Punkte (grün markiert), in denen Kraftpfade enden; im Bereich dazwischen ist der Dip.

Mit der in Abschnitt 2.3.4 beschriebenen Methode werden nun die Spannungstensoren im Inneren der einzelnen Haufen berechnet und anschließend gemittelt. Die Werte der Diagonalelemente  $\sigma_{xx}$  und  $\sigma_{zz}$  sind in den Abbildung 4.21 und 4.22 farbkodiert<sup>3</sup>. Man sieht am Bild für  $\sigma_{zz}$  sehr gut die Existenz des Dips. Zu beachten ist, daß die Erniedrigung des Drucks unter der Spitze des Haufens nicht nur am Boden vorhanden ist, sondern auch in höheren Schichten.

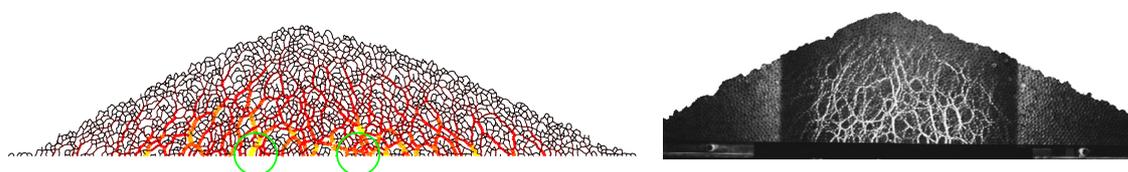
In der Abbildung 4.23 wurde  $\sigma_{zz}(x)$  für verschiedene Höhen dargestellt. Der Dip ist bis zur halben Höhe des Haufens deutlich zu erkennen. Wie in den Experimenten wurde der Druck mit  $\rho gh$  skaliert.

Für die Spannungstensoren wird der Winkel  $\Psi$  der Hauptachsen des Spannungsellipsoiden berechnet und für alle Systeme gemittelt. In Abbildung 4.24 sind die Werte als Kontourplot dargestellt,

<sup>3</sup>Die Form der Flanken in der Darstellung stammt aus der Anpassung der Testflächen an den Rand und hat keine weitere Bedeutung.



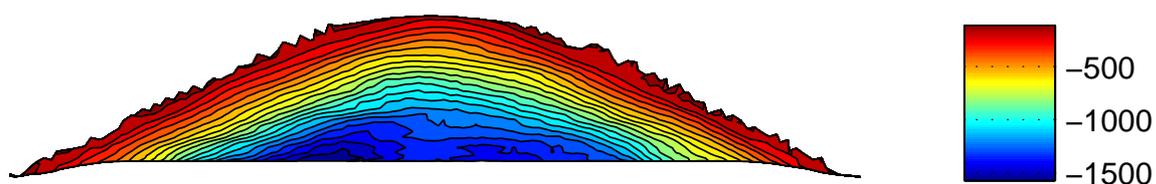
**Abbildung 4.19:** Bild eines Sandhaufens, der aus einer polydispersen Teilchenmischung aufgeschüttet wurde.



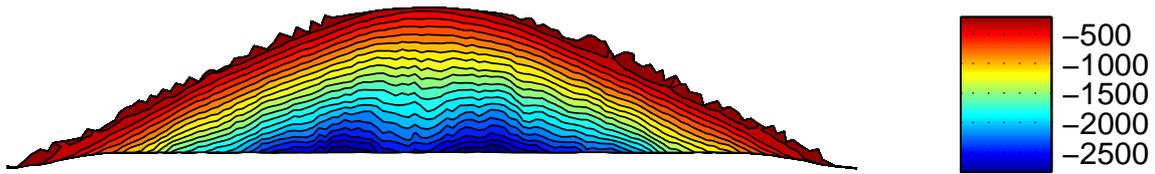
(a) Kraftnetzwerk des Systems aus Abbildung 4.19

(b) Kraftnetzwerke im Experiment (aus [85])

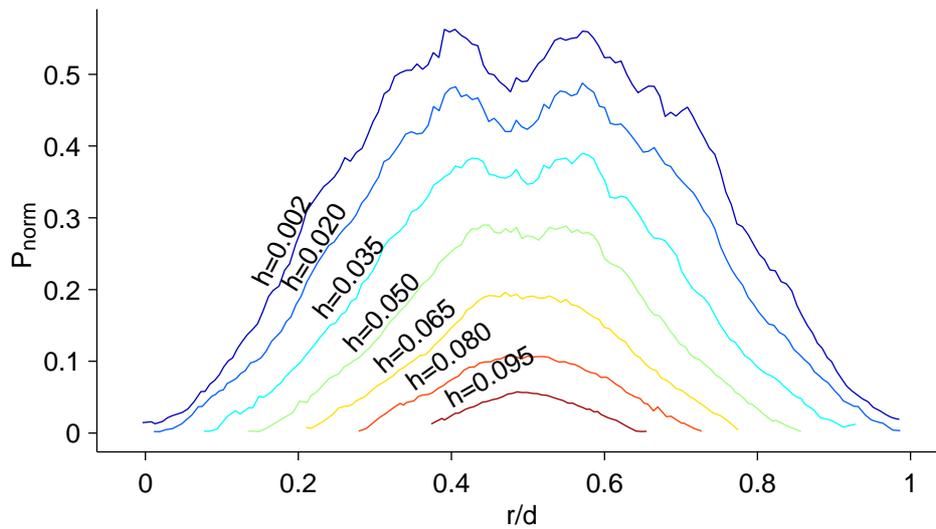
**Abbildung 4.20:** Kraftnetzwerke



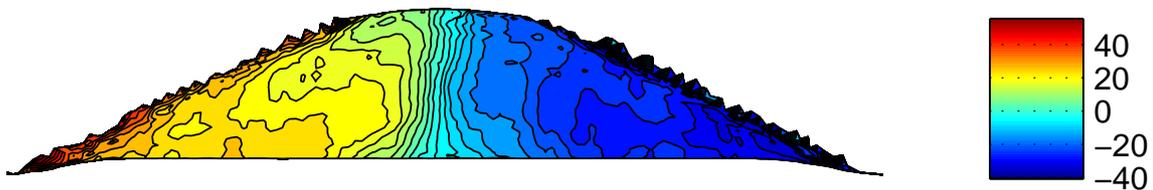
**Abbildung 4.21:** Farbkodierte Darstellung von  $\sigma_{xx}$ . Es wurde über neun Systeme gemittelt.



**Abbildung 4.22:** Farbkodierte Darstellung von  $\sigma_{zz}$ . Es wurde über neun Systeme gemittelt.



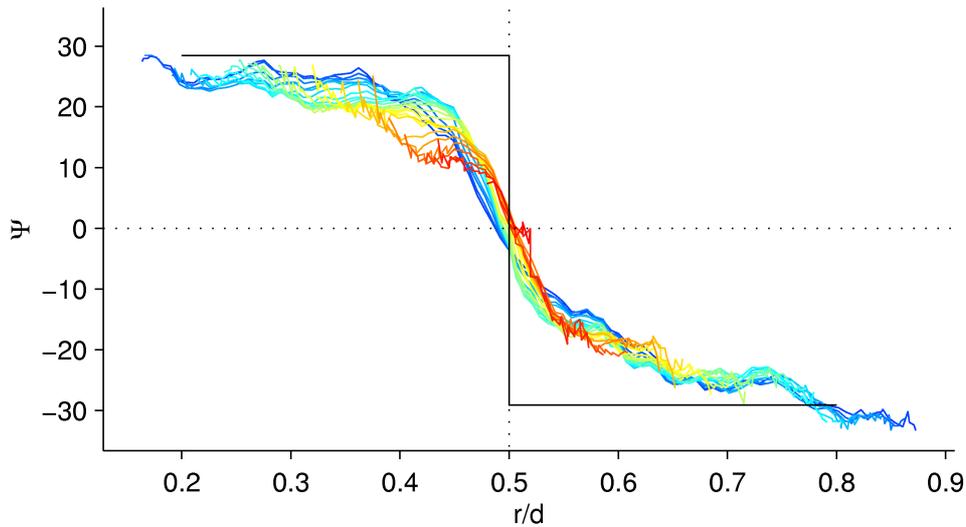
**Abbildung 4.23:** Der Druck auf einzelne Ebenen in verschiedenen Höhen der gemittelten Haufen.



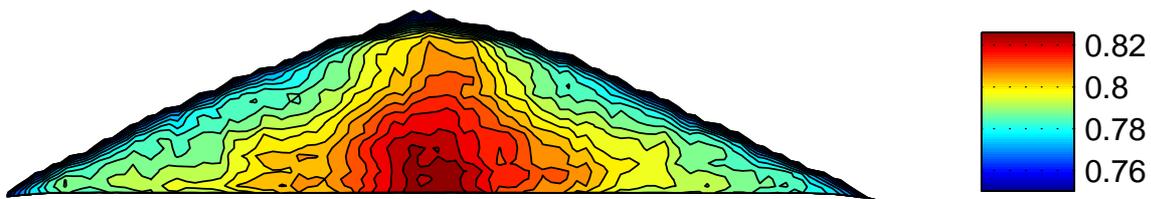
**Abbildung 4.24:** Darstellung der Winkel  $\Psi$  zwischen einer Hauptachse des Spannungstensors und der Senkrechten.

Bereiche gleicher Farbe haben den gleichen Winkel. Deutlich ist zu erkennen, daß der Winkel in den einzelnen Hälften eine bevorzugte Richtung hat. Nach Wittmer et al. [93] müßte der Winkel nahe der Oberfläche gleich  $\Psi = \frac{1}{2}(\frac{\pi}{2} - \Phi) = 28.5^\circ$  sein. Außerdem sollte  $\Psi$  entsprechend der FPA-Hypothese überall konstant sein, Bei der OSL-Hypothese muß dies zumindest im Außenbereich  $\frac{c_1}{c} < S$  gelten (siehe Abschnitt 1.3). In Abbildung 4.25 wurde  $\Psi(\frac{r}{d})$  für alle Höhen des Systems aufgetragen. Die Winkel fallen alle auf einer Kurve zusammen, die aber nur sehr ungenau von der theoretischen Vorhersage beschrieben wird.

Bei der Bestimmung der mittleren Packungsdichte in diesem und anderen Systemen ergibt sich, daß die Dichte des Haufens nicht konstant ist. Die Packungsdichte ist in Abbildung 4.26 dargestellt. Man sieht, daß der Haufen im Kernbereich dichter ist als in den Randbereichen. Auffällig ist, daß der



**Abbildung 4.25:** Darstellung der Winkels  $\Psi$  für alle Höhen des Systems. Die Kurven für die einzelnen Höhen sind in unterschiedlichen Farben (identisch mit denen in Abbildung 4.23) dargestellt. Die schwarze Linie entspricht der theoretischen Vorhersage von Wittmer für  $\mu = 0.65$ .



**Abbildung 4.26:** Gemittelte Packungsdichte des Sandhaufens.

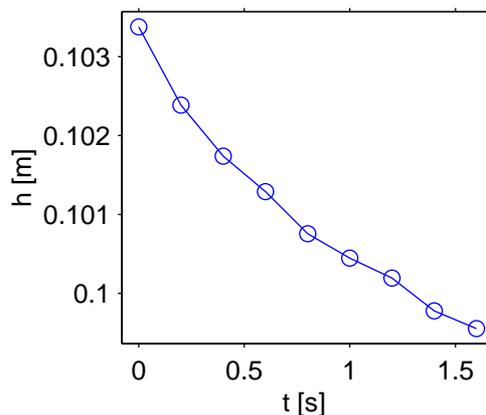
Bereich der erhöhten Dichte in der unteren Schicht gerade dort ist, wo das Druckminimum ist.

Die Gründe sind in der Entstehungsgeschichte des Haufens zu suchen. Eine Ursache kann sein, daß die Teilchen, die auf den schon entstandenen Haufen fallen, diesem wieder Energie in Form von Stößen zuführen. Dadurch kann das System weiter kompaktieren. Der andere Mechanismus kann die Volumenänderung des Granulates bei der durch die Lawinen verursachten Scherung sein. Die Voraussetzung, daß ein Haufen aus Granulat im Inneren homogen ist, ist also nicht zutreffend, wenn der Sandhaufen entsprechend aufgeschüttet wurde.

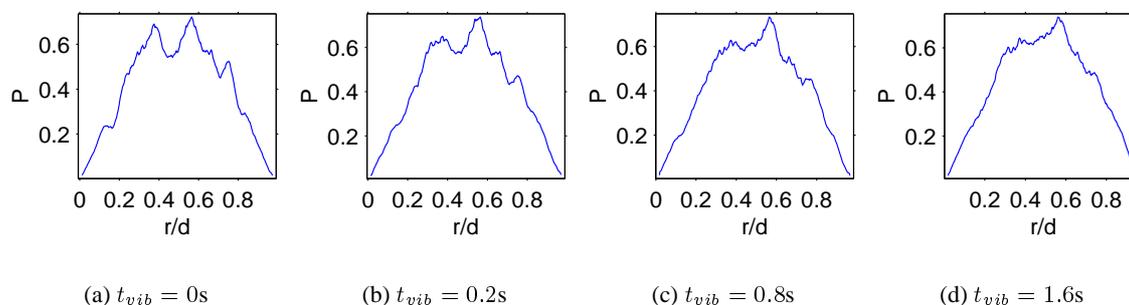
## 4.3 Beeinflussung eines Sandhaufens

### 4.3.1 Vibration

Jäger bezeichnet einen ruhenden Sandhaufen als “dead state” (vergl. Abbildung 1.1). Die “Aktivierung” des Systems erfolgt dementsprechend durch Zuführen von Energie. Um zu untersuchen, welche Auswirkungen dies auf das Druckminimum in einem Sandhaufen hat, wurden Simulationen durchgeführt, wobei die Energie dem System durch Vibration der Bodenplatte zugeführt wird. Zuerst wurde ein Sandhaufen mit `gc2d` erzeugt und die Endsituation gespeichert. Dann wurden diese Daten



**Abbildung 4.27:** Höhe eines Sandhaufens nach unterschiedlichen Vibrationszeiten  $t$



**Abbildung 4.28:** Geglättete Darstellung des Drucks auf den Boden nach unterschiedlichen Vibrationszeiten.

als Ausgangssituation für weitere Programmläufe verwendet, wobei der Boden jetzt als vibrierende Wand berechnet wurde. Die Messungen zeigen nicht einen Lauf, bei dem zu unterschiedlichen Zeitpunkten gemessen wurde, sondern verschiedene Simulationen mit gleicher Anfangsbedingung und unterschiedlichen Vibrationszeiten. Die Bodenplatte oszilliert mit 20 Hz, die Amplitude ist 0.1 mm, die Form der Schwingung läßt sich als “Klopfen” beschreiben (siehe Anhang B.1.5.2).

Der Haufen ändert durch die Vibration seine Form, die Höhe nimmt mit zunehmender Vibrationszeit ab (siehe Abbildung 4.27).

Durch die Vibrationen kommt es im Inneren des Sandhaufens zu einer Umordnung, die Dichte wird homogener und das Druckminimum nimmt ab. Ebenso werden die Fluktuationen an den Flanken des Systems kleiner. Dies stimmt mit den Beobachtungen von Aoki [158] überein, die ebenfalls das Verschwinden des Dips bei Vibration des Bodens beobachtet hat.

### 4.3.2 Externe Kontrolle des Druckminimums

In diesem Abschnitt soll die Änderung der Randbedingungen auf einen Haufen aus Granulat untersucht werden. Dazu wird in einem kleinen Bereich an der Sohle des Haufens eine zusätzliche Spannung  $\sigma_{zz}$  angesetzt. Dies geschieht dadurch, daß ein kleiner Teil des Bodens beweglich ist und solange nach unten (oben) bewegt wird, bis die gewünschte Druckerniedrigung (Druckerhöhung)

erreicht ist.

Es wurden zwei verschiedene Systeme verwendet. Zum ersten ein Sandhaufen, der einen Dip zeigt. Hier ist eine naheliegende Frage, ob der Dip kompensiert werden kann. Das zweite System ist ein schichtweise aufgeschütteter Haufen, der keinen Dip zeigt. Kann man künstlich einen Dip erzeugen, ohne den Haufen zu zerstören?

Bei der Simulation wurde in einem ersten Programmlauf von `gc2d` der gewünschte Sandhaufen aufgeschüttet, der Stempel im Boden ist noch unbeweglich. Die Breite des beweglichen Bodens wurde so gewählt, daß sie der des Dips in einem Haufen, der auf einem durchgehenden Boden aufgeschüttet wurde, entspricht. Bei den hier vorgestellten Systemen beträgt diese Breite 4.5 cm. Dann wurde der Druck  $P_0$ , der auf den Stempel wirkt, bestimmt. Die Daten dieser einzelnen Simulation dienen dann als Ausgangsbasis für viele weitere Simulationen, in denen das bewegliche Bodenelement dann so eingestellt wurde, daß ein vorgegebenes Verhältnis  $p = \frac{\Delta P}{P_0}$  erreicht wurde. Die Verschiebungen der Bodenplatte waren dabei kleiner als ein Teilchendurchmesser. Aus den fertigen Simulationen wurde dann die Verschiebung  $\Delta u$  in  $z$ -Richtung bestimmt. Positive Werte bedeuten ein Hineinpressen des Stempels in den Haufen, negative Werte ein Herausziehen.

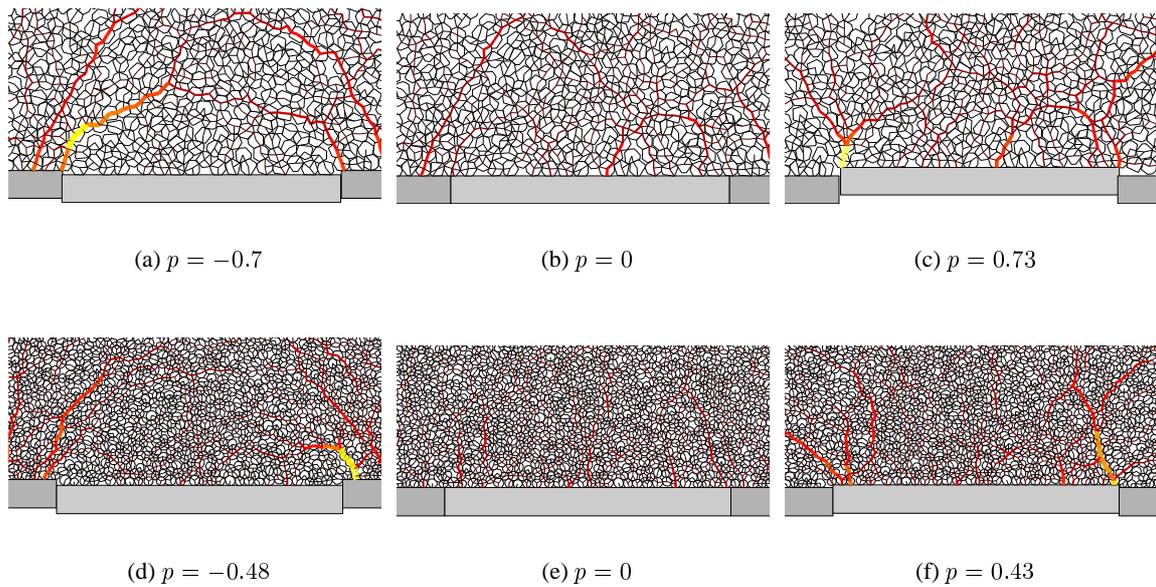
Nun die Ergebnisse der Simulation des von einer Punktquelle aufgeschütteten Haufen mit Dip: in Abbildung 4.30 ist der Druckverlauf des unbeeinflussten Systems ( $p = 0$ ) blau dargestellt. Der Graph 4.30(a) zeigt dann den Druckverlauf für den kleinsten simulierten Wert von  $p = -0.7$ . Die Gesamtkraft auf den Boden muß gleich dem Gewicht des Sandhaufens und damit konstant sein, woraus folgt, daß der Mittelwert über den Bodendruck einen festen Wert hat. Da der Stempel also weniger Kräfte aufnimmt, müssen diese von den feststehenden Teilen des Bodens aufgenommen werden. Der Druck ist dann nur in einem schmalen Bereich links und rechts des Stempels erhöht, weiter außen liegende Regionen werden nicht beeinflusst. Die Pfade, entlang derer die Kräfte zur Seite geleitet werden, sind in Abbildung 4.29(a) deutlich zu sehen, man erkennt einen Bogen. Preßt man den Boden in den Sandhaufen, läßt sich das Druckminimum kompensieren. An der Spitze in der Druckverteilung (Abbildung 4.30(b)) und am Kraftnetzwerk (Abbildung 4.29(c)) sieht man, daß der Hauptteil der Kräfte nur auf sehr wenige Teilchen ausgeübt wird.

In Abbildung 4.31 sind die Ergebnisse für einen geschichteten Haufen ohne Dip dargestellt. Es ist interessant, daß es möglich ist, auch in diesem System durch Änderung der Randbedingungen künstlich einen Dip zu erzeugen (Abbildung 4.31(a)). Ebenso wie bei dem anderen Haufen werden die Kräfte nur von der Umgebung des Stempels getragen, weiter entfernte Bereiche werden nicht beeinflusst. Am Kraftnetzwerk in Abbildung 4.29(d) sieht man, daß durch die Bewegung des Bodens ein Bogen entsteht, der die Kräfte zur Seite leitet. Für  $p = 0$ , also den unbeeinflussten Sandhaufen, sind die Pfade der Kräfte nicht so deutlich ausgeprägt und verlaufen nicht zur Seite hin. Beim Hineindrücken des Stempels wurde nur der Punkt  $p = 0.4$  erreicht. Auch hier wird der Großteil der Kräfte nur von wenigen Teilchen getragen. Wurde versucht, größere Kräfte anzuwenden, durchstieß der Stempel den Haufen.

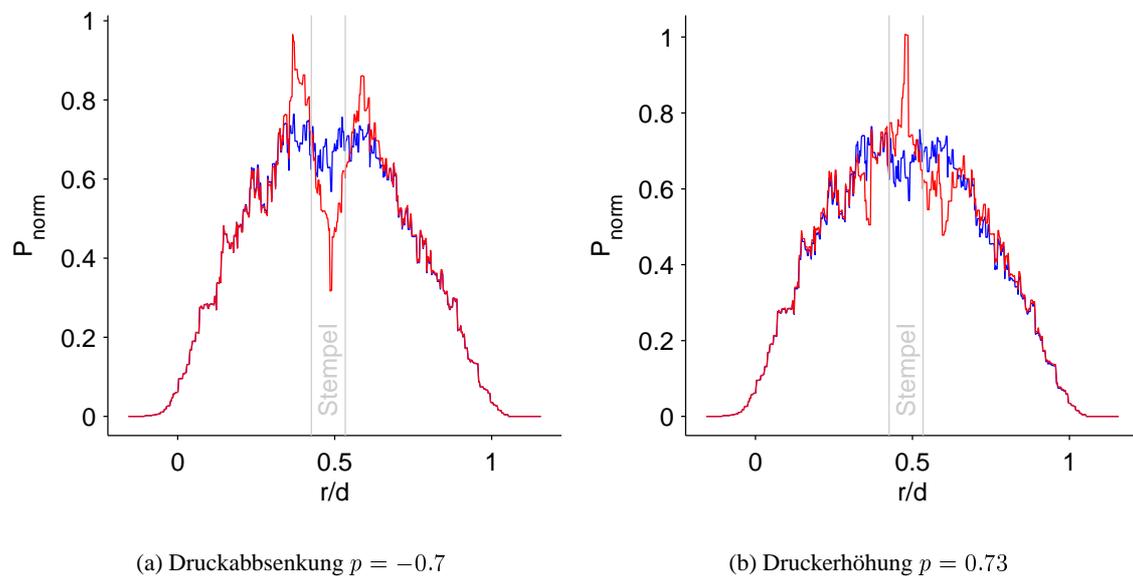
Abbildung 4.32 zeigt den Zusammenhang zwischen den Verschiebungen und den Druckänderungen. Die Kurve für die Simulation der Punktquelle ist ungefähr punktsymmetrisch zum Punkt  $(0, -0.1)$ , die für den geschichteten Haufen zum Punkt  $(0, 0)$ . Die Antwort des Systems auf die Verschiebung hängt also bei den Systemen von der Vorgeschichte ab, die ja unterschiedlich ist. Untersucht werden soll in Zukunft, ob man durch zyklisches Durchfahren der Kurven am Schluss dieselbe Kurve erhält.

### 4.3.3 Elastische Konstanten

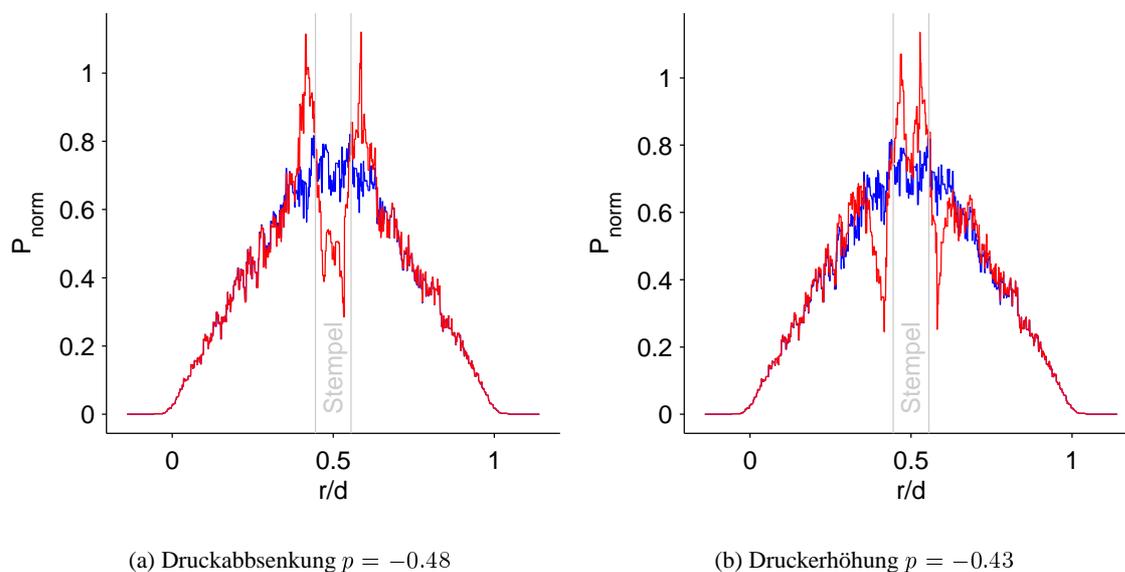
Liegt Spannungs- und den Verzerrungstensor eines Systems vor, so läßt sich für ein isotropes System daraus der Elastizitätsmodul und die Querkontraktionszahl berechnen.



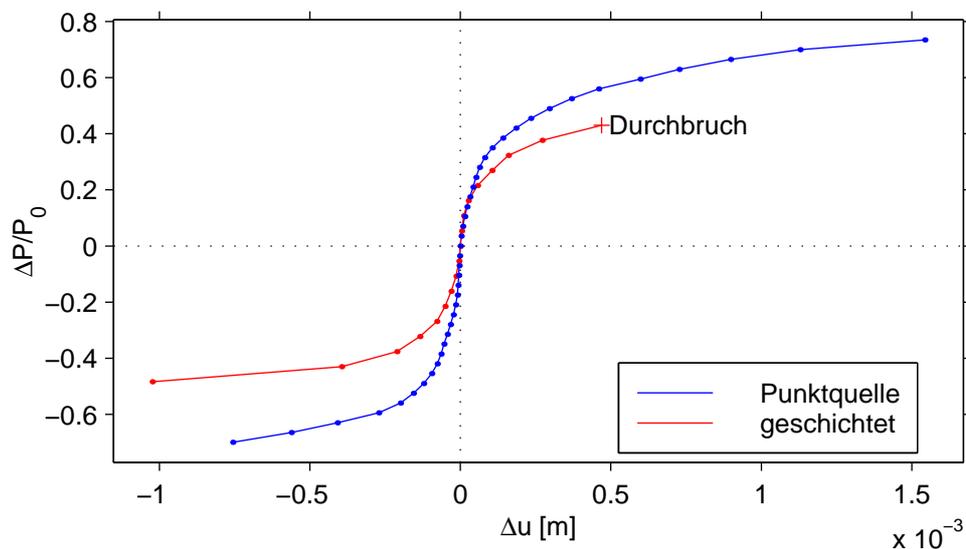
**Abbildung 4.29:** Die Kraftnetzwerke im Inneren eines Haufens mit Dip (a,b,c) und eines geschichteten Haufens ohne Druckminimum (d,e,f). Für die Darstellung der Kräfte wurde für jeden Haufen jeweils die selbe Skalierung der Farben verwendet. Zur besseren Erkennbarkeit wird der Betrag der Kräfte nicht nur durch die Linienbreite symbolisiert, sondern auch durch die Farbe. Betragmäßig grosse Kräfte sind gelb, mittlere sind rot und kleine Kräfte sind schwarz dargestellt.



**Abbildung 4.30:** Druckverteilung am Boden eines Haufens, der von einer Punktquelle aufgeschüttet wurde. Die blaue Kurve ist die Druckverteilung der Ausgangssituation  $p = 0$ , die rote Kurve zeigt den Druck nach dem Bewegung des Stempels.



**Abbildung 4.31:** Druckverteilung am Boden eines schichtweise aufgebauten Haufens. Die blaue Kurve ist die Druckverteilung der Ausgangssituation  $p = 0$ , die rote Kurve zeigt den Druck nach dem Bewegung des Stempels.



**Abbildung 4.32:** Die Abhängigkeit der Druckänderung  $p = \frac{\Delta P}{P_0}$  von der Verschiebung  $u$

Dabei gilt für den Zusammenhang zwischen Spannungstensor und Verzerrungstensor in isotropen Systemen folgende Gleichung [122]:

$$\sigma_{ik} = \frac{E}{1 + \nu} \left( u_{ik} + \frac{\nu}{1 - 2\nu} u_{ll} \delta_{ik} \right) \quad (4.1)$$

wobei  $E$  der Elastizitätsmodul (Youngmodul) und  $\nu$  der Querkontraktionskoeffizient (Poissonsche Zahl) ist. Für den zweidimensionalen Fall ergeben sich bei ebener Deformation die zwei Gleichungen

$$\sigma_{xx} = \frac{E}{(1 + \nu)(1 - 2\nu)} ((1 - \nu)u_{xx} + \nu u_{zz}) \quad (4.2)$$

$$\sigma_{zz} = \frac{E}{(1 + \nu)(1 - 2\nu)} ((1 - \nu)u_{zz} + \nu u_{xx}) \quad (4.3)$$

mit den Unbekannten  $E$  und  $\nu$ . Durch Umformen wird  $E$  eliminiert und für die Poisson Zahl ergibt sich

$$\nu = \frac{\sigma_{zz}u_{xx} - \sigma_{xx}u_{zz}}{(u_{xx} - u_{zz})(\sigma_{xx} + \sigma_{zz})}. \quad (4.4)$$

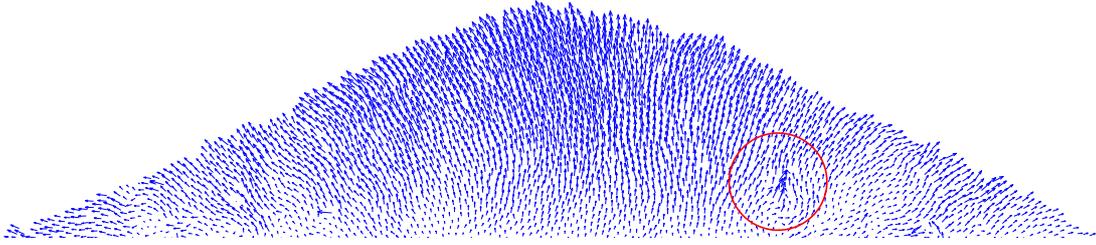
Durch Einsetzen von Gl. 4.4 in Gl. 4.2 wird dann der Elastizitätsmodul berechnet:

$$E = \sigma_{xx} \frac{(\nu + 1)(1 - 2\nu)}{u_{xx}(1 - \nu) + \nu u_{zz}} \quad (4.5)$$

Damit sollen nun die elastischen Konstanten eines Sandhaufens bestimmt werden. Allerdings ist dazu das Wissen um die Deformation des Sandhaufens unter dem Einfluß der Gravitation notwendig. Hierzu wiederum müßte man den spannungsfreien Zustand des Systems kennen. Bei einigen der regulären Sandhaufen aus Abschnitt 4.1 läßt sich ein spannungsfreier Zustand erraten. Zu Beginn der Simulation werden dazu die einzelnen Partikel auf den entsprechenden Gitterpunkten  $\mathbf{x}_i$  so plziert, daß sie ihre Nachbarn berühren, die Überlappfläche aber nur ein Punkt ist, also  $A_{ij} = 0$ . Dann wirken auf die Teilchen keine Kontaktkräfte, das System ist spannungsfrei. Im Lauf der Simulation kommt jedes Teilchen in der Position  $\mathbf{x}'_i$  zur Ruhe. Für diesen Zustand lokalen Gleichgewichts werden die Spannungstensoren berechnet. Die Verschiebung  $\mathbf{u}_i = \mathbf{x}'_i - \mathbf{x}_i$  ist definierbar.

Nicht geklärt ist aber, wie man den spannungsfreien Zustand für einen Haufen bestimmen kann, der aufgeschüttet wurde. Einen entsprechenden Anfangszustand gibt es nicht. Also muß versucht werden, aus dem Endzustand des Systems einen spannungsfreien Zustand zu bestimmen, der bei Anlegen der tatsächlich vorhandenen Spannungen in den verzerrten Zustand führt. In der Simulation ist dies erreichbar, indem man die Gravitation reduziert, bis  $g \approx 0$  gilt. Dies muß so langsam geschehen, daß der Haufen zu jedem Zeitpunkt im Gleichgewicht ist. Dabei wird dann die aus den Kontakten freiwerdende Energie durch Reibung und Dämpfung dissipiert. Numerisch ist  $g = 0$  in endlicher Zeit nicht zu erreichen, man muß bei einem hinreichend kleinen Wert für  $g'$  die Simulation stoppen. In dem hier vorgestellten Beispiel ist dies  $g' = 2.45 \cdot 10^{-3}$ . Diese Methode kann nur funktionieren, wenn es nicht zu Umlagerungen im Sandhaufen kommt. Es muß ein (nahezu) reversibler Pfad zwischen beiden Zuständen existieren. Dies bedeutet, wenn die Gravitation mehrfach reduziert und wieder auf den ursprünglichen Wert gebracht wird, muss der Endzustand gleich sein. Aufgrund der Methode bestimmt man also die negative Verschiebung.

Dabei haben sich die Partikel vom Punkt  $\mathbf{x}'_i$  mit  $g = 9.81$  zum Punkt  $\mathbf{x}_i$  mit  $g \approx 0$  bewegt. Dargestellt ist dies in Abbildung 4.33. An dieser Darstellung läßt sich erkennen, daß es durch die Änderung von  $g$  zu einigen kleinen Umordnungen im System kam.



**Abbildung 4.33:** Bewegung der einzelnen Körner bei der Reduzierung von  $g = 9.81$  auf  $g \approx 0$ . Die Pfeile deuten vom Startpunkt  $\mathbf{x}'_i$  in Richtung des Endpunkts  $\mathbf{x}_i$ . Die Pfeile wurden so skaliert, daß sie darstellbar sind. Typische Verschiebungen sind kleiner als ein Hundertstel des Teilchendurchmessers. Der rote Kreis markiert einen Bereich, in dem es zu einer Umordnung kam.

Allgemein gilt, das zwei Schritte notwendig sind, um aus den Verschiebungen der einzelnen Teilchen den Verzerrungstensor zu berechnen. Im ersten Schritt geht man von einzelnen Teilchen zu einer mittleren Verschiebung über. Dazu mittelt man die einzelnen Verschiebungsvektoren im Inneren von Testflächen (ähnlich wie in Abschnitt 2.3.4), so daß man dann die Verschiebung  $u$  auf den Punkten eines Gitters erhält. Sinnvollerweise verwendet man dazu die gleichen Testflächen wie zur Berechnung der Spannungstensoren, zusätzliche numerische Interpolationen werden im weiteren nicht notwendig.

Im zweiten Schritt wird aus der Verschiebung der Verzerrungstensor

$$u_{ik} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i} + \frac{\partial u_l}{\partial x_i} \frac{\partial u_l}{\partial x_k} \right) \quad (4.6)$$

berechnet. Da die gemischten Ableitungen hinreichend klein sind, gilt für die Diagonalelemente

$$u_{xx} = \frac{\partial u_x}{\partial x} \quad \text{und} \quad u_{zz} = \frac{\partial u_z}{\partial z}. \quad (4.7)$$

Die Ableitungen werden folgendermaßen numerisch berechnet:

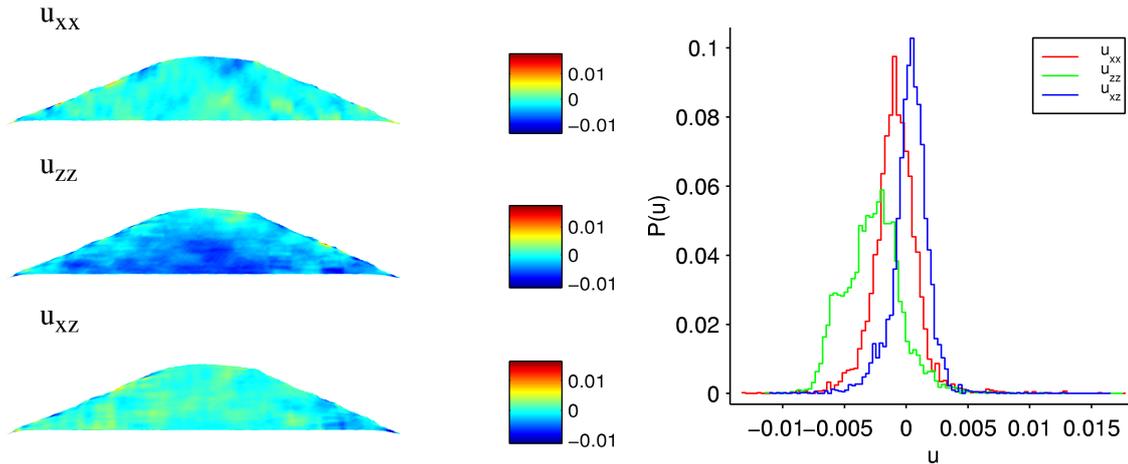
$$u_{xx}(x, z) = \frac{u_x(x + h_x, z) - u_x(x - h_x, z)}{2h_x} + \mathcal{O}(h_x^2) \quad (4.8)$$

$$u_{zz}(x, z) = \frac{u_z(x, z + h_z) - u_z(x, z - h_z)}{2h_z} + \mathcal{O}(h_z^2), \quad (4.9)$$

wobei  $h_{x,z}$  aus dem Abstand der Gitters folgt, auf dem  $u$  bestimmt wurde. Damit sind alle Voraussetzungen gegeben, um  $E$  und  $\nu$  für diese speziellen Systeme zu bestimmen.

Aus diesen Verschiebungen werden dann die Elemente des Verzerrungstensors berechnet. Dargestellt sind  $u_{xx}$  und  $u_{yy}$  in Abbildung 4.34. Es ist, wie zu erwarten, die Verzerrung in Richtung der Gravitation größer als senkrecht dazu.

Mit den Gleichungen 4.4 und 4.5 kann nun für jeden Punkt im Sandhaufen den Elastizitätsmodul und die Querkontraktionszahl berechnet werden. In Abbildung 4.36 ist  $\nu$  im Intervall  $[-1, 1]$  dargestellt. Eigentlich muß  $\nu < \frac{1}{2}$  gelten. Im Bereich der Oberfläche des Sandhaufens trifft dies jedoch nicht mehr zu. Hier kann der Sandhaufen nicht mehr als elastisches System betrachtet werden. Dies



**Abbildung 4.34:** Farbkodierte Darstellung der Elemente  $u_{xx}, u_{zz}$  und  $u_{xz}$  des Verzerrungstensor. Daneben ist deren Verteilung dargestellt.

gilt auch für die weißen Flecken im Inneren, wo es während der Änderung von  $g$  zu einer Umordnung der Partikel kam. Die Verteilung von  $\nu$  ist in Abbildung 4.35(b) dargestellt, der Mittelwert der Querkontraktionszahl ist  $\nu_{avg} = 0.446 \pm 0.137$ . Dies bedeutet, daß der Sandhaufen eine relativ geringe Kompressibilität hat. Auch bei der farbkodierten Darstellung 4.37 wurde der Youngmodul nur in einem Intervall  $[0 \dots 5 \cdot 10^6]$  dargestellt. In den Bereichen der Teilchenumlagerung und an der Oberfläche ist die Definition einer Elastizität nicht mehr sinnvoll,  $E$  springt um bis zu 12 Größenordnungen, sicher auch eine Folge der numerischen Ableitung. Aus Abbildung 4.35(a) wird der mittlere Wert des Elastizitätsmoduls zu  $E = 1,48 \cdot 10^5 \pm 1,08 \cdot 10^5$  bestimmt. Für die Partikel galt  $E_{part} = 10^7$ . Der simulierte Haufen ist damit um zwei Größenordnungen weicher als seine einzelnen Partikel.

Nimmt man einen isotropen Festkörper an, kann der Kompressionsmodul entweder ueber die Spannungen

$$u_{ii} = \frac{1}{2K} \sigma_{ii} \quad (4.10)$$

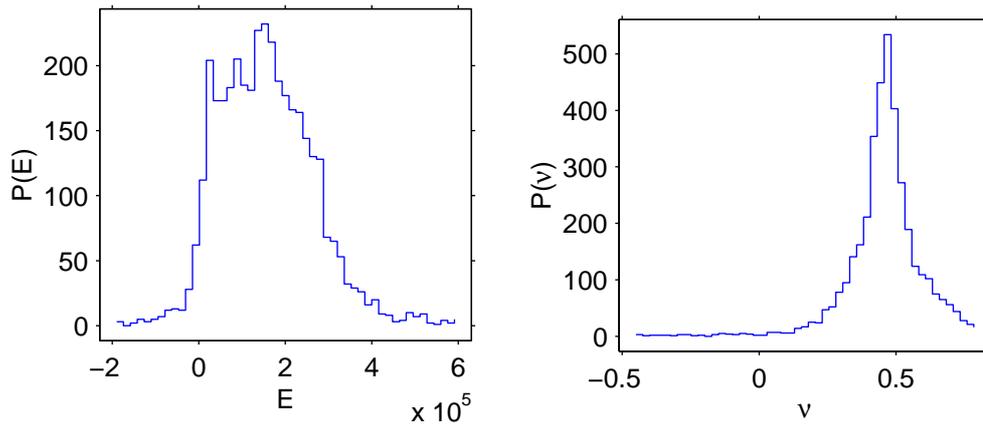
oder direkt aus  $E$  und  $\nu$

$$K = \frac{1}{2} \frac{E}{(1 + \nu)(1 - 2\nu)} \quad (4.11)$$

bestimmt werden. Das Ergebnis ist in Abbildung 4.38 dargestellt. Offensichtlich ist der Haufen im Inneren härter als im Außenbereich, außerdem ist der Bereich des Dips deutlich zu erkennen. Im Bereich des Minimums ist das Granulat weicher als in den Bereichen der Maxima. Dies ist ein deutlicher Hinweis auf die Bildung eines großen Bogens, der die Kräfte von der Mitte weg zur Seite hin ableitet.

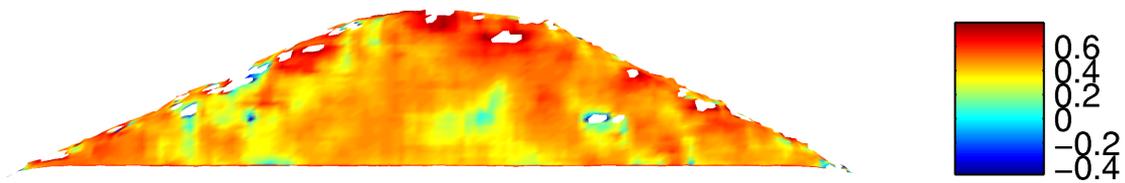
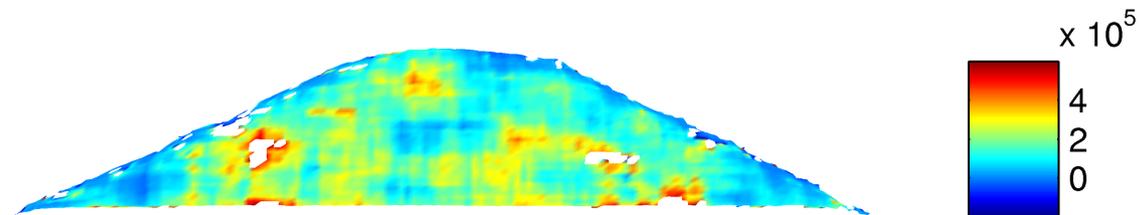
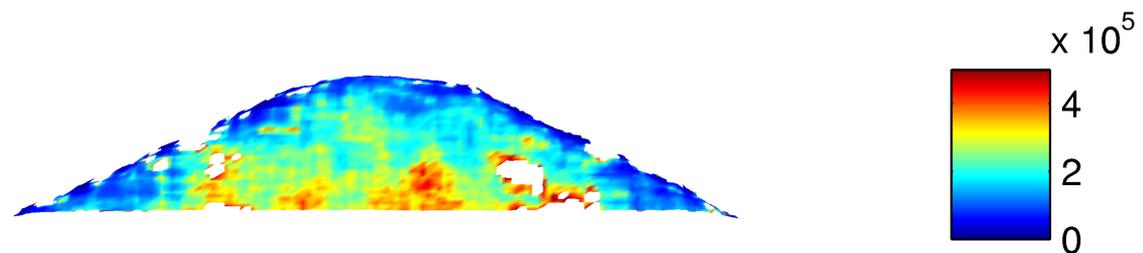
## 4.4 Eigenschaften kohäsiver Aufschüttungen

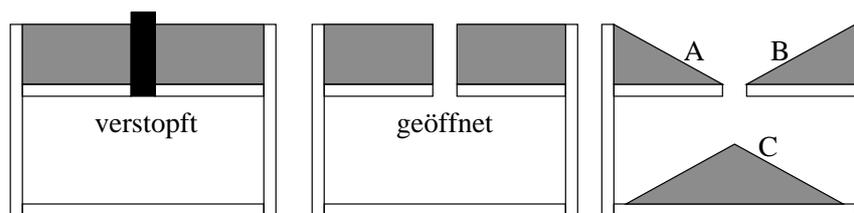
Die durch Gleichung 2.28 beschriebene Modellierung für die Kohäsionskräfte zwischen zwei Teilchen soll nun in der Simulation angewandt werden. Zuerst werden die Simulationen mit Experimen-



(a) Verteilung des Elastizitätsmoduls

(b) Verteilung der Poissonzahl

**Abbildung 4.35:** Verteilungen der elastischen Konstanten.**Abbildung 4.36:** Farbkodierte Darstellung der Poissonzahl  $\nu$ .**Abbildung 4.37:** Farbkodierte Darstellung des Youngmoduls  $E$ .**Abbildung 4.38:** Der Kompressionsmodul  $K$  in farbkodierter Darstellung.



**Abbildung 4.39:** Experimentiermethode, um den Böschungswinkel von kohäsiven Granulaten zu messen.

ten verglichen, dann werden experimentell nur schwer zugängliche Ergebnisse wie Korrelationszeiten und Koordinationszahlen vorgestellt.

In den letzten Jahren hat das Interesse an den Effekten zugenommen, die durch Kohäsion ausgelöst werden [19, 21, 49, 160]. Beispiele dafür sind die langsame Zunahme der statischen Reibung bei längeren Kontakten von Sandkörnern [19], Kristalloberflächen [161] und Felsen [20], die durch die Luftfeuchtigkeit verursacht wird. Gleiches gilt auch für die Experimente von Albert [21]. Die beigefügten Ölmengen führen nur zu einem sehr dünnen, 20-25 nm dicken Film auf der Kugeloberfläche. Die experimentellen Untersuchungen legen dabei nahe, daß sich mit zunehmender Kohäsion eine Änderung des Flußverhaltens ergibt. Bei geringer Kohäsion bewegen sich die Körner einzeln, bei großer Kohäsion bewegen sich die Teilchen in kleinen Clustern. Dieser Aspekt soll im weiteren beleuchtet werden.

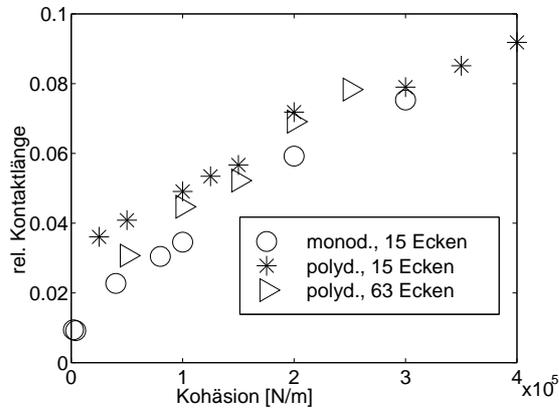
Albert et al. [21, 49, 160] maßen den Böschungswinkel mit der “draining-crater”-Methode [162]. Dabei wird ein Gefäß, welches eine verschlossene Öffnung am Boden hat, mit dem Granulat gefüllt. Anschließend wird das Loch im Boden geöffnet und das Material kann frei herausfließen (siehe Abbildung 4.39). Sobald der Fluß zum Stillstand gekommen ist, wird der Böschungswinkel gemessen. Dabei wurde ein linearer Zusammenhang zwischen dem Böschungswinkel und der Kohäsion gefunden.

Die Größe der Box entspricht in den Experimenten [21, 49, 160] etwa 80-250 Teilchendurchmessern, in der Simulation sind es etwa 160-200. Die Größe des Ausflusses war 12-25 Teilchendurchmesser, in der Simulation 12-40.

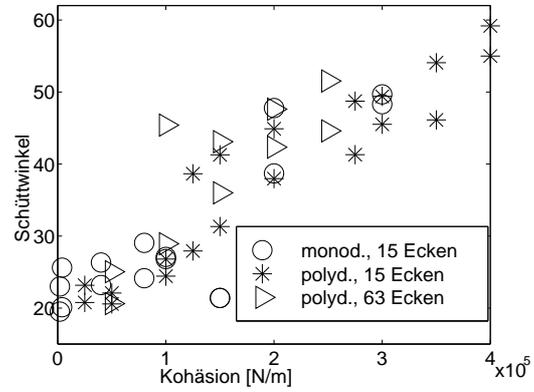
Eine Simulationsreihe wurde mit monodispersen, regulären Polygonen mit 15 Ecken durchgeführt, eine mit einer polydispersen Mischung mit einer Gleichverteilung der Radien  $r$  im Bereich  $[0.75 \cdot r, 1.25 \cdot r]$ . Die Reibungskoeffizienten für alle Simulationen waren  $\mu = 0.6$ , der Youngmodul  $E = 10^7 \frac{\text{N}}{\text{m}}$  und der Zeitschritt  $\Delta t = 0.2 \cdot 10^{-5} \text{s}$ . Um den Einfluß der Teilchenform zu untersuchen, wurde eine weitere Reihe mit der gleichen Größenverteilung, aber mit 63 Ecken gerechnet. Alle Messungen geben konsistente Daten für mittlere bis große Werte der Kohäsion. Für kleine Werte hingegen haben die monodispersen Teilchen die Tendenz, sich in Dreiecksgittern anzuordnen. Dies dominiert dann die Physik des ganzen Systems.

In den Simulationen nehmen die Kontaktfläche und die Kontaktlänge (siehe Abbildung 4.40) linear mit der Größe des Kohäsionsparameters  $k_{\text{coh}}$  zu. Eine Kontaktlänge von etwa 0.01–0.1 Teilchendurchmessern entspricht einer Eindringtiefe von etwa  $2.5 \cdot 10^{-3}$  bis  $2.5 \cdot 10^{-4}$  Teilchendurchmessern, also 3-4 Größenordnungen kleiner als die Teilchen selbst. Dies paßt mit den von Albert [21] in den Experimenten gefundenen Dicken der Ölschichten zusammen, die ca. 4 Größenordnungen kleiner sind als die Teilchengröße.

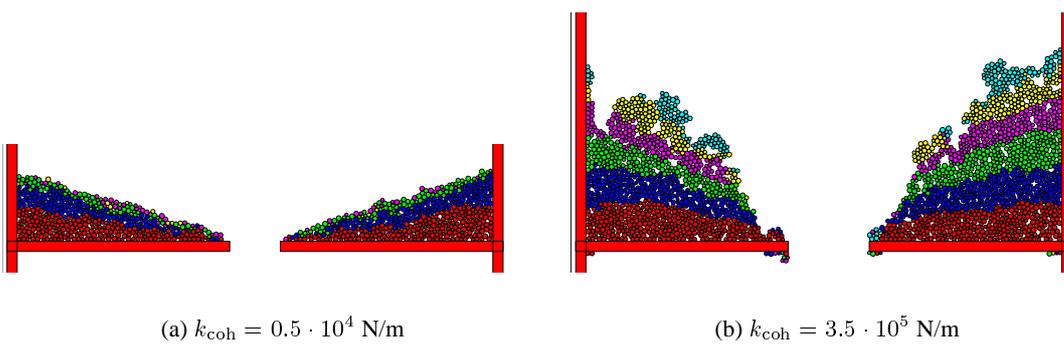
Abbildung 4.41 zeigt den Böschungswinkel für die Bereiche A und B in Bild 4.39. Gemessen wurde mit der zu Gleichung 2.42. führenden Methode. Im Bereich zwischen  $k_{\text{coh}} = 0.5 \cdot 10^5 \frac{\text{N}}{\text{m}}$  und  $k_{\text{coh}} = 2.5 \cdot 10^5 \frac{\text{N}}{\text{m}}$  ist der Böschungswinkel praktisch linear abhängig von der Kohäsion. Im Bereich für



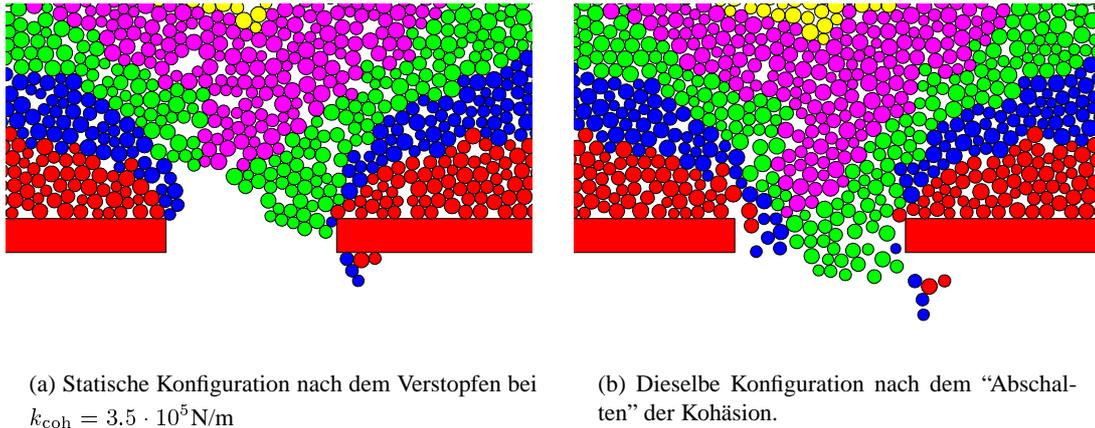
**Abbildung 4.40:** Kontaktlänge zweier kollidierender Teilchen im Verhältnis zum Teilchendurchmesser.



**Abbildung 4.41:** Der Böschungswinkel in den Bereichen A und B aus Abbildung 4.39 in Abhängigkeit vom Kohäsionsparameter  $k_{\text{coh}}$ .



**Abbildung 4.42:** Die Oberfläche des Granulates bei verschiedenen Werten für  $k_{\text{coh}}$ .



**Abbildung 4.43:** Der Einfluss der Kohäsion auf das Verstopfen eines Trichter kann in der Simulation durch Änderung des Kohäsionsparamters gezeigt werden.

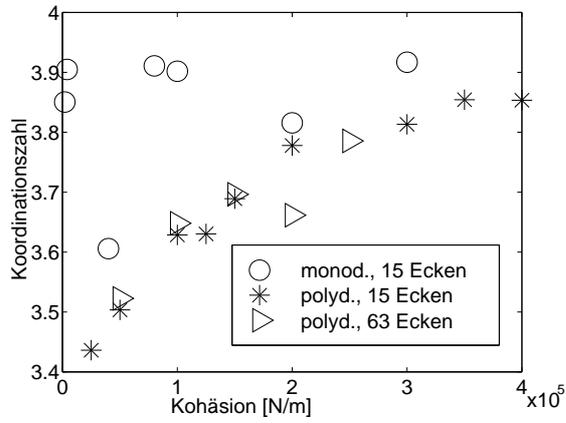
$k_{\text{coh}} < 0.5 \cdot 10^5 \frac{\text{N}}{\text{m}}$  ist der Böschungswinkel konstant. Dabei ist der berechnete Böschungswinkel für verschwindende Kohäsion in Übereinstimmung mit den Messungen von Hornbaker [49] von  $22^\circ$ - $23^\circ$  für trockene Granulate. Wenn man dann die Daten der Simulation an die experimentellen Werte anfitzt, entspricht eine Schicht von 1 nm Öl einer Zunahme von  $k_{\text{coh}}$  um  $0.05 \frac{\text{N}}{\text{m}}$ .

Der Böschungswinkel im Bereich C ist kleiner als in den Bereichen A und B. Dies hängt wahrscheinlich damit zusammen, daß die aus dem Ausfluß kommenden Teilchen eine höhere kinetische Energie haben, die dissipiert werden muß. Dies führt zu einem Auseinanderlaufen des Haufens in diesem Bereich. Gleichzeitig ist die Oberfläche in diesem Bereich aber für größere Werte von  $k_{\text{coh}}$  sehr zerklüftet, eine Bestimmung des Böschungswinkels für größere Kohäsion ist daher sehr unzuverlässig.

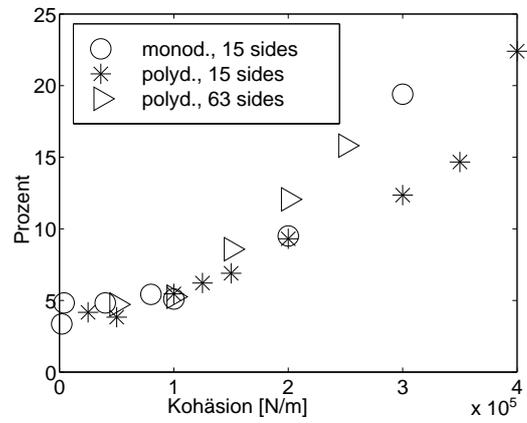
Die Tendenz des fließenden Granulates, die Auslaßöffnung zu verstopfen, nimmt mit der Kohäsion zu. Ist diese groß genug ( $k_{\text{coh}} > 2 \cdot 10^5 \frac{\text{N}}{\text{m}}$ ), so verstopft eine Öffnung der Größe von 15-20 Teilchendurchmessern sehr leicht. Dabei ist die Kohäsion stark genug, die Teilchen auch an der Unterseite des Behälters haften zu lassen. In Abbildung 4.43(a) sieht man, daß es einige Teilchen gibt, die ohne Kohäsion sicher noch fallen würden, aber es stellt sich nun die Frage, ob die Kohäsionskräfte die Bildung der Verstopfung nur fördern, oder ob sie diese auch aktiv tragen. Hier zeigt sich einer der großen Vorteile der Simulation, denn man kann die Kohäsion einfach auf Null setzen. In der Simulation beginnt das Material wieder zu fließen. Dies ist ein erster Hinweis auf die Bildung von Gruppen aus mehreren Teilchen. Der Durchmesser der Auslaßöffnung ist dann für diese Cluster nicht groß genug, er verstopft. Für hinreichend große Öffnungen (20-25 Teilchendurchmesser) wurde kein Verstopfen beobachtet.

Nachdem die bisherigen Daten mit den Experimenten vergleichbar waren, sollen nun einige Größen dargestellt werden, die nur über die Simulation auf Teilchenebene zugänglich sind.

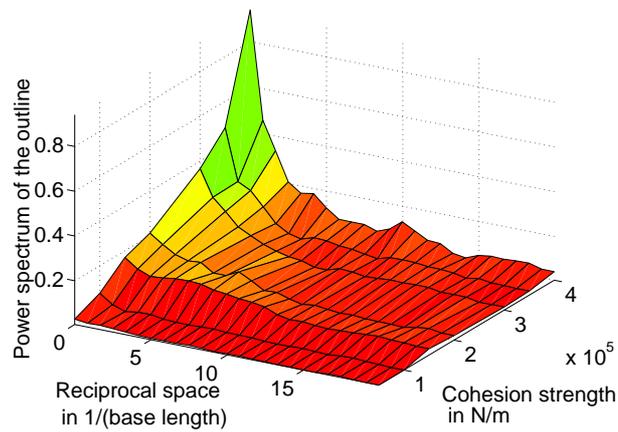
Bei der Auswertung der Koordinationszahlen muß man die Systeme getrennt betrachten. Für monodisperse Systeme sieht man in Abbildung 4.44, daß die Koordinationszahl meist sehr nahe bei 4 ist. Der Grund dafür ist, daß die Teilchen sich in diesen Systemen auf einem Dreiecksgitter anordnen, aber mit den Teilchen links und rechts im allgemeinen keinen Kontakt haben. Dies wird durch das Kraftnetzwerk aus Abbildung 4.5(b) bestätigt. Größere Abweichungen sind aber möglich. Für die polydispersen Mischungen ist die Koordinationszahl proportional zur Kohäsion. Aus den Daten der Kraftnetzwerke läßt sich aber nicht folgern, ob dieser Anstieg für die Stabilität des Haufens notwendig ist. Es wäre auch möglich, daß viele Teilchen Kontakte haben, die aber keine statische Last mehr



**Abbildung 4.44:** Die Koordinationszahl gemittelt für die Bereich A, B und C.



**Abbildung 4.45:** Dargestellt ist der Anteil der Kontakte, die zu Beginn der Simulation im oberen Bereich und zum Ende im Bereich C noch existieren.



**Abbildung 4.46:** Spektrum der Oberflächen der Regionen A und B. Die Einheiten im k-Raum sind angegeben in  $\frac{1}{l}$ , wobei  $l$  die Basislänge der Regionen A, B ist.

tragen und sich im kohäsionslosen Fall bereits gelöst hätten.

Um das Auftreten von sich bewegenden Clustern zu untersuchen, wurde folgende Methode angewandt: es wurden zu dem Zeitpunkt, als die Öffnung des Behälters noch verschlossen war, für jedes Teilchen bestimmt, welche Nachbarn es hat. Nachdem das System nach Öffnen des Stopfens wieder zum Stillstand gekommen ist, wurden im Bereich C wieder die Nachbarschaften bestimmt. Dann wurde ausgewertet, wieviele dieser Nachbarschaften schon zu Beginn bestanden. Diese Daten sind in Abbildung 4.45 dargestellt. Die Zahl der Kontakte, die erhalten bleiben, nimmt um den Faktor 4-5 gegenüber dem kohäsionsfreien System zu. Dies legt den Schluß nahe, daß sich nicht mehr einzelne Teilchen bewegen, sondern größere Cluster. Daraus folgt dann auch, daß die Durchmischung im nichtkohäsiven Fall stärker ist als für ein System mit Kohäsion.

In der Verfahrenstechnik wird die Kohäsion und die Oberflächenrauigkeit der Teilchen in einem Parameter zusammengefaßt. Nach Carr [163] hängen der Böschungswinkel und dieser Wert nicht zusammen (siehe auch [164]). Die Tatsache, daß die statische Reibung in einem granularen System von der Teilchenrauigkeit unabhängig sein kann, findet sich auch schon im Lehrbuch von Rabinowicz [125]. Zusammengefaßt bedeutet dies, daß Reibung und Rauigkeit/Kohäsion unterschiedliche Eigenschaften sind.

Die Modellierung von Reibung durch die Bildung von "Polymeren", wie sie Buchholtz und Pöschel vorschlagen [48], kann dann im Hinblick auf diese Daten auch als die Simulation von Kohäsion interpretiert werden.

Um die Oberfläche des Systems genauer zu untersuchen, wurde dessen Rauigkeit bestimmt. Dazu wurde zuerst die Abweichung der Kontur von der durch den Böschungswinkel (Gleichung 2.42) gegebenen Geraden bestimmt. Von diesen Daten wurde dann das Powerspektrum berechnet. Dabei wurden dann Region A und B gemittelt. In Abbildung 4.46 sieht man, daß die Rauigkeit auf allen Größenskalen zunimmt. Dabei nimmt die Rauigkeit besonders stark für sehr kleine Wellenzahlen zu. Dies kann so interpretiert werden, daß es bei zunehmender Kohäsion größere Cluster gibt, die dann an der Oberfläche zu den Nachbarclustern hin aufbrechen (siehe Abbildung 4.42(b)). Andererseits wird aber auch die Oberfläche der einzelnen Cluster rauher, so daß die Beiträge für grössere Wellenzahlen nicht verschwinden.

Zusammenfassend kann man sagen, das Systeme mit geringer Kohäsion sich nicht deutlich anders verhalten als Systeme ohne Kohäsion. Bei größerer Kohäsion allerdings gibt es deutliche Veränderungen des Verhaltens in Bezug auf Clusterbildung und Oberflächenstruktur.



## Kapitel 5

# Zusammenfassung und Ausblick

Was ich bisher getan hatte  
war ein Bruchteil von gar nichts.  
Es war alles nur Sand,  
und der leiseste Wind würde es fortwehen.  
Paul Auster *Hinter verschlossenen Türen*

In dieser Arbeit wurde ein Modell zur Simulation von zweidimensionalen Granulaten vorgestellt. Mit diesem Modell kann die Statik und Dynamik von verschiedenartigen Systemen untersucht werden. Die einzelnen Partikel wurden dabei durch konvexe Polygone beschrieben. Somit ist man in der Lage, sowohl besondere Teilchenformen (Stäbchen, ellipsenartige Teilchen etc.) zu untersuchen, als auch die Oberflächenrauigkeit (Eckenzahl) zu variieren. Die Kraftgesetze wurden so modelliert, daß neben der elastischen Wechselwirkung mit Reibung auch Kohäsion berücksichtigt werden kann. Das Simulationsprogramm kann zusätzlich die Größenänderung aufgrund von Temperatureffekten berechnen. Damit konnten verschiedenste Systeme untersucht werden. Ausführlich betrachtet wurden Granulate in einer Röhre und Sandhaufen.

Zunächst wird die *Discrete-Element-Methode* vorgestellt, die die Basis des Programms bildet. Nach Darstellung der zur Simulation notwendigen Teilcheneigenschaften wird deren Wechselwirkung bei einer Kollision beschrieben. Die vorgestellten Kraftgesetze beschreiben eine elastische Wechselwirkung, Dämpfung in Normalen- und Tangentialrichtung, Reibung und Kohäsion. Dann wird erläutert, wie man die nach einem Programmlauf verfügbaren Daten, ähnlich den Meßwerten eines Experiments, auswerten kann. Interessante Informationen sind beispielsweise die lokale Dichte oder die Spannungstensoren.

Bei einer Simulation müssen verschiedene Aspekte der *Numerik* berücksichtigt werden. Nur so wird sichergestellt, daß man einen effektiven und zuverlässigen Code entwickelt. Es werden spezielle Algorithmen und deren Parallelisierbarkeit gezeigt, die die Untersuchung größerer und damit aussagekräftigerer Systeme erlauben. Zum Beweis der Zuverlässigkeit und der Stabilität des Gesamtsystems werden die Ergebnisse verschiedenster Testsysteme gezeigt.

Im Anschluß daran werden das Problem der *Schüttgutsäule* und die klassische Lösung von Janssen vorgestellt. Diese Lösung wird auf erweiterte Wandformen verallgemeinert. Die Lösungen für drei spezielle Systeme werden vorgestellt und entsprechende Ergebnisse von Simulationen gezeigt.

Der nächste Abschnitt ist der Untersuchung des *Dips* von Sandhaufen gewidmet. Zuerst werden regelmäßige, konstruierte Systeme gezeigt, mit bekannten Lösungen verglichen und bewertet. Die

Simulationen geschichteter Systeme zeigen mit dem Experiment vergleichbare Ergebnisse. Es wird die Homogenität der Dichte und die Ausrichtung der Spannungstensoren beschrieben. Anschließend werden die Ergebnisse für Haufen gezeigt, die von einer punktförmigen Quelle aufgeschüttet wurden. Diese Systeme zeigen ein Druckminimum unter der Spitze des Sandhaufens. Die Winkel der Hauptachsen der Spannungstensoren werden mit den theoretischen Vorhersagen von Wittmer et al. verglichen. Es wurde die Dichte berechnet, die, anders als bisher angenommen, nicht homogen ist. Vielmehr weisen Haufen, die einen Dip haben, einen kompakten Kern auf. Dabei entsteht die instabile Situation, daß der Druck in Bereichen höherer Dichte geringer ist. Dieser Effekt legt die Idee nahe, daß die Dip-Entstehung mit einer Instabilität verknüpft ist.

Der Einfluß von Vibrationen auf die Druckverteilung sowie die Änderung der Randbedingungen des Sandhaufens werden beschrieben. Dann wird an einem Beispielsystem gezeigt, wie in einer Simulation die elastischen Konstanten eines granularen Systems bestimmt werden können. Dazu wird die Gravitation adiabatisch auf Null reduziert, so daß man einen spannungsfreien Referenzzustand hat. Die Untersuchung der Einflüsse der Kohäsion schließen das Kapitel ab.

Im Anhang findet sich eine detaillierte Darstellung des Simulationssystems `gc2d`. Eine Beschreibungssprache für granulare Systeme wird vorgestellt. Die Formate der Ausgabedateien werden genau beschrieben, so daß zukünftige Anpassungen und Erweiterungen durch Dritte komfortabel möglich sind.

Insgesamt wurde gezeigt, daß die Simulationen unter Berücksichtigung der mikroskopischen Effekte wie Teilchenform, Reibung und Kohäsion einen weiten Bereich an makroskopischen Effekten beschreiben kann. Dabei wird nicht nur die Dynamik, sondern auch die Statik von granularen Systemen untersucht. Eine systematische Abweichung der zweidimensionalen Beschreibung von den Experimenten wurde nicht gefunden.

Die Möglichkeiten für die zukünftige Nutzung der vorgestellten Methoden sind vielfältig. Die Untersuchungen der elastischen Konstanten eines Granulates sollen erweitert und die Entstehung und Einflüsse des kompakten Kerns untersucht werden. Die Untersuchung des Kompaktionsverhaltens von Granulaten in vibrierten Schüttgutsäulen ist in Vorbereitung, eine entsprechende Zusammenarbeit ist im Entstehen. Für die Simulationen der Kohäsion ist geplant, die Dauer der Kontakte zu bestimmen, damit genauere Aussagen über die Bildung der Cluster getroffen werden können. Bereits in der Entwicklung ist die Anwendung für die Simulation der Abnutzung des Schotteruntergrundes von Hochgeschwindigkeitsbahnstrecken. Die breite Anwendbarkeit des gezeigten Systems wird die Untersuchung von verschiedenen Systemen ermöglichen. Es kann den Zugang zu experimentell schwer zugänglichen Informationen ermöglichen und die Grundlage für zukünftige Theorien liefern.

## Anhang A

# Namenskonventionen bei Dateinamen

Die Steuerung einer Simulation mit `gc2d` erfolgt fast ausschließlich durch Dateien. Sowohl die Beschreibung des zu berechnenden Systems als auch die Simulationsergebnisse werden in einzelnen Dateien gespeichert. Insgesamt gibt es etwa 10 verschiedene Dateitypen, die sich alle durch ihre Funktion und ihr Dateiformat unterscheiden. Jede Ein- und Ausgabedatei hat eine eigene Endung, durch den ihr Typ bestimmt wird. Zusätzlich enthalten die Dateinamen einiger Ausgabedateien noch eine fortlaufende Nummer. Dies bedeutet dann, daß die Information mehrfach, zu verschiedenen Zeitschritten der Simulation, abgespeichert wurde. Folgende Dateitypen sind dem Programm bekannt:

<code>.gsi</code>	<b>Granular Setup Input</b>	Anh. B.1
Die Eingabedatei mit dieser Endung enthält die Beschreibung des zu simulierenden Systems. Dies ist normalerweise die einzige Datei, die der Nutzer von <code>gc2d</code> erstellt.		
<code>.gci</code>	<b>Granular Collision Input</b>	Anh. B.2
Diese Eingabedatei enthält die Beschreibung von Kollisionen des zu simulierenden Systems. Diese Datei wird normalerweise nicht per Hand, sondern aus einer <code>gco</code> -Datei erstellt.		
<code>.gso</code>	<b>Granular Setup Output</b>	Anh. C.1
Diese Ausgabedatei enthält eine von <code>gc2d</code> generierte Beschreibung des simulierten Systems. Diese Datei kann wiederum als Eingabedatei für weitere Programmläufe verwendet werden.		
<code>.gco</code>	<b>Granular Collision Output</b>	Anh. C.1
Diese Ausgabedatei enthält eine von <code>gc2d</code> generierte Beschreibung der Kollisionen des simulierten Systems. Diese Datei kann wiederum als Eingabedatei für weitere Programmläufe verwendet werden.		
<code>.ggf</code>	<b>Granular Geometry File</b>	Anh. C.2
In dieser Ausgabedatei ist die Beschreibung der Geometrie jedes in der Simulation verwendeten Polygone enthalten.		
<code>.gff</code>	<b>Granular Force File</b>	Anh. C.3
In dieser Ausgabedatei ist die Beschreibung aller wirkenden Kräfte zu einem bestimmten Zeitpunkt enthalten.		
<code>.gpf</code>	<b>Granular Position File</b>	Anh. C.4
In dieser Ausgabedatei sind von allen Partikeln Daten wie Position, Geschwindigkeit etc. enthalten.		
<code>.gtf</code>	<b>Granular Time File</b>	Anh. C.5
In dieser Ausgabedatei wird der Zeitverlauf einiger physikalischer Größen protokolliert.		

<code>.gpl</code>	<b>Granular Particle Line</b>	Anh. C.6
-------------------	-------------------------------	----------

In dieser Ausgabedatei wird in kompakter Form die Trajektorie für jedes Teilchen gespeichert.

<code>.glk</code>	<b>Granular Lock File</b>	-
-------------------	---------------------------	---

Diese Sicherheits-Datei enthält keinerlei Information. Allein ihre Existenz ist von Bedeutung. `gc2d` prüft, bevor es irgendeine andere Datei schreibt, ob diese Datei existiert. Ist sie vorhanden, bedeutet dies, daß diese Simulation schon gelaufen ist, bereits erzeugte Daten also überschrieben werden könnten. In diesem Fall bricht `gc2d` die Simulation sicherheitshalber ab<sup>1</sup>. Wenn die Datei nicht existiert, wird sie erzeugt.

<code>.ps</code>	<b>PostScript File</b>	-
------------------	------------------------	---

Hier wird ein Bild der Simulation im Postscriptformat abgelegt.

Den Namen der Eingabedateien teilt man `gc2d` über die Kommandozeilenoption `-i` mit. Dort muß der vollständige Dateiname der `gsi`-Datei inklusive der Endungen angegeben werden. Aus dem angegebenen String wird dann der Name der entsprechenden `gci`-Datei bestimmt. Da die `gsi`-Dateien beim Einlesen noch durch einen Filter laufen können, gibt es 4 Möglichkeiten:

<code>gsi</code> -Dateiname	<code>gci</code> -Dateiname	verwendeter Filter
<code>foo-in.gsi</code>	<code>foo-in.gci</code>	Es wird kein Filter verwendet
<code>foo-in.gsi.gz</code>	<code>foo-in.gci</code>	Die <code>gsi</code> -Datei wird vor dem Lesen noch dekomprimiert
<code>foo-in.gsi.m4</code>	<code>foo-in.gci</code>	<b>Veraltet!</b> Die <code>gsi</code> -Datei wird vor dem Lesen durch den Makroprozessor M4 geleitet.
<code>foo-in.gsi.pl</code>	<code>foo-in.gci</code>	Die Datei wird als Perl-programm ausgeführt. Die Ausgabe dieses Programms wird als Eingabe verwendet (siehe Anhang D.1).

Der Name der Ausgabedateien wird durch den Befehl **Basename** in der `gsi`-Datei festgelegt. An diesen Namen werden dann für die entsprechenden Dateien eventuell noch eine laufende Nummer und die Endung eingefügt. Würde man den Befehl **Basename** `foo-out` geben, könnten die Ausgabedateien des Programms wie folgt aussehen:

```
doublon:~/gc2d/data>ls
foo-out.0001.gco  foo-out.0002.gco  foo-out.0001.gff
foo-out.0002.gff  foo-out.0002.gso  foo-out.0001.gso
foo-out.0001.gpf  foo-out.0002.gpf  foo-out.gco
foo-out.gso      foo-out.ggf      foo-out.glk
foo-out.gpl      foo-out.gtf      foo-out.ps
```

<sup>1</sup>Dieses Verhalten kann durch die Kommandozeilenoption `-l` verändert werden, siehe Anhang E.1

## Anhang B

# Format der Eingabedateien

### B.1 Eingabedatei zur Simulationsbeschreibung (`gsi`)

#### B.1.1 Einführung

Dateien vom Typ `gsi` sind die wichtigsten Eingabedateien für die Simulation. Die Aufbau der Simulation wird dabei mit einer Beschreibungssprache festgelegt. Mit Hilfe dieser Sprache wird das physikalische System detailliert beschrieben, so daß es dann mit `gc2d` berechnet werden kann. Die Syntax selbst ist sehr einfach gehalten; Berechnungen, Kontrollstrukturen wie Schleifen oder Abfragen sind nicht vorhanden. Stattdessen existiert ein Interface zur Skriptsprache Perl (siehe Anhang D.1).

Der Aufbau einer `gsi`-Datei ist im Prinzip der folgende:

```

PARAMETER v1.4
Identifikationstext

# Kommentar
$ Spezieller Kommentar

Steuerwort Parameter
OBJEKTNAME
Steuerwort Parameter Parameter Kommentar
Steuerwort Parameter
Steuerwort Parameter Kommentar

OBJEKTNAME
Steuerwort

End_Of_File

```

Die erste Zeile der Datei muß immer den Text `PARAMETER v1.4` enthalten. Anhand dieses Strings wird die Datei identifiziert. Durch die Versionsnummer wird sichergestellt, daß nicht alte Steuerdateien, die evtl. nicht mehr der aktuellen Syntax folgen, bearbeitet werden. Die zweite Zeile enthält einen beliebigen Text, der als Identifikation benützt wird. Ausgewertet wird dieser Eintrag erst in der entsprechenden `gci`-Datei (siehe Anh. B.2). Wird keine `gci`-Datei erstellt, so kann diese Zeile auch leer bleiben.

Der ganze Aufbau ist an einzelnen Objekten ausgerichtet. Ein typisches Objekt ist z.B. `Wall`. Auf einen Objektnamen folgen dann Befehle, die das jeweilige Objekt genauer beschreiben. Standardmäßig wird am Anfang der Datei das Objekt `GENERAL` bearbeitet. Es ist nicht möglich, im weiteren Verlauf der `gsi`-Datei zu ihm zurückzukehren; ein entsprechendes Befehlswort ist nicht vorgesehen. Dies hat den Grund, daß hier Parameter gesetzt werden müssen, die im weiteren Verlauf der `gsi`-Datei schon benötigt werden. Speziell sind dies der Name der Ausgabedatei und der Zeitschritt. Die weiteren Objekte können in beliebiger Reihenfolge angegeben werden. Bei `Display` wird allerdings nur das letzte in der Eingabedatei vorkommende Objekt ausgewertet. Die Datei wird abgeschlossen mit dem Befehlswort `END_OF_FILE`; evtl. folgender Text wird ignoriert.

Ein Kommentar, der mit einem `#` beginnt, wird einfach ignoriert. Beginnt eine Zeile allerdings mit einem `$`, so wird dieser Kommentar von `gc2d` am Bildschirm ausgegeben.

Es gelten die folgenden Regeln:

1. Groß- und Kleinschreibung haben innerhalb der `gsi`-Datei keinen Einfluß. Ausnahmen hiervon sind extra vermerkt.
2. Wird bei einem Befehl, der eigentlich 2 Parameter erwartet, nur ein Wert übergeben, wird der zweite automatisch auf 0 gesetzt.
3. Auf die Parameter folgende Texte werden ignoriert
4. Leerzeilen werden ignoriert.
5. Vor und hinter einem Befehl oder einem Sektionsnamen können beliebig viele Leerzeichen oder Tabulatoren stehen.
6. Die Parameter werden durch eine beliebige Zahl von Leerzeichen oder Tabulatoren getrennt.
7. Die Zeilenlänge darf 1023 Zeichen nicht überschreiten.

Die nachfolgende Beschreibung der Befehle entspricht dem Schema:

BEFEHLSNAME	Typ	Typ	Kategorie
<code>Perlbehl(Vorgabewert, Vorgabewert);</code>			Einheit

Als Parametertypen kommen **float**, **integer**, und **string** in Frage; eine entsprechende Typenkonvertierung wird vorgenommen. Die Kategorie eines Befehls ist entweder **USER** oder **System**. Fällt ein Befehl in letztere Kategorie, ist es nicht empfehlenswert, den Befehl zu verwenden, da falsche Werte zu völlig unphysikalischen Simulationen führen können. Diese Befehle werden hauptsächlich von `gc2d` verwendet, um eine Simulation wieder abzuspeichern (siehe Anh C.1). In der zweiten Zeile findet sich dann der zugehörige Perl-Befehl, genaueres zu deren Nutzung findet sich im Anhang D.1. In Klammern angegeben sind die Werte, die standardmäßig verwendet werden. Ebenfalls mit angegeben ist dann gegebenenfalls die Einheit der angegebenen Werte. Die Angabe  $\infty$  bei manchen Werten bedeutet, daß hier lediglich eine sehr große Zahl<sup>1</sup> verwendet wird, die bei normalen Simulationen nicht erreicht wird.

### B.1.2 Das Objekt GENERAL

Mit den folgenden Befehlen werden globale Parameter der Simulation festgelegt. Auch wenn vernünftige Werte als Standard vorgegeben sind, so ist es doch sinnvoll, diese Parameter bei jeder Simulation selbst festzulegen und aufeinander abzustimmen.

<sup>1</sup>Dies ist im allgemeinen der Wert  $(2^{32} - 1)$  von `BIG_NUMBER`, der in der Datei `gsd.h` festgelegt wird.

**B.1.2.1 Allgemeine Parameter**

<b>DELTA_T</b>	<b>float</b>	<b>User</b>
Delta_T(0.0001);		s

Durch dieses Kommando wird die Größe des Zeitschritts für den Predictor-Corrector festgelegt (siehe Gleichung 2.9).

<b>TIME</b>	<b>float</b>	<b>float</b>	<b>User</b>
Time(0.0,1.0);			s

Durch dieses Kommando wird die Simulationszeit festgelegt. Der erste Wert gibt den Startzeitpunkt der Simulation an, der zweite den Endzeitpunkt.

<b>MUE_STATISCH</b>	<b>float</b>	<b>User</b>
Mue_Statisch(0.6);		

Durch dieses Kommando wird der statische Reibungskoeffizient festgelegt. Es muß  $Mue\_Statisch \geq 0$  und  $Mue\_Statisch < 1.0$  gelten.

<b>MUE_DYNAMISCH</b>	<b>float</b>	<b>User</b>
Mue_Dynamisch(0.6);		

Durch dieses Kommando wird der dynamische Reibungskoeffizient festgelegt. Es muß  $MUE\_DYNAMISCH \geq 0$  und  $MUE\_DYNAMISCH < 1.0$  gelten. Dieser Wert wird z.Z. im Programm nicht genutzt!

<b>E_MODUL</b>	<b>float</b>	<b>User</b>
E_Modul(1e7);		$\frac{N}{m}$

Durch dieses Kommando wird der Youngsche Elastizitätsmodul festgelegt. Es muß  $E\_Modul > 0$  gelten.

<b>GAMMA</b>	<b>float</b>	<b>User</b>
Gamma(0.5);		

Durch dieses Kommando wird der Dämpfungskoeffizient für die Kollision festgelegt. Es muß  $GAMMA \geq 0$  gelten.

<b>COHESION</b>	<b>float</b>	<b>User</b>
Cohesion(0.0);		$\frac{N}{m}$

Durch dieses Kommando wird der Kohäsionskoeffizient festgelegt. Es muß  $COHESION \geq 0$  gelten.

<b>STEPS_DATA</b>	<b>integer</b>	<b>User</b>
Steps_Data(10000);		$\Delta t$

Durch dieses Kommando wird festgelegt, alle wieviele Zeitschritte Ausgabefiles vom Typ `gff` und `gpff` geschrieben werden bzw. die Datei vom Typ `gtf` ergänzt wird.

<b>STEPS_CHECKPOINT</b>	<b>integer</b>	<b>User</b>
Steps_Checkpoint(10000);		$\Delta t$

Durch dieses Kommando wird festgelegt, alle wieviele Zeitschritte Dateien vom Typ `gso` und `gco` zum Wiederaufsetzen des Laufs geschrieben wird.

<b>STEPS_SETUP</b>	<b>integer</b>	<b>User</b>
Steps_Setup( $\infty$ );		$\Delta t$

Durch dieses Kommando wird festgelegt, alle wieviele Zeitschritte eine neue Datei vom Typ `gco` und `gso` geschrieben wird.

<b>STEPS_DUMPLINE</b>	<b>integer</b>	<b>User</b>
Steps_Dumpline( $\infty$ );		$\Delta t$

Durch dieses Kommando wird festgelegt, alle wieviele Zeitschritte eine neue Datei vom Typ `gp1` geschrieben wird. Die Unterstützung für diesen Befehl muß mit `#define USE_DUMPLINE` einkompiliert sein.

<b>SEED</b>	<b>float</b>	<b>float</b>	<b>User</b>
Seed(0,0);			

Durch dieses Kommando werden die Initialisierungswerte des Zufallsgenerators festgelegt.

<b>BASENAME</b>	<b>string</b>	<b>User</b>
Basename("default");		

Durch dieses Kommando wird festgelegt, unter welchem Namen die Ausgabedateien angelegt werden. Es kann auch ein Pfad relativ zum Verzeichnis, in dem `gc2d` aufgerufen wurde, angegeben werden. Bei diesem Befehl wird die Groß- und Kleinschreibung des Parameters berücksichtigt.

<b>EMAIL</b>	<b>string</b>	<b>User</b>
Email("NONE");		

Durch dieses Kommando wird festgelegt, ob und an wen eine Email geschickt wird, wenn `gc2d` ordnungsgemäß beendet wurde. Wird der String **"NONE"** angegeben, wird keine Email versandt. Bei diesem Befehl wird die Groß- und Kleinschreibung des Parameters berücksichtigt.

<b>NEXT_DATA_FILENUMBER</b>	<b>integer</b>	<b>System</b>
Next_Data_Filenumber(1);		

Durch dieses Kommando wird festgelegt, mit welcher laufenden Nummer die nächsten Files vom Typ `gff` und `gpf` geschrieben werden.

<b>NEXT_SETUP_FILENUMBER</b>	<b>integer</b>	<b>System</b>
Next_Setup_Filenumber(1);		

Durch dieses Kommando wird festgelegt, mit welcher laufenden Nummer die nächsten Files vom Typ `gso` und `gco` geschrieben werden.

### B.1.2.2 Variation der Erdbeschleunigung

Durch entsprechende Konfiguration bei der Kompilation (siehe Anhang E.2) ist es möglich, den Wert von  $g$  während der Simulation zu variieren.

Ist diese Funktionalität nicht aktiviert, wird nur der Befehl **G** mit einem Parameter akzeptiert.

<b>G</b>	<b>float</b>	<b>User</b>
G(9.81);		$\frac{m}{s^2}$

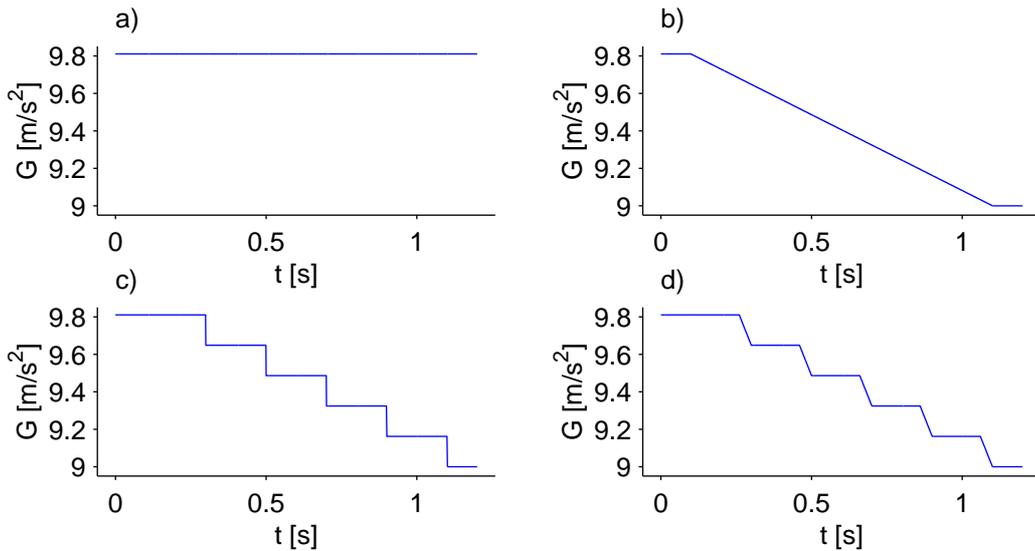
Durch dieses Kommando wird der Wert für die Erdbeschleunigung festgelegt. Der entsprechende

Vektor ist dann  $\begin{pmatrix} 0 \\ -g \end{pmatrix}$ .

Andernfalls kennt `gc2d` noch folgende Befehle:

<b>G</b>	<b>float</b>	<b>float</b>	<b>User</b>
G(9.81,9.81);			$\frac{m}{s^2}$

Durch dieses Kommando werden der Anfangswert  $g_1$  und der Endwert  $g_2$  der Erdbeschleunigung festgelegt.



**Abbildung B.1:** Die vier Varianten, wie die die Erdbeschleunigung  $G$  variiert werden kann. Die Parameter sind

```
G 9.81 9.0 G_TIME 0.1 1.1 G_Parameter 5 0.2
```

- a) `G_Type CONSTANT` b) `G_Type LINEAR`
- c) `G_Type STEP` d) `G_Type SMOOTH_STEP`

<b>G_TIME</b>	<b>float</b>	<b>float</b>	<b>User</b>
G_Time(0.0, 1.0);			s

Durch dieses Kommando werden Anfangszeit  $t_1$  und Endzeit  $t_2$  für die Änderung der Gravitation festgelegt.

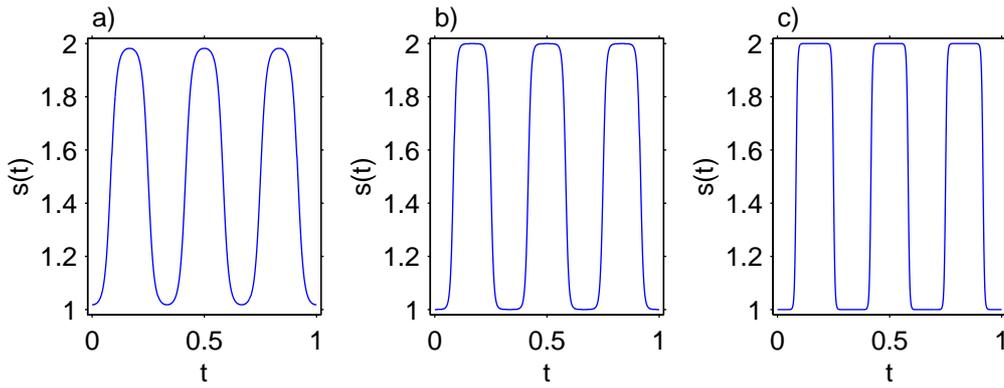
<b>G_PARAMETER</b>	<b>float</b>	<b>float</b>	<b>User</b>
G_Parameter(5, 0.5);			N/A

Durch dieses Kommando werden zwei Parameter  $p_1$  und  $p_2$  für die Änderung der Gravitation festgelegt (siehe Abbildung B.1). Die Bedeutung der Parameter kann sich je nach Art der Manipulation von  $g$  ändern.

<b>G_TYPE</b>	<b>string</b>	<b>User</b>
G_Type("CONSTANT");		

Durch dieses Kommando wird die Art festgelegt, wie  $g$  variiert wird (siehe Abbildung B.1).

<b>CONSTANT</b>	$g = g_1$
<b>LINEAR</b>	$g$ wird im Zeitraum $[t_1, t_2]$ linear von $g_1$ auf $g_2$ geändert.
<b>STEP</b>	$g$ wird im Zeitraum $[t_1, t_2]$ stufenweise von $g_1$ auf $g_2$ geändert. Insgesamt werden $p_1$ Stufen gemacht.
<b>SMOOTH_STEP</b>	$g$ wird im Zeitraum $[t_1, t_2]$ stufenweise von $g_1$ auf $g_2$ geändert. Insgesamt werden $p_1$ Stufen gemacht. Die einzelnen Stufen sind keine Sprünge, sondern werden linear ausgeführt. Der Parameter $p_2$ gibt die Breite der linearen Änderung an wobei $p_2 = 0$ der Grenzfall <b>STEP</b> ist, $p_2 = 1$ entspricht <b>LINEAR</b>



**Abbildung B.2:** Dargestellt ist die zeitliche Verlauf der Größenskalierung mit  $f = 3$ ,  $a = 1$  und a)  $S = 2$  b)  $S = 4$  und c)  $S = 8$

### B.1.2.3 Temperaturveränderung

Mit den folgenden Befehlen kann eine Größenänderung der Teilchen aufgrund einer Temperaturänderung simuliert werden. Die Unterstützung hierfür muß mit `#define USE_HEATING` einkompiliert werden.

<b>HEAT_TIME</b>	<b>float</b>	<b>float</b>	<b>User</b>
Heat_Time( $\infty, \infty$ );			s
<b>HEAT_AMPLITUDE</b>	<b>float</b>		<b>User</b>
Heat_Amplitude(0.01);			
<b>HEAT_FREQUENCY</b>	<b>float</b>		<b>User</b>
Heat_Frequency(1.0);			$s^{-1}$
<b>HEAT_SLOPE</b>	<b>float</b>		<b>User</b>
Heat_Slope(10.0);			

Durch dieses Kommandos wird festgelegt, wie die Größe der Partikel skaliert wird. Mittels `HEAT_TIME` wird der Zeitraum  $[t_1, t_2]$  der Größenskalierung  $s(t)$  festgelegt. Die Skalierung für den Zeitpunkt  $t_{\text{akt}}$  erfolgt dann nach der Formel (siehe Abbildung B.2)

$$t = \begin{cases} t_1 & t_{\text{akt}} < t_1 \\ t_{\text{akt}} & t_1 \leq t_{\text{akt}} \leq t_2 \\ t_2 & t_2 < t_{\text{akt}} \end{cases} \quad (\text{B.1})$$

$$s(t) = 1 + a \cdot \left( \frac{1}{2} - \frac{1}{2} \tanh(S \cos(2\pi f t)) \right). \quad (\text{B.2})$$

$a \hat{=} \text{HEAT\_AMPLITUDE}$

$f \hat{=} \text{HEAT\_FREQUENCY}$

$S \hat{=} \text{HEAT\_SLOPE}$

### B.1.3 Das Objekt DISPLAY

Dieses Objekt beschreibt die Parameter für die graphische Darstellung der Simulation.

<b>X_RANGE</b>	<b>float</b>	<b>float</b>	<b>User</b>
X_Range(-0.1, 0.1);			m

Durch dieses Kommando wird festgelegt, welcher Bereich der X-Achse graphisch dargestellt wird. Der zweite Wert muß größer als der erste sein.

<b>Y_RANGE</b>	<b>float</b>	<b>float</b>	<b>User</b>
Y_RANGE(-0.1, 0.1);			m

Durch dieses Kommando wird festgelegt, welcher Bereich der Y-Achse graphisch dargestellt wird. Der zweite Wert muß größer als der erste sein.

<b>SCREEN</b>	<b>integer</b>	<b>integer</b>	<b>User</b>
Screen(640, 0);			Pixel

Durch dieses Kommando wird festgelegt, mit welcher Größe das Ausgabefenster geöffnet wird. Ist einer der beiden Werte  $\leq 0$  wird der andere so festgelegt, daß die Grafik im Seitenverhältnis 1 : 1 dargestellt wird.

<b>INTERVAL</b>	<b>integer</b>		<b>User</b>
Interval(100);			$\Delta t$

Durch dieses Kommando wird festgelegt, alle wieviele Zeitschritte eine Bildschirmdarstellung erfolgen soll.

<b>MODUS</b>	<b>integer</b>		<b>User</b>
Modus(3);			

Durch dieses Kommando wird festgelegt, welche Elemente zu Beginn dargestellt werden. Der Wert wird durch das logische ODER der in der Datei `button_mask.h`<sup>2</sup> angegebenen Parameter berechnet. Die Standardeinstellung ist  $0x00000001 \vee 0x0000002l = 3$ , d.h. es werden Umrandung und Fläche der Teilchen dargestellt.

## B.1.4 Das Objekt PARTICLE

Mit diesem Objekt beschreibt man die Teilchen, die mit `gc2d` simuliert werden.

### B.1.4.1 Allgemeine Parameter

<b>CORNERS</b>			<b>User</b>
Corners();			m

Die Angabe der Ecken erfolgt nach einer speziellen Methode. Der Zeile, in der der Befehl `CORNERS` steht, folgen weitere Zeilen, die jeweils die X- und Y-Koordinaten eines Punktes enthalten. Nach der letzten Ecke folgt der Befehl `END`. Zwischen `CORNERS` und `END` sind keine anderen Befehle oder Kommentare zulässig. Defaultmäßig wird das Polygon von `gc2d` so verschoben, daß der Schwerpunkt an dem durch `COM` vorgegebenen Punkt liegt. Die Punkte müssen in der Reihenfolge gegen den Uhrzeigersinn angegeben werden und konvex sein. Dies wird in der gegenwärtigen Programmversion nicht überprüft, `gc2d` verläßt sich auf korrekte Angaben des Nutzers.

<b>COM</b>	<b>float</b>	<b>float</b>	<b>User</b>
Com(0.0, 0.0);			m

Dieser Befehl gibt die Position des Schwerpunktes vor. Achtung! Wenn das Kommando `POSITION_ABSOLUT` gegeben wird, wird diese Zeile ignoriert.

<sup>2</sup>Die Datei `button_mask.h` befindet sich auf der beigelegten CDROM.

<b>SPECIAL</b>			<b>System</b>
Special();			

Nur zu Debuggingzwecken!

<b>VELOCITY</b>	<b>float</b>	<b>float</b>	<b>User</b>
Velocity(0.0,0.0);			<i>m/s</i>

Dieser Befehl gibt die Translationsgeschwindigkeit des Teilchens vor.

<b>ACCELERATION</b>	<b>float</b>	<b>float</b>	<b>User</b>
Acceleration(0.0,0.0);			<i>m/s<sup>2</sup></i>

Dieser Befehl gibt die Beschleunigung des Teilchens vor.

<b>PC_TRANS3</b>	<b>float</b>	<b>float</b>	<b>System</b>
Pc_Trans3(0.0,0.0);			<i>m/s<sup>3</sup></i>

<b>PC_TRANS4</b>	<b>float</b>	<b>float</b>	<b>System</b>
Pc_Trans4(0.0,0.0);			<i>m/s<sup>4</sup></i>

<b>PC_TRANS5</b>	<b>float</b>	<b>float</b>	<b>System</b>
Pc_Trans5(0.0,0.0);			<i>m/s<sup>5</sup></i>

Diese Befehle geben die höheren Ableitungen des Predictor-Corrector für die Translation wieder, sie sollten nicht durch den Nutzer vorgegeben werden.

<b>ANGLE</b>	<b>float</b>		<b>User</b>
Angle(0.0);			

Dieser Befehl gibt den Winkel des Teilchens vor. Zulässig sind Werte von 0 bis  $2\pi$ . Die Drehung erfolgt im Uhrzeigersinn.

<b>OMEGA</b>	<b>float</b>		<b>User</b>
Omega(0.0);			<i>1/s</i>

Dieser Befehl gibt die Winkelgeschwindigkeit des Teilchens vor.

<b>OMEGA_DOT</b>	<b>float</b>		<b>User</b>
Omega_Dot(0.0);			<i>1/s<sup>2</sup></i>

Dieser Befehl gibt die Winkelbeschleunigung des Teilchens vor.

<b>PC_ROT3</b>	<b>float</b>		<b>System</b>
Pc_Rot3(0.0);			

<b>PC_ROT4</b>	<b>float</b>		<b>System</b>
Pc_Rot4(0.0);			

<b>PC_ROT5</b>	<b>float</b>		<b>System</b>
Pc_Rot5(0.0);			

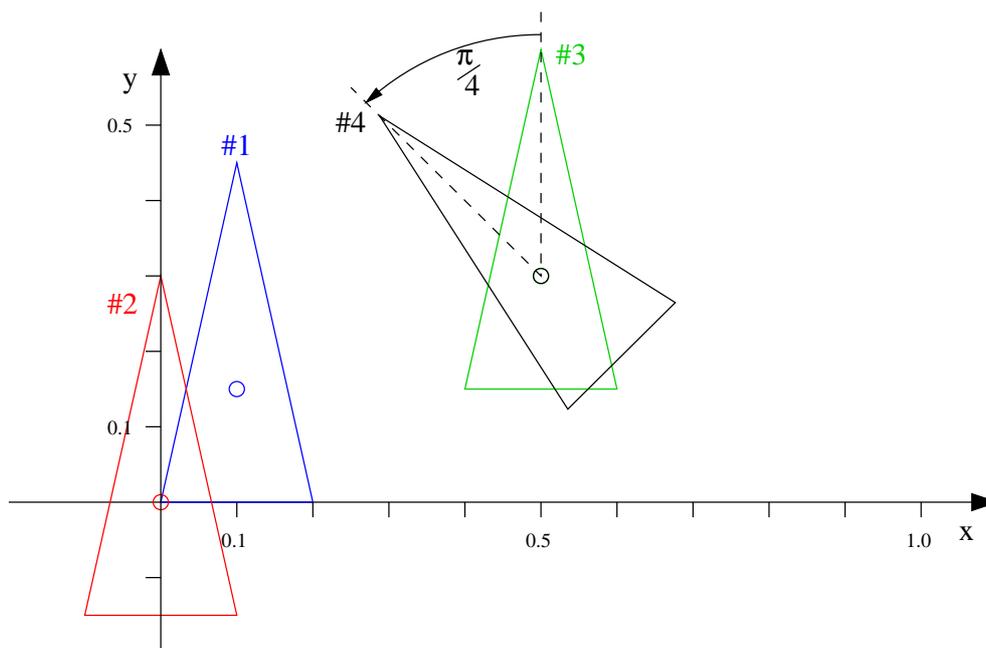
Diese Befehle geben die höheren Ableitungen des Predictor-Corrector für die Rotation wieder, sie sollten nicht durch den Nutzer vorgegeben werden.

<b>ID</b>	<b>integer</b>		<b>System</b>
Id(?);			

Durch dieses Kommando wird festgelegt, welche interne ID das Partikel erhält. Dieser Wert sollte nicht durch den Nutzer vorgegeben werden. Einen Standardwert gibt es nicht, stattdessen wird eine freie Nummer bestimmt und dem Teilchen zugewiesen.

<b>POSITION_ABSOLUT</b>			<b>User</b>
Position_Absolut();			

Dieser Befehl gibt an, daß die mit **CORNERS** angegebenen Eckkoordinaten als absolute Koordinaten



**Abbildung B.3:** Die verschiedenen Schritte bei der Positionierung eines Teilchens.

betrachtet werden; die Angabe **COM** wird in diesem Fall ignoriert.

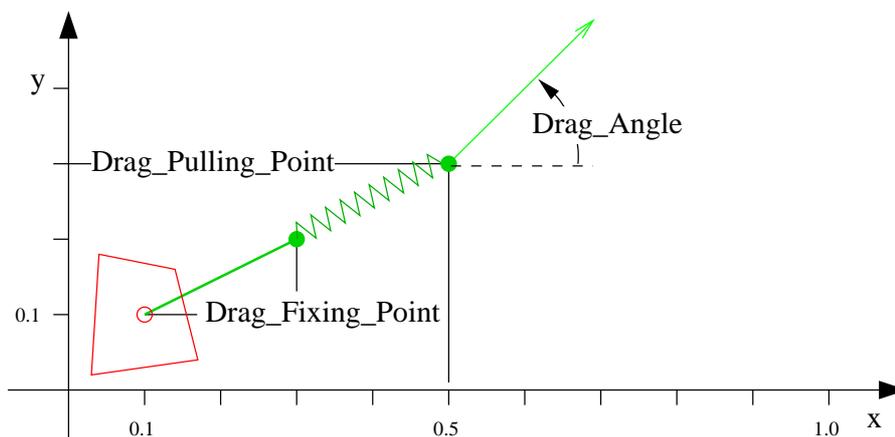
Am Beispiel eines Dreiecks seien die notwendigen Befehle gezeigt: die Basislänge beträgt 0.2m und die Höhe 0.45m, der Schwerpunkt soll am Punkt (0.5, 0.3) liegen. Das Teilchen soll um  $\pi/4$  gedreht werden.

```
PARTICLE
Com 0.5 0.3 # Position des Schwerpunktes
Corners # Vorgabe der Ecken
0.0 0.0
0.0 0.2
0.1 0.45
END
Angle 0.7853975 # Neigung des Quadrats
```

In Abbildung B.3 sind die einzelnen Schritte dargestellt. Im ersten Schritt (blau) werden durch **CORNERS** die Eckpunkte des Dreiecks festgelegt. `gc2d` verschiebt das Teilchen dann automatisch so, daß der Schwerpunkt am Punkt (0, 0) liegt (rot). Durch den Befehl **COM** wird die Position des Schwerpunktes festgelegt (grün). Der Befehl **ANGLE** dreht das Teilchen dann in seine endgültige Position.

#### B.1.4.2 Teilchen mit Feder

An einem einzelnen Teilchen kann eine virtuelle Feder befestigt werden. Das eine Ende der Feder ist am Teilchen selbst befestigt, das andere kann entlang einer Geraden mit fester Geschwindigkeit gezogen werden.



**Abbildung B.4:** Beispiel für die beschreibenden Parameter eines mit einer Feder versehenen Teilchens.

<b>DRAG_ANGLE</b>	<b>float</b>	<b>User</b>
<code>Drag_Angle(0.0);</code>		

Dieser Befehl gibt an, in welche Richtung sich das durch **DRAG\_PULLING\_POINT** vorgegebene Ende der Feder bewegen soll.

<b>DRAG_VELOCITY</b>	<b>float</b>	<b>float</b>	<b>User</b>
<code>Drag_Velocity(0.1,);</code>			$\frac{m}{s}$

Dieser Befehl gibt an, mit welcher Geschwindigkeit sich das freie Ende der Feder bewegen soll.

<b>DRAG_K</b>	<b>float</b>	<b>User</b>	
<code>Drag_K(100);</code>			N/m

Dieser Befehl gibt die Federkonstante vor.

<b>DRAG_TIME</b>	<b>float</b>	<b>float</b>	<b>User</b>
<code>Drag_Time(0.0,1.0);</code>			s

Dieser Befehl gibt das Zeitintervall vor, in dem sich das freie Ende der Feder bewegt.

<b>DRAG_FIXING_POINT</b>	<b>float</b>	<b>float</b>	<b>User</b>
<code>DRAG_FIXING_POINT(0.0,0.0);</code>			m

Dieser Befehl gibt einen Punkt relativ zum Schwerpunkt vor, an dem die Feder angreifen soll. Der Punkt darf auch außerhalb des Teilchens liegen. Die Feder wird dann über einen masselosen Stab am Teilchen befestigt.

<b>DRAG_PULLING_POINT</b>	<b>float</b>	<b>float</b>	<b>User</b>
<code>DRAG_PULLING_POINT(0.0,0.0);</code>			m

Dieser Befehl gibt die Startposition des freien Endes der Feder an. Dieser Punkt darf auch innerhalb des Teilchens liegen.

Hier nun ein Beispiel, wie man einen kleinen Block mit einer Feder bewegen kann. Das freie Ende der Feder wird für 1 Sekunde mit der Geschwindigkeit  $1 \frac{m}{s}$  unter einem Winkel von  $\frac{\pi}{4}$  bewegt. Zu Beginn befindet sich dieses Federende an den Koordinaten (0.5, 0.3). Das am Teilchen befestigte Ende der Feder hängt an einer masselosen Stange, die im Schwerpunkt befestigt ist. Die Stange reicht 0.2 nach rechts und 0.1 nach oben. Folgender Code erzeugt den gewünschten Aufbau der in Abbildung B.4 zu sehen ist.

```

PARTICLE
Com          0.1  0.1 # Schwerpunkt
Corners      # Vorgabe der Ecken
...          # usw.
END
Drag_Angle   0.785397 # Winkel
Drag_Velocity 1.0     # Geschwindigkeit
Drag_K       100      # Federkonstante
Drag_Time    1        2 # Zeitintervall
DRAG_FIXING_POINT 0.2  0.1 # Befestigungspunkt
DRAG_PULLING_POINT 0.5  0.3 # Zugpunkt

```

### B.1.5 Das Objekt WALL

Mit diesem Objekt beschreibt man eine Wand, also ein unbewegliches Teilchen, in der Simulation.

CORNERS	User
Corners();	m

Die Angabe der Ecken erfolgt nach einer speziellen Methode. Der Zeile, in der der Befehl CORNERS steht, folgen weitere Zeilen, die jeweils die X- und Y-Koordinaten eines Punktes enthalten. Nach der letzten Ecke folgt der Befehl END. Zwischen CORNERS und END sind keine anderen Befehle zulässig. Die Punkte müssen in der Reihenfolge gegen den Uhrzeigersinn angegeben werden und konvex sein. Dies wird in der gegenwärtigen Programmversion nicht überprüft, sondern gc2d verläßt sich auf korrekte Angaben des Nutzers. Im Gegensatz zu einem Teilchen wird eine Wand nicht automatisch so verschoben, daß ihr Schwerpunkt bei (0, 0) liegt.

OFFSET	float	float	User
Offset(0.0,0.0);			m

Gibt eine Verschiebung der Wand um den angegebenen Vektor an.

ANGLE	float	User
Angle(0.0);		

Die Wand wird um den angegebenen Winkel um ihren Schwerpunkt gedreht, zulässig sind Werte von 0 bis  $2\pi$ . Die Drehung erfolgt im Uhrzeigersinn.

VELOCITY	float	float	User
Velocity(0.0,0.0);			m/s

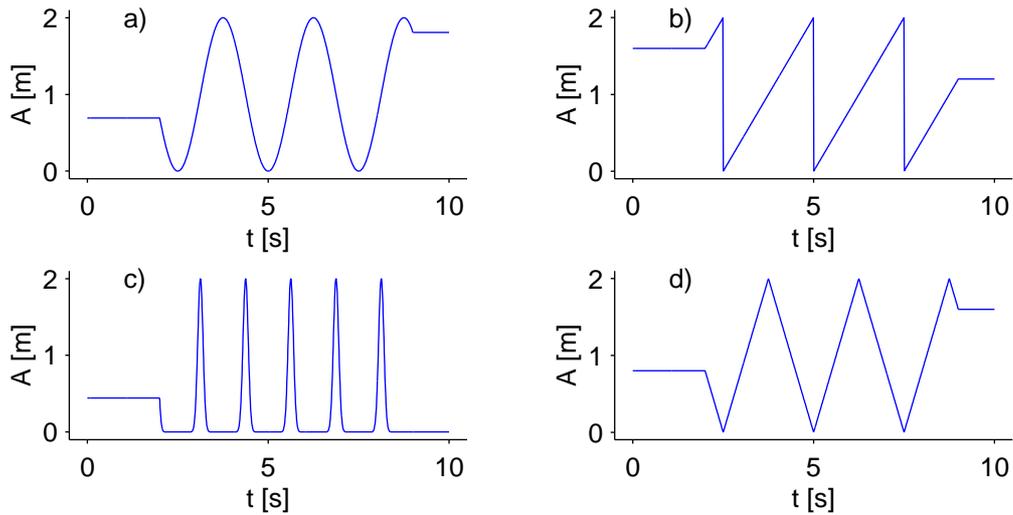
Gibt eine virtuelle Geschwindigkeit der Oberfläche wieder. Achtung! Diese Geschwindigkeit wirkt nur tangential zur Oberfläche!

ID	integer	System
Id(?);		

Durch dieses Kommando wird festgelegt, welche interne ID das Partikel erhält. Dieser Wert sollte nicht durch den Nutzer vorgegeben werden. Einen Standardwert gibt es nicht, stattdessen wird eine freie Nummer bestimmt und der Wand zugewiesen.

#### B.1.5.1 Wand als Stempel

Neben den normalen Wänden gibt es auch noch die Möglichkeit, die Wand als Stempel zu erzeugen. Die Wand bewegt sich dann solange, bis sie die vorgegebene Kraft ausübt. Eine Wand kann nicht gleichzeitig auch noch vibrieren (Abschnitt B.1.5.2).



**Abbildung B.5:** Amplituden für die verschiedenen Schwingungstypen bei vibrierenden Wänden.

<b>PISTON_ANGLE</b>	<b>float</b>	<b>User</b>
Piston_Angle(0.0);		

Durch dieses Kommando wird die Richtung festgelegt, in die sich der Stempel bewegen soll.

<b>PISTON_FORCE</b>	<b>float</b>	<b>User</b>
Piston_Force(0.0);		N

Durch dieses Kommando wird die Kraft festgelegt, die der Stempel in Richtung **PISTON\_ANGLE** ausüben soll.

<b>PISTON_V_MAX</b>	<b>float</b>	<b>User</b>
Piston_V_Max(1.0);		m/s

Durch dieses Kommando wird der Betrag der Geschwindigkeit des Stempels begrenzt. So kann man erreichen, daß der Stempel vorsichtig auf eine granulare Oberfläche aufsetzt.

### B.1.5.2 Vibrierende Wände

Neben den normalen Wänden gibt es auch noch die Möglichkeit, die Wand vibrieren zu lassen. Eine Wand kann nicht gleichzeitig auch noch als Stempel dienen (Abschnitt B.1.5.1).

<b>VIB_ANGLE</b>	<b>float</b>	<b>User</b>
Vib_Angle( $\pi/2$ );		

Durch dieses Kommando wird die Richtung festgelegt, in die die Wand schwingt.

<b>VIB_TIME</b>	<b>float</b>	<b>float</b>	<b>User</b>
Vib_Time(0, $\infty$ );			s

<b>VIB_WIDTH</b>	<b>float</b>	<b>User</b>
Vib_Width(1);		m

<b>VIB_FREQUENCY</b>	<b>float</b>	<b>User</b>
Vib_Frequency(1);		1/s

<b>VIB_TYPE</b>	<b>string</b>	<b>User</b>
Vib_Type("COSINUS");		

Durch diese Kommandos wird die Art der Vibration festgelegt. Mittels VIB\_TIME wird der Zeitraum  $[t_1, t_2]$  bestimmt.

$$t = \begin{cases} t_1 & t_{\text{akt}} < t_1 \\ t_{\text{akt}} & t_1 \leq t_{\text{akt}} \leq t_2 \\ t_2 & t_2 < t_{\text{akt}} \end{cases} . \quad (\text{B.3})$$

Die Amplitude der Auslenkung kann dann in Abhängigkeit der Art der Vibration bestimmt werden:

$W \hat{=} \text{VIB\_WIDTH}$

$f \hat{=} \text{VIB\_FREQUENCY}$

<b>COSINUS</b>	$A(t) = W \cdot 0.5 \cdot (1 - \cos(t_{\text{akt}} f 2\pi))$	Abbildung B.5 a)
<b>SAWTOOTH</b>	$A(t) = W \cdot (t_{\text{akt}} \bmod \frac{1}{f}) f$	Abbildung B.5 b)
<b>TAPPING</b>	$A(t) = W \cdot \sin(t_{\text{akt}} f 2\pi)^{30}$	Abbildung B.5 c)
<b>TRIANGLE</b>	$A_{\text{temp}}(t) = (t_{\text{akt}} \bmod \frac{1}{f}) f$ $A(t) = \begin{cases} 2W \cdot A_{\text{temp}} & A_{\text{temp}} < 0.5 \\ 2W \cdot t(1 - A_{\text{temp}}) & \text{sonst} \end{cases}$	Abbildung B.5 d)

### B.1.6 Das Objekt GENERATE\_BLOCK

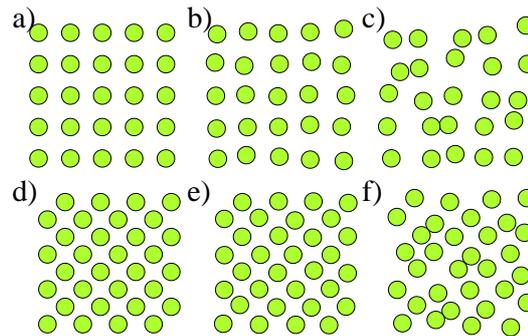
Mit diesem Objekt kann eine große Zahl von ähnlichen Teilchen beschrieben werden, die in einem Rechteck positioniert werden.

<b>X_RANGE</b>	<b>float</b>	<b>float</b>	<b>User</b>
X_Range(0.0, 10.0);			m
<b>Y_RANGE</b>	<b>float</b>	<b>float</b>	<b>User</b>
Y_Range(0.0, 10.0);			m

Durch diese Kommandos werden die Grenzen des Rechtecks vorgegeben, in dem die Teilchen generiert werden sollen.

<b>GRID_DISTANCE</b>	<b>float</b>	<b>float</b>	<b>User</b>
Grid_Distance(2.0, 2.0);			m

Durch diesen Befehl werden die Gitterabstände  $\Delta x$  und  $\Delta y$  bestimmt. Wie diese beiden Parameter genutzt werden, hängt vom verwendeten Gittertyp ab. Bei den Rechteckgittern werden direkt die Gitterabstände festgelegt. Bei den gegeneinander versetzten Quadratgittern gibt  $\Delta x$  den Abstand der Gitterpunkte in der X-Richtung vor,  $\Delta y$  bestimmt den Abstand zur nächsten Gitterebene.

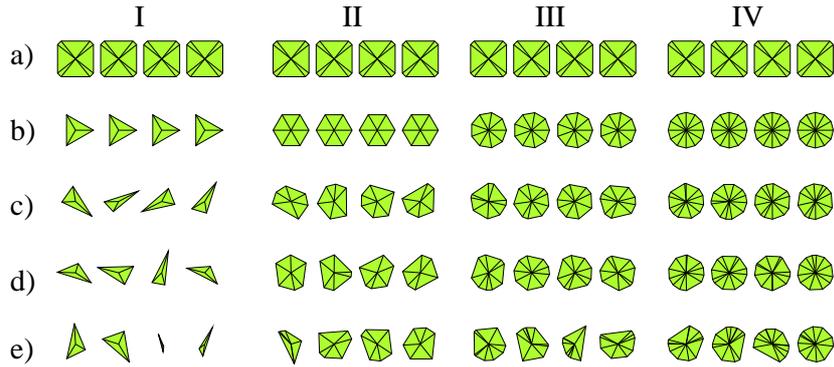


**Abbildung B.6:** Beispiele für die 6 verschiedenen Gittertypen, auf denen Teilchen angeordnet werden können. a) **RECT** b) **RECT\_GAUSS** c) **RECT\_EQUAL** d) **FCC** e) **FCC\_GAUSS** f) **FCC\_EQUAL**

<b>GRID_TYPE</b>	<b>string</b>	<b>User</b>
Grid_Type("FCC");		m
<b>RECT</b>	Ein Rechteckgitter: die Schwerpunkte der Teilchen liegen genau auf den Gitterpunkten.	
<b>RECT_GAUSS</b>	Ein Rechteckgitter: die Schwerpunkte der Teilchen sind in X-Richtung in einem Intervall $[-\frac{1}{2}\Delta x, \frac{1}{2}\Delta x]$ , in Y-Richtung im Intervall $[-\frac{1}{2}\Delta x, \frac{1}{2}\Delta x]$ um den Gitterpunkt mit einer abgeschnittenen Gaußkurve <sup>3</sup> verteilt.	
<b>RECT_EQUAL</b>	Ein Rechteckgitter: die Schwerpunkte der Teilchen sind in X-Richtung in einem Intervall $[-\frac{1}{2}\Delta x, \frac{1}{2}\Delta x]$ , in Y-Richtung im Intervall $[-\frac{1}{2}\Delta y, \frac{1}{2}\Delta y]$ um den Gitterpunkt gleichverteilt.	
<b>FCC</b>	Zwei gegeneinander versetzte Quadratgitter: die Schwerpunkte der Teilchen liegen genau auf den Gitterpunkten.	
<b>FCC_GAUSS</b>	Zwei gegeneinander versetzte Quadratgitter: die Schwerpunkte der Teilchen sind in X-Richtung in einem Intervall $[-\frac{1}{2}\Delta x, \frac{1}{2}\Delta x]$ , in Y-Richtung im Intervall $[-\frac{1}{2}\Delta y, \frac{1}{2}\Delta y]$ um den Gitterpunkt mit einer speziellen abgeschnittenen Gaußkurve verteilt.	
<b>FCC_EQUAL</b>	Zwei gegeneinander versetzte Quadratgitter: die Schwerpunkte der Teilchen sind in X-Richtung in einem Intervall $[-\frac{1}{2}\Delta x, \frac{1}{2}\Delta x]$ , in Y-Richtung im Intervall $[-\frac{1}{2}\Delta y, \frac{1}{2}\Delta y]$ um den Gitterpunkt gleichverteilt.	
<b>PART_SIZE</b>	<b>float</b> <b>float</b>	<b>User</b>
Part_Size(1.0,1.0);		m
<b>PART_VAR_SIZE</b>	<b>float</b>	<b>User</b>
Part_Var_Size(0.5);		
<b>RANDOM_SIZE</b>	<b>string</b>	<b>User</b>
Random_Size("CONST");		

<sup>3</sup>Ziel dieser Verteilung ist es, "schöne" Anfangsbedingungen auszuwählen. Einen Einfluß auf die Physik des Systems gibt es dabei nicht. Wichtig dabei ist, daß die Zufallszahl im Intervall  $[0, 1]$  liegt. Könnten die Zufallszahlen einen beliebigen Wert annehmen, wäre es möglich, daß nicht sinnvolle Parameter entstehen, zum Beispiel negative Eckenzahlen. Ausserdem sollte die Verteilung symmetrisch und relativ schmal sein.

Als praktikabel erwiesen sich dafür dann die Gaußverteilung  $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$  mit den Parametern  $\mu = 0.5$  und  $\sigma = 0.1$ . Das Einhalten der Intervallgrenzen  $[0, 1]$  kann dadurch erzwungen werden, daß Zufallszahlen, die zu groß oder zu klein sind, vom Programm einfach verworfen werden. Da  $\sigma$  hinreichend klein ist, ist die Wahrscheinlichkeit dafür auch so gering, daß die Verteilung nicht wesentlich gestört wird.



**Abbildung B.7:** Beispiele für verschiedenen Teilchenformen. In den einzelnen Zeilen sind Teilchen vom Type a) **SPECIAL\_RECTANGLE** b) **REGULAR** c) **RAND1** d) **RAND2** e) **FULL**, in den Spalten jeweils Beispiele für Teilchen mit I) 3 II) 6 III) 9 und III) 12 Ecken dargestellt.

Das Kommando **Part\_Size** gibt 2 Längen  $l_x$  und  $l_y$  zur Bestimmung der Teilchen vor, **Part\_Var\_Size** einen Wert  $I$ , der die Breite des Intervalls beschreibt.

<b>CONST</b>	Konstante Teilchengröße: <b>Part_Var_Size</b> wird ignoriert
<b>EQUAL</b>	Für jedes Teilchen $i$ werden die aktuellen Längen $l_x^i$ bzw. $l_y^i$ im Intervall $[l_x \cdot (1 - I), l_x \cdot (1 + I)]$ bzw. $[l_y \cdot (1 - I), l_y \cdot (1 + I)]$ gleichverteilt ausgewürfelt. Die Bestimmung der Werte $l_x^i$ und $l_y^i$ erfolgt unabhängig voneinander; dies hat zur Folge, daß die Teilchengröße nicht einfach skaliert wird, sondern die Form sich ändert.
<b>GAUSS</b>	Für jedes Teilchen $i$ werden die aktuellen Längen $l_x^i$ bzw. $l_y^i$ im Intervall $[l_x \cdot (1 - I), l_x \cdot (1 + I)]$ bzw. $[l_y \cdot (1 - I), l_y \cdot (1 + I)]$ mit einer speziellen abgeschnittenen Gaußkurve ausgewürfelt. Die Bestimmung der Werte $l_x^i$ und $l_y^i$ erfolgt unabhängig voneinander; dies hat zur Folge, daß die Teilchengröße nicht einfach skaliert wird, sondern die Form sich ändert.

<b>PART_NUM_CORNERS</b>	<b>integer</b>	<b>User</b>
Part_Num_Corners ( 6 ) ;		

<b>PART_VAR_CORNERS</b>	<b>integer</b>	<b>User</b>
Part_Var_Corners ( 2 ) ;		

<b>RANDOM_NUM_CORNERS</b>	<b>integer</b>	<b>User</b>
Part_Random_Num_Corners ( "CONST" ) ;		

**Part\_Num\_Corners** gibt eine Eckenzahl  $N_E$  vor, **Part\_Var\_Corners** bestimmt das Intervall  $N_I$  und **Random\_Num\_Corners** die Art der Verteilung.

<b>CONST</b>	Alle Teilchen haben die Eckenzahl $N_E$ .
<b>EQUAL</b>	Die Eckenzahl ist gleichverteilt im Intervall $[N_E - N_I, N_E + N_I]$ .
<b>GAUSS</b>	Die Eckenzahl wird im Intervall $[N_E - N_I, N_E + N_I]$ mit einer speziellen abgeschnittenen Gaußkurve ausgewürfelt .

<b>PART_RANDOM_POS_CORNERS</b>	<b>integer</b>	<b>User</b>
Part_Random_Pos_Corners ( "REGULAR" ) ;		

Zur Berechnung der Teilchenform wird folgendermaßen vorgegangen: die Halbachsen einer Ellipse  $a$  und  $b$  werden durch die entsprechenden Befehle vorgegeben. Dann werden, der Zahl der Ecken  $N$

entsprechend, die Winkel  $\varphi_0, \dots, \varphi_{(N-1)}$  ausgewürfelt und der Größe nach aufsteigend sortiert. Die Position der Ecken  $x_i, y_i$  ergeben sich dann aus der Parameterdarstellung der Ellipse [165] durch

$$x_i = a \cdot \cos(\varphi_i) \quad (\text{B.4})$$

$$y_i = b \cdot \sin(\varphi_i) \quad (\text{B.5})$$

Dies bedeutet, daß die Ecken eines Teilchens auf einer Ellipse liegen und damit das Polygon sicher konvex ist. Die Winkel werden wie folgt festgelegt, wobei dann  $i = 0 \dots N - 1$  gilt:

<b>SPECIAL_RECTANGLE</b>	Sonderfall; es werden Rechtecke mit abgeschrägten Kanten generiert
<b>REGULAR</b>	$\varphi_i = i \cdot \frac{2\pi}{N}$
<b>RAND1</b>	$\varphi_i$ zufällig aus dem Intervall $[(i - 0.5) \cdot \frac{2\pi}{N}, (i + 0.5) \cdot \frac{2\pi}{N}[$
<b>RAND2</b>	$\varphi_i$ zufällig aus dem Intervall $[(i - 1) \cdot \frac{2\pi}{N}, (i + 1) \cdot \frac{2\pi}{N}[$
<b>FULL</b>	$\varphi_i$ zufällig aus dem Intervall $[0, 2\pi[$

Typische Beispiele für die verschiedenen Teilchenformen sind in Abbildung B.7 gezeigt. Man erkennt, daß **RAND1** und **RAND2** sehr ähnlich sind, wobei die zweite Form etwas mehr zu überlangen Seiten tendiert. **FULL** wiederum generiert sehr unregelmäßige Teilchen, die bei niedriger Eckenzahl nur bedingt als typische Sandkörner gelten können.

<b>PART_ROT</b>	<b>float</b>	<b>User</b>
Part_Rot(0.0);		
<b>RANDOM_ROT</b>	<b>string</b>	<b>User</b>
Random_Rot("EQUAL");		

Durch **Part\_Rot** wird ein Drehwinkel  $\varphi_0$  vorgegeben. Der tatsächliche Winkel  $\varphi$  wird dann durch das Kommando **Random\_Rot** festgelegt.

<b>CONST</b>	$\varphi = \varphi_0$
<b>EQUAL</b>	$\varphi$ zufällig aus dem Intervall $[0, 2\pi[$
<b>GAUSS</b>	$\varphi$ wird mit der speziellen abgeschnittenen Gaußkurve aus dem Intervall $]\varphi_0 - \frac{\pi}{2}, \varphi_0 + \frac{\pi}{2}[$ ausgewählt.

<b>ANGLE_RANGE</b>	<b>float</b>	<b>float</b>	<b>User</b>
Angle_Range(0, 2π);			
<b>VELOCITY_RANGE</b>	<b>float</b>	<b>float</b>	<b>User</b>
Velocity_Range(0.0, 0.0);			$\frac{m}{s}$

Mit diesen Befehlen ist es möglich, den Teilchen eine individuelle Anfangsgeschwindigkeit zuzuweisen. Die Richtung wird in dem durch **Angle\_Range** gegebenen Intervall, der Betrag in dem durch **Velocity\_Range** vorgegebenen Bereich ausgewürfelt.

### B.1.7 Das Objekt DROPSOURCE

Mit diesem Objekt beschreibt man eine Teilchenquelle. Im Lauf der Simulation werden dabei an einer bestimmten Stelle Teilchen erzeugt.

<b>POS</b>	<b>float</b>	<b>float</b>	<b>User</b>
Pos(0.0,0.01);			m
<b>WIDTH</b>	<b>float</b>		<b>User</b>
Width(0.0);			
<b>TYPE</b>	<b>string</b>		<b>User</b>
Type('`ABS`');			s

Der Befehl **Pos** gibt zwei Werte  $x$  und  $y$ , der Befehl **width** eine Länge  $b$  vor. Je nach Art der Positionierung wird dann die Stelle bestimmt, an der ein Teilchen eingefügt wird.

<b>REL</b>	Relativer Positionierung
<b>ABS</b>	Absolute Positionierung

Bei beiden Arten der Positionierung wird die die X-Koordinate des Einfügepunktes für jedes Teilchen im Intervall  $[x - \frac{b}{2}, x + \frac{b}{2}]$  ausgewürfelt. Bei der absoluten Positionierung wird das neue Teilchen dann an der Y-Koordinate  $y$  eingesetzt. Bei der relativen Positionierung hingegen wird der höchste Schwerpunkt  $y_{max}$  eines Teilchens im Intervall  $[x - \frac{b}{2}, x + \frac{b}{2}]$  gesucht. Der Einsetzpunkt ist dann  $y_{max} + y$ . Wird kein Schwerpunkt gefunden, so ist  $y_{max} = 0$ .

<b>TIME</b>	<b>float</b>	<b>float</b>	<b>User</b>
Time(0.0,1.0);			s

Durch diesen Befehl wird der Zeitraum, in dem Teilchen eingefügt werden, festgelegt.

<b>DELTA_T</b>	<b>float</b>		<b>User</b>
Delta_T(0.1);			s

Durch diesen Befehl werden die Intervalle, die zwischen zwei Einfügevorgängen liegen, festgelegt.

<b>PART_SIZE</b>	<b>float</b>	<b>float</b>	<b>User</b>
Part_Size(0.01,0.01);			m

<b>PART_VAR_SIZE</b>	<b>float</b>		<b>User</b>
Part_Var_Size(0.1);			

<b>RANDOM_SIZE</b>	<b>string</b>		<b>User</b>
Random_Size("CONST");			

<b>PART_NUM_CORNERS</b>	<b>integer</b>		<b>User</b>
Part_Num_Corners(8);			

<b>PART_VAR_CORNERS</b>	<b>integer</b>		<b>User</b>
Part_Var_Corners(2);			

<b>RANDOM_NUM_CORNERS</b>	<b>integer</b>		<b>User</b>
Part_Random_Num_Corners("CONST");			

<b>PART_RANDOM_POS_CORNERS</b>	<b>integer</b>		<b>User</b>
Part_Random_Pos_Corners("REGULAR");			

<b>PART_ROT</b>	<b>float</b>		<b>User</b>
Part_Rot(0.0);			

<b>RANDOM_ROT</b>	<b>string</b>		<b>User</b>
Random_Rot("CONST");			

Diese Befehle sind den gleichnamigen Befehlen im Abschnitt B.1.6 äquivalent, die dortigen Beschreibungen sind auch hier gültig.

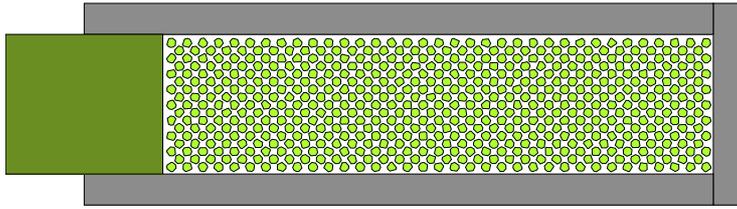
### B.1.8 Beispiel für eine Eingabedatei

Die folgende Datei erzeugt ein Setup, bei dem ein Stempel Teilchen in einem Kolben zusammenpreßt. Dabei ist die Gravitation abgeschaltet. Vor jeder Zeile sind Zeilennummern angegeben, die bei der eigentlichen Datei nicht dort stehen dürfen.

```

01: PARAMETER v1.4
02:
03: # A piston, compressing some particles
04:     Delta_T           0.000015
05:     Time              0           0.2
06:     E_Modul          1E7
07:     Gamma            0.5
08:     Steps_Data       1000
09:     Steps_Checkpoint 100
10:     Basename         data/piston2
11:     G                0
12:
13: DISPLAY
14:     X_Range          -0.35      0.15
15:     Y_Range          -0.05      0.15
16:     Screen           800        0
17:     Interval         100
18:
19: WALL
20:     Offset           -0.3        0.0
21:     Piston_Angle    -0.0
22:     Piston_Force    10000
23:     Piston_V_Max    1
24:     Corners
25:         -0.05      0.01
26:         0.05       0.01
27:         0.05       0.1
28:         -0.05      0.1
29:     END
30:
31: WALL
32:     Offset           0.0         0.0
33:     Corners
34:         -0.3       -0.01
35:         0.1        -0.01
36:         0.1        0.01
37:         -0.3       0.01
38:     END
39:
40: WALL
41:     Offset           0.0         0.11
42:     Corners
43:         -0.3       -0.01
44:         0.1        -0.01
45:         0.1        0.01
46:         -0.3       0.01
47:     END
48:
49: WALL
50:     Offset           0.0         0.0

```



**Abbildung B.8:** Die Abbildung zeigt das System, das `gc2d` aus der vorgestellten Eingabedatei erzeugt. Die starren Wände erscheinen grau, die Partikel sind hellgrün und der Stempel dunkelgrün dargestellt.

```

51:      Corners
52:          0.1      -0.01
53:          0.12     -0.01
54:          0.12     0.12
55:          0.1      0.12
56:      END
57:
58: GENERATE_BLOCK
59:     X_Range          -0.247   0.098
60:     Y_Range          0.012   0.098
61:     Grid_Distance    0.005   0.005
62:     Part_Var_Corners 1
63:     Part_Num_Corners 8
64:     Part_Rot         1
65:     Part_Size        0.003   0.003
66:     Grid_Type        FCC
67:     Random_Size      CONST
68:     Random_Rot       EQUAL
69:     Random_Num_Corners EQUAL
70:     Random_Pos_Corners RAND1
71: END_OF_FILE

```

Die erste Zeile ist fest vorgeschrieben, die zweite Zeile bleibt leer, da keine `gci`-Datei vorgegeben wird. Zeile 3 enthält einen Kommentar, der ignoriert wird. In den Zeilen 4-11 folgen dann Befehle, die zur Sektion `GENERAL` gehören. Es werden der Zeitschritt, der Name für die Ausgabedateien und auch der Wert für  $g$  gesetzt. Das Schlüsselwort `DISPLAY` in Zeile 13 weist `gc2d` an, daß jetzt Befehle folgen, die die Bildschirmdarstellung steuern. Zeile 16 gibt zum Beispiel an, daß das Bildschirmfenster 800 Pixel breit sein soll, die Höhe des Fensters wird dynamisch angepaßt.

Ab Zeile 19 wird der Stempel beschrieben, der die Partikel komprimiert. Die Zeilen 21-23 beschreiben dabei die Eigenschaften des Stempels, die Zeilen 24-29 seine Geometrie. Die Zeilen 31-56 dienen dazu, drei weitere Wände festzulegen. Ab Zeile 58 bis Zeile 70 wird eine Gruppe von Teilchen beschrieben, die auf einem Gitter (Zeile 66) angeordnet werden. Zeile 71 schließt die Datei ordnungsgemäß ab.

## B.2 Eingabedateien für Kollisionen (gci)

Die Dateien vom Type `gci` enthalten die Beschreibungen für Kollisionen, die sich aus der zugehörigen `gsi`-Datei ergeben. Von der Struktur her ist eine `gci`-Datei genauso aufgebaut wie eine `gsi`-Datei, allerdings gibt es nur ein Objekt mit dem Namen **COLLISION**. Eine typische (stark gekürzte) Datei sieht in etwas so aus:

```
PARAMETER v1.4
Wed May 24 18:31:54 2000 --- doublon --- schinner

COLLISION
Particles      3512 3473
Summ_Friction -0.03836233163280577951
Old_Area       0.00000000001731635668

COLLISION
Particles      3511 3496
Summ_Friction -0.00175875105087826668
Old_Area       0.00000000000989987259

# hier wurde gekürzt

COLLISION
Particles      1 2
Summ_Friction -0.07117878025238574824
Old_Area       0.00000000001401684324
END_OF_FILE
```

In jeder dieser Sektionen werden die Daten des letzten Zeitschritts gespeichert, die zur Berechnung der Kräfte aus einer Kollision notwendig sind. Alle diese Befehle gehören zur Kategorie **SYSTEM**. Es ist nicht sinnvoll, eine derartige Datei von Hand zu erstellen; auch ein Perl-Interface für diesen Dateityp gibt es nicht. Stattdessen sollte eine `gsi`-Datei nur aus einer `gso`-Datei (siehe Anhang C.1) erstellt werden.

Der Identifikationstext in der zweiten Zeile der Datei muß identisch sein mit demjenigen aus der zugehörigen `gsi`-Datei. So soll gewährleistet werden, daß beide Dateien wirklich zusammengehören und auch vom selben Programmlauf stammen.

<b>PARTICLES</b>	<b>integer</b>	<b>integer</b>	<b>System</b>
(?, ?);			

Hier werden die zwei Identifikationsnummern der Teilchen angegeben, für die die Kollisionsdaten gültig sind. Wird keine Kollision zwischen den Teilchen mit den angegebenen Nummern gefunden, so meldet `gc2d` dies und ignoriert die entsprechende Sektion.

<b>SUMM_FRICTION</b>	<b>float</b>	<b>System</b>
(0.0);		n

Diese Zeile enthält den Wert, der zur Berechnung von  $F_{||}(t_c - \Delta t)$  notwendig ist (siehe Abschnitt 2.2.2).

<b>OLD_AREA</b>	<b>float</b>	<b>System</b>
(0.0);		$m^2$

Diese Zeile enthält den Wert, der zur Berechnung von  $dA$  notwendig ist (siehe Abschnitt 2.2.2).

## Anhang C

# Format der Ausgabedateien

### C.1 Ausgabedateien für die Systembeschreibung (`gso`) und die Kollisionen (`gco`)

Neben den in den Anhängen C.2, C.3 und C.4 beschriebenen Ausgabeformaten, die dazu dienen, im Hinblick auf physikalische Größen ausgewertet zu werden, schreibt `gc2d` auch noch eine zweite Kategorie von Dateien. Die Aufgabe dieser Files ist es, den aktuellen Stand der Simulation so abzuspeichern, daß an dieser Stelle weitergerechnet werden kann. Dies bietet verschiedene Vorteile:

1. Muß der aktuelle Lauf aus irgendeinem Grund (Stromausfall, Systemwartung, etc.) vorzeitig abgebrochen werden, so kann die Simulation an diesem Punkt wiederaufgesetzt werden.
2. Man kann einmalig eine Konfiguration berechnen, die dann für verschiedenen weitere Simulationen als Ausgangsbasis dient.

Es ist also sinnvoll, daß man diese Ausgabedateien im Format der Eingabedateien generiert.

<code>gso</code> -Datei	Format einer <code>gsi</code> -Datei
<code>gco</code> -Datei	Format einer <code>gci</code> -Datei

Hier gewinnt auch der in den Anhängen B.1 und B.2 beschriebene Identifikationsstring seine Bedeutung. `gc2d` schreibt in die zweite Zeile beider Dateien eine Kombination aus Datum, Uhrzeit, Rechnername und Nutzernamen. Somit ist gewährleistet, daß man nicht aus Versehen eine neue `gso`-Datei zusammen mit einer alten `gco`-Datei gleichen Namens verwendet.

Um eine Simulation durch Nutzung der Dateien `foo.gso` und `foo.gco` wieder zu starten, ist es nur notwendig, diese Dateien umzubenennen.

Folgendes ist allerdings bei `gco`-Dateien zu berücksichtigen:

1. Die Sektion **Generate\_Block** wird nicht geschrieben; stattdessen werden die einzelnen Teilchen ausgegeben.
2. Es kann nicht das Format für das Perl-Interface ausgegeben werden. Sollte es notwendig sein, diese Ausgabedatei über das Perlinterface zu nutzen, kann man die Datei mit dem Skript `gsi2pl`<sup>1</sup> die Datei in umwandeln.

<sup>1</sup>Die Datei `gsi2pl` befindet sich auf der beigelegten CDROM.

3. Es ist ratsam, den Namen der Ausgabedatei **Basename** zu ändern, ansonsten besteht die Gefahr, daß die Ausgabedateien durcheinander geraten.

## C.2 Beschreibung der Teilchengeometrie (ggf)

Dateien mit der Endung `ggf` enthalten die Beschreibung der Geometrie aller in der Simulation existierenden Teilchen. Dabei wird pro Teilchen eine Zeile verwendet. Die Koordinaten der Polygonecken werden so abgespeichert, daß der Teilchenschwerpunkt bei  $(0, 0)$  ist, der Rotationswinkel ist  $\varphi = 0$ .

Diese Datei wird nur einmal für eine Simulation angelegt, für jedes neu hinzukommende Teilchen wird eine zusätzliche Zeile angehängt. Dies bedeutet, daß diese Datei während der Laufzeit der Simulation ständig wachsen kann. Bei der Kompilierung wird der Wert  $N_{max} = \text{MAX\_NUM\_CORNERS}$  vorgegeben. Er gibt die Maximalzahl der Ecken an, die ein Polygon haben darf. Jede Zeile hat dann  $3 + 2 \cdot N_{max}$  Felder.

Feld	Inhalt	Beschreibung
1	Id	Die Teilchennummer. Über diese Zahl kann ein bestimmtes Teilchen in jeder Zeile identifiziert werden.
2	Typ	Der Typ des Teilchens. Normale freie Teilchen haben hier eine 0, Wände eine 2. Weitere Typen sind im Quellcode in der Datei <code>type_particle.h</code> beschrieben.
3	Eckenzahl	In diesem Feld wird gespeichert, wieviele Ecken $N$ das Polygon hat
4 bis $4 + N - 1$	X-Koordinaten	Die X-Koordinaten der Ecken.
$4 + N$ bis $4 + N_{max} - 1$	void	
$4 + N_{max}$ bis $4 + N_{max} + N - 1$	Y-Koordinaten	Die Y-Koordinaten der Ecken.
$4 + N_{max} + N$ bis $4 + 2N_{max} - 1$	void	

Die Algorithmen von `gc2d` kennen keine Beschränkung der Eckenzahl, `MAX_NUM_CORNERS` kann auf jeden beliebigen positiven Wert  $\geq 3$  gesetzt werden. Die Festlegung einer maximalen Eckenzahl dient nur dazu, daß diese Ausgabedatei eine feste Zahl an Feldern hat, so daß sie bequem mit `Matlab` eingelesen werden kann. Dazu muß in `Matlab` nur die Länge der Zeile bekannt sein.

## C.3 Beschreibung der Kräfte (gff)

Dateien mit der Endung `gff` enthalten die Beschreibung der Kräfte im System zu einem bestimmten Zeitpunkt. Für jeden Kontakt wird eine Zeile gespeichert, die folgendes Format hat:

Feld	Inhalt	Beschreibung
1	ID <sub>1</sub>	Die Teilchennummer des ersten Teilchens einer Kollision.
2	ID <sub>2</sub>	Die Teilchennummer des zweiten Teilchens einer Kollision. Ist dieser Wert = -1, so ist die angegebene Kraft eine Einteilchenkraft, z.B. die Gravitation.
3	$x_s$	Die Koordinaten des Schwerpunktes des ersten Teilchens.
4	$y_s$	
5	$x_f$	Der Aufpunkt der Kraft.
6	$y_f$	
7	$f_x$	Die Komponenten der Kraft.
8	$f_y$	

## C.4 Position und Geschwindigkeit (gpf)

Dateien mit der Endung `gpf` enthalten die Beschreibung der Positionen und Geschwindigkeiten im System zu einem bestimmten Zeitpunkt. Für jedes Partikel wird eine Zeile abgespeichert, die folgendes Format hat:

Feld	Inhalt	Beschreibung
1	ID	Die Teilchennummer des Teilchens.
2	$x$	Die Koordinaten des Schwerpunktes des Partikels.
3	$y$	
4	$\varphi$	Den Rotationswinkel des Partikels.
5	$\dot{x}$	Die Translationsgeschwindigkeit des Partikels.
6	$\dot{y}$	
7	$\dot{\varphi}$	Die Rotationsgeschwindigkeit des Partikels.
8	$\ddot{x}$	Die Beschleunigung des Partikels.
9	$\ddot{y}$	
10	$\ddot{\varphi}$	Die Winkelbeschleunigung des Partikels.

## C.5 Zeitabhängige Größen (gtf)

Diese Datei mit der Endung `gtf` enthält die Information, die den Zeitverlauf einiger wichtiger Systemgrößen beschreibt. Jedesmal, wenn ein Satz von `gff`- und `gpff`-Dateien geschrieben wird, wird eine Zeile mit 5 Feldern hinten an dieser Datei angehängt.

Feld	Inhalt	Beschreibung
1	Filenumber	Dieses Feld enthält die laufende Nummer der <code>gff</code> - und <code>gpff</code> -Dateien, die geschrieben wurden.
2	Time	Dieses Feld enthält den Zeitpunkt der Simulation, für die die folgenden Werte gelten.
3	G	Der Wert von $g$ . Dieser kann sich nur ändern, wenn <code>gc2d</code> mit <code>USE_VARIABLE_G</code> kompiliert wurde.
4	reserviert	Das vierte Feld ist für zukünftige Erweiterungen reserviert. Im Moment wird dort der Wert 0.0 geschrieben.
5	Temperatur	Mit <code>gc2d</code> kann die Änderung der Teilchenausdehnung aufgrund einer Temperaturänderung simuliert werden. In diesem Feld wird der aktuelle Skalierungsfaktor für die Teilchengröße gespeichert. Wird <code>USE_HEATING</code> nicht genutzt, so steht hier der Wert 1.0.

Mit dem folgenden kurzen **Matlab**-Skript kann man sich also den Verlauf von Temperatur und Erdbeschleunigung darstellen:

```
load foo.pgl;
t=foo(:, 2);      # Vektor mit den Zeitpunkten
g=foo(:, 3);      # Vektor mit den Erdbeschleunigung
T=foo(:, 5);      # Vektor mit den Temperaturen

hold on
plot(t, g);       # g(t)
plot(t, T);       # T(t)
```

## C.6 Trajektorien (gp1)

Diese Ausgabedatei mit der Endung `gp1` erlaubt es, in sehr einfacher Form die Trajektorien der einzelnen Partikel darzustellen. Da diese Datei sehr schnell sehr groß werden kann, gleichzeitig aber nicht bei jeder Simulation benötigt wird, kann man `gc2d` auch so kompilieren (siehe Anh. E.2), daß diese Funktionalität nicht enthalten ist.

Die Datei ist zeilenweise orientiert; je Zeitschritt wird nur eine Zeile zur Datei hinzugefügt. Die ersten beiden Zahlen einer Zeile enthalten den Zeitschritt und den Wert von  $g$ . Danach folgt für jedes Teilchen eine Sechsergruppe von Zahlen. Folgende Werte werden pro Teilchen abgespeichert, wobei  $i = 1 \dots N - 1$ , wenn  $N$  Teilchen simuliert werden.

Feld	Inhalt	Beschreibung
1	t	Der Zeitpunkt, zu dem diese Zeile geschrieben wurde.
2	g	Der Wert der Erdbeschleunigung, der zu diesem Zeitpunkt gültig war.
$3 + 6i$	Id	Die Teilchennummer. Über diese Zahl kann ein bestimmtes Teilchen in jeder Zeile identifiziert werden.
$4 + 6i$	Typ	Der Typ des Teilchens. Normale freie Teilchen habe hier eine 0, Wände eine 2. Weitere Typen sind im Quellcode in der Datei <code>type_particle.h</code> beschrieben.
$5 + 6i$	x	Die X-Koordinate des Schwerpunktes.
$6 + 6i$	y	Die Y-Koordinate des Schwerpunktes.
$7 + 6i$	$v_x$	Die X-Komponente des Geschwindigkeitsvektors.
$8 + 6i$	$v_y$	Die Y-Komponente des Geschwindigkeitsvektors.

Ist die Teilchenzahl während der Simulation unverändert, so läßt sich mit **Matlab** besonders leicht die Trajektorie aller Teilchen bestimmen:

```
load foo.pgl;
t=foo(:, 1);      # Vektor mit den Zeitpunkten
x=foo(:, 5:6:end); # Matrix mit allen X-Koordinaten
y=foo(:, 6:6:end); # Matrix mit allen Y-Koordinaten

plot(x, y);       # Trajektorie
plot(t, x);       # x(t)
plot(t, y);       # y(t)
```

## Anhang D

# Ein Präprozessor für die Eingabedateien

Im Anhang B.1 wird der Aufbau der Beschreibungssprache für granulare Simulationen beschrieben. Wie bereits dort erwähnt, enthält diese Sprache aber keine Kontrollstrukturen, so daß das Schreiben einer derartigen Eingabedatei nicht sehr komfortabel ist. In ein Programm zur physikalischen Simulation von Granulaten allerdings noch eine vollständige Programmiersprache einzubauen, erscheint nicht sinnvoll. Vielmehr liegt es nahe, diese Dateien nicht direkt per Hand zu schreiben, sondern über einen Präprozessor die endgültige Form zu erzeugen. Hier kamen verschiedenen Möglichkeiten in Betracht.

1. Der Präprozessor der Programmiersprache C ist weit verbreitet und hat eine einfache Syntax. Allerdings fehlen Elemente wie Schleifen und Unterprogramme. Insgesamt ist dieser Präprozessor zu unflexibel und zu sehr auf eine vollständige Programmiersprache ausgerichtet. Daher wurde diese Variante nicht genutzt.
2. M4 ist ein Makroprozessor, der standardmäßig auf jedem UNIX-System vorhanden ist. Verschiedene Kontrollstrukturen fehlen, können aber über interne Makros nachgebildet werden. Für einfache Erweiterungen ist M4 geeignet. Bei komplexeren Aufgabestellungen wird die Syntax von M4 allerdings sehr schwer überschaubar. Ein Interface zu diesem Präprozessor ist in `gc2d` noch enthalten, dient allerdings nur noch der Kompatibilität mit alten Eingabedateien. Die Nutzung dieses Interfaces ist nicht mehr empfehlenswert, entsprechend wird diese Möglichkeit auch nicht weiter beschrieben.
3. Die Skriptsprache Perl ist auf so gut wie jedem UNIX-System anzutreffen. Perl ist eine vollständige Programmiersprache, es gibt keinerlei Einschränkungen und sie läßt sich durch die Benutzung von Bibliotheken beliebig erweitern. Ein Interface zu dieser Sprache ist in `gc2d` implementiert und wird empfohlen.

### D.1 Das Perl-Interface

Die Zusammenarbeit von `gc2d` mit Perl wurde so gestaltet, daß man mit Hilfe der Bibliothek `gsi.pm`<sup>1</sup> ein Perlprogramm [153] erstellt, welches von `gc2d` dann beim Start aufgerufen wird. Dazu startet es den Perlinterpretierer und übergibt ihm die Eingabedatei. Perl selbst führt dann das Programm aus. Als Ausgabe wird die normale `gsi`-Syntax produziert. Diesen Text liest `gc2d` dann

<sup>1</sup>Die Datei `gsi.pm` befindet sich auf der beigelegten CDROM.

ein und verwendet ihn zum Erstellen der Simulation. Erstellt man also eine Eingabedatei mit Hilfe dieser Bibliothek, so schreibt man ein Perl-Programm, das die eigentliche Eingabedatei erstellt. Entsprechend sind folgende Befehlssequenzen

```
doublon:~/gc2d> perl foo.gsi.pl > foo.gsi
doublon:~/gc2d> gc2d -i foo.gsi
```

oder

```
doublon:~/gc2d> gc2d -i foo.gsi.pl
```

gleichwertig.

Zu Erläuterung nun ein einfaches Beispiel: Das Perlprogramm

```
use gsi;                # Nutzung der Bibliothek

$K = 1;                # Ein beliebiger Parameter

General;               # Kopfzeile der gsi-Datei
Delta_T(0.000001);     # Zeitschritt
Mue_Dynamisch(0.1*$K); # mue mit Hilfe von K berechnen
Basename("output_$K"); # K im Dateiname einbauen
Email("foo@test.bar"); # eine eMail-Adresse
```

hat als Ausgabe:

```
PARAMETER v1.4

# gsi.pm Version 0.99
# original version by Heiko Bauke, extended by A. Schinner
# May 2000 Magdeburg

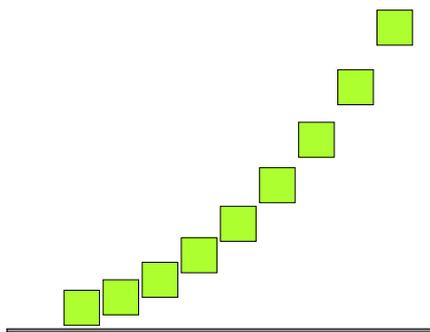
      Delta_t           1.000000e-06
      Mue_Dynamisch     1.000000e-01
      Basename          output_1
      Email              foo@test.bar
```

Die Grundfunktion für einen beliebigen Befehl entspricht dabei im wesentlichen einem Programm mit der Form:

```
sub Befehl{
  my $parameter =$_[0];
  printf("Befehl %s\n", $parameter);
}
```

Zusätzlich wird noch überprüft, ob der Befehl in dieser Sektion überhaupt zulässig ist und ob die Parameter gewissen Bedingungen entsprechen. Die Befehlsnamen selbst sind im Anhang B.1 angegeben, wobei zu beachten ist, daß Perl Groß- und Kleinschreibung unterscheidet und die einzelnen Befehle mit einem Semikolon abgeschlossen werden. Zusätzlich ist noch zu beachten:

1. Als erster Befehl kommt immer **GENERAL**. Er erzeugt den nötigen Header für die `gsi`-Dateien



**Abbildung D.1:** Beispielsetup erzeugt mit dem Perl-Interface.

2. Der Befehl **Comment** erzeugt eine Kommentarzeile, beginnend mit einem \$.
3. Der Befehl **Macro** ist ein reiner Dummy, er tut nichts. Er dient nur zur optischen Strukturierung der Eingabedatei.
4. Dem Befehl **Corners** übergibt man zwei Listen mit den X- bzw. den Y-Koordinaten der Ecken.
5. Findet `gc2d` einen Fehler in der Eingabedatei, so gibt es die Zeilennummer der `gsi`-Datei an, nicht des Perlfiles!

Als Beispiel nun eine Eingabedatei in Perl-Syntax, es sind Zeilennummern mit angegeben, die in der eigentlichen Datei nicht vorhanden sein dürfen.

```

01: use gsi;
02:
03: General;
04: Delta_T(0.000001);
05: E_Modul(1E8);
06: Gamma(0.1);
07: Basename("demo");
08: Mue_Dynamisch(0.6);
09: Steps_Data(1000000);
10: Steps_Checkpoint(1000000);
11: Steps_Setup(1000000);
12: Time(0, 0.175);
13: Email("none");
14:
15: Display;
16: Screen(600, 0);
17: Interval(1000);
18: X_Range(-0.1, 1.1);
19: Y_Range(-0.1, 1.1);
20:
21: Wall;
22: Corners([-0.1, 1.1, 1.1, -0.1],[-0.01, -0.01, 0., 0.0]);
23:
24: @ecken_x=(0.0, 0.1, 0.1, 0.0);
25: @ecken_y=(0.0, 0.0, 0.1, 0.1);
26:

```

```

27: for ($cnt=1;$cnt<10;$cnt++){
28:   Particle;
29:   Com($cnt*0.11, 0.05+$cnt*$cnt*0.01);
30:   Corners(\@ecken_x, \@ecken_y);
31: }
32: EOF;

```

Zeile 1 bindet die notwendige Perlbibliothek ein, mit Zeile 3 wird der Header geschrieben. Zeile 4 bis 21 entsprechen direkt den `gsi` Befehlen, nur daß die Parameter in Klammern geschrieben werden, Strings auch noch in Anführungszeichen. Zeile 22 zeigt den etwas veränderten Befehlsaufbau bei der Angabe von Ecken. In Zeile 24 und 25 werden die Ecken für die Teilchen vordefiniert. In Zeile 27 beginnt eine Schleife mit der Zählervariablen `$cnt`. Zeile 29 setzt dann das Partikel auf eine Position in Abhängigkeit des Zählers. Zeile 30 beschreibt die Ecken, wobei die Liste der Ecken als Referenz übergeben wird. Die erzeugte Simulation ist in Abbildung D.1 gezeigt.

In Kombination mit der Kommandozeilenoption `-P 'string'` ist es auch noch möglich, dem Perlskript zusätzliche Parameter zu übergeben. Beim Aufruf des Perlinterfaces gibt `gc2d` den String als zusätzliche Kommandozeilenoption mit an. In Perl selbst stehen diese Werte dann im Array `@ARGV` zur Verfügung. Wird also `gc2d` z.B. mit der Option `-P "0.6 20"` aufgerufen, können dieses Werte dann folgendermaßen in der Steuerdatei weiterverarbeitet werde:

```

$mue = $ARGV[0];
$N   = $ARGV[1];

```

Es wäre dann `$mue=0.6` und `$N=20`.

Eine hervorragende Einführung in Perl bietet das "Kamel Buch" von Wall et al. [166]. Es existiert auch noch eine Datei `xemacs-addon`<sup>2</sup> für den Editor XEmacs. Fügt man diese Datei in die private `.emacs`-Datei ein, so beherrscht der Editor noch Syntaxhighlighting für diese Eingabedateien.

## D.2 Makrobibliotheken

Um die Vorteile des Perlinterfaces (siehe Anhang D.1) noch bequemer nutzen zu können, wurden weitere Bibliotheken erstellt. Mit diesen Bibliotheken ist es möglich mit einem einzigen Befehl spezielle Teilchen, komplizierte Teilchenanordnungen oder aufwendige Wandgeometrien aufzusetzen. Die einzelnen Bibliotheken sollen im folgenden erklärt werden. Für die Beschreibung aller Befehle gelten folgende Abkürzungen:

- $x$  X-Koordinate des Teilchenschwerpunktes
- $y$  Y-Koordinate des Teilchenschwerpunktes
- $\varphi$  Rotationswinkel des Teilchens
- $N$  Die Zahl der Ecken bei einer Rundung

## D.3 Kreise

Mit den Befehlen dieser Bibliothek `spheres.pm`<sup>3</sup> kann man einfach kreisförmige Teilchen aufsetzen. Da bei `gc2d` immer mit Polygonen gearbeitet wird, sind werden Kreise durch regelmäßige  $N$ -Ecke approximiert.

<sup>2</sup>Die Datei `xemacs-addon` befindet sich auf der beigelegten CDROM.

<sup>3</sup>Die Datei `spheres.pm` befindet sich auf der beigelegten CDROM.



**Abbildung D.2:** Abgebildet sind hier typische Teilchenformen, die in den Abschnitten D.3, D.5 und D.4 beschrieben werden. a) sphere b) ellipse c) rectangle d) rectangle\_cut e) rectangle\_round

```
gc2d_sphere(x, y, r, N);
```

Erzeugt wird ein kreisförmiges Teilchen mit dem Radius  $r$ .

```
gc2d_sphere_wall(x, y, r, N);
```

Erzeugt wird eine kreisförmige Wand mit dem Radius  $r$ .

```
gc2d_sphere7(x, y, r);
```

```
gc2d_sphere10(x, y, r);
```

```
gc2d_sphere20(x, y, r);
```

```
gc2d_sphere50(x, y, r);
```

Diese Befehle entsprechen dem einfachen Befehl für kreisförmige Teilchen, wobei  $N$  im Befehlsnamen kodiert ist.

## D.4 Ellipsen

Mit den Befehlen dieser Bibliothek `ellipses.pm`<sup>4</sup> kann man einfach ellipsenförmige Partikel erzeugen.

```
gc2d_ellipse(x, y, r1, r2, N);
```

Die Werte  $r_1$  und  $r_2$  geben die Längen der Halbachsen der Ellipse dar, in die das  $N$ -Eck eingeschrieben wird.

```
gc2d_ellipse7(x, y, r);
```

```
gc2d_ellipse10(x, y, r);
```

```
gc2d_ellipse20(x, y, r);
```

```
gc2d_ellipse50(x, y, r);
```

Diese Befehle entsprechen dem einfachen Befehl für elliptische Teilchen, wobei  $N$  im Befehlsnamen kodiert und  $\varphi = 0$  ist.

<sup>4</sup>Die Datei `ellipses.pm` befindet sich auf der beigelegten CDROM.

## D.5 Rechtecke

Der Bibliothek `rectangles.pm`<sup>5</sup> stellt Befehle bereit, um rechteckige Teilchen zu generieren. Die Rechtecke mit abgeschrägter und abgerundeter Ecke sind bei regulären Anordnungen numerisch etwas "gutmütiger" als die einfachen Rechtecke, da die Schnittpunkte der Umrisse eindeutig definiert sind. Dies ist bei Rechtecken, deren Kanten exakt uebereinanderliegen, nicht der Fall.

```
gc2d_rectangle(x, y, a, b, φ);
```

Die Werte  $a$  und  $b$  geben die Breite und die Höhe des Rechtecks an.

```
gc2d_rectangle_cut(x, y, a, b, φ, c);
```

Die Werte  $a$  und  $b$  geben die Breite und die Höhe des Rechtecks an. Bei dieser Form des Rechtecks werden die Ecken abgeschrägt, man erhält also ein Achteck, die Kanten werden auf die Länge  $a \cdot (1 - 2c)$  bzw.  $b \cdot (1 - 2c)$  verkürzt.

```
gc2d_rectangle_round(x, y, a, b, φ, r, N);
```

Die Werte  $a$  und  $b$  geben die Breite und die Höhe des Rechtecks an. Bei dieser Form des Teilchens werden die Ecken abgerundet. Der Wert  $r$  gibt den Radius des Viertelkreises an, der in jede Ecke eingepaßt wird,  $N$  die Zahl seiner Ecken.

```
gc2d_square(x, y, a, φ);
```

```
gc2d_square_cut(x, y, a, φ, c);
```

```
gc2d_square_round(x, y, a, φ, r, N);
```

Diese drei Befehle generieren quadratische Teilchen; sie entsprechen den Kommandos der Rechtecke für den Fall  $a = b$ .

## D.6 Reguläre Haufen

Der Bibliothek `heap.pm`<sup>6</sup> stellt Funktionen bereit, um reguläre Sandhaufen aufzusetzen.

Die ersten vier Parameter sind bei den folgenden Routinen gleich,  $x_l$  und  $x_r$  geben den linken und rechten Rand des Haufens an,  $y_0$  seine Basis sowie  $\psi$  den "Böschungswinkel".

```
gc2d_heap_hexagonal_spheres(x_l, x_r, y_0, ψ, a);
```

```
gc2d_heap_rectangular_spheres(x_l, x_r, y_0, ψ, a);
```

Mit diesen Befehlen können Haufen mit kreisförmigen Partikeln generiert werden. Der Parameter  $r$  gibt dabei den Radius der Scheiben an. Auch hier ist es sinnvoll, bei der hexagonalen Variante die Bedingung  $\psi < \arctan(2)$  einzuhalten.

```
gc2d_heap_hexagonal_rectangles(x_l, x_r, y_0, ψ, a, b);
```

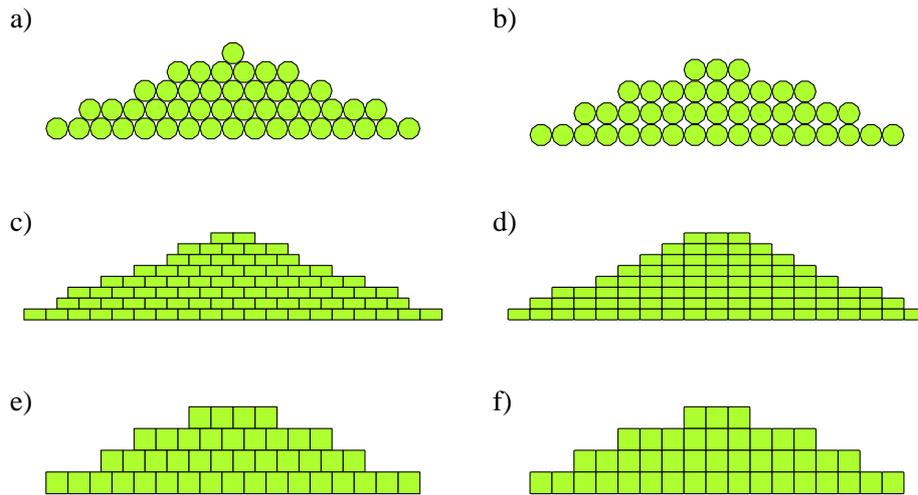
```
gc2d_heap_rectangular_rectangles(x_l, x_r, y_0, ψ, a, b);
```

Die Parameter  $a$  und  $b$  bestimmen die Breite und die Höhe der eingesetzten Teilchen. Bei der hexagonalen Variante des Befehls ist es sinnvoll, die Bedingung  $\psi < \arctan(\frac{2b}{a})$  einzuhalten, da sonst überhängende Partikel entstehen.

```
gc2d_heap_hexagonal_squares(x_l, x_r, y_0, ψ, r, N);
```

<sup>5</sup>Die Datei `rectangles.pm` befindet sich auf der beigelegten CDROM.

<sup>6</sup>Die Datei `heap.pm` befindet sich auf der beigelegten CDROM.



**Abbildung D.3:** Hier sind die verschiedenen Anordnungen von regulären Haufen dargestellt, die in Abschnitt D.6 beschrieben werden. a) `hexagonal_spheres` b) `rectangular_spheres` c) `hexagonal_rectangles` d) `rectangular_rectangles` e) `hexagonal_squares` f) `rectangular_squares`

```
gc2d_heap_rectangular_squares( $x_l, x_r, y_0, \psi, r, N$ );
```

Diese Befehle sind den Befehlen für die Rechtecke sehr ähnlich, es werden nur Quadrate als Teilchen eingesetzt. Bei der hexagonalen Variante des Befehls ist es sinnvoll, die Bedingung  $\psi < \arctan(2)$  einzuhalten, da sonst überhängende Partikel entstehen.

Beispiele für die verwendeten Befehle sind in Abbildung D.3 abgebildet. Alle diese Routinen setzen auf zwei Kernroutinen auf, die die Funktionalität für ein Rechtecks- bzw. ein Dreiecksgitter bereitstellen.

```
gc2d_heap_hexagonal( $x_l, x_r, y_0, \psi, \Delta x, \delta y, \text{Perl-Kommando}$ );
```

```
gc2d_heap_rectangular( $x_l, x_r, y_0, \psi, \Delta x, \delta y, \text{Perl-Kommando}$ );
```

Die genaue Funktionalität dieser beiden Routinen ist dem Quellcode der Bibliothek zu entnehmen, eine genauere Erklärung des letzten Parameters würde zu sehr in programmiertechnische Details führen.

## D.7 Rotierende Trommeln

Mit der Bibliothek `drum.pm`<sup>7</sup> kann man eine rotierende Trommel simulieren. Dabei drehen sich die Wandteilchen nicht, haben aber eine virtuelle Geschwindigkeit an der Oberfläche. Das System ist dann mit einer feststehenden Trommel, die mit einer Art Fließband ausgekleidet ist, zu vergleichen.

```
gc2d_drum( $r, d, x, y, v, N$ );
```

$r$  gibt den Radius der Trommel an,  $d$  die Wandstärke.  $x$  und  $y$  legen das Zentrum der Trommel fest.

<sup>7</sup>Die Datei `drum.pm` befindet sich auf der beigelegten CDROM.

Die Geschwindigkeit  $v$  bestimmt den Betrag der Tangentialgeschwindigkeit an der Trommelinnenseite. Der Wert  $N$  gibt an, aus wievielen Polygonen die Trommel konstruiert werden soll.

## D.8 Einlesen von XFig-Dateien

Die Bibliothek `read_xfig.pm`<sup>8</sup> schafft eine Schnittstelle zum Programm XFig<sup>9</sup>. Damit ist es möglich, ein System mit XFig zu zeichnen und es dann mit `gc2d` zu simulieren.

```
gc2d_read_xfig(Dateiname);
```

Dabei muß die XFig-Datei einigen Anforderungen genügen, um sinnvoll simuliert werden zu können:

1. Es werden nur Objekte simuliert, die als Polygone abgespeichert sind. Dies werden ausschließlich mit den Befehlen `POLYGON drawing`, `POLYLINE drawing`, `rectangular BOX drawing` und `Regular Polygon` erzeugt. Alle anderen Objekte werden ignoriert.
2. Ist ein Objekt zu 100% mit Farbe gefüllt, so wird es als freies Teilchen simuliert, ansonsten als Wand.
3. Da XFig sein Koordinatensystem anders orientiert hat als `gc2d`, werden die Y-Werte der Zeichnung mit -1 multipliziert, man zeichnet also im vierten Quadranten.
4. Der Zeichenmaßstab ist 10:1.

---

<sup>8</sup>Die Datei `read_xfig.pm` befindet sich auf der beigelegten CDROM.

<sup>9</sup>Die verwendete Version von XFig ist "Xfig 3.2 patchlevel 2 (Protocol 3.2)".

---

## Anhang E

# Interna von gc2d

### E.1 Kommandozeilenoptionen von gc2d

Das Programm `gc2d` hat einige Kommandozeilenoptionen, mit denen sich der Ablauf des Programms beeinflussen läßt.

<code>-h</code>	Eine kurze Hilfe zu den Kommandozeilenoptionen wird ausgegeben.
<code>-v</code>	Das Programm gibt mehr Statusinformationen aus als normalerweise. (verbose)
<code>-g</code>	Aktivierung der graphischen Oberfläche.
<code>-s</code>	Hält die graphische Ausgabe an, man kann schrittweise die Simulation ablaufen lassen. Diese Option wirkt nur in Kombination mit <code>-g</code> .
<code>-l</code>	Ignoriert den vorhandenen Lockfile. Wird diese Option angegeben, ist es möglich, bereits vorhandene Simulationsergebnisse zu überschreiben!
<code>-t Zahl</code>	Mit diesem Befehl kann man vorgeben, wieviele Threads parallel laufen sollen.
<code>-i Dateiname</code>	Diese Option gibt den Dateinamen der zu simulierenden <code>gsi</code> -Datei an. Wird diese Option nicht gegeben, ist <code>default.gsi</code> als Standardname vorgegeben.
<code>-P Text</code>	Der mit diesem Befehl angegebene String wird dem Perlinterface als Parameter mitgegeben. Zur Auswertung siehe Anhang D.1.

### E.2 Konfiguration bei der Kompilierung

Einige Freiheiten in der Manipulation von Parametern werden nicht bei jeder Simulation benötigt. Ein Beispiel hierfür ist die Veränderbarkeit von `g`. In den meisten Fällen wird man wohl `g = const` annehmen, eine Neuberechnung von `g` würde nur Rechenzeit verschwenden.

Außerdem ist es erforderlich, daß `gc2d` auch auf Rechnern zu übersetzen ist, die bestimmte Bibliotheken für die Grafik oder die Parallelisierung nicht haben. Aus diesem Grund ist es möglich, durch bestimmte Definitionen in der Datei `gstd.h` nur die benötigten Programmfähigkeiten zu kompilieren und einige Parameter anzupassen. Dies hilft, unnötige Rechenzeit zu sparen und ermöglicht eine verbesserte Portabilität.

USE_THREADS	Das Programm wird so übersetzt, daß Threads zur Parallelisierung verwendet werden. Die Kommandozeilenoption "-t" wird aktiviert.
USE_SYSTEMINFO	Beim Start werden die systemspezifischen Daten ausgegeben. Dieser Parameter ist nur auf SOLARIS-Rechnern sinnvoll.
USE_GTK	Das Programm wird so übersetzt, daß die GTK-Bibliothek zur graphischen Darstellung verwendet wird. Die Kommandozeilenoption "-g" und "-s" werden aktiviert.
USE_HEATING	Das Programm wird so übersetzt, daß eine Größenveränderung der Partikel aufgrund einer Temperaturänderung simuliert werden kann. Die gsi-Befehle HEAT_* werden aktiviert.
USE_LOCALSCALE	Das Programm wird so übersetzt, daß jedes Partikel einen eigenen Größenskalierungsfaktor berücksichtigt. Dieses Feature wird allerdings noch nicht genutzt; der Skalierungsfaktor ist standardmäßig gleich eins und wird nicht verändert.
USE_VARIABLE_G	Das Programm wird so übersetzt, daß eine Änderung des Wertes von g möglich ist. Die gsi-Befehle G_* werden aktiviert, der Befehl G ändert die Zahl seiner Parameter.
USE_DUMPLINE	Die Ausgabe von Dateien des Typs .gpl wird aktiviert. Der gsi-Befehl Steps_Dumpline wird aktiviert. Im allgemeinen ist es sinnvoll, diese Ausgabedatei nicht zu erzeugen, da sie sehr schnell sehr groß werden kann.
NICENESS	Auf einem UNIX-System versucht gc2d seine Priorität soweit herunterzusetzen, daß es den laufenden Betrieb nicht stört. Der mögliche Bereich ist 0 bis 19, bei 19 ist gc2d besonders "nett" zu anderen Programmen, bei 0 versucht es möglichst viel Rechenzeit zu bekommen.
MAX_NUM_THREADS	Dieser Wert gibt die maximale Zahl von Threads an, die über die Kommandozeile aktiviert werden können.
MAX_NUM_CORNERS	Dieser Wert gibt die maximale Zahl der Ecken für ein Polygon an. Gleichzeitig bestimmt er die Zahl der Felder in den ggf-Dateien. Dieser Wert muß bei der Auswertung der ggf-Dateien durch Matlab berücksichtigt werden, der Standard ist 64.

### E.3 Graphische Übersicht der Funktionsaufrufe

In Abbildung E.1 sind die Funktionsaufrufe von drei wichtigen Unterprogrammen dargestellt. Von der Routine main sind die wichtigen Aufrufe der Hauptschleife dargestellt; Initialisierung, Grafik und Dateimanagement sind nicht dargestellt. Anhand der Funktion initialize\_particle kann man erkennen, welche Vorbereitungsmaßnahmen getroffen werden müssen, wenn ein neues Teilchen in die Simulation eingefügt wird. Die graphische Darstellung der Routine closest\_features läßt sehr schön die einzelnen logischen Ebenen des Algorithmuses erkennen,

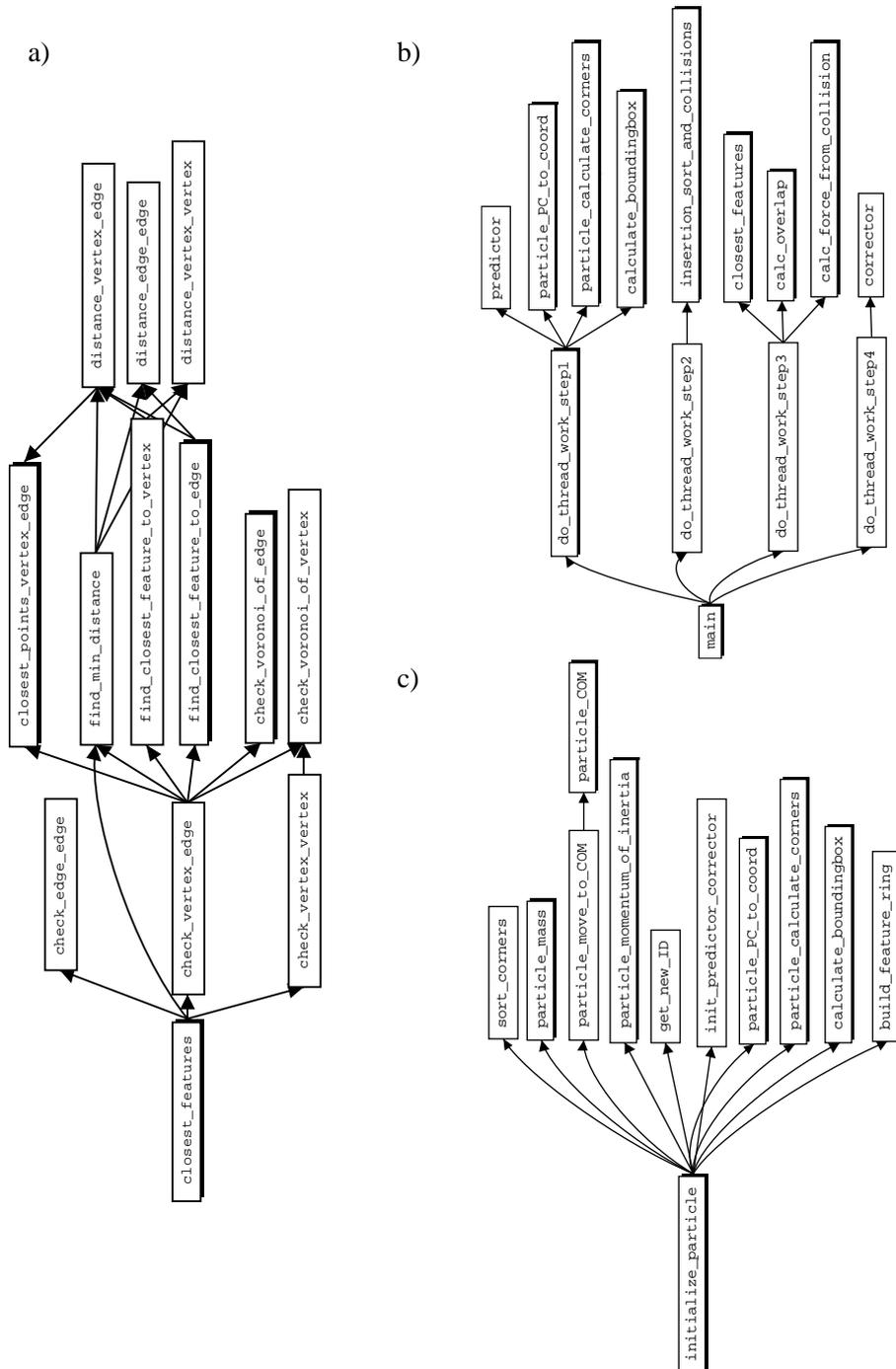
### E.4 Strukturen

Alle logischen Datengruppen wurden bei der Entwicklung von gc2d in Strukturen zusammengefaßt. So ist eine Erweiterung der Programmfunktionalität deutlich erleichtert. In der folgenden Tabelle sind die wichtigsten Strukturen kurz beschrieben:

Struktur	Dateiname	Beschreibung
particle	type_particle.h	Diese Struktur beschreibt die zentralen Eigenschaften eines Teilchens, wie die Lage der Ecken, Maße, Geschwindigkeit, Beschleunigungen etc.
boundingbox	type_boundingbox.h	Diese Struktur enthält eine Begrenzung einer Bounding-Box.
feature	type_feature.h	Diese Struktur enthält die Beschreibung der Ecken oder Kanten des Teilchens.
collision	type_collision.h	Diese Struktur enthält Daten über die Kollision zweier Teilchen. Diese Struktur wird sowohl für die Kollision zweier Bounding-Boxes als auch der Teilchen selber verwendet.
thread_job	type_threadchunk.h	Mit dieser Struktur wird eine Arbeitseinheit für einen Thread (siehe 2.4.4) verwaltet.
drop_source	type_dropsource.h	Eine Teilchenquelle

Weitere Strukturen für die Grafik oder das Dateimanagement finden sich im Quellcode. Alle Strukturen sind in eigenen Dateien deklariert, die mit `type_` beginnen. Jede dieser Dateien enthält auch alle notwendigen Definitionen, die zur einfachen Nutzung der entsprechenden Datentypen notwendig sind.

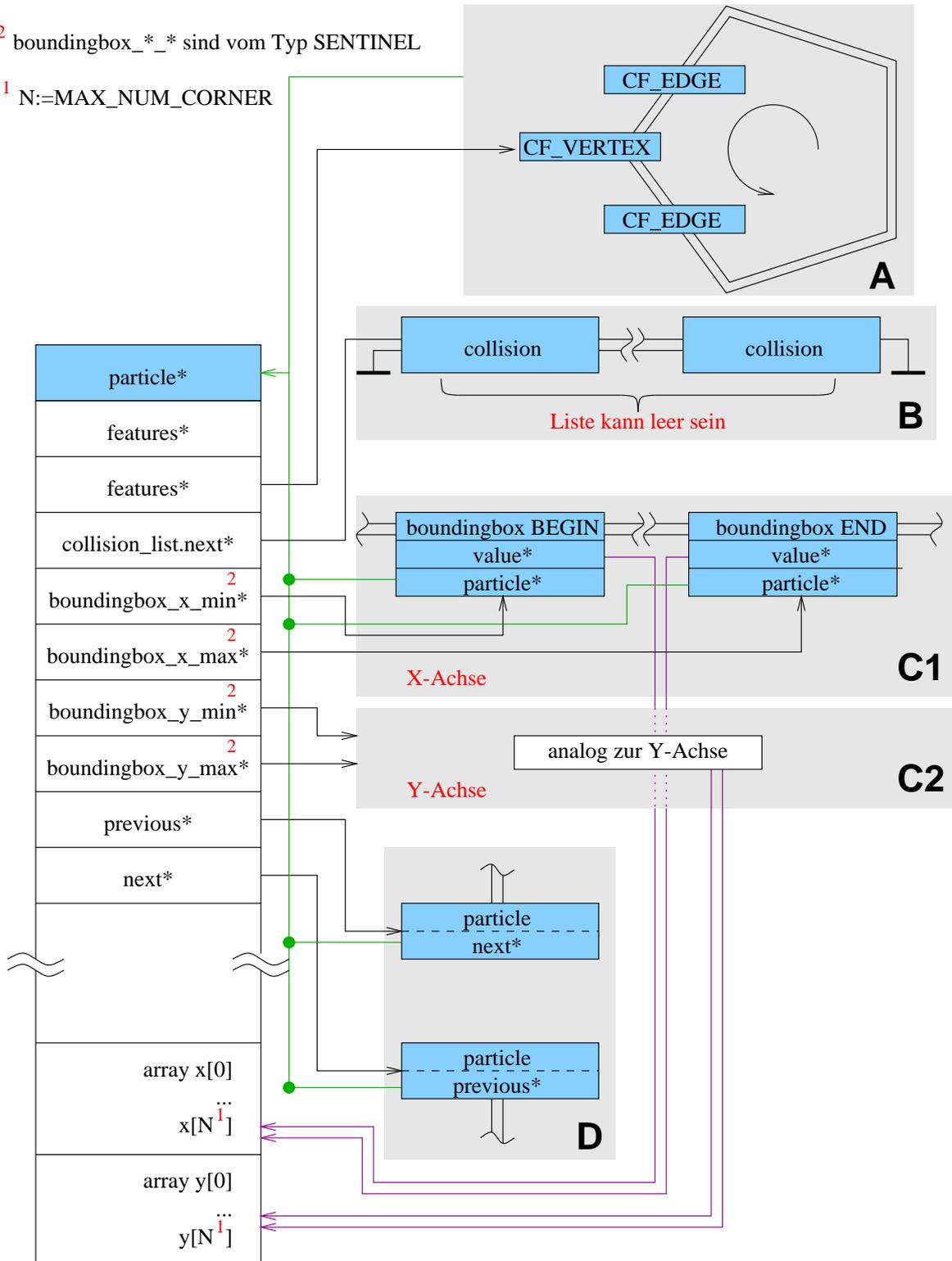
In Abbildung E.2 ist schematisch der Zusammenhang der Strukturen dargestellt. Man erkennt dort beispielsweise die Organisation der Ecken und Kanten als Ring oder daß die Boundingboxes der x- und y-Achse in zwei separaten Listen verwaltet werden.



**Abbildung E.1:** Dargestellt werden die Funktionsaufrufe dreier wichtiger Programmteile: a) `closest_feature` zur Abstandsberechnung b) `initialize_particle` zeigt die notwendigen Aufrufe zur Initialisierung eines einzelnen Sandkorns und c) die Hauptschleife.

<sup>2</sup> boundingbox\_\*\_\* sind vom Typ SENTINEL

<sup>1</sup> N:=MAX\_NUM\_CORNER



**Abbildung E.2:** Zusammenhang der Strukturen innerhalb von `gc2d` : A Ring der Ecken und Kanten, B Liste aller Kollisionen, die zu diesem Teilchen gehören, C1,C2 Liste der Bounding-Box-Begrenzungen für die x- und y-Achse D die Liste aller Teilchen



## Anhang F

# Gear Predictor-Corrector

Gegeben sei die Differentialgleichung  $y^{(l)} = f(t, y, y', y'', \dots, y^{(l-1)})$  und ein  $k < l$ . Dann sei  $\mathbf{y}$  ein Vektor, der  $y$  und die ersten  $k$  Ableitungen von  $y$  enthält, und die Form

$$\mathbf{y} = \begin{pmatrix} y \\ y' \\ y'' \\ \vdots \\ y^{(k)} \end{pmatrix} \quad (\text{F.1})$$

hat.  $B$  sei eine  $(k+1) \times (k+1)$  Matrix, sowie  $\mathbf{c}$  ein Vektor mit der  $(k+1)$  Elementen. Dann kann nach [116] jede Multivalue-<sup>1</sup>Predictor-Corrector-Methode in der Form

$$\mathbf{y}_{n,(0)} = B\mathbf{y}_{n-1} \quad \text{Predictor} \quad (\text{F.2})$$

$$\mathbf{y}_{n,(m+1)} = \mathbf{y}_{n,(m)} + \mathbf{c}G(\mathbf{y}_{n,(m)}) \quad m \geq 0 \quad \text{Corrector} \quad (\text{F.3})$$

geschrieben werden. Das Prinzip des Verfahrens ist es, zuerst mit Hilfe der ersten  $k$  Ableitungen von  $y$  einen neuen Wert für diese Ableitungen vorauszusagen (Predictor). Dann wird mit Hilfe von  $f(t, y, y', y'', \dots, y^{(l-1)})$  an dieser vorhergesagten Stelle eine Korrektur für diese Werte berechnet (Corrector), dies kann  $m$ -mal geschehen. Auffällig ist, daß der Predictor *nicht* von der zu lösenden Differentialgleichung abhängt, sondern nur von  $\mathbf{y}$ .

Für den Predictor beim 6-Value-Gear-Predictor-Corrector geht man von der Taylorentwicklung

$$f(a+h) = \sum_{n=0}^{\infty} \frac{h^n}{n!} f^{(n)}(a) \quad (\text{F.4})$$

<sup>1</sup>Hier werden nur die sog. Multivalue-Verfahren betrachtet. Bei Multistep-Verfahren erfolgt der Predictorschritt mit Hilfe der letzten Schritte der Lösungsverfahren. Im Vergleich zu den Multivalue-Verfahren haben sie den Nachteil, nur sehr schwer für einen „Adaptive Stepsize“ geeignet zu sein. Außerdem müssen vor Beginn der Simulationen z.B. mit Hilfe des Runge-Kutta-Verfahrens Startwerte für die „letzten“ Schritte bestimmt werden, damit das Verfahren korrekt beginnen kann. Zudem ist es sehr aufwendig, während des Laufs die Ordnung des Verfahrens zu ändern. In [116, 117] wird noch erwähnt, daß bei Multivalue meist die Koeffizienten  $\mathbf{c}$  kleiner sind als bei Multistep. Dies kommt der Genauigkeit in Bezug auf Rundungsfehler zugute.

aus. Dabei werden  $f(t)$ ,  $\dot{f}(t)$  bis  $f^{(4)}(t)$  jeweils entwickelt, wobei immer nach  $f^{(5)}$  abgebrochen wird.

$$f(t + \Delta t) = f(t) + \Delta t \frac{df(t)}{dt} + \frac{\Delta t^2}{2} \frac{d^2 f(t)}{dt^2} + \frac{\Delta t^3}{6} \frac{d^3 f(t)}{dt^3} + \frac{\Delta t^4}{24} \frac{d^4 f(t)}{dt^4} + \frac{\Delta t^5}{120} \frac{d^5 f(t)}{dt^5} \quad (\text{F.5})$$

$$\frac{df(t + \Delta t)}{dt} = \frac{df(t)}{dt} + \Delta t \frac{d^2 f(t)}{dt^2} + \frac{\Delta t^2}{2} \frac{d^3 f(t)}{dt^3} + \frac{\Delta t^3}{6} \frac{d^4 f(t)}{dt^4} + \frac{\Delta t^4}{24} \frac{d^5 f(t)}{dt^5} \quad (\text{F.6})$$

$$\frac{d^2 f(t + \Delta t)}{dt^2} = \frac{d^2 f(t)}{dt^2} + \Delta t \frac{d^3 f(t)}{dt^3} + \frac{\Delta t^2}{2} \frac{d^4 f(t)}{dt^4} + \frac{\Delta t^3}{6} \frac{d^5 f(t)}{dt^5} \quad (\text{F.7})$$

$$\vdots \quad (\text{F.8})$$

$$\frac{d^5 f(t + \Delta t)}{dt^5} = \frac{d^5 f(t)}{dt^5} \quad (\text{F.9})$$

Die einzelnen Ableitungen werden dann zeitskaliert, so daß mit  $\mathbf{r}_0 = f(t)$  gilt

$$\mathbf{r}_n = \frac{\Delta t^n}{n!} \frac{d^n \mathbf{r}_0}{dt^n}, \quad (\text{F.10})$$

also  $\mathbf{r}_1(t) = \Delta t \frac{d\mathbf{r}_0}{dt}$ ,  $\mathbf{r}_2(t) = \frac{\Delta t^2}{2} \frac{d^2 \mathbf{r}_0}{dt^2}$  usw. Damit läßt sich dann das Gleichungssystem F.9 in der Form

$$\begin{pmatrix} \mathbf{r}_0^p(t + \Delta t) \\ \mathbf{r}_1^p(t + \Delta t) \\ \mathbf{r}_2^p(t + \Delta t) \\ \mathbf{r}_3^p(t + \Delta t) \\ \mathbf{r}_4^p(t + \Delta t) \\ \mathbf{r}_5^p(t + \Delta t) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 1 & 3 & 6 & 10 \\ 0 & 0 & 0 & 1 & 4 & 10 \\ 0 & 0 & 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{r}_0(t) \\ \mathbf{r}_1(t) \\ \mathbf{r}_2(t) \\ \mathbf{r}_3(t) \\ \mathbf{r}_4(t) \\ \mathbf{r}_5(t) \end{pmatrix} \quad (\text{F.11})$$

schreiben. Dabei ist die Matrix die Pascalsche Dreiecksmatrix, die allgemein bestimmt ist durch

$$P_{i,j} = \begin{cases} 1 & \text{falls } i = j \\ 0 & \text{falls } i < j \\ P_{i,j-1} + P_{i-1,j-1} & \text{sonst} \end{cases} \quad (\text{F.12})$$

mit  $i, j = 1 \dots n$ .

Damit ist der Predictor passend zu Gleichung F.3 bestimmt.

Beim Gear-PC wird nur einmal korrigiert. Dies hat für die Simulation dann den Vorteil, daß auch die Kraftberechnung nur einmal durchgeführt wird.

Jetzt ist zu beachten, daß der Korrekturschritt abhängig von der Ordnung der zu lösenden Differentialgleichung ist.

Es wird  $\Delta \mathbf{r}$  berechnet, das den Unterschied zwischen dem aus Gleichung F.11 vorhergesagten  $\mathbf{r}_n^p$  und dem Wert, der durch Anwendung der DGL auf die vorhergesagten Werte berechnet wurde, angibt.

Für eine Differentialgleichung der Form

$$\begin{array}{l} \dot{\mathbf{r}} = f(\mathbf{r}) \\ \ddot{\mathbf{r}} = f(\mathbf{r}) \\ \ddot{\mathbf{r}} = f(\mathbf{r}, \dot{\mathbf{r}}) \end{array} \quad \text{ist} \quad \begin{array}{l} \Delta \mathbf{r} = \mathbf{r}_1^c - \mathbf{r}_1^p \\ \Delta \mathbf{r} = \mathbf{r}_2^c - \mathbf{r}_2^p \\ \Delta \mathbf{r} = \mathbf{r}_2^c - \mathbf{r}_2^p \end{array} \quad \text{mit} \quad \begin{array}{l} \mathbf{r}_1^c = f(\mathbf{r}_0^p) \\ \mathbf{r}_2^c = f(\mathbf{r}_0^p) \\ \mathbf{r}_2^c = f(\mathbf{r}_0^p, \mathbf{r}_1^p). \end{array}$$

p	$c_0$	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$
1	$\frac{95}{288}$	1	$\frac{25}{24}$	$\frac{35}{72}$	$\frac{5}{48}$	$\frac{1}{120}$
2	$\frac{3}{20}$	$\frac{251}{360}$	1	$\frac{11}{18}$	$\frac{1}{6}$	$\frac{1}{60}$
3	$\frac{3}{80}$	$\frac{19}{40}$	$\frac{9}{8}$	1	$\frac{3}{8}$	$\frac{1}{20}$
4	$\frac{1}{30}$	$\frac{1}{10}$	1	$\frac{5}{3}$	1	$\frac{1}{5}$

**Tabelle F.1:** In dieser Tabelle sind die Korrekturfaktoren für den Gear-Predictor-Corrector 6der Ordnung 6 dargestellt. p gibt die Ordnung der zu lösenden Differentialgleichung an. Weitere Koeffizienten für andere Ordnungen finden sich in [116, 118].

Dann ist der Korrekturschritt durch

$$\begin{pmatrix} \mathbf{r}_0^c(t + \Delta t) \\ \mathbf{r}_1^c(t + \Delta t) \\ \mathbf{r}_2^c(t + \Delta t) \\ \mathbf{r}_3^c(t + \Delta t) \\ \mathbf{r}_4^c(t + \Delta t) \\ \mathbf{r}_5^c(t + \Delta t) \end{pmatrix} = \begin{pmatrix} \mathbf{r}_0^p(t + \Delta t) \\ \mathbf{r}_1^p(t + \Delta t) \\ \mathbf{r}_2^p(t + \Delta t) \\ \mathbf{r}_3^p(t + \Delta t) \\ \mathbf{r}_4^p(t + \Delta t) \\ \mathbf{r}_5^p(t + \Delta t) \end{pmatrix} + \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix} \Delta \mathbf{r} \quad (\text{F.13})$$

gegeben.

Die Parameter  $c$  sind aus Tabelle F.1 ersichtlich. Im Fall der DGL der Form  $\ddot{\mathbf{r}} = f(\mathbf{r}, \dot{\mathbf{r}})$  wird in [104] vorgeschlagen, beim 5-Value-PC für  $c_0$  den Wert  $\frac{19}{120}$  durch  $\frac{19}{90}$ , und beim 6-Value-PC  $\frac{3}{20}$  durch  $\frac{3}{16}$  zu ersetzen.



---

## Literaturverzeichnis

- [1] F. Spahn, J. M. Hertzsch, and N. V. Brilliantov. The role of particle collisions for the dynamics in planetary rings. *Chaos, Solitons & Fractals*, 5:1945, 1995.
- [2] S. Dippel, G. G. Batrouni, and D. E. Wolf. Collision-induced friction in the motion of a single particle on a bumpy inclined line. *Phys.Rev. E*, 54:6845, 1996.
- [3] S. Dippel. *Microscopic Dynamics of Granular Materials*. PhD thesis, Forschungszentrum Jülich – Zentralbibliothek, 1998.
- [4] A. Betat and I. Rehberg. Dry friction of a solid body in a rotating drum. In *Friction, Arching, Contact Dynamics*, page 301, Singapore, 1997. World Scientific.
- [5] A. Schinner and K. Kassner. Dynamics of a sliding particle in a rotating drum. In *Friction, Arching, Contact Dynamics*, pages 305–309, Singapore, 1997. World Scientific.
- [6] M. A. Scherer, V. Buchholtz, T. Pöschel, and I. Rehberg. Swirling granular matter: From rotation to reptation. *Phys. Rev. E*, 54:R4560, 1996.
- [7] Michael A. Scherer, Thomas Mahr, Andreas Engel, and Ingo Rehberg. Granular dynamics in a swirled annulus. *Physical Review E*, 58(5):6061–6072, 1998.
- [8] Michael A. Scherer. *Der Schwenkeffekt - Experimentelle Untersuchungen zur granularen Dynamik*. PhD thesis, Otto-von-Guericke-Universität-Magdeburg, 1998.
- [9] S. Schwarzer. Sedimentation and flow through porous media: Simulating dynamically coupled discrete and continuum phases. *Phys. Rev. E*, 52(6):6461, 1995.
- [10] W. Kalthoff, S. Schwarzer, G. H. Ristow, and H. J. Herrmann. On hydrodynamic diffusion and velocity fluctuations in two-dimensional simulations of sedimentation. *Int. J. Mod. Phys. C*, 7:543–561, 1996.
- [11] Stefan Schwarzer. Particles in liquids. In H. J. Herrmann, J.-P. Hovi, and S. Luding, editors, *Physics of Dry Granular Media*, volume 350 of *Series E: Applied Sciences*. NATO ASI, Kluwer Academic Publishers, September 1997.
- [12] C. Völtz, M. Schröter, G. Iori, A. Betat, A. Lange, A. Engel, and I. Rehberg. Finger-like patterns in sedimenting water-sand suspensions. accepted for publication in *Physics Reports*.
- [13] X. I. Wu and K. J. Maløy. Why hour glasses tick. *Physical Review Letters*, 71(9):1363–1366, August 1993.
- [14] Alexander Schinner and Hans-Georg Mattutis. Molecular dynamics of cohesive granular materials. In D. Helbing, H.J. Herrmann, M. Schreckenberg, and D. E. Wolf, editors, *Traffic and Granular Flow*, pages 505–510. Springer, 1999.

- [15] R. Gutfraind and S. B. Savage. Smoothed particle hydrodynamics for the simulation of broken-ice fields: Mohr-coulomb type rheology and frictional boundary conditions. unpublished, 1997.
- [16] R. Gutfraind and S. B. Savage. Marginal ice zone rheology: Comparison of results from continuum-plastic models and discrete-particle simulations. unpublished, 1997.
- [17] R. A. Bagnold. Experiments on a gravity-free dispersion of large solid spheres in a Newtonian fluid under shear. *Proc. Royal Soc. London*, 225:49–63, 1954.
- [18] Tim Niclas Scheffler. *Kollisionskühlung in elektrisch geladener granularer Materie*. PhD thesis, Universität Duisburg, 2000.
- [19] L. Bocquet, E. Charlaix, S. Ciliberto, and J. Crassous. Moisture induced ageing in granular media. unpublished, 1998.
- [20] J. Dieterich and G. Conrad. Effect of humidity on time- and velocity-dependent friction in rocks. *J. Geophysical Research*, 89:4196–4202, 1984.
- [21] R. Albert, I. Albert, D. Hornbaker, P. Schiffer, and A.-L. Barabási. Maximum angle of stability in wet and dry spherical granular media. *Phys. Rev. E*, 56(6):R6271–R6274, 1997.
- [22] Heinrich M. Jaeger. Chicago experiments on convection, compaction, and compression. In H. J. Herrmann, J.-P. Hovi, and S. Luding, editors, *Physics of Dry Granular Media*, volume 350 of *Series E: Applied Sciences*. NATO ASI, Kluwer Academic Publishers, September 1997.
- [23] D. E. Wolf and P. Grassberger, editors. *Friction, Arching and Contact Dynamics*. World Scientific, 1996.
- [24] G. G. Batrouni, S. Dippel, and L. Samson. Stochastic model for the motion of a particle on an inclined rough plane and the onset of viscous friction. *Phys. Rev. E*, 53:6496, 1996.
- [25] S. Dippel, G. G. Batrouni, and D. E. Wolf. Friction experienced by a particle moving on a rough inclined plane. In *Friction, Arching, Contact Dynamics*, page 311, 1997.
- [26] S. Dippel, L. Samson, and G. G. Batrouni. Motion of a ball on a rough inclined plane. unpublished, 1996.
- [27] F. J. Elmer. Macroscopic stick-slip motion in the Frenkel-Kontorova-Tomlinson model. In D. E. Wolf and P. Grassberger, editors, *Friction, Arching and Contact Dynamics*, pages 119–123. World Scientific, 1997.
- [28] Franz-Josef Elmer. Controlling friction. *Physical Review E*, 57(5):4903–4906, 1998.
- [29] A. Schinner, M. Scherer, I. Rehberg, and K. Kassner. Rotation and reptation. In H. J. Herrmann, J.-P. Hovi, and S. Luding, editors, *Physics of Dry Granular Media*, volume 350 of *Series E: Applied Sciences*, pages 639–644. NATO ASI, Kluwer Academic Publishers, September 1997.
- [30] M. Coulomb. Theorie des Machines Simples. *Academie des Sciences*, 10:166, 1781.
- [31] M. Faraday. On a peculiar class of acoustical figures and on certain forms assumed by groups of particles upon vibrating elastic surfaces. *Philos. Trans. R. Soc. London*, 52:299, 1831.
- [32] O. Reynolds. On the dilatancy of media composed of rigid particles in contact. *Philosophical Magazine and Journal of Science, fifth Series*, 50-20:469, 1885.

- [33] D. E. Wolf, M. Schreckenberg, and A. Bachem, editors. *Traffic and Granular Flow*. World Scientific, 1995.
- [34] D. Helbing, H.J. Herrmann, M. Schreckenberg, and D. E. Wolf, editors. *Social, Traffic, and Granular Dynamics*. Springer, 1999.
- [35] P. Bak, C. Tang, and K. Wiesenfeld. Self-organized criticality: An explanation of  $1/f$  noise. *Phys. Rev. Lett.*, 59(4):381–384, 1987.
- [36] P. Bak, C. Tang, and K. Wiesenfeld. Self-organized criticality. *Phys. Rev. A*, 38(1):364–375, 1988.
- [37] V. Frette, K. Christensen, A. Malthe-Sorensen, J. Feder, T. Jossang, and P. Meakin. Avalanche dynamics in a pile of rice. *Nature*, 379:49–51, 1996.
- [38] H. J. Herrmann. On the shape of a sandpile. In H. J. Herrmann, J.-P. Hovi, and S. Luding, editors, *Physics of Dry Granular Media*, volume 350 of *Series E: Applied Sciences*. NATO ASI, Kluwer Academic Publishers, September 1997.
- [39] M. Rimmele. *Modellierung von Böschungswinkeln und Stratifikation von Schüttgütern*. PhD thesis, Stuttgart, 1997.
- [40] J. J. Alonso J.-P. Hovi and H. J. Herrmann. Lattice model for the calculation of the angle of repose from microscopic grain properties. *Physical Review E*, 55(1):672–680, 1998.
- [41] G. A. Held, D. H. Solina, H. Solina, D. T. Keane, W. J. Haag, P. M. Horn, and G. Grinstein. Experimental study of critical-mass fluctuations in an evolving sandpile. *Physical Review Letters*, 65:1120–1123, August 1990.
- [42] Grasselli Y. and Herrmann H.J. On the angles of dry granular heaps. *Physica A*, 246:301, 1997.
- [43] Christian M. Dury, Gerald H. Ristow, Jamie L. Moss, and Masami Nakagawa. Boundary effects on the angle of repose in rotating cylinder. *Physical Review E*, 57(4):4491–4497, April 1998.
- [44] J.M.N.T. Gray and Y.C. Tai. Particle size segregation, granular shocks and stratification patterns. In H. J. Herrmann, J.-P. Hovi, and S. Luding, editors, *Physics of Dry Granular Media*, volume 350 of *Series E: Applied Sciences*. NATO ASI, Kluwer Academic Publishers, September 1997.
- [45] G. C. Barker and A. Mehta. Rotated sandpiles: The role of grain reorganization and inertia. *Phys. Rev. E*, 53(6):5704–5713, 1996.
- [46] G. Baumann, I.M. Janosi, and D. E. Wolf. Surface properties and flow of granular material in a 2D rotating drum model. *Phys.Rev.E*, 51:1879, 1995.
- [47] G. Baumann, T. Scheffler, I.M. János, and D. E. Wolf. Angle of repose in a two-dimensional rotating drum model. In D. E. Wolf, M. Schreckenberg, and A. Bachem, editors, *Traffic and Granular Flow*, page 347, Singapore, 1996. World Scientific.
- [48] V. Buchholtz, T. Pöschel, and H.-J. Tillemans. Simulation of rotating drum experiments using non-circular particles. *Physica A*, 216:199, 1995.
- [49] D. J. Hornbaker, R. Albert, I. Albert, A.-L. Barabasi, and P. Schiffer. What keeps sandcastles standing? *Nature*, 387:765, 1997.

- [50] Stéphane Douady and Adrian Daerr. Formation of sandpiles, avalanches on an inclined plane. In H. J. Herrmann, J.-P. Hovi, and S. Luding, editors, *Physics of Dry Granular Media*, volume 350 of *Series E: Applied Sciences*. NATO ASI, Kluwer Academic Publishers, September 1997.
- [51] F.-X. Riguidel, R. Jullien, G. H. Ristow, A. Hansen, and D. Bideau. Behaviour of a sphere on a rough inclined plane. *J. Phys. I*, 4:261, 1994.
- [52] F.-X. Riguidel, A. Hansen, and D. Bideau. Gravity-driven motion of a particle on an inclined plane with controlled roughness. *Europhys. Lett.*, 28:13, 1994.
- [53] L. Quartier, B. Andreotti, S. Douady, and A. Daerr. On the dynamics of grains on a sandpile. in preparation, 2000.
- [54] S. S. Manna and D. V. Khakhar. Internal avalanches in granular medium. *Physical Review E*, 58(6):6935–6938, December 1998.
- [55] M. A. Aguirre, N. Nerone, A. Calvo, I. Ippolito, and D. Bideau. Avalanche parameters: Dependence on the size of the granular packing. In D. Helbing, H.J. Herrmann, M. Schreckenberg, and D. E. Wolf, editors, *Traffic and Granular Flow*, pages 489–494. Springer, 1999.
- [56] M. A. Aguirre, N. Nerone, A. Calvo, I. Ippolito, and D. Bideau. Influence of the number of layers on the equilibrium of a granular packing. *Physical Review E*, 62(1):489–494, 2000.
- [57] A. Daerr and S. Douady. Sensitivity of granular surface flows to preparation. *Euro. Phys. Lett.*, 47(3):324–330, 1999.
- [58] A. Daerr and S. Douady. Two types of avalanche behaviour in granular media. *Letters to Nature*, 399:241–243, 1999.
- [59] Adrian Daerr. Dynamical equilibrium of avalanches on a rough plane. in preparation, 2000.
- [60] G. C. Barker and Anita Mehta. Avalanches at rough surfaces. *Physical Review E*, 61(6):6765–6772, June 2000.
- [61] P. Biswas, A. Majumdar, A. Mehta, and J. K. Bhattacharjee. Smoothing of sandpile surfaces after intermittent and continuous avalanches: three models in search of an experiment. *Physical Review E*, 58(2):1266–1285, 1998.
- [62] H. Nishimori and N. Ouchi. Formation of ripple patterns and dunes by wind-blown sand. *Phys. Rev. Lett.*, 71(1):197, 1993.
- [63] H. Nishimori and M. Yamasaki. A minimal model approach for the morphodynamics of dunes. In H. J. Herrmann, J.-P. Hovi, and S. Luding, editors, *Physics of Dry Granular Media*, volume 350 of *Series E: Applied Sciences*. NATO ASI, Kluwer Academic Publishers, September 1997.
- [64] J. J. Alonso and H. J. Herrmann. Shape of a two-dimensional sandpile. *Physical Review Letters*, 76(26):4911–4914, June 1996.
- [65] James B. Knight, E. E. Ehrichs, Vadim Yu. Kuperman, Janna K. Flint, Heinrich M. Jaeger, and Sidney R. Nagel. Experimental study of granular convection. *Physical Review E*, 54:5726–5738, November 1996.
- [66] Keiko M. Aoki, Tetsuo Akiyama, Yoji Maki, and Tatsuyuki Watanabe. Convective roll patterns in vertically vibrated beds of granules. *Physical Review E*, 54:874–883, July 1996. Konvektionsrollen in vibriertem Sand.

- [67] J. Duran. Depletion and multiparticle segregation. In H. J. Herrmann, J.-P. Hovi, and S. Luding, editors, *Physics of Dry Granular Media*, volume 350 of *Series E: Applied Sciences*. NATO ASI, Kluwer Academic Publishers, September 1997.
- [68] G. H. Ristow. Dynamics of granular materials in a rotating drum. *Europhysics Letters*, 34(4):263–268, 1996.
- [69] Gerald H. Ristow and Masami Nakagawa. Shape dynamics of interfacial front in rotating cylinders. *Physical Review E*, 59(2):2044–2048, February 1999.
- [70] A. Karolyi, J. Kertesz, H. Makse, H. E. Stanley, and S. Halvon. Cellular automata models for granular media. In H. J. Herrmann, J.-P. Hovi, and S. Luding, editors, *Physics of Dry Granular Media*, volume 350 of *Series E: Applied Sciences*. NATO ASI, Kluwer Academic Publishers, September 1997.
- [71] J. P. Koeppe, M.ENZ, and J. Kakalios. Phase diagram for avalanche stratification of granular media. *Physical Review E*, 58(4):4104–4107, 1998.
- [72] J. Baxter, U. Tüzün, D. Heyes, I. Hayati, and P. Fredlund. Stratification in poured granular heaps. *Nature*, 391:136, 1998.
- [73] Y. Grasselli and H. J. Herrmann. Experimental study of granular stratification. *Granular Matter*, 1(1), 1998. in press.
- [74] D. A. Head and G. J. Rodgers. Slowly driven sandpile formation with granular mixtures. *Phys. Rev. E*, 56:1976, 1997.
- [75] H. A. Makse, S. Havlin, P. R. King, and H. E. Stanley. Spontaneous stratification in granular mixtures. *Nature*, 386:379–381, 1997.
- [76] H. A. Makse, P. Cizeau, and H. E. Stanley. Self-stratification mechanism in granular mixtures. *Phys. Rev. Lett.*, 78:3298, 1997.
- [77] F. H. Hummel and E. J. Finnan. The distribution of pressure on surfaces supporting a mass of granular material. *Proc. Instn. Civil Engn.*, 212:369–392, 1921. the former: Minutes of Proc. of the Instn. of civil Engineers with other selected papers.
- [78] Tomosada Jotaki and Ryuichi Moriyama. On the bottom pressure distribution of the bulk materials piled with the angle of repose. *J. of the Soc. of Powder Technol., Japan*, 16(4):184–191, 1979. in Japanese.
- [79] R. Brockbank, J. M. Huntley, and R.C. Ball. Contact force distribution beneath a three-dimensional granular pile. *J. Phys. II France*, 7:1521–1532, 1997.
- [80] J. Šmid and J. Novosad. Pressure distribution under heaped bulk solids. *I. Chem. E. Symposium Series*, 63:D3/V/1–12, 1981.
- [81] J. Šmid. Stress and friction forces in the base of a heap of bulk solids. In *Proceedings of International Symposium on Powder Technology 1981*, pages 177–183. , 1981.
- [82] I.K.Lee and J.R. Herrington. Stresses beneath granular embankments. *Proceedings of the first Australian-New Zealand Conference on Geomechanics: Melbourne*, 1:291–296, August 1971.
- [83] Bernhard Lackinger. *Das Tragverhalten von Staudämmen mit membranartigen Dichtungen*. PhD thesis, Mitteilungen des Instituts für Bodenmechanik, Felsmechanik und Grundbau an der Fakultät für Bauingenieurwesen und Architektur der Universität Innsbruck, 1980.

- [84] H.-G. Matuttis and A. Schinner. Influence of the geometry on the pressure distribution of granular heaps. *Granular Matter*, 1(4):195–201, 1999.
- [85] Loic Vanel, Daniel Howell, D. Clark, R. P. Behringer, and Eric Clement. Memories in sand: Experimental tests of construction history on stress distributions under sandpiles. *Physical Review E*, 60(5):R5040–R5043, November 1999.
- [86] D. C. Hong. Stress distribution of a hexagonally packed granular pile. *Phys. Rev. E*, 47(1):760–762, 1993.
- [87] H.-G. Matuttis and S. Luding. The effect of particle shape and friction on the stresses in heaps of granular media. In D. E. Wolf and P. Grassberger, editors, *Friction, Arching and Contact Dynamics*, pages 201–206, Singapore, 1997. World Scientific.
- [88] S. Luding and H.-G. Matuttis. The effect of interaction laws on the stresses in frictionless granular media. In D. E. Wolf and P. Grassberger, editors, *Friction, Arching and Contact Dynamics*, pages 207–211, Singapore, 1997. World Scientific.
- [89] J. P. Wittmer, M. E. Cates, and P. Claudin. Stress propagation and arching in static sandpiles. *J. Phys. I*, 7:39–80, 1997.
- [90] S. F. Edwards and R. B. S. Oakeshott. The transmission of stress in an aggregate. *Physica D*, 38:88–92, 1989.
- [91] J.-P. Bouchaud, P. Claudin, M. E. Cates, and J. P. Wittmer. Models of stress propagation in granular media. In H. J. Herrmann, J.-P. Hovi, and S. Luding, editors, *Physics of Dry Granular Media*, page 97, Dordrecht, 1998. Kluwer Academic Publishers.
- [92] J.-P. Bouchaud, M. E. Cates, and P. Claudin. Stress distribution in granular media and nonlinear wave equation. *J. Phys. I*, 5:639–656, 1995.
- [93] J. P. Wittmer, P. Claudin, M. E. Cates, and J.-P. Bouchaud. An explanation for the central stress minimum in sand piles. *Nature*, 382:336–338, 1996.
- [94] J. P. Wittmer, M. E. Cates, P. Claudin, and J.-P. Bouchaud. Arching effects in the stress distribution of sandpiles. In R. P. Behringer and J. T. Jenkins, editors, *Powders & Grains 97*, page 303. Balkema, Rotterdam, 1997.
- [95] Steve Morris. private communication, 2000.
- [96] D. Howell and R. P. Behringer. Fluctuations in a 2D granular Couette experiment: A critical transition. *Phys. Rev. Lett.*, 82:5241, 1999.
- [97] Brian Miller, Corey O’Hern, and R. P. Behringer. Stress fluctuations for continuously sheared granular materials. *Physical Review Letters*, 77:3110–3113, 1996.
- [98] O. Tsoungui, D. Vallet, and J.-C. Charmet. Experimental study of the force distributions inside 2D granular systems. In H. J. Herrmann, J.-P. Hovi, and S. Luding, editors, *Physics of Dry Granular Media*, volume 350 of *Series E: Applied Sciences*. NATO ASI, Kluwer Academic Publishers, September 1997.
- [99] R. A. Bagnold. *The Physics of Blown Sand and Desert Dunes*. Chapman and Hall, 1956.
- [100] M. D. Shattuck, C. Bizon, P. B. Umbanhowar, J. B. Swift, and H. L. Swinney. Pattern formation in vertically vibrated granular layers: Experiments and simulation. In H. J. Herrmann, J.-P. Hovi, and S. Luding, editors, *Physics of Dry Granular Media*, volume 350 of *Series E: Applied Sciences*, pages 613–618. NATO ASI, Kluwer Academic Publishers, September 1997.

- [101] H. J. Herrmann, J.-P. Hovi, and S. Luding, editors. *Physics of Dry Granular Media*. Kluwer Academic Publishers, 1997.
- [102] Jacques Duran. *Sands, Powders, and Grains*. Springer, 2000.
- [103] T. Pöschel and S. Luding, editors. *Granular Gases*. Springer Verlag, Berlin, 2000.
- [104] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. Oxford University Press, Oxford, 1987.
- [105] Alejandro L. Garcia. *Numerical Methods for Physics*. Prentice Hall, 1994.
- [106] Matthias Müller and Hans J. Herrmann. DSMC - a stochastic algorithm for granular matter. In H. J. Herrmann, J.-P. Hovi, and S. Luding, editors, *Physics of Dry Granular Media*, volume 350 of *Series E: Applied Sciences*. NATO ASI, Kluwer Academic Publishers, September 1997.
- [107] H. J. Herrmann and M. Müller. Molecular dynamics simulations of a charged granular medium. In Rüdiger Esser, Peter Grassberger, Johannes Grotendorts, and Marius Lewerenz, editors, *Molecular Dynamics Simulations of a Charged Granular Medium*, pages 239–249. World Scientific, 1999.
- [108] M. Müller and H. J. Herrmann. How much can we simplify a system of grains? In *Granular Gases*. Lecture Notes in Physics, Springer, 1999.
- [109] P. A. Cundall and O. D. L. Strack. A discrete numerical model for granular assemblies. *Géotechnique*, 29(1):47–65, 1979.
- [110] S. Luding. *Models and Simulations of Granular Materials*. PhD thesis, Universität Freiburg, 1994.
- [111] S. McNamara and W. R. Young. Inelastic collapse in two dimensions. *Phys. Rev. E*, 50(1):R28–R31, 1994.
- [112] S. McNamara and W. R. Young. Inelastic collapse and clumping in a one-dimensional granular medium. *Phys. Fluids A*, 4(3):496, 1992.
- [113] S. Luding and S. McNamara. How to handle the inelastic collapse of a dissipative hard-sphere gas with the TC model. *Granular Matter*, 1(3):113–128, 1998.
- [114] S. Luding and H. J. Herrmann. Cluster growth in freely cooling granular media. *Chaos*, 9(3):673–681, 1999.
- [115] C. Bizon, M. D. Shattuck, J. B. Swift, W. D. McCormick, and H. L. Swinney. Patterns in 3D vertically oscillated granular layers: Simulation and experiment. *Phys. Rev. Lett.*, 80(1):57–60, 1998.
- [116] C. William Gear. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, Inc., 1971.
- [117] Peter Henrici. *Discrete Variables Methods in Ordinary Differential Equations*. John Wiley & Sons, Inc., 1968.
- [118] Alexander Schinner. Numerische Simulationen für granulare Medien. Diplomarbeit, Sep 1995. Univ. Regensburg, Naturwissenschaftlichen Fakultät II, Physik.
- [119] Alexander V. Potapov and Charles S. Campbell. A fast model for the simulation of non-round particles. *Granular Matter*, 1(1):9–14, 1998.

- [120] V. Buchholtz and T. Pöschel. Numerical investigations of the evolution of sandpiles. *Physica A*, 202:390, 1994.
- [121] Hans-Jürgen Tillemans. *Molekulardynamik-Simulationen beliebig geformter Teilchen in zwei Dimensionen*. PhD thesis, Forschungszentrum Jülich, 1995.
- [122] L. D. Landau and E. M. Lifschitz. *Elastizitätstheorie*. Akademie Verlag Dresden, Berlin, 1989.
- [123] Torsten Gröger. *Partikelmechanische Untersuchungen zur senkrechten Schlauchgutförderung*. PhD thesis, Otto-von-Guericke-Universität-Magdeburg, 1999.
- [124] Hans-Georg Matuttis. private communication.
- [125] Ernest Rabinowicz. *Friction and Wear of Materials*. John Wiley, New York, London, Sidney, 1965.
- [126] S. Luding. Collisions & contacts between two particles. In H. J. Herrmann, J.-P. Hovi, and S. Luding, editors, *Physics of dry granular media - NATO ASI Series E350*, page 285, Dordrecht, 1998. Kluwer Academic Publishers.
- [127] Ferenc Kun and Hans J. Herrmann. Transition from damage to fragmentation in collision of solids. *Phys. Rev. E*, 59(3):2623–2632, 1999.
- [128] Jean-Christophe Geminard, Wolfgang Losert, and Jerry P. Gollub. Frictional mechanics of wet granular material. *Physical Review E*, 59(5):5881–5890, 1999.
- [129] W. Losert, J.-C. Geminard, S. Nasuno, and J. P. Gollub. Mechanisms for slow strengthening in granular materials. *Physical Review E*, 61:4060, 2000.
- [130] Holger Kruse, Karl Popp, Hans-Georg Matuttis, and Alexander Schinner. Behandlung des Schotters als Vielkörpersystem mit wechselnden Bindungen. *EI - Der Eisenbahningenieur*, 2000. submitted.
- [131] H.-G. Matuttis. *Determinanten Quanten Monte Carlo und analytische Fortsetzung für das Hubbard Modell*. PhD thesis, Universität Regensburg, 1995.
- [132] D. J. Hornbaker. Private Kommunikation.
- [133] Alexander Schinner. Fast algorithms for the simulation of polygonal particles. *Granular Matter*, 2,1:35–43, 1999.
- [134] D. Baraff. Rigid body simulation. In *Course 60, An introduction to Physically Based Modeling*, ACM Siggraph, pages H1–H68, 1993.
- [135] Sedgewick. *Algorithmen*. Addison-Wesley, 1992.
- [136] Alexander Schinner and Klaus Kassner. Shared memory parallelization for molecular-dynamics simulations of non-spherical granular material. In Rüdiger Esser, Peter Grassberger, Johannes Grotendorts, and Marius Lewerenz, editors, *Molecular Dynamics on Parallel Computers*, pages 239–249. World Scientific, 1999.
- [137] M. C. Lin. *Efficient Collision Detection for Animation and Robotics*. PhD thesis, University of California at Berkeley, 1993.
- [138] M.C. Lin and J.F. Canny. A fast algorithm for incremental distance calculation. In *International Conference on Robotics and Automation*, pages 1008–1014. IEEE, 1991.

- [139] M.C. Lin. *Efficient Collision Detection for Animation and Robotics*. PhD thesis, University of California at Berkeley, December 1993.
- [140] M.C. Lin and D. Manocha. Interference detection between curved objects for computer animation. *Models and Techniques in Computer Animation*, pages 43–57, 1993.
- [141] S. Kleinman, D. Shah, and B. Smaalders. *Programming with Threads*. SunSoft Press A Prentice Hall Title, 1996.
- [142] Tim Scheffler and D.E. Wolf. Shared memory parallelization for molecular-dynamics simulations of non-spherical granular material. In Rüdiger Esser, Peter Grassberger, Johannes Grotendorts, and Marius Lewerenz, editors, *Molecular Dynamics Simulations of a Charged Granular Medium*, pages 239–249. World Scientific, 1999.
- [143] Steven Brawer. *Introduction to Parallel Programming*. Academic Press, Inc, 1989.
- [144] Almasi and Gottlieb. *Highly parallel Computing*. The Benjamin/Cummings Publishing Company, Inc, 1994.
- [145] I. T. Foster. *Designing and Building Parallel Programs*. Addison-Wesley, Reading, Massachusetts, 1995.
- [146] Lou Baker and Bradley J. Smith. *Parallel Programming*. McGraw-Hill, 1996.
- [147] T. Ungerer. *Parallelrechner und Parallele Programme*. Spektrum Akademischer Verlag, Heidelberg, 1997.
- [148] David Spector. *Building Linux Clusters*. O Reilly, 2000.
- [149] Donald E. Knuth. *The Art of Computer Programming*, volume 1-3. Addison-Wesley, 1968.
- [150] Gerthsen, Kneser, and Vogel. *Physik*. Springer, 16 edition, 1989.
- [151] B. Painter, S. Tennakoon, and R. P. Behringer. Collisions and fluctuations for granular materials. In H. J. Herrmann, J.-P. Hovi, and S. Luding, editors, *Physics of Dry Granular Media*, volume 350 of *Series E: Applied Sciences*. NATO ASI, Kluwer Academic Publishers, September 1997.
- [152] Walter Greiner. *Mechanik 2*. Verlag Harri Deutsch, 1989.
- [153] Heiko Bauke. `gsi.pm` - Ein Perl-Modul für `gc2d`. unpublished, July 2000.
- [154] J. Eggers. Sand as Maxwell's demon. *Physical Review Letters*, 83:5322, 1999.
- [155] H. A. Janssen. Versuche über Getreidedruck in Silozellen. *Zeitschr. d. Vereines deutscher Ingenieure*, 39(35):1045–1049, 1895.
- [156] G. Oron and H. J. Herrmann. Exact calculation of force networks in granular media. *Phys. Rev. E*, 58(2):2079–2089, 1998.
- [157] G. Oron and H. J. Herrmann. Exact determination of force networks in a static assembly of disks. In H. J. Herrmann, J.-P. Hovi, and S. Luding, editors, *Physics of Dry Granular Media*, page 143, Dordrecht, 1998. Kluwer Academic Publishers.
- [158] Alexander Schinner, Hans-Georg Matuttis, Tetso Akiyama, Junya Aoki, Satoshi Takahashi, Keiko M. Aoki, and Klaus Kassner. History-dependent structure in granular piles. eingereicht für POWDERS & GRAINS 2001.

- [159] K. Aoki. private communication, 2000.
- [160] R. Albert, M. A. Pfeifer, and A.-L. Barabási. Drag force in a granular medium. unpublished, 1998.
- [161] J.H Westbrook and P.J Jorgensen. Effect of water desorption on indentation microhardness anisotropy in minerals. *The American Mineralogist*, 53:1899–1914, 1968.
- [162] R. L. Brown and J. C. Richards. *Principles of Powder Mechanics*. Pergamon Press, Oxford, 1970.
- [163] R. L. Carr. . *Journ. Chem. Engin.*, 18:163, 1965.
- [164] T. Akiyama and Y. Tanijiri. Criterion for re-entrainment of particles. *Powder Technology*, 57:21–26, 1989.
- [165] H. Stöcker. *Taschenbuch mathematischer Formeln und moderner Verfahren*. Verlag Harri Deutsch, 1995.
- [166] Larry Wall, Tom Christiansen, and Randal L. Schwartz. *Programming Perl*. OReilly, 1996.

# Lebenslauf

Name: Schinner  
Vorname: Alexander  
Geburtsdatum: 3.12.1968  
Geburtsort: Wasserburg am Inn  
Staatsangehörigkeit: Deutsch  
Wohnsitz: Valpichlerstr. 121, 80689 München  
1975-1979 Grundschule an der Camerloherstraße, München  
1979-1988 Max-Planck-Gymnasium, München  
11/1988-10/1989: Hochschulstudium Technische Universität München  
Studienfach "Elektrotechnik (Diplom)"  
11/1989-10/1995: Hochschulstudium Universität Regensburg  
Studienfach "Physik (Diplom)"  
Abschluß als Dipl.-Phys.  
4/1996-4/2001 wissenschaftlicher Mitarbeiter am Institut für Theoretische Physik  
an der Otto-von-Guericke-Universität Magdeburg  
seit 4/2001 Institut für Theoretische Physik  
Johann Wolfgang Goethe-Universität  
Frankfurt am Main