

Simulation von hoch polydispersen zufällig dichten Partikelpackungen unter Berücksichtigung der Agglomeration im Feinstkornbereich

Dissertation
zur Erlangung des Doktorgrades
der Naturwissenschaften

vorgelegt von
Stephan Mock
aus Heilbad Heiligenstadt

genehmigt von der Fakultät für
Mathematik/Informatik und Maschinenbau
der Technischen Universität Clausthal

Tag der mündlichen Prüfung

.....

Vorsitzende/Vorsitzender der Promotionskommission:

.....

Hauptberichterstatter:

Prof. Dr. Michael Kolonko, Technische Universität Clausthal

Mitberichterstatter:

Prof. Dr. Gabriel Zachmann, Universität Bremen

Inhaltsverzeichnis

Danksagungen	3
1 Einleitung	5
1.1 Aufbau der Arbeit	7
2 Grundlagen	9
2.1 Der Baustoff Beton	9
2.2 Wichtige Kenngrößen	11
2.2.1 Die Raumausfüllung	11
2.2.2 Die Korngrößenverteilung	13
2.2.3 Die fraktale Dimension	16
2.3 Reguläre Kugelpackungen	18
2.3.1 Monodisperse reguläre Kugelpackungen	19
2.3.2 Apollonische Kugelpackungen	21
2.4 Zufällig dichte Packungen	22
2.4.1 Der Wandeffekt	23
2.5 Spezielle Korngrößenverteilungen	29
2.6 Methoden zur Bestimmung der Raumausfüllung	31
2.6.1 Experimentelle Bestimmung der Raumausfüllung	31
2.6.2 Mathematisch empirische und semiempirische Bestimmung der Raumausfüllung	32
2.6.3 Bestimmung der Raumausfüllung per Simulation	36
3 Die Packungssimulation „RaSim“	39
3.1 Die Fraktionierung	40
3.1.1 Arten des Sampling	42
3.1.2 Hierarchische Aufteilung der Samples	43
3.1.3 Werte und Parameter für die Fraktionierung	47
3.2 Die Simulation	50
3.2.1 Die Diskrete Elemente Methode	50
3.2.2 Collective Rearrangement	51
3.3 Die Nachbarschaftsverwaltung	52
4 Simulation besonderer Korngrößenverteilungen	57
4.1 Dominante große Partikel	58
4.1.1 Simulationen mit kritischen Verteilungen	60
4.1.2 Auswirkungen der Samplinggröße auf die Raumausfüllung	65
4.1.3 Geeignete Gegenmaßnahmen	67

4.2	Online-Fraktionierung	68
4.2.1	Implementierung	70
4.3	Auswertung	74
5	Agglomeration	77
5.1	Ursache und Wirkung	79
5.2	Gegenmaßnahmen	85
5.3	Berücksichtigung in der Simulation	86
5.4	Simulative Bestimmung der agglomerierten Korngrößenverteilung . . .	88
5.4.1	Ausgangsdaten für die Simulation	91
5.4.2	Bestimmung der agglomerierenden Primärpartikel	92
5.4.3	Bestimmung der Anteile der Agglomerate	93
5.4.4	Endergebnis: die stabilen Anteile	95
5.4.5	Festlegung wichtiger Größen für die Simulation	96
5.5	Details der Implementierung	100
5.5.1	Berücksichtigung der Streuung der Agglomerategrößen	104
5.6	Simulative Bestimmung der Packungsdichte von agglomerierten Partikel- mischungen	108
5.6.1	Kombination von PSD-Morph und RaSim	109
5.6.2	Simulationsergebnisse	111
6	Simulation des Wachstums natürlicher Muster	115
6.1	Der DLA Algorithmus	115
6.1.1	Varianten des DLA-Algorithmus	118
6.1.2	Fazit	120
6.2	Weitere Modelle	121
6.2.1	Der Eden-Growth Algorithmus	121
6.2.2	Perkolation	122
6.3	Implementierungen	124
6.3.1	Ein Off-Lattice DLA-Algorithmus	124
6.3.2	Erweiterungen des Algorithmus	128
6.3.3	Clusterkugelverwaltung	129
6.4	Laufzeitverhalten des DLA-Algorithmus	133
6.5	Eigenschaften der DLA-Agglomerate	135
7	Zukünftige Entwicklungen	143
7.1	Parallele Simulation	146
7.1.1	Parallele Simulation der Fraktionen	146
7.1.2	Parallele Berechnung der Entlappung	147
7.1.3	Parallele Simulation auf der GPU	148
A	Eingesetzte Software	151
A.1	RaSim	151
A.1.1	RaSim mit grafischer Oberfläche	157
A.2	PSD-Morph	159
A.3	Ein- und Ausgabedateien	162
A.4	Sonstige Software	165

Danksagungen

An dieser Stelle möchte ich mich bei allen bedanken, die zum Gelingen dieser Arbeit beigetragen haben. Insbesondere gilt mein Dank Herrn Prof. Dr. Michael Kolonko für die Betreuung meiner Arbeit, für viele Anregungen und Ideen und Rat bei schwierigen Problemen. Ebenso danke ich den Mitarbeitern vom Institut für Nichtmetallische Werkstoffe: Herrn Prof. Dr. Albrecht Wolter, dessen Fachwissen unverzichtbar war für das Verständnis der Thematik, und Dipl. Chem. Thomas Bohne für die gute Zusammenarbeit und die Durchführung der labortechnischen Messungen, deren Ergebnisse in dieser Arbeit dargestellt sind. Besonders bedanken möchte ich mich auch bei Dr. Steffen Raschdorf für die Beantwortung vieler Fragen, die bei der Einarbeitung in die Simulationssoftware auftraten.

Diese Arbeit wurde durch ein Stipendium der Dyckerhoff-Stiftung (Projektnummer T218/20381/2010) ermöglicht. Mein herzlicher Dank gilt deshalb der Stiftung und Herrn Schaaf vom Stifterverband für die Deutsche Wissenschaft für die freundliche und unbürokratische Hilfe.

Kapitel 1

Einleitung

Beton ist heute einer der am meisten eingesetzten Baustoffe. Er kommt bei Straßen, Brücken, Tunneln und fast allen Arten von Gebäuden zum Einsatz. Hoch entwickelte Rezepturen und eine Vielzahl zur Verfügung stehender Zusätze erlauben eine Anpassung an die unterschiedlichsten Anforderungen. Die vorliegende Arbeit entstand aus der Problemstellung der Verbesserung dieses Hochleistungsbaustoffes. Wir betrachten dabei nicht die chemischen Reaktionen, die zur Bildung des festen Betonsteins führen. Statt dessen konzentrieren wir uns auf die Mischung der trockenen granularen Ausgangsstoffe. Ihre Komposition hat entscheidenden Einfluss auf die Eigenschaften des fertigen Baustoffes. Von besonderem Interesse ist die Raumauffüllung (auch Packungsdichte genannt) der Partikelpackung, sie beeinflusst die Festigkeit und Dauerhaftigkeit des fertig ausgehärteten Betons. Bei der experimentellen Bestimmung einer optimalen Mischung sind eine Vielzahl von Messungen verschiedener Kenngrößen im Labor notwendig. Um diese aufwendigen Untersuchungen teilweise ersetzen bzw. ergänzen zu können, wurden die Simulationssysteme „RaSim“ und „PSD-Morph“ entwickelt. Das von Raschdorf in seiner Arbeit [Ras10] entwickelte „RaSim“ ermöglicht es, ein 3-dimensionales Computermodell der Partikelpackung einer gegebenen Mischung zu erstellen und aus diesem die Raumauffüllung zu ermitteln. Die Fortsetzung der Arbeit Raschdorfs und die Weiterentwicklung von „RaSim“, sind Gegenstand der vorliegenden Arbeit. Von besonderer Bedeutung ist dabei die Entwicklung von „PSD-Morph“, mit dem sich die Wirkung der Agglomeration der kleinsten Partikel in die Packungssimulation einbeziehen lässt.

Der Einsatz einer Partikelsimulation zur Bestimmung von Kenngrößen einer granularen Mischung ist heute dank leistungsfähiger Computertechnik und effizienter Algorithmen möglich und sinnvoll. Vor allem aber bietet sie den Vorteil, dass nur wenige Ausgangsdaten benötigt werden und somit die experimentelle Arbeit im Labor deutlich reduziert werden kann. Die technische Umsetzung einer entsprechenden Si-

mulation stellt jedoch eine große Herausforderung dar. Ein Grund dafür liegt in der besonderen Korngrößenverteilung wie sie in Baustoffen wie Betonmischungen vorkommt. Sie besteht aus einer sehr großen Bandbreite unterschiedlicher Korngrößen: von den weniger als $1\mu\text{m}$ großen Partikeln der feinen Zusatzstoffe bis zu den mehr als 1cm großen Partikeln der groben Gesteinskörnung (Kies). Eine virtuelle Partikelpackung, die alle in der realen granularen Mischung vorkommenden Korngrößen im richtigen Verhältnis abbildet, benötigt eine so hohe Anzahl Partikel, dass die Simulation die Möglichkeiten heutiger Standardhardware bei weitem übersteigt. Das System „RaSim“ löst dieses Problem durch eine hierarchische Herangehensweise - der sogenannten Fraktionierung der Korngrößenverteilung - und den Einsatz moderner Datenstrukturen und ist so in der Lage, Partikelpackungen mit nahezu beliebiger Korngrößenverteilung zu simulieren.

Der Einsatz dieser Simulation für die Optimierung von Betonmischungen ist jedoch nur in begrenztem Umfang möglich, da das Programm bisher keine Möglichkeit bot, die Wirkung der interpartikulären Kohäsionskräfte zu berücksichtigen. Diese Kräfte sind vor allem im Feinstkornbereich der granularen Mischungen wirksam und sorgen dort für eine Verklumpung - die sogenannte Agglomeration - der Partikel. Die Agglomeration beeinflusst die Struktur der Partikelpackung und sorgt für eine Senkung der Packungsdichte. Sie ist der Hauptgrund für starke Abweichungen, die man zwischen den Ergebnissen der bisherigen Simulation und experimentell im Labor ermittelten Packungsdichten von Mischungen mit hohem Feinstoffanteil beobachten kann. Raschdorf erarbeitete bereits einige Ansätze, um die Wirkung der Agglomeration in die Simulation mit einzubeziehen. Er musste allerdings erkennen, dass sich diese nicht in der Praxis anwenden lassen, da die für die Berechnungen notwendigen Daten nur schwer oder gar nicht zu ermitteln waren.

Der Schwerpunkt der vorliegenden Arbeit liegt auf der Entwicklung von Datenstrukturen und Algorithmen, die eine Simulation des Wachstums von Agglomeraten und ihrer Wirkung auf die Partikelpackung ermöglichen. Von zentraler Bedeutung ist dabei die Entwicklung der Simulation „PSD-Morph“. Sie ist als eine zusätzliche Komponente der Packungssimulation anzusehen. Mit „PSD-Morph“ können die durch die Agglomeration verursachten Veränderungen in der Korngrößenverteilung (engl. Particle Size Distribution, PSD) approximiert werden. Aus den so gewonnenen Daten kann mit „RaSim“ eine Packung simuliert werden, in der die Auswirkungen der Agglomeration berücksichtigt sind.

Die Agglomeration von feinen Partikeln ist ein komplexer Vorgang, der von vielen verschiedenen Faktoren beeinflusst wird. Viele der chemischen und physikalischen Kenngrößen, die mit der Agglomeration in Verbindung stehen, lassen sich messtech-

nisch nur schwer ermitteln. Die PSD-Morph Simulation bildet deshalb den Vorgang des Agglomeratewachstums in abstrakter Form nach. Dazu wird der Wachstumsprozess in diskrete Schritte unterteilt, die jeweils Agglomerate bis zu einer bestimmten Größe betrachten. In jedem dieser Schritte wird der Anteil stabiler Partikel - also Partikel die nicht agglomerieren - und der Anteil der agglomerierenden Partikel - die sich verändern, indem sie sich mit anderen zu größeren Partikeln zusammenschließen - bestimmt. Die Agglomerate jedes Simulationsschrittes wachsen bis zum Erreichen einer Sättigungsgrenze. Dieser Vorgang wiederholt sich, bis die Mischung frei von zur Agglomeration neigenden Partikeln ist.

Das Anwachsen der Agglomerate durch die zufällige Anlagerung weiterer Partikel wird durch einen speziellen DLA-Algorithmus (engl. Diffusion Limited Aggregation) simuliert. Dieser - in seiner Grundform simple Algorithmus - ist in der Lage, die poröse fraktalartige Struktur vieler Agglomerate nachzubilden. In seiner speziell für den Einsatz innerhalb der PSD-Morph Simulation entwickelten Form kann dieser Algorithmus eine vorgegebene Korngrößenverteilung verarbeiten und somit die reale Größenverteilung der Primärpartikel innerhalb der fertigen Agglomerate virtuell nachbilden. Damit ist es möglich, den Anteil der Agglomerate innerhalb einer agglomerierten Korngrößenverteilung ebenso zu approximieren wie die Porosität dieser Agglomerate. Mit diesen Daten kann dann die durch die Agglomeration verringerte Packungsdichte in der von „RaSim“ simulierten Packung berechnet werden.

Einen weiteren Schwerpunkt dieser Arbeit bildet die Entwicklung der Online-Fraktionierung, die eine Ergänzung der von Raschdorf entwickelten Fraktionierung darstellt. Sie ist in der Lage, flexibel auf die Besonderheiten der gegebenen Korngrößenverteilung zu reagieren. Packungen die von großen Partikeln dominiert werden, und deren Simulation mit der ursprünglichen Fraktionierung problematisch ist, erzielen mit der Online-Fraktionierung verlässlichere Ergebnisse.

Um den praktischen Einsatz der Simulation zu erleichtern, wurde sie mit einer grafischen Oberfläche versehen. Die Benutzung ist dadurch auch für Nichtexperten möglich. Zum Zeitpunkt der Ausarbeitung dieser Arbeit wird das Programm am Institut für Nichtmetallische Werkstoffe der TU-Clausthal zur Unterstützung der Optimierung von Baustoffmischungen sowie am Institut für Ziegelforschung in Essen im Rahmen eines Projektes zur Verbesserung der Scherbenfestigkeit eingesetzt.

1.1 Aufbau der Arbeit

Diese Arbeit gliedert sich in 8 Kapitel. Im Anschluss an diese Einleitung erklärt das Kapitel Grundlagen die grundlegenden Fakten, die für diese Arbeit wichtig sind. Der

Baustoff Beton wird beschrieben und welche Bedeutung die Raumausfüllung seiner granularen Bestandteile für seine Eigenschaften hat. Dann werden die Kenngrößen definiert, die innerhalb dieser Arbeit verwendet werden. Danach erfolgt die Beschreibung einiger grundlegender Kugelpackungsarten sowie wichtiger Korngrößenverteilungen. Schließlich folgt eine Klassifizierung der Methoden zur Bestimmung der Raumausfüllung von granularen Stoffen. Das Kapitel 3 verschafft einen Überblick über das zuvor von Raschdorf entwickelte Simulationssystem „RaSim“. Im Kapitel 4 werden besondere Korngrößenverteilungen betrachtet, die in der Packungssimulation Probleme bereiten können. Dann wird die Online-Fraktionierung vorgestellt, die flexibel auf die Besonderheiten solcher Verteilungen reagieren kann und damit verlässlichere Simulationen erlaubt. Kapitel 5 beschäftigt sich dann mit dem Phänomen der Agglomeration. Es werden die Ursachen, die verschiedenen Einflussfaktoren und die Wirkung auf die Partikelpackung betrachtet. Im Anschluss erfolgt die Beschreibung der „PSD-Morph“ Simulation von ihren theoretischen Grundlagen bis zu den Implementierungsdetails. Schließlich wird der kombinierte Einsatz von „PSD-Morph“ und „RaSim“ zur simulativen Bestimmung der Packungsdichte von agglomerierten Partikelmischungen erklärt mit der abschließenden Vorstellung einiger Simulationsergebnisse. Kapitel 6 beschreibt Algorithmen, mit denen sich komplexe, oft fraktalartige, Wachstums- und Ausbreitungsmuster simulieren lassen. Dabei wird insbesondere ein DLA-Algorithmus vorgestellt, der von der „PSD-Morph“ Simulation für das Wachstum der Agglomerate eingesetzt wird. In Kapitel 7 wird ein Fazit über den Entwicklungsstand des Simulationssystems zum Abschluss dieser Arbeit gezogen und es werden einige kurze Ideenskizzen für mögliche zukünftige Entwicklungen vorgestellt. Im Anhang A folgt schließlich eine kurze Beschreibung der Bedienelemente der Simulationsprogramme.

Kapitel 2

Grundlagen

In diesem Kapitel werden die Grundlagen beschrieben, auf denen die Computersimulationen von Betonmischungen als Kugelpackungen basieren. Zunächst wird der Baustoff Beton betrachtet, seine Zusammensetzung, seine Funktionen und wie die Raumauffüllung seiner trockenen granularen Bestandteile seine Eigenschaften beeinflusst. Dann werden wichtige Kenngrößen vorgestellt, die der Charakterisierung von Partikelmischungen dienen. Danach folgt die Beschreibung besonderer Kugelpackungsarten mit ihren Eigenschaften und Effekten. Basierend auf den Überlegungen zur Struktur von Kugelpackungen mit hoher Raumauffüllung werden dann spezielle Korngrößenverteilungen vorgestellt. Zum Schluss folgt eine Übersicht über die Techniken zur Bestimmung der Raumauffüllung von Partikelmischungen.

2.1 Der Baustoff Beton

Als Beton bezeichnet man einen künstlichen Stein, der aus der Verbindung einer Gesteinskörnung mit einem Bindemittel - in der Regel Zement - besteht. Unter Zugabe von Wasser kommt es zu einer exothermen chemischen Reaktion, bei der sich der trockene, pulverförmige Zement zuerst verflüssigt, um schließlich zu erhärten und zusammen mit der Gesteinskörnung ein disperses Baustoffgemisch zu bilden. Solange der Beton flüssig ist, kann er auf vielfältige Art verarbeitet werden: er kann in beliebige Formen gegossen werden, lässt sich durch Leitungen pumpen um beispielsweise hoch gelegene Stockwerke beim Bau von Hochhäusern zu erreichen und er kann als Spritzbeton beim Ausbau von Tunneln oder zur Stabilisierung von Felswänden eingesetzt werden. Diese Vielfältigkeit erlaubt die Realisierung von Bauprojekten, die ohne Beton nur schwer oder gar nicht umsetzbar wären.

Die Entwicklung des neuzeitlichen Betons begann erst im 18. Jahrhundert. Vorläufer unserer heutigen modernen Betone gab es jedoch schon in der Antike. Besonders

die Römer waren für ihre hochentwickelte Betonkultur bekannt. Sie verwendeten ein Gemisch aus gebranntem Kalkstein, Sand und Steinen sowie den aus vulkanischer Asche gewonnenen Puzzolanen. Zahlreiche bis heute erhaltene Bauwerke wie das Kolosseum, die Kuppel des Pantheon und die Aquädukte zeugen von der Festigkeit und Dauerhaftigkeit dieses Baustoffs.

Heutige Betone sind für ihren jeweiligen Anwendungszweck optimiert und können aus einer Vielzahl unterschiedlicher Stoffe bestehen. Grundsätzlich lassen sich diese Stoffe in 5 Gruppen einteilen. Die wichtigsten 3 Bestandteile eines Betons sind:

Zuschläge: Gesteinskörnung, d.h. Gesteinsbrocken und Sand

Zement: als Bindemittel

Wasser: für die Hydratation

Darüber hinaus können auch eine Reihe fester oder flüssiger Zusätze verwendet werden, die sowohl die Verarbeitbarkeit als auch die Endigenschaften des Betons beeinflussen [Wol11]. Diese Zusätze unterteilt man in 2 Gruppen: die Zusatzmittel und die Zusatzstoffe.

Zusatzmittel: die durch chemische oder physikalische Wirkung die Eigenschaften des Betons ändern. Ihr Volumenanteil ist dabei aber so gering, dass er auf die Gesamtmenge des Betons keinen Einfluss hat. Einige wichtige Gruppen von Zusatzmitteln sind:

- Fließmittel: zur Verminderung des Wasseranspruchs
- Verzögerer: verlängern die Erstarrungszeit
- Stabilisierer: verhindern die Wasserabsonderung
- Dichtungsmittel: reduzieren die kapillare Wasseraufnahme

Zusatzstoffe: feine anorganische Stoffe, die sowohl inaktiv als Füllmaterial das Porenvolumen verringern, als auch aktiv in der Mischung reagieren können. Ihre Zugabemenge ist in der Regel so hoch, dass sie bei der Stoffraumrechnung berücksichtigt werden muss.

Mithilfe dieser Zusätze kann für jede Anwendung die optimale Betonmischung erstellt werden. Man strebt dabei meist eine möglichst geringe Porosität des fertig ausgehärteten Betons an, da sich diese negativ auf die Druckfestigkeit auswirkt. Außerdem widersteht ein Beton mit weniger und kleineren Poren besser den Umwelteinflüssen, der Baustoff wird dadurch dauerhafter und nachhaltiger. Eine wichtige Rolle spielt dabei

die Raumauffüllung der trockenen Bestandteile der Mischung. Das Volumen der verbleibenden Hohlräume zwischen den Partikeln sollte möglichst gering gehalten werden. Diese Hohlräume werden beim Anmachen des Betons durch das Anmachwasser gefüllt, wenn dieses der Mischung hinzugefügt wird. Je größer das Hohlraumvolumen ist, umso größer ist auch der Wasseranspruch der Mischung.

Natürlich ist Wasser ein wesentlicher Bestandteil des Betons, es löst einerseits die Hydratation des Zements aus, die letztlich zur Erhärtung des Gemisches führt, andererseits macht es den Beton auch fließfähig und ist damit wichtig für die Verarbeitbarkeit. Doch Wasser, das nicht für die Hydratation aufgebraucht wird, verbleibt zunächst im Beton, es verlangsamt die Festigkeitsentwicklung und führt zu einer höheren Porosität. Das Verhältnis zwischen Anmachwasser und Zement - der sogenannte w/z -Wert - sollte deshalb möglichst gering sein. Typische Werte für den w/z -Wert liegen zwischen 0,4 und 0,55. Durch eine hohe Raumauffüllung der trockenen Bestandteile sinkt also der Wasseranspruch der Mischung wodurch ein niedriger w/z -Wert eingehalten werden kann, was sich wiederum positiv auf die Eigenschaften des fertigen Betons auswirkt. Eine ausführlichere Beschreibung der Zusammenhänge zwischen Raumauffüllung, Wasseranspruch, Hydratation sowie Porosität und Dauerhaftigkeit des Betons findet sich in [Pal09].

2.2 Wichtige Kenngrößen

Die folgenden Kapitel definieren die für diese Arbeit wichtigen Kenngrößen. Die Darstellung der Symbole und Formeln entspricht der in Raschdorfs Arbeit [Ras10], insofern sie dort bereits verwendet wurden. Alle Fakten stammen aus [Sti09, VDZ08, BDM⁺88].

2.2.1 Die Raumauffüllung

Stoffsysteme wie Betonmischungen werden allgemein als disperse Systeme bezeichnet. Solche Systeme setzen sich aus zwei Phasen zusammen: die festen Partikel der Gesteinskörnung, des Zements (vor der Hydratation) und eventuell weiterer Zusatzstoffe bilden die *disperse Phase*. Das die Partikel umgebende Medium nennt man die *kontinuierliche Phase*. Die kontinuierliche Phase kann sowohl gasförmig, flüssig als auch fest sein. So sind die Partikel vor der Zugabe des Anmachwassers nur von Luft umgeben, nach der Zugabe des Anmachwassers besteht das umgebende Medium aus dem flüssigen Zementleim der schließlich zum festen Zementstein aushärtet. Disperse Stoffsysteme kann man unter anderem durch ihre chemische Zusammensetzung,

ihre disperse Struktur sowie die Wechselwirkungen an den Phasengrenzen charakterisieren. In der hier vorliegenden Arbeit beschränken wir uns auf die Betrachtung der trockenen Betonmischung vor der Zugabe des Anmachwassers, das heißt, wir gehen stets von einer gasförmigen kontinuierlichen Phase aus. Außerdem untersuchen wir ausschließlich die disperse Struktur des Gemisches und welche Faktoren diese Struktur beeinflussen, andere Charakteristika werden nicht näher betrachtet.

Die für uns wichtigste Kenngröße ist die Raumausfüllung, die wir mit Φ bezeichnen. Sie beschreibt den Anteil der dispersen Phase am Gesamtvolumen des Stoffsystems. Um eine technisch klarere Definition zu ermöglichen gehen wir davon aus, dass sich das Partikelgemisch in einem geschlossenen Container befindet, den es vollständig ausfüllt. Das führt zu folgender Definition:

Definition 2.1 (Raumausfüllung) Die **Raumausfüllung** Φ einer Partikelpackung ist der Anteil des Volumens aller Partikel am Gesamtvolumen, das die Packung beansprucht. Es sei V_K das Gesamtvolumen aller Partikel der Packung und V_C das Volumen des Containers, der die Packung enthält, dann beträgt Φ :

$$\Phi = \frac{V_K}{V_C}.$$

Da Partikelpackungen auch in vielen verschiedenen anderen Forschungsrichtungen zur Anwendung kommen, existieren für die Raumausfüllung mehrere Synonyme. Ein häufig in der Literatur auftauchender Begriff, der auch in dieser Arbeit verwendet wird, ist die **Packungsdichte** (engl. packing density). Hauptsächlich in der mechanischen Verfahrenstechnik wird der Begriff **Volumenkonzentration** benutzt.

Den Anteil der kontinuierlichen Phase am Gesamtvolumen des Stoffsystems ($1 - \Phi$) bezeichnen wir als Hohlraumanteil oder auch **Porosität** ϵ .

Definition 2.2 (Porosität) Es sei V_K das Gesamtvolumen aller Partikel der Packung und V_C das Volumen des Containers, der die Packung enthält, dann ist mit $V_H = V_C - V_K$ das Hohlraumvolumen der Packung gegeben. Die **Porosität** ϵ der Packung beträgt:

$$\epsilon = \frac{V_H}{V_C} = 1 - \Phi.$$

Eine weitere Größe, die eng mit der Raumausfüllung und der Porosität zusammenhängt, ist die **Koordinationszahl** k .

Definition 2.3 (Koordinationszahl) Die **Koordinationszahl** k eines Partikels in einer Packung gibt die Zahl seiner Berührungspunkte mit den Nachbarpartikeln an.

Partikel in stark verdichteten Packungen haben eine hohe Koordinationszahl, während Agglomerate eine hohe Porosität aufweisen, ihre Partikel haben eine niedrige Koordinationszahl. Eine besondere Rolle spielt die Koordinationszahl in regulären Kugelpackungen, die in Kapitel 2.3 vorgestellt werden. In solchen Packungen sind die Koordinationszahlen durch die Packungsstruktur genau festgelegt. Bei den 5 in Kapitel 2.3.1 vorgestellten monodispersen regulären Kugelpackungen hat sogar jede Kugel einer Packung dieselbe Koordinationszahl. In diesen besonderen Fällen hängt die Koordinationszahl direkt mit der Raumausfüllung zusammen, das heißt die Packungen mit der höchsten Koordinationszahl haben auch die höchste Raumausfüllung.

Raumausfüllung und Porosität eines Partikelgemisches hängen von mehreren verschiedenen Faktoren ab, im wesentlichen sind dies:

Partikelform: in den in dieser Arbeit verwendeten Modellen gehen wir in der Regel von kugelförmigen Partikeln aus.

Verdichtende Kräfte: von außen auf die Partikelpackung einwirkende Kräfte wie z.B. Schwerkraft, Verdichtungsdruck oder Rütteln drängen die Partikel enger zusammen und erhöhen somit die Raumausfüllung.

Interpartikuläre Kräfte: zwischen den Partikeln wirkende Haftkräfte führen zur Agglomeration und verhindern die Entstehung einer dichten Packung.

Korngrößenverteilung: um die Hohlräume der Packung aufzufüllen, müssen entsprechend kleine Partikel im richtigen Mengenverhältnis vorhanden sein.

Die Korngrößenverteilung hat eine ganz grundlegende Bedeutung, sie entscheidet über die maximal mögliche Raumausfüllung der Partikelpackung. In den Kapiteln 2.3, 2.4 und 2.5 werden einige wichtige Packungsarten und die zugehörigen Korngrößenverteilungen beschrieben. Das folgende Kapitel definiert die für uns wichtigen Darstellungen der Korngrößenverteilung.

2.2.2 Die Korngrößenverteilung

Besteht eine Partikelpackung nur aus Partikeln derselben Größe, so nennt man sie *monodispers*. Vermischt man zwei oder drei monodisperse Partikelsysteme mit unterschiedlichen Korngrößen miteinander, so nennt man die entstehenden Mischungen *bidispers* bzw. *tridispers*. Solche noch relativ einfachen Partikelsysteme sind vor allem für theoretische Überlegungen und Experimente interessant, da man an ihnen wichtige grundlegende Packungsstrukturen untersuchen kann. Die Korngrößen der Partikel in einer Betonmischung sind jedoch in der Regel in einem sehr breiten Intervall

Index	Mengenart	Verteilungen	Bemerkungen
$r = 0$	Anzahl ($\sim x^0$)	$Q_0(x), q_0(x)$	sehr häufig
$r = 1$	Länge ($\sim x^1$)	$Q_1(x), q_1(x)$	sehr selten
$r = 2$	Fläche ($\sim x^2$)	$Q_2(x), q_2(x)$	häufig
$r = 3$	Volumen ($\sim x^3$)	$Q_3(x), q_3(x)$	häufig
$r = 3^*$	Masse ($\sim \rho \cdot x^3$)	$Q_{3^*}(x), q_{3^*}(x)$	sehr häufig

Tabelle 2.1: Systematische Indizierung der Mengenarten nach [Sti09]

verteilt. Die kleinsten Partikel der Zusatzstoffe können kleiner als $1\mu\text{m}$ sein, während die größten Partikel der Gesteinskörnung größer als 1cm sind. Wir bezeichnen solche Mischungen als *polydispers*. Als **Polydispersität** können wir anschaulich die Breite des Intervalls verstehen, in dem die Korngrößen aller Partikel einer Mischung verteilt sind.

Definition 2.4 (Polydispersität) Ist mit x_{\min} die kleinste und mit x_{\max} die größte Korngröße einer Partikelmischung gegeben, dann nennen wir

$$P = \frac{x_{\max}}{x_{\min}}$$

die Polydispersität der Mischung. Für $P = 1$ heißt die Mischung *monodispers*, für $P > 1$ heißt sie *polydispers*.

Neben der Polydispersität benötigen wir vor allem noch die **Korngrößenverteilung** (abgek.: KGV; engl. particle size distribution, abgek.: PSD) als wesentliches Charakteristikum einer Partikelmischung. Die Korngrößenverteilung gibt die Menge der Partikel pro Korngröße an, sie wird meistens als **Verteilungssumme** $Q_r(x)$ angegeben.

Definition 2.5 (Verteilungssumme) Eine KGV lässt sich durch die **Verteilungssumme** $Q_r : \mathbb{R} \rightarrow [0, 1]$ beschreiben. $Q_r(x)$ gibt den Mengenanteil der Partikel mit Korngröße $d \leq x$ an. Der Index r gibt dabei an, um welche Mengenart es sich handelt. Eine Auflistung der möglichen Mengenarten findet sich in Tabelle 2.1.

Eine weitere Möglichkeit der Darstellung ist die als **Verteilungsdichte** $q_r(x)$. Sie lässt sich aus der Verteilungssumme $Q_r(x)$ ermitteln.

Definition 2.6 (Verteilungsdichte) Eine KGV lässt sich durch die **Verteilungsdichte** $q_r : \mathbb{R} \rightarrow [0, 1]$ mit

$$q_r(x) = \frac{dQ_r(x)}{dx}$$

beschreiben, wenn Q_r stetig differenzierbar ist. Der Index r gibt an, um welche Mengenart es sich handelt (Tabelle 2.1).

In der Praxis werden meistens Volumen- oder Massenverteilungen verwendet, d.h. $Q_3(x)$, $q_3(x)$ bzw. $Q_{3^*}(x)$, $q_{3^*}(x)$. Zur Bestimmung dieser Verteilungen existieren verschiedene bewährte Messverfahren. Bei grobkörnigem Material wiegt man den Durchgang durch Siebe mit unterschiedlicher Maschenweite, so erhält man $Q_{3^*}(x)$, wobei x die jeweilige Maschenweite ist. Ist die Reindichte ρ des Materials für jede Korngröße gleich, so ist $Q_3(x) \approx Q_{3^*}(x)$. Bei feinkörnigem Material benutzt man Messverfahren, die auf der Laserbeugung basieren.

Es ist hier zu beachten, dass keines der in der Praxis verwendeten Verfahren eine unendlich hohe Auflösung liefert. Es können daher nur die Mengenanteile von Partikeln gemessen werden, deren Korngrößen innerhalb genau definierter Intervalle liegen. Wir bezeichnen diese Intervalle hier als *Korngrößenklassen*¹. Bei der Messung von Siebdurchgängen beispielsweise wird eine Korngrößenklassen durch eine obere Maschenweite x_o und die nächst kleinere Maschenweite x_u definiert. Die Verteilungsdichte aus Definition 2.6 berechnet sich dann wie folgt:

$$q_r(x) = \frac{Q_r(x_o) - Q_r(x_u)}{x_o - x_u}.$$

Um eine Partikelpackung mit einer vorgegebenen KGV per Computer simulieren zu können, muss zunächst eine gewisse Anzahl von virtuellen Partikeln (in der Regel Kugeln) erzeugt werden, deren Durchmesser der vorgegebenen Verteilung folgt. Dazu wird eine Anzahlverteilung benötigt. Es ist also erforderlich, die Verteilung von einer Volumen- in eine Anzahlverteilung (und umgekehrt) konvertieren zu können. Das ist mit:

$$q_3(x) = \frac{x^3 \cdot q_0(x)}{\int_{x_{\min}}^{x_{\max}} y^3 \cdot q_0(y) dy}$$

und

$$q_0(x) = \frac{x^{-3} \cdot q_3(x)}{\int_{x_{\min}}^{x_{\max}} y^{-3} \cdot q_3(y) dy}$$

möglich. Mit x_{\min} bzw. x_{\max} wird der kleinste bzw. größte Partikeldurchmesser der Verteilung bezeichnet. Allgemein gilt für die Umrechnung zwischen verschiedenen

¹In der Praxis ist dafür der Begriff *Fraktionen* gebräuchlich. Da wir diesen Begriff innerhalb dieser Arbeit aber auch für die hierarchische Simulation (siehe Kapitel 3.1.2) verwenden, benutzen wir statt dessen *Korngrößenklassen*.

Mengenarten:

$$q_r(x) = \frac{x^{r-s} \cdot q_s(x)}{\int_{x_{\min}}^{x_{\max}} y^{r-s} \cdot q_s(y) dy} \quad \text{mit } r, s = 0, \dots, 3,$$

wobei r und s die Indizes der Ziel- bzw. Quellmengenart gemäß der Tabelle 2.1 sind.

2.2.3 Die fraktale Dimension

Viele der Objekte, mit denen sich diese Arbeit beschäftigt, haben eine fraktale Struktur. Das trifft sowohl auf die Kugelpackungen, wie beispielsweise die apollonischen Packungen, insbesondere aber auch auf die Agglomerate zu. Der Begriff der Fraktale stammt aus der von Benoît B. Mandelbrot² begründeten fraktalen Geometrie. Fraktale sind komplexe, selbstähnlich Gebilde. Die Selbstähnlichkeit bedeutet, dass jeder noch so kleine Ausschnitt des Gebildes, wenn man ihn entsprechend vergrößert, dem Ausgangsgebilde ähnelt. Nach dieser Definition gibt es in der Natur keine echten Fraktale, denn jedes komplexe Objekt erreicht irgendwann eine Maximalgröße, ab der es nicht weiter wächst. Umgekehrt gelangt man bei der Zerlegung eines jeden komplexen Objektes schließlich zu einem kleinsten Grundmuster, das sich nicht weiter zerlegen lässt. Trotzdem weisen viele Objekte innerhalb eines bestimmten Skalierungsintervalls eine fraktalartige Struktur auf.

Fraktale können über ihre **fraktale Dimension** D charakterisiert werden. Für 3-dimensionale Gebilde nimmt die fraktale Dimension Werte zwischen 1 und 3 an. Von einem Fraktal spricht man aber nur dann, wenn $D \notin \mathbb{N}$ gilt. (D muss also eine „gebrochene“ Zahl sein, woraus sich auch der Begriff „Fraktal“ herleitet.) Drei einfache Beispiele für 2-dimensionale fraktale Gebilde sind in Abbildung 2.1 dargestellt. Jeweils ausgehend von einem einfachen, aus mehreren Kreisscheiben zusammengesetzten, Grundmuster wird ein komplexeres, größeres Gebilde erzeugt, indem mehrere Grundmuster in der gleichen Weise zusammengesetzt werden, wie die Kreisscheiben im Grundmuster selbst. Dieses neue Gebilde übernimmt dann selbst die Rolle des Grundmusters, aus dem das nächst größere Gebilde zusammengesetzt wird. Auf der Grundlage dieser Erklärung kann die fraktale Dimension definiert werden.

Definition 2.7 (fraktale Dimension) *Ist N die Anzahl der Teile, aus denen ein fraktales Gebilde besteht, und ist S der Skalierungsfaktor, um den das Gebilde größer ist als jedes seiner Teile, dann berechnet sich die fraktale Dimension D des Gebildes zu:*

$$D = \frac{\log(N)}{\log(S)}.$$

²Benoît B. Mandelbrot, 1924 - 2010, französisch-US-amerikanischer Mathematiker

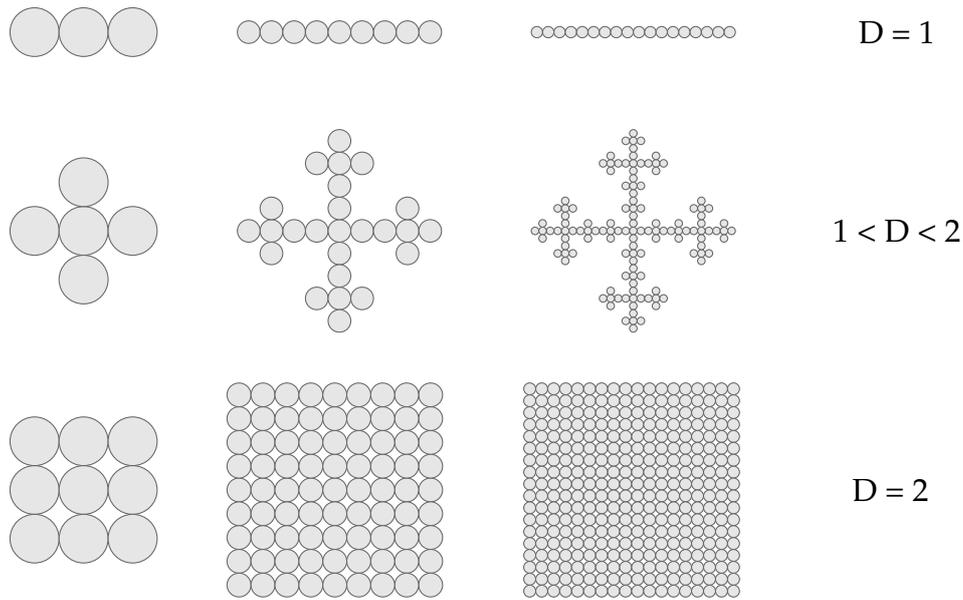


Abbildung 2.1: Selbstähnliche (fraktale) zweidimensionale Gebilde (nach [BDM⁺88])

Für das obere und untere Gebilde aus Abbildung 2.1 ergeben sich als Werte für die fraktale Dimension $D = 1$ bzw. $D = 2$. Das schneeflockenartige Gebilde in der Mitte hat die fraktale Dimension $D = \frac{\log(5)}{\log(3)} = 1,46497$. Zwischen der Anzahl Partikel N_p (in Abbildung 2.1 die Anzahl der Kreisscheiben) mit Radius r , die ein fraktales Gebilde mit Radius R bilden und der fraktalen Dimension dieses Gebildes D besteht der Zusammenhang:

$$N_p = \left(\frac{R}{r}\right)^D.$$

Angewendet auf die Berechnung der Raumaufüllung Φ für eine 3-dimensionale Flocke (Agglomerat) ergibt sich:

$$\Phi = (1 - \epsilon) = \frac{N_p \cdot r^3}{R^3} = \frac{\left(\frac{R}{r}\right)^D}{\left(\frac{R}{r}\right)^3} = \left(\frac{R}{r}\right)^{D-3}.$$

Das bedeutet, mit zunehmenden Durchmesser der Flocke sinkt ihre Raumaufüllung Φ bzw. ihre Porosität ϵ steigt an. Diese Tatsache hat Auswirkungen auf die Raumaufüllung von Partikelmischungen mit hohem Feinstoffanteil, wie dies bei Betonmischungen der Fall ist. Feine Partikel neigen dazu, Agglomerate mit fraktalartiger Struktur zu bilden, wodurch die Raumaufüllung der gesamten Partikelpackung gesenkt wird. In Kapitel 5 wird diese Thematik detailliert behandelt.



Abbildung 2.2: Reguläre Packung aus Weihnachtsbaumkugeln

2.3 Reguläre Kugelpackungen

Bei der Simulation von Partikelpackungen modellieren wir die Partikel in der Regel als Kugeln. Die Packungen sind dann also immer Kugelpackungen. Einige verschiedene Arten dieser Packungen und ihre Eigenschaften werden in diesem und den folgenden Kapiteln genauer beschrieben.

Von regulären Kugelpackungen spricht man, wenn die Kugeln in einem regelmäßigen Gitter angeordnet sind. Solche Packungen entstehen durch die Einwirkung von Kräften, die jede der Kugeln an eine festgelegte Position innerhalb der Packungsstruktur bewegen. Das ist beispielsweise bei Atomen in einem Kristallgitter der Fall. Insbesondere können Metalle, die in reiner Form verwendet werden, wie Kupfer oder Aluminium, als monodisperse Packungen angesehen werden, da sie nur aus gleichartigen Atomen aufgebaut sind. Die Art der zwischen diesen Atomen wirkenden Bindungskräfte führen zu einer Anordnung in einem bestimmten, regelmäßigen Gitter.

Weitere, mehr aus dem Alltag vertraute Beispiele findet man bei der Verpackung von kugelförmigen Objekten. Dabei kann es sich um Früchte, wie Äpfel oder Orangen handeln, die zum Transport in Kisten verpackt werden; oder um Weihnachtsbaumkugeln, die in zylinderförmigen Plastikverpackungen untergebracht sind wie in Abbildung 2.2. Bei der Lagerung solcher Objekte ist man häufig bestrebt, sie nach einem bestimmten Muster anzuordnen. Das liegt vor allem daran, dass bestimmte Anordnungen eine höhere Raumausfüllung erzielen als andere. Man spart also Platz. In einigen Fällen kann aber auch die bessere Verteilung der Kräfte eine Rolle spielen. Da z.B. Weihnachtsbaumkugeln sehr zerbrechlich sind, packt man sie am besten so, dass sie eine möglichst hohe Koordinationszahl haben. Die Wirkung der Schwerkraft und Stöße können sich so auf viele Punkte verteilen und die Kugeln sind besser vor Be-

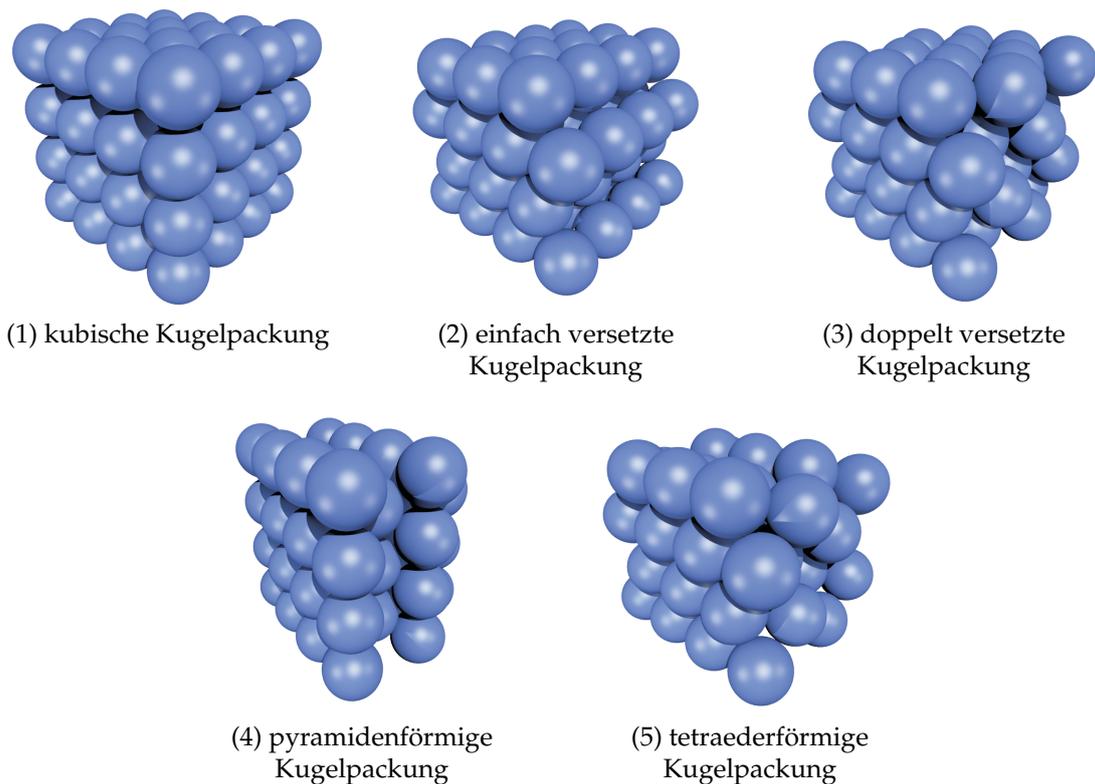


Abbildung 2.3: Die 5 möglichen Kugelpackungen nach [WW37]

schädigungen geschützt.

Eine Besonderheit von geordneten Kugelpackungen ist die mathematisch analytische Zugänglichkeit ihrer Eigenschaften. Je nach Art der Anordnung kann die Position jeder Kugel exakt ermittelt werden. Damit ist es auch möglich, Eigenschaften wie die Koordinationszahl oder die Raumauffüllung direkt zu berechnen.

2.3.1 Monodisperse reguläre Kugelpackungen

Laut White und Walton [WW37] kann man kugelförmige Partikel grundsätzlich auf 5 unterschiedliche Arten packen. In Abbildung 2.3 sind diese Packungsarten dargestellt. In allen 5 Packungen liegen die Kugeln in Ebenen, von denen mehrere übereinander gestapelt sind. In der Abbildung sind diese Ebenen von vorne links nach hinten rechts gestapelt. In kubischen Kugelpackungen (1) berührt jede Kugel 4 Nachbarkugeln in der selben Ebene. Alle Ebenen sind gleich und so übereinander gestapelt, dass der höchste Punkt jeder Kugel der unteren Ebene den tiefsten Punkt der Kugel in der Ebene darüber berührt. Die Koordinationszahl in diesen Packungen beträgt also $k = 6$. In einfach versetzten Kugelpackungen (2) berührt jede Kugel 6 Nachbarkugeln in der selben Ebene. Die Ebenen sind aber in der selben Weise übereinander gestapelt wie

Fig. No.	Art der Kugelpackung	Einheitsprisma (Vol.)	Hohlraumant. (%)
(1)	kubisch	$8r^3$	47,64
(2)	einfach versetzt	$4r^3\sqrt{3}$	39,55
(3)	doppelt versetzt	$6r^3$	30,20
(4)	pyramidenförmig	$4r^3\sqrt{2}$	25,95
(5)	tetraederförmig	$4r^3\sqrt{2}$	25,95

Tabelle 2.2: Hohlraumanteil der 5 möglichen Kugelpackungen nach [WW37]

bei den kubischen Kugelpackungen, das ergibt $k = 8$. Bei doppelt versetzten Kugelpackungen (3) sind die Kugeln einer Ebene genauso angeordnet wie im einfach versetzten Fall. Die Ebenen sind aber in der Weise versetzt übereinander gestapelt, dass jede Kugel in der Senke zwischen 2 Kugeln der Ebene darunter zu liegen kommt. Das ergibt die Koordinationszahl $k = 10$. In einer pyramidenförmigen Kugelpackung (4) sind die Kugeln einer Ebene genau wie bei einer kubischen Kugelpackung angeordnet. Die Kugeln der jeweils nächst höheren Ebene liegen aber immer in der Senke zwischen 4 Kugeln der Ebene unter ihnen. Jede Kugel bildet also, zusammen mit den 4 Kugeln in deren Senke sie liegt, die 5 Eckpunkte einer 4-seitigen Pyramide. In tetraederförmigen Kugelpackungen (5) sind die Kugeln einer Ebene schließlich wieder wie bei den einfach oder doppelt versetzten Kugelpackungen angeordnet. Die Kugeln der nächst höheren Ebene liegen dann aber in der Senke zwischen 3 Kugeln der Ebene unter ihnen. Zusammen mit den 3 Kugeln, in deren Senke sie liegt, bildet jede Kugel den Eckpunkt eines Tetraeders. Sowohl die pyramiden- als auch die tetraederförmigen Kugelpackungen haben die Koordinationszahl $k = 12$.

Welchen Hohlraumanteil diese 5 Kugelpackungsarten enthalten ist in Tabelle 2.2 aufgelistet. Für die Berechnungen wurde um die Packungen jeweils ein Einheitsprisma gelegt, dessen Volumen, bezogen auf den Radius r der Kugeln, in Spalte 3 der Tabelle angegeben ist. Die pyramidenförmige und die tetraederförmige Kugelpackung haben beide zusammen den niedrigsten Hohlraumanteil und damit auch die höchste Raumauffüllung mit 74,05%. Die Tatsache, dass es keine Packung aus gleichgroßen Kugeln geben kann mit einer höheren Raumauffüllung als diesem Wert, ist ein sehr altes und viel diskutiertes Problem. Es ist bekannt als die „Keplersche Vermutung“³ für die Thomas Hales einen Computerbeweis lieferte [HF11, Hal05, Hal00a, Hal06].

Weitere Packungsarten lassen sich aus den 5 vorgestellten durch Kombination und Variation generieren. Beispielsweise kann aus der tetraederförmigen Kugelpackung durch Drehung der 3ten Ebene um 60° eine weitere Variante erzeugt werden, die jedoch die gleiche Raumauffüllung und die gleiche Koordinationszahl aufweist, wie die ursprüngliche Version.

³Johannes Kepler, 1571 - 1630, deutscher Mathematiker und Astronom

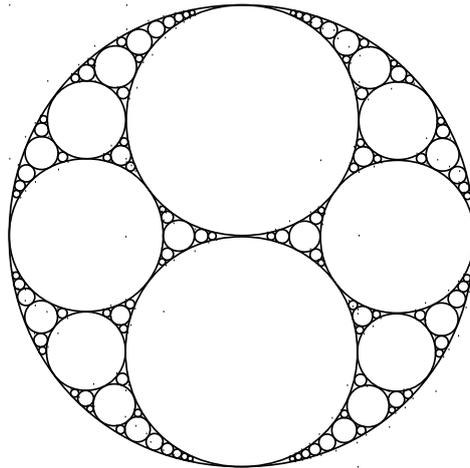


Abbildung 2.4: 2-dimensionale Apollonische Packung

2.3.2 Apollonische Kugelpackungen

Fügt man den in Kapitel 2.3.1 beschriebenen Kugelpackungen kleinere Kugeln hinzu, deren Radien und Mittelpunkte so gewählt sind, dass sie die verbleibenden Hohlräume zwischen den großen Kugeln optimal ausfüllen, kann man die Raumausfüllung deutlich über 74,05% anheben. Dieser Vorgang lässt sich beliebig fortsetzen indem die, immer kleiner werdenden, noch verbleibenden Hohlräume ebenfalls durch noch kleinere Kugeln gefüllt werden. Die Raumausfüllung kann so beliebig nahe an den theoretischen Grenzwert von 100% angenähert werden. Die dabei entstehende fraktale Struktur wird als „Apollonische Packung“⁴ bezeichnet. In der Regel werden apollonische Packungen so definiert, dass der umschließende Container selbst eine Kugel ist. In Abbildung 2.4 ist dies für den 2-dimensionalen Fall dargestellt. Die Packung beginnt hier mit 3 Initialkreisen. Die erste ist ein Einheitskreis und bildet den Container, der die Packung enthält. Die zwei anderen Kreise sind so in diesen Container eingefügt, dass sie sowohl den Container als auch sich gegenseitig berühren. In diese Struktur können nun sequentiell weitere Kreise eingefügt werden, wobei jeder dieser Kreise innerhalb des Containers liegt, disjunkt zu allen anderen Kreisen ist und maximalen Radius hat. Auf die gleiche Weise lassen sich auch beliebig höher dimensionale Packungen aufbauen.

Obwohl apollonische Packungen zu den regulären Packungen gehören, wird allgemein davon ausgegangen, dass sich ihre fraktale Dimension D nicht analytisch ermitteln lässt. Borkovec, Paris und Peikert [BPP94] haben für den 3-dimensionalen Fall numerisch einen Wert von $D \approx 2,4739465$ ermittelt. Apollonische Packungen sind interessant, da sie ein mögliches Modell für Korngrößenverteilungen mit einer hohen

⁴Apollonios von Perge, ca. 262 - 190 v. Chr., griechischer Mathematiker

Raumausfüllung bilden.

2.4 Zufällig dichte Packungen

Sind die Kugeln einer Packung nicht in einem regelmäßigen Muster angeordnet, sondern an zufälligen Positionen, wie es bei granularen Schüttungen der Fall ist, dann spricht man von zufällig dichten Packungen (engl. random close packing, RCP). Anders als bei regulären Packungen sind hier die Eigenschaften nicht rein mathematisch analytisch zugänglich. Laut Torquato [TTD00] ist der in der Literatur weit verbreitete Begriff „random close packing“ nicht einmal wohl definiert. Er hängt von der Art und Weise ab, wie die Packung erzeugt wurde und eventuell von weiteren Systemparametern. Torquato schlägt deshalb vor, von maximal zufällig gepackten Packungen (engl. maximally random jammed) zu sprechen. Dieser Begriff lässt sich präziser formulieren: er bezeichnet Packungen, in denen jede Kugel von ihren Nachbarkugeln so eingeklemmt ist, dass sie nicht verschoben werden kann ohne dafür die Positionen der Nachbarkugeln zu verändern.

Die Eigenschaften zufällig dichter Kugelpackungen sind für viele technische Anwendungen relevant, da sie ein gutes Modell für granuläre Materialien darstellen. Kenngrößen wie die Packungsdichte oder die Koordinationszahlen erlauben Aussagen über die Stabilität von Schüttkegeln oder das Fließverhalten durch einen Trichter [JN92]. Speziell für monodisperse Kugelpackungen existieren viele Arbeiten in denen die Packungsdichte mit Hilfe von Kugeln aus Blei, Stahl [BH46, Sco60], Glas oder Nylon [Rut62] ermittelt wurde. Unabhängig von Material und Größe der Kugeln ergaben sich stets ähnliche Werte, so dass ϕ_c für monodisperse Kugelpackungen mit $\approx 0,639$ [Lee70] angegeben werden kann. Der Wert liegt damit deutlich unter der für monodisperse Kugelpackungen maximal möglichen Raumausfüllung von 74,05%.

Für polydisperse zufällig dichte Kugelpackungen gilt ähnliches wie für apollonische Kugelpackungen, durch das Hinzufügen immer kleinerer Kugeln kann die Raumausfüllung beliebig nahe an den theoretischen Grenzwert von 100% angenähert werden. Bedingt durch die zufällige Lage der Kugeln treten hier aber zwei Effekte auf, die sich negativ auf die Dichte der Packung auswirken können. Der erste Effekt ist die Keilung. Sie tritt auf, wenn im Hohlraum zwischen mehreren großen Kugeln eine kleine Kugel liegt, die zu groß für diesen Hohlraum ist. Die großen Kugeln werden dann auseinander gedrängt, so dass zwischen ihnen zusätzlicher Hohlraum entsteht. Damit die Raumausfüllung der Packung dadurch nicht sinkt, müssen ausreichend kleinere Kugeln vorhanden sein, um diesen zusätzlichen Hohlraum zu füllen. Polydisperse Kugelpackungen mit hoher Raumausfüllung sind deshalb häufig so strukturiert, dass die

großen Kugeln vollständig von kleineren Kugeln umgeben sind. Hier kann jedoch der zweite Effekt auftreten, der Wandeffekt. Er sorgt für eine lokale Senkung der Raumausfüllung in den Bereichen der Packung, in denen sich kleinere Kugeln in der Nachbarschaft sehr viel größerer Kugeln befinden. Das folgende Kapitel untersucht diesen Effekt anhand eines Modells.

2.4.1 Der Wandeffekt

Große Partikel spielen innerhalb einer Packung eine zwiespältige Rolle. Einerseits nehmen sie ein großes Volumen ein, das sie - vorausgesetzt sie sind selbst nicht porös - zu 100 % ausfüllen. Das heißt, in dieser Rolle erhöhen sie die Raumausfüllung der gesamten Packung. Andererseits haben sie aufgrund ihrer Größe auch eine große Oberfläche, die eine geringe Krümmung aufweist.

Definition 2.8 (Krümmung) [BSMM99] *Die Krümmung einer Kurve im Punkt M wird eine Zahl K genannt, die die Abweichung der Kurve in der unmittelbaren Umgebung dieses Punktes von einer Geraden angibt.*

$$K = \lim_{\overline{MN} \rightarrow 0} \left| \frac{\Delta \vec{f}}{\overline{MN}} \right|$$

Mit N wird ein Punkt auf der Kurve bezeichnet, so dass der von M und N begrenzte Kurvenabschnitt die Länge $\overline{MN} > 0$ hat. Die Abweichung des Punktes N von der Geraden wird mit $\Delta \vec{f}$ bezeichnet.

Die Oberfläche eines Körpers kann in einem Punkt M je nach Richtung eine andere Krümmung aufweisen. Wir gehen aber davon aus, dass die Packung nur aus kugelförmigen Partikeln besteht. Die Oberfläche einer Kugel hat in jedem beliebigen Punkt und in jeder Richtung die gleiche Krümmung, sie berechnet sich aus dem Kehrwert des Radius zu:

$$K = \frac{1}{r}.$$

Mit zunehmendem Radius einer Kugel konvergiert ihre Krümmung also gegen den Wert 0, der Krümmung einer ebenen Fläche. Wie tief ein Partikel in die Zwischenräume seiner Nachbarpartikel eindringen kann, hängt davon ab, wie groß seine Krümmung im Verhältnis zur Krümmung seiner Nachbarpartikel ist. Entlang der Oberfläche eines großen Partikels, der sich in der Nachbarschaft kleinerer Partikel befindet, sinkt deshalb die Raumausfüllung lokal ab. Durch diesen sogenannte Wandeffekt sinkt auch die Gesamtraumausfüllung der Packung.

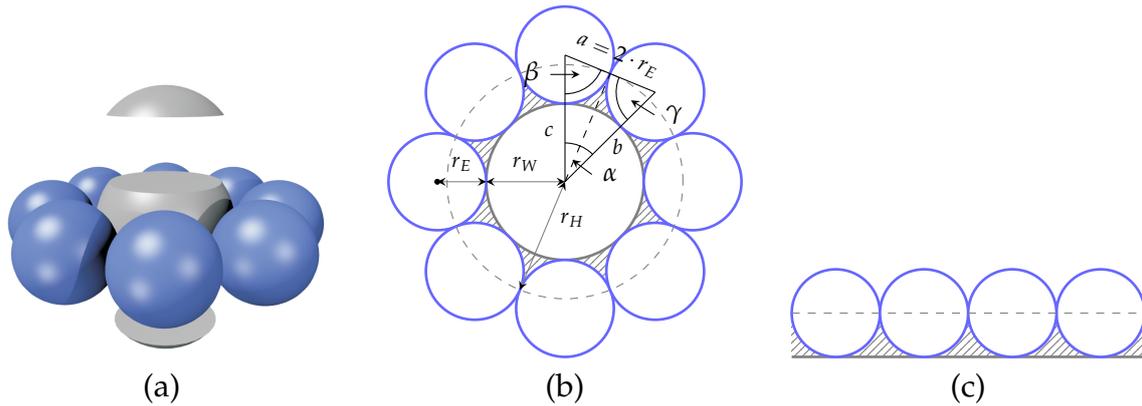


Abbildung 2.5: Modell zur Berechnung der Auswirkungen des Wandeffektes. Die Kugelkonfiguration in 3D (a), in 2D (b) und die Bestimmung des Grenzwertes an einer Ebene (c).

Im Folgenden werden zwei Modelle vorgestellt, die geeignet sind, die lokale Wirkung des Wandeffektes zu untersuchen. Beide Modelle beruhen auf vereinfachten Ausschnitten von regulären Kugelpackungen, so dass sich die Positionen der Kugeln durch die Anwendung einfacher geometrischer Gesetze berechnen lassen. Beim ersten Modell, dessen Aufbau in Abbildung 2.5 dargestellt ist, gehen wir von einer im Zentrum platzierten Kugel (grau dargestellt) aus. An ihrer Oberfläche soll der Wandeffekt berechnet werden, dazu wird sie von einem Ring aus Einheitskugeln (Radius $r_E = 1$) umgeben. Wir bezeichnen die zentrale Kugel als **Wandkugel**. Wenn die Wandkugel den Radius 1 hat, wird sie von einem Ring aus 6 Einheitskugeln umgeben. Wir wollen nun untersuchen, wie sich der verbleibende Hohlraum zwischen der Wandkugel und den sie umgebenden Einheitskugeln verändert, wenn wir die Wandkugel vergrößern. Wir vergrößern sie schrittweise, indem wir dem Einheitskugelring weitere Kugeln hinzufügen. Besteht der Ring aus n Kugeln, dann können wir den Winkel α zwischen 2 benachbarten Kugeln ermitteln, indem wir die 360° des Vollkreises entsprechend unterteilen:

$$\alpha = \frac{360^\circ}{n}.$$

Da der Mittelpunkt der Wandkugel zusammen mit den Mittelpunkten der beiden benachbarten Einheitskugeln ein gleichseitiges Dreieck aufspannen (siehe Teil b in Abbildung 2.5) gilt, dass die beiden Seiten b und c ebenso wie die gegenüberliegenden Winkel β und γ gleich groß sind. Da der Winkel α bereits bekannt ist und die Innenwinkelsumme in jedem Dreieck 180° beträgt gilt:

$$\beta(= \gamma) = \frac{180^\circ - \alpha}{2}$$

Die Länge der dem Winkel α gegenüberliegende Seite a ergibt sich aus der Summe der Radien der beiden benachbarten Einheitskugeln:

$$a = 2 \cdot r_E.$$

Laut Sinussatz ist in jedem Dreieck das Verhältnis aus der Länge einer Seite und dem Sinus des ihr gegenüberliegenden Winkels für alle drei Seiten und Winkel gleich.

Sinussatz:

$$\frac{a}{\sin \alpha} = \frac{b}{\sin \beta} = \frac{c}{\sin \gamma}$$

Damit kann die Länge der Seiten b und c berechnet werden:

$$\begin{aligned} b(=c) &= \frac{a}{\sin \alpha} \cdot \sin \beta \\ &= \frac{2 \cdot r_E}{\sin \alpha} \cdot \sin \left(\frac{180^\circ - \alpha}{2} \right) \end{aligned}$$

Der Radius der Wandkugel r_W ergibt sich zu $b - r_E$, also:

$$r_W = \frac{2 \cdot r_E}{\sin \alpha} \cdot \sin \left(\frac{180^\circ - \alpha}{2} \right) - r_E.$$

Von der Wandkugel werden das obere und das unter Kugelsegment (Polkappen) abgetrennt, so dass nur eine Scheibe zurück bleibt, deren Dicke dem Durchmesser der Einheitskugeln im umgebenden Ring entspricht. Diese Konfiguration ist in Teil a der Abbildung 2.5 dargestellt. Das Volumen der verbleibenden Kugelscheibe beträgt dann:

$$V_R = \left(\frac{4}{3} \cdot \pi \cdot r_R^3 \right) - 2 \cdot \left(\frac{\pi \cdot h^2}{3} (3 \cdot r_R - h) \right).$$

Dabei ist $h = r_W - r_E$ die Höhe der Kugelsegmente. Um den verbleibenden Hohlraum zwischen der Wandkugel und den Kugeln im umgebenden Ring berechnen zu können, muss noch eine äußere Grenze für den zu betrachtenden Raum definiert werden. Dafür wird dem Modell eine Hilfskugel hinzugefügt, deren Mittelpunkt mit dem der Wandkugel übereinstimmt. Der Radius r_H dieser Hilfskugel wird so gewählt, dass ihre Oberfläche die Berührungspunkte der Einheitskugeln schneidet. Er stimmt dann mit der Höhe des gleichseitigen Dreiecks überein (in Teil b der Abbildung 2.5 als gestrichelte Linie dargestellt). Diese Höhe teilt das gleichseitige Dreieck in zwei rechtwinklige Dreiecke, deren Hypotenusen mit b bzw. c gegeben sind. Ebenso ist mit r_E je eine der beiden Katheten gegeben. Die Höhe ($= r_H$) bildet dann die zweite Kathete, die gemäß

des Satzes des Pythagoras zu

$$r_H = \sqrt{b^2 - r_E^2}$$

berechnet werden kann. Das obere und untere Kugelsegment der Hilfskugel wird in gleicher Weise abgetrennt wie bei der Wandkugel. Das Volumen der verbleibenden Kugelscheibe bezeichnen wir mit V_H . Vom Volumen des Raumes zwischen der Hilfskugelscheibe und der Wandkugelscheibe $V_H - V_R$ müssen noch die Schnittvolumen der Einheitskugeln mit der Hilfskugel abgezogen werden, um das verbleibende Hohlraumvolumen zu ermitteln.

Bei der Überschneidung mehrerer Kugeln entstehen Schnittvolumen, die sehr unterschiedlich geformt sein können. Die Vielfalt dieser Formen nimmt überproportional mit der Anzahl der am Schnitt beteiligten Kugeln zu. Die Berechnung des Schnittvolumens kann deshalb schon bei nur 3 beteiligten Kugeln sehr komplex werden [GS87b, GS87a, GS88]. Bei dem hier benutzten Modell finden jedoch nur paarweise Überschneidungen zwischen den Einheitskugeln und der Hilfskugel statt. Die dabei entstehenden Schnittvolumen haben immer die Form einer - aus zwei Kugelsegmenten zusammengesetzten - Linse und berechnen sich zu:

$$V(K_1 \cap K_2) = \frac{\pi(r_1 + r_2 - l)^2(l^2 + 2lr_2 - 3r_2^2 + 2lr_1 + 6r_2r_1 - 3r_1^2)}{12l} \quad \text{mit } l < r_1 + r_2.$$

Dabei sind K_1 und K_2 die beiden sich schneidenden Kugeln mit den zugehörigen Radien r_1 und r_2 sowie dem Abstand ihrer Mittelpunkte l . Bezeichnen wir die Einheitskugeln mit K_1 bis K_n und die Hilfskugel mit K_H , dann beträgt das verbleibende Hohlraumvolumen V_L :

$$V_L = V_H - V_R - \sum_{i=1}^n V(K_H \cap K_i).$$

Den Anteil dieses Hohlraumvolumens am Raum zwischen der Wand- und der Hilfskugel bezeichnen wir mit H , er berechnet sich zu:

$$H = \frac{V_L}{V_H - V_R}.$$

Gemäß dieser Überlegungen wurde nun H für verschieden große Wandkugeln ermittelt. Alle notwendigen Berechnungen können mit Hilfe eines Tabellenkalkulationsprogramms durchgeführt werden. Wie sich H in Abhängigkeit von der Größe der Wandkugel verändert ist in Abbildung 2.6 dargestellt. Die Abbildung zeigt, dass H

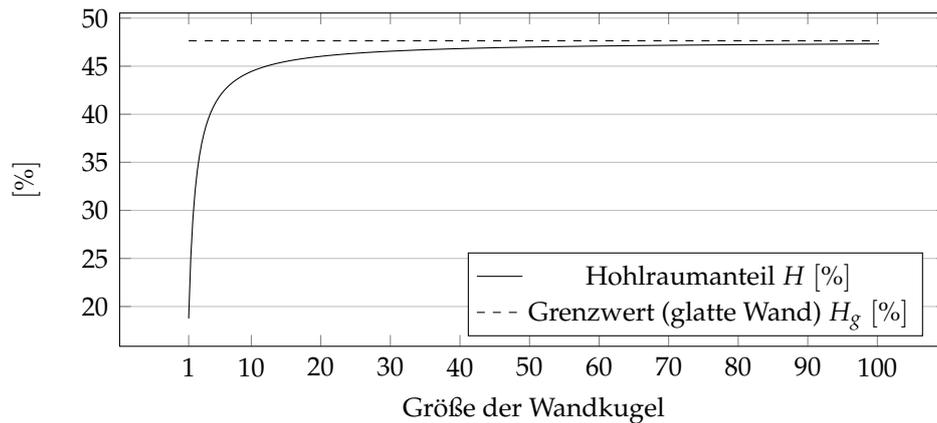
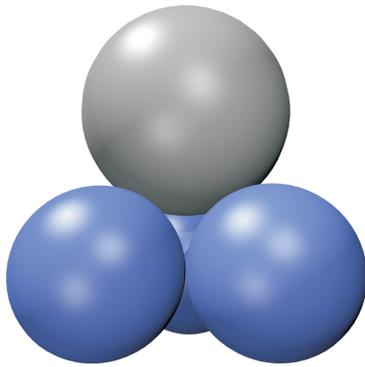


Abbildung 2.6: Auswirkungen des Wandeffektes in Abhängigkeit von der Größe der Wandkugel

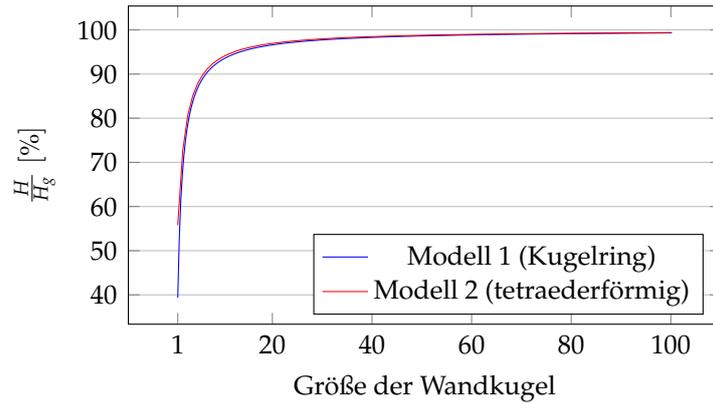
mit zunehmender Größe der Wandkugel gegen einen Grenzwert $H_g = 47,64\%$ konvergiert. Dieser Grenzwert ergibt sich, wenn man die Einheitskugeln nicht in einem Ring um die Wandkugel anordnet, sondern nebeneinander auf einer Ebene, wie es in Teil c der Abbildung 2.5 dargestellt ist.

Natürlich ergeben sich diese Werte nur bei dieser speziellen Konfiguration der Kugeln. Doch das gleiche Konvergenzverhalten des Hohlraumanteils H gegen den Grenzwert H_g lässt sich auch bei anderen Kugelanordnungen beobachten. Ein alternatives Modell für die Berechnung der Auswirkungen des Wandeffektes ist in Teil a der Abbildung 2.7 dargestellt. Bei diesem Modell sind drei Einheitskugeln zusammen mit einer Wandkugel so angeordnet, dass sie sich gegenseitig berühren und die Eckpunkte eines Tetraeders bilden. Die Hilfskugel, die zur Berechnung des verbleibenden Hohlraumvolumens gebraucht wird, ist hier im Zentrum des Tetraeders platziert. Ihr Radius ist so gewählt, dass ihre Oberfläche alle Berührungspunkte zwischen den Einheitskugeln sowie zwischen den Einheitskugeln und der Wandkugel schneidet. Die Berechnung der Positionen der Einheitskugeln, der Wand- und Hilfskugel sowie die Berechnung der Schnittvolumen erfolgt hier wieder nach den selben einfachen geometrischen Gesetzen wie schon beim ersten Modell.

Vergrößert man die Wandkugel, so erhöht sich das verbleibende Hohlraumvolumen innerhalb der Hilfskugel und konvergiert gegen einen Grenzwert, der in diesem Modell bei $H_g = 62,89\%$ liegt. Bestimmt wird dieser Wert, in dem die Wandkugel durch eine ebene Fläche ersetzt wird, die das obere Kugelsegment (Polkappe) der Hilfskugel abschneidet. Um das Konvergenzverhalten beider Modell miteinander vergleichen zu können, wurden beide Hohlraumanteile, jeweils als Anteil vom zugehörigen Grenzwert $\frac{H}{H_g}$, im Teil b der Abbildung 2.7 dargestellt. Beide Kurven verlaufen nahezu identisch. Das heißt, obwohl sich die Grenzwerte, abhängig von der Positio-



(a)



(b)

Abbildung 2.7: Ein zweites, alternatives Modell zur Berechnung der Auswirkungen des Wandeffektes mit tetraederförmiger Kugelanordnung (a) und der Vergleich der beiden Modelle (b).

nierung von Wand- und Einheitskugeln, stark unterscheiden können, konvergiert der Hohlraumanteil bei Vergrößerung der Wandkugel in jedem Fall gleich schnell gegen den jeweiligen Grenzwert. Demnach nimmt der Wandeffekt zunächst sehr stark zu, bis die Wandkugel etwa die 25-fache Größe der Einheitskugeln erreicht hat. Bei diesem Punkt hat sich der Hohlraumanteil bereits bis auf weniger als 3% dem Grenzwert genähert. Bei der 50-fachen Größe sind es weniger als 2% und bei 100-facher Größe weniger als 1%. Das heißt, dass derart große Wandkugeln aus der Sicht der Einheitskugeln - unabhängig davon, wie sie um die Wandkugel arrangiert sind - schon fast wie eine glatte Wand wirken. Für die lokale Wirkung des Wandeffektes ist die tatsächliche Größe der Wandkugel dann nicht mehr entscheidend. Es ist aber zu beachten, dass dies nicht für die globale Wirkung innerhalb einer Partikelpackung gilt. Da die Wandkugel mit zunehmender Größe auch eine immer größere Oberfläche hat, in deren Nachbarschaft entsprechend mehr kleine Kugeln Platz finden, senkt der Wandeffekt die lokale Raumauffüllung in einem größeren Bereich der globalen Packung.

Die von Raschdorf [Ras10] entwickelte Packungssimulation benutzt eine hierarchische Zerlegung (Fraktionierung) der Korngrößenverteilung, um auch Partikelmischungen mit sehr hoher Polydispersität simulieren zu können. Um die Interaktionen zwischen den Fraktionen berücksichtigen zu können, werden sogenannte Randkugeln eingesetzt, die vor allem für die Simulation des Wandeffektes notwendig sind. Die Daten aus den hier vorgestellten Modellen können als Anhaltspunkte dienen, um zu bestimmen, bis zu welcher Größe Randkugeln in den Fraktionen vorhanden sein müssen, um realistische Simulationsergebnisse zu bekommen. In den Kapiteln 3.1 und 4 wird die Problematik der Fraktionierung detailliert behandelt.

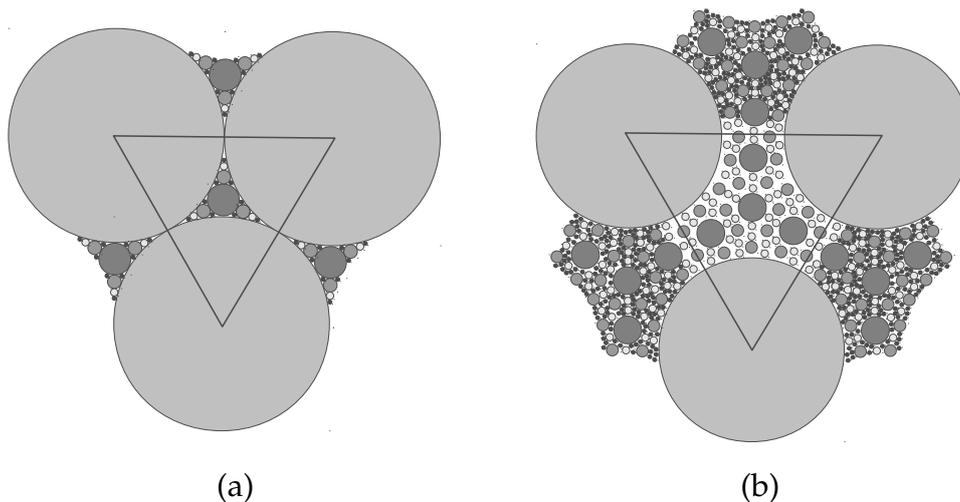


Abbildung 2.8: Apollonische Packung (a) und Gesperrte Packung (b) aus [Ver04]

2.5 Spezielle Korngrößenverteilungen

Um die für Betonmischungen angestrebten hohen Raumauffüllungen zu erreichen, benötigt man eine günstige Korngrößenverteilung. In Abbildung 2.8 sind einige grundlegende Überlegungen dargestellt, welche Eigenschaften eine Verteilung haben muss, um als günstig zu gelten. Teil a der Abbildung zeigt eine apollonische Packung, wie sie schon in den vorangegangenen Kapiteln vorgestellt wurde. Bei diesen Packungen bildet die größte Korngrößenklasse ein dichtes Packungsskelett. Die Partikel der jeweils nächst kleineren Korngrößenklasse sind so platziert, dass sie die noch verbleibenden Hohlräume optimal ausfüllen. Diese Struktur erzeugt eine hohe Raumauffüllung bei minimalem Feinstoffanteil. Allerdings ist sie in der Praxis nicht realisierbar. Da die trockenen Bestandteile einer Betonmischung eine zufällig dichte Packung bilden, würde die Korngrößenverteilung einer apollonischen Packung keine besonders hohe Raumauffüllung ergeben. Schon die Partikel der größten Korngrößenklasse würden sich nicht in der dichtest möglichen Struktur anordnen (vergleiche dichteste reguläre Kugelpackung $\approx 74\%$ [WW37] und zufällig dichte Kugelpackung $\approx 63\%$ [Lee70]). Damit würde ein größeres Hohlraumvolumen übrig bleiben als von den Partikeln der jeweils nächst kleineren Korngrößenklasse - die ihrerseits auch nicht in der dichtest möglichen Anordnung vorliegen - gefüllt werden könnte.

Eine Alternative zeigt der Teil b der Abbildung 2.8. In dieser gesperrten Packung bilden die Partikel der größten Korngrößenklasse kein dichtes Packungsskelett, sie sind auseinander gedrängt. Jedes Partikel ist hier vollständig von den Partikeln der nächst kleineren Korngrößenklasse eingehüllt. Auf diese Weise entsteht ebenfalls eine Packung mit hoher Raumauffüllung, der Feinstoffanteil ist aber sehr viel höher als bei einer apollonischen Packung. Mit dieser Korngrößenverteilung lässt sich auch in der

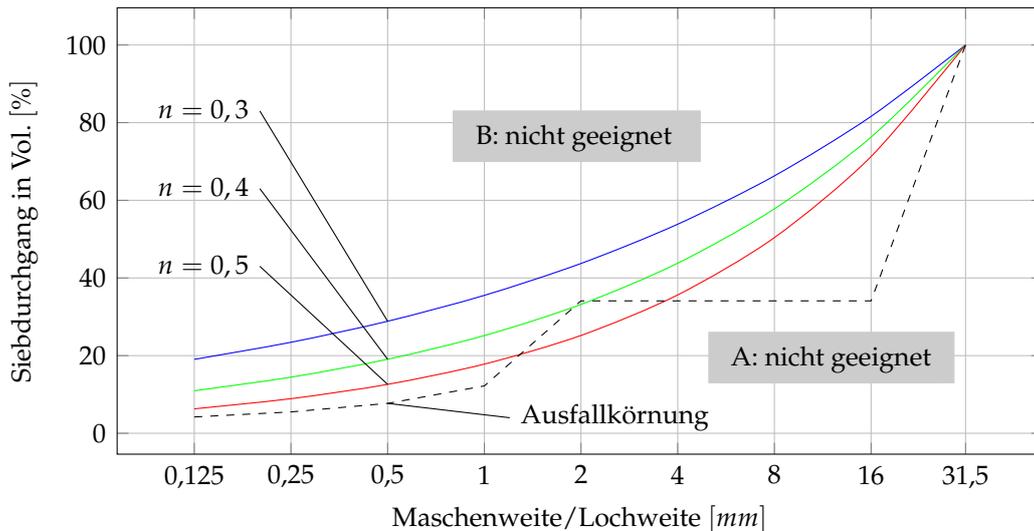


Abbildung 2.9: Fullerverteilung (Abbildung nach [VDZ08, Wol11])

Praxis eine hohe Raumausfüllung erreichen. Die große Anzahl feiner Partikel reicht auch dann noch aus die verbleibenden Hohlräume zwischen den groben Partikeln zu füllen, wenn diese nicht in der dichtest möglichen Anordnung vorliegen. Allerdings vergrößert der hohe Feinstoffanteil die Oberfläche der Packung sehr stark, was auch zu einem erhöhten Wasseranspruch der Betonmischung führt.

Sowohl die Korngrößenverteilung einer apollonischen Packung als auch die einer gesperrten Packung sind ungünstig für eine Betonmischung. Sie bilden aber - bezüglich des Feinstoffanteils - eine untere und obere Grenze, zwischen denen sich günstige Korngrößenverteilungen finden lassen. Als das klassische Modell für günstige Korngrößenverteilungen gilt die von Fuller und Thompson [FT06] entwickelte Fuller-Verteilung. Sie beschreibt den Durchgang des Partikelgemisches durch Siebe mit unterschiedlicher Maschenweite - also $Q_3(x)$ - mit:

$$Q_3(x) = \left(\frac{x}{x_{max}} \right)^n$$

Dabei ist x_{max} das Größtkorn. Einige für Betonzuschlag typische Sieblinien mit Exponenten $n = 0,3 \dots 0,5$ sind in Abbildung 2.9 dargestellt. Die Korngrößenverteilung der apollonischen Packung wäre hier im unteren Teil der Grafik zu finden (A: nicht geeignet), die Korngrößenverteilung der gesperrten Packung im oberen Teil der Grafik (B: nicht geeignet). Für welchen Exponenten man die optimale Raumausfüllung erhält, hängt auch von der Partikelform ab. So arbeitet man bei runden Partikeln mit $n \approx 0,4$, während man bei gebrochenen Materialien einen niedrigeren Wert benutzt: $n \approx 0,3$. Die Fuller-Verteilung ist in der DIN 1045-2 genormt, dort sind die Sieblinien

A($n = 0,7$), B($n = 0,35$) und C($n = 0,22$) aufgeführt.

Ebenfalls in der Abbildung 2.9 enthalten ist eine sogenannte Ausfallkörnung. Ausfallkörnungen haben eine unstetige Sieblinie, da einige Korngrößen absichtlich nicht besetzt sind. Solche Verteilungen können in einigen Fällen zu höheren Raumausfüllungen führen als die Fuller-Verteilungen. Das kann dadurch erklärt werden, dass während des Verdichtungsprozesses die feinsten Partikel nicht in alle Hohlräume zwischen den groben Partikeln gelangen können, wenn der Zugang von Partikeln mit mittlerer Größe blockiert wird. Entfernt man diese blockierenden Partikel aus der Mischung, kann die Packung viel stärker verdichtet werden. Ausfallkörnungen können zu sehr guten Rezepturen führen, diese müssen aber im Einzelfall überprüft werden.

2.6 Methoden zur Bestimmung der Raumausfüllung

Um die Raumausfüllung einer Partikelmischung zu bestimmen, existieren eine Reihe sehr unterschiedlicher Verfahren. Grundsätzlich lassen sie sich aber in 3 Kategorien einordnen. Bei den experimentellen Verfahren wird die Raumausfüllung der Mischung mit speziellen Messgeräten im Labor ermittelt. Mathematisch empirische bzw. semiempirische Verfahren ermitteln die Raumausfüllung auf Grund der gegebenen Korngrößenverteilung mittels eines mathematischen Modells. Die dritte Kategorie bilden die Simulationen, bei ihnen wird die Partikelpackung mit Hilfe eines Algorithmus als virtuelles 3D-Modell im Computer erzeugt. Die Raumausfüllung (und eventuell auch andere Eigenschaften) können dann aus diesem 3D-Modell ermittelt werden. Die folgenden Kapitel vermitteln einen kurzen Überblick über die Verfahren dieser 3 Kategorien.

2.6.1 Experimentelle Bestimmung der Raumausfüllung

Die Messverfahren zur experimentellen Bestimmung der Raumausfüllung lassen sich grob in nasse und trockene Methoden unterteilen. Bei den nassen Methoden [WK08a, WK08b] bestimmt man das verbleibende Hohlraumvolumen einer Partikelpackung, indem man dieses Volumen mit einer Flüssigkeit füllt. Die benötigte Flüssigkeitsmenge (also das Volumen) wird aus der Gewichtszunahme der Packung ermittelt. Bei den trockenen Methoden ermittelt man die Raumausfüllung aus der Reindichte ρ_R , also der Dichte ohne Hohlräume, und der komprimierten Schüttdichte ρ_{KS} des Materials zu

$$\Phi = \frac{\rho_{KS}}{\rho_R}.$$

Alle in der vorliegenden Arbeit verwendeten experimentell bestimmten Raumausfüllungen wurden mit Hilfe des Pyknometerverfahrens [Pal09] ermittelt. Es handelt sich dabei um eine trockene Methode, die zur Bestimmung der beiden benötigten Dichten ρ_{KS} und ρ_R zwei spezielle Pyknometer einsetzt. Für die Messung der komprimierten Schüttdichte ρ_{KS} wird ein Großpyknometer eingesetzt, bei dem eine zylindrische Kammer eine Probe des Materials aufnimmt. In diese Kammer wird dann ein Kolben eingefahren, während die Kammer selbst in Drehung versetzt wird. Aus der Eindringtiefe des Kolbens und der Masse der Probe lässt sich die gesuchte Dichte berechnen. Die Reindichte ρ_R wird mit Hilfe eines Heliumpyknometers gemessen. Dazu wird eine Probe des Materials in eine Kammer gegeben, die anschließend mit Helium gespült wird. In einer zweiten Referenzkammer wird ein Heliumüberdruck erzeugt. Stellt man nun einen Druckausgleich zwischen den beiden Kammern her, so füllt das Helium alle verfügbaren Hohlräume. Aus dem Druckabfall sowie den bekannten Volumen der Kammern lässt sich so das genaue Volumen der Probe (ohne Hohlräume) ermittelt, das zusammen mit der Masse der Probe die Reindichte ergibt.

2.6.2 Mathematisch empirische und semiempirische Bestimmung der Raumausfüllung

Um für eine Mischung aus mehreren granularen Stoffen, wie z.B. eine Betonrezeptur, eine Zusammensetzung zu finden, die eine optimale Raumausfüllung ergibt, ist eine sehr hohe Anzahl von Messungen notwendig. Die rein experimentelle Bestimmung der Raumausfüllung stößt hier schnell an ihre Grenzen. Ideal wäre deshalb ein rein mathematisches Modell, das aus einigen wenigen Kenngrößen einer Mischung, beispielsweise der Korngrößenverteilung, deren Raumausfüllung berechnen kann. Ein solches Modell ließe sich dann als Zielfunktion in einer Optimierungsaufgabe einsetzen. Die zu untersuchenden Mischungen müssen aber als zufällig dichte Packungen aufgefasst werden, da ihre Partikel an zufälligen Positionen und nicht in einem Kristallgitter angeordnet sind. Wie bereits in Kapitel 2.4 beschrieben wurden, sind die Eigenschaften solcher Packungen nicht rein mathematisch analytisch zugänglich. Allein aus der Kenntnis der in der Mischung auftretenden Korngrößen und deren Häufigkeit, kann also nicht auf die Raumausfüllung der Packung geschlossen werden. Alle existierenden mathematischen Modelle benutzen deshalb, zusätzlich zur Korngrößenverteilung, noch weitere Kenngrößen der zu untersuchenden Mischung, die auf experimentellem Wege ermittelt werden müssen. Je nach dem, ob diese Kenngrößen als allgemeingültig für alle Mischungen angenommen werden können oder ob sie individuell für verschiedene Arten von Partikelmischungen bestimmt werden müssen,

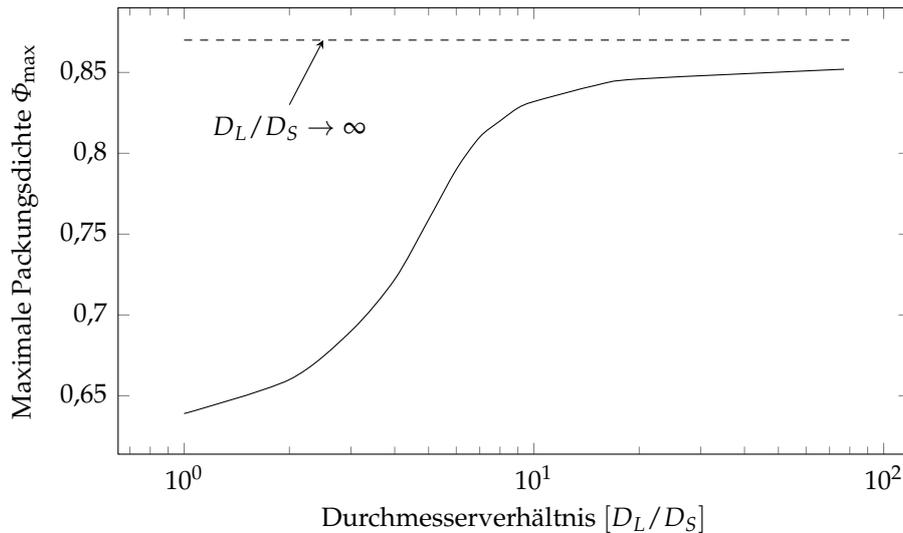


Abbildung 2.10: Maximale Packungsdichte von binären Mischungen nach [Lee70]

können die darauf aufbauenden Modelle in semiempirische bzw. empirische Modelle unterteilt werden.

Um die Grundidee hinter dieser Methode zur Bestimmung der Raumaufüllung verständlich zu machen, verwenden wir das Modell von Lee [Lee70], das als semiempirisch bezeichnet werden kann. Lee stützt sich auf eine Reihe von experimentellen Ergebnissen zur Raumaufüllung von monodispersen Kugelpackung [BH46, Sco60, Rut62]. Diese ergeben den bekannten Wert von $\approx 0,639$. Geht man davon aus, dass die gesamte Mischung nur aus kugelähnlichen Partikeln besteht und ihre Korngrößenverteilung in sehr eng verteilte (bzw. monodisperse) Fraktionen zerlegt ist, dann darf dieser Wert als konstante Eigenpackungsdichte für alle Fraktionen angenommen werden. Erzeugt man nun eine binäre Mischung, indem man in eine bestehende Fraktion aus kleinen Kugeln eine einzelne große Kugel einfügt, so ersetzt diese Kugel eine gewisse Menge kleiner Kugeln. Mit diesen kleinen Kugeln entfallen aber auch die zwischen ihnen liegenden Hohlräume, wodurch die Raumaufüllung der Mischung ansteigt. Durch das Einfügen immer weiterer großer Kugeln kann die Raumaufüllung weiter gesteigert werden, bis die großen Kugeln schließlich selbst eine zufällig dichte Packung bilden. Umgekehrt ist es auch möglich einer Fraktion aus großen Kugeln kleine Kugeln hinzuzufügen, die die verbleibenden Hohlräume auffüllen und so ebenfalls die Raumaufüllung erhöhen. Auch hier wird eine Grenze erreicht, wenn die kleinen Kugeln selbst eine zufällig dichte Packung in den Hohlräumen bilden.

Lee betrachtet die Raumaufüllung von binären Kugelpackungen als Funktion aus dem Durchmesser Verhältnis D_L/D_S der beiden beteiligten Kugelgrößen und den Volumenanteilen X_L und X_S der beiden Mischungskomponenten. Dabei bezeichnen D_L

bzw. X_L die Werte für die großen und D_S bzw. X_S die Werte für die kleinen Kugeln. Die maximal mögliche Packungsdichte in Abhängigkeit vom Durchmesser Verhältnis wurde experimentell ermittelt und ist in Abbildung 2.10 dargestellt. Die Funktion verläuft zwischen 2 Grenzwerten. Der eine ist die Raumausfüllung einer zufällig dichten monodispersen Packung (0,639) und stellt sich ein, wenn das Durchmesser Verhältnis 1 ist. Der andere Grenzwert ergibt sich, wenn das Durchmesser Verhältnis so groß ist ($\rightarrow \infty$), dass Effekte wie die Keilung und der Wandeffekt vernachlässigt werden können und sowohl die großen als auch die kleinen Kugeln in den Hohlräumen jeweils ihre eigenen zufällig dichten Packungen bilden, ohne sich gegenseitig zu beeinflussen. Dieser Grenzwert ist $\Phi_{P,\max} = 0,87$ und berechnet sich zu:

$$\Phi_{P,\max} = \Phi_{P,L} + (1 - \Phi_{P,L}) \cdot \Phi_{P,S}.$$

Dabei sind $\Phi_{P,L}$ und $\Phi_{P,S}$ jeweils die Eigenpackungsdichten der großen bzw. kleinen Kugeln, die wir beide mit 0,639 annehmen. Für $\Phi_{P,\max}$ berechnen sich die zugehörigen Volumenanteile der beiden Mischungskomponenten wie folgt:

$$X_L = \frac{\Phi_{P,L}}{\Phi_{P,\max}} = 0,735$$

$$X_S = \frac{(1 - \Phi_{P,L}) \cdot \Phi_{P,S}}{\Phi_{P,\max}} = 0,265$$

Für beliebige Volumenanteile kann die resultierende Packungsdichte gebildet werden indem man die beiden Werte:

$$(\Phi_P)_S = 0,639 \cdot X_S + \Phi_{SL} \cdot X_L$$

$$(\Phi_P)_L = \Phi_{LS} \cdot X_S + 0,639 \cdot X_L$$

berechnet. Die beiden Koeffizienten Φ_{SL} und Φ_{LS} ergeben sich wie folgt:

$$\Phi_{SL} = 0,639 + (\Phi_{\max}(D_L/D_S) - 0,639)/0,735$$

$$\Phi_{LS} = 0,639 + (\Phi_{\max}(D_L/D_S) - 0,639)/0,265$$

Dabei ist Φ_{\max} die maximale Packungsdichte für binäre Mischungen mit dem entsprechenden Durchmesser Verhältnis aus der Abbildung 2.10. Die beiden Werte $(\Phi_P)_S$ und $(\Phi_P)_L$ spiegeln die Vorstellung wieder, dass man entweder in eine Fraktion aus kleinen Kugeln große Kugeln einfügt oder in einer Fraktion aus großen Kugeln die verbleibenden Hohlräume mit kleinen Kugeln auffüllt. Oder in einer anderen Betrachtungsweise: die Mischung wird entweder von den kleinen Kugeln dominiert, die ein festes Packungsskelett bilden, in das an einigen Stellen große Kugeln eingelassen sind; oder

die Mischung wird von den großen Kugeln dominiert, die ein festes Packungsskelett bilden, dessen Hohlräume teilweise von kleinen Kugeln aufgefüllt werden. Der kleinere dieser beiden Werte $(\Phi_P)_S$ bzw. $(\Phi_P)_L$ ist dann die gesuchte Raumaufüllung der Mischung.

Ausgehend von diesem Grundmodell für binäre Mischungen erweitert Lee [Lee70] seine Methode durch das Hinzufügen weiterer Komponenten zu einem Modell für n -äre Mischungen. Dadurch wird es möglich, beliebige Korngrößenverteilungen zu verarbeiten, in dem sie in viele einzelne eng verteilte Fraktionen zerlegt werden. Wenn n die Anzahl dieser Fraktionen ist, dann sind die n Werte $(\Phi_P)_i$ mit:

$$(\Phi_P)_i = \sum_{j=1}^n \Phi_{ij} \cdot X_j \quad \text{mit } i = 1, \dots, n$$

zu berechnen. Wie schon beim Grundmodell für binäre Mischungen ist der kleinste dieser Werte die gesuchte Raumaufüllung. Durch die Koeffizienten Φ_{ij} :

$$\begin{aligned} \Phi_{ii} &= 0,639 \\ \Phi_{ij} &= 0,639 + (\Phi_{\max}(D_i/D_j) - 0,639)/0,265 \\ \Phi_{ji} &= 0,639 + (\Phi_{\max}(D_i/D_j) - 0,639)/0,735 \\ \text{es gilt: } D_i/D_j &\geq 1 \end{aligned}$$

werden alle möglichen Kombinationen der Komponenten berücksichtigt. Bei sehr breiten polydispersen Korngrößenverteilungen, wie sie bei Betonmischungen üblich sind, werden schnell mehrere hundert Fraktionen benötigt, um die notwendige enge bzw. (fast) monodisperse Verteilung in den einzelnen Fraktionen zu erreichen. Die Berechnungen müssen also auf jeden Fall am Computer durchgeführt werden.

Lee stützt sich bei seinem Modell nur auf die Daten der Packungsdichte von monodispersen Kugelpackungen sowie die maximalen Packungsdichte von daraus erzeugten binären Mischungen (Abbildung 2.10). Da diese Daten als gegeben und konstant angenommen werden und keine weiteren individuellen Messungen für konkrete Mischungen aus unterschiedlichen Stoffen mehr vorgenommen werden müssen, wird dieses Modell als semiempirisch bezeichnet. Seine Einfachheit macht es interessant für den Einsatz in Projekten, in denen es um die Bestimmung der maximalen Packungsdichte von granularen Mischungen geht. Beispielsweise wird es in [Übl02] eingesetzt, um die optimale Packungsdichte von Füllmaterialien zu bestimmen, die in das Material von elektrischen Isolationen eingebracht werden, um deren Wärmeleitfähigkeit zu erhöhen. Die Einfachheit des Modells bildet aber auch einen Schwachpunkt. Bei Mischungen mit sehr breiter Korngrößenverteilung können die Wechselwirkungen zwi-

schen den unterschiedlich großen Partikeln in der Packung sehr komplex sein. Die realen Zustände lassen sich dann nur teilweise auf das Verhalten von binären Mischungen zurückführen. Abhängig von der konkreten Korngrößenverteilung können so unterschiedlich starke Abweichungen von der realen Raumauffüllung auftreten.

Neben dem Modell von Lee existieren auch eine Reihe empirischer Modelle, wie beispielsweise die Arbeiten von Furnas [Fur29] und Yu und Standish [YS88, YS91]. Diese Arbeiten basieren ebenfalls auf dem Grundgedanken eine Partikelmischung mit umfangreicher Korngrößenverteilung auf die Kombination von jeweils 2 mehr oder weniger eng verteilten Partikelmischungen zurückzuführen. Sie unterscheiden sich aber darin, wie diese Kombination modelliert wird. Zusätzlich benötigen sie, im Gegensatz zu Lee, genauere und eventuell für die konkrete Stoffmischung zu bestimmende empirische Daten, weshalb sie nicht als semiempirisch bezeichnet werden können. Eine detaillierte Betrachtung dieser und weiterer Modelle findet sich bei Raschdorf [Ras10]. Allgemein lässt sich sagen, dass besonders solche Modelle in ihren Ergebnissen Realitätsnähe und Zuverlässigkeit beweisen, die auf detailliertere empirische Daten zurückgreifen. Diese Modelle benötigen damit aber auch einen höheren zusätzlichen Aufwand für die experimentelle Bestimmung der benötigten Daten. Mitunter gelten diese Daten auch nur unter bestimmten Voraussetzungen, was diese Modelle dann weniger allgemeingültig macht.

2.6.3 Bestimmung der Raumauffüllung per Simulation

Eine andere Möglichkeit, die Raumauffüllung einer Partikelmischung per Berechnung am Computer zu bestimmen, bilden die Simulationen. Sie benötigen keine zusätzlichen empirischen Daten, da sie die zu untersuchende Packung virtuell nachbilden. Der Rechenaufwand für diese Simulationen ist in jedem Fall um ein Vielfaches höher als bei den mathematisch empirischen Methoden, wodurch die Einsatzmöglichkeiten ursprünglich beschränkt waren. Die stetig steigende Rechenleistung der Computer einerseits und neue verbesserte Algorithmen andererseits machen Simulationen aber mehr und mehr zu einem wertvollen Werkzeug für die Untersuchung der Eigenschaften von Partikelmischungen. Da die Packung virtuell nachgebildet wird, ist es nicht nur möglich, ihre Raumauffüllung zu bestimmen. Auch andere, sonst nur schwer zu ermittelnde Daten, sind auf diese Weise zugänglich, wie beispielsweise die Koordinationszahl der Partikel, die Porenstruktur oder die Auswirkung des Wandeffektes.

Es existieren mehrere unterschiedliche Simulationsmodelle, die für diesen Einsatz in Frage kommen. Grob kann man sie entweder den physikalischen Modellen, die in der Regel DEM-basiert [CS79] sind, oder den abstrakten Modellen zuordnen. Die physikalischen Modelle simulieren die Bewegungen der Partikel auf der Grundlage

der physikalischen Gesetze. Sie sind dadurch sehr vielseitig, ihre Möglichkeiten gehen weit über den Aufbau einer statischen Partikelpackung hinaus. Auch das Fließverhalten der Partikel unter verschiedenen Bedingungen kann simuliert werden. Solche Simulationen benötigen aber auch einen entsprechend hohen Rechenaufwand. Die abstrakten Modelle beschränken sich nur auf den Aufbau der Partikelpackung, die physikalischen Gesetze, die diesem Aufbau zugrunde liegen, werden nur teilweise und nur in abstrakter Form berücksichtigt. Zwei wesentliche Vertreter dieser abstrakten Modelle sind die Random-Sequential-Addition Simulationen (RSA) [TCW68, AM72], die die Packung durch iteratives Ablagern der einzelnen Partikel in einem Container erzeugen, und die Collective-Rearrangement Simulationen (CR) [HE98, HEC99, BBS02, BS06], die alle Partikel gleichzeitig in einem Container platzieren und durch iteratives Verschieben der einzelnen Partikel deren endgültige Position ermitteln. In [Els09] findet sich eine Arbeit, die sich detailliert mit dem simulativen Erzeugen und Auswerten von Kugelpackungen beschäftigt. Das in der vorliegenden Arbeit verwendete Simulationssystem für Partikelpackungen basiert auf einem Collective-Rearrangement, es wurde von Raschdorf entwickelt und ist detailliert in [Ras06] beschrieben. Eine kurze Beschreibung des Ablaufs sowie der wichtigsten Funktionen findet sich auch in Kapitel 3.2.

Kapitel 3

Die Packungssimulation „RaSim“

Der Schwerpunkt meiner Arbeit liegt in der Weiterentwicklung des Simulationssystems „RaSim“, das ursprünglich von Steffen Raschdorf im Rahmen seiner Promotion entwickelt wurde. Damit ist es möglich, aus vorgegebenen Korngrößenverteilungen zufällig dichte Kugelpackungen zu erzeugen, zu analysieren und zu visualisieren. Der Aufbau und die Funktion dieser Software werden in diesem Kapitel erklärt, dabei wird auch auf die besonderen Probleme eingegangen, die sich aus den Korngrößenverteilungen ergeben, die typischerweise bei Betonmischungen auftreten.

Der grundsätzliche Aufbau der Software ist in Abbildung 3.1 dargestellt. Die drei wichtigsten Komponenten sind: die Simulation, die Fraktionierung und die Nachbarschaftsverwaltung. In der Simulation wird eine Stichprobe aus Kugeln innerhalb eines Containers zu einer zufällig dichten Packung arrangiert. Um auch hoch polydisperse Partikelmischungen simulieren zu können, wurde die Fraktionierung entwickelt. Sie zerlegt die vorgegebene Korngrößenverteilung in mehrere kleinere Abschnitte, zu denen sich repräsentative Stichproben von moderatem Umfang erstellen lassen. Die Nachbarschaftsverwaltung beinhaltet eine Datenstruktur, die einen effizienten Zugriff auf die räumliche Nachbarschaft jeder Kugel bietet. Dieser Zugriff ist entscheidend für ein gutes Laufzeitverhalten der Simulation. Damit bildet die Nachbarschaftsverwaltung, genau wie die Fraktionierung, eine wichtige Voraussetzung, um hoch polydisperse Korngrößenverteilungen verarbeiten zu können. Die Arbeitsweise und die zugrunde liegenden Modelle dieser drei Komponenten werden in den folgenden Kapiteln erklärt.

Alle notwendigen Eingabedaten werden in Form von zwei Dateien bereitgestellt. Die Korngrößenverteilung ist in der Mischungsdatei enthalten. Diese Daten liegen, bedingt durch die verwendeten Messverfahren, meistens als Volumenverteilung vor. Da die Simulation aber eine Anzahlverteilung für das Sampling benötigt, werden die notwendigen Konvertierungen von einer eigenen Komponente - der Mischung - über-

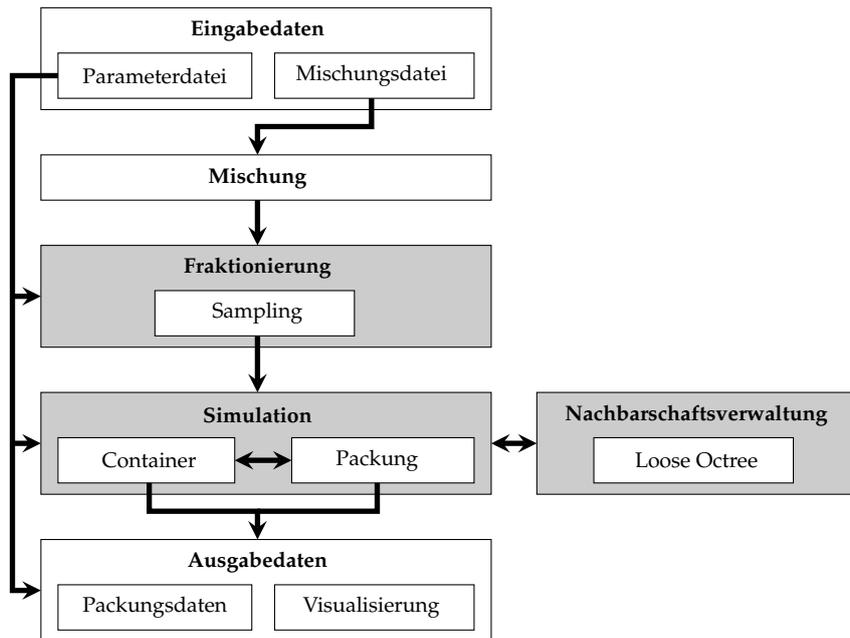


Abbildung 3.1: Grundsätzlicher Aufbau des Simulationssystems „RaSim“

nommen. In der Parameterdatei können verschiedene Einstellungen vorgenommen werden, mit denen sich bestimmte Aspekte der Simulation steuern lassen. Die Ergebnisse einer abgeschlossenen Simulation können auf mehrere Arten dargestellt werden. Die ermittelte Raumauffüllung wird zusammen mit einer Übersicht über die eingestellten Parameter, den berechneten Zwischenergebnissen und der benötigten Zeit in einer Protokolldatei ausgegeben. Zusätzlich können die erzeugte Packungsstruktur und verschiedene Visualisierungen dieser Packung jeweils als Datei ausgegeben werden.

3.1 Die Fraktionierung

Eine Grundvoraussetzung für die simulative Erzeugung einer zufällig dichten Partikelpackung und die Ermittlung ihrer Eigenschaften ist das Sampling. Dabei wird aus den vorgegebenen Daten eine virtuelle Kugelmengung erzeugt, die die Eigenschaften der realen Partikelmischung in ausreichendem Maße abbildet. Die Raumauffüllung der Packung kann von vielen Partikeleigenschaften beeinflusst werden. Im Allgemeinen beschränkt man sich aber auf die Abbildung der realen Korngrößenverteilung und lässt andere Faktoren, wie z.B. die Kornform, zugunsten einer schnellen Simulation, außer Acht. Das Sampling muss also nur die passenden Durchmesser für die einzelnen Elemente der Kugelmengung ermitteln. Trotz dieser Vereinfachung ist das Vorgehen

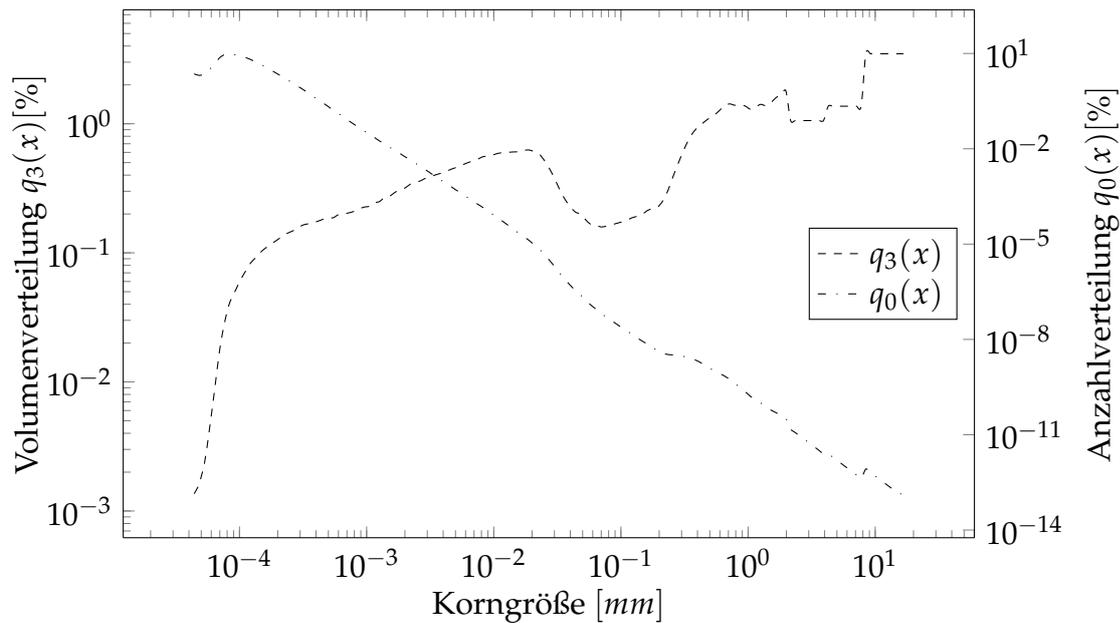


Abbildung 3.2: Volumen- und Anzahlverteilungsdichte einer Betonmischung aus 17% Zement (CEM II), 5% Flugasche und einer Gesteinskörnung bis 15mm Korngröße.

nicht trivial. Die Besonderheiten der in Betonmischungen vorkommenden Korngrößenverteilungen stellen ein erhebliches Problem bei der Erzeugung einer repräsentativen Stichprobe dar. Einen Eindruck davon vermittelt die Abbildung 3.2, in der die labortechnisch gemessene Korngrößenverteilung einer Betonmischung dargestellt ist.

Die Korngrößen in dieser Mischung reichen von weniger als $1\mu\text{m}$ bis zu mehr als 1cm . Betrachtet man die Unterschiede im Verlauf der Volumen- und der Anzahlverteilung so fällt auf, dass die größten Partikel am seltensten auftreten. Die Partikel von $0,1 - 15\text{mm}$ machen nur $1,25 \cdot 10^{-7}\%$ aller Partikel der Mischung aus. Sie können jedoch nicht vernachlässigt werden, denn sie machen aufgrund ihrer Größe c.a. 77% des Gesamtvolumens aller Partikel aus. Genau umgekehrt verhält es sich mit den kleinsten Korngrößen: 95% aller Partikel sind kleiner als $0,25\mu\text{m}$, sie machen aber nur c.a. 1,3% des Volumens aus. Die Raumauffüllung einer zu dieser Korngrößenverteilung passenden Kugelpackung ergibt sich aus der Interaktion aller Korngrößen:

- **Die größten Korngrößen** nehmen den größten Teil des Volumens ein.
- **Die mittleren Korngrößen** füllen die Hohlräume zwischen den größten Partikeln oder (wenn nicht genügend Hohlraum zur Verfügung steht) sie drängen deren Packungsskelett auseinander.
- **Die kleinsten Korngrößen** machen nur einen kleinen Teil des Volumens aus aber durch ihre geringe Größe und ihre große Anzahl können sie die verbleibenden Hohlräume füllen und so die Raumauffüllung der Packung steigern.

Keine der Korngrößen kann vernachlässigt werden. Für die simulative Ermittlung der Raumausfüllung benötigt man also eine repräsentative Stichprobe über alle Durchmesser der Korngrößenverteilung. Das heißt, die Stichprobe muss Kugeln für alle vorgegebenen Korngrößenintervalle im richtigen Verhältnis enthalten.

3.1.1 Arten des Sampling

Wenn sich zu einer gegebenen Verteilungsfunktion F die zugehörige Inverse F^{-1} ohne großen Aufwand bestimmen lässt, dann bietet die Inversionsmethode [Kol08] eine gute Möglichkeit, eine Stichprobe zu erzeugen, die der Verteilung folgt.

Satz 3.1 (Inversionsprinzip) *Es sei F eine Verteilungsfunktion und U eine $U(0,1)$ -verteilte Zufallsvariable. Dann gilt: $Y := F^{-1}(U)$ hat die Verteilungsfunktion F , d.h.*

$$\mathbf{P}(Y \leq t) = F(t), \quad t \in \mathbb{R}.$$

Die Verteilungsfunktion ist hier die Verteilungssummenfunktion der Anzahlanteile $Q_0(x)$ wobei x die Korngröße ist. Die Stichprobe wird also unabhängig gleichverteilt auf der y -Achse gezogen. Mithilfe der Inversen der Verteilungssummenfunktion $Q_0^{-1}(y)$ können dann die Korngrößen für die Stichprobe bestimmt werden. Das Inversionsprinzip garantiert, dass die gezogenen Korngrößen der gegebenen Verteilung folgen.

Aufgrund der besonderen Gestalt der hier verwendeten Verteilungen, bei denen die größten Partikel extrem selten vorkommen, kann die Inversionsmethode nicht direkt angewendet werden. Bei einer $U(0,1)$ -verteilte Zufallsvariable wäre es extrem unwahrscheinlich, wenigstens einen Partikel aus dem größten Korngrößenintervall zu ziehen. Für eine repräsentative Stichprobe müssen deshalb andere Methoden verwendet werden, um eine gleichverteilte Variable auf der y -Achse zu erzeugen. Man kann grundsätzlich folgende Möglichkeiten unterscheiden:

Simple Random Sampling With Replacement Bei dieser Methode werden alle Punkte der Stichprobe unabhängig und zufällig gleichverteilt auf dem zur Verfügung stehenden Intervall gezogen. Es wird also eine $U(0,1)$ -verteilte Zufallsvariable simuliert, so dass dieses Vorgehen genau dem Inversionsprinzip entspricht. Das Sampling muss aber gegebenenfalls so lange wiederholt werden, bis auch die seltenen Korngrößen in der Stichprobe vorhanden sind.

Stratifiziertes Sampling Dabei wird das zur Verfügung stehende Intervall in mehrere Schichten (Strata) unterteilt, um anschließend in jeder Schicht eine separate

Stichprobe zu ziehen. Diese Methode reduziert die Varianz und garantiert auch das Vorkommen der seltenen Korngrößen in der Gesamtstichprobe.

Systematic Sampling [MM44, Yat48] Das Systematic Sampling bildet einen Spezialfall des Stratifizierten Sampling. Bei einem Stichprobenumfang von N Partikeln wird das zur Verfügung stehende Intervall in N Strata unterteilt. Innerhalb des ersten Stratum wird ein Startpunkt p_1 gewählt, der damit alle weiteren Punkte der Stichprobe p_i gemäß:

$$p_i = p_1 + \frac{i-1}{N}, \quad i = 2, \dots, N$$

festlegt.

Deterministisches Sampling Bei diesem Spezialfall des Systematic Sampling wird darauf verzichtet, den Startpunkt p_1 zufällig zu bestimmen, stattdessen wird ein prä-determiniertes Quantil benutzt. Somit kann beim deterministischen Sampling auf einen Zufallsgenerator verzichtet werden.

Eine detaillierte Beschreibung aller Sampling-Methoden und ihrer Spezialfälle sowie eine Untersuchung ihrer Vor- und Nachteile findet sich bei Raschdorf [Ras10]. Testsimulationen mit verschiedenen Korngrößenverteilungen haben gezeigt, dass die Varianzen in den Stichproben, die durch das stratifizierte- oder das systematic Sampling entstehen, keinen nennenswerten Einfluss auf die simulierte Raumauffüllung haben. Im praktischen Einsatz arbeitet das Simulationsprogramm deshalb in der Regel mit einem deterministischen Sampling.

3.1.2 Hierarchische Aufteilung der Samples

Das Hauptproblem bei der Erzeugung eines repräsentativen Samples, das alle Korngrößen im korrekten Verhältnis abbildet, liegt im benötigten Stichprobenumfang. Die in Abbildung 3.2 dargestellte Verteilung verdeutlicht dies. Die größten Partikel dieser Mischung (15mm) sind auch die seltensten, ihr Anteil an der Gesamtanzahl aller Partikel beträgt nur $1,39 \cdot 10^{-15}$ ($1,39 \cdot 10^{-13}\%$). Der Kehrwert davon ist $7,2 \cdot 10^{14}$, das ist der Stichprobenumfang der notwendig ist, um im Mittel wenigstens eine Kugel im größten Durchmesserintervall zu erhalten. Die Simulation wäre dann mit nur einer - die gesamte Korngrößenverteilung umfassenden - Stichprobe mit heutiger Standardhardware nicht durchführbar.

Zur Lösung des Problems wurde der Algorithmus RESOS (recursive estimation with systematic overlapping sampling) [KRW10] entwickelt, der die Simulation hierarchisch mit mehreren Stichproben durchführt. Dieser Algorithmus basiert auf der Idee,

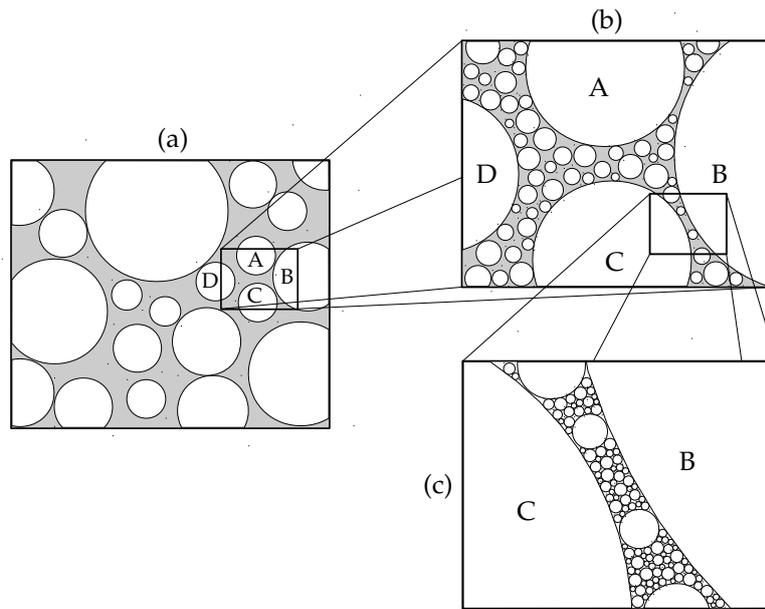


Abbildung 3.3: Prinzip der Fraktionierung nach [Ras10]: Die Gesamtpackung wird in mehrere Detailstufen (Fraktionen) zerlegt. Einzelne Randkugeln aus höheren Fraktionen sorgen für die Interaktionseffekte zwischen den Fraktionen.

die Gesamtpackung in mehrere Detailstufen zu zerlegen und diese dann unabhängig voneinander zu simulieren. In Abbildung 3.3 ist dieses Prinzip dargestellt. Zunächst werden - in Teil (a) der Abbildung - nur die größten Kugeln der Packung betrachtet, dabei werden alle kleineren Kugeln als graue Masse dargestellt. Will man die Gesamtraumausfüllung der Packung berechnen, muss man die Raumausfüllung dieser Masse kennen. Dazu ist eine weitere Simulation in einer niedrigeren Detailstufe notwendig, wie in Teil (b) dargestellt. Da diese Stichprobe die größten Kugeln nicht mehr notwendiger Weise im korrekten Verhältnis enthalten muss, genügt ein sehr viel kleinerer Ausschnitt der Gesamtpackung. Auch dieser Ausschnitt enthält wieder eine graue Masse, die hier die kleinsten Kugeln der Korngrößenverteilung repräsentiert. Da in keinem der drei Teile alle Korngrößen im korrekten Verhältnis abgebildet werden müssen, sondern nur die eines kleinen Abschnittes, kann der Stichprobenumfang in einer für die Simulation sinnvollen Größe gehalten werden.

Somit ergibt sich für die Berechnung der Gesamtraumausfüllung der Packung ein rekursiver Algorithmus, der die Gesamtpackung hierarchisch, beginnend mit der niedrigsten Detailstufe, aufbaut. Die Detailstufen entsprechen einer Fraktionierung der Korngrößenverteilung, wobei aber jedes Sample zusätzlich auch Korngrößen aus höheren Fraktionen als Randkugeln enthalten muss. Die Abbildung zeigt, dass die Kugeln A, B, C und D aus Teil (a) auch in Teil (b) vorhanden sind, die Kugeln B und C

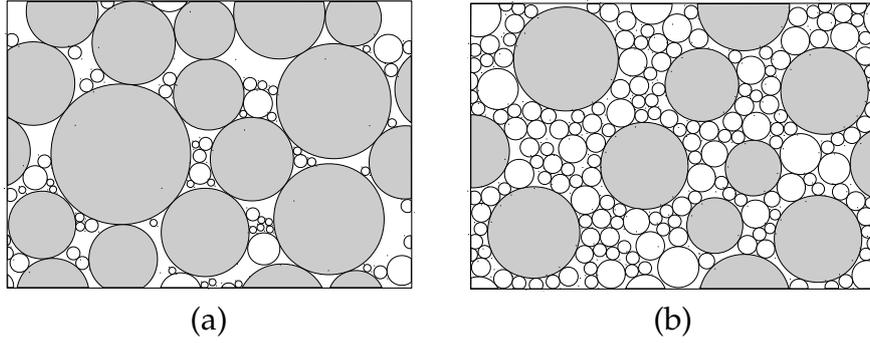


Abbildung 3.4: Kombination der Fraktionen zur Gesamtpackung nach [Ras10]: Die Packung wird entweder von den Kugeln der aktuell höchsten Fraktion F_i (a) oder von den Kugeln der unteren Fraktionen $F_1 \dots F_{i-1}$ (b) dominiert.

sind außerdem, zusammen mit einigen Kugeln aus Teil (b), auch in Teil (c) vorhanden. Durch dieses Vorgehen können die Interaktionen zwischen den Korngrößen der unterschiedlichen Fraktionen simuliert werden. Das betrifft vor allem den Wandeffekt, ohne den die ermittelte Raumaufüllung zu hoch ausfallen würde.

Bezeichnen wir die Fraktionen, nach aufsteigender Partikelgröße, mit F_1, \dots, F_n , wobei eine Fraktion F_i eine Mischung aus Partikeln des Durchmesserintervalls $[d_{\min}^{(i)}, d_{\max}^{(i)}]$ repräsentiert, dann berechnet sich die Gesamtraumaufüllung aus:

$$\phi = \frac{V(F_1, \dots, F_n)}{C(F_1, \dots, F_n)} = \frac{\sum_{i=1}^n V(F_i)}{C(F_1, \dots, F_n)} \quad (3.1)$$

Das von der Gesamtpackung benötigte Volumen $C(F_1, \dots, F_n)$ ergibt sich durch die Art und Weise, wie sich die Kugeln der unteren Fraktionen in die jeweils nächst höhere Fraktion einfügen. Dafür gibt es zwei Möglichkeiten, die in Abbildung 3.4 dargestellt werden: Im Fall (a) reicht das Volumen der unteren Fraktionen nicht aus, um das Hohlraumvolumen $L(F_i)$ der Fraktion F_i zu füllen. Die Packung der Kugeln von F_i bestimmt dann alleine das benötigte Volumen. Im Teil (b) werden die Kugeln von F_i auseinander gedrängt, sie „schwimmen“ zwischen den Kugeln der unteren Fraktionen. Da sich die Kugeln von F_i nicht gegenseitig berühren, können sie keine eigene Packung bilden wie im Fall (a), ihre Position innerhalb der Gesamtpackung spielt daher keine Rolle, sie vergrößern deren benötigtes Volumen nur um den Betrag ihres eigenen Nettovolumens. Damit ergibt sich:

$$C(F_1, \dots, F_i) = \begin{cases} C(F_i) & \text{für } C(F_1, \dots, F_{i-1}) \leq L(F_i) \\ C(F_1, \dots, F_{i-1}) + V(F_i) & \text{für } C(F_1, \dots, F_{i-1}) > L(F_i) \end{cases} \quad (3.2)$$

Hier bietet sich eine Möglichkeit, den Simulationsvorgang abzukürzen. Wenn die Kugeln der Fraktion F_i keine eigene Packung bilden können, weil sie zwischen den Kugeln der unteren Fraktionen „schwimmen“ ($C(F_1, \dots, F_{i-1}) > L(F_i)$), wird das von ihrer Packung eingenommene Volumen $C(F_i)$ nicht für die Berechnung des Volumens der Gesamtpackung $C(F_1, \dots, F_i)$ benötigt. In diesem Fall könnte also auf die Simulation dieser Fraktion verzichtet werden. Dazu müsste jedoch schon vor Beginn der Simulation feststehen, dass der verbleibende Hohlraum in der Fraktion F_i nicht ausreichen wird, um die Kugeln der unteren Fraktionen aufzunehmen. Das ist nicht möglich, da die Raumauffüllung und damit auch das Hohlraumvolumen der Fraktion erst durch die Simulation bestimmt wird. Mit Hilfe empirischer Daten können wir jedoch eine Untergrenze für die Raumauffüllung und damit eine Obergrenze für den verbleibenden Hohlraum festsetzen. Ausgehend von den aus der Literatur bekannten Untersuchungen für zufällig dichte monodisperse Kugelpackungen [Sco60, BH46, Rut62, Lee70], können wir die Raumauffüllung für solche Packungen mit 0,639 festlegen. Der verbleibende Hohlraum beträgt also ca. 36,1% des von der Packung insgesamt eingenommenen Volumens. Da wir voraussetzen, dass alle Partikelgrößen durch Kugeln dargestellt werden und störende Effekte wie Reibung und Agglomeration vernachlässigbar sind, können wir für polydisperse Packungen eine höhere Raumauffüllung und damit einen kleineren Hohlraumanteil erwarten. Die 36,1% bilden also eine obere Grenze des für die unteren Fraktionen zur Verfügung stehenden Hohlraums. Reicht dieses Volumen nicht aus, kann auf die Simulation der Fraktion F_i verzichtet werden.

Da die Sample zu jeder Fraktion (mit Ausnahme der letzten Fraktion) auch Kugeln aus höheren Fraktionen enthalten, muss das Volumen der Packung $C(F_i)$ aus Formel 3.2 angepasst werden.

$$C^*(F_i) := C(F_i, \dots, F_n) - V(F_{i+1}, \dots, F_n) \quad (3.3)$$

Mit $C^*(F_i)$ wird das Volumen einer Packung aus Kugeln der Fraktion F_i bezeichnet, die zusätzlich Kugeln aus höheren Fraktionen F_{i+1}, \dots, F_n enthält. Diese zusätzlichen Kugeln werden wie Löcher innerhalb der Packung behandelt, durch sie wird der Wandeffekt der großen Kugeln aus F_{i+1}, \dots, F_n auf die kleinen Kugeln aus F_i vermittelt, ihr Volumen zählt jedoch nicht zum Volumen der Packung. Wie durch Simulationen mit verschiedenen Glaskugelpackungen [KRW10] gezeigt werden konnte, liegt die mit RESOS ermittelte Raumauffüllung, dank der Randkugeln, sehr dicht bei den empirisch ermittelten Werten. Ohne die Randkugeln wird jedoch eine teilweise mehr als 10% zu hohe Raumauffüllung erreicht.

3.1.3 Werte und Parameter für die Fraktionierung

Um die Anzahl und Lage der Fraktionen festzulegen ist es naheliegend, die x-Achse in mehrere gleich breite Abschnitte zu unterteilen und anschließend zu jedem dieser Abschnitte ein eigenes Sample zu erzeugen. Dieses Vorgehen ist zwar durchaus möglich, es geht jedoch nicht auf die Besonderheiten der Korngrößenverteilung ein. In den hier verwendeten Mischungen liefern die größten Partikel auch den größten Volumenanteil (Abbildung 3.2), deshalb sind sie besonders wichtig für die Raumauffüllung der gesamten Packung. Ihre Anzahl ist jedoch sehr viel geringer als die der kleinen Partikel. Hinzu kommt, dass - aufgrund der verwendeten Messverfahren - die Korngrößen für die größeren Partikel in deutlich breiteren Intervallen gegeben sind als für die kleinen Partikel. Aufgrund dieser Tatsachen wird die Einteilung so vorgenommen, dass die Fraktionen mit zunehmender Korngröße schmaler und damit genauer werden.

Zunächst muss die Korngrößenverteilung als Verteilungssumme für die Anzahlanteile $Q_0(x)$ vorliegen. Dann kann, unter Benutzung eines der in Kapitel 3.1.1 vorgestellten Verfahren, ein Sample über die gesamte Korngrößenverteilung erzeugt werden, also vom Anteil 0 bis 1 (100%). Dieses Sample wird dann gemäß eines fest vorgegebenen Anteils a in aktive Kugeln und Randkugeln unterteilt. Da aktive Kugeln und Randkugeln zusammen als ein Sample erzeugt werden, ist das korrekte Verhältnis zwischen den Korngrößen gewährleistet. Der Korngrößenumfang der aktiven Kugeln definiert die Lage der Fraktion. Damit ist der Startanteil p_i , von dem aus das Sample für die Fraktion F_i erzeugt wird, genau der Endanteil der aktiven Kugeln von F_{i-1} .

Diese Art der Fraktionierung wird in Abbildung 3.5 veranschaulicht. Die verwendete Verteilungssumme gehört zur selben Betonmischung, die in Abbildung 3.2 dargestellt wurde. Jedes der erzeugten Sample besteht aus 50.000 Kugeln, eine Größenordnung, die sich mit vertretbarem Zeitaufwand simulieren lässt. Dabei stellt jedes Sample die Korngrößen ihrer Fraktion im korrekten Verhältnis dar, sowie zusätzlich die Korngrößen einiger höher liegender Fraktionen. Die Relationen der einzelnen Sample zueinander können jedoch nicht die korrekten Anteile ihrer Fraktionen an der Gesamtanzahl aller Partikel wiedergeben. Jedes Sample hat die gleiche Größe, und - bei einheitlich festgelegtem Anteil aktiver Kugeln a - ist auch die Anzahl der aktiven Kugeln in jedem Sample gleich. Damit hätte jede Fraktion den gleichen Anzahlanteil an der Gesamtpackung, was jedoch nicht der vorgegebenen Verteilung entspricht. Aus Abbildung 3.5 wird deutlich, dass die Fraktionen mit wachsendem Index einen immer geringeren Anteil an der Gesamtanzahl aller Partikel haben. Deshalb ist es notwendig zwischen der Fraktion F_i und dem zugehörigem Sample S_i zu unterscheiden.

Um aus der Simulation eines Samples S_i die in den Gleichungen 3.1 - 3.3 benötigten Werte $C(F_i)$, $L(F_i)$ und $V(F_i)$ zu erhalten, können die ermittelten Simulationsergebnis-

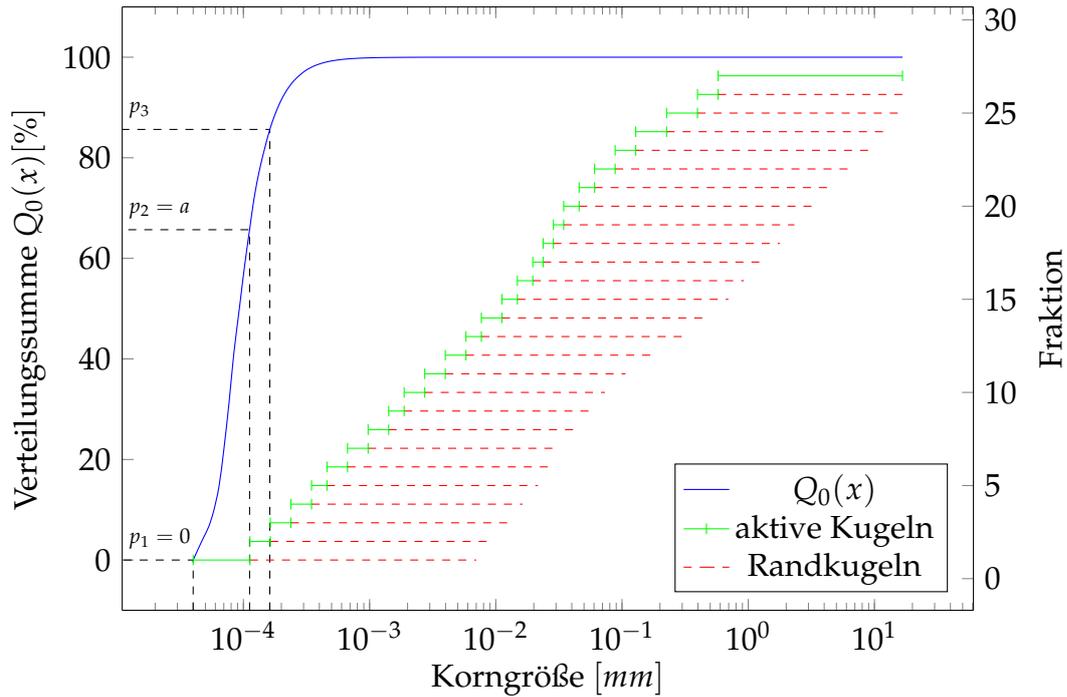


Abbildung 3.5: Fraktionierung einer Betonmischung anhand ihrer Verteilungssumme. Der Startanteil einer Fraktion F_i ist genau der Endanteil der aktiven Kugeln aus F_{i-1} .

se nicht direkt verwendet werden. Sie müssen zuerst von den Relationen des Samples auf die korrekten Relationen der Fraktion transformiert werden. Dazu werden sie entsprechend des Anzahlanteils des Samples S_i an der Gesamtverteilung gewichtet. Das Gewicht w_i kann ermittelt werden aus:

$$w_i = 1 - p_i = (1 - a)^{i-1}, \quad 1 \leq i \leq n \quad (3.4)$$

Die Aufteilung der Verteilung in die Fraktionen wird von drei Parametern gesteuert: der Anzahl Kugeln in einem Sample s , dem Anteil aktiver Kugel in einem Sample a und der Anzahl der benötigten Fraktionen n . In der praktischen Anwendung des Modells ist es sinnvoll, mit Rücksicht auf die Laufzeit der Simulation, die Parameter s und a fest vorzugeben. Die Anzahl der benötigten Fraktionen n muss dann so bestimmt werden, dass auch der seltenste Korngrößenintervall der gegebenen Verteilung mit wenigsten einer Kugel vertreten ist. Bei den hier verwendeten Korngrößenverteilungen ist der seltenste Korngrößenintervall auch gleichzeitig der größte, es handelt sich also um das letzte Intervall $[d_{\min}^{(n)}, d_{\max}^{(n)}]$ aus der letzten Fraktion F_n . Laut Gleichung 3.4 hat diese Fraktion die Gewichtung:

$$w_n = (1 - a)^{n-1}$$

Das letzte Sample kann keine Randkugeln aus höheren Fraktionen mehr enthalten, jede der s Kugeln k_1, \dots, k_s ist als aktive Kugel Teil der Fraktion F_n . Und jede dieser Kugeln repräsentiert damit einen Anteil von:

$$h_0(k_i) = \frac{w_n}{s} = \frac{(1-a)^{n-1}}{s}$$

an der Gesamtanzahl aller Partikel. Der maximale Anteil, den eine Kugel dieses Samples an der Gesamtmischung haben darf, ist durch die vorgegebene Korngrößenverteilung bekannt, es handelt sich um den gemessenen Anzahlanteil der Partikel im größten Durchmesserintervall. Geben wir diesen Wert als β vor, so erhalten wir die Gleichung:

$$\frac{(1-a)^{n-1}}{s} \leq \beta$$

Für die mindestens benötigte Anzahl Fraktionen erhalten wir dann:

$$n \geq \left\lceil 1 + \frac{\log(s\beta)}{\log(1-a)} \right\rceil \quad (3.5)$$

Wie Gleichung 3.5 zeigt, kann bei Festlegung einer ausreichend hohen Anzahl Fraktionen der benötigten Stichprobenumfang drastisch gesenkt werden. Ohne Beschränkung auf einen bestimmten Algorithmus kann man voraussetzen, dass dadurch auch die simulative Erzeugung der Packung deutlich beschleunigt wird. Für einen Einsatz des Simulationssystems innerhalb eines Optimierungsverfahrens ist dies von besonderer Bedeutung. Doch bei der Entscheidung „hohe Anzahl Fraktionen“ contra „großer Stichprobenumfang“ sind insbesondere zwei Faktoren zu berücksichtigen. Zum einen darf die Samplegröße nicht zu klein gewählt werden. Aufgrund der Zufälligkeit in der Anordnung der Kugeln innerhalb der Packung kommt es zu lokalen Schwankungen der Raumauffüllung. Damit global ein stabiler, reproduzierbarer Wert entsteht, müssen alle für die Packungsstruktur relevanten Korngrößen in ausreichendem Umfang im Sample vorhanden sein. Welche Folgen eine unzureichende Präsenz von wichtigen Korngrößen hat, wird in Kapitel 4 dargestellt. Zum anderen stellt die Simulation der Interaktionen zwischen den Fraktionen mittels Randkugeln nur eine Approximation der realen Zustände dar. Für eine hohe Realitätsnähe der simulierten Packung sollte deshalb eine möglichst geringe Anzahl Fraktionen angestrebt werden. Eine detailliertere Darstellung der hier vorgestellten Fraktionierung sowie eine Bewertung der Qualität der Simulationsergebnisse in Abhängigkeit von der Wahl der Parameter findet sich in [Ras10] und [KRW10].

3.2 Die Simulation

Die Eigenschaften einer zufällig dichten Partikelpackung simulativ am Computer zu ermitteln heißt, diese Packung virtuell nachzubauen. Meistens wird dazu der Entstehungsprozess der Packung mit einem geeigneten Modell simuliert. Eine Partikelpackung entsteht in einer Art von Sedimentationsprozess, bei dem die Partikel in einen Container eingebracht werden, auf den Boden fallen, sich dort ablagern und schließlich durch den Druck der anderen, auf ihnen lastenden Partikel verdichtet werden. Im Verlauf dieses Prozesses wirken verschiedene Kräfte auf die Partikel ein, die ihre endgültige Position innerhalb der Packung beeinflussen. Dominierend dabei ist die Schwerkraft, sie bewirkt die Ablagerung am Boden und die Verdichtung der so entstehenden Packung. Weitere Kräfte können durch die Materialeigenschaften ausgelöst werden: so beeinflussen auch die Reibung und magnetische oder elektrostatische Kräfte die Bewegung der Partikel.

Es existieren sehr viele unterschiedliche Modelle, mit denen sich das Verhalten von granularen Materialien, insbesondere die Entstehung einer dichten Packung, simulieren lassen. Dieses Kapitel stellt nur einige der wichtigsten, in Bezug auf ihre Anwendbarkeit für unser Simulationssystem, vor.

3.2.1 Die Diskrete Elemente Methode

Eines der vielseitigsten und mächtigsten Modelle um das Verhalten von Partikeln unter dem Einfluss verschiedener Kräfte zu untersuchen, ist die Diskrete Elemente Methode (DEM)¹. Es handelt sich dabei um ein numerisches Verfahren das ursprünglich von Cundall und Strack [CS79] entwickelt wurde. Die Positionen und Geschwindigkeiten der Partikel sowie die auf sie wirkenden Kräfte und die daraus resultierenden Beschleunigungen werden als Differentialgleichungssystem dargestellt, das mit Hilfe eines geeigneten numerischen Integrationsverfahrens gelöst wird. Da sich die DEM sehr eng an den physikalischen Gesetzen orientiert, sind die darauf basierenden Simulationen sehr realitätsnah. Die Anwendungsfelder gehen mittlerweile weit über die Simulation von granularen Materialien hinaus, so beschreiben Allen und Tildesley [AT91] die Möglichkeiten des Einsatzes in der Molekulardynamik. Besonders vielfältig sind auch die in der Computergrafik eingesetzten Partikelsysteme, deren Simulation oft auf der DEM basiert. Durch geeignete Datenstrukturen für die Kraftübertragung innerhalb fester Atomgitter lässt sich beispielsweise auch das Verhalten fester Körper nachbilden. In [YMWZ12] wird eine Methode beschrieben, bei der eine vorgegebene Geometrie durch Kugeln ausgefüllt wird, die ihrerseits in eine Federstruktur einge-

¹engl.: discrete element methode (manchmal auch „distinct element method“ genannt)

bunden sind. Dadurch ist die Simulation von elastischen Verformungen von Körpern möglich.

Die hohe physikalische Korrektheit der DEM erfordert einen entsprechend hohen Rechenaufwand. Die simulative Erzeugung einer stark polydispersen Partikelpackung, die trotz der in Kapitel 3.1 vorgestellten Fraktionierung eine hohe Anzahl Partikel erfordert, ist deshalb zeitaufwändig. Dennoch ist eine Simulation auf der Basis dieses Modells möglich und kann sinnvoll sein wenn z.B. die Vorgänge bei der Entstehung der Packung genauer untersucht werden sollen.

3.2.2 Collective Rearrangement

Im Rahmen dieser Arbeit sind hauptsächlich die Werte der fertigen - also statischen - Partikelpackung interessant. Die Vorgänge, die zur Entstehung dieser Packung führen, können auch stark abstrahiert werden, solange das Endergebnis realistisch ist. Deshalb können wir uns für die Simulation auf einfachere Modelle zurückziehen, die zwar nicht die realen physikalischen Vorgänge bei der Entstehung der Packung nachbilden, dafür aber eine schnellere Berechnung erlauben.

Das in unserem Simulationssystem eingesetzte Verfahren gehört zu den Collective Rearrangement-Algorithmen (CR). In der Literatur werden verschiedener Versionen von CR-Algorithmen beschrieben [HE98, HEC99, BBS02, BS06], die sich zwar in ihrer konkreten Arbeitsweise unterscheiden, dabei aber stets dem gleichen Grundprinzip folgen. Der Aufbau der Partikelpackung erfolgt in zwei Phasen:

1. **Anfangsplatzierung:** Die Partikel werden nacheinander an zufälligen Positionen innerhalb eines gegebenen Containers platziert. Überlappungen der Partikel sind dabei erlaubt. In der Regel ist die Größe des Containers so gewählt, dass er zu klein ist um die gesamte Partikelmenge aufzunehmen. Überlappungen sind somit unvermeidlich.
2. **Entlappung:** Nachdem der Container gefüllt ist, werden die Überlappungen zwischen den Partikeln schrittweise beseitigt. Dazu wird jeder Partikel so verschoben, dass die Überlappungen mit seinen Nachbarpartikeln minimiert werden. Die Vektoren für diese Verschiebungen können direkt aus den Überlappungen der Partikel mit ihren Nachbarn berechnet werden. Es wird also eine gegenseitige Abstoßung und Verdrängung der Partikel simuliert. Da der Volumenanspruch der Packung während der Entlappung steigt, muss zusätzliches Volumen geschaffen werden. Dies kann entweder durch Vergrößern des Containers, durch Verkleinern der Partikel oder durch Löschen einzelner Partikel geschehen.

Die wichtigsten Eigenschaften des für diese Arbeit entwickelten CR-Algorithmus sind:

Periodische Randbedingungen: Als Containerform wird in der Regel ein Würfel benutzt. Dadurch ist es möglich, die Packung an den Seitenflächen des Containers periodisch fortzusetzen und so den Wandeffekt zu neutralisieren. Konkret bedeutet das: ein Partikel der beispielsweise aus der linken Seitenfläche des Containers herausragt, ragt gleichzeitig zur rechten Seitenfläche des Containers hinein. Es entsteht also eine scheinbar unendlich große Packung.

Optimierte Anfangsplatzierung: Während der Anfangsplatzierung wird jeder Partikel mehrfach zufällig positioniert. Am Ende behält er die Position, in der er die wenigsten Überlappungen mit seinen Nachbarn hat. Durch diesen erhöhten Aufwand kann die anschließende Entlappungsphase deutlich beschleunigt werden.

Größenänderungen des Containers: Während der Entlappung werden weder die Größe noch die Anzahl der Partikel geändert. So ist sichergestellt, dass sich das Sample nicht von der vorgegebenen Korngrößenverteilung entfernen kann. Statt dessen wird der Container vergrößert, um sich dem wachsenden Volumenanspruch der Packung anzupassen. Es ist auch möglich, den Container während der Simulation mehrfach wieder zu verkleinern. Auf diese Weise kann, wenn auch nur indirekt, die verdichtende Wirkung der Schwerkraft auf die Packung simuliert werden.

Sequentielle Entlappung: In jedem Iterationsschritt der Entlappungsphase wird jeder Partikel genau einmal verschoben. Diese Verschiebung wird für jeden Partikel sofort nach der Berechnung des Vektors durchgeführt und nicht erst am Ende des Iterationsschritts, nachdem die Vektoren für alle Partikel berechnet wurden. Die Packung befindet sich also nicht erst am Ende eines Iterationsschritts in einem neuen Zustand mit weniger Überlappungen, sondern bereits nach jeder einzelnen Verschiebung. Durch dieses Vorgehen ist der Algorithmus sehr schnell in der sequentiellen Verarbeitung. Er eignet sich aber nicht für die parallele Verarbeitung auf mehreren Prozessoren.

3.3 Die Nachbarschaftsverwaltung

Ein in dieser Arbeit immer wiederkehrendes Problem ist das Handling einer großen Kugelmenge. Die Laufzeit der hier verwendeten Algorithmen hängt oft entscheidend davon ab, wie schnell eine bestimmte Kugel gefunden werden kann bzw. wie effizient der Zugriff auf einzelne Teil(-kugel-)mengen ist. So müssen während der Entlappungsphase des CR-Algorithmus in jedem Iterationsschritt pro Kugel alle überlappenden

Nachbarkugeln gefunden werden. Bei naivem Vorgehen müssten in jedem Iterationsschritt alle möglichen Kugelpaare untersucht werden, das sind bei N Kugeln $\binom{N}{2}$ Paare. Der Aufwand für diese Operationen beträgt dann also $O(N^2)$.

Da sich die Kugeln von Iterationsschritt zu Iterationsschritt aber nur geringfügig bewegen, verändert sich auch ihre Nachbarschaft nur allmählich über die Zeit. Dieser Umstand wird durch den Einsatz von Verlet-Listen [Ver67] ausgenutzt. Jede Kugel verfügt über so eine Liste, in der ihre Nachbarkugeln gespeichert werden. Die Liste einer Kugel wird nur dann aktualisiert, wenn sich die Nachbarschaft der Kugel tatsächlich geändert hat. Um das feststellen zu können, wird eine Verlet-Distanz δ_V festgelegt, die für alle Kugeln k_i , mit $0 < i \leq N$, konstant ist. Zusammen mit dem Radius einer Kugel ergibt die Verlet-Distanz den Radius einer Hüllkugel $r_i + \delta_V$. Alle Kugeln, die in dieser Hüllkugel, auch Verlet-Kugel $V(k_i)$ genannt, liegen oder sie überlappen, gehören zur Nachbarschaft der Kugel k_i . Diese Nachbarschaft muss erst dann aktualisiert werden, wenn sich der Mittelpunkt der Kugel k_i um mehr als $\delta_V/2$ von seinem Ursprung entfernt hat. Da die Nachbarschaftsrelation symmetrisch ist, muss nach jeder Aktualisierung k_i aus den Verlet-Listen aller Kugeln gelöscht werden, die ihrerseits nicht mehr zur Nachbarschaft von k_i gehören, und umgekehrt zu den Verlet-Listen der Kugeln hinzugefügt werden, die neu in der Nachbarschaft von k_i sind.

Durch den Einsatz der Verlet-Listen muss die Nachbarschaft der Kugeln zwar seltener aktualisiert werden, der Aufwand für die Aktualisierung selbst verringert sich dadurch aber nicht. Dafür werden Datenstrukturen benötigt, mit denen sich die Nachbarschaftssuche auf eine möglichst kleine Teilmenge der Kugeln beschränken lässt. Diese Strukturen basieren auf einer Zerlegung des zu untersuchenden Raumes (Container) in kleinere Teilräume. Bei der Suche nach den Nachbarkugeln einer Kugel k_i kann man sich so auf die Teilräume beschränken, die in räumlicher Nähe zu k_i liegen.

Eine einfache und weit verbreitete Methode ist die Verwendung eines Gitters [AW59, HGE74, GDK89, Wac06, SY07, BM91, YMT96], das gleichgroße Teilräume (Zellen) erzeugt. Wenn die Simulation innerhalb eines würfelförmigen Containers abläuft, ist die Verwendung eines orthogonalen Gitters naheliegend, so dass die Zellen ebenfalls würfelförmig sind. Zu jeder Zelle gehört eine Liste, in der die in der Zelle liegenden Kugeln gespeichert werden. Diese Struktur ist besonders für die Verwaltung einer monodispersen Kugelmenge geeignet. In diesem Fall genügt es, die Kantenlänge der Zellen auf den Durchmesser der Kugeln festzulegen und in jeder Zelle nur die Kugeln zu speichern, deren Mittelpunkt innerhalb der Zelle liegt. Die Nachbarschaft einer Kugel k_i ist dann mit der Zelle, die k_i enthält, sowie den 26 direkt angrenzenden Zellen gegeben. Die Verarbeitung einer polydispersen Kugelmenge, insbesondere mit einer für den Baustoff Beton typischen Korngrößenverteilung, ist jedoch weit weniger einfach.

Bereits die Festlegung einer geeigneten Kantenlänge für die Zellen, stellt ein Problem dar. Die Verwendung des größten Kugeldurchmessers ist nicht sinnvoll, denn es können Packungen auftreten, in denen sich die größte enthaltene Kugel über mehr als die Hälfte der Kantenlänge des Containers erstreckt. Für eine sinnvolle Zerlegung des Containers muss die Kantenlänge der Zellen also viel kleiner sein als der größte Kugeldurchmesser. Dann genügt es aber nicht mehr, eine Kugel nur in der Zelle zu speichern, in der ihr Mittelpunkt liegt. Alle Kugeln, deren Durchmesser größer ist als die Kantenlänge der Zellen, können sich über mehr als die 26 direkt an ihre Zelle grenzenden Nachbarzellen erstrecken. Sie müssen deshalb in allen Zellen gespeichert werden, in die sie hinein ragen. Durch diese Redundanz können sie bei der Erstellung bzw. Aktualisierung einer Verlet-Liste für eine Kugel k_i mehrfach auftreten. Diese Mehrfacheinträge müssen nachträglich gefunden und wieder entfernt werden, was einen zusätzlichen Aufwand bedeutet.

Als eine mögliche Lösung für diese Probleme schlägt Raschdorf vor, die größten Kugeln der Packung nicht in den Gitterzellen sondern in einer separaten Liste zu speichern. Diese Liste wird dann grundsätzlich der Nachbarschaft jeder Kugel hinzugefügt. Da die Anzahl der größten Kugeln in der Packung sehr gering ist, stellt diese (eventuell unnötige) Erweiterung der Nachbarschaft jeder Kugel innerhalb der Simulation kein großes Problem dar. Als eine weitere Verbesserung führt Raschdorf eine Einteilung in Sektoren ein. Dabei werden die Kugeln einer Zelle nach ihrer Lage innerhalb der Zelle klassifiziert. Kugeln gehören zum selben Sektor wenn sie aus der gleichen Seitenwand, der gleichen Kante oder der gleichen Ecke herausragen. Kugeln die vollständig innerhalb der Zelle liegen gehören zum zentralen Sektor. Bei der Nachbarschaftssuche für eine Kugel k_i müssen so alle Zellen berücksichtigt werden, die k_i enthalten und nur die Sektoren der angrenzenden Zellen, in denen Kugeln in Richtung k_i herausragen.

Als besonders gut geeignet für die Verwaltung einer polydispersen Kugelmenge haben sich aber Octrees [JT80, Mea82] erwiesen; Datenstrukturen, die den Raum (Container) hierarchisch in unterschiedlich große Zellen zerlegen. Das Prinzip eines Octrees ist in Abbildung 3.6 dargestellt. Der Container selbst bildet den Wurzelknoten eines Baumes. Er wird in 8 gleichgroße Zellen zerlegt, die ihrerseits die Kindknoten der Wurzel auf der 1ten Ebene des Baumes bilden. Jede dieser 8 Zellen ist ebenfalls wieder in 8 gleichgroße Zellen zerlegt, die dann die entsprechenden Kindknoten auf der 2ten Ebene des Baumes bilden. Diese Struktur setzt sich bis zu einer gewissen Tiefe fort. Die maximal notwendige Tiefe ist auf natürliche Weise durch die kleinsten Kugeln der Simulation festgelegt, da die kleinsten Knoten (auf der tiefsten Ebene des Baumes) noch groß genug sein müssen, um diese Kugeln aufzunehmen. Im praktischen Einsatz

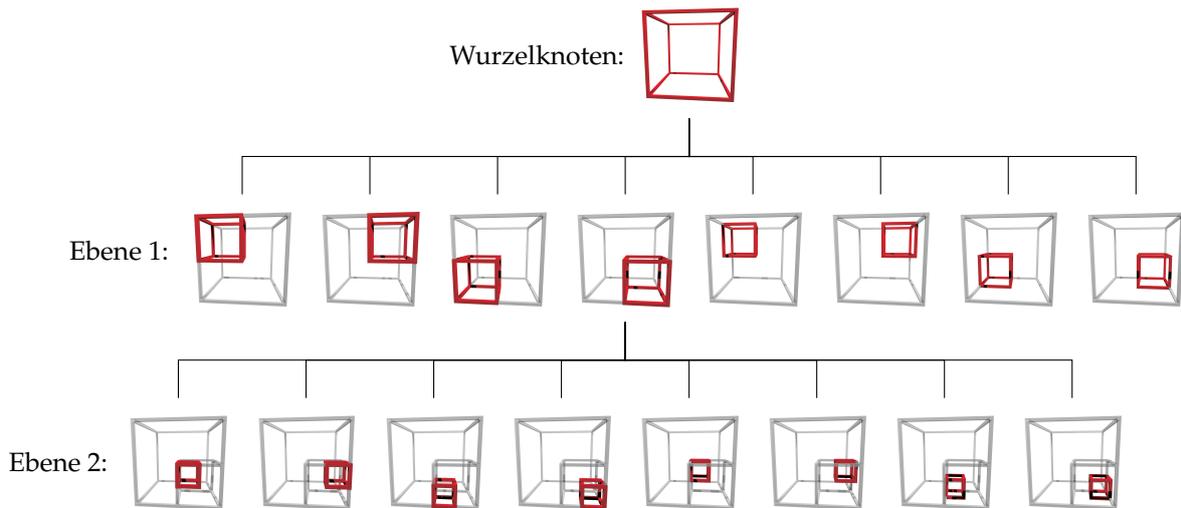


Abbildung 3.6: Prinzip des Octree

wird in der Regel ein Grenzwert für die Tiefe des Octrees festgesetzt, der nie überschritten werden darf. Auf diese Weise kann der Speicherbedarf der Datenstruktur begrenzt werden, denn die Anzahl der Knoten wächst mit der 8ten Potenz der Tiefe des Baumes.

Die Kugeln werden in diese Struktur gemäß ihrer Lage und Größe eingefügt. Dabei gilt die Regel: jede Kugel wird in den kleinsten (in der tiefsten Ebene liegenden) Knoten eingefügt, in den sie vollständig hineinpasst. Die Nachbarschaft einer Kugel k_i ist dann durch den Knoten gegeben, der die Kugel enthält, sowie durch den Teilbaum, der aus allen Vorfahren und allen Nachfahren dieses Knotens besteht. Der große Vorteil eines Octrees gegenüber einem Gitter besteht darin, dass die Verteilung seiner Zellgrößen gut mit den von uns verwendeten Korngrößenverteilungen harmonisiert. In jeder Ebene des Baumes halbiert sich die Kantenlänge der Zellen während sich ihre Anzahl um den Faktor 8 erhöht im Vergleich zur Vorgängerebene. In der Regel wird sich also der allergrößte Teil der zu verwaltenden Kugeln auf den tiefsten Ebenen des Baumes (also in den kleinsten Zellen) befinden. Dadurch muss für die Nachbarschaftssuche nur ein sehr kleiner Teilbaum untersucht werden.

Ein Nachteil des Octrees ist die strenge Forderung, dass eine Kugel vollständig in eine Zelle hinein passen muss, um in ihr gespeichert zu werden. Kugeln, die die Zellgrenzen überschneiden, müssen deshalb in höheren Ebenen und damit in unnötig großen Zellen abgelegt werden. Dadurch vergrößert sich auch der für diese Kugeln zu untersuchende Teilbaum, was sich negativ auf die Laufzeit der Simulation auswirkt. Um diesen Nachteil zu beheben, wurde der Octree zum sogenannten „Loose-Octree“ [Ulr00] erweitert. Beim Loose-Octree dürfen die Kugeln die Zellgrenzen auch über-

schneiden. Um zu gewährleisten, dass die Kugeln, die in einem Knoten gespeichert werden, eine bestimmte Größe nicht überschreiten, wird zusätzlich eine äußere Abgrenzung definiert. Die ursprüngliche Zellgrenze bildet dann die innere Abgrenzung. Eine Kugel kann dann in einem Knoten gespeichert werden, wenn sie vollständig innerhalb der äußeren Abgrenzung liegt und sich ihr Mittelpunkt innerhalb der inneren Abgrenzung befindet. Da eine Kugel nun auch in die Nachbarzellen hinein ragen kann, müssen diese inklusive ihrer Vorfahren und Nachfahren bei der Nachbarschaftssuche der Kugel mit berücksichtigt werden. Dies ist ein Nachteil, der jedoch mehr als ausgeglichen wird durch die Tatsache, dass sich nun ein Großteil der Kugeln auf die tieferen Ebenen des Baumes verteilt.

Datenstrukturen für die effiziente Verwaltung von umfangreichen Kugelmengen nehmen einen großen Raum ein sowohl in der Arbeit von Raschdorf [Ras10, KR11] als auch in der hier vorliegenden Arbeit. In Kapitel 6.3.3 werden weitere Datenstrukturen beschrieben, die bei einer ähnlichen Problemstellung, jedoch mit einer anderen Dynamik und in einer anderen geometrischen Struktur, zum Einsatz kommen. Alle diese Datenstrukturen bilden eine wichtige Voraussetzung für die Untersuchungen, die im Rahmen dieser Arbeit durchgeführt wurden und für den praktischen Einsatz des Simulationssystems innerhalb einer Optimierungsaufgabe.

Kapitel 4

Simulation besonderer Korngrößenverteilungen

Das im letzten Kapitel vorgestellte Simulationssystem löst zwei wesentliche Probleme, die der simulativen Bestimmung der Raumausfüllung von granularen Baustoffen entgegen steht. Die hierarchische Zerlegung der Korngrößenverteilung in Fraktionen beseitigt die Limitation der Simulation durch die maximal mögliche Samplinggröße. Grundsätzlich können so jetzt beliebig umfangreiche Korngrößenverteilungen verarbeitet werden. Außerdem macht die - auf einem Octree basierende - Nachbarschaftsverwaltung Simulationen mit sehr umfangreichen Kugelmengen in akzeptabler Zeit möglich. Diese beiden Faktoren machen das Simulationssystem mächtig genug, um praxisrelevante Korngrößenverteilungen virtuell per Computer zu testen. Die Zuverlässigkeit der simulativ ermittelten Ergebnisse stellt jedoch noch ein großes Problem dar. Das ist vor allem darauf zurückzuführen, dass Packungen aus kugelförmigen Partikeln mit glatter Oberfläche, wie sie von der Simulation erzeugt werden, einen Idealzustand darstellen. Einige Stoffe kommen diesem Idealzustand nahe. Beispielsweise zeigen Schüttungen aus Glaskugeln eine sehr gute Übereinstimmung mit der Simulation. Auch die Raumausfüllung vieler Sande, solange sie eine abgerundete Kornform haben, kann gut per Simulation ermittelt werden.

Sobald aber zwischen den Partikeln einer Schüttung, neben der verdichtend wirkenden Schwerkraft, noch weitere Kräfte relevant werden, können die Simulationsergebnisse weit von den tatsächlichen Werten abweichen. Herrscht zum Beispiel eine sehr starke Reibung zwischen den Partikeln, können sie sich nur schwer gegeneinander verschieben. Die Mischung kann dann nur unzureichend verdichtet werden und die Raumausfüllung der Packung bleibt deutlich niedriger als sie laut Simulation sein müsste. Auch Adhäsions- und Kohäsionskräfte, die besonders zwischen sehr feinen Partikeln zum tragen kommen, wirken der Entstehung einer dichten Packung entgegen.

gen.

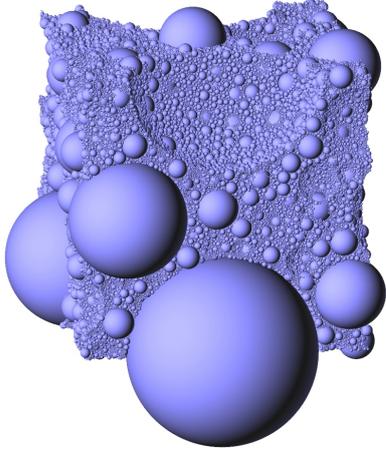
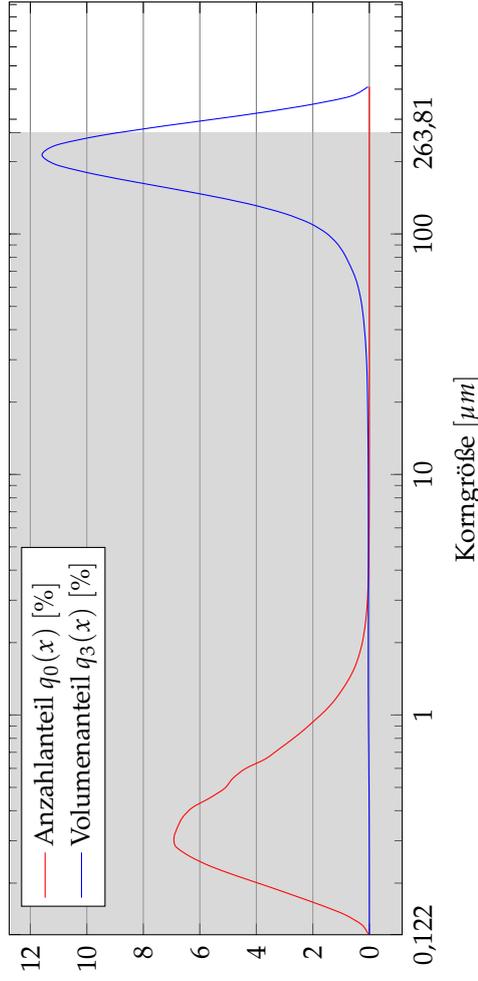
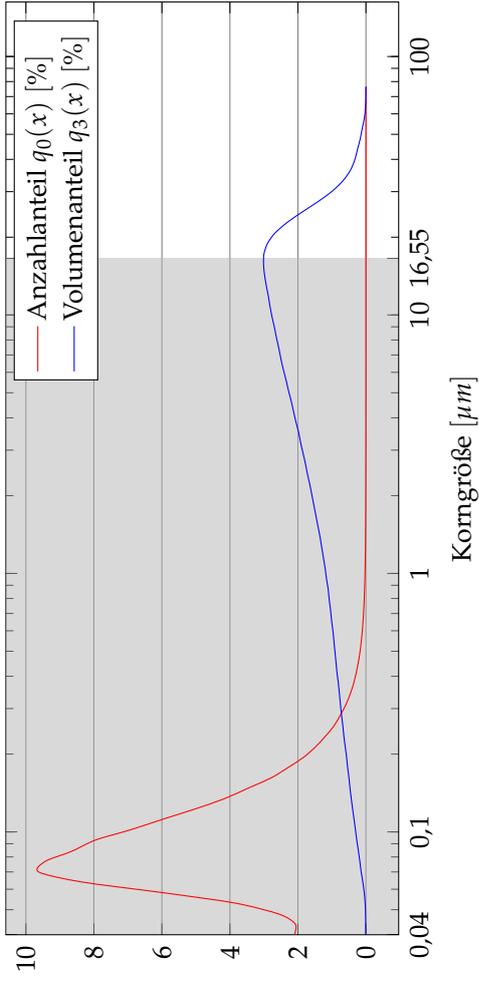
Diese Phänomene und ihre Berücksichtigung in der Simulation werden in einem späteren Kapitel behandelt. Darüber hinaus kann aber auch die Art der Korngrößenverteilung selbst zu einem Problem für die Simulation werden. Dieses Kapitel beschreibt die Ursachen dafür und zeigt mögliche Lösungen.

4.1 Dominante große Partikel

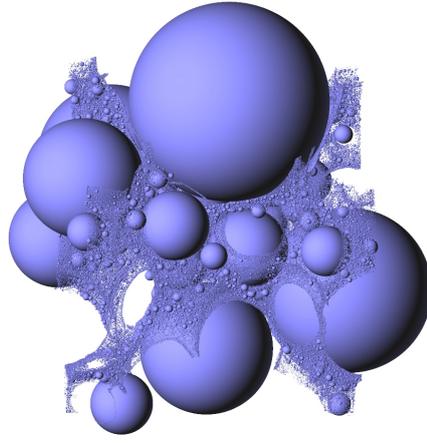
In der Praxis vorkommende Partikelschüttungen verhalten sich in der Regel nicht in der Weise ideal, wie es apollonische Packungen tun. Das heißt, dass sich die Partikel jeder Korngrößenklasse nicht so in die Lücken zwischen den nächst größeren Partikeln einfügen, dass sie diese optimal ausfüllen ohne die größeren Partikel auseinander zu drängen. Partikelschüttungen werden also in der Praxis meistens von einigen Korngrößenklassen dominiert. Abhängig davon welche Korngrößenklassen dominant sind, ergeben sich grundsätzlich zwei mögliche Packungsstrukturen. Diese Tatsache wird auch schon im hierarchischen Simulationsmodell bei der rekursiven Berechnung des Containervolumens (Kapitel 3.1.2, Gleichung 3.2) berücksichtigt.

Wie ein CR-Algorithmus diese beiden Packungsstrukturen verarbeiten kann, wird in Abbildung 4.1¹ dargestellt. Der Teil (a) der Abbildung zeigt die typische Verteilung eines Zement CEM I. Aus dieser Verteilung wurde ein Sample mit 1.000.000 Kugeln erzeugt, die dann per Simulation zu einer zufällig dichten Packung arrangiert wurden, die in Teil (b) dargestellt ist. Wie bereits in Kapitel 3.1.2 gezeigt wurde, sind extreme Samplinggrößen notwendig, um alle Korngrößen einer derartigen Verteilung abzubilden. Deshalb ist klar, dass in einem Sample aus 1.000.000 Kugeln nicht alle Korngrößen enthalten sein können. Der graue Bereich im Verteilungsdiagramm markiert alle Korngrößen, die vom Sample erfasst werden konnten. Immerhin reicht die Samplinggröße, um sowohl die vom Anzahlanteil ($q_0(x)$) als auch die vom Volumenanteil ($q_3(x)$) stärksten Korngrößenintervalle abzubilden. Die Charakteristik der Verteilung kann so ausreichend gut dargestellt werden. Mit 99,9% an der Gesamtanzahl liegen die meisten Partikel bei dieser Verteilung zwischen $0,044\mu m$ und $1,15\mu m$, sie nehmen gleichzeitig auch 20,7% des Gesamtvolumens aller Partikel ein. Betrachten wir die daraus erzeugte Packung so fällt auf, dass sie von den kleineren Partikeln dominiert wird. Die wenigen großen Partikel sind vollständig zwischen den kleinen Partikeln eingebettet, sie „schwimmen“ zwischen ihnen ohne sich gegenseitig zu berühren. Die von der Simulation ermittelte Raumausfüllung dieser Packung beträgt 86,9%.

¹Die dargestellten Packungsbilder sowie einige weitere Bilder in dieser Arbeit, wurden mit einer modifizierten Version von QuteMol [TCM06, Qut07] erstellt. Siehe auch Kapitel A.4.



(b)



(d)

Abbildung 4.1: Verteilungskurven und simulierte Packung eines Zement CEM I (a und b) sowie eines feinen Quarzsandes (c und d). Die beiden Packungen wurden aus Stichproben mit jeweils 1 Mio. Kugeln erzeugt. Wieviel der gegebenen Korngrößenverteilungen damit abgedeckt werden können, wird durch den grau hinterlegten Bereich markiert.

Eine andere Packungsstruktur entsteht durch Verteilungen, die dem Beispiel im Teil (c) der Abbildung entsprechen. Es handelt sich dabei um die Verteilung eines feinen Quarzsandes, die ähnlich umfangreich ist wie die des Zements. Die meisten Partikel - mit 99,95% an der Gesamtanzahl - liegen hier zwischen $0,122\mu\text{m}$ und $5,1\mu\text{m}$, ihr Anteil am Gesamtvolumen aller Partikel beträgt aber nur 0,87%. Auch hier wurde per Simulation aus einem Sample von 1.000.000 Kugeln eine Packung erzeugt, die im Teil (d) der Abbildung zu sehen ist. Diese Packung wird von wenigen großen Partikeln dominiert. Das Gesamtvolumen aller feinen Partikel reicht nicht aus, um die Lücken zwischen diesen großen Partikeln zu füllen oder sie sogar auseinander zu drängen. Trotzdem bilden die dominanten großen Partikel hier keine dichte Packungsstruktur. Genau wie bei der Packung im Teil (b) der Abbildung berühren sie sich nicht gegenseitig. Die von der Simulation ermittelte Raumauffüllung liegt bei nur 55,09% und damit sogar deutlich unter dem Wert, den man für eine zufällig dichte monodisperse Kugelpackung erwarten kann. Der Grund für dieses ungewöhnliche Ergebnis liegt in der besonderen Korngrößenverteilung des Samples. Tatsächlich zeigt dieses Beispiel, dass Verteilungen existieren, aus denen CR-Algorithmen keine dichten Packungen erzeugen können. Um dieses Phänomen und seine Auswirkungen genauer zu untersuchen, wurden Simulationen mit unterschiedlichen, kritischen Verteilungen durchgeführt. Die Ergebnisse sowie geeignete Gegenmaßnahmen werden in den folgenden Kapiteln vorgestellt.

4.1.1 Simulationen mit kritischen Verteilungen

Ausgehend von den im vorherigen Kapitel vorgestellten Beispielen wurden zwei unterschiedliche Korngrößenverteilungen generiert. Die erste dieser Verteilungen wird von kleinen Partikeln dominiert und wird mit „KKDom“ bezeichnet. Bei der zweiten Verteilung dominieren die großen Partikel, wir bezeichnen sie mit „GKDom“. „KKDom“ und „GKDom“ entsprechen weitestgehend den Verteilungen des Zements und des Quarzsandes aus den Beispielen in Abbildung 4.1. Sie wurden lediglich entlang der x-Achse in einen anderen Korngrößenbereich verschoben, so dass die jeweiligen Bereiche mit den höchsten Anzahlanteilen ($q_0(x)$) übereinander zu liegen kommen. Dadurch werden Vergleiche zwischen den beiden Verteilungen vereinfacht. Da unser CR-Algorithmus keinerlei Faktoren berücksichtigt, bei denen die absolute Größe der Partikel eine Rolle spielt (z.B. Schwerkraft, Trägheit, Reibung oder an der Oberfläche wirkende Anziehungskräfte), hat solch eine relative Größenveränderung keinerlei Auswirkungen auf die Eigenschaften der fertigen Packung. Da die Größenverhältnisse zwischen den einzelnen Partikeln erhalten bleiben, wirkt die Verschiebung einer Ver-

teilung wie eine Skalierung der gesamten Packung. Für die Raumausfüllung gilt:

$$\Phi = \frac{V_k}{V_c} = \frac{\sum_i \frac{1}{6} \pi d_i^3}{l^3}$$

Wobei d_i der Durchmesser der i -ten Kugeln der Packung und l die Kantenlänge des würfelförmigen umgebenden Containers ist. Wird nun die gesamte Packung mit einem festen Faktor S skaliert, ergibt sich:

$$\Phi = \frac{\sum_i \frac{1}{6} \pi (Sd_i)^3}{(Sl)^3} = \frac{S^3 \sum_i \frac{1}{6} \pi d_i^3}{S^3 l^3} = \frac{\sum_i \frac{1}{6} \pi d_i^3}{l^3}$$

Das heißt, es ändern sich zwar die absoluten Werte für die Größen und Volumen von Kugeln und Container, ihre Relation zueinander und damit die Raumausfüllung Φ bleibt aber gleich. Wir dürfen also davon ausgehen, dass sich „KKDom“ und „GKDom“ in der Simulation genauso verhalten wie die Verteilungen des Zements und des Quarzsandes aus dem Beispiel.

Aus den beiden Verteilungen „KKDom“ und „GKDom“ wurden verschiedene Mischungen mit unterschiedlichen Anteilen erzeugt. Diese Mischungen sind in Abbildung 4.2 aufgelistet. Die Anteile, bezogen auf das Volumen, die jede der beiden Komponenten an der Mischung hat, sind unter den Verteilungen aufgeführt. Betrachtet man, wie sich die Kurven für die Anzahlanteile ($q_0(x)$) und die Volumenanteile ($q_3(x)$) bei unterschiedlicher Mischung von „KKDom“ und „GKDom“ verändern so fällt auf, dass sich die Anzahlanteile nur geringfügig verändern. Der Korngrößenbereich mit den höchsten Anzahlanteilen liegt bei allen Mischungen zwischen etwa $0,1\mu m$ und $2,0\mu m$ mit einem Maximum bei etwa $0,25\mu m$. Alle größeren Partikel kommen nur sehr selten in den Mischungen vor mit Anteilen von weniger als 1%.

Ein völlig anderes Verhalten zeigen die Kurven für die Volumenanteile. Partikel im Korngrößenbereich zwischen $40\mu m$ und $400\mu m$, nehmen in einer Mischung aus 100% „KKDom“ noch einen mäßig großen Anteil des Volumens ein. Das Maximum beträgt 2.98% und liegt bei $48\mu m$. Bei zunehmenden Anteil „GKDom“ in der Mischung nimmt dieser Anteil deutlich zu. Das Maximum beträgt am Ende 11.58% und liegt bei $213\mu m$, also bei Partikeln die mehr als 4 mal größer sind. Obwohl die Anzahl der Partikel in diesem Korngrößenbereich um weniger als $2 \cdot 10^{-3}\%$ ansteigt, reicht diese Veränderung aus, um nun fast das gesamte Volumen der Mischung in diesem Bereich zu konzentrieren.

Um herauszufinden, wie ein CR-Algorithmus Partikelmischungen mit den dargestellten Verteilungen verarbeitet, wurden je 10 Simulationsläufe mit einer Samplinggröße von 50.000 Kugeln durchgeführt. Die Ergebnisse sind in Abbildung 4.3 darge-

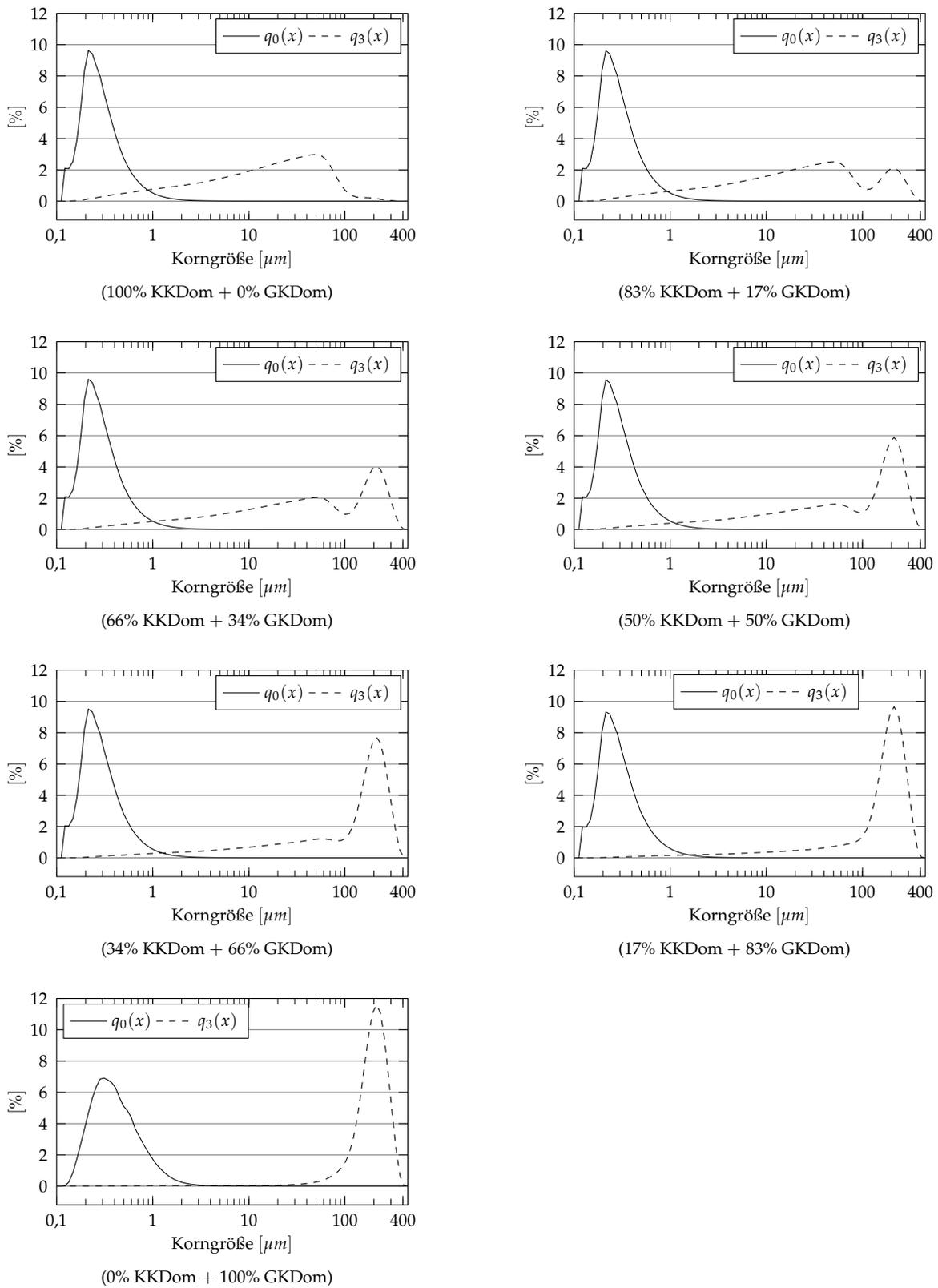


Abbildung 4.2: Verschiedene Mischungen aus einer von kleine Kugeln domininierten- (KKDom) und einer von großen Kugeln dominierten (GKDom) Korngrößenverteilung

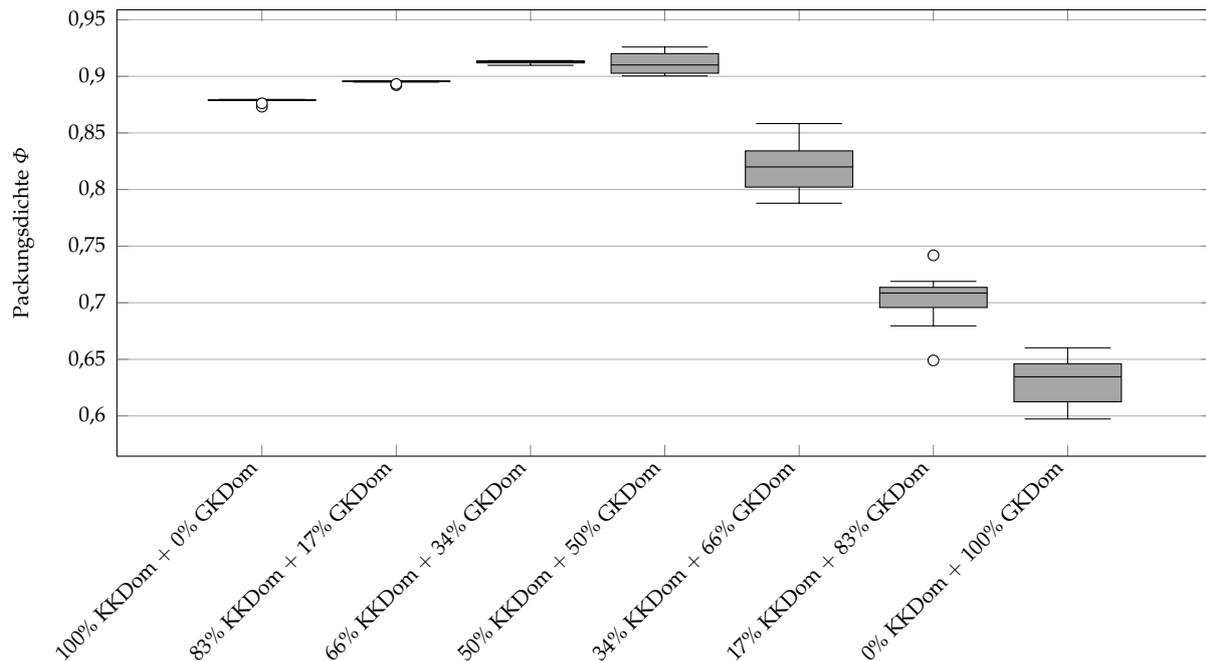


Abbildung 4.3: Per Simulation ermittelte Packungsdichten für verschiedene Mischungen aus einer von kleine Kugeln dominierten- (KKDom) und einer von großen Kugeln dominierten (GKDom) Korngrößenverteilung

stellt. Natürlich unterscheiden sich die ermittelten Packungsdichten für jede der 7 Mischungen, interessanter ist jedoch, welche Varianzen die Ergebnisse aufweisen. Die ersten 3 Mischungen, in denen der „KKDom“-Anteil überwiegt, zeigen keine Auffälligkeiten in ihren Simulationsergebnissen. Alle Werte streuen um weniger als 1%. Ab einem Mischungsverhältnis von 50% „KKDom“ und 50% „GKDom“ nimmt die Varianz jedoch deutlich zu. In den letzten 3 Mischungen, bei denen der „GKDom“-Anteil überwiegt, sind die Simulationsergebnisse in Bereichen von bis zu 10% gestreut.

Einen detaillierteren Einblick in eine der Simulationen liefert die Abbildung 4.4. Es handelt sich dabei um die Mischung aus 0% KKDom und 100% GKDom. Dargestellt sind die Packungen der Fraktionen 1, 3 und 10 sowie ihre Lage innerhalb der Korngrößenverteilungen. Die Simulation bestand insgesamt aus 13 Fraktionen. In der Packung der Fraktion 1 reichen die Korngrößen von $0,122\mu\text{m}$ bis $39,589\mu\text{m}$. Sie bleiben damit knapp unterhalb des Bereiches mit den höchsten Volumenanteilen. Die Packung enthält zwar einige sehr große Kugeln, diese werden jedoch von der dichten Packung der kleineren Kugeln eingehüllt, woraus eine hohe Packungsdichte und eine hohe Kontaktzahl resultieren. Die Packung der Fraktion 3 enthält Kugeln von $0,743\mu\text{m}$ bis $188,518\mu\text{m}$ Größe und reicht damit weit hinein in den Bereich mit den höchsten Volumenanteilen. Der größte Teil des Volumens der Packung wird hier von nur 3 sehr großen Kugeln gebildet. Obwohl die Anzahl der restlichen Kugeln sehr hoch ist, kön-

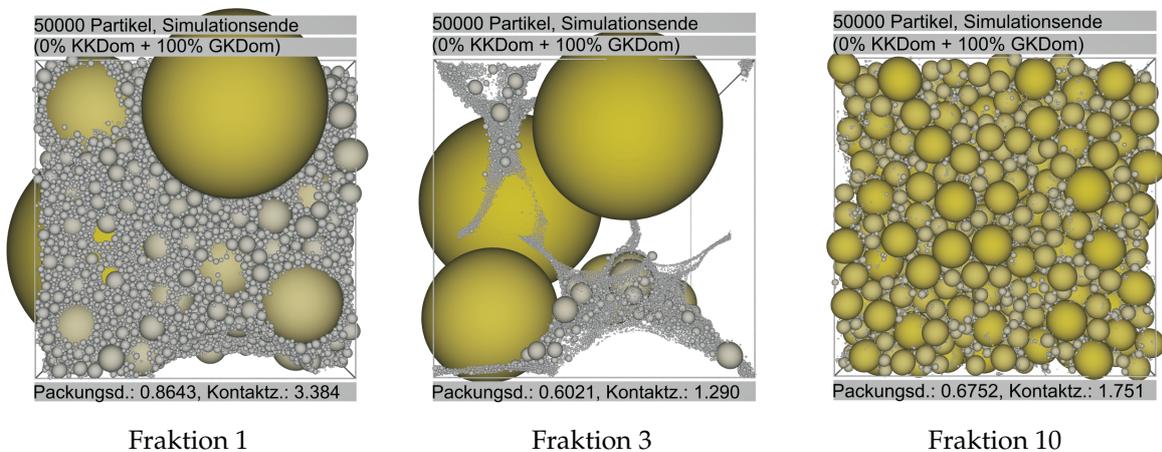
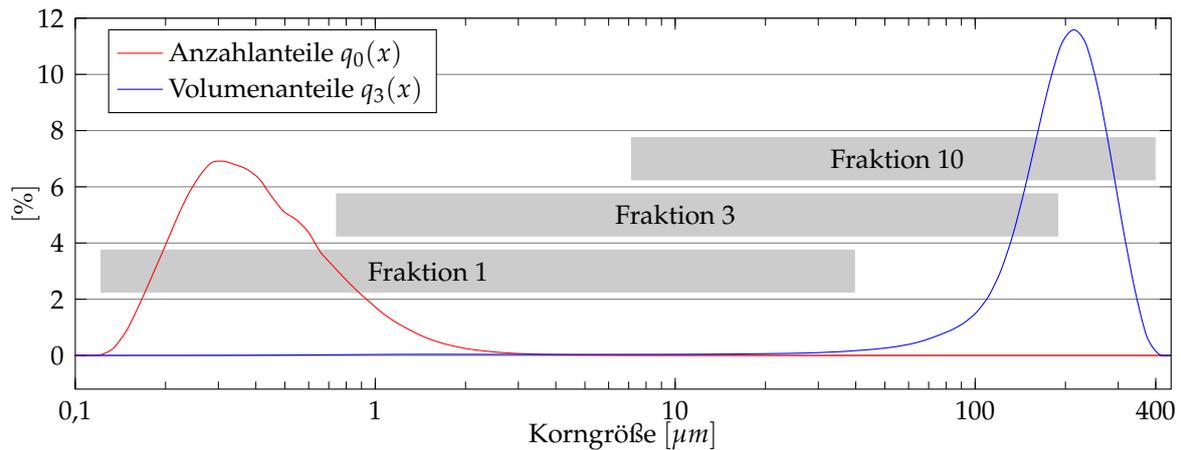


Abbildung 4.4: Ausschnitte aus einer Simulation der Mischung aus 0% KKDom und 100% GKDom

nen sie keine dichte Packung bilden. Ihre Größe ist zu gering, um das große Leervolumen des Containers auszufüllen. Wenn der CR-Algorithmus im Verlauf der Simulation ihre Überlappungen beseitigt, steht sehr viel Platz zur Verfügung. Statt einer dichten Packung, in der kleine Kugeln in die Lücken zwischen den größeren Kugeln gedrängt werden, entsteht eine sehr lose Packung mit einer geringen Kontaktzahl. Ebenso können auch die großen Kugeln keine dichte Packung bilden. Zwar nehmen sie durch ihre Größe auch sehr viel Volumen ein, dafür ist aber ihre Anzahl viel zu gering. Ein völlig anderes Verhalten zeigt die Packung der Fraktion 10. Hier reichen die Korngrößen von $7,154\mu\text{m}$ bis $397,74\mu\text{m}$. Auch diese Packung wird von den großen Kugeln dominiert. Doch in diesem Fall ist ihre Anzahl groß genug, um eine dichte Packung zu bilden.

Die hohen Varianzen in einigen der per Simulation ermittelten Packungsdichten sind darauf zurückzuführen, dass sich die Packungen einiger ihrer Fraktionen ähnlich verhalten, wie die Packung der Fraktion 3 in Abbildung 4.4. Das heißt, dass dort die kleinen Kugeln lediglich den Effekt haben, die Leerraum des Containers geringfügig

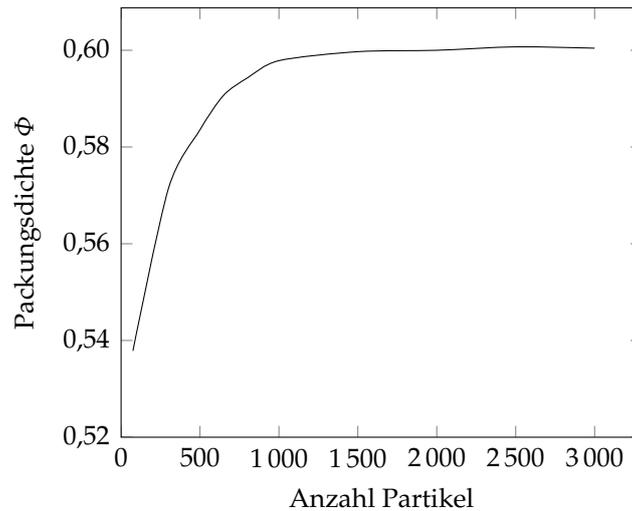


Abbildung 4.5: Abhängigkeit der Packungsdichte von der Anzahl Partikel nach [HE98]

aufzufüllen. Ihr Volumen ist jedoch, im Verhältnis zu den großen Kugeln, so gering, dass sie deren Bewegungen während der Simulation kaum beeinflussen können. Die großen Kugeln können sich damit fast so verhalten, als wären die kleinen Kugeln nicht vorhanden. Das entspricht einer Simulation, die ein sehr kleines Sample verarbeitet, das nur aus großen Kugeln besteht.

4.1.2 Auswirkungen der Samplinggröße auf die Raumausfüllung

Hinweise, welche Auswirkungen sehr kleine Samplinggrößen auf die Simulationsergebnisse haben, finden sich schon in anderen Arbeiten. So beschreiben He und Ekere [HE98] Simulationen, bei denen unterschiedlich große monodisperse Kugelmengen in einem kugelförmigen Container gepackt werden. Die jeweils erzielten Packungsdichten sind in Abbildung 4.5 dargestellt. Es zeigt sich, dass die Packungsdichte deutlich abnimmt, wenn die Anzahl kugelförmiger Partikel unter 1000 sinkt. Das ist vor allem auf die zunehmende Wirkung des Wandeffektes zurückzuführen, da ein kleinerer Container eine, im Verhältnis zum Volumen der Packung, größere Wandfläche hat. Darüber hinaus beschreiben He und Ekere aber auch, dass bei kleinen Samplinggrößen höhere Varianzen bei den Simulationsergebnissen auftreten. Durch die zufällige Lage der einzelnen Kugeln innerhalb der Packung unterliegt die Packungsdichte lokalen Schwankungen (an einigen Stellen liegen die Kugeln zufällig etwas dichter, an anderen Stellen etwas weniger dicht). Betrachtet man die Packung global, so gleichen sich diese Schwankungen gegenseitig aus und führen zu relativ stabilen Werten für die Packungsdichte. Ist die Samplinggröße aber zu gering, ist ein Ausgleich nicht mehr möglich. Die lokalen Schwankungen führen dann zu erheblichen Varianzen in den Si-

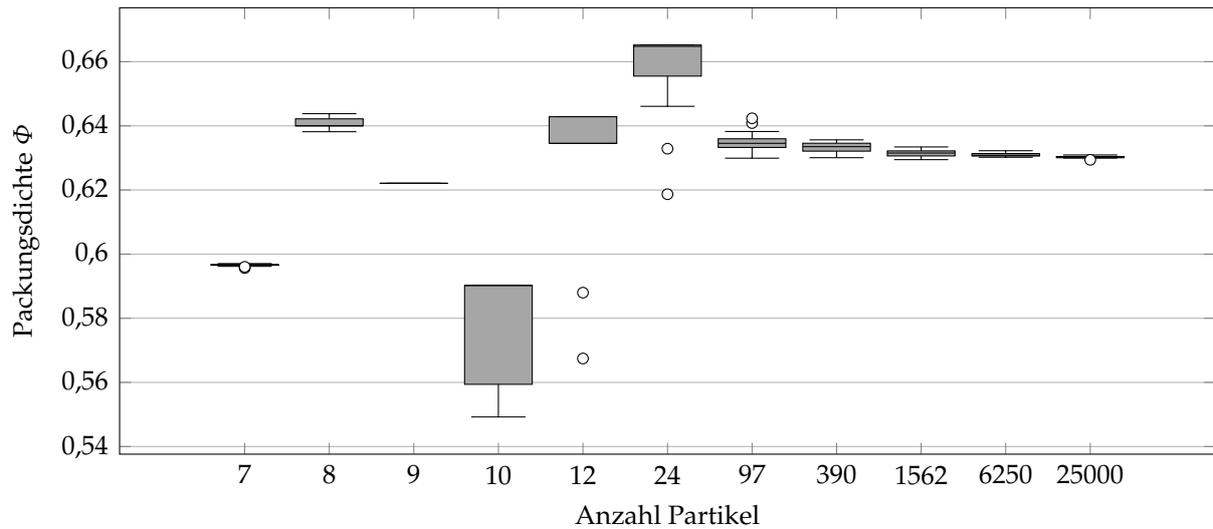


Abbildung 4.6: Per Simulation ermittelte Packungsdichten einer monodispersen Packung bei unterschiedlicher Samplinggröße

mulationsergebnissen.

Eine ähnliche Messreihe für unser Simulationsprogramm ist in Abbildung 4.6 dargestellt. Für jede der Samplinggrößen wurden hier 20 Simulationen durchgeführt. Auch hier liefern Packungen mit mehr als 1000 Kugeln eine Packungsdichte von etwa 63% mit nur sehr geringen Varianzen. Werden weniger als 1000 Kugeln in der Simulation verwendet, zeigen sich aber mehrere Effekte. Zunächst (von 97 bis 390 Kugeln) nimmt nur die Varianz zu. Da wir in unserem Simulationssystem aber einen würfelförmigen Container mit periodischen Randbedingungen verwenden, kann keine Abnahme der Packungsdichte aufgrund des Wandeffektes beobachtet werden. Bei noch kleineren Packungen (von 10 bis 24 Kugeln) ist nicht alleine eine deutliche Zunahme der Varianzen zu beobachten, zusätzlich weichen die Simulationsergebnisse auch stark von den 63% ab. Da hier nur noch sehr wenige Kugeln zur Verfügung stehen, wirken sich nicht mehr nur die lokalen Schwankungen auf die Packungsdichte aus. Das Fehlen einer einzigen Kugel kann nun eine Lücke in der Packungsstruktur hinterlassen, die von den verbleibenden Kugeln nicht mehr aufgefüllt werden kann. Solche Packungen sind dann im Mittel weniger dicht. Entzieht man aber noch weitere Kugeln, gibt es für die verbleibende Kugeln wieder neue Möglichkeiten sich dichter anzuordnen. Noch deutlicher wird dies bei den Ergebnissen für Packungen aus 7 bis 9 Kugeln. Hier gibt es nur noch geringe Varianzen. Aufgrund der sehr geringen Anzahl Kugeln und des entsprechend kleinen Containers, gibt es nicht mehr sehr viele Möglichkeiten eine Packung zu bilden. Da die Anordnung der Kugeln an den Containerwänden periodisch wiederholt wird, ergeben sich so Kristallstrukturen, die den in Kapitel 2.3.1 vorgestellten Packungen nahe kommen.

Betrachten wir z.B. die Ergebnisse für Packungen aus 8 Kugeln: diese Kugeln bilden die Eckpunkte eines (möglicherweise verzerrten) Würfels. Die entstehende Struktur könnte also einer kubischen, einer einfachen oder einer doppelt versetzten Kugelpackung (Abbildung 2.3) ähnlich sehen. Zu Beginn jeder Simulation wird der Container auf eine so kleine Größe gesetzt, dass die Kugeln nicht genug Platz finden und Überlappungen unvermeidlich sind. Außerdem benutzt unsere Simulation eine optimierte Anfangsplatzierung, bei der aus mehreren zufällig gewählten Positionen für eine Kugel diejenige gewählt wird, die die geringsten Überlappungen aufweist. Diese Faktoren machen es extrem unwahrscheinlich, dass die 8 Kugeln jeweils in den 8 Eckpunkten des Würfels platziert werden, denn dort würden sie den meisten Platz beanspruchen. Die Struktur wird also nicht einer kubischen sondern eher einer einfachen oder doppelt versetzten Kugelpackung ähneln. Tatsächlich zeigen die Simulationsergebnisse eine Packungsdichte von etwa 64% mit nur geringer Varianz. Mit einem verbleibenden Hohlraumanteil von 36% liegen diese Werte etwa zwischen denen der einfachen und der doppelt versetzten Kugelpackung (Tabelle 2.2). Entfernt man aus dem Sample nur eine Kugel bzw. fügt eine Kugel hinzu, entstehen andere Strukturen mit deutlich geringeren Packungsdichten wie man an den Ergebnissen für Packungen aus 7 bzw. 9 Kugeln sehen kann.

4.1.3 Geeignete Gegenmaßnahmen

Die vorgestellten Beispiele aus Mischungen der „KKDom“- und „GKDom“-Verteilungen sowie der monodispersen Packungen mit extrem kleinen Samplinggrößen machen deutlich, dass Simulationsergebnisse aus Packungen, die von wenigen großen Kugeln dominiert werden, nicht zuverlässig sind. Als problematisch haben sich nicht nur die starken Varianzen in den Ergebnissen erwiesen. Aufgrund ihrer sehr geringen Anzahl führt das Vorhandensein oder Fehlen von nur einer einzigen der großen dominanten Kugeln, zu einem erheblichen Ansteigen oder Absinken der Packungsdichte. Deshalb kann selbst der Mittelwert aus mehreren Simulationen mit dem gleichen Sample nicht als verlässlicher Wert angesehen werden.

Damit die Simulation aus jeder gegebenen Korngrößenverteilung Packungen mit verwertbaren, realistischen Eigenschaften erzeugen kann, muss dafür gesorgt werden, dass die dominierenden Korngrößen in ausreichend hoher Anzahl vorhanden sind, um eine dichte Packungsstruktur zu bilden. Das ist immer dann gegeben, wenn die kleinen Korngrößen dominant sind, da diese dann überproportional oft in der Verteilung vorkommen. Wenn jedoch die großen Korngrößen die Packung dominieren, kann dies, bei den bisher benutzten Samplingverfahren, nicht garantiert werden. Alle in Kapitel 3.1.1 vorgestellten Verfahren ziehen ihre Stichproben auf einem festgelegten Bereich

der y-Achse der Verteilungssumme $Q_0(x)$. Dieser Bereich reicht von einem Startanteil - der liegt normalerweise bei 0%, bei mehr als einer zu simulierenden Fraktion ist es der Endanteil der aktiven Kugeln der vorherigen Fraktion - bis zum Ende der Verteilungsfunktion also 100%. Da in den für die granularen Bestandteile von Betonmischungen typischen Korngrößenverteilungen die Anzahlanteile der einzelnen Korngrößenklassen mit steigendem Durchmesser drastisch sinken, führen diese Verfahren dazu, dass die größte in einem Sample vorhandene Korngrößenklasse in der Regel von nur einer einzelnen Kugel repräsentiert wird.

Eine einfache und naheliegende Maßnahme, um die Anzahl der größten Kugeln in einem Sample zu erhöhen, wäre die Vergrößerung des Samples. Wie aber schon in Kapitel 3.1.2 erklärt wurde, ist dies durch das Laufzeitverhalten der Simulation und die Möglichkeiten der verfügbaren Hardware begrenzt. Außerdem zeigt bereits das Beispiel in Abbildung 4.1, dass es Verteilungen gibt, bei denen selbst ein Sample aus 1 Mio. Kugeln eine für die Simulation zu geringe Anzahl der dominanten großen Kugeln enthält. Besser wäre es, die Sample flexible an die gegebene Korngrößenverteilung anzupassen. Dazu wird die Kugelmenge jeder Fraktion noch vor ihrer Simulation überprüft. Enthält sie große dominante Kugeln, deren Anzahl aber zu gering für eine sinnvolle Simulation ist, so müssen diese entfernt werden. Die Korngrößenverteilung in den folgenden Fraktionen muss sich entsprechend verändern. Das Sample jeder Fraktion wird gemäß des vorgegebenen Anteils a in aktive Kugeln und Randkugeln unterteilt. Der jeweilige Endanteil der aktiven Kugeln, der mit Hilfe der Verteilungssumme $Q_0(x)$ ermittelt werden kann, ist gleichzeitig der Startanteil der nächsten Fraktion. Wird also eine Fraktion schmaler, da einige Kugeln aus ihrem Sample entfernt werden mussten, wirkt sich das auf alle folgenden Fraktionen aus.

Da sich bei dieser Methode die Anzahl der benötigten Fraktionen sowie deren Start- und Endanteile erst im Verlauf der Simulation ergeben, ist es nicht mehr möglich diese Parameter wie in Kapitel 3.1.3 (Gleichung 3.5) vorab zu berechnen. Die folgenden Kapitel erklären wie die neue Fraktionierungsmethode im Detail funktioniert.

4.2 Online-Fraktionierung

Da bei der bisherigen Fraktionierungsmethode die Anzahl der benötigten Fraktionen noch vor dem Programmstart berechnet wird, und dieser Wert der Simulation beim Start als Parameter übergeben wird, können wir sie als „Offline-Fraktionierung“ bezeichnen. Die neue Methode bezeichnen wir als „Online-Fraktionierung“, da der entsprechende Startparameter von der Simulation nicht mehr verwendet wird. Statt dessen wird die Anzahl der benötigten Fraktionen nun vom Programm selbständig er-

mittelt. Dadurch kann die Simulation flexibler auf die Besonderheiten der gegebenen Korngrößenverteilung reagieren. Außerdem müssen so weniger Vorarbeiten für den Simulationsstart geleistet werden, was die Bedienung für den Anwender erleichtert.

Abgesehen von einigen Erweiterungen arbeitet die Online-Fraktionierung genau wie die Offline-Fraktionierung. Das bedeutet, dass es keine Auswirkungen auf das in Kapitel 3.1.2 vorgestellte Modell der hierarchischen Simulation gibt. Lediglich die Ermittlung der Gewichte w_i (Gleichung 3.4), zur Transformation der Simulationsergebnisse auf die korrekten Relationen der Fraktionen, muss entsprechend angepasst werden.

Die Online-Fraktionierung benutzt im Kern dieselben Samplingverfahren wie die Offline-Fraktionierung. Die einzige Änderung besteht hier darin, dass die Stichproben auf einem Bereich der y -Achse der Verteilungssumme $Q_0(x)$ gezogen werden, der vom Startanteil der aktuellen Fraktion bis zu einer Obergrenze o reicht. Diese Obergrenze ist zu Beginn, wie bei der Offline-Fraktionierung, das Ende der Verteilungsfunktion 1. Um die Breite der Fraktion auf den Korngrößen zu verringern, und somit große dominante Kugeln auszuschließen, kann die Obergrenze gesenkt werden. Würde man die entsprechenden Kugeln einfach nachträglich aus dem Sample löschen, würde das zu einer Verzerrung der Korngrößenverteilung führen. Die Methode der Beschränkung durch eine Obergrenze stellt sicher, dass alle Korngrößenklassen, die im Sample enthalten sind, mit den korrekten Anzahlanteilen repräsentiert sind. Allerdings bricht das Sample mit der Korngröße x , deren Anteil der Obergrenze entspricht, ab:

$$Q_0^{-1}(o) = x$$

Im Sample sind also keine Kugeln enthalten, die größer sind als dieses x . Die Gleichung 3.4 verändert sich zu:

$$w_i = o - p_i$$

Dabei ist p_i der Startanteil der Fraktion F_i . Um ein gültiges Sample für die Simulation zu erzeugen, wird dieses Verfahren gegebenenfalls mehrfach wiederholt. Dazu wird das Sample auf einige Kriterien geprüft. Erfüllt es diese Kriterien nicht, wird die Obergrenze o auf die nächst kleinere Korngrößenklasse gesenkt und das Sampling wird wiederholt. Sobald ein Sample erzeugt wurde, dass die Kriterien erfüllt, kann damit die Simulation gestartet werden.

Die Formulierung von Kriterien, die ein für die Simulation geeignetes Sample garantieren, stellt hier ein Problem dar. Da es nicht möglich ist, mit mathematisch analytischen Methoden die Raumauffüllung einer zufällig dichten polydispersen Kugel-

packung allein aufgrund ihrer Korngrößenverteilung zu bestimmen, ist es ebenfalls nicht möglich vorherzusagen, welche Korngrößenklasse in der Packung dominant sein wird. Diese Informationen erhält man erst durch die Simulation. Ein sehr simples Kriterium, dass bei Testläufen mit verschiedenen Partikelmischung gute Ergebnisse produzierte, überprüft den Größenunterschied zwischen der größten und der kleinsten Kugel im Sample. Unterscheiden sich diese Kugeln um mehr als einen festgelegten Faktor, wird das Sample verworfen. Die Wirksamkeit dieses Kriteriums beruht auf der Tatsache, dass das Volumen eines Partikels in der dritten Potenz des Durchmessers wächst. Eine um den Faktor 50 größere Kugel hätte beispielsweise ein 125000-fach größeres Volumen. Bei einer Packung aus 50000 Kugeln, die sich zum größten Teil aus den kleineren Korngrößen zusammensetzt, könnte also eine einzige, um das 50-fache größere Kugel mehr Volumen aufbringen als der Rest der Packung. Eine simple Beschränkung des Größenunterschiedes zwischen größter und kleinster Kugeln ist geeignet, solch eine Dominanz zu verhindern.

4.2.1 Implementierung

Für die Online-Fraktionierung wird hier ein deterministisches Sampling verwendet, dass im Algorithmus 1 aufgelistet ist. Aber auch die anderen, in Kapitel 3.1.1 aufgelisteten, Samplingverfahren lassen sich in gleicher Weise um eine Obergrenze erweitern.

Algorithmus 1 : Deterministisches Sampling / `determSampling(p_i, o)`

Input : Die Verteilungssummenfunktion der Anzahlanteile: ($Q_0(x)$)

Input : Der Startanteil der aktuellen Fraktion F_i : (p_i)

Input : Die Obergrenze: ($o = 1$)

Input : Die Fraktionsgröße: (s) (Stichprobenumfang)

Output : Ein Vektor aus Kugeldurchmessern: (**K**)

```

1 for  $j = 0$  to  $s - 1$  do
2    $y_j = p_i + (o - p_i) * (j + 0,5) / s$ 
3    $d_j = Q_0^{-1}(y_j)$            // Der zu  $y_j$  passende Durchmesser
4   K  $\leftarrow d_j$ 
5 return K

```

Der Algorithmus unterteilt das zur Verfügung stehende Intervall - also den Bereich zwischen dem Startanteil der aktuellen Fraktion p_i und der Obergrenze o - in gleich große Strata gemäß der gegebenen Fraktionsgröße s (Zeile 2). Die einzelnen Punkte der Stichprobe werden dann jeweils in der Mitte jedes Stratums gezogen und dem Vektor **K** hinzugefügt (Zeile 4).

Die Online-Fraktionierung ist in Algorithmus 2 aufgelistet. Sie besteht aus einer inneren Schleife (Zeilen 7 - 11) und einer äußeren Schleife (Zeilen 3 - 18). Die innere Schleife führt das deterministische Sampling von Algorithmus 1 aus, senkte die Obergrenze (für den nächsten Schleifendurchlauf), prüft das erzeugte Sample und wiederholt die Schritte, falls die Prüfung ein negatives Ergebnis lieferte. Da die Korngrößenverteilung, bedingt durch die verwendeten Messverfahren, stets als diskrete Werte in festgelegten Korngrößenintervallen (Korngrößenklassen) gegeben sind, verwenden wir diese Intervalle als Schrittlänge für die Senkung der Obergrenze des Samplings. Die Korngrößenintervalle sind geordnet in einem Vektor \mathbf{KG} gespeichert, so dass jede obere Intervallgrenze gleich der unteren Intervallgrenze des nächsten Intervalls ist. Die Obergrenze des Samplings wird bei jeder Senkung auf den Anteil Q_0 der oberen Intervallgrenze des nächst kleineren Intervalls gesetzt (Zeile 10).

Algorithmus 2 : Online Fraktionierung

Input : Die Verteilungssummenfunktion der Anzahlanteile: $(Q_0(x))$
Input : Die Anzahl der gegebenen Korngrößen: (k)
Input : Der Vektor der gegebenen Korngrößen: $(\mathbf{KG}[i])$ mit $0 \leq i < k$
Input : Die Fraktionsgröße: (s)
Input : Der Anteil aktiver Kugeln: (a)

```

1  $p_{i+1} = 0$  // Startanteil der nächsten Fraktion  $F_{i+1}$ 
2  $letzteFraktion = \mathbf{false}$ 
3 repeat
4    $p_i = p_{i+1}$  // Startanteil der aktuellen Fraktion  $F_i$ 
5    $o = 1$ 
6    $grenzIndex = k$ 
7   repeat
8      $\mathbf{K} \leftarrow \text{determSampling}(p_i, o)$ 
9      $grenzIndex --$ 
10     $o = Q_0(\mathbf{KG}[grenzIndex])$  // Obergrenze senken
11   until  $\text{pruefSample}(\mathbf{K})$ 
// wenigstens eine Kugel der Stichprobe im letzten KG-Intervall
12   if  $\mathbf{K}[s - 1] \in (\mathbf{KG}[k - 1], \mathbf{KG}[k])$  then
13      $letzteFraktion = \mathbf{true}$ 
14      $p_{i+1} = 1$  // Endanteil der letzten Fraktion
15   else
16      $p_{i+1} = Q_0(\mathbf{K}[s * a])$  // Endanteil der aktuellen Fraktion
17    $\text{simulation}(\mathbf{K})$ 
18 until  $letzteFraktion$ 

```

Die äußere Schleife legt zunächst den Startanteil der aktuellen Fraktion p_i und die Obergrenze o fest, beides wird vom Algorithmus 1 für das Sampling verwendet. Der

Startanteil der aktuellen Fraktion ist 0, wenn es sich um die erste Fraktion handelt. In allen folgenden Fraktionen ist es der Endanteil der aktiven Kugeln der letzten Fraktion (Zeile 4). Die Obergrenze wird auf 1 gesetzt (Zeile 5) und die innere Schleife mit dem Sampling wird gestartet. Nachdem die innere Schleife mit einem gültigen Sample beendet wurde, wird überprüft, ob es sich hier bereits um die letzte Fraktion handelt. Das ist der Fall, wenn die letzte - und damit seltenste - Korngrößenklasse von mindestens einer Kugel repräsentiert wird. Da das deterministische Sampling ein der Größe nach sortiertes Sample liefert, genügt es zu überprüfen, ob die letzte gezogene Korngröße $K[s - 1]$ in der letzten Korngrößenklasse $KG[k - 1]$ liegt. In diesem Fall wird der Endanteil der Fraktion auf 1 gesetzt (Zeile 14), sie enthält dann keine Randkugeln mehr. Das Programm wird dann nach der Simulation dieser letzten Fraktion beendet (Zeile 18). Handelt es sich aber nicht um die letzte Fraktion, wird das Sample gemäß des gegebenen Anteils a in aktive Kugeln und Randkugeln unterteilt. Der Endanteil der Fraktion ist dann der Anteil Q_0 der letzten Korngröße der aktiven Kugeln $K[s * a]$ (Zeile 16).

Um die vom Algorithmus 1 erzeugten Sample überprüfen zu können, gibt es die Funktion `pruefSample()`, deren grundsätzlicher Aufbau in Algorithmus 3 aufgelistet ist. Die Funktion sucht zunächst im Vektor der Korngrößenintervalle das Intervall, in dem das Sample beginnt (Zeilen 2 - 4). Anschließend wird in einer Schleife (Zeile 5 - 12) das gesamte Sample durchlaufen. Dabei wird überprüft, ob jedes Korngrößenintervall, das im Sample enthalten ist, auch tatsächlich durch mindestens eine Kugel repräsentiert wird. Ist ein Intervall nicht im Sample repräsentiert (Zeile 8), wird geprüft, ob es sich um eine Ausfallkörnung handelt. Das ist der Fall wenn der Anzahlanteil q_0 für dieses Intervall 0 ist (Zeile 9). Handelt es sich aber nicht um eine Ausfallkörnung, wird das Sample als ungültig zurückgewiesen.

Da die größten in einem Sample vorkommenden Korngrößenklassen auch die seltensten sind, werden sie in der Regel nur von einigen wenigen Kugeln repräsentiert. Dadurch kann es in diesem Bereich zu Ungenauigkeiten in der Darstellung der gegebenen Korngrößenverteilung kommen. Beispielsweise kann es vorkommen, dass die größte Korngrößenklasse eines Samples von einer einzigen Kugel repräsentiert wird, während die nächst kleinere Korngrößenklasse nicht im Sample vorkommt, obwohl sie einen etwas höheren Anteil hat. Das heißt, diese Korngröße wurde vom Samplingalgorithmus nicht getroffen, da auf die größten Korngrößen nur sehr wenige Kugeln fallen. Bei der Offline-Fraktionierung können wir diese Ungenauigkeit vernachlässigen. Bei den unteren Fraktionen dienen die größten Kugeln nur als Wandkugeln, und wie im Kapitel 2.4.1 gezeigt wurde, konvergiert die lokale Raumauffüllung von kleinen Kugeln an einer Wandkugel gegen den Grenzwert einer glatten Wand. Bei sehr

Algorithmus 3 : Prüfen der Stichprobe / `pruefSample()`

Input : Die Verteilungsdichte der Anzahlanteile ($q_0(x)$)

Input : Die Anzahl der gegebenen Korngrößen: (k)

Input : Der Vektor der gegebenen Korngrößen: ($\mathbf{KG}[i]$) mit $0 \leq i < k$

Input : Die Fraktionsgröße (s)

Input : Der maximal erlaubte Größenunterschied zwischen größter und kleinster Kugel (maxMinKugelRatio)

Input : Die zu testende Stichprobe als Vektor aus Kugeldurchmessern ($\mathbf{K}[j]$) mit $0 \leq j < s$

```
1  $i = 0$ 
2 repeat
3    $i++$  // Sucht das Intervall in dem die Stichprobe beginnt
4 until  $\mathbf{K}[0] \in (\mathbf{KG}[i], \mathbf{KG}[i+1])$ 
5 for  $j = 0$  to  $s - 1$  do
6   if  $\mathbf{K}[j] \in (\mathbf{KG}[i], \mathbf{KG}[i+1])$  then
7      $i++$  // Dieses Intervall enthält wenigstens eine Kugel
8   if  $\mathbf{K}[j] > \mathbf{KG}[i+1]$  then
9     // Dieses Intervall enthält keine Kugel
10    if  $q_0(\mathbf{KG}[i+1]) = 0$  then
11       $i++$  // Dieses Intervall ist eine Ausfallkörnung
12    else
13      return = false // Dieses Intervall ist keine Ausfallkörnung
14 if  $(\mathbf{K}[s-1]/\mathbf{K}[0] > \text{maxMinKugelRatio})$  then
15   return false // Die Stichprobe enthält zu große Kugeln
16 return true
```

großen Wandkugeln ist also ihre genaue Größe vernachlässigbar, nur ihr Vorhandensein ist wichtig. Das in der letzten Fraktion auch die seltenste Korngröße vorhanden ist, wird bei der Art der Berechnung der benötigten Anzahl Fraktionen vorausgesetzt (Gleichung 3.5). Bei der Online-Fraktionierung wird aber die Anzahl der Fraktionen erst im Verlauf der Simulation ermittelt. Die letzte Fraktion wird daran erkannt, dass die letzte Korngrößenklasse der Korngrößenverteilung durch mindestens eine Kugel repräsentiert ist. Es muss also sichergestellt sein, dass auch die größten in einem Sample enthaltenen Korngrößen genau abgebildet werden.

Zuletzt wird noch überprüft ob das Größenverhältnis zwischen größter und kleinster Kugel die vorgegebene Grenze nicht überschreitet (Zeile 13). Diese Grenze kann der Simulation als Parameter übergeben werden, was eine individuelle Anpassung an die gegebene Korngrößenverteilung erlaubt.

4.3 Auswertung

Durch die Online-Fraktionierung ist nun auch eine zuverlässige hierarchische Simulation von Mischungen möglich, bei denen die Packungsstruktur von großen Kugeln dominiert wird. Die Eigenschaften der Simulation, insbesondere ihre Laufzeit, werden dabei aber nicht negativ beeinflusst. Die Tatsache, dass die Fraktionen während des Samplings in ihrer Breite durch eine Obergrenze beschränkt werden, kann zu einer Erhöhung ihrer Anzahl führen. In Tabelle 4.1 sind die Ergebnisse einiger Simulationen aufgelistet, die die Wirkung der Obergrenze verdeutlicht.

Max. KG.Unterschied pro Fraktion	Anzahl der Fraktionen n	Raumausfüllung \pm Standardabweichung [%]	Gesamt- laufzeit [sec]
unbegrenzt	1	67,2311 \pm 1,5807	4457
100	3	69,1188 \pm 0,3376	8579
50	5	68,9059 \pm 0,1654	7090
25	7	68,4423 \pm 0,0532	6583

Tabelle 4.1: Simulationsergebnisse für einen Quarzsand bei Online-Fraktionierung mit unterschiedlichen Einstellungen

Diese Simulationen wurden mit der Verteilung des feinen Quarzsandes aus Abbildung 4.1 (Teil c) durchgeführt. Dafür wurden aber sowohl die feinsten ($< 1,384\mu m$) als auch die größten ($> 213,2\mu m$) Partikel entfernt, so dass sich die Mischung mit nur einem Sample aus 50.000 Kugeln darstellen lässt. Für jede Einstellung wurden 10 Simulationen durchgeführt. Die Ergebnisse zeigen, dass sich durch eine Begrenzung des maximal erlaubten Korngrößenunterschiedes innerhalb der Fraktionen, die Streuung der Simulationsergebnisse drastisch senken lässt. Das geht jedoch zu Lasten der Anzahl der benötigten Fraktionen und erhöht somit auch die Gesamtlaufzeit der Simulation. Den höchsten Zeitaufwand benötigten Simulationen, bei denen der Größenunterschied zwischen größter und kleinster Kugel auf den Faktor 100 begrenzt ist. Senkt man diese Grenze weiter, sinkt die Laufzeit trotz steigender Anzahl Fraktionen. Der Grund dafür liegt im Unterschied zwischen der Entlappung von großen und kleinen Kugeln durch den CR-Algorithmus. Für Kugeln, die im Verhältnis zu den anderen Kugeln der Packung sehr groß sind, ist die Entlappung aufwändig. Aufgrund ihrer Größe ist auch ihre Nachbarschaft, die bei ihrer Entlappung berücksichtigt werden muss, sehr groß. Außerdem können bei ihren Bewegungen innerhalb der Packung immer wieder viele neue Überlappungen entstehen, wodurch es länger dauert, bis sie eine überlappungsfreie Position eingenommen haben. Durch die Begrenzung des Korngrößenunterschiedes entstehen Packungen, die weniger polydispers sind. Die größten Kugeln in diesen Packungen haben eine deutlich kleinere Nachbarschaft, als das ohne die Be-

grenzung der Fall wäre. Der Entlappungsprozess wird dadurch vereinfacht, was zu einer kürzeren Laufzeit führt.

Das Simulationssystem ist jedoch besonders darauf ausgerichtet, sehr umfangreiche Korngrößenverteilungen zu verarbeiten. Dabei werden in der Regel größere Anzahlen von Fraktionen benötigt als dies bei den Simulationen aus Tabelle 4.1 der Fall ist. Die Anzahl der benötigten Fraktionen, sowie deren Lage, für die Simulation der vollständigen Verteilung des Quarzsand ist in Tabelle 4.2 aufgelistet. Sowohl bei der

Frak.- nr.	Offline-Fraktionierung			Online-Fraktionierung		
	von KG [μm]	bis KG [μm]	Randkgl. bis KG [μm]	von KG [μm]	bis KG [μm]	Randkgl. bis KG [μm]
1	0,1224	0,4575	39,589	0,1224	0,4561	2,9174
2	0,4575	0,7435	83,322	0,4561	0,7417	10,605
3	0,7435	1,0768	188,52	0,7417	1,0745	18,382
4	1,0767	1,4764	253,86	1,0745	1,4737	26,783
5	1,4764	1,9801	287,40	1,4737	1,9760	35,762
6	1,9801	2,6492	316,48	1,9760	2,6423	47,589
7	2,6492	3,6136	339,23	2,6423	3,5936	63,171
8	3,6136	5,0279	363,63	3,5936	4,9885	83,747
9	5,0279	7,1543	377,01	4,9885	7,0426	121,71
10	7,1543	10,686	397,74	7,0426	10,388	161,16
11	10,686	18,337	406,03	10,388	17,421	256,84
12	18,337	45,773	409,35	17,421	41,967	409,19
13	45,773	431,59	-	41,967	430,26	-

Tabelle 4.2: Fraktionierung der Korngrößenverteilung eines Quarzsandes. Unterschiedliche Lage der Fraktionen bei Offline- und Online-Fraktionierung.

Offline- als auch bei der Online-Fraktionierung dieser Verteilung werden 13 Fraktionen benötigt. Es gibt also in diesem Fall keine negativen Auswirkungen der Online-Fraktionierung auf das Laufzeitverhalten der Simulation. Betrachtet man die Korngrößen, die jede der Fraktionen umfasst, so fällt auf, dass sich diese Werte zwischen der Offline- und der Online-Fraktionierung nur geringfügig unterscheiden. Deutliche Unterschiede gibt es aber bei den maximalen Korngrößen der Randkugeln jeder Fraktion. Die Begrenzung bei der Online-Fraktionierung wurde so eingestellt, dass hier Kugeln bis maximal der 25-fachen Größe der kleinsten Kugel der Fraktion erlaubt wurden. Die größeren Kugeln, die dadurch in den Fraktionen fehlen, haben aber einen so geringen Anteil, dass dies kaum Auswirkungen auf die Start- und Endanteile der nachfolgenden Fraktionen hat.

Bei der Festlegung einer geeigneten Begrenzung der maximalen Korngrößen ist zu beachten, dass durch das Fehlen größerer Randkugeln die Interaktionen zwischen den Fraktionen reduziert werden. Das betrifft vor allem die Simulation des Wandeffektes.

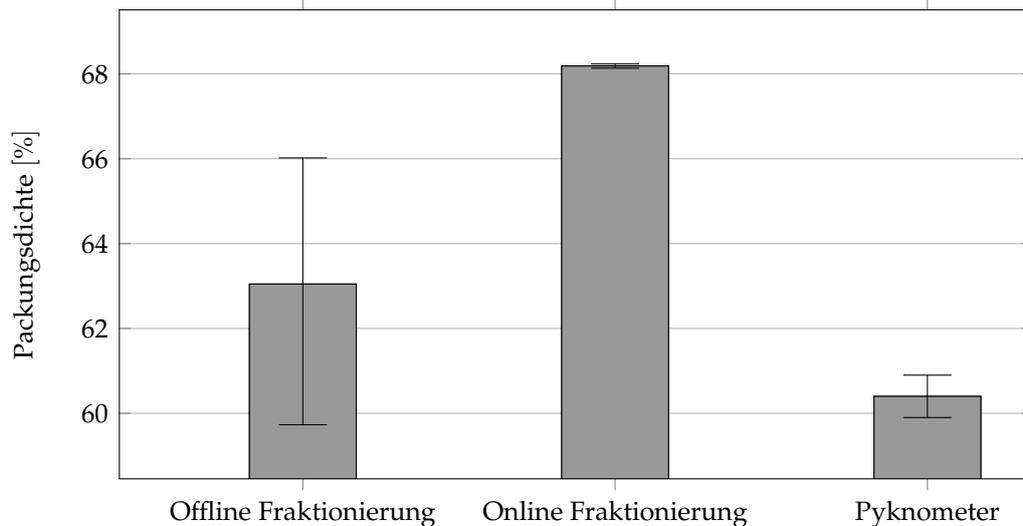


Abbildung 4.7: Packungsdichte eines feinen Quarzsandes mit verschiedenen Methoden ermittelt

Auf eine Begrenzung sollte deshalb bei allen Korngrößenverteilungen verzichtet werden, die sich auch mit der Offline-Fraktionierung problemlos simulieren lassen. Dort wo eine Begrenzung notwendig ist, sollte sie nicht zu klein gewählt werden. Auf der Grundlage des in Kapitel 2.4.1 vorgestellten Modells zur Wirkung des Wandeffektes kann man annehmen, dass Begrenzungen der maximalen Korngröße zwischen dem 25-fachen bis 50-fachen der kleinsten Kugel der Fraktion vertretbar sind. Oberhalb dieser Werte ändert sich die lokale Raumauffüllung nahe der Randkugel nur noch geringfügig.

Die Simulationsergebnisse für die Verteilung des Quarzsandes sind für die unterschiedlichen Fraktionierungsarten in Abbildung 4.7 dargestellt. Jede der Simulationen wurde 10mal wiederholt. Dank der Online-Fraktionierung kann die Simulation eine deutlich höhere Packungsdichte ermitteln. Außerdem unterliegen die Simulationsergebnisse, im Gegensatz zur Offline-Fraktionierung, nur noch einer sehr geringen Streuung. Vergleicht man diese Werte jedoch mit der per Pyknometer gemessenen Packungsdichte, so fällt auf, dass die Simulation den realen Wert in beiden Fällen deutlich überschätzt. Der Grund dafür ist die Kornform des hier verwendeten Quarzsandes. Die Körner sind nicht kugelförmig, wie von der Simulation angenommen wird, stattdessen haben sie eine gebrochene, splittrige Form. Da sie sich ineinander verhaken können, widerstehen sie den Verdichtungsprozessen und sorgen so für eine deutlich geringere Packungsdichte. Wie dieses Verhalten in der Simulation berücksichtigt werden kann, wird in den folgenden Kapiteln dargestellt.

Kapitel 5

Agglomeration

Bei polydispersen Partikelmischungen, deren Korngrößen sich zu erheblichen Teilen im Mikrometerbereich befinden, kann die Agglomeration einen großen Einfluss auf die Packungsdichte haben. Damit die per Simulation ermittelten Ergebnisse eine gute Approximation der realen Verhältnisse in einer Partikelpackung liefern können, muss der Einfluss der Agglomeration berücksichtigt werden. Dies stellt aber ein großes Problem dar, denn die Agglomerate beeinflussen die Struktur und die Dichte der Partikelpackung auf mehrere Arten. Das Wachstum der Agglomerate wiederum wird von vielen verschiedenen Faktoren beeinflusst. Viele wichtige Kenngrößen, die für die Simulation einer agglomerierten Partikelpackung benötigt werden, wie z.B. der Anteil der Agglomerate an der Korngrößenverteilung oder die innere Raumauffüllung der Agglomerate, sind messtechnisch nur schwer oder gar nicht zu ermitteln. Dieses und die folgenden Kapitel zeigen einige der komplexen Zusammenhänge zwischen verschiedenen Partikeleigenschaften, äußeren Einflussfaktoren und dem Wachstum von Agglomeraten. Es wird erklärt, wie die Agglomeration die Korngrößenverteilung der Partikel und das Porenvolumen der Packung verändert. Schließlich werden Wege gezeigt, wie die Auswirkungen der Agglomeration in der Simulation berücksichtigt werden können. Dabei wird insbesondere ein Algorithmus vorgestellt, der die komplexen Vorgänge der Agglomeration auf einfache Weise simuliert. Er ist damit in der Lage, alle von der Packungssimulation benötigten Kenngrößen für die Berechnung einer agglomerierten Partikelpackung, aus wenigen Ausgangsdaten zu approximieren.

Unter Agglomeration versteht man die Zusammenballung kleiner Partikel zu größeren Aggregaten, den sogenannten Agglomeraten. Relevant ist sie vor allem bei Partikeln, deren Größe im Mikrometerbereich und darunter liegt, da die Bindungskräfte in der Regel zu schwach sind, um die Masse größerer Partikel zusammen zu halten. Die folgenden Beispiele zeigen die Vielfältigkeit dieses Phänomens und bieten einen kleinen - wenn auch unvollständigen - Überblick über die Forschungsbereiche, die sich

mit der Agglomeration beschäftigen.

Ein einfaches aus dem Alltag bekanntes Beispiel, das die Wirkungsweise der Agglomeration verdeutlicht, ist die Zusammenballung mikroskopisch kleiner Staubteilchen zu Zentimeter großen Staubflusen, wie man sie häufig beim Staubwischen finden kann. Die Primärpartikel, die diese Flusen bilden, sind nur wenige Mikrometer groß. Aufgrund ihrer geringen Masse genügt schon eine geringe Luftströmung, um sie aufzuwirbeln. Befinden sie sich in der Luft, bewegen sie sich, aufgrund der brownischen Bewegung der Moleküle um sie herum, auf einer chaotischen Bahn während sie langsam auf den Boden sinken. Kollidieren dabei zwei Partikel in einem unelastischen Stoß miteinander, sorgen Kohäsionskräfte an ihrer Oberfläche für ein Anhaften. Auf diese Weise bilden sich immer größere Cluster aus Staubpartikeln, die wir als Staubflusen wahrnehmen.

Die Physik hinter der Entstehung von Agglomeraten bildet die Basis für viele Wachstumsprozesse in der Natur. So spielt sie beispielsweise auch eine wichtige Rolle bei der Planetesimalbildung in der Frühphase des Sonnensystems [Blu06, DBCW07]. Nach dem heutigen Stand der Forschung geht man davon aus, dass sich Planeten aus dem Material einer kosmischen Gas-Staub-Scheibe bilden, die einen Protostern umkreist. Die Partikel dieses Materials sind zu Beginn nur wenige Mikrometer groß. Im Laufe eines Prozesses, der über mehr als 100 Mio. Jahre andauert, verdichten sie sich zu immer größeren Objekten, die schließlich zu Planeten anwachsen. Da die Partikel aber in der Frühphase dieses Wachstumsprozesses eine zu geringe Masse haben, um ein signifikantes Gravitationsfeld auszubilden, kann die Verdichtung des Materials nicht durch die Schwerkraft initiiert werden. Wie man in Experimenten in der Schwerelosigkeit [HSS99] nachweisen konnte, wachsen die ersten größeren Strukturen statt dessen, genau wie bei den Partikeln des Hausstaubes, durch zufällige Kollisionen. Diese Form der Staubagglomeration wird in der Astrophysik als „Koagulation“ [Spe03] bezeichnet.

Die Vergrößerung der Partikel ist ein wesentliches Merkmal der Agglomeration. Durch die Konzentration der Masse sehr vieler kleiner Partikel in wenigen großen Agglomeraten verändert sich das Verhalten von granularen Stoffen. Viele Vorgänge wie z.B. Transport, Ausfiltern oder Abscheiden können so erleichtert werden. In Zyklonabscheidern, die in vielen technischen Anlagen zur Reinigung der Luft oder der Abgase eingesetzt werden, macht man sich dieses Prinzip zu nutze [Anh04]. Bei diesem Verfahren werden flüssige oder feste Partikel aus einem Gasstrom mittels Zentrifugalkraft abgeschieden. Da die Abscheideleistung mit der Größe der Partikel steigt, strebt man eine möglichst hohe Agglomeration bei feinen Partikeln an.

Neben den genannten Beispielen existieren aber auch Anwendungen, bei denen

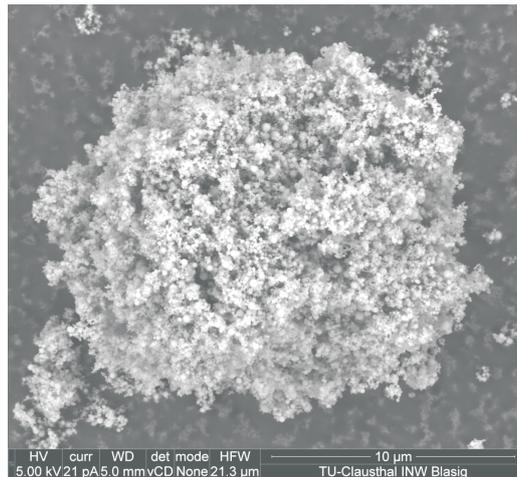


Abbildung 5.1: Agglomerat aus Mikrosilica-Pulver

man bestrebt ist, die Agglomeration zu verhindern oder wenigsten zu verringern. So werden von der pharmazeutischen Industrie inhalierbare Wirkstoffe entwickelt, die aus feinen granularen Pulvern bestehen [WLD⁺03]. Die Korngröße dieser Pulver muss so klein sein, dass sie von den Lungenbläschen aufgenommen werden können. Agglomeration würde hier zu einer Vergrößerung der Partikel führen und ist deshalb unerwünscht. Auch viele Stoffe, die in Betonmischungen verwendet werden, liegen in Korngrößen unterhalb des Mikrometerbereichs vor. Hier kann es ebenfalls zur unerwünschten Agglomeration kommen. Dabei können Strukturen wie in Abbildung 5.1 auftreten. Es handelt sich bei dem Material um ein Pulver aus Siliciumdioxid [BAS09], das bei Mörteln, Klebern, Spachtelmassen und Beton zum Einsatz kommt. Es gilt als gutes Füllmaterial und kann die Dichtigkeit und Festigkeit von Festbeton bzw. Mörtel verbessern. Wegen seiner hohen Feinheit neigt es jedoch zur Agglomeration, was in der Abbildung zu sehen ist. Solche Agglomerate sind unerwünscht, da sie aufgrund ihrer Größe und ihrer oft fraktalartigen Struktur die Eigenschaften des Baustoffes negativ beeinflussen können.

5.1 Ursache und Wirkung

Die Ursache der Agglomeration sind Kohäsionskräfte, die an der Oberfläche der Partikel wirken. Im Detail betrachtet handelt es sich um mehrere Arten von Kräften, die auch in Kombination auftreten können. Je nach ihrer Natur werden die unterschiedlichen Kräfte von verschiedenen Faktoren beeinflusst. Elektrostatische Kräfte entstehen, wenn elektrische Ladungen von außen in die Partikel eingebracht werden. Sie können gezielt eingesetzt werden, um beispielsweise durch unterschiedliche Ladung die

Oberfläche feiner Partikel mit Nanopartikeln zu überziehen [WLD⁺03] oder um durch gleiche Ladung den Agglomerationsvorgang zu hemmen [LWD⁺06]. Allgemein sind jedoch besonders die Van-der-Waals¹ Kräfte von Bedeutung. Sie können grundsätzlich zwischen allen Oberflächen auftreten, ohne dass dafür eine Induktion von außen erforderlich ist. Als Van-der-Waals Kräfte bezeichnet man die zwischen Dipol-Molekülen wirkenden Anziehungskräften, wobei man hier zwischen drei unterschiedlichen Szenarien unterscheiden muss:

1. Die Wechselwirkung zwischen zwei permanenten Dipolen.
2. Die Wechselwirkung zwischen einem permanentem Dipol, der ein entgegengesetztes Dipolmoment in einem nicht polaren Molekül induziert.
3. Die Wechselwirkung zwischen zwei nicht polaren Molekülen, durch gegenseitiges Induzieren von temporären Dipolmomenten.

Die letztere wird als London-Kraft bezeichnet, sie wird oft mit der Van-der-Waals Kraft gleichgesetzt.

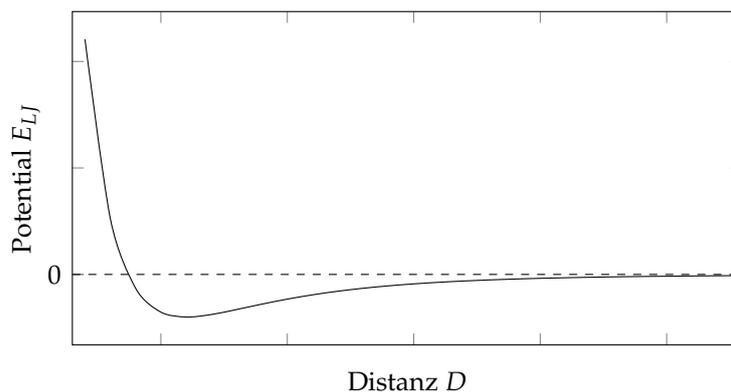


Abbildung 5.2: Partikel-Partikel Interaktionsenergie als Lennard-Jones Potential

Die Berechnung der Anziehungskraft zwischen zwei Partikeln ist nicht trivial, sie hängt von mehreren Faktoren ab. Neben den physikalisch-chemischen Eigenschaften des Stoffes sind dies auch die Größe und die Form der Partikel. Betrachten wir zunächst die mikroskopische Ebene: die Wechselwirkungen zwischen zwei sich annähernden Molekülen können durch ein Lennard-Jones² Potential modelliert werden, wie es in Abbildung 5.2 dargestellt ist. Allgemein hat das Lennard-Jones Potential die Form:

$$E_{LJ} = \frac{C_1}{D^n} - \frac{C_2}{D^6} \quad \text{für } n > 6$$

¹Johannes Diderik van der Waals, 1837 - 1923, niederländischen Physiker

²John Edward Lennard-Jones, 1894 - 1954, englischer Mathematiker und theoretischer Physiker

Es besteht aus einem anziehenden Term $\frac{C_2}{D^6}$ und einem abstoßenden Term $\frac{C_1}{D^n}$. In die Konstanten C_1 und C_2 gehen die spezifischen Materialeigenschaften ein. Das Potential ist abhängig von der Distanz D zwischen den beiden Molekülen. Der abstoßende Term entfaltet seine Wirkung erst, wenn sich die Elektronenwolken der beiden Moleküle sehr nahe kommen. Das Potential steigt dann sehr schnell an, so dass eine abstoßende Kraft entsteht, die verhindert, dass die Moleküle ineinander eindringen können. Zwischen dem abstoßenden und dem anziehenden Teil bildet sich eine Senke im Potential. Entfernt man sich von dieser Senke nach außen, so nimmt die Anziehungskraft im Verhältnis $\frac{1}{D^6}$ zur Distanz ab. Das bedeutet, dass die Van-der-Waals Kräfte nur über sehr kurze Distanzen relevant sind. Über dieses Verhalten auf mikroskopischer Ebene kann man auch auf die Interaktionen zwischen makroskopischen Strukturen schließen. Dort spielt aber auch die Form der Partikeloberfläche eine wichtige Rolle. Zwei Partikel, deren Durchmesser im Nano- oder Mikrometerbereich liegt, können aneinander anhaften, wenn sich ihre Oberflächen nahe genug kommen. Hamaker [HAM37] gibt folgendes vereinfachtes Modell für die Berechnung der Interaktionsenergie zwischen 2 Partikeln an:

$$E = - \int_{V_1} dv_1 \int_{V_2} dv_2 \frac{C}{D^6} \quad (5.1)$$

Dabei stehen V_1 und V_2 für die Gesamtvolumen der beiden Partikel und v_1 bzw. v_2 für einzelne Volumenelemente (induzierte Dipole) innerhalb der jeweiligen Gesamtvolumen. Das Modell geht davon aus, dass sich die Potentiale zwischen den einzelnen paarweise induzierten Dipolen einfach aufsummieren lassen, was aber nicht ganz der Realität entspricht. Die komplexen Interaktionen zwischen den Dipolen innerhalb jeder Partikeloberfläche werden hier außer Acht gelassen. Es wird aber deutlich, dass die resultierende Gesamtenergie E nicht alleine von den Materialeigenschaften abhängt, die mit der Konstanten C ausgedrückt werden, sondern auch davon, wie weit sich die einzelnen Elemente der beiden Volumen V_1 und V_2 annähern können. Bei Partikeln, deren Oberflächen nur geringe Krümmungen aufweisen, können sich verhältnismäßig große Volumenteile sehr nahe kommen. Damit ergibt sich auch eine große Gesamtinteraktionsenergie, was die Agglomeration begünstigt. Umgekehrt können durch eine raue Oberfläche die Haftkräfte deutlich reduziert werden. Abstehende Volumenelemente auf Partikeloberflächen können wie Abstandhalter wirken, die verhindern, dass sich die Hauptanteile der Partikelvolumen zu nahe kommen. Dieses Prinzip ist in Abbildung 5.3 dargestellt: die Kontaktfläche zwischen den beiden Partikeln ist auf die Auswüchse auf ihren Oberflächen beschränkt, nur dort können die Van-der-Waals Kräfte eine relevante Größe erreichen. Die Hauptanteile der beiden Volumen bleiben

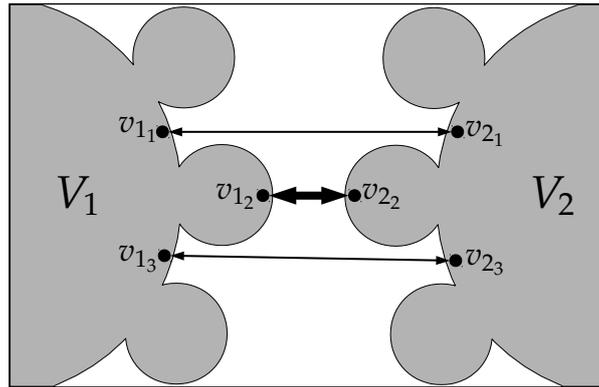


Abbildung 5.3: Verminderung der interpartikulären Kräfte durch Verkleinerung der Kontaktfläche.

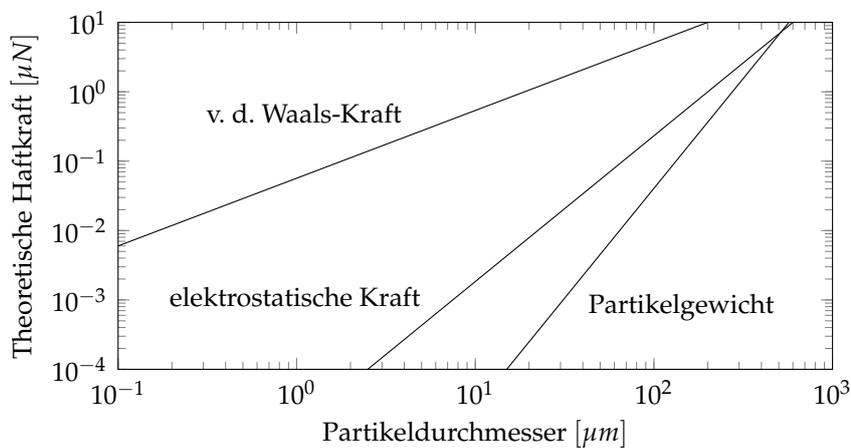


Abbildung 5.4: Vergleich verschiedener interpartikulärer Kräfte in Abhängigkeit zur Partikelgröße nach [BPW⁺91].

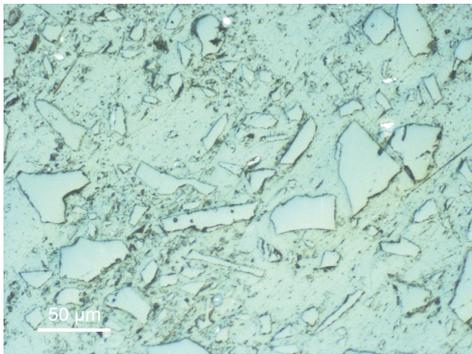
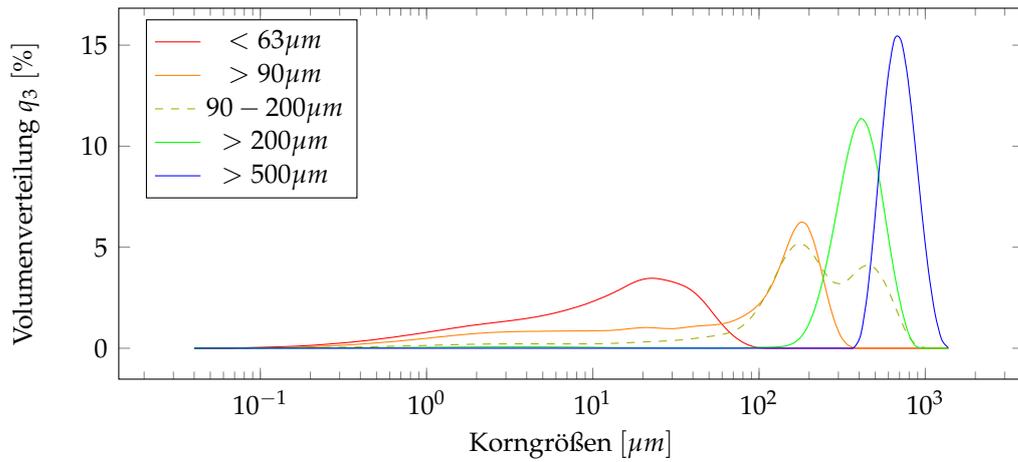
aber zu weit von einander entfernt, so dass ein Anhaften verhindert wird. Man kennt diese Wirkung auch als Lotuseffekt von der Epidermis einiger besonderer Pflanzen. Eine speziell strukturierte Oberfläche mit stark reduzierter Kontaktfläche verhindert ein Anhaften von Wasser und Schmutzpartikeln [NB97]. In [WLD⁺03] wird eine Technik beschrieben, bei der die Oberfläche feiner Partikel von noch kleineren Nanopartikeln überzogen wird, um so die Kontaktfläche zu verkleinern und die Agglomeration zu vermindern. Die Form und Größe der Partikeloberfläche spielt also eine wichtige Rolle für das Agglomerationsverhalten. Lösungen für die Berechnung der Kohäsionskräfte zwischen kugelförmigen Partikeln finden sich in [Bra32, HAM37].

Der Zusammenhang zwischen interpartikulären Haftkräften und Partikeloberfläche ist auch der Grund dafür, dass die Agglomerationsneigung mit zunehmender Größe der Partikel abnimmt. In Abbildung 5.4 ist die Zunahme der theoretischen Haft-

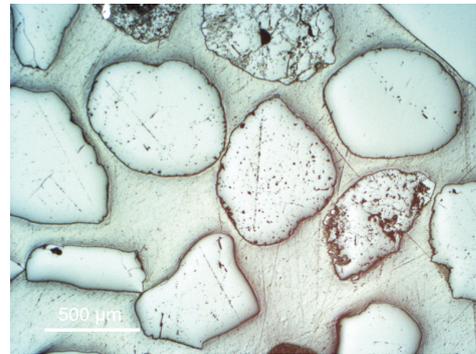
kräfte zwischen 2 Partikeln im Vergleich mit dem Partikelgewicht bei wachsendem Partikeldurchmesser dargestellt. Das Partikelgewicht verhält sich proportional zum Partikelvolumen, welches in der 3ten Potenz zum Partikeldurchmesser wächst. Die Van-der-Waals- und die elektrostatische Kraft hingegen wirken entlang der Partikeloberfläche, die nur in der 2ten Potenz zum Partikeldurchmesser (und damit deutlich langsamer als das Partikelgewicht) anwächst. Da große Partikel eine im Verhältnis zu ihrem Volumen (und damit auch zu ihrem Gewicht) kleinere Oberfläche haben als ähnlich geformte kleine Partikel, dominiert bei ihnen die Schwerkraft über die interpartikulären Haftkräfte. Sie neigen deshalb kaum dazu Agglomerate zu bilden, die laut Valverde und Castellanos [VC06, VC07] vor allem bei Korngrößen unterhalb von $50\mu\text{m}$ auftreten. Die Schwerkraft sorgt innerhalb einer Partikelpackung für deren Verdichtung und erzeugt so weit höhere Packungsdichten als die Van-der-Waals Kräfte, die eher lose Strukturen erzeugen. Auch das Wachstum der Agglomerate ist begrenzt. Überschreitet ein Agglomerat eine bestimmte Größe, reichen die Haftkräfte zwischen seinen einzelnen Partikeln nicht mehr aus, um den auftretenden Scherkräften zu widerstehen. Diese Scherkräfte können sowohl durch das Gewicht der Primärpartikel innerhalb des Agglomerates entstehen als auch durch den Druck anderer Partikel, die innerhalb einer Partikelpackung auf dem Agglomerat lasten. Agglomerate können also auch durch einfache Verdichtungsprozesse zerstört werden.

Neben den bereits beschriebenen Faktoren können auch rein mechanische Kräfte zur Agglomeration feiner Partikel führen. So gibt beispielsweise Fazekas [Faz07] für die Agglomeration von Sand eine Obergrenze von $250\mu\text{m}$ an. Bei diesen Korngrößen hat jedoch das mechanische Verhalten aufgrund der Partikelform einen größeren Einfluss als die Van-der-Waals Kräfte. Die Abbildung 5.5 zeigt Vergleiche zwischen gemessenen und simulativ ermittelten Packungsdichten verschieden feiner Sande sowie die Mikroskopaufnahmen der zugehörigen Kornformen. Hierfür wurde ein Quarzsand mittels verschieden feiner Siebe in 4 Fraktionen zerlegt. Diese Fraktionen sind gemäß der zugehörigen Siebweite mit $< 63\mu\text{m}$, $> 90\mu\text{m}$, $> 200\mu\text{m}$ und $> 500\mu\text{m}$ bezeichnet. Die entsprechenden Korngrößenverteilungen sind im oberen Teil von Abbildung 5.5 dargestellt. Zusätzlich wurden auch die Werte einer Mischung aus zwei Fraktionen ermittelt, sie besteht zu $1/3$ aus Sand der Korngröße $> 200\mu\text{m}$ und zu $2/3$ aus Sand mit $> 90\mu\text{m}$ Korngröße.

Bei Sand mit einer Korngröße von mehr als $500\mu\text{m}$ haben die Partikel eine glatte, runde, teilweise sogar kugelähnliche Form. Das Verhalten der Partikelpackung in der Realität entspricht deshalb fast dem in der Simulation, die beiden ermittelten Packungsdichten weichen nur um 2% voneinander ab. Diese Abweichung ist wohl vor allem durch die von der Kugelgestalt abweichenden Kornform zu erklären. Wie in der



Korngröße: $< 63\mu\text{m}$



Korngröße: $> 500\mu\text{m}$

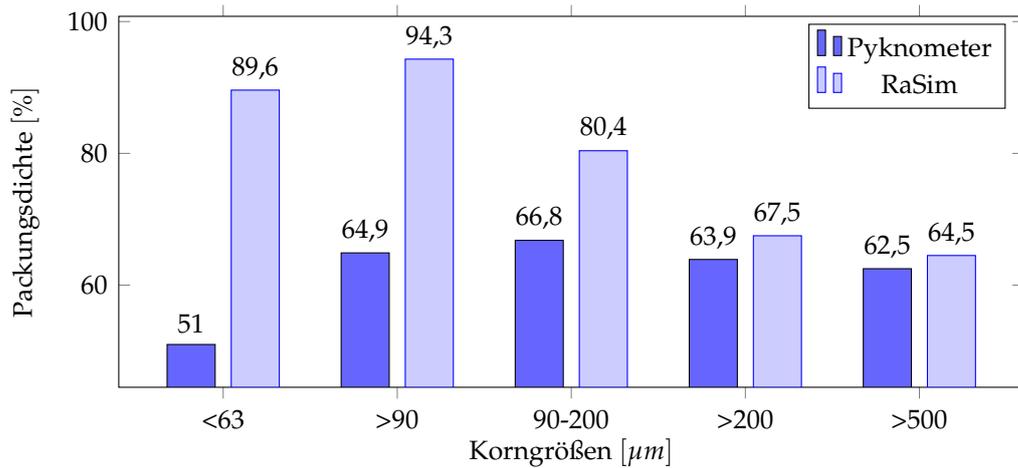


Abbildung 5.5: Verschiedene Sandgemische: Vergleich der Korngrößenverteilungen, der Kornformen und der simulierten und gemessenen Packungsdichten.

zugehörigen Mikroskopaufnahme zu sehen ist, haben einige Partikel eine teilweise konkave Form. Sie können so zusätzliches Porenvolumen in die Packung einbringen (Vergleiche Abbildung 7.1 Form 1 und Form 2). Ganz anders verhält sich hier der Sand mit einer Korngröße von weniger als $63\mu\text{m}$. In der Mikroskopaufnahme sind kaum Partikel mit einer annähernd runden Form zu finden. Die meisten von ihnen haben eine gebrochene, splittrige Form, was den Reibungswiderstand an ihrer Oberfläche stark erhöht. Dadurch sind sie in der Packung weniger beweglich und Verdichtungsprozesse werden erschwert. Zusätzlich können sie sich auch ineinander verhaken und so kleine Agglomerate bilden. Diese Agglomerate sind so stabil, dass sie auch dem von außen eingebrachten Verdichtungsdruck bei der Messung im Pyknometer widerstehen. Die Abweichung zwischen Simulation und Messung beträgt hier mehr als 38%.

Vergleichen wir die Ergebnisse aller 5 Sande im unteren Teil von Abbildung 5.5 so fällt auf, dass bei den letzten 3 Sanden, die vor allem aus größeren Partikeln bestehen, der Trend bei den Mess- und den Simulationsergebnissen gleich ist. Allerdings nimmt die Abweichung zwischen Simulation und Messung deutlich zu, je mehr feine Partikel der Sand enthält. Nimmt man jedoch auch die beiden feinsten Sande mit den Korngrößen $< 63\mu\text{m}$ und $> 90\mu\text{m}$ hinzu, so stimmt auch der Trend zwischen Messung und Simulation nicht mehr überein. Beispielsweise ermittelt die Simulation für den feinsten Sand mit 89,6% die zweithöchste Packungsdichte der Versuchsreihe. Laut Messung hat dieser Sand aber mit 51% die niedrigste Packungsdichte. Die Simulation kann also in ihrer bisherigen Form die realen Verhältnisse in vielen Fällen nicht korrekt wiedergeben.

5.2 Gegenmaßnahmen

Es existieren verschiedene Möglichkeiten, die Agglomeration bei feinen Pulvern zu verhindern oder zumindest zu verringern. In einigen Anwendungen sind Verfahren zur Veränderung der Partikeloberfläche durch Beschichtung mit Nanopartikeln [WLD⁺03] sinnvoll. Das wurde bereits im letzten Kapitel erwähnt. Bei der Verarbeitung von Beton ist man aber letztlich an einer hohen Dichte des fertigen Betonsteins interessiert. In den trockenen Ausgangsstoffen können also durchaus Agglomerate vorhanden sein. Erst bei der Verarbeitung werden diese Agglomerate durch geeignete Maßnahmen zerstört. In einigen Fällen reicht dafür die Dispergierung der Stoffe, da die interpartikulären Haftkräfte in flüssiger Umgebung oft ausreichend reduziert sind, um die Agglomerate zerfallen zu lassen [YXBI⁺08]. Man ist bei dieser Technik nicht alleine auf Wasser beschränkt. Auch durch den Einsatz spezieller Fließmittel [Kje07] können Agglomerate zerstört und so die Dichte der Mischung erhöht werden. Auch durch

die Art der rein mechanischen Verarbeitung der Betonmischung können Agglomerate deutlich reduziert werden. Beispielsweise neigen Mikrosilicapulver [BAS09] wegen ihrer Feinheit stark zur Agglomeration, was durch lange Lagerzeiten noch begünstigt wird. Laut Angaben der Hersteller sorgt eine ausreichend lange Gesamtmischzeit bei Mikrosilica-Betonen und gegebenenfalls eine trockene Vormischung zusammen mit der groben Gesteinskörnung bei Trockenmörteln dafür, dass die Agglomerate weitestgehend zerrieben werden.

Trotz all dieser Gegenmaßnahmen kann man in der Regel nicht von einer völlig agglomeratfreien Mischung ausgehen. Auch kleine Agglomerate sorgen für eine anders strukturierte Packung als dies in der Simulation der Fall ist. Die Agglomeration oder zumindest ihre Auswirkung muss also in geeigneter Form in der Simulation berücksichtigt werden, wenn diese realistische Ergebnisse liefern soll.

5.3 Berücksichtigung in der Simulation

Einige Strategien, um die Wirkung der Agglomeration in der Simulation zu berücksichtigen, wurden bereits von Raschdorf in seiner Dissertation [Ras10] dargestellt. Eine interessante und naheliegende Möglichkeit ist die Ermittlung eines Faktors a , der das fertige Simulationsergebnis korrigiert:

$$\Phi_{Exp} = a\Phi_{Sim}$$

Der Faktor a ist dabei abhängig vom Feinstoffanteil in der untersuchten Mischung und zerfällt deshalb in die Komponenten:

$$a = b - c \cdot f, \quad \text{mit } f = \int_{x=0\mu m}^{100\mu m} q_0(x) dx$$

In b sind dabei verschiedene Einflussfaktoren berücksichtigt, die zu einer grundsätzlichen Abweichung zwischen Experiment und Simulation führen, wie etwa die Partikelform. Der Faktor c gibt alleine an, wie stark die Raumauffüllung durch den Anteil von Partikeln unter $100\mu m$ in der Mischung gesenkt wird. Raschdorf ermittelte die beiden Faktoren b und c durch Regression. Er verglich dazu die real gemessenen Raumauffüllungen einer Reihe von Mörtelmischungen mit unterschiedlichem Feinstkornanteil mit den zugehörigen Simulationsergebnissen. Sind die Faktoren einmal ermittelt, können sie verwendet werden, um die Raumauffüllung anderer Mischungen mit der selben Stoffzusammensetzung aber anderen Anteilen zu bestimmen. Diese Vorgehensweise ist deshalb interessant, weil sie nicht in die Simulation und deren Parameter eingreift.

Sie ist sogar völlig unabhängig von der Art der Simulation, da sie nur das Ergebnis verwendet. Allerdings setzt sie voraus, dass für jede zu untersuchende Stoffzusammensetzung eine ausreichend große Datenbasis von Messergebnissen vorhanden ist, um die jeweiligen Faktoren bestimmen zu können. Da in modernen Betonmischungen eine Vielzahl unterschiedlicher Stoffe zum Einsatz kommen können, und die experimentelle Messung der Raumausfüllung mit erheblichem Aufwand verbunden ist, stehen zu Beginn der Entwicklung neuer Rezepturen oft nur wenige Daten zur Verfügung. Raschdorf zeigte deshalb einen Weg auf, um die Agglomeration direkt in der Simulation bzw. in ihren Parametern zu berücksichtigen.

Betrachten wir, welche Auswirkungen die Agglomeration auf die Struktur einer Partikelpackung hat, so lassen sich deutlich zwei Wirkungsweisen unterscheiden, die in Abbildung 5.6 dargestellt sind. Die linke Seite zeigt die Verteilungsdichte q_0 einer nicht agglomerierten Mischung, die einen gewissen Anteil zur Agglomeration neigenden Feinstkorn enthält. Dieser Anteil, den wir mit g_0 bezeichnen, verschwindet aus der Verteilung, die darin enthaltenen Partikel schließen sich zu Agglomeraten zusammen, die dann als zusätzlicher Anteil in einem höheren Korngrößenbereich erscheinen. Als Ergebnis dieses Vorgangs erhalten wir die Verteilungsdichte \hat{q}_0 , die einen geringeren Anteil Feinstkorn dafür aber einen gewissen Anteil an Agglomeraten, den wir mit h_0 bezeichnen, enthält. Ihr Schwerpunkt ist deshalb im Vergleich zur Ausgangsverteilung q_0 deutlich zu den höheren Korngrößen verschoben.

Allein die veränderte Korngrößenverteilung kann die Raumausfüllung einer Packung senken: da der Anteil der kleinsten Partikel an der Gesamtanzahl gesunken ist, reicht ihre Anzahl eventuell nicht mehr aus, um die verbleibenden Hohlräume zwischen den großen Partikeln zu füllen. Hinzu kommt, dass die Agglomerate andere Eigenschaften haben als Primärpartikel der gleichen Größe. Durch ihren Aufbau enthalten sie viele Hohlräume und bringen damit zusätzliches Porenvolumen in die Packung ein, wodurch sie deren Raumausfüllung senken.

Raschdorf beschreibt in seiner Arbeit, wie aus der mittels \hat{q}_0 simulativ ermittelten

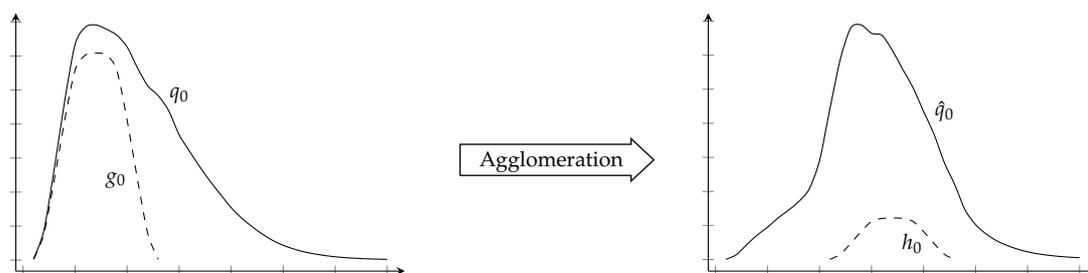


Abbildung 5.6: Auswirkung der Agglomeration auf eine Korngrößenverteilung nach [Ras10]

Raumausfüllung sowie dem Agglomerateanteil h_0 und dem Feststoffanteil der Agglomerate ρ die Raumausfüllung einer Packung mit Agglomeraten Φ_A ermittelt werden kann. Diese von ihm vorgeschlagene Methode ist jedoch nicht praxistauglich, da sie einige Fragen unbeantwortet lässt. So konnte er die ungefähre Lage von h_0 nur durch die Auswertung von Experimenten schätzen, was einen sehr hohen Aufwand bedeutet. Ebenso musste unklar bleiben, wie der Feststoffanteil der Agglomerate ρ , den wir auch als deren innere Raumausfüllung ansehen können, bestimmt werden kann.

Im Rahmen der hier vorliegenden Arbeit wurde eine Simulation entwickelt, die die Verteilung \hat{q}_0 und den in ihr enthalten Agglomerateanteil h_0 mit Hilfe eines iterativen Prozesses bestimmt. Dabei kommen spezielle Algorithmen zum Einsatz, die das Wachstum der Agglomerate aus den Primärpartikeln virtuell nachbilden. Zu den Vorteilen dieses Verfahrens gehört, dass es nur sehr wenige Ausgangsdaten benötigt. Darüber hinaus liefert es auch die Strukturen der erzeugten Agglomerate als 3D-Modelle, was genauere Untersuchungen, und insbesondere auch eine Abschätzung von ρ , ermöglicht.

5.4 Simulative Bestimmung der agglomerierten Korngrößenverteilung

Erste grundlegende Arbeiten zur mathematischen Beschreibung von Aggregationen wurden in der ersten Hälfte des letzten Jahrhunderts von Smoluchowski [Smo17] entwickelt. Von ihm stammt eine Koagulationsgleichung [Smo16], die die Entstehung und das Weiterwachsen von Clustern aus Primärpartikeln modelliert.

$$\partial_t f(x) = \frac{1}{2} \int_0^x a(y, x-y) f(y) f(x-y) dy - \int_0^\infty a(x, y) f(x) f(y) dy$$

Sie beschreibt das Verhalten einer Korngrößenverteilung im Verlauf einer Koagulation über die Zeit t . Man geht dabei davon aus, dass sich jeweils genau zwei Partikel (Primärpartikel oder Cluster) zu einem größeren Cluster verbinden. Im einfachsten Fall sind die Partikel vollständig durch ihr Volumen beschrieben. Die Verteilungsfunktion $f = f(t, x) \geq 0$ gibt also den Anteil von Partikeln mit Volumen $x > 0$ an der Gesamtanzahl aller Partikel zum Zeitpunkt $t \geq 0$ an. Das erste Integral der Gleichung berechnet den Anteil aller Partikel mit Volumen x , die aus der Verbindung zweier Partikel mit Volumen y und $x - y$ mit $y \in (0, x)$ hervorgehen. Der Koeffizient $a = a(y, x - y)$ gibt dabei die Wahrscheinlichkeit an, mit der sich Partikel mit Volumen y und $x - y$ miteinander verbinden. Das zweite Integral berechnet den Anteil aller Partikel mit Volumen x , die sich mit anderen Partikeln verbinden und so größere Partikel bilden. Analyti-

sche Lösungen dieser Gleichung existieren nur für einige Spezialfälle, in der Regel ist man aber auf numerische Simulationen [FL04] angewiesen.

Die Koagulationsgleichung von Smoluchowski beruht auf einem wichtigen Prinzip: das Nettovolumen, also das Volumen aller Partikel ohne Hohlräume, bleibt während des gesamten Wachstumsprozesses gleich. Kennt man zu jedem Zeitpunkt die Anzahl von (Primär-) Partikeln, die sich jeweils zu einem größeren (Agglomerate-) Partikel verbinden, dann kann man auch die Anzahl der entstehenden (Agglomerate-) Partikel berechnen. Haben wir also eine Korngrößenverteilung für nicht agglomerierte Primärpartikel q_0 gegebene und kennen wir außerdem die Agglomerationsneigung der Partikel $a = a(x) \in (0, 1)$, dann lässt sich der Anteil der entstehenden Agglomerate berechnen wenn wir wissen, wie viele Primärpartikel einer Größe sich zu einer bestimmten Agglomeratgröße verbinden. Wissen wir beispielsweise, dass sich m Partikel der Größe x zu jeweils einem Agglomerat der Größe y verbinden, dann beträgt der entsprechende Agglomerateanteil:

$$h_0(y) = \frac{1}{m} g_0(x), \quad \text{mit } g_0(x) = q_0(x) \cdot a(x)$$

Basierend auf diesen Überlegungen wurde das Simulationssystem PSD-Morph (PSD = engl. particle size distribution) entwickelt, dessen Prinzip in Abbildung 5.7 dargestellt ist. Die Simulation erzeugt in einem iterativen Prozess aus einer Verteilung von nicht agglomerierten Primärpartikeln (in der Abbildung blau dargestellt) eine stabile, nicht weiter agglomerierende Verteilung, die sowohl Anteile aus Primärpartikeln als auch aus stabilen Agglomeraten (rot dargestellt) enthält. Die Menge aller während eines Simulationsschrittes erzeugten Agglomerate bezeichnen wir als Agglomeratengeneration oder kurz Generation. Wie setzen voraus, dass die gegebene Korngrößenverteilung in diskrete Intervalle I_j , mit $j = 1, \dots, K$, zerlegt ist. In der Praxis ist dies, bedingt durch die Messverfahren zur Bestimmung der Korngrößenverteilung, in der Regel immer gegeben.

Die Simulation ermittelt zunächst den Anteil der zur Agglomeration neigenden Primärpartikel im ersten Intervall I_1 (in der Abbildung grau dargestellt). Im ersten Simulationsschritt werden diese Partikel zu Agglomeraten zusammengesetzt. Dabei gilt stets: das Wachstum eines Agglomerates stoppt, sobald seine Größe im nächst höheren Intervall liegt. Die so entstandenen Agglomeratpartikel bilden die Generation 1. In allen folgenden Simulationsschritten wird dieser Vorgang in gleicher Weise fortgesetzt. Es wird der Anteil der zur Agglomeration neigenden Primärpartikel im kleinsten der verbleibenden (noch nicht agglomerierten) Intervalle ermittelt, um sie anschließend zur nächsten Agglomeratengeneration zu formieren. Im Gegensatz zum ersten Simula-

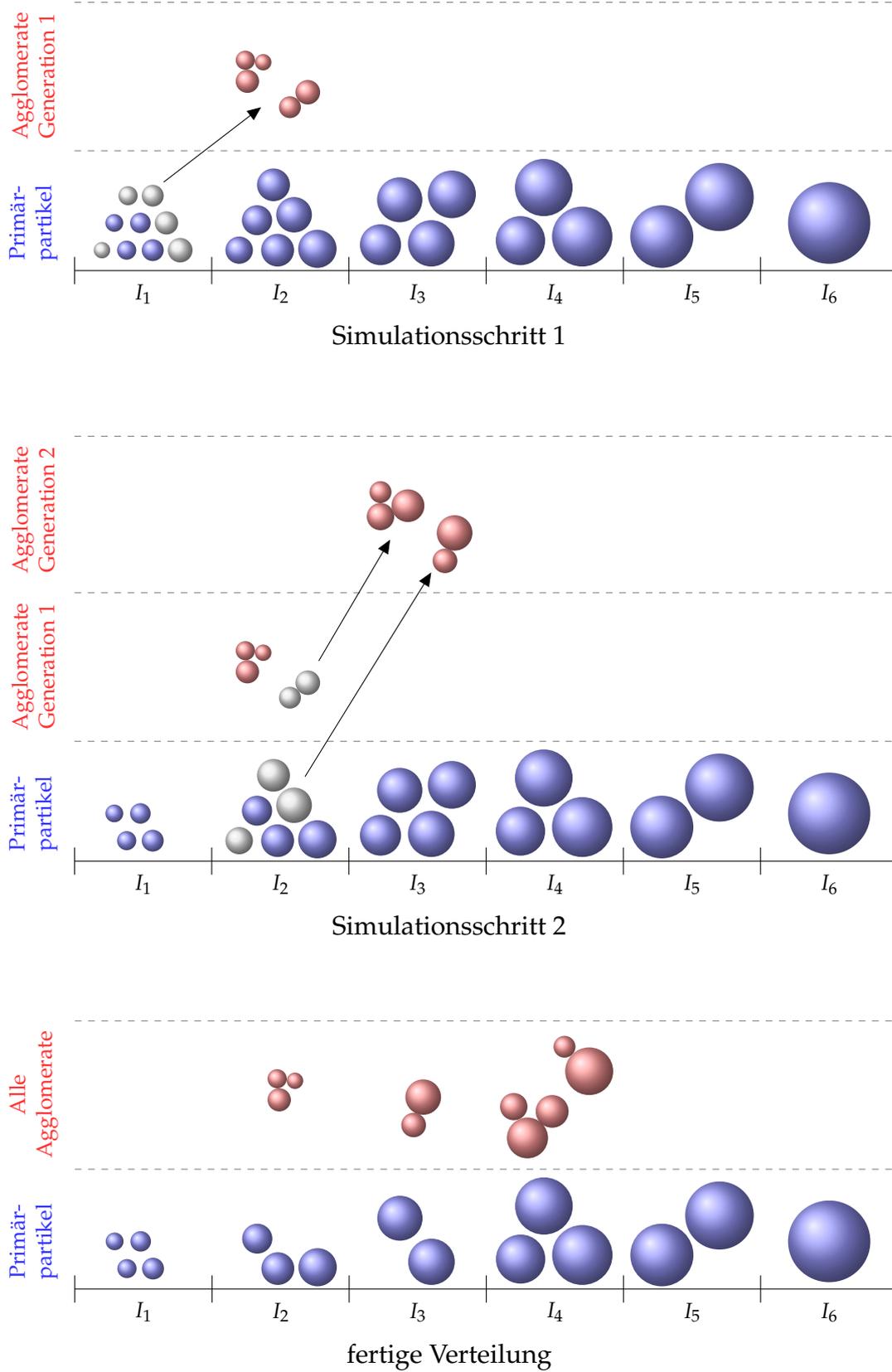


Abbildung 5.7: Prinzip des PSD-Morph Algorithmus

tionsschritt ist hier aber zu berücksichtigen, dass die Agglomerate der eben erzeugten Generation nicht stabil sein müssen, ein Teil von ihnen könnte noch weiter wachsen. Dieser Anteil wird bestimmt und die entsprechenden Agglomerate werden aus der Generation wieder entfernt. Die Primärpartikel dieser Agglomerate werden dann zu den bereits bestimmten agglomerierenden Primärpartikeln hinzugefügt, um aus ihnen gemeinsam die nächste Agglomerate-Generation zu erzeugen.

Die Simulation ist beendet, wenn keine weiteren zur Agglomeration neigenden Primärpartikel mehr in der Verteilung vorhanden sind und ebenso keine weiter wachsenden Agglomerate mehr. Die Partikel sind hier nicht vollständig durch ihr Volumen beschrieben, zwei weitere wichtige Faktoren sind ihre Größe und ihre Form. Für die Primärpartikel gehen wir von einer Kugelform aus. Die aus ihnen geformten Agglomerate können jedoch sehr komplexe, fraktalartige Formen haben. Es ist deshalb nicht möglich, auf rein analytischem Weg zu bestimmen, wie sich die Größe eines Agglomerates ändert, wenn ein weiteres Primärpartikel hinzugefügt wird. Denn das ist abhängig von der Oberflächenstruktur des Agglomerates und dem Punkt auf seiner Oberfläche, an dem das Primärpartikel andocken wird. Beides wird vom Zufall bestimmt. Um dieses zufällige Anwachsen simulieren zu können, setzen wir einen „Agglomerategenerator“ ein. Dieser Generator implementiert ein geeignetes Modell, das die physikalischen Vorgänge, die zur Agglomeration führen, virtuell nachbildet. Damit ist es möglich, eine repräsentative Stichprobe von Agglomeraten zu erzeugen, aus denen dann die benötigten Daten gewonnen werden können. Aus der Sicht des Simulationssystems PSD-Morph ist es nicht relevant, mit welchem Modell die Agglomerate erzeugt werden. Es ist deshalb auch möglich, unterschiedliche Modelle einzusetzen, um beispielsweise die Simulation an andere Anwendungsbereiche anzupassen. Wir benutzen hier einen DLA-Algorithmus (engl. diffusion limited aggregation), um das Wachstum der Agglomerate zu simulieren. Dieser Algorithmus sowie einige Alternativen werden in einem späteren Kapitel beschrieben.

5.4.1 Ausgangsdaten für die Simulation

Für unser Simulationssystem benötigen wir folgende Ausgangsdaten:

- $[0, S] \subset \mathbb{R}$, Intervall der vorhandenen Korngrößen (Primärpartikeldurchmesser).
- $q : [0, S] \rightarrow \mathbb{R}_+$, die **Korngrößenverteilung** der Primärpartikel in Anzahlanteilen, es gilt: $\int_{[0,S]} q(x) dx = 1$.
- $a : [0, S] \rightarrow [0, 1]$, $a(x)$ ist die **Agglomerationsneigung** eines Primärpartikels der Größe x .

- $b : [0, S] \rightarrow [0, 1]$, $b(x)$ ist die **Weiter-Agglomerationsneigung** eines (Teil-)Agglomerates der Größe x .

Ferner eine **Diskretisierung** in Intervalle:

- $0 = s_0 < s_1 < \dots < s_{K-1} < s_K = S$
- $I_j := [s_{j-1}, s_j)$, $j = 1, \dots, K$
- q, a und b sind konstant auf den Intervallen I_j :

$$q(x) \equiv: \tilde{q}(i), \quad a(x) \equiv: \tilde{a}(i), \quad b(x) \equiv: \tilde{b}(i) \quad \text{für } x \in I_i, i = 1, \dots, K.$$

Die Diskretisierung ist eine wichtige Voraussetzung für die Simulation. Gleichzeitig ist sie aber auch durch das Format der gemessenen Werte der Korngrößenverteilung vorgegeben. Sie ist Ausdruck der begrenzten Auflösung der verwendeten Messverfahren, die nicht jede tatsächlich vorhandene Korngröße erfassen können, sondern nur die Anteile festgelegter Intervalle an der Gesamtmenge. Die tatsächliche in jedem Intervall I_i vorhandene Verteilung ist, bedingt durch die Auflösung des Messverfahrens, nicht bekannt.

Darüber hinaus setzen wir voraus, dass ein **Agglomerategenerator** existiert, der gemäß der vorgegebenen Parameter virtuelle Agglomerate erzeugen kann, indem er den natürlichen Wachstumsprozess simuliert. Die von diesem Generator erzeugten Agglomerate können sehr komplexe Formen haben. Um sie in unser Simulationsmodell einbinden zu können, abstrahieren wir sie als Kugeln. Wie wir die Größe dieser Kugeln und die Weiteragglomerationsneigung $b(x)$ der Agglomerate bestimmen können, wird in Kapitel 5.4.5 erklärt.

5.4.2 Bestimmung der agglomerierenden Primärpartikel

Mit Hilfe der gegebenen Ausgangsdaten definieren wir iterativ die Funktionen $g_i, h_i : [0, S] \rightarrow \mathbb{R}_+$. Dabei beschreibt $g_i(x)$ die Anzahlanteile der in Iteration i zur Agglomeration zur Verfügung stehenden Primärpartikel des Durchmessers x .

$$g_1(x) = a(x) \cdot q(x) = \tilde{a}(1) \cdot q(x), \quad x \in [0, s_1) = I_1$$

$$g_{i+1}(x) = \begin{cases} b(x) \cdot g_i(x) = \tilde{b}(i+1) \cdot g_i(x) & \text{mit } x \in [0, s_i) = I_1 \cup \dots \cup I_i \\ a(x) \cdot q(x) = \tilde{a}(i+1) \cdot q(x) & \text{mit } x \in [s_i, s_{i+1}) = I_{i+1} \end{cases}$$

Im ersten Iterationsschritt stehen also alle in I_1 agglomerierenden Primärpartikel zur Verfügung. In späteren Schritten stehen neben den agglomerierenden Primärpartikeln

des aktuellen Intervalls, auch Primärpartikel aus früheren Intervallen zur Verfügung. Es handelt sich dabei um die Primärpartikel, die in den weiter wachsenden Agglomeraten des aktuellen Intervalls enthalten sind. Für $g_i(x)$ ergibt sich somit folgende Rekursion:

$$g_i(x) = \begin{cases} \tilde{a}(i)q(x) & \text{für } x \in [s_{i-1}, s_i) = I_i \\ \tilde{b}(i)\tilde{a}(i-1)q(x) & \text{für } x \in [s_{i-2}, s_{i-1}) = I_{i-1} \\ \tilde{b}(i)\tilde{b}(i-1)\tilde{a}(i-2)q(x) & \text{für } x \in [s_{i-3}, s_{i-2}) = I_{i-2} \\ \dots & \\ \tilde{b}(i)\tilde{b}(i-1)\dots\tilde{b}(3)\tilde{a}(2)q(x) & \text{für } x \in [s_1, s_2) = I_2 \\ \tilde{b}(i)\tilde{b}(i-1)\dots\tilde{b}(2)\tilde{a}(1)q(x) & \text{für } x \in [s_0, s_1) = I_1 \end{cases} \quad (5.2)$$

Das bedeutet, der Anteil der kleineren Primärpartikel in den großen Agglomeraten sinkt umso mehr, je mehr von ihnen bereits in stabilen (nicht weiter wachsenden) kleineren Agglomeraten gebunden sind.

Es sei

$$K_0 := \min \{k | 1 \leq k \leq K, \tilde{a}(k') = \tilde{b}(k') = 0 \text{ für alle } k' = k + 1, \dots, K\}$$

der größte Index, für den beide Agglomerationsneigungsfunktionen $\neq 0$ sind. Dann gilt offensichtlich $g_i \equiv 0$ für $i > K_0$. Die Simulation muss also für alle $i = 1, \dots, K_0$ durchgeführt werden. Für $i > K_0$ findet keine Agglomeration mehr statt.

5.4.3 Bestimmung der Anteile der Agglomerate

Um aus den ermittelten agglomerierenden Primärpartikelanteilen $g_i(x)$, $x \in [0, s_i)$ die Anteile der daraus entstehenden Agglomerate bestimmen zu können, benötigen wir den Agglomerategenerator. Dazu wird zunächst aus $g_i(x)$ mittels Normierung eine Wahrscheinlichkeitsverteilung auf $[0, s_i)$ ermittelt:

$$G_i(x) = \frac{g_i(x)}{\int_{[0, s_i)} g_i(x') dx'}, \quad x \in [0, s_i)$$

Anschließend können Kugeln als Primärpartikel erzeugt werden, deren Durchmesser der Verteilung $G_i(x)$ folgen. Diese Primärpartikel fügt dann der Generator zu Agglomeraten zusammen. Er lässt dabei jedes Agglomerat solange wachsen, bis seine Größe $> s_i$ ist, das heißt, es verlässt das aktuelle Intervall nach oben. Die so entstehende Agglomerategeneration muss einen ausreichend großen Umfang erreichen, um für fol-

gende Eigenschaften stabile Werte zu liefern:

- $d(i) :=$ Index des Intervalls in das die mittlere Agglomeratgröße fällt
- $m(i) :=$ mittlere Anzahl an Primärpartikeln die ein Agglomerat enthält
- $\rho(i) :=$ mittlere innere Raumauffüllung der Agglomerate

Wobei $\rho(i)$ nicht für die Berechnungen innerhalb der PSD-Morph Simulation gebraucht wird. Es wird aber später für die simulative Bestimmung der Packungsdichte der agglomerierten Partikelmischung benötigt (siehe Kapitel 5.6.1). Beträgt die Anzahl der erzeugten Agglomerate n und sind die beobachteten Durchmesser d_1, \dots, d_n , die Anzahl Primärpartikel m_1, \dots, m_n und die inneren Raumauffüllungen ρ_1, \dots, ρ_n , dann können die entsprechenden Werten mit:

$$d(i) = \frac{1}{n} \sum_{i=1}^n d_i, \quad m(i) = \frac{1}{n} \sum_{i=1}^n m_i \quad \text{und} \quad \rho(i) = \frac{1}{n} \sum_{i=1}^n \rho_i$$

berechnet werden. Wie eine ausreichend hohe Anzahl Agglomerate n festgelegt werden kann, wird in Kapitel 5.4.5 unter „Festlegung der Anzahl der Agglomerate pro Generation“ erklärt.

Mit diesen Daten können dann die Anteile der Agglomerate bestimmt werden. Wenn die Partikelmischung ursprünglich aus M Primärpartikeln bestand, dann stehen dem Generator im Simulationsschritt i :

$$M \cdot \int_{[0, s_i)} g_i(x) dx = M \cdot \sum_{j=1}^i \prod_{l=j+1}^i b(l) a(j) \int_{[s_{j-1}, s_j)} q(x) dx$$

Partikel zur Verfügung. Enthalten die daraus entstandenen Agglomerate im Mittel $m(i)$ Primärpartikel, so können daraus im Mittel:

$$\frac{M \cdot \int_{[0, s_i)} g_i(x) dx}{m(i)} \tag{5.3}$$

Agglomerate gebildet werden. Ihr Anteil an der Anzahl ursprünglicher Primärpartikel beträgt also:

$$\frac{\int_{[0, s_i)} g_i(x) dx}{m(i)} \tag{5.4}$$

Diese Agglomerate haben eine mittlere Größe die in $I_{d(i)}$ liegt. Mit Hilfe dieser Daten

kann nun die Agglomerate-Anteilsfunktion h_i aufgebaut werden:

$$h_1(x) = 0, \quad x \in [0, S] \quad (5.5)$$

$$h_{i+1}(x) = \begin{cases} h_i(x) & \text{falls } x \in [0, s_i) = I_1 \cup \dots \cup I_i \\ (1 - \tilde{b}(i+1)) \cdot h_i(x) & \text{falls } x \in I_{i+1} = [s_i, s_{i+1}) \\ h_i(x) + \frac{\int_{[0, s_i)} g_i(y) dy}{m(i+1)} & \text{falls } x \in I_{d(i+1)} \end{cases} \quad (5.6)$$

Der Anteil der Agglomerate mit Durchmesser x (an der Gesamtanzahl der ursprünglichen Primärpartikel) zu Beginn der i -ten Iteration wird mit $h_i(x)$ bezeichnet. Nach der Ausführung des Simulationsschrittes besteht der Agglomerateanteil $h_{i+1}(x)$ aus den Agglomerateanteilen aus früheren Intervallen $h_i(x)$. Diese sind bereits aus früheren Simulationsschritten als stabile Agglomerate übrig geblieben und werden deshalb nicht mehr verändert. Im aktuellen Intervall verbleibt der Anteil $(1 - \tilde{b}(i+1))h_i(x)$ in stabilen nicht weiter wachsenden Agglomeraten. Die restlichen $\tilde{b}(i+1)h_i(x)$ wachsen weiter, wobei sie nun auch Primärpartikel aus dem aktuellen Intervall I_{i+1} enthalten können. Diese Agglomerate werden wieder solange wachsen, bis sie das aktuelle Intervall nach oben verlassen. Ihre Größe wird dann im Mittel im Intervall $I_{d(i+1)}$ liegen.

Da mit dem Fortschreiten der Iterationen in $g_i(x)$ immer größere Primärpartikel zur Agglomeration zur Verfügung stehen, würde in den weiter wachsenden Agglomeraten eine unnatürliche Sortierung der Primärpartikelgrößen stattfinden. Im Zentrum der Agglomerate würden sich nur die kleinsten Primärpartikel befinden, während die größeren weiter außen eingebaut würden. Der Agglomerategenerator erzeugt die Agglomerate deshalb für jeden Iterationsschritt neu. Die verschiedenen Primärpartikelgrößen werden so, bezüglich der Reihenfolge ihres Hinzufügens zum Agglomerat, alle gleich behandelt. Außerdem wird so sichergestellt, dass die Primärpartikel jeder Agglomerategeneration auch tatsächlich der für ihren Iterationsschritt berechneten Verteilung $G_i(x)$ folgen.

5.4.4 Endergebnis: die stabilen Anteile

Wurde die Simulation für alle $i = 1, \dots, K_0$ durchgeführt, findet keine weitere Agglomeration mehr statt. In

$$(1 - a(x))q(x), \quad x \in [0, S]$$

sind dann nur noch die Anteile der stabilen nicht agglomerierenden Primärpartikel enthalten. Die restlichen $a(x)q(x)$ sind nun in stabilen nicht weiter wachsenden Agglomeraten gebunden, deren Anteile mit h_{K_0} beschrieben werden. Damit lässt sich die

Funktion q' zusammensetzen:

$$q'(x) := (1 - a(x)) \cdot q(x) + h_{K_0}(x), \quad x \in [0, S) \quad (5.7)$$

Die Anzahl der stabilen Agglomerate ist geringer als die Anzahl der Primärpartikel aus denen sie entstanden sind. Ausgedrückt wird dies durch die Division durch $m(i)$ in den Formeln 5.3 und 5.4. Dadurch ist auch die Gesamtanzahl der Partikel gesunken, was aber bei der Berechnung von h_{K_0} und q' nicht berücksichtigt wurde. So beschreibt q' keine Dichtefunktion der agglomerierten Korngrößenverteilung. Wir setzen deshalb q' und die darin enthaltene h_{K_0} durch Normierung in \hat{q} und h um:

$$\hat{q}(x) = \frac{q'(x)}{\int_{[0,S)} q'(x') dx'}, \quad x \in [0, S)$$

$$h(x) = \frac{h_{K_0}(x)}{\int_{[0,S)} q'(x') dx'}, \quad x \in [0, S)$$

Mit $\hat{q}(x)$ werden nun die Anzahl-Anteile der Korngrößen der stabilen agglomerierten Partikelmischung beschrieben, die sowohl aus Primärpartikeln als auch aus Agglomeraten besteht. Die Anteile der in ihr enthaltenen Agglomerate sind durch $h(x)$ gegeben.

5.4.5 Festlegung wichtiger Größen für die Simulation

Für einige der von der Simulation benötigten Daten können wir keine konkreten Angaben machen. Wir sind deshalb darauf angewiesen Festlegungen zu treffen, die aufgrund von Daten aus der Literaturrecherche sowie eigenen Mess- und Simulationsergebnissen sinnvoll sind. Auf der Grundlage dieser Festlegungen wurden auch die Simulationen für die in Kapitel 5.6.2 vorgestellten Ergebnisse durchgeführt.

Die Funktion der Agglomerationsneigung

Aus den in Kapitel 5.1 genannten Fakten geht bereits hervor, dass es sich bei der Agglomerationsneigung $a(x) \rightarrow [0, 1]$ um eine monoton fallende Funktion handeln muss. Über den genauen Verlauf lässt sich jedoch keine allgemeingültige Aussage treffen. In [YZYC06] finden sich Untersuchungen zur Packungsdichte und Porenstruktur von monodispersen Partikelpackungen bei unterschiedlicher Korngröße. Die Daten wurden mit Hilfe einer DEM-Simulation ermittelt, die auch die Van-der-Waals Kräfte berücksichtigt. In Abbildung 5.8 sind einige der Daten dargestellt. Bei Korngrößen von über $200\mu m$ stellt sich eine stabile Packungsdichte ein bei etwa 0,61 (nahe dem bekannten Wert für monodisperse zufällig dichte Kugelpackungen von 0,63). Darunter

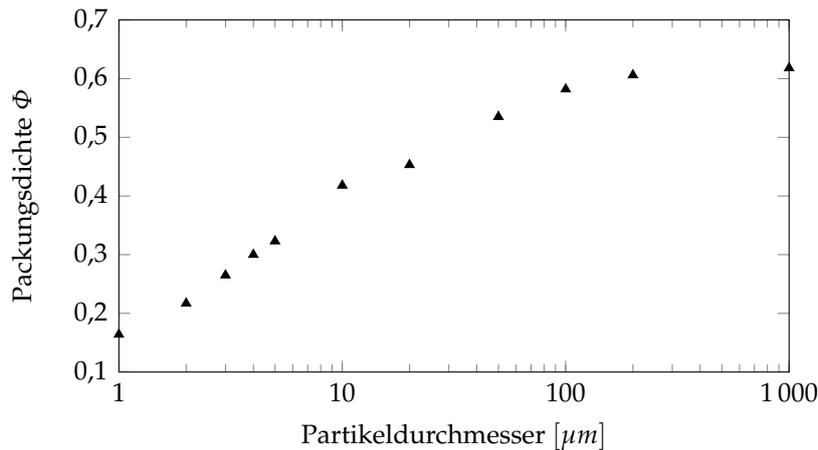


Abbildung 5.8: Zusammenhang zwischen Packungsdichte und Partikelgröße (Daten der DEM Simulation aus [YZYC06])

nimmt die Packungsdichte aber aufgrund der zunehmend dominierenden interpartikulären Haftkräfte ab. Aufgrund des Verlaufs der Packungsdichte schlug Raschdorf in seiner Arbeit [Ras10] vor, die Agglomerationsneigung auf der Grundlage einer logistischen Verteilung zu modellieren. Das führt zu der Gleichung:

$$a(x) = 1 - \frac{1}{e^{-\frac{x-u}{v}}}.$$

Die beiden Parameter u und v müssten dann individuell für verschiedene Stoffe oder Stoffgruppen mit ähnlichem Agglomerationsverhalten experimentell bestimmt werden.

Betrachten wir beispielsweise den Vergleich zwischen gemessenen und simulativ bestimmten Packungsdichten im unteren Teil der Abbildung 5.5. Die Differenz zwischen den beiden Datenreihen lässt darauf schließen, dass ab einer Grenze, die etwa zwischen $100 - 200 \mu m$ liegt, mit einer erheblichen Agglomeration der Partikel zu rechnen ist. Unterhalb dieser Grenze nimmt die Agglomeration dann mit sinkendem Partikeldurchmesser stark zu. Im Hinblick auf den praktischen Einsatz der PSD-Morph Simulation, sollte sich dieses Agglomerationsverhalten möglichst einfach modellieren lassen. Als Parameter sollten das Größt- und Kleinstkorn der Agglomeration sowie die Steilheit der Agglomerationsneigung $a(x)$ dienen. Sie sollten, ähnlich wie bei einer Fuller-Verteilung, direkt in der Funktionsgleichung verwendet werden. Die Funktion für die Agglomerationsneigung hat dann die Form:

$$a(x) = \left[1 - \left(\frac{x - x_{\min}}{x_{\max} - x_{\min}} \right)^u \right] \cdot a_{\max} \quad \text{mit: } x_{\min} \leq x \leq x_{\max}. \quad (5.8)$$

Mit x_{\min} bezeichnen wir die kleinste Korngröße der vorgegebenen Verteilung, x_{\max} bezeichnet die größte noch agglomerierende Korngröße. Über den Parameter u lässt sich die Steilheit der Funktion einstellen. Ein kleiner Wert (< 1) sorgt für eine (mit zunehmender Korngröße) schnell abfallende Agglomerationsneigung, während ein großer Wert (> 1) einen eher flachen Verlauf erzeugt. Da wir nicht davon ausgehen können, dass die kleinsten Korngrößen x_{\min} einer gegebenen Verteilung zu 100% agglomerieren, kann die Agglomerationsneigung über den zusätzlichen Parameter $a_{\max} \in [0, 1]$ skaliert werden; er gibt die maximale Agglomerationsneigung an, die bei x_{\min} anliegt. Wie diese Parameter aus dem Vergleich der Packungsdichten zwischen Simulation und Messung für einen Stoff bestimmt werden können, wird in Kapitel 5.6.2 beschrieben. Sind die Parameter einmal bekannt, können sie für Simulationen von unterschiedlichen Korngrößenverteilungen desselben Stoffes oder Stoffgruppen mit ähnlichem Agglomerationsverhalten eingesetzt werden.

Nun bleibt noch die Frage zu klären, wie die Weiteragglomerationsneigung von Agglomeraten $b(x)$ bestimmt werden kann. Da die Agglomerate die gleichen physikalischen und chemischen Eigenschaften haben, wie die Primärpartikel, aus denen sie zusammengesetzt sind, muss $b(x)$ grundsätzlich den gleichen Verlauf haben wie $a(x)$. Die Weiteragglomerationsneigung eines Agglomerates der Größe x ließe sich also aus $a(x)$ bestimmen. Die auffälligste Unterschied zwischen einem Agglomerat und einem Primärpartikel der gleichen Größe ist die starke Porosität des Agglomerates. Durch seine Struktur kann es große Hohlräume enthalten, wodurch sein Gewicht sehr viel kleiner ist, als das eines gleichgroßen Primärpartikels. Wie in Kapitel 5.1 erklärt wurde, wird das Wachstum der Agglomerate durch ihr Volumen bzw. Gewicht begrenzt. Je größer und voluminöser (und damit auch schwerer) eine Struktur wird, umso mehr dominiert die Schwerkraft (und nicht mehr die interpartikulären Haftkräfte) ihr Verhalten (siehe Abbildung 5.4). Wir beschränken uns deshalb auf die Betrachtung des Netto-Volumens der Agglomerate, das heißt das Volumen ohne die Hohlräume. Dieses Volumen ergibt sich aus der Summe aller Primärpartikelvolumen, aus denen das Agglomerat besteht. Weitere Eigenschaften wie die Größe und Struktur der Oberfläche werden vernachlässigt. Um die Weiteragglomerationsneigung zu bestimmen, benutzen wir den Durchmesser einer Kugel, die das Volumen des Agglomerates hat. Wenn also x_a und V_a die Größe und das Netto-Volumen eines Agglomerates bezeichnen, dann ermitteln wir $b(x_a)$ mit:

$$b(x_a) = a(x'), \quad \text{mit } x' = \sqrt[3]{\frac{V_a}{\frac{1}{6}\pi}}$$

Da die Agglomerationsneigung $a(x)$ eine monoton fallende Funktion ist, haben Ag-

glomerate somit eine deutlich höhere Wahrscheinlichkeit (weiter) zu agglomerieren als Primärpartikel der gleichen Größe.

Bestimmung der Größe der Agglomerate

Grundsätzlich können wir die Größe eines Agglomerates über die minimale Hüllkugel bzw. „Bounding-Sphere“ bestimmen, also eine Kugel minimalen Durchmessers, die das Agglomerat vollständig einhüllt. Den Durchmesser dieser „Bounding-Sphere“ legen wir dann als Partikeldurchmesser des Agglomerates fest. Es ist jedoch zu beachten, dass die Agglomerate oft sehr komplexe Formen aufweisen, die stark von der Kugelform abweichen. In Abbildung 6.10 sind einige Agglomerateformen aufgelistet, die von dem von uns benutzten Agglomeratengenerator erzeugt wurden. Sie haben eine fraktalartige Form mit einer stark zerklüfteten Oberfläche. Innerhalb einer Packung könnten also andere Primär- oder Agglomeratekugeln teilweise in diese Oberfläche eindringen. Außerdem sind reale Agglomerate im Gegensatz zu ihren Primärpartikeln keine starren Gebilde, sie können durch den Verdichtungsdruck innerhalb einer Packung komprimiert werden. Um diese Fakten berücksichtigen zu können, führen wir einen Skalierungsfaktor $\tau \in (0, 1]$ ein. Mit diesem Faktor wird der Durchmesser der Bounding-Sphere so skaliert, dass die Agglomerate in der Packungssimulation durch entsprechend kleinere Kugeln dargestellt werden. Diese Skalierung wird innerhalb der PSD-Morph Simulation, bei der Generierung der Agglomerate, vorgenommen und ist somit bereits in den Verteilungen $\hat{q}(x)$ und $h(x)$ berücksichtigt. Anpassungen innerhalb der Packungssimulation RaSim sind dadurch unnötig. Ein sinnvoller Wert für τ kann aus der Verteilung der inneren Dichte der Agglomerate aus Abbildung 6.9 in Kapitel 6.5 abgeleitet werden.

Festlegung der Anzahl der Agglomerate pro Generation

Der Agglomeratengenerator innerhalb der PSD-Morph Simulation hat die Aufgabe, pro Generation i die mittlere Größe $d(i)$, die mittlere Anzahl enthaltener Primärpartikel $m(i)$ und (für die anschließende Packungssimulation) die mittlere innere Raumausfüllung $\rho(i)$ der Agglomerate zu bestimmen. Dafür ist es notwendig, eine ausreichend große Stichprobe von Agglomeraten zu erzeugen, um die entsprechenden Werte abschätzen zu können. Die einfachste Möglichkeit, um die Größe der Stichprobe zu bestimmen, besteht darin, sie als Parameter fest vorzugeben. Wird der Parameter auf einen ausreichend hohen Wert (> 100) gesetzt, kann man gute Abschätzungen für die Mittelwerte erwarten. Diese Strategie ist vor allem dann sinnvoll, wenn die Performance der Simulation getestet werden soll. Da durch den fixen Parameter immer die

gleiche Anzahl Agglomerate erzeugt werden, lassen sie die Laufzeiten unterschiedlicher Simulationen miteinander vergleichen.

Die Festlegung einer fixen Anzahl Agglomerate pro Generation erfolgt jedoch rein willkürlich und erlaubt keine Aussagen über die tatsächliche Zuverlässigkeit der Schätzung der Mittelwerte. Deshalb werden für solche Problemstellungen üblicherweise Intervallschätzer eingesetzt. Um das 95%-Konfidenzintervall für die mittlere Größe der Agglomerate zu ermitteln, geht man folgendermaßen vor: Ist n die Anzahl der bisher erzeugten Agglomerate der aktuellen Generation und sind die beobachteten Durchmesser der Agglomerate d_1, \dots, d_n , dann können wir die Varianz σ^2 aus diesen Daten mit:

$$\sigma_n := \sqrt{\frac{1}{n-1} \sum_{i=1}^n (d_i - \bar{d}_n)^2}, \quad \text{mit} \quad \bar{d}_n := \frac{1}{n} \sum_{i=1}^n d_i$$

abschätzen. Legen wir eine Standardnormalverteilung zugrunde, dann liegt das $(1 - \alpha/2)$ Fraktile für $\alpha = 5\%$ bei $\approx 1,96$, das heißt bei diesem Wert erreicht die Verteilung 95%. Wir erzeugen nun solange Agglomerate, bis zum ersten Mal gilt:

$$n \geq 1,96 \cdot \frac{\sigma_n}{0,05 \cdot \bar{d}_n}, \quad n \geq n_0$$

wobei n_0 eine kleine Zahl von Vorläufen beschreibt. Wir können dann die Aussage treffen, dass das erzeugte Intervall von Agglomerategrößen den tatsächlichen Mittelwert mit 95%-iger Wahrscheinlichkeit enthält. Der aus der Stichprobe geschätzte Wert $d(i) := \bar{d}_n$ liegt also mit hoher Wahrscheinlichkeit nahe beim tatsächlichen Mittelwert für die Größe von Agglomeraten der Generation i . Da es nicht möglich ist, aus der Güte der Schätzung des Mittelwertes für die Agglomerategröße, auf die Güte der entsprechenden Werte für die Anzahl der enthaltenen Primärpartikel und der inneren Raumauffüllung zu schließen, sollte das Verfahren unabhängig für alle zu ermittelnden Werte durchgeführt werden. Die Erzeugung der Agglomerate kann dann beendet werden, wenn alle drei Werte mit der entsprechend hohen Sicherheit geschätzt werden können.

5.5 Details der Implementierung

Die Implementierung des PSD-Morph Algorithmus besteht im wesentlichen aus 3 Objekten. Das Objekt „Agglomerationsneigung“ speichert die Verteilung der Anzahlanteile der nicht-agglomerierten Primärpartikel $q(x)$ sowie die Funktion der Agglomerationsneigung pro Korngröße $a(x)$. Aus diesen Eingabedaten sowie den Daten, die

die Simulation in jedem Iterationsschritt liefert, kann das Objekt die Verteilung der agglomerierenden Primärpartikelgrößen der jeweils nächsten Agglomeratengeneration $g_i(x)$ bestimmen (Formel 5.2). Ebenso werden hier die Anteile der stabilen nicht weiter wachsenden Agglomerate $h(x)$ berechnet. Das Objekt kann damit nach Abschluss der Simulation die agglomerierte Korngrößenverteilung $\hat{q}(x)$ als Ergebnis ausgeben.

Das zweite Objekt ist der „Agglomeratengenerator“, er erzeugt zunächst per Sampling eine Menge aus Kugeln mit zufällig verteilten Durchmessern. Als Verteilung für dieses Sampling dient die vom Objekt „Agglomerationsneigung“ berechnete Verteilung der agglomerierenden Primärpartikelgrößen. Aus diesen Kugeln werden dann Agglomerate zusammen gesetzt, bis deren Größe die Grenze zum nächst höheren Korngrößenintervall überschreitet.

Das dritte Objekt ist der „Collector“, er dient als Vermittler zwischen den beiden ersten Objekten. Seine Aufgabe ist es, alle vom „Agglomeratengenerator“ erzeugten Agglomerate aufzusammeln und Funktionen für deren Auswertung zur Verfügung zu stellen. Vor allem berechnet er die Größe jedes Agglomerates gemäß der Beschreibung in Kapitel 5.4.5, außerdem die mittlere Anzahl Primärpartikel, die in jedem Agglomerat verbaut sind, und die innere Raumauffüllung ρ jedes Agglomerates. Die Größe der Agglomerate und die Anzahl der in ihnen enthaltenen Primärpartikel werden vom Objekt „Agglomerationsneigung“ benutzt, um die Anteile der stabilen Agglomerate $h(x)$ sowie die in $g_i(x)$ enthaltenen weiter agglomerierenden Primärpartikelanteile zu berechnen. Die innere Raumauffüllung ρ kann zusammen mit dem fertigen Simulationsergebnis (der agglomerierten Korngrößenverteilung $\hat{q}(x)$) benutzt werden, um in der Packungssimulation die Agglomeration mit einzuberechnen. Dieses Vorgehen wird in Kapitel 5.6.1 beschrieben. Zusätzlich zu den von der Simulation benötigten Werten kann der „Collector“ aber auch noch weitere Auswertungsfunktionen bereitstellen, wie beispielsweise eine Visualisierung der erzeugten Agglomerate. Das Zusammenspiel der 3 Objekte ist im Algorithmus 4 dargestellt.

Der Algorithmus besteht aus einer äußeren Schleife (Zeilen 4 - 18) und einer inneren Schleife (Zeilen 12 - 16). Das zentrale Element ist die Anweisung in Zeile 5, das Objekt „Agglomerationsneigung“ (**ANeigung**) berechnet hier die Korngrößenverteilung der agglomerierenden Primärpartikel für den nächsten Iterationsschritt. Die Anteile für diese Verteilung stammen aus den instabilen noch weiter wachsenden Agglomeraten der letzten Generation sowie den nächst größeren noch nicht agglomerierten Primärpartikeln. Für diese Berechnungen benötigt das Objekt die Daten der letzten Agglomeratengeneration, die es aus dem Objekt „Collector“ entnimmt. Während des ersten Schleifendurchlaufs enthält der „Collector“ noch keine Daten. In diesem Fall wird in Zeile 5 nur eine minimale Verteilung erzeugt, sie besteht mindestens aus der kleinsten

Algorithmus 4 : PSD-Morph

Input : Die Verteilungsdichte der nicht agglomerierten Anzahlanteile ($q(x)$)

Input : Die Agglomerationsneigung der Primärpartikel ($a(x)$)

Output : Die Verteilungsdichte der agglomerierten Anzahlanteile ($\hat{q}(x)$)

Output : Die Agglomerateanteile ($h(x)$)

Output : Die mittleren Raumausfüllung jeder Agglomeratgröße ($\rho(x)$)

```
1 new ANeigung( $q(x), a(x)$ )
2 new Collector
3  $g(x) = 0$  // Korngrößenverteilung der agglomerierenden Primärpartikel
4 repeat
5    $g(x) \leftarrow$  ANeigung.updatePKG (Collector)
6   if  $g(x) \neq 0$  then
7     // Es existieren noch agglomerierende Partikel
8      $abbruchgroesse =$  Collector.midAGroesse ()
9     if  $abbruchgroesse = 0$  then
10       $abbruchgroesse = \max(x) \in g(x)$ 
11    delete Collector // Löschen des alten Collectors
12    new Collector // Neuen leeren Collector erzeugen
13    repeat
14      new AGenerator( $g(x)$ ) // Agglomeratengenerator initialisieren
15      // Erzeugen eines Agglomerates und hinzufügen zum Collector
16      Collector  $\leftarrow$  AGenerator.start (abbruchgroesse ( $\cdot \frac{1}{\tau}$ ))
17      delete AGenerator
18    until ABRUCHKRITERIUM
19     $\rho(x) \leftarrow$  Collector.midRaumausfuellung ()
20 until  $g(x) = 0$ 
21  $\hat{q}(x) \leftarrow$  ANeigung.AKGV ()
22  $h(x) \leftarrow$  ANeigung.AAnt ()
23 return ( $\hat{q}(x), h(x), \rho(x)$ )
```

agglomerierenden Korngröße und zusätzlich allen folgenden Korngrößen x , deren Agglomerationsneigung $a(x) = 1$ ist. Da alle Korngrößen mit Agglomerationsneigung 1 keine stabilen Primärpartikel enthalten, können sie (in dem von uns benutzten Modell) auch keine stabilen Agglomerate enthalten. Agglomerate können also erst in Korngrößen x mit $a(x) < 1$ auftreten. In allen Iterationsschritten $i > 1$ enthält der „Collector“ die Daten der letzten Agglomeratengeneration. Damit kann das Objekt „Agglomerationsneigung“, gemäß der Formel 5.6, die Verteilung der fertigen stabilen Agglomerate $h(x)$ aufbauen. Die Korngrößenverteilung der agglomerierenden Primärpartikel $g(x)$ wird gemäß der Formel 5.2 berechnet.

Solange noch instabile weiter-agglomerierende Partikel existieren, dass heißt es gilt: $g(x) \neq 0$ für $x \in [0, S]$ (Zeilen 6 und 18), muss die Simulation fortgesetzt werden. In diesem Fall wird zunächst die Abbruchgröße für das Wachstum der Agglomerate der nächsten Generation festgelegt (Zeilen 7 - 9). Dafür wird die mittlere Größe der Agglomerate der letzten Generation benutzt, die aus dem „Collector“ entnommen werden kann. Das heißt, die nächsten Agglomerate werden ihr Wachstum stoppen, sobald sie größer sind als die Agglomerate der letzten Generation. Da der „Collector“ während des ersten Schleifendurchlaufs noch keine Daten enthält, muss die Abbruchgröße dann auf andere Weise bestimmt werden. In diesem Fall wird die größte Korngröße benutzt, die in der Verteilung der agglomerierenden Primärpartikel $g(x)$ noch enthalten ist (Zeile 9). Die Agglomerate stoppen also ihr Wachstum, sobald sie größer werden als die Primärpartikel aus denen sie bestehen. Bevor eine neue Agglomeratengeneration erzeugt werden kann, wird der „Collector“ gelöscht und ein neuer leerer „Collector“ angelegt (Zeilen 10 - 11). Alle im „Collector“ enthaltenen Daten sind zu diesem Zeitpunkt ausgewertet und werden nicht länger gebraucht. Der neue „Collector“ speichert die Daten des nächsten Iterationsschrittes. Innerhalb der inneren Schleife werden mit Hilfe des „Agglomeratengenerators“ (**AGenerator**) solange Agglomerate erzeugt (und dem „Collector“ übergeben), bis ein festgelegtes Abbruchkriterium erreicht ist. Für das Abbruchkriterium gelten die Erklärungen aus Kapitel 5.4.5 für die „Festlegung der Anzahl der Agglomerate pro Generation“. Beim Start des „Agglomeratengenerators“ (Zeile 14) wird die Abbruchgröße noch mit dem Faktor $\frac{1}{\tau}$ skaliert. Da die Agglomeratengrößen später, gemäß den Erklärungen in Kapitel 5.4.5 zur „Bestimmung der Größe der Agglomerate“, um den Faktor τ skaliert werden, wird so sichergestellt, dass alle Agglomerate der Generation tatsächlich größer als die vorgegebene Abbruchgröße sind.

Sobald eine Agglomeratengeneration fertig gestellt ist, kann die mittlere innere Raumausfüllung der Agglomerate ρ bestimmt werden. Sie wird in einem Vektor gespeichert, der die innere Raumausfüllung jeder Agglomeratengröße enthält (Zeile 17). Wie die

Raumausfüllung im Inneren der simulierten Agglomerate verteilt ist und wie sie bestimmt wird, wird in Kapitel 6.5 beschrieben.

Wenn keine instabilen Partikel mehr existieren und keine weitere Agglomeration mehr stattfinden kann, wird die äußere Schleife beendet. Das Objekt „Agglomerationsneigung“ kann nun die fertig agglomerierte Korngrößenverteilung $\hat{q}(x)$ berechnen, indem der von agglomerierenden Primärpartikeln befreiten Verteilung $(1 - a(x))q(x)$ die berechneten Anteile der Agglomerate $h(x)$ zugefügt werden (Formel 5.7). Die fertig zusammengestellte und normierte Verteilung kann dann zusammen mit den Anteilen und der inneren Raumausfüllung der Agglomerate $h(x)$ und $\rho(x)$ ausgegeben werden (Zeilen 19 - 21).

5.5.1 Berücksichtigung der Streuung der Agglomerategrößen

Um ein erstes vorläufiges Ergebnis einer PSD-Morph Simulation zu erhalten, wurden die Korngrößenverteilung eines feinen Quarzsandes und eine einfache linear fallende Agglomerationsneigung benutzt. Diese Eingabedaten sind im oberen Teil der Abbildung 5.9 dargestellt. Die Korngrößen des Quarzsandes erstrecken sich von $0,122\mu\text{m}$ bis $409,6\mu\text{m}$, wobei über 99,9% der Anzahlanteile unter $4\mu\text{m}$ liegen. Die sich aus den Anzahlanteilen der Primärpartikel $q(x)$ und der Agglomerationsneigung $a(x)$ ergebenden Anteile der agglomerierenden Primärpartikel $g(x)$ sind als graue Fläche abgebildet. Alle 3 Funktionen sind konstant auf den Korngrößenklassen, so dass deren Anzahl und Lage gut zu erkennen sind. Das Ergebnis der Simulation ist im unteren Teil der Abbildung 5.9 dargestellt. Wie zu erwarten, haben sich die Anteile deutlich in den Bereich der höheren Korngrößenklassen verschoben. Aus den Anteilen der agglomerierenden Primärpartikel $g(x)$ sind die Anteile der Agglomerate $h(x)$ geworden, die ebenfalls als graue Fläche abgebildet sind. Die Funktion $h(x)$ hat den Verlauf einer logarithmischen Normalverteilung, wie dies auch schon von Raschdorf [Ras10] durch die Auswertung von experimentellen Ergebnissen geschätzt wurde. Auffällig ist hier aber, dass nicht in allen Korngrößenklassen Agglomerateanteile enthalten sind, sie beschränken sich auf einige steile „Peaks“, die in regelmäßigen Abständen auftreten. In den Korngrößen dazwischen ist $h(x) = 0$. Der Grund dafür ist in Abbildung 5.10 dargestellt. Sie zeigt die Streuung der Agglomerate der ersten 10 Generationen der Simulation über mehrere Korngrößenklassen. Jede der Generationen besteht aus 100 Agglomeraten. Die Durchmesser dieser Agglomerate sind aber nicht alleine über die Korngrößenklasse verteilt, in der sich ihre mittlere Größe befindet. Die Agglomerategrößen sind hier über bis zu 7 Korngrößenklassen verteilt. Da die mittlere Größe der Agglomerate jeder Generation (in der Abbildung jeweils durch eine gestrichelte Linie in der Farbe der Generation markiert), gleichzeitig auch als Abbruchgröße des

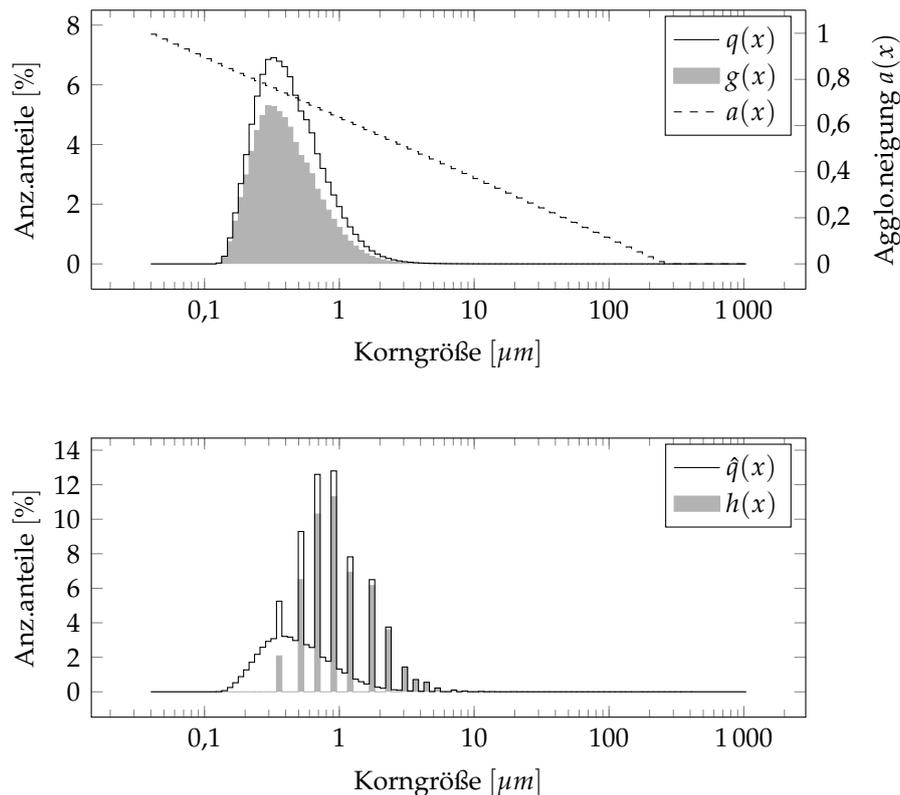


Abbildung 5.9: PSD-Morph Simulation am Beispiel der Korngrößenverteilung eines feinen Quarzsandes: nicht agglomerierte Ausgang-Verteilung $q(x)$ mit Agglomerationsneigung $a(x)$ und daraus resultierende agglomerierende Primärpartikelanteile $g(x)$ (oben) und agglomerierte Ergebnis-Verteilung $\hat{q}(x)$ mit Agglomerateanteilen $h(x)$ (unten).

Wachstums für die Agglomerate der nächsten Generation dient, kommt es zur Überschneidung der Agglomerategrößen mehrerer Generationen.

Die relativ weite Streuung der Durchmesser der Agglomerate lässt sich durch die Art ihres Wachstums erklären, das im Agglomeratengenerator implementiert ist. Die entstehenden Agglomerate bestehen aus mehreren Primärpartikeln und sind sehr viel poröser als es eine zufällig dichte Packung dieser Partikel wäre. Da bedeutet, dass sich zwischen den einzelnen Primärpartikeln eines Agglomerates sehr große Hohlräume befinden können. Überschreitet der Durchmesser eines Agglomerates während seines Wachstums im Agglomeratengenerator die Abbruchgröße, so bestimmen die Größe des letzten hinzugefügten Primärpartikels und seine Andockstelle am Agglomerat, welche Größe das Agglomerat haben wird. In Abbildung 5.11 wird dieser Umstand dargestellt. Dockt das letzte Primärpartikel an einer exponierten Stelle des Agglomerates an (Teil (a) der Abbildung), so wird das Agglomerat ungefähr um den Durchmesser dieses Primärpartikels größer sein als die Abbruchgröße. Es kann also auch mehrere Korngrößenklassen über der Abbruchgröße liegen, wenn das letzte Primärpartikel ent-

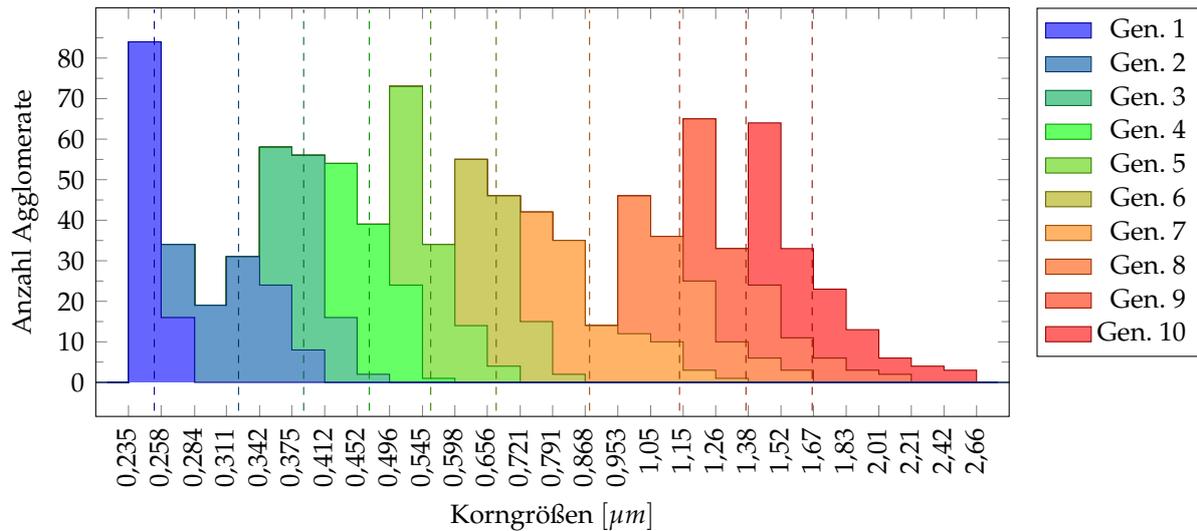


Abbildung 5.10: Anzahl der pro Korngröße erzeugten Agglomerate (farblich getrennt nach Generationen) für die ersten 10 Generationen einer Simulation. Die mittlere Größe der Agglomerate jeder Generation ist jeweils durch eine gestrichelte Linie markiert.

sprechend groß ist. Taucht das letzte Primärpartikel statt dessen tief in einen Hohlraum des Agglomerates ein (Teil (b) der Abbildung), wird das Agglomerat dadurch nur unwesentlich vergrößert, sein Durchmesser wird sich dann nahe an der Abbruchgröße befinden.

Die in Kapitel 5.4.3 vorgestellte Methode zur Bestimmung der Agglomerateanteile (Formeln 5.5 und 5.6) berücksichtigt diese Streuung der Agglomerategrößen nicht. Alle Agglomerateanteile eines Iterationsschrittes i werden stets dem Korngrößenintervall der mittleren Größe der Agglomerate $I_{d(i)}$ hinzugefügt. Dabei wird davon ausgegangen, dass die Agglomerategrößen einer Generation i nur in $I_{d(i)}$ verteilt sind. Das ist jedoch nicht der Fall wie in Abbildung 5.10 zu sehen ist. Wegen dieser relativ weiten Streuung der Agglomerategrößen, muss auch die mittlere Größe der Nachfolgeneration d_{i+1} nicht unbedingt im nächst folgenden Intervall $I_{d(i)+1}$ liegen. Das erklärt die vereinzelt „Peaks“ im unteren Teil der Abbildung 5.9. Da die Agglomerate jeder Generation i über mehr als das Intervall $I_{d(i)}$ verteilt sind, müssen auch die Anzahlanteile der Agglomerate entsprechend verteilt werden. Es seien $s_{a(i)}$ und $s_{A(i)}$ die Durchmesser des kleinsten und größten Agglomerates im Iterationsschritt i mit: $0 < s_{a(i)} \leq s_{A(i)} \leq S$. Ferner benötigen wir eine Funktion die angibt, wie die Anteile der agglomerierenden Primärpartikel $g_i(x)$ in Form von Agglomeraten über das Intervall $[s_{a(i)}, s_{A(i)})$ verteilt sind. Dafür definieren wir die Funktion $k_i(x, y)$, die angibt, welcher Anteil an Primärpartikeln der Größe x in Agglomeraten der Größe y verbaut

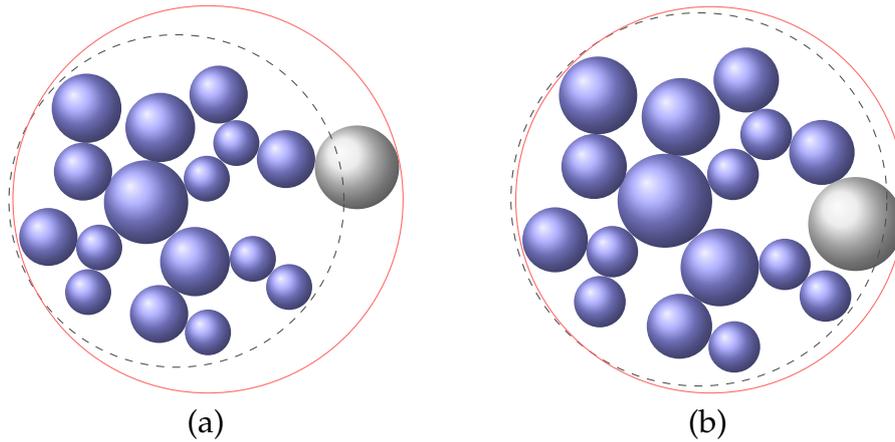


Abbildung 5.11: Streuung der Agglomeratgrößen: bei gleicher Abbruchgröße (gestrichelter Kreis) entstehen unterschiedliche Agglomeratgrößen (roter Kreis) in Abhängigkeit von der Andockstelle des letzten Primärpartikels (graue Kugel).

ist. Es gilt:

$$\int_{[0,S]} k_i(x,y)dy = g_i(x).$$

Diese Daten stehen im „Collector“-Objekt der PSD-Morph Simulation zur Verfügung. Nach dem Erzeugen einer Agglomeratengeneration ist bekannt, wie viele Primärpartikel einer bestimmten Größe gesammelt wurden und ebenfalls, wie viele davon in Agglomeraten einer bestimmten Größe enthalten sind. Damit ist es nun möglich, die Agglomerateanteile $h_i(x)$ mit Berücksichtigung der Streuung der Agglomeratgrößen zu berechnen.

$$h_1(x) = 0, \quad x \in [0, S]$$

$$h_{i+1}(x) = \begin{cases} h_i(x) & \text{falls } x < s_{a(i)} \\ h_i(x) + \frac{\int_{[0,s_i]} k_i(z,x)dz \cdot (1-\tilde{b}(i+1))}{m(i+1)} & \text{falls } x \in [s_{a(i)}, s_{A(i)}] \end{cases}$$

Mit:

$$\int_{[0,s_i]} k_i(z,x)dz \cdot \tilde{b}(i+1)$$

ist der Anteil der in nicht stabilen (also weiter wachsenden) Agglomeraten der Größe x verbauten Primärpartikel gegeben. Dieser Anteil steht für die Agglomerate der nächsten Generation zur Verfügung.

Das Ergebnis der Simulation mit einem entsprechend erweiterten PSD-Morph Algorithmus ist in Abbildung 5.12 dargestellt. Hier gibt es keine Lücken mehr in der Verteilung der Agglomerateanteile $h(x)$. Dennoch zeigen sich auffällige „Zacken“ im

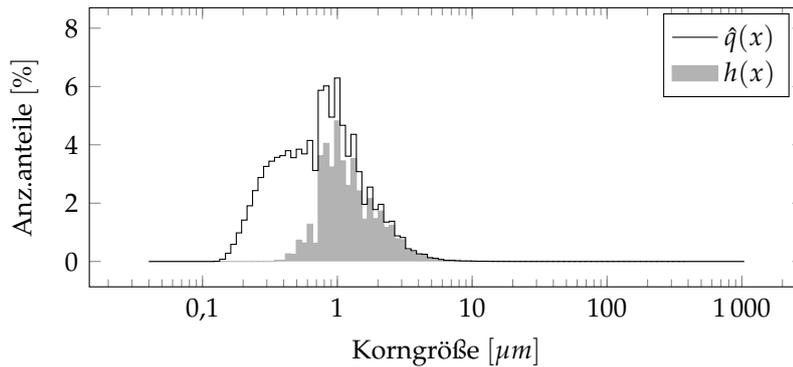


Abbildung 5.12: Ergebnis einer PSD-Morph Simulation mit Berücksichtigung der Streuung der Agglomerategrößen.

Korngrößenbereich mit den höchsten Agglomerateanteilen zwischen $0,6\mu m$ und $3\mu m$. Dies ist eine Besonderheit der PSD-Morph Simulation, die durch die Aufteilung des eigentlich kontinuierlichen Agglomeratewachstums in Agglomerategenerationen verursacht wird. Für einen natürlicheren Verlauf der Verteilungskurven könnte, in einer Nachbearbeitung der Daten, eine Glättung durchgeführt werden. Dies würde, in der hier verwendeten Korngrößenverteilung, nur eine geringfügige Korrektur der Agglomerateanteile ($\leq 1\%$) in jeder Korngrößenklasse bedeuten. In den folgenden Kapiteln wird beschrieben, wie aus den per PSD-Morph ermittelten Daten und der Packungssimulation RaSim die Packungsdichte einer Partikelmischung mit Berücksichtigung der Agglomeration bestimmt werden kann. Welchen Einfluss der genaue Verlauf der agglomerierten Korngrößenverteilung $\hat{q}(x)$, im Vergleich zu anderen Faktoren, auf die Senkung der Packungsdichte hat, wird in Kapitel 5.6.2 anhand einiger Simulationsergebnisse dargestellt.

5.6 Simulative Bestimmung der Packungsdichte von agglomerierten Partikelmischungen

Um die Packungsdichte einer agglomerierten Partikelmischung bestimmen zu können, müssen wir die PSD-Morph mit der RaSim Simulation kombinieren. Diese Kombination, die im Folgenden erklärt wird, ist schematisch in Abbildung 5.13 dargestellt. Die Berechnung der agglomerierten Packungsdichte Φ_A basiert dabei wieder auf der Arbeit von Raschdorf [Ras10].

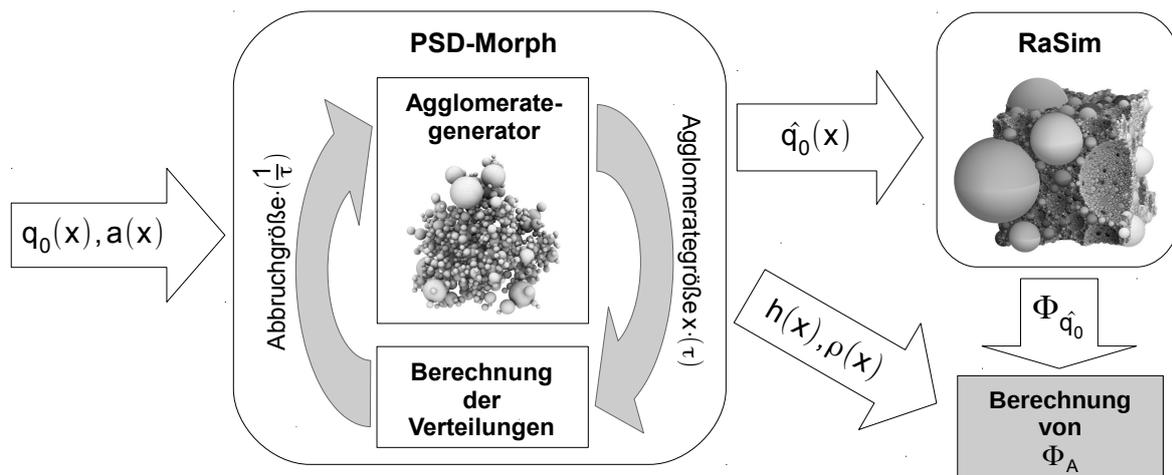


Abbildung 5.13: Kombination von PSD-Morph und RaSim zur Bestimmung der Packungsdichte einer agglomerierten Partikelmischung Φ_A

5.6.1 Kombination von PSD-Morph und RaSim

Wie in den vorangegangenen Kapiteln deutlich wurde, benötigen wir die nicht agglomerierte Korngrößenverteilung der zu simulierenden Mischung $q(x)$, die messtechnisch ermittelt werden muss. Darüber hinaus benötigen wir noch die Funktion der Agglomerationsneigung $a(x)$, deren Verlauf aus dem Vergleich mit den messtechnisch ermittelten Packungsdichten geschätzt werden kann. Diese Funktion muss nur einmal für einen Stoff oder eine Gruppe von Stoffen mit ähnlichem Agglomerationsverhalten bestimmt werden. Diese Daten genügen bereits, um mittels PSD-Morph Simulation die agglomerierte Korngrößenverteilung $\hat{q}_0(x)$, die darin enthaltenen Verteilung der Agglomerateanteile $h(x)$, sowie die zugehörige innere Raumaufüllung der Agglomerate pro Korngröße $\rho(x)$ zu approximieren.

Wir setzen voraus, dass wir sowohl die nicht-agglomerierten Primärpartikel als auch die Agglomerate innerhalb der Packungssimulation als Kugeln darstellen. Damit kann die Packungsdichte der agglomerierten Partikelmischung auf der Grundlage einer normalen Kugelpackungssimulation bestimmt werden. Weitere Anpassungen des Simulationssystems sind nicht notwendig. Die von der PSD-Morph Simulation ermittelte agglomerierte Korngrößenverteilung $\hat{q}_0(x)$ kann direkt als Eingabe für die Packungssimulation RaSim verwendet werden, die daraus die Packungsdichte einer entsprechenden Kugelpackung $\Phi_{\hat{q}_0}$ bestimmt. In $\Phi_{\hat{q}_0}$ wird jedoch nur die durch die Agglomeration veränderte Korngrößenverteilung berücksichtigt, die Agglomerate werden innerhalb der Packungssimulation als solide Kugeln behandelt. Tatsächlich enthalten die Agglomerate aber aufgrund ihrer Struktur zusätzlichen Hohlraum. Dieser Hohlraum muss zum Leervolumen der Kugelpackung hinzu addiert werden, um

die Packungsdichte der agglomerierten Partikelmischung Φ_A zu erhalten. Laut Definition ist die Packungsdichte gegeben durch:

$$\Phi = \frac{V_K}{V_C} = \frac{V_K}{V_K + V_{leer}}.$$

Dabei ist V_K das Nettovolumen aller Kugeln der Packung und V_C das Gesamtvolumen der Packung, das sich aus dem Nettovolumen der Kugeln und dem Porenvolumen V_{leer} zusammensetzt. Für die Berechnung von Φ_A müssen wir berücksichtigen, dass sich die Packung aus Primärpartikeln und Agglomeraten zusammensetzt, damit ergibt sich:

$$\Phi_A = \frac{V_{Prim} + \rho V_{Agg}}{V_K + V_{leer}}.$$

Mit V_{Prim} wird das Volumen aller Kugeln bezeichnet, die nicht agglomerierte Primärpartikel darstellen und mit V_{Agg} das Volumen aller Kugeln, die Agglomerate darstellen. Da die Agglomerate keine soliden Kugeln sind, sondern Hohlräume aufweisen, wird ihr Volumen mit ihrer inneren Dichte ρ multipliziert. Da wir mit $h(x)$ auch die Verteilung der Agglomerateanteile gegeben haben, können wir den Anteil der Agglomerate an der Gesamtanzahl aller Partikel A_h einfach mit:

$$A_h = \int_0^\infty h(x) dx$$

berechnen. Der Anteil ihres Volumens (inklusive der Hohlräume) am Gesamtvolumen aller Partikel V_h beträgt:

$$V_h = \frac{\int_0^\infty \frac{\pi}{6} h(x) x^3 dx}{\int_0^\infty \frac{\pi}{6} \hat{q}_0(x) x^3 dx} = \frac{\int_0^\infty h(x) x^3 dx}{\int_0^\infty \hat{q}_0(x) x^3 dx}.$$

Damit ergibt sich:

$$\begin{aligned} \Phi_A &= \frac{(1 - V_h)V_K + \rho V_h V_K}{V_K + V_{leer}} \\ &= \frac{V_K \cdot (1 - V_h + \rho V_h)}{V_K + V_{leer}} \\ &= \frac{V_K}{V_K + V_{leer}} \cdot (1 - (1 - \rho)V_h) \end{aligned}$$

Die Packungsdichte der agglomerierten Mischung Φ_A kann damit einfach aus $\Phi_{\hat{q}_0}$ be-

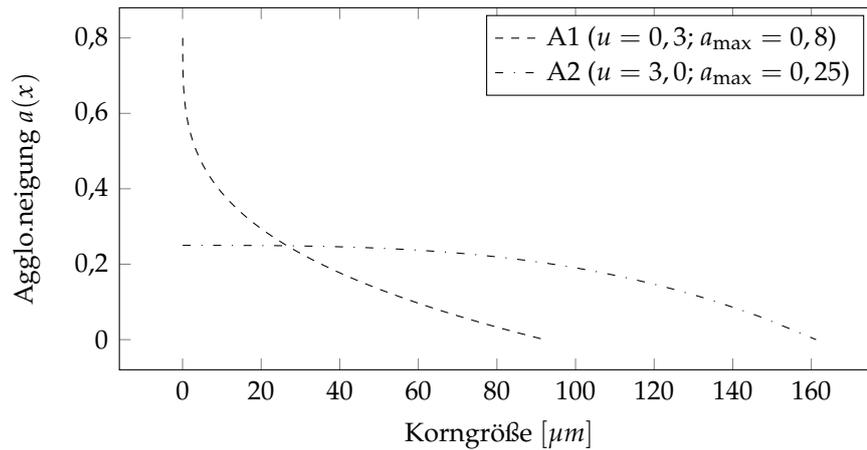


Abbildung 5.14: Zwei mögliche Verläufe für die Funktion der Agglomerationsneigung (nach Gleichung 5.8)

rechnet werden mit:

$$\Phi_A = \Phi_{\hat{q}_0} \cdot (1 - (1 - \rho)V_h).$$

5.6.2 Simulationsergebnisse

Die hier vorgestellten Simulationsergebnisse wurden aus den Korngrößenverteilungen von Sandgemischen mit unterschiedlichen Feinkornanteilen gewonnen. Diese Sandgemische wurden bereits in Abbildung 5.5 vorgestellt. Der Vergleich der simulativ mit RaSim ermittelten Packungsdichten mit den per Pyknometer gemessenen Werten (auch in Abbildung 5.15 noch einmal dargestellt) erlaubt einige Aussagen über den ungefähren Verlauf der Agglomerationsneigung $a(x)$ dieser Gemische. Die gemessenen und simulativ ermittelten Werte bei Gemischen mit Korngrößen über $200\mu m$ bzw. über $500\mu m$ weichen kaum von einander ab. Die ermittelten Differenzen lassen sich auch vor allem auf die von der Kugelgestalt abweichenden Kornform zurückführen und weniger auf die Auswirkungen der Agglomeration. Wir können daraus schließen, dass $a(x) \approx 0$ für $x > 200\mu m$ gilt. Für Gemische mit kleineren Korngrößen nimmt die Differenz zwischen den beiden Werten jedoch deutlich zu. Da der genaue Verlauf der Agglomerationsneigung aber dennoch nicht bekannt ist, wurden mehrere Simulationläufe mit unterschiedlichen $a(x)$ durchgeführt. Zwei mögliche Verläufe der Agglomerationsneigung, die wir mit A1 und A2 bezeichnen, sind in Abbildung 5.14 dargestellt. Für die Berechnung der beiden Funktionen aus einer geschätzten maximalen Agglomerationsneigung a_{\max} und einem Exponenten für die Steilheit u wurde die Gleichung 5.8 aus Kapitel 5.4.5 benutzt. In A1 gehen wir mit $a_{\max} = 0,8$ von einer sehr hohen Agglomerationsneigung für die kleinsten Korngrößen mit $0,04\mu m$ aus. Sie fällt

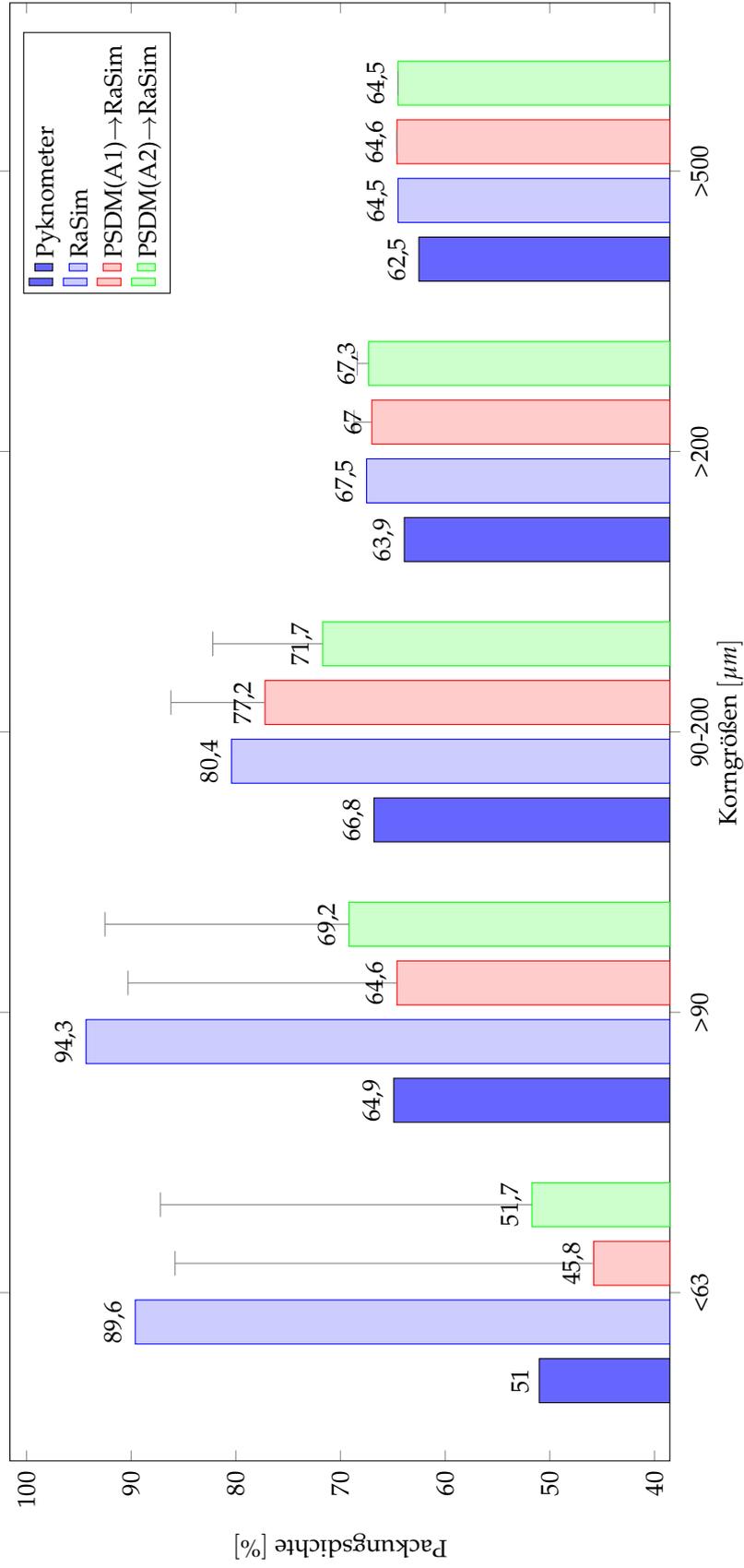


Abbildung 5.15: Vergleich der gemessenen und simulierten Raumausfüllungen verschiedener Quarzsandgemische. (Die zusätzlichen Markierungen über den „PSD-Morph → RaSim“ Simulationsergebnissen geben die Raumausfüllung Φ_{q_0} an, die nur auf der veränderten Korngrößenverteilung beruht.)

jedoch mit steigender Korngröße sehr schnell ab und erreicht bereits bei $x = 92\mu\text{m}$ ihren Nullpunkt. Mit diesen Daten wurden die Packungsdichten der Sandgemische per Simulation ermittelt. Dabei wurde die Kombination der PSD-Morph und der RaSim Simulation eingesetzt, die im vorigen Kapitel beschrieben wurde. Die Ergebnisse sind in Abbildung 5.15 unter der Bezeichnung „PSDM(A1)→RaSim“ dargestellt. Obwohl es sich bei A1 nur um eine erste grobe Schätzung des Verlaufs der Agglomerationsneigung handelt, zeigt sich eine deutliche Verbesserung der Simulationsergebnisse gegenüber einer reinen RaSim Simulation. Insbesondere findet sich der Trend in den gemessenen Werten jetzt auch in den Simulationsergebnissen wieder. Die Simulation ermittelt für das Sandgemisch mit Korngrößen $< 63\mu\text{m}$ die geringste Packungsdichte aller Gemische, was durch die per Pyknometer gemessenen Werte bestätigt wird. Dennoch gibt es deutliche Abweichungen zwischen den gemessenen und den simulativ ermittelten Werten. So unterschätzt die Simulation die reale Packungsdichte der Mischungen mit Korngrößen $< 63\mu\text{m}$ und $> 90\mu\text{m}$ während sie die Packungsdichte der Mischung aus Korngrößen $90 - 200\mu\text{m}$ deutlich überschätzt. Aus diesen Abweichungen lässt sich schließen, dass die tatsächliche Funktion der Agglomerationsneigung einen deutlich flacheren Verlauf haben muss als A1. In A2 haben die kleinsten Korngrößen mit $a_{\text{max}} = 0,25$ eine sehr viel geringere Agglomerationsneigung als in A1. Dafür fällt A2 deutlich langsamer ab und erreicht seinen Nullpunkt erst bei $x = 161\mu\text{m}$. Die zugehörigen Simulationsergebnisse sind in Abbildung 5.15 unter der Bezeichnung „PSDM(A2)→RaSim“ dargestellt. Diese Ergebnisse liegen noch einmal deutlich näher an den gemessenen Werten, die Differenz zwischen Simulation und Messung liegt bei allen 5 Mischungen unter 5%. Mit A2 haben wir also bereits eine gute Approximation der tatsächlichen Agglomerationsneigung der hier verwendeten Sandgemische gefunden.

Die nur auf der veränderten Korngrößenverteilung \hat{q}_0 beruhende Packungsdichte $\Phi_{\hat{q}_0}$ ist ebenfalls in Abbildung 5.15 dargestellt. Sie wird jeweils als dünne graue Linie über den „PSD-Morph→RaSim“ Simulationsergebnissen angezeigt. In $\Phi_{\hat{q}_0}$ wird das zusätzliche, in den Agglomeraten eingeschlossene Hohlraumvolumen nicht berücksichtigt. Wie man sieht, führt die durch die Agglomeration veränderte Korngrößenverteilung alleine nur zu einer geringfügigen Senkung der Packungsdichte und in einigen Fällen sogar zu einer leichten Steigerung. Den größten Beitrag zur Senkung der Packungsdichte liefert somit das in den Agglomeraten eingeschlossene zusätzliche Hohlraumvolumen. Das bedeutet, dass der genaue Verlauf der agglomerierten Korngrößenverteilung $\hat{q}(x)$ nur einen geringen Einfluss auf die Senkung der Packungsdichte hat. Entscheidender ist der Anteil der Agglomerate am Gesamtvolumen der Packung V_h , der aus $h(x)$ berechnet wird, und die innere Raumauffüllung der Agglo-

merate ρ . Die in Abbildung 5.12 dargestellten „Zacken“ in den Verteilungen $\hat{q}(x)$ und $h(x)$ können also vernachlässigt werden. Eine Glättung der Verteilungskurven würde die Simulationsergebnisse nur unwesentlich verändern, solange der in $h(x)$ enthaltene Gesamtanteil der Agglomerate unverändert bleibt .

Kapitel 6

Simulation des Wachstums natürlicher Muster

In der Natur gibt es eine Vielzahl unterschiedlicher Wachstumsprozesse wie beispielsweise die Entstehung einer Schneeflocke, die Ablagerung von Rußteilchen im Inneren eines Schornsteins, die Ausbreitung eines Tumors oder das Wachstum einer Bakterienkolonie in einer Petrischale. Dabei können sehr komplexe - oft auch fraktalartige - Strukturen entstehen, die sich meist nur schwer vermessen oder mathematisch beschreiben lassen. Den Wachstumsprozessen selbst liegen aber oft nur sehr einfache Algorithmen zugrunde, wodurch es möglich ist, die Strukturen virtuell am Computer nachzubilden. Das eröffnet interessante Einblicke in natürliche Prozesse, die normalerweise nur schwer zugänglich sind.

Dieses Kapitel stellt einige Algorithmen vor, mit denen sich das Wachstum natürlicher Muster simulieren lässt. Besonders wird dabei auf die DLA-Algorithmen mit ihren verschiedenen Versionen eingegangen. Anschließend wird die Implementierung einer speziell angepassten Version eines DLA-Algorithmus beschrieben, der als Agglomeratengenerator für die PSD-Morph Simulation eingesetzt wird.

6.1 Der DLA Algorithmus

Mit der Abkürzung „DLA“ wird das Phänomen des diffusionsbegrenzten Wachstums (engl. Diffusion Limited Aggregation) bezeichnet. Damit ist die Ablagerung feiner Partikel gemeint, die durch ein Medium (ein Gas oder eine Flüssigkeit) diffundieren. Die brownische Molekularbewegung in diesem Medium versetzt die Partikel in eine zufällige, ungerichtete Bewegung. Kommen sich dabei zwei Partikel nahe genug, dass die interpartikulären Haftkräfte wirken können, bleiben sie aneinander haften und bilden einen Cluster. Durch das Anhaften weiterer Partikel wächst dieser Cluster an.

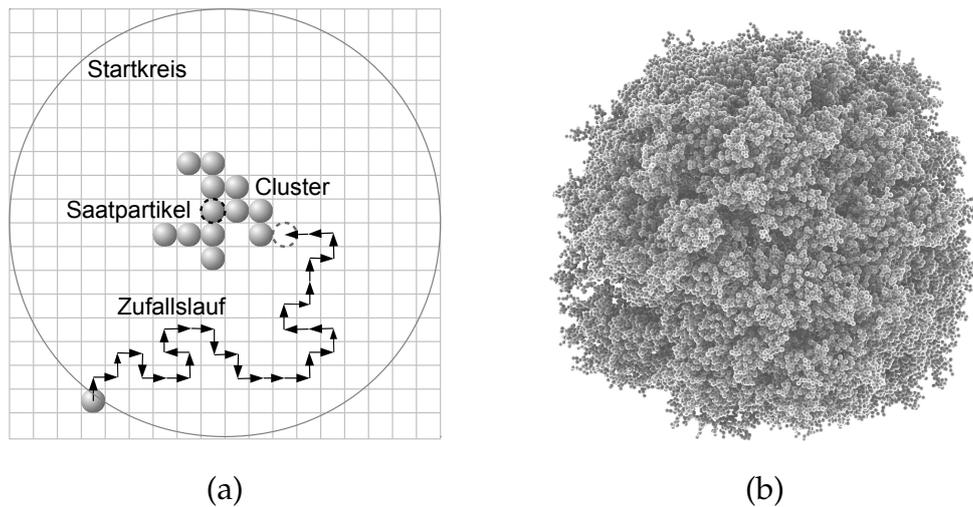


Abbildung 6.1: Prinzip des DLA-Algorithmus (a) und ein per DLA-Algorithmus erzeugter Cluster (b)

Eine erste Beschreibung eines Algorithmus, der das diffusionsbegrenzte Wachstum nachbildet, findet sich in der Arbeit von Witten und Sander [WS81]. In der Literatur wird deshalb auch häufig der Begriff Witten-Sander Aggregate verwendet. Das Funktionsprinzip des Algorithmus, der allgemein als DLA-Algorithmus bezeichnet wird, ist in Abbildung 6.1 dargestellt. In seiner Grundform verwendet er ein quadratisches Gitter, auf dem sich die Partikel entlang bewegen. Dabei wird folgendermaßen vorgegangen:

1. Zu Beginn wird ein erstes Partikel - das Saatpartikel - im Zentrum des Gitters platziert.
2. Ein weiteres Partikel wird an einer zufälligen Position in einigem Abstand zum Saatpartikel platziert. Dafür wird in der Regel ein Startkreis (bzw. Startsphäre im 3-dimensionalen Raum) mit ausreichend großem Radius um das Saatpartikel definiert. Die Startposition wird dann zufällig auf diesem Startkreis gewählt.
3. Das zufällig positionierte Partikel absolviert einen Zufallslauf (engl. Random-Walk), dabei wird es in jedem Schritt auf eine zufällig gewählte Nachbarzelle des Gitters verschoben. Entfernt sich das Partikel dabei zu weit vom Saatpartikel, wird es gelöscht. Berührt das laufende Partikel irgendwann das Saatpartikel oder ein anderes bereits abgelagertes Partikel, so dockt es dort an und vergrößert somit den Cluster.
4. Die Punkte 2. und 3. werden solange wiederholt, bis der Cluster die gewünschte Größe hat.

Die dabei entstehenden Cluster sind nicht kompakt. Der Zufallslauf, den die Partikel vor ihrem Andocken durchlaufen, sorgt für eine sehr poröse fraktalartige Struktur. Die ersten Partikel bilden zunächst nur einen kompakten Kern. Die Teile des Kerns, die besonders exponiert sind, werden dann mit höherer Wahrscheinlichkeit von weiteren Partikeln getroffen als solche, die weiter im Inneren liegen. Die exponierten Teile wachsen so zu Ästen an, die sich weiter verzweigen, wenn sie eine bestimmte Größe erreicht haben. Schließlich besteht der Cluster aus einem kompakten Kern in der Mitte, der von vielen, sich immer weiter verzweigenden, Ästen umgeben ist. Jedes neue Partikel trifft auf seinem Zufallslauf mit hoher Wahrscheinlichkeit die äußeren Äste, die dadurch weiter wachsen und die tiefer liegenden Äste abschirmen, diese werden deshalb nur selten getroffen. Der entscheidende Faktor, der zur Entstehung der fraktalartigen Struktur der DLA-Aggregate führt, ist also der Zufallslauf, der die brownische Molekularbewegung simuliert. Diese Strukturen werden deshalb auch als „Brownsche Bäume“ bezeichnet.

Die baumartigen fraktalen Strukturen, die von DLA-Algorithmen erzeugt werden, bilden ein allgemeines Modell für viele sehr unterschiedliche Wachstums- und Ausbreitungsprozesse. Einige Beispiele dafür sind: die Bildung von Flocken aus Staub-, Ruß- oder andern feinen Partikeln; das Wachstum von Schneeflocken oder anderen Dendriten wie auskristallisierte Mangan-Oxide auf Gesteinsoberflächen [Hal00b] sowie das Ausbreitungsmuster einer Flüssigkeit mit niedriger Viskosität (Wasser) innerhalb einer hoch viskosen Flüssigkeit (Öl). Das letztgenannte Phänomen wird als „Hele-Shaw-Fluss“¹ bezeichnet und tritt z.B. bei der sekundären Rohölgewinnung auf [Hal00b]. Dabei wird Wasser in eine Öl führende poröse Gesteinsschicht (häufig Sandstein) injiziert. Der Ausbreitungsprozess des Wassers im Öl kann im Labor in einer sogenannten „Hele-Shaw-Zelle“ nachgebildet werden. Sie besteht aus zwei Glasscheiben, die an den Rändern aufeinander gepresst werden. Eine der beiden Scheiben besitzt in der Mitte ein Öffnung, durch die zuerst die hoch viskose Flüssigkeit gepresst wird. Anschließend wird das Wasser (zur besseren Sichtbarkeit meist mit Farbstoff vermischt) injiziert. Das Wasser durchdringt die hoch viskose Flüssigkeit an einigen bestimmten Stellen, die durch zufällig verteilte Schwankungen innerhalb der Versuchsanordnung verursacht werden. An diesen Stellen breitet sich das Wasser in Form von Ästen aus, die sich schließlich immer weiter verzweigen. Dadurch entsteht ein fraktales Gebilde, dessen Struktur der eines DLA-Clusters entspricht.

Aufgrund der vielen verschiedenen Anwendungsgebiete existieren eine Vielzahl unterschiedlicher Arbeiten über DLA-Cluster, so dass sie zu den am besten untersuchten fraktalen Strukturen zählen. Die meisten Arbeiten beschreiben die Eigenschaf-

¹nach Henry Selby Hele-Shaw, 1854 - 1941, britischer Ingenieur

ten von 2-dimensionalen, in weniger Fällen auch von 3-dimensionalen, DLA-Clustern [Mea85, Mea95]. Höher dimensionale DLA-Cluster wie in [TM89] sind weniger gut untersucht. Die fraktale Dimension von DLA-Clustern konnte für 2-dimensionale Strukturen mit $D \approx 1,71$ und für 3-dimensionale Strukturen mit $D \approx 2,5$ bestimmt werden. Bei höheren Raumdimensionen zeigt sich zwischen der fraktalen Dimension D und der Raumdimension d der Zusammenhang: $D \rightarrow d - 1$ [Hal00b]. Diese Werte stehen für eine hohe Komplexität und Porosität der DLA-Cluster. Doch neben der vorgestellten Grundform existieren noch verschiedene Varianten und Erweiterungen des DLA-Algorithmus, die andere (nicht-quadratische) Gitterformen benutzen oder ganz ohne Gitter auskommen und zusätzliche Faktoren neben dem Zufallslauf berücksichtigen. Diese Algorithmen können unterschiedliche - und auch dichter gepackte - Varianten der fraktalen Struktur der DLA-Cluster erzeugen. Einige von ihnen werden im folgenden Kapitel beschrieben.

6.1.1 Varianten des DLA-Algorithmus

Die Verwendung eines Gitters für den DLA-Algorithmus kann ein Nachteil sein, da sich die Partikel so in jedem Schritt des Zufallslaufs nur um die Kantenlänge einer Gitterzelle bewegen können. Um eine bessere Laufzeit der Simulation zu erreichen, insbesondere bei der Generierung sehr großer Cluster, ist es sinnvoll, die Länge jedes Schrittes im Zufallslauf an die Entfernung der Partikel zum Cluster anzupassen. So können sich die Partikel in großer Distanz mit entsprechend großen Schritten dem Cluster nähern. Der Zufallslauf benötigt dadurch im Mittel weniger Schritte. Nahe am Cluster ist dann die Schrittlänge entsprechend reduziert. So benötigen die Partikel dort wieder etwas mehr Schritte, um die restliche Distanz bis zum Andocken am Cluster zurück zu legen. Ihre dadurch erhöhte Beweglichkeit gibt ihnen die Möglichkeit, auch an den exponierten Ästen vorbei in die tiefer liegenden Hohlräume vorzudringen. Ein entsprechend angepasster Algorithmus wird in [Mea85] beschrieben. Die Bewegung der Partikel ist dort nicht mehr streng an das Gitter gebunden, so dass in einem Schritt auch mehrere Zellen übersprungen werden können. Dazu wird in jeder Zelle des Gitters die maximale Weite gespeichert, um die sich ein Partikel, dessen Mittelpunkt in dieser Zelle liegt, bewegen kann ohne den Cluster zu berühren. Diese Weite ist dann die Distanz, die der nächste Schritt des Zufallslaufs zurücklegt. Sobald sich der Cluster durch das Andocken eines weiteren Partikels vergrößert, werden die Werte in den Gitterzellen entsprechend angepasst. Versionen des DLA-Algorithmus, deren Zufallslauf nicht auf einem Gitter erfolgt, werden in der Literatur als **Off-Lattice** bezeichnet.

Eine weitere Besonderheit, die sich aus der Verwendung eines Gitters für den Zufallslauf ergibt, ist die Anisotropie in der Form der entstehenden Cluster. Durch die



Abbildung 6.2: Anisotroper DLA-Cluster aus 400 000 kugelförmigen Partikeln

Art des verwendeten Gitters sind die möglichen Bewegungsrichtungen für die Partikel vorgegeben. Äste des Clusters, die in eine dieser Richtungen wachsen, werden deshalb häufiger von Partikeln getroffen und wachsen entsprechend schneller. Bei einem quadratischen Gitter entstehen so diamantförmige Cluster wie in Abbildung 6.2. Der Effekt tritt besonders deutlich bei sehr großen Clustern (>100000 Partikel) auf, wobei die Form des Clusters von der Art des Gitters abhängt [BB85]. Anisotrope DLA-artige fraktale Strukturen treten auch in einigen realen Anwendungen auf, deshalb existieren auch entsprechende Versionen des DLA-Algorithmus. In [MV87] ist ein Algorithmus beschrieben, bei dem durch die Festlegung eines Grenzwertes die Winkel, in denen ein Partikel am Cluster andocken kann, eingeschränkt werden. Dadurch entstehen sternförmige Cluster, deren Äste radial aus dem Zentrum herauswachsen. Das erlaubt die Simulation des Wachstums von Formen, die bei Experimenten mit Flüssigkristallen beobachtet wurden. Sollen aber isotrope Formen generiert werden, dürfen keine Beschränkungen für die Richtung der Schritte im Zufallslauf existieren. Im Kapitel 6.3 wird eine Methode beschrieben, die ein gleichmäßiges Wachstum der DLA-Cluster in alle Richtungen ermöglicht.

In einigen Anwendungen haben, neben den interpartikulären Haftkräften, auch andere Faktoren Einfluss auf das Wachstum der Aggregate. Ein Beispiel dafür sind kolloide Suspensionen, die eine Art von dispersen Stoffsystemen bilden. Die kontinuierliche Phase bildet hier stets eine Flüssigkeit. Die disperse Phase bilden Tröpfchen oder feste Partikel deren Korngröße im Micro- oder Nanometerbereich liegt. Solche Suspensionen treten beispielsweise bei Nahrungsmitteln (Pudding, Sahne) bei Kosmetika und Medikamenten (Cremes, Salben) und bei Baustoffen wie Beton (Zusatzstoffe) auf. Aufgrund ihrer geringen Größe unterliegen die Partikel der dispersen Phase der

brownschen Molekularbewegung und können auch Agglomerate bilden. Die Interaktionen an der Grenzschicht zur flüssigen Phase können aber dazu führen, dass die Partikel nicht bei jeder Berührung mit einem anderen Partikel anhaften. Das führt zu einem langsameren Anwachsen der Agglomerate und einer dichteren Struktur als dies bei DLA-Clustern der Fall ist. Diese Art des Wachstums wird als „Reaction-Limited Aggregation“ (RLA) [RSK05] bezeichnet. Sie kann mit einem entsprechenden Algorithmus simuliert werden, der eine angepasste Version des DLA-Algorithmus ist. Dabei wird ein Partikel bei Annäherung an den Cluster nicht sofort andockt, sondern gemäß einer festgelegten Wahrscheinlichkeit. So lässt sich der Einfluss unterschiedlicher anziehender und abstoßender Kräfte mit kurzer Reichweite, wie sie an der Grenzschicht zwischen disperser und kontinuierlicher Phase wirken, simulieren. Da die Partikel nicht zwangsläufig sofort beim ersten Kontakt andocken, bewegen sie sich länger und können so mit höherer Wahrscheinlichkeit an den exponierten Ästen des Clusters vorbei in die tiefer liegenden Hohlräume vordringen. Die so entstehenden Agglomerate wachsen langsamer und haben eine höhere Dichte.

Eine Variante des DLA-Algorithmus, die die Bindungsfähigkeit unterschiedlicher Partikeltypen berücksichtigt, wird in [Mue04] vorgestellt. Dabei werden alle Partikel bestimmten Typen zugeordnet. Zusätzlich wird eine Typenfunktion definiert, die die Verträglichkeit von je zwei Partikeln gemäß vorgegebener Andockregeln liefert. Zwei Partikel werden nur dann miteinander verbunden, wenn ihre Typen miteinander verträglich sind. Diese Version des Algorithmus wird als „Typed DLA“ bezeichnet, mit ihm können auch chemische Bindungseigenschaften unterschiedlicher Moleküle oder Atome simuliert werden. In [Mue04] werden Simulationsergebnisse vorgestellt die zeigen, dass die fraktale Dimension D mit der Anzahl der Partikeltypen steigt. Abhängig von den Andockregeln, die in der Typenfunktion implementiert sind, können sehr unterschiedlich geformte Cluster entstehen.

6.1.2 Fazit

Als Agglomeratengenerator für die PSD-Morph Simulation wird eine angepasste Version des DLA-Algorithmus eingesetzt. Ausschlaggebend für diese Entscheidung sind einige besondere Eigenschaften des Algorithmus, insbesondere die Tatsache, dass sich DLA-Cluster auch ohne Gitter - also Off-Lattice - erzeugen lassen. Die Primärpartikel, die in jedem Iterationsschritt der PSD-Morph Simulation zur einer Agglomeratengeneration formiert werden, liegen in einer bestimmten Korngrößenverteilung $g_i(x)$ vor, die aus der vorgegebenen Korngrößenverteilung der Mischung $q_0(x)$ und der Funktion der Agglomerationsneigung $a(x)$ berechnet wird. Die Durchmesser des größten und kleinsten Partikels innerhalb eines Agglomerates können deshalb erheblich dif-

ferieren. Ein regelmäßiges Gitter mit gleichgroßen Zellen ist hier ungeeignet für die Anordnung der Partikel. Wie der Zufallslauf in der hier benutzten Version des DLA-Algorithmus berechnet wird, sowie weitere Details der Implementierung, sind in Kapitel 6.3 beschrieben.

6.2 Weitere Modelle

Der Einsatz eines DLA-Algorithmus als Agglomeratengenerator wird von der PSD-Morph Simulation nicht zwingend festgelegt, er kann auch durch einen anderen geeigneten Algorithmus ersetzt werden. Das ist beispielsweise dann sinnvoll, wenn die Auswirkungen der Entstehung einer ganz bestimmten Agglomeratstruktur auf die Korngrößenverteilung eines Gemisches simuliert werden soll. Lässt sich diese Struktur mit einem anderen Algorithmus besser wiedergeben als mit dem DLA-Algorithmus, ist ein Austausch sinnvoll.

Ein interessante Alternative zur Generierung von virtuellen Agglomeraten stellt vor allem die „Diskrete Elemente Methode“ (DEM) dar, die bereits in Kapitel 3.2.1 vorgestellt wurde. Da sie die physikalischen Vorgänge realitätsnah simulieren kann, erlaubt sie die Untersuchung der Auswirkung unterschiedlicher Kräfte auf die Struktur der entstehenden Agglomerate. Es muss jedoch beachtet werden, dass die DEM einen wesentlich höheren Rechenaufwand erfordert als beispielsweise ein DLA-Algorithmus. Da die PSD-Morph Simulation aber in jedem Iterationsschritt eine große Anzahl von Agglomeraten benötigt, würde dies die Simulation von stark agglomerierenden Mischungen erheblich erschweren.

Neben dem DLA-Algorithmus existieren einige weitere statistische Wachstumsalgorithmen, die auf dem gleichen Grundprinzip beruhen, der Belegung von Zellen in einem Gitter gemäß vorgegebener auf Zufallsereignissen beruhender Regeln. Einige von ihnen können fraktale Strukturen erzeugen. Im Folgenden werden zwei typische Vertreter dieser Algorithmen beschrieben.

6.2.1 Der Eden-Growth Algorithmus

Eine frühe Version eines 2-dimensionalen Wachstumsalgorithmus wurde von Murray Eden [Ede61] beschrieben. Er wird als Eden-Growth bezeichnet und dient der Untersuchung der Eigenschaften des Zellwachstums, insbesondere der dabei entstehenden Strukturen. Als Grundlage dient ein quadratisches 2-dimensionales Gitter (das jedoch auch durch eine andere Gitterform oder höher dimensionale Gitter ersetzt werden kann). Zum Zeitpunkt der Initialisierung wird eine Zelle im Zentrum des Gitters

belegt. Anschließend wird eine der angrenzenden Nachbarzellen zufällig ausgewählt und belegt. Dieser Vorgang wird nun wiederholt. Die so entstehende Struktur wächst weiter, indem sequentiell weitere Gitterzellen an ihrer Oberfläche belegt werden.

Der Eden-Growth Algorithmus kann als eine vereinfachte Grundform des DLA-Algorithmus angesehen werden, bei dem ein Partikelcluster durch die Anlagerung weiterer Partikel auf seiner Oberfläche anwächst. Beim Eden-Growth müssen die Partikel aber vor dem Andocken am Cluster keinen Zufallslauf absolvieren, ihre Andockpunkte werden direkt gemäß einer vorgegebenen Wahrscheinlichkeitsverteilung bestimmt. Die entstehende Struktur hängt von dieser Wahrscheinlichkeitsverteilung ab. Werden die Nachbarzellen des Clusters gemäß einer Gleichverteilung ausgewählt, entstehen runde kompakte Strukturen, die nur an ihrer Oberfläche kleine fraktalartige Verästelungen aufweisen. Diese Verästelungen wachsen aber mit zunehmender Größe des Clusters nicht weiter.

Bei einem DLA-Algorithmus verteilt sich die Andockwahrscheinlichkeit für neue Partikel, bedingt durch den Zufallslauf, in besonderer Weise über die Clusteroberfläche. Exponierte Teile des Clusters, das heißt kleine Verästelungen auf der Oberfläche, werden mit höherer Wahrscheinlichkeit getroffen als tiefer liegende Senken. Grundsätzlich ließe sich diese Form des Clusterwachstums auch direkt mit dem Eden-Growth Algorithmus simulieren, es ist jedoch unklar, wie die dafür notwendige Wahrscheinlichkeitsverteilung für das Andocken der Partikel auf der Clusteroberfläche bestimmt werden soll.

6.2.2 Perkolation

Die Perkolation ist eine Methode, die die Simulation der Ausbreitung von Mustern in einer vorgegebenen Struktur ermöglicht. Die Grundlage bildet auch hier wieder ein Gitter. Bei der einfachsten Form der Perkolation werden alle Zellen des Gitters nacheinander durchlaufen und gemäß einer vorgegebenen Wahrscheinlichkeit belegt. Diese Form wird auch als Knotenperkolation bezeichnet. Eine Alternative dazu ist die Kantenperkolation, bei der alle Kanten des Gitters durchlaufen und gemäß der vorgegebenen Wahrscheinlichkeit belegt werden. In beiden Fällen ist die Belegung einer Zelle bzw. Kante unabhängig vom Zustand der Nachbarzellen bzw. -kanten. Auf diese Weise entstehen auf dem Gitter viele fraktale Gebilde, deren Größe von der Höhe der Wahrscheinlichkeit abhängt, mit der eine Zelle oder Kante belegt wurde. Mit diesen Formen der Perkolation kann beispielsweise die Ausbreitung von Rissen im Felsgestein, die Aneinanderreihung von Molekülen zu Makromolekülen oder die Atomstruktur in Metalllegierungen simuliert werden.

In [BJ00] werden die fraktalen Strukturen von Tumoren und die Möglichkeit der

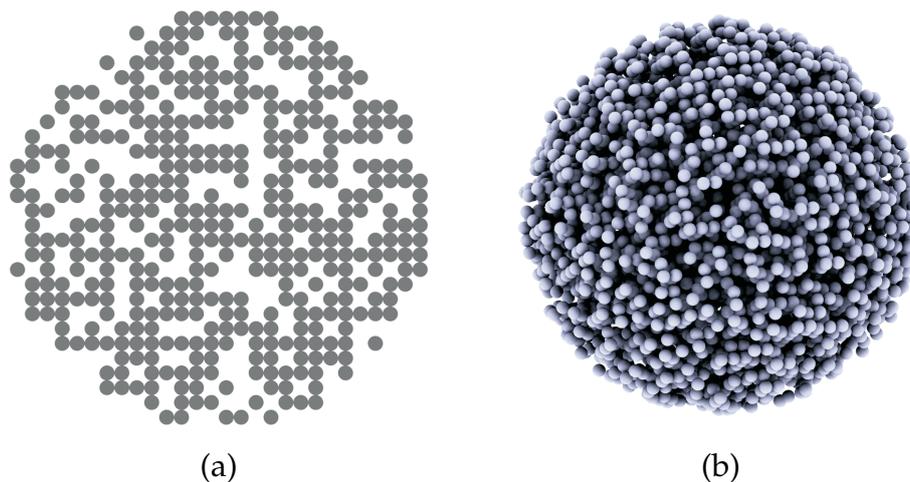


Abbildung 6.3: Perkolationscluster in 2D (a) und 3D (b)

Simulation dieser Strukturen mit Hilfe verschiedener statistischer Wachstumsalgorithmen untersucht. Dabei wird eine sogenannte Invasionsperkolations zur Simulation der irregulären Gefäßstrukturen innerhalb von Tumoren vorgeschlagen. Bei dieser Form der Perkolations wird jede Zelle des Gitters mit einem zufälligen Wert versehen, der als Stärke der Zelle interpretiert werden kann. Um das Wachstum zu starten, wird eine erste Zelle des Gitters belegt. Während des Wachstumsprozesses werden dann in jedem Iterationsschritt alle Nachbarzellen der bereits belegten Gitterzellen durchsucht. Dabei wird die Zelle ausgewählt und belegt, die den geringsten Wert hat. Auf diese Weise entsteht eine zusammenhängende fraktale Struktur, die viele Löcher unterschiedlicher Größe umschließt. Die fraktale Dimension für solche 2-dimensionalen Strukturen beträgt $D \approx 1,89$ [BJ00]. Die fraktale Dimension für den kürzesten Pfad zwischen zwei Punkten in einem Perkolationscluster beträgt laut [HS88] für den 2-dimensionalen Fall $D \approx 1,13$ und für den 3-dimensionalen Fall $D \approx 1,34$.

Perkolationsalgorithmen sind weniger geeignet um das Wachstum von Agglomeraten zu simulieren. Sie können aber eingesetzt werden, um besonders poröse Partikelpackungen zu generieren. Solche Packungen können beispielsweise für die Simulation von Filtern oder Katalysatoren verwendet werden. Dort braucht man Strukturen die eine hohe Porosität in Verbindung mit einer großen Oberfläche bieten. In Abbildung 6.3 ist ein Beispiel für solch eine Packung dargestellt. Hierfür wurde ein einfacher Algorithmus implementiert, der die kugelförmigen Partikel per Knotenperkolations in einem quadratischen Gitter angeordnet. Die Anordnung entspricht damit der einer kubischen Kugelpackung (siehe Abbildung 2.3). Der Algorithmus ließe sich aber auch auf eine zufällig dichte Partikelpackung anwenden. Dort bilden die Kugeln ebenfalls eine - wenn auch sehr irreguläre - Gitterstruktur.

6.3 Implementierungen

Dieses Kapitel beschreibt die Implementierung des in Kapitel 5.4 verwendeten Agglomerategenerators. Grundsätzlich basiert dieser Generator auf dem bereits beschriebenen DLA-Modell. Der PSD-Morph Algorithmus stellt jedoch einige besondere Anforderungen bezüglich der zu generierenden Agglomerate:

- Die Agglomerate müssen sich aus Kugeln unterschiedlichen Durchmessers zusammensetzen lassen. Die Häufigkeit der einzelnen Durchmesser soll dabei der vorgegebenen Verteilung $g_i(x)$ folgen.
- Das Wachstum eines Agglomerates muss stoppen, sobald eine festgelegte Abbruchgröße überschritten wird.
- Es muss eine Möglichkeit geben, die innere Raumausfüllung ρ der Agglomerate zu bestimmen.

Darüber hinaus ist es wichtig, dass für jede Agglomerategeneration i , die der PSD-Morph Algorithmus benötigt, eine ausreichend hohe Anzahl Agglomerate generiert werden kann, um aussagekräftige Werte für die mittlere Agglomerategröße $d(i)$ und die mittlere Anzahl der enthaltenen Primärpartikel $m(i)$ zu erhalten. Dieses und die folgenden Kapitel beschreiben die verwendeten Techniken, mit denen alle Anforderungen des PSD-Morph erfüllt werden können. Wie bei der Packungssimulation „Ra-Sim“ gehen wir auch hier stets davon aus, dass die Primärpartikel, aus denen ein Agglomerat zusammengesetzt wird, kugelförmig sind. Da dieser Agglomerategenerator auch als eigenständige Anwendung zur Generierung von DLA-Clustern betrieben werden kann, verwenden wir in den folgenden Kapiteln den Begriff DLA-Cluster bzw. Cluster statt Agglomerat.

6.3.1 Ein Off-Lattice DLA-Algorithmus

Ein wichtiges Element der DLA-Algorithmen ist der Zufallslauf. Jede Kugel muss diesen absolvieren, bevor sie am Cluster angedockt werden kann. Dieser Vorgang ist zu zeitaufwändig, wenn sehr große Cluster mit mehr als 100 000 Kugeln erzeugt werden sollen. Ein weiteres Problem liegt in der Benutzung eines Gitters: da der Zufallslauf ausreichend Platz benötigt, muss es deutlich größer sein als der Cluster. Für sehr große 3-dimensionale Cluster kann es schnell auf mehrere Millionen Zellen anwachsen. Werden die Gitterzellen benutzt, um zusätzliche Daten zu speichern, wie z.B. den Abstand zum Cluster [Mea85], entsteht eine Datenstruktur, die leicht mehrere hundert Megabyte umfassen kann. Um diese Probleme zu vermeiden ist es besser, auf das Gitter zu verzichten und den Zufallslauf statt dessen auf andere Weise zu berechnen.

Algorithmus 5 : Off-Lattice DLA-Algorithmus

Input : Die Korngrößenverteilung der Primärpartikel ($g_i(x)$)

Input : Die (*abbruchgroesse*) für das zu erzeugende Agglomerat

Output : Der Vektor, der das fertige Agglomerat enthält (**K**)

```
1  $k = \text{sampling}(g_i(x))$  // Erzeugen der Saatkugel
2 verschiebeNach( $k, (0,0,0)$ ) // Platzieren der Saatkugel
3  $clusterRadius = k.radius$  // Radius des Clusters initialisieren
4  $\mathbf{K} \leftarrow k$  // Saatkugel dem Vektor hinzufügen
5 repeat
6    $k = \text{sampling}(g_i(x))$  // Erzeugen einer weiteren Kugel
7    $neustart = \mathbf{true}$ 
8    $angedockt = \mathbf{false}$ 
9   while Not  $angedockt$  do
10    if  $neustart$  then
11       $startSphaerenRadius = 2 * clusterRadius + k.radius$ 
12      zufallsPosAufSphäre( $startSphaerenRadius, k$ )
13       $neustart = \mathbf{false}$ 
14    else
15       $angedockt = \text{zufälligerSchritt}(k)$ 
16    if Not  $angedockt$  then
17      // Hat sich die Kugel zu weit vom Cluster entfernt?
18      if  $distanz(k, (0,0,0)) > 10 * startSphaerenRadius$  then
19         $neustart = \mathbf{true}$ 
20      else
21        // Radius des Clusters gegebenenfalls korrigieren
22        if  $distanz(k, (0,0,0)) + k.radius > clusterRadius$  then
23           $clusterRadius = distanz(k, (0,0,0)) + k.radius$ 
24     $\mathbf{K} \leftarrow k$  // Kugel dem Vektor hinzufügen
25 until  $clusterRadius \geq \text{abbruchgroesse}$ 
26 return  $\mathbf{K}$ 
```

Wir verwenden hier einen Off-Lattice DLA-Algorithmus, dessen Kern in Algorithmus 5 aufgelistet ist. Er arbeitet gemäß den Beschreibungen für DLA-Algorithmen aus der Literatur. Zunächst wird ein Saatpartikel erzeugt und im Zentrum platziert (Zeilen 1-4). Danach folgt die äußere Schleife des Algorithmus (Zeilen 5-23), in der dem Agglomerat solange weitere Kugeln hinzugefügt werden (Zeilen 6 und 22), bis die vorgegebene Abbruchgröße überschritten wird (Zeile 23). Alle Kugeln (nach dem Saatpartikel) müssen einen Zufallslauf absolvieren, der in der inneren Schleife des Algorithmus (Zeilen 9-21) ausgeführt wird. Jede neue Kugel wird von einer zufälligen Position auf einer Startsphäre (Zeile 12) gestartet, die in ausreichendem Abstand den Cluster umgibt (Zeile 11). Die Berechnung der einzelnen Schritte des Zufallslaufs werden von der Funktion `zufälligerSchritt` durchgeführt. Berührt eine Kugel nach einem Zufallsschritt den Cluster, wird sie angedockt und die Simulation fährt mit der nächsten Kugel fort bzw. sie wird beendet wenn das Agglomerat bereits die Abbruchgröße überschritten hat.

Auf einem Gitter (On-Lattice Version) müsste die Funktion `zufälligerSchritt` nur eine der 6 Nachbarzellen der Kugel auswählen und sie dorthin verschieben. Ohne Gitter muss der zufällige Schritt jedoch auf andere Weise berechnet werden. Ähnlich der in [Mea85] beschriebenen Methode verwenden wir hier für die Weite eines zufälligen Schrittes die maximale Distanz, die die Kugel auf jeden Fall zurück legen kann, bevor sie den Cluster berührt. Der Vorteil liegt darin, dass sich die Kugel in großer Entfernung zum Cluster in entsprechend großen Schritten bewegt und sich so dem Cluster schnell annähern kann. Die mittlere Anzahl Schritte, die der Zufallslauf zum Andocken einer Kugel braucht, ist somit unabhängig vom Durchmesser der Startsphäre.

Wir beschränken die möglichen Bewegungen, die die laufende Kugel in jedem Schritt des Zufallslaufs machen kann, auf eine Verschiebung parallel zu einer der 3 Raumachsen. Wie in Abbildung 6.4 dargestellt, gestaltet sich dann die Berechnung der Distanz zu den bereits zum Cluster gehörenden Kugeln (im Folgenden Clusterkugeln genannt) sehr einfach. Da sich die Kugel dabei entweder vorwärts oder rückwärts bewegen kann, ergeben sich insgesamt 6 mögliche Richtungen pro Zufallsschritt. Die möglichen Bewegungsrichtungen jedes Zufallsschrittes sind also die selben wie in der On-Lattice Version des Algorithmus. Für jede der 6 möglichen Richtungen wird überprüft, welche Clusterkugeln getroffen werden können. Zu diesen wird dann die Distanz ermittelt. Da sich die laufende Kugel immer parallel zu jeweils einer der 3 Raumachsen bewegt, wird sie die Clusterkugeln in der Regel nicht zentral treffen, sondern seitlich versetzt. Wie in diesem Fall die Distanz d berechnet werden kann, ist in Abbildung 6.4 dargestellt. Die kleinste der so ermittelten Distanzen wird dann als Weite des nächsten Zufallsschrittes benutzt. Befindet sich die laufende Kugel irgendwann

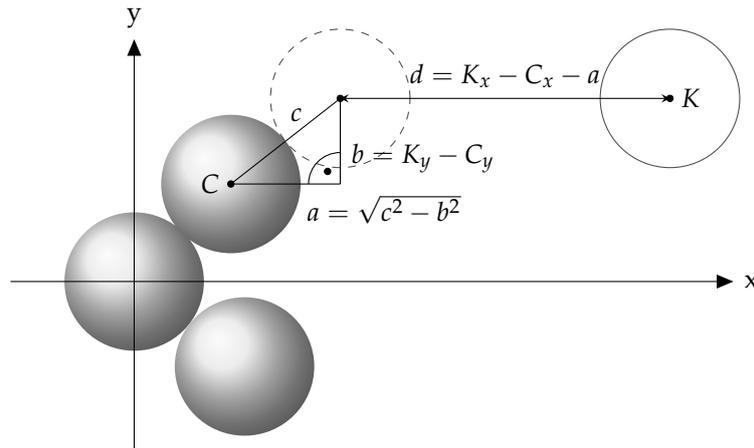


Abbildung 6.4: Berechnung der Distanz d zwischen der laufenden Kugel K und der bereits zum Cluster gehörenden Kugel C

in einer so ungünstigen Position, dass sie in keiner der 6 möglichen Richtungen den Cluster treffen kann, wird die Weite des nächsten Schrittes auf die Hälfte des Radius der Startsphäre gesetzt. Diese Schrittweite ist in jedem Fall groß genug, um der Kugel die Möglichkeit zu geben, im nächsten Schritt wieder nahe genug an den Cluster und damit in eine günstigere Position zu kommen. Die Richtung, in die der Schritt führt, wird in jedem Fall zufällig gleichverteilt unter den 6 möglichen Richtungen ausgewählt. Entfernt sich eine Kugel während ihres Zufallslaufs zu weit vom Cluster, wird sie erneut von einer zufälligen Position auf der Startsphäre gestartet (Zeilen 17, 18).

Diese Art der Berechnung des Zufallslaufs bietet zwei wesentliche Vorteile. Zum einen benötigt der Zufallslauf so im Mittel weniger Schritte (im Vergleich zur On-Lattice Version des Algorithmus), um sich dem Cluster zu nähern. Außerdem können so - ohne weitere Anpassungen - auch Agglomerate aus unterschiedlich großen Kugeln zusammengesetzt werden. Bei Verwendung eines Gitters hat jeder Zufallsschritt die Weite einer Gitterzelle, ist also immer konstant. Soll das Agglomerat aus Kugeln bestehen die sowohl größer als auch kleiner sein können als eine Gitterzelle, muss der Zufallslauf in der Nähe des Clusters entsprechend korrigiert werden, denn viele der von den Clusterkugeln belegten Zellen sind dann nur zum Teil ausgefüllt. Bei der hier vorgestellten Off-Lattice Version sind keine weiteren Anpassungen notwendig, da die Weite jedes Zufallsschrittes individuell berechnet wird. Die Größe jeder neu hinzugefügten Kugel wird gemäß der vorgegebenen Korngrößenverteilung der agglomerierenden Primärpartikel $g_i(x)$ bestimmt (Zeilen 1 und 6). Dafür wird das „Simple Random Sampling“ eingesetzt, dass bereits in Kapitel 3.1.1 vorgestellt wurde.

6.3.2 Erweiterungen des Algorithmus

Um die Entstehung anisotroper Formen während des Clusterwachstums zu vermeiden und um eine Möglichkeit zu schaffen, auch Cluster mit einer höheren inneren Dichte (RLA-Version) zu generieren, wurde der Algorithmus mit einigen Erweiterungen ausgestattet, die im Folgenden beschrieben sind.

Zufällige Rotation des Clusters

Beim vorgestellten Algorithmus verlaufen die Schritte im Zufallslauf jeweils parallel zu einer der 3 Raumachsen. Wie aber bereits erklärt wurde, entstehen anisotrope Clusterformen (siehe Abbildung 6.2) wenn die möglichen Bewegungsrichtungen des Zufallslaufs eingeschränkt werden. Um dies zu vermeiden, muss diese Einschränkung aufgehoben werden. Eine einfache und effiziente Möglichkeit besteht darin, denn Cluster in regelmäßigen Abständen zu rotieren. Die anisotrope Form entsteht dadurch, dass Äste des Clusters, die in einer der möglichen Bewegungsrichtungen des Zufallslaufs liegen, häufiger getroffen werden und dadurch schneller wachsen als andere Clusterteile. Die regelmäßige Rotation des Clusters ändert immer wieder die Richtungen seiner Äste und sorgt so für ein richtungsunabhängiges Wachstum.

Bei dieser Methode ist keine Änderung der Berechnung des Zufallslaufs notwendig. Die Auswirkungen auf das Laufzeitverhalten des Algorithmus sind vernachlässigbar. Da eine anisotrope Form erst nach dem Andocken einer ausreichend großen Anzahl von Kugeln am Cluster entstehen kann genügt es, die Rotation nach jeweils 100 angedockten Kugeln durchzuführen. Auch die Rotation selbst kann sehr effizient durchgeführt werden. Wir greifen dafür auf die mathematischen Methoden der 3D Computergrafik [Len04] zurück. Die Rotation eines Punktes im 3-dimensionalen Raum um eine beliebige Achse kann so als eine simple Vektor-Matrix-Multiplikation dargestellt werden. Zunächst legen wir für jede der 3 Raumachsen eine Transformationsmatrix fest:

$$R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{bmatrix} \quad \text{mit } \theta_x := \text{rand}(0^\circ, 360^\circ)$$
$$R_y(\theta_y) = \begin{bmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \quad \text{mit } \theta_y := \text{rand}(0^\circ, 360^\circ)$$
$$R_z(\theta_z) = \begin{bmatrix} \cos \theta_z & -\sin \theta_z & 0 \\ \sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{mit } \theta_z := \text{rand}(0^\circ, 360^\circ).$$

Jede der drei Matrizen R_x , R_y und R_z beschreibt eine Rotation um die x-, y- bzw. z-Achse. Die zugehörigen Winkel θ_x , θ_y und θ_z werden zufällig gleichverteilt zwischen 0° und 360° ausgewählt. Anschließend wird aus den drei Matrizen eine Gesamttransformationsmatrix T berechnet:

$$T = R_x \cdot R_y \cdot R_z.$$

Die Multiplikation der drei Matrizen R_x , R_y und R_z entspricht einer Hintereinanderausführung der 3 von ihnen dargestellten Rotationen. Um die Rotation des Clusters durchzuführen genügt es nun, die Vektoren der Mittelpunkte aller Clusterkugeln mit der Matrix T zu multiplizieren. Die Matrix T muss für jede auszuführende Rotation nur einmal berechnet werden. Da die 3 Rotationswinkel θ_x , θ_y und θ_z gemäß einer Gleichverteilung über einem Vollkreis von 360° ausgewählt werden, kann der Cluster jede räumliche Orientierung mit der gleichen Wahrscheinlichkeit einnehmen. Damit wird keine Wachstumsrichtung bevorzugt und es entstehen isotrope Cluster.

Anhaftwahrscheinlichkeit der Primärpartikel

Um auch dichtere Clusterstrukturen, gemäß der Beschreibung der „Reaction-Limited Aggregation“ (RLA), erzeugen zu können, wurde der Algorithmus um eine Anhaftwahrscheinlichkeit der Primärpartikel erweitert. Diese wird vom Benutzer als fester Parameter vorgegeben. Berührt eine laufende Kugel nach einem Zufallsschritt den Cluster, so ist sie - gemäß der Darstellung in Algorithmus 5 - angedockt. Mit dieser Erweiterung wird jedoch nun (per Zufallsgenerator) bestimmt, ob die Kugel entsprechend der festgelegten Anhaftwahrscheinlichkeit tatsächlich angedockt werden soll. Ist dies nicht der Fall, wird die Weite des Zufallsschrittes nachträglich etwas verkürzt, so dass die laufende Kugel den Cluster nicht mehr berührt. Dies kann als ein Abprallen der Kugel vom Cluster interpretiert werden. Der so gewonnene Abstand gibt der Kugel genügend Bewegungsfreiraum, um ihren Zufallslauf fortzusetzen. Der nächste Schritt des Zufallslaufs, kann damit wie in der bisherigen Version berechnet werden.

6.3.3 Clusterkugelverwaltung

Eine Besonderheit des hier vorgestellten Algorithmus ist, dass in jedem Schritt des Zufallslaufs die Distanz der laufenden Kugel zum Cluster berechnet werden muss. In der einfachsten Variante dieser Berechnung werden einfach alle Clusterkugeln darauf überprüft, ob sie von der laufenden Kugel getroffen werden können oder nicht. Damit verschlechtert sich aber das Laufzeitverhalten des Algorithmus mit zunehmender Größe des Clusters. Um das zu vermeiden, wird eine geeignete Datenstruktur zur

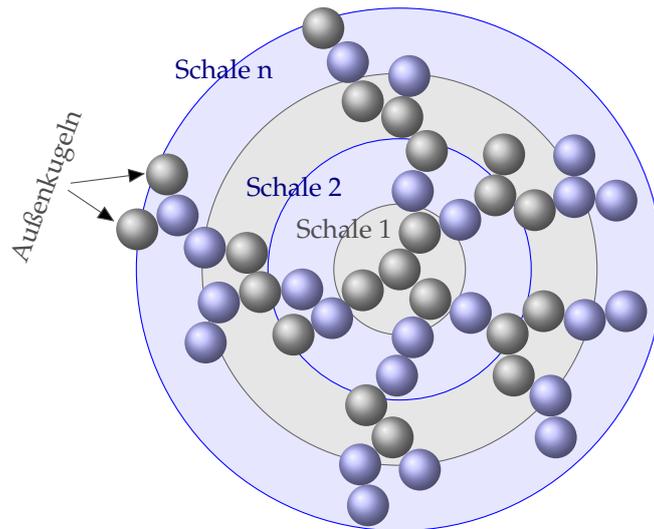


Abbildung 6.5: Schalensystem zur Verwaltung der Clusterkugeln

Verwaltung der Clusterkugeln benötigt, die eine effiziente Berechnung der Distanzen unabhängig von der Größe des Clusters erlaubt.

Das Wachstum eines DLA-Clusters findet vor allem an seinem äußeren Rand statt. Dort befinden sich die exponierten Äste der fraktalen Struktur, die am häufigsten von Kugeln während ihres Zufallslaufs getroffen werden. Wir bezeichnen diesen Bereich als aktive Zone. Die Teile des Clusters, die sich unterhalb dieser aktiven Zone befinden, werden nur selten getroffen und können deshalb kaum weiter wachsen. Diese Tatsache kann man ausnutzen, um die Berechnung der Distanzen für den Zufallslauf effizienter zu gestalten. Dazu werden Ausschnitte des Clusters jeweils mit einer eigenen Hüllkugel umgeben. Kann eine dieser Hüllkugeln im aktuellen Schritt des Zufallslaufs nicht getroffen werden, braucht der in ihr liegende Ausschnitt des Clusters nicht überprüft zu werden. Die Hüllkugeln werden so angeordnet, dass sie den Cluster in Zonen mit unterschiedlichen Trefferwahrscheinlichkeiten einteilen. Das ergibt ein Schalensystem, wie es in Abbildung 6.5 dargestellt wird, sein Aufbau ähnelt einer Zwiebel.

Definition 6.1 (Schalensystem S) Ein Schalensystem S dient der Aufteilung der Kugeln eines DLA-Clusters. Es besteht aus Hüllkugeln unterschiedlichen Durchmessers, deren gemeinsamer Mittelpunkt im Koordinatenursprung liegt. Der Inhalt jeder Hüllkugel bildet eine Schale des Systems. Die Schalen sind von innen nach außen (also mit steigendem Durchmesser der jeweiligen Hüllkugel) nummeriert. Eine Clusterkugel gehört zur Schale i , wenn sie vollständig innerhalb der betreffenden Hüllkugel liegt aber nicht zu einer tiefer liegenden Schale ($i - 1, \dots, 1$) gehört.

Kugeln die zu keiner Schale gehören, weil sie außerhalb der größten Hüllkugel liegen, werden als Außenkugeln bezeichnet.

Die Berechnung eines Schritts im Zufallslauf, die in Algorithmus 6 dargestellt wird, basiert darauf, die Clusterkugeln der einzelnen Schalen von außen nach innen abzuarbeiten. Dabei werden die Hüllkugeln benutzt, um zu prüfen, ob die Kugeln einer Schale im aktuellen Schritt des Zufallslaufs getroffen werden können und weitere Überprüfungen notwendig sind oder nicht. Die Funktion benutzt ein Schalensystem \mathcal{S} , um für

Algorithmus 6 : Die Funktion zufälligerSchritt

Input : Ein Schalensystem (\mathcal{S})
Input : Die aktuelle Kugel im Zufallslauf (k)

```

1 < richtung, distanz > = minDistanz (k,  $\mathcal{S}$ .ausenkugeln ())
2 s =  $\mathcal{S}$ .anzahlSchalen ()
3 schalePruefen = true
4 while s > 0 and schalePruefen do
5     if Not kugelInHüllkugel (k,  $\mathcal{S}$ .hüllkugel (s)) then
6         schalePruefen = false
7     if Not schalePruefen then
8         < tmpRichtung, tmpDistanz > = minDistanz (k,  $\mathcal{S}$ .hüllkugel (s))
9         if (Not tmpRichtung = KEINE) and (tmpDistanz < distanz) then
10            schalePruefen = true
11    if schalePruefen then
12        < tmpRichtung, tmpDistanz > = minDistanz (k,  $\mathcal{S}$ .schalenkugeln (s))
13        if (Not tmpRichtung = KEINE) and (tmpDistanz < distanz) then
14            richtung = tmpRichtung
15            distanz = tmpDistanz
16    s --
17 verschiebeRichtung = rand (6)
18 verschiebe (k, verschiebeRichtung, distanz)
19 if verschiebeRichtung = richtung then
20      $\mathcal{S}$ .kugelHinzufügen (k)
21     return true
22 else
23     return false

```

die aktuell laufende Kugel k die nächste Position im Zufallslauf zu berechnen. Zuerst werden die Außenkugeln (wenn sie existieren) überprüft (Zeile 1). Die Funktion `minDistanz` überprüft für alle 6 möglichen Richtungen, ob die Kugel k eine der als Parameter übergebenen Kugeln treffen kann. Falls ein Treffer möglich ist, werden die Richtung und die kürzeste ermittelte Distanz als Ergebnis geliefert.

Das Konzept der Außenkugeln wird normalerweise nur für Performance-Messungen verwendet. Wenn ein Schalensystem in ausreichender Größe definiert wurde, wird

jede Clusterkugel immer innerhalb einer Schale liegen. Ohne Schalensystem (Zeile 2: Anzahl Schalen = 0) werden jedoch alle Kugeln als Außenkugeln behandelt, was einen direkten Vergleich des Laufzeitverhaltens des Algorithmus mit und ohne Schalensystem erlaubt.

Die einzelnen Schalen werden nun von außen nach innen durchlaufen (Zeile 4 bis Zeile 16). Dabei wird zunächst geprüft, ob sich die laufende Kugel k bereits innerhalb der Hüllkugel einer Schale befindet (Zeile 5) oder ob diese Hüllkugel im aktuellen Schritt (also in einer der 6 möglichen Bewegungsrichtungen) von der laufenden Kugel k getroffen werden kann (Zeile 8 und 9). Wenn beides nicht der Fall ist, kann keine der Clusterkugeln innerhalb dieser Schale getroffen werden. Das gilt auch für alle tiefer liegenden Schalen, so dass die Schleife (Zeile 4) beendet werden kann. Ist aber ein Treffen der Hüllkugel möglich, wird zusätzlich die Distanz geprüft (Zeile 9), ist sie kleiner als die bisherige minimale Distanz, dann könnten auch die Kugeln in dieser Schale in einer kürzeren Distanz getroffen werden. In diesem Fall ist es notwendig, die Kugeln zu überprüfen (Zeile 11) und die Schleife fortzusetzen.

Nach dem Beenden der Schleife wird zufällig eine der 6 möglichen Richtungen gewählt (Zeile 17) und die Kugel k um die ermittelte kürzeste Distanz in diese Richtung verschoben (Zeile 18). Wenn die zufällig gewählte Richtung mit der für die kürzeste Distanz ermittelten Richtung überein stimmt (Zeile 19), dann ist die Kugel nach diesem Schritt am Cluster angedockt. Die Kugel kann dann dem Schalensystem hinzugefügt werden (Zeile 20).

Dieser Ablauf zur Berechnung eines Schrittes im Zufallslauf und die Verwendung eines Schalensystems bieten mehrere Vorteile:

- Da die Schleife sofort beendet wird, sobald keine Hüllkugel mehr getroffen werden kann oder die Distanz zu ihnen größer ist als die bisher ermittelte kürzeste Distanz, werden in den allermeisten Fällen nur noch die am äußeren Rand des Clusters angedockten Kugeln überprüft. Der Algorithmus wird dadurch wesentlich beschleunigt.
- Ein Rotation des Clusters um den Koordinatenursprung während des Wachstums, um die Entstehung einer anisotropen Form zu vermeiden, ist ohne weitere Anpassungen möglich. Da die Mittelpunkte aller Hüllkugeln des Schalensystems im Koordinatenursprung liegen, ändert sich die Zuordnung der Clusterkugeln zu den Schalen nicht durch eine Rotation.
- Über die Hüllkugeln des Schalensystems kann sehr einfach ermittelt werden, wann der Cluster die vorgegebene Abbruchgröße überschreitet. Dazu wird der Durchmesser der Hüllkugel der größten Schale auf die Abbruchgröße gesetzt.

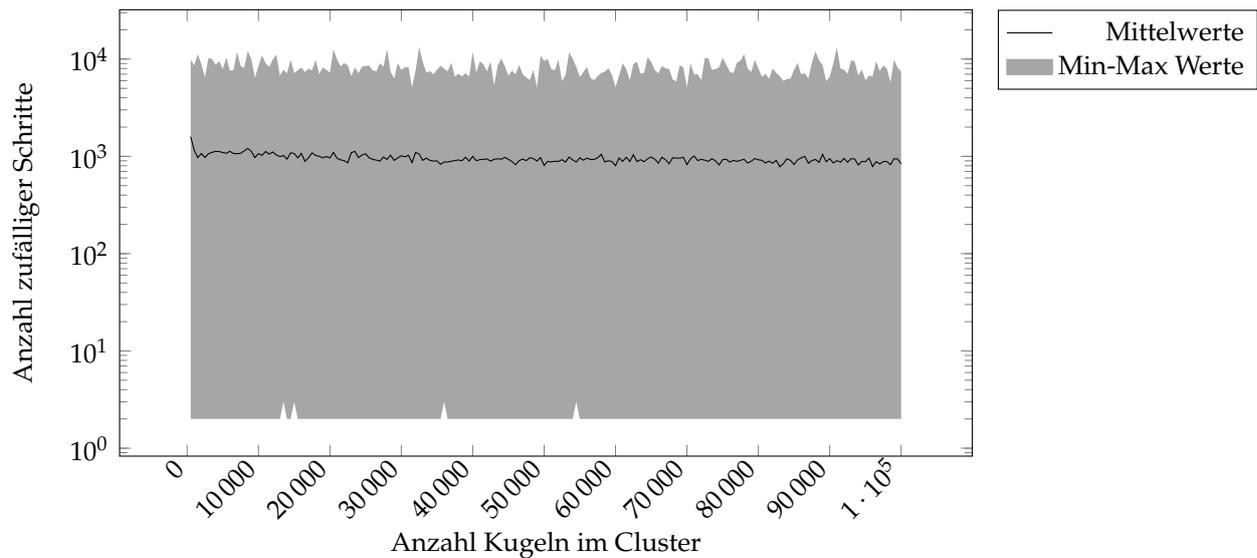


Abbildung 6.6: Anzahl Schritte im Random-Walk bei unterschiedlicher Clustergröße

Sobald eine erste angedockte Kugel außerhalb des Schalensystems liegt und somit zur Außenkugel wird, ist dies ein Hinweis auf das Überschreiten der Abbruchgröße. Da DLA-Cluster aber in der Regel eine sehr unregelmäßige Form haben, ist damit nicht garantiert, dass ihr Durchmesser tatsächlich schon oberhalb der Abbruchgröße liegt. Deshalb muss der Cluster, nach dem Auftreten der ersten Außenkugel, im Koordinatenursprung zentriert werden. Weil dabei die Clusterkugeln verschoben werden, muss anschließend ihre Zuordnung zu den Schalen des Schalensystems aktualisiert werden. Wenn nach diesem Vorgang noch immer Clusterkugeln außerhalb des Schalensystems liegen, hat der Cluster die Abbruchgröße tatsächlich überschritten und sein Wachstum wird beendet. Ist das nicht der Fall, wird das Wachstum fortgesetzt.

6.4 Laufzeitverhalten des DLA-Algorithmus

Die Laufzeit des DLA-Algorithmus hängt grundsätzlich von zwei Faktoren ab: der Anzahl der Kugeln N , die zu einem Cluster formiert werden und der Anzahl der Schritte des Zufallslaufs M , die notwendig sind, um eine Kugel am Cluster anzudocken. Wie sich M bei zunehmender Clustergröße verhält, ist in Abbildung 6.6 dargestellt. Für die Bestimmung dieser Daten wurde ein DLA-Cluster aus 100 000 Kugeln generiert, dabei wurden nach jeweils 500 angedockten Kugeln die minimale, die maximale und die mittlere Anzahl Schritte im Zufallslauf notiert. Die Abbildung zeigt, dass es einen sehr großen Unterschied zwischen der minimalen und der maximalen Anzahl Schritte gibt. So benötigt der Zufallslauf im besten Falle nicht mehr als 2 Schritte, um eine Kugel am

Cluster anzudocken, im schlechtesten Fall sind es aber über 10 000 Schritte. Die mittlere Anzahl der benötigten Schritte liegt dabei zwischen 800 und 1 200. Auffällig ist, dass alle drei Werte annähernd konstant bleiben und nicht mit der Clustergröße zunehmen. Verantwortlich dafür ist die in den vorangegangenen Kapiteln beschriebene Art der Berechnung der Zufallsschritte. Da die Weite jedes Zufallsschrittes immer an die aktuelle Entfernung der laufenden Kugel zum Cluster angepasst wird, ist die Anzahl der für den Zufallslauf benötigten Schritte stets unabhängig von den Durchmessern des Clusters und der Startsphäre. Wir können M also als konstant annehmen.

Für die Berechnung der Weite eines Zufallsschrittes muss der Algorithmus die Entfernungen zu den Clusterkugeln ermitteln, die in diesem Schritt getroffen werden können. Ohne Verwendung einer Clusterkugelverwaltung müssen für jeden Zufallsschritt alle Clusterkugeln überprüft werden. Für das Laufzeitverhalten ergibt sich dann:

$$O(M \cdot N^2).$$

Bei Verwendung eines Schalensystems wird jedoch nur noch der äußere Rand - also die aktive Zone - des Clusters überprüft. Bezeichnen wir die Anzahl der Clusterkugeln, die in den zu überprüfenden Schalen liegen mit N_S , dann ergibt sich für das Laufzeitverhalten:

$$O(M \cdot N \cdot N_S) \quad \text{mit } N_S < N.$$

Dies bedeutet eine erhebliche Beschleunigung des Algorithmus. Ein Vergleich der Laufzeiten des Algorithmus mit und ohne Schalensystem ist in Abbildung 6.7 dargestellt. Die Simulationen wurden auf einem Computer mit einer Intel Core i5 M450 CPU mit 2,4 GHz und 4 GB Arbeitsspeicher durchgeführt. Ohne Schalensystem benötigt der Algorithmus für die Generierung eines DLA-Clusters aus 100 000 Kugeln über 42 Stunden, mit Schalensystem sind es nur noch 11 Minuten. Für die Verwendung des Algorithmus als Agglomeratengenerator innerhalb der PSD-Morph Simulation ist diese Geschwindigkeit ausreichend, da die dort zu generierenden Agglomerate in der Regel sehr viel kleiner sind als 100 000 Kugeln. Es ist jedoch zu beachten, dass die Anzahl der in einer Schale des Schalensystems S liegenden Clusterkugeln mit der Oberfläche der zugehörigen Hüllkugel wächst. Damit wächst die Anzahl der in jedem Zufallsschritt zu überprüfenden Clusterkugeln N_S quadratisch zum Clusterdurchmesser. Dieser Algorithmus ist also nur bedingt für die Generierung sehr großer Cluster ($\gg 100\,000$ Kugeln) geeignet. Eine Verbesserung der Laufzeit für sehr große Cluster ist möglich, indem N_S aufgeteilt wird. Dies geschieht durch eine Zerlegung der Schalen in kleinere Segmente. Der Algorithmus muss dann nicht mehr die gesamte Schale, sondern nur

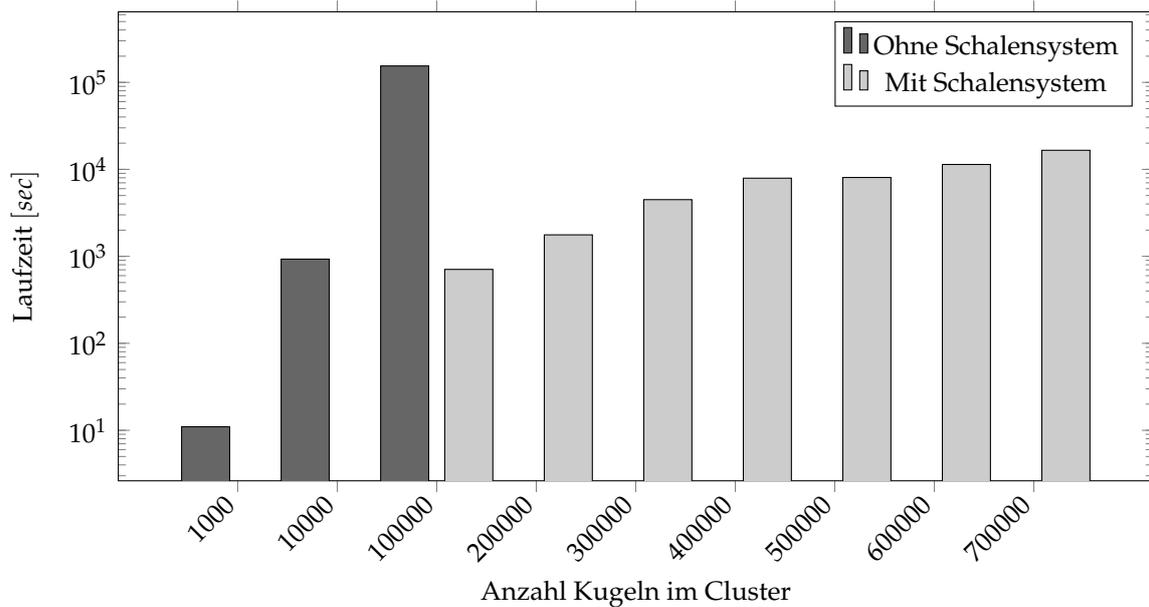


Abbildung 6.7: Vergleich der Laufzeiten des DLA-Algorithmus mit und ohne Schalensystem.

noch die Segmente überprüfen, die er in einem Zufallsschritt treffen kann. Zu bedenken ist dabei, ob ein effizientes Rotieren des Clusters auch mit segmentierten Schalen noch möglich ist.

6.5 Eigenschaften der DLA-Agglomerate

Der in den vorangegangenen Kapiteln vorgestellte DLA-Algorithmus generiert ausschließlich dreidimensionale Cluster. Die innere Struktur dieser Cluster kann sichtbar gemacht werden, indem eine dünne Scheibe aus ihnen heraus geschnitten wird. Mehrere dieser Scheiben, für verschiedene Wachstumsphasen eines DLA-Clusters, sind in Abbildung 6.8 dargestellt. Die einzelnen Kugeln des Clusters sind unterschiedlich eingefärbt. Die erste Kugel (Saatpartikel) ist blau, die letzte dem Cluster hinzugefügte Kugel rot gefärbt. Alle Kugeln dazwischen haben eine gemischte Farbe, deren Blau- und Rotanteil der Reihenfolge ihres Hinzufügens zum Cluster entspricht. Die Abbildung zeigt die fraktalartige Struktur des Clusters, mit den typischen DLA-Ästen. Diese Äste werden mit zunehmender Größe des Clusters immer ausgeprägter. Die Färbung der Kugeln macht die aktive Zone (rot bis violett gefärbt) sichtbar, in der das Wachstum des Clusters stattfindet. Da dieser Cluster während seines Wachstums nicht rotiert wurde, entstand eine anisotrope Form. Man erkennt, dass die heraus geschnittene Scheibe mit zunehmender Größe immer deutlicher die Form einer Raute annimmt, die auf die Koordinatenachsen ausgerichtet ist. Eine weitere Besonderheit der inneren Struktur

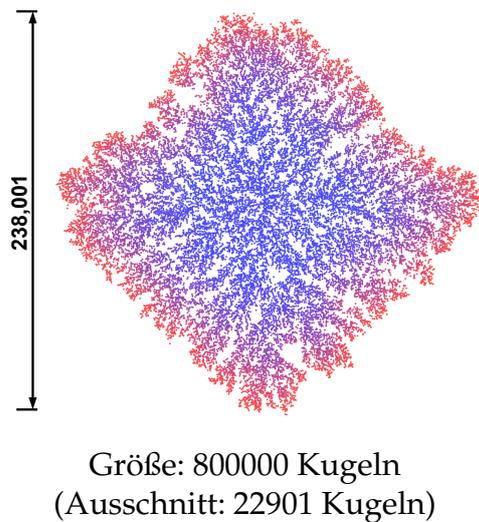
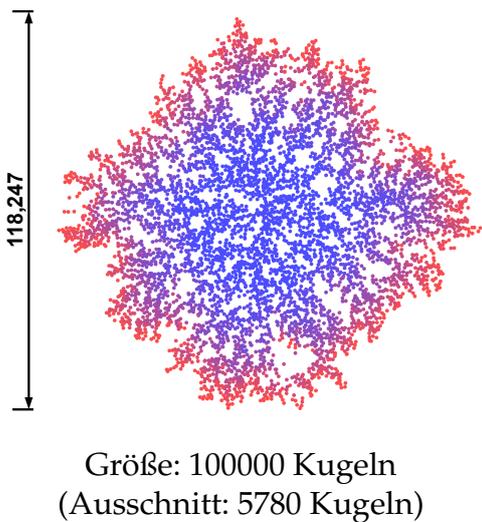
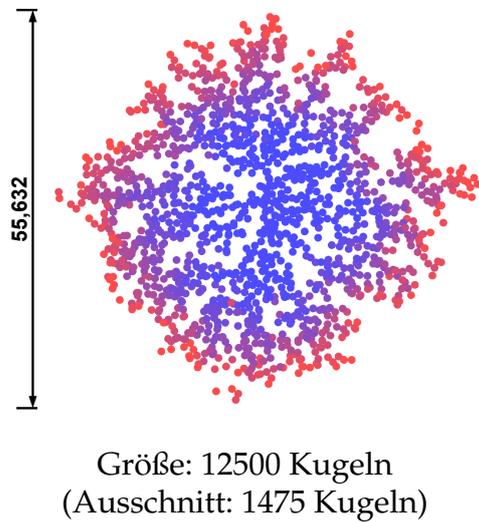
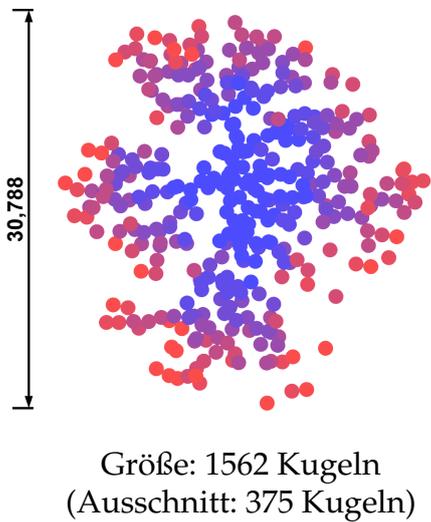
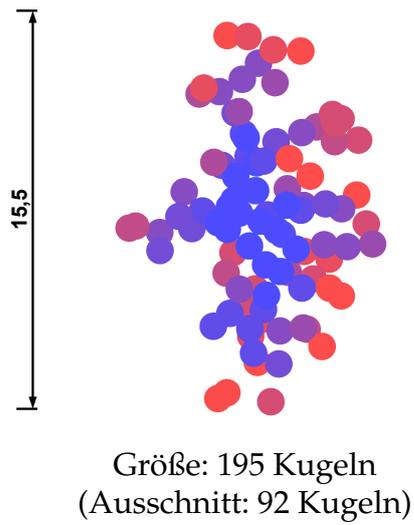
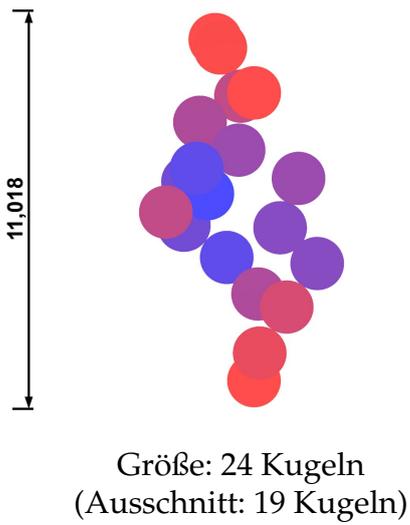


Abbildung 6.8: Schnitt einer dünnen Scheibe aus einem DLA-Agglomerat in verschiedenen Wachstumsphasen

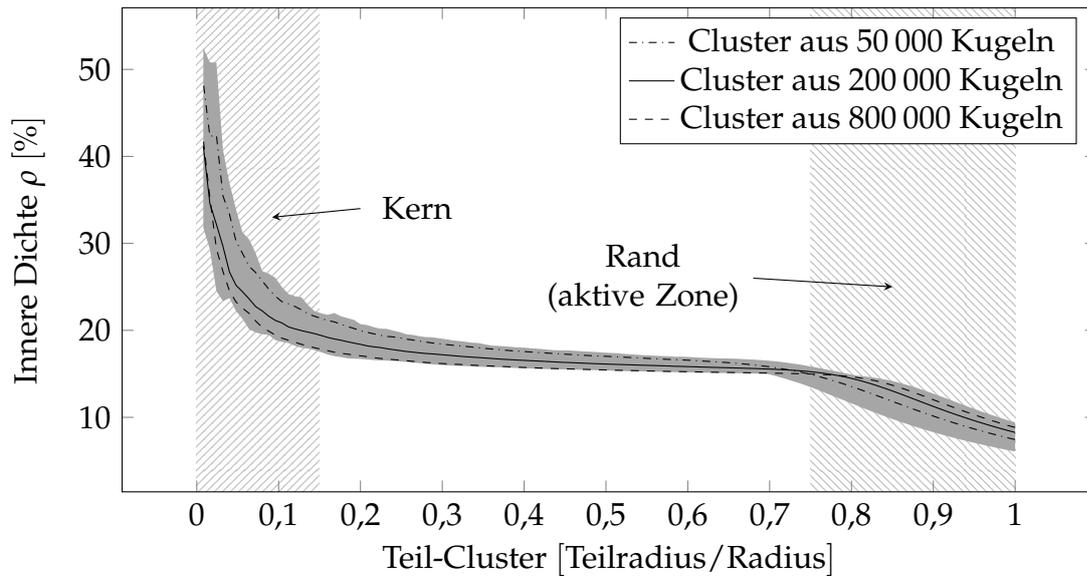


Abbildung 6.9: Verteilung der inneren Dichte in DLA-Clustern

erkennt man, wenn man die zwischen den Ästen liegenden Hohlräume des Clusters betrachtet. Sie werden mit zunehmendem Abstand vom Clusterkern immer größer. Das ist der Hauptgrund dafür, dass die Packungsdichte einer Partikelmischung, mit zunehmender Größe der in ihr enthaltenen Agglomerate immer weiter absinkt. Das zeigt auch, dass die DLA-Cluster ein gutes Modell für reale Agglomerate bilden.

Einen genaueren Eindruck von der Verteilung der inneren Dichte in einem DLA-Cluster vermittelt die Abbildung 6.9. Für die dargestellten Messergebnisse wurden jeweils 10 Cluster aus 50 000, 200 000 und 800 000 Kugeln generiert. Die Messungen wurden für Ausschnitte der Cluster mit unterschiedlich großen Teilradien vorgenommen. Dazu wurde zunächst eine Hüllkugel um den Cluster gelegt, die alle Clusterkugeln enthält und somit den größten Ausschnitt des Clusters darstellt. Der Anteil des Volumens aller Clusterkugeln am Volumen der Hüllkugel ist der Wert der inneren Dichte ρ . Anschließend wurden kleinere Ausschnitte des Clusters betrachtet, indem der Radius der Hüllkugel schrittweise verkleinert wurde. Für diese Ausschnitte ergibt sich der Wert ρ dann aus dem Volumen aller innerhalb der verkleinerten Hüllkugel liegenden Clusterkugeln sowie dem Volumen dieser Hüllkugel. Da die Cluster mit unterschiedlicher Anzahl Clusterkugeln auch unterschiedliche Größen aufweisen, wurden die Verläufe der inneren Dichte nicht abhängig vom absoluten sondern vom relativen Teilradius (Teilradius/Radius) dargestellt. Mit Radius ist dabei der Radius der Hüllkugel gemeint, die alle Clusterkugeln enthält, und mit Teilradius der Radius der verkleinerten Hüllkugel. Auf diese Weise ist ein direkter Vergleich der Ergebnisse möglich. Die Verläufe der inneren Dichte aller generierten Cluster liegen innerhalb des grau dargestellten Bereichs. Der Verlauf der Mittelwerte für die drei Clustergrößen

sind als Linien dargestellt.

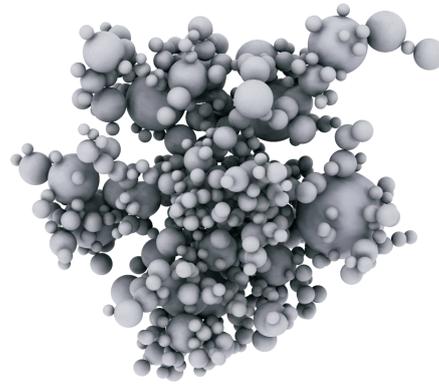
Anhand der Werte können deutlich 3 Zonen unterschieden werden. Im Inneren jedes Clusters liegt ein Kern, die Dichte steigt in Richtung dieses Kerns exponentiell an. In der Zone außerhalb des Kerns fällt die Dichte nur langsam zur aktiven Zone hin ab. In der aktiven Zone fällt die Dichte dann deutlich schneller nach außen ab. Stellt der Cluster ein Agglomerat innerhalb einer Partikelpackung dar, dann können Nachbarpartikel in die aktive Zone des Clusters eindringen, aufgrund der geringen Dichte dort. Wie in Kapitel 5.4.5 beschrieben, können wir dies berücksichtigen, indem die Kugel, die das Agglomerat in der Partikelpackung ersetzt, einen etwas kleineren Durchmesser aufweist als die Hüllkugel des Agglomerates. In Abbildung 6.9 erkennt man, dass die aktive Zone ungefähr zwischen 75% und 80% der Clustergröße beginnt. Der Skalierungsfaktor τ für die Bestimmung der Größe der Agglomerate sollte also auf einen Wert zwischen 0,75 und 0,8 eingestellt werden.

Die bis hier gezeigten Beispiele für DLA-Cluster bestanden alle aus Kugeln der gleichen Größe. Als Agglomeratengenerator für die PSD-Morph Simulation muss unser DLA-Algorithmus aber Cluster generieren, deren Kugeln einer vorgegebenen Korngrößenverteilung folgen. Beispiele für solche Cluster mit unterschiedlichem Durchmesser sind in Abbildung 6.10 dargestellt. Die Größen der Clusterkugeln folgen einer Fullerverteilung mit Exponent 0,01 und reichen dabei von $0,048\mu\text{m}$ bis $0,868\mu\text{m}$. Alle dargestellten Cluster wurden während ihres Wachstums rotiert und haben deshalb eine isotrope kugelähnliche Form, was besonders am größten Cluster (Teil c) der Abbildung deutlich wird. Im Teil d der Abbildung ist zum Vergleich die Mikroskopaufnahme eines realen Mikrosilica Agglomerates dargestellt.

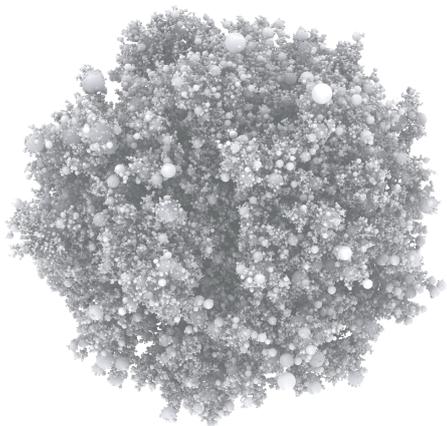
Alle diese Cluster haben grundsätzlich die selbe fraktale Struktur, ihre Wachstumsgeschwindigkeit und ihre innere Dichte hängen jedoch von der Korngrößenverteilung der Kugeln ab, aus denen sie bestehen. In Abbildung 6.11 sind die Größen und die Anzahlen der enthaltenen Primärpartikel (bzw. Clusterkugeln) für verschiedene DLA-Agglomerate dargestellt. Dafür wurden 4 unterschiedliche Agglomeratetypen simuliert. Der erste Typ enthält Primärpartikel, deren Größen einer Fullerverteilung mit den bereits für Abbildung 6.10 beschriebenen Parametern folgen. Der zweite Typ unterscheidet sich nur durch den Exponenten der Fullerverteilung, er beträgt hier 0,9, wodurch die Agglomerate mehr große Primärpartikel enthalten. Die Typen drei und vier entsprechen jeweils dem ersten bzw. zweiten Typ. Sie unterscheiden sich aber dadurch, dass hier eine geringere Anhaftwahrscheinlichkeit (0,05) für die Primärpartikel festgelegt wurde (RLA-Version), was zur Entstehung dichter Cluster (bzw. Agglomerate) führt. Für die Simulationen wurden 6 verschiedenen Abbruchgrößen festgelegt, bei deren Überschreiten das Wachstum gestoppt wird. Für jede Abbruchgröße und



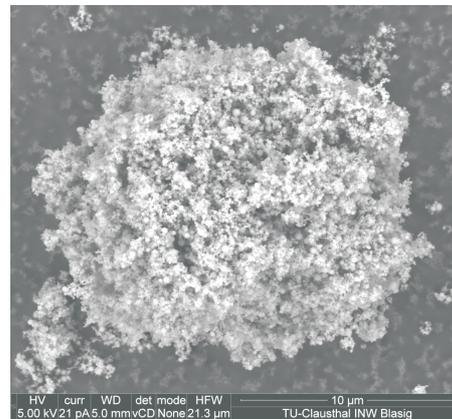
(a)
26 Kugeln
Durchmesser: $0,6\mu\text{m}$



(b)
758 Kugeln
Durchmesser: $2,27\mu\text{m}$



(c)
311 144 Kugeln
Durchmesser: $18,16\mu\text{m}$



(d)
Mikrosilica-Agglomerat
Durchmesser: $17\mu\text{m}$

Abbildung 6.10: DLA-Agglomerate verschiedener Größe (a)-(c) im Vergleich mit einem Mikrosilica-Agglomerat (d)

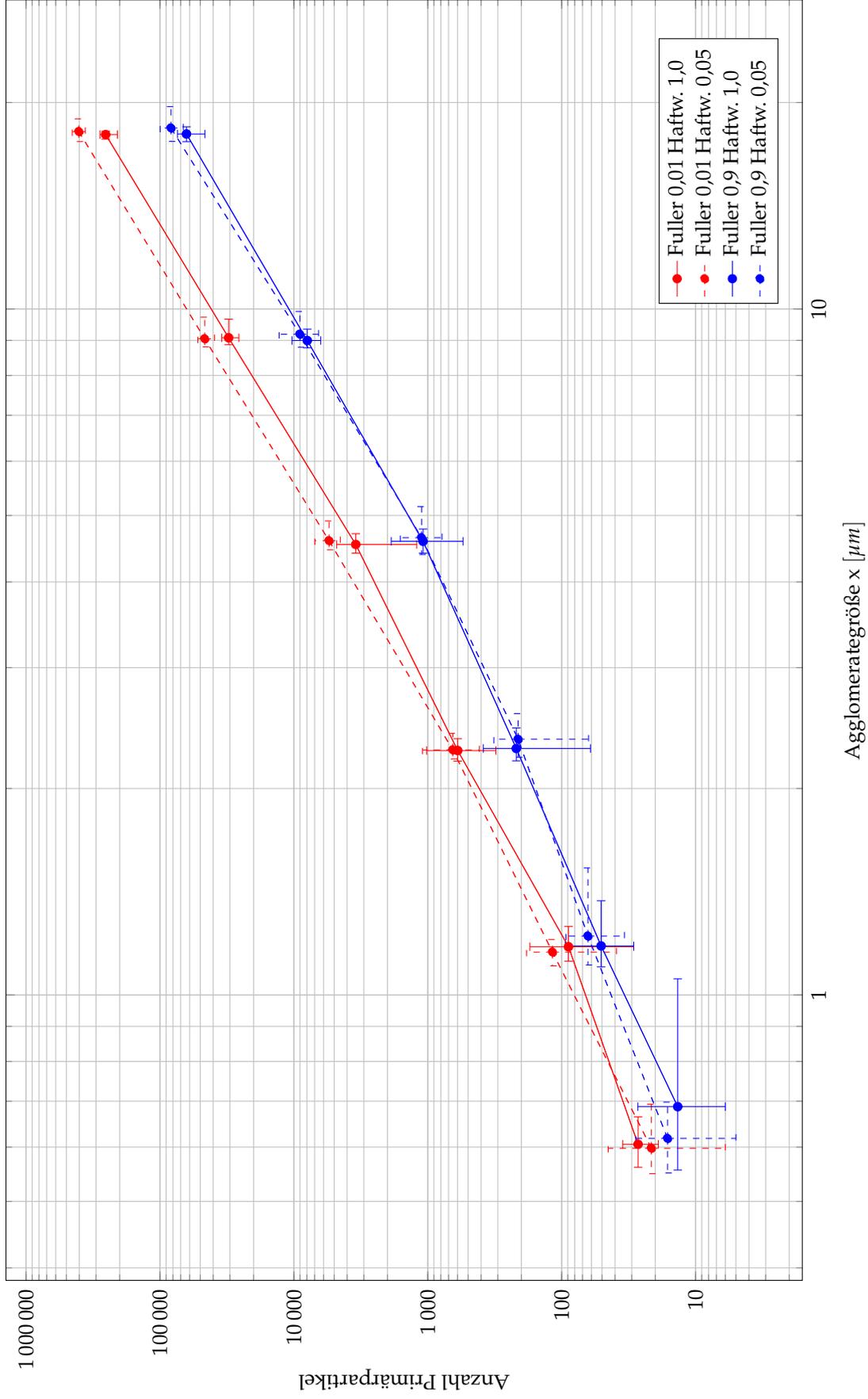


Abbildung 6.11: Einfluß der Korngrößenverteilung und der Anhaftwahrscheinlichkeit auf das Wachstum von DLA-Agglomeraten. Die minimal geforderten Agglomeratgrößen (Abbruchgrößen) in der Simulation betragen: 0,5474 μm ; 1,0948 μm ; 2,1897 μm ; 4,3794 μm ; 8,7588 μm und 17,5177 μm .

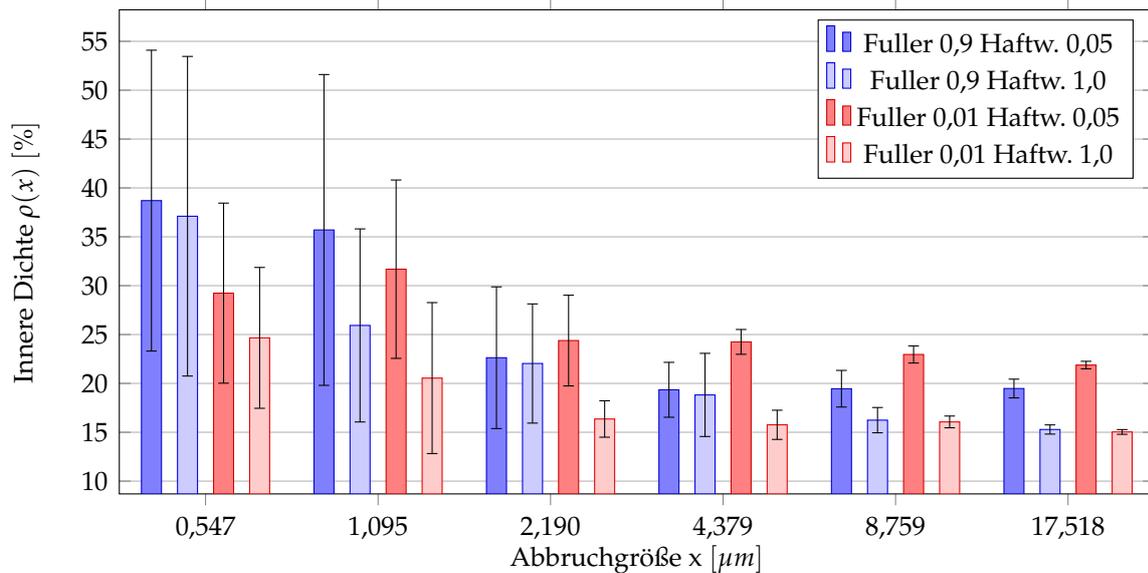


Abbildung 6.12: Innere Dichte von DLA-Agglomeraten bei unterschiedlicher Größe und Primärpartikelverteilung

jeden Agglomeratetyp wurden jeweils 10 Simulationen durchgeführt. Die Mittelwerte für die generierten Agglomeratgrößen und die zugehörige Anzahl der enthaltenen Primärpartikel sowie die jeweilige Streuung der Ergebnisse sind in der Abbildung dargestellt. Wie man erwarten konnte, wachsen Agglomerate, die mehr große Primärpartikel enthalten, wesentlich schneller als Agglomerate die überwiegend aus kleinen Primärpartikeln bestehen. Mit einer niedrigeren Anhaftwahrscheinlichkeit wiederum entstehen dichtere Agglomerate, die mehr Primärpartikel benötigen, um die Abbruchgröße zu überschreiten. Solche Agglomerate wachsen also auch langsamer, allerdings ist hier der Unterschied nicht so deutlich wie zwischen unterschiedlichen Korngrößenverteilungen der Primärpartikel.

Die mittlere innere Dichte, die für jede der Agglomeratgrößen ermittelt wurde, sowie die jeweilige Streuung der Ergebnisse ist in Abbildung 6.12 dargestellt. Auffällig ist die sehr starke Streuung bei den kleinsten Agglomeraten. Diese bestehen fast nur aus einem Kern, die fraktalen Äste sind bei ihnen noch nicht sehr ausgeprägt, was auch ihre relativ hohe Dichte erklärt. Mit zunehmender Größe der Agglomerate sinkt ihre Dichte und die Streuung in den Ergebnissen lässt deutlich nach. Bei den kleinsten Größen erreichen die Agglomerate, deren Primärpartikelgrößen einer Fullerverteilung mit Exponent 0,9 folgen, die höchsten Dichten. Bei ihnen ist der Anteil der großen Primärpartikel höher, deshalb besteht auch ihr Kern mit höherer Wahrscheinlichkeit aus mindestens einem großen Primärpartikel. In diesem Fall sind nur sehr wenige Primärpartikel notwendig, um die Abbruchgröße zu überschreiten und es können sich noch keine größeren Äste bilden. Diese Agglomerate haben deshalb häufiger eine kompakte

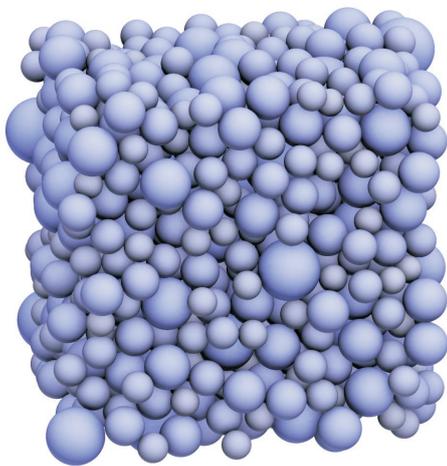
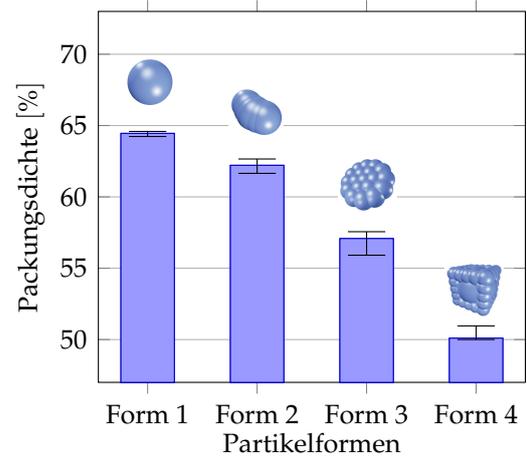
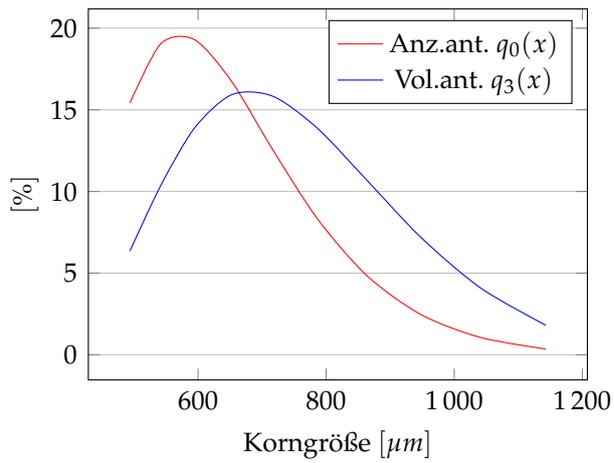
tere Form mit einer hohen Dichte. Anders verhält es sich bei den größeren Agglomerategrößen. Ein höherer Anteil an kleinen Primärpartikeln führt hier zu höheren Dichten, auch wenn die Unterschiede weniger ausgeprägt sind als bei den kleinen Agglomeraten. Kleine Primärpartikel können mit höherer Wahrscheinlichkeit tiefer in die Hohlräume zwischen den Ästen des Agglomerates eintauchen als große Primärpartikel, die kaum zwischen den Ästen hindurch passen. So können dichtere Agglomerate entstehen.

Kapitel 7

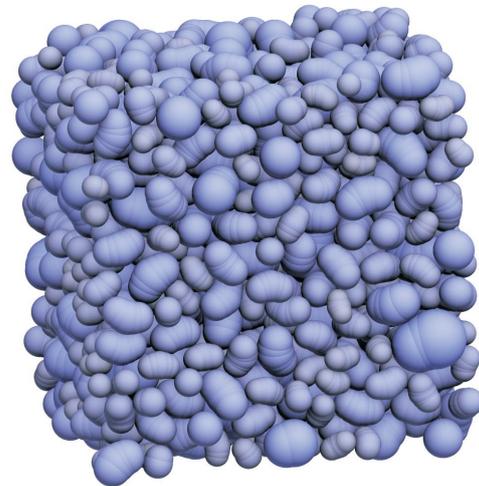
Zukünftige Entwicklungen

Mit dem Abschluss dieser Arbeit steht - in Form der beiden Programme „RaSim“ und „PSD-Morph“ - ein System zur Verfügung, das alle wichtigen Anforderungen für die simulative Bestimmung der Packungsdichte von Betonmischungen erfüllt. Die Packungssimulation „RaSim“ kann die für diese Mischungen typischen hoch polydispersen Korngrößenverteilungen dank einer hierarchischen Zerlegung (Fraktionierung) verarbeiten. Durch die Erweiterung zur Online-Fraktionierung erfolgt die Zerlegung der Korngrößenverteilung nicht mehr starr, wie in der ursprünglichen Version der Simulation, sie passt sich dynamisch an die Verteilung an. Dabei wird sichergestellt, dass alle zu einer Fraktion gehörenden Korngrößen durch eine ausreichende Partikelanzahl repräsentiert sind. Fehlerhafte Ergebnisse, die durch die Dominanz großer Partikel innerhalb einer Packung entstehen können, werden so vermieden. Die Auswirkungen der Agglomeration auf die Packungsstruktur können mit „PSD-Morph“ simuliert werden. Da das Wachstum der Agglomerate dabei mithilfe des stochastischen Wachstumsmodells der „Diffusion Limited Aggregation“ dargestellt wird, sind für die Simulation nur wenige, relativ einfach zu bestimmende, Ausgangsdaten notwendig. Der kombinierte Einsatz von „PSD-Morph“ und „RaSim“ erlaubt nun auch eine realistische Simulation von Partikelpackungen mit hohem Feinstkornanteil. Eine rechnergestützte Optimierung von Betonmischungen ist somit möglich.

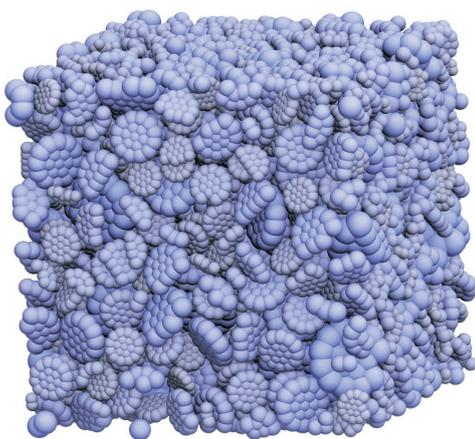
Alle bisher in dieser Arbeit vorgestellten Simulationsergebnisse stammen aus Simulationen mit kugelförmigen Partikeln. Der von Raschdorf [Ras10] entwickelte Packungsalgorithmus ist jedoch auch in der Lage andere Partikelformen zu verarbeiten, wenn diese Formen als Kugelcluster dargestellt werden. Einige Beispiele für Packungen aus nicht kugelförmigen Partikeln sind in Abbildung 7.1 dargestellt. Die Korngrößen in jeder dieser Packungen folgen der Verteilung im oberen linken Teil der Abbildung. Alle vier abgebildeten Packungen bestehen aus je 1000 Partikeln. Als Referenz dient die erste Packung, die nur aus einfachen kugelförmigen Partikeln besteht. Die



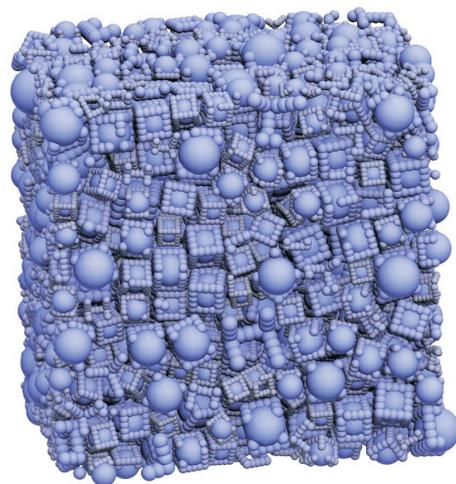
Packung Form 1



Packung Form 2



Packung Form 3



Packung Form 4

Abbildung 7.1: Simulationen von Packungen aus nicht kugelförmigen Partikeln

jeweils ermittelten Simulationsergebnisse sind im oberen rechten Teil der Abbildung dargestellt. Sie zeigen, dass die Packungsdichten bei unterschiedlichen Partikelformen ganz erheblich voneinander abweichen können. Solange die Partikel aber keine flache oder eckige Form haben (Form 3 und 4) sondern eine kompakte und eher kugelhähnliche Form (Form 2) liegt die ermittelte Packungsdichte nahe an der einer Kugelpackung (Form 1). Das wir uns bei allen anderen im Rahmen dieser Arbeit durchgeführten Simulationen auf einfache kugelförmige Partikel beschränken, ist dadurch zu rechtfertigen, dass viele der in Betonmischungen vorkommende Stoffe aus Partikeln bestehen, die nur unerheblich von der Kugelform abweichen. Die Auswirkung der Partikelform auf die Packungsdichte kann in diesem Fall vernachlässigt werden. Trotzdem ist die Simulation von nicht kugelförmigen Partikeln als Kugelcluster eine interessante Option. Sie kann immer dann eingesetzt werden, wenn Packungen aus Partikeln, die stark von der Kugelform abweichen, simuliert werden sollen. Auch für die Simulation von agglomerierten Mischungen ergeben sich damit interessante Möglichkeiten. Da die „PSD-Morph“ Simulation virtuelle Agglomerate als Kugelcluster erzeugt, könnten diese Agglomerate auch direkt als Partikelform von „RaSim“ verarbeitet werden. Damit wäre es möglich, neben der Packungsdichte, auch den Einfluss der Agglomerate auf die Koordinationszahlen der Partikel in der Packung zu untersuchen. Beim Einsatz dieser Technik ist jedoch zu beachten, dass die Gesamtanzahl der von der Simulation zu verarbeitenden Kugeln mit der Anzahl der in einem Partikel enthaltenen Kugeln steigt, sie beträgt:

$$\text{Gesamtanzahl Kugeln} = \text{Anzahl Partikel} \cdot \text{Anzahl Kugeln pro Partikel.}$$

So bestehen beispielsweise die würfelförmigen Partikel (Form 4) in Abbildung 7.1 aus 45 Kugeln. Die Packung aus diesen Partikeln enthält also 45000 Kugeln. Im Gegensatz zu einer Kugelpackung muss die Simulation hier, neben der Position jedes Partikels, auch dessen räumliche Ausrichtung berechnen. Der Rechenaufwand für die Simulation einer Packung aus nicht kugelförmigen Partikeln ist also um ein vielfaches höher als bei Kugelpackungen. Solche Simulationen können deshalb bisher nur in engen Grenzen, das heißt mit einer geringen Anzahl Partikel, durchgeführt werden.

Die zukünftigen Entwicklungen sollten sich vor allem auf die weitere Beschleunigung der Packungssimulation „RaSim“ konzentrieren. Dies ist vor allem für den Einsatz des Simulationssystems innerhalb einer Optimierungsaufgabe notwendig. Unabhängig davon, welche Technik man für diese Optimierung verwendet, sind dafür auf jeden Fall eine große Anzahl von Packungssimulationen erforderlich. Da „RaSim“ für die Verarbeitung einer umfangreichen Korngrößenverteilung, wie sie für Betonmischungen typisch ist, oft mehrere Stunden benötigt, würde sich die Optimierung

einer entsprechenden Mischung über viele Tage erstrecken. Durch die Beschleunigung der Simulation ließe sich dieser Vorgang auf eine akzeptablere Zeitspanne verkürzen. Gleichzeitig erweitern sich dadurch die Einsatzmöglichkeiten der Software. Da die Simulation dann auch höhere Kugelanzahlen in moderater Zeit verarbeiten könnte, wäre auch die bisher kaum genutzte Möglichkeit der Simulation nicht kugelförmiger Partikel in größerem Umfang anwendbar.

7.1 Parallele Simulation

Bisher wurde die Packungssoftware durch die reine Verbesserung ihres Laufzeitverhaltens beschleunigt. Durch die von Raschdorf entwickelte Nachbarschaftsverwaltung mit Verlet-Listen und Loose-Octree, die in Kapitel 3.3 beschrieben wurde, ist diese Möglichkeit weitestgehend ausgeschöpft. Weitere Möglichkeiten ergeben sich durch die Parallelisierung der Simulation. Diese beruht auf dem Prinzip des „Divide and Conquer“, wobei die zu lösende Aufgabe in mehrere kleinere Teilaufgaben zerlegt wird, deren jeweilige Lösungen unabhängig voneinander berechnet und anschließend zu einem Gesamtergebnis zusammengefasst werden. Die Lösung der Teilaufgaben kann dabei entweder auf unterschiedlichen Rechnern innerhalb eines Rechenclusters oder innerhalb eines einzigen Rechners mit mehreren Prozessorkernen erfolgen. Besonders die letzte Variante ist hierbei interessant, da CPUs mit mehreren Rechenkernen heute bereits zum Standard gehören. Der Geschwindigkeitsgewinn durch die Parallelisierung beruht darauf, dass die einzelnen Rechenschritte der Simulation nicht mehr sequentiell - also zeitlich nacheinander von nur einem Rechenkern - sondern parallel und damit gleichzeitig von mehreren Rechenkernen bearbeitet werden. Wie hoch dieser Geschwindigkeitsgewinn ausfällt, hängt nicht alleine von der Anzahl der eingesetzten Rechenkerne ab, sondern auch davon, wie die zu lösende Aufgabe in Teilaufgaben zerlegt wird. Für die Packungssimulation lassen sich drei verschiedene Strategien unterscheiden, die im Folgende beschrieben werden.

7.1.1 Parallele Simulation der Fraktionen

Eine naheliegende Strategie, die zudem nur geringe Änderungen an der Implementierung der Packungssimulation erfordert, ist die parallele Simulation der einzelnen Fraktionen. Die Zerlegung der gegebenen Korngrößenverteilung in Fraktionen muss nur einmal zu Beginn im Hauptthread der Simulation durchgeführt werden. Jede Fraktion wird dann einem eigenen Thread übergeben, der die Generierung der Packung durchführt. Die Berechnung der Gesamtpackungsdichte aus den Teilergebnissen er-

folgt dann wieder im Hauptthread. Die Ausführung jedes dieser Threads kann unabhängig voneinander auf unterschiedlichen Rechenkernen erfolgen. Dabei gilt grundsätzlich, je mehr Rechenkerne zur Verfügung stehen, umso mehr Threads können gleichzeitig ausgeführt werden. Wie viele Threads vorhanden sind hängt davon ab, in wie viele Fraktionen die gegebene Korngrößenverteilung gemäß der vorgegebenen Parameter (maximale Anzahl Kugeln pro Fraktion und Anteil aktiver Kugeln) zerlegt wurde. Natürlich lässt sich die Anzahl der Fraktionen erhöhen, indem die Anzahl Kugeln pro Fraktion gesenkt wird. Dieses Vorgehen ist jedoch nicht empfehlenswert, da die Interaktionen zwischen den Fraktionen umso schlechter simuliert werden können, je weniger Kugeln (und damit auch: je weniger Randkugeln) in einer Fraktion enthalten sind. Außerdem ist zu beachten, dass in der bisherigen sequentiellen Simulation die Daten zum benötigten Volumen (Containervolumen) aller vorherigen Fraktionen $C(F_1, \dots, F_{i-1})$ benutzt werden konnten, um zu entscheiden, ob die Simulation der aktuellen Fraktion F_i notwendig ist oder nicht. Wie in Kapitel 3.1.2 beschrieben wurde, kann auf die Simulation der Fraktion F_i verzichtet werden, wenn bereits vorher klar ist, dass das Hohlraumvolumen ihrer Packung nicht ausreichen wird, um die Partikel der unteren Fraktionen aufzunehmen. Bei einer parallelen Simulation der Fraktionen stehen diese Daten aber nicht mehr vorab zur Verfügung. Es müssen also in jedem Fall alle Fraktionen simuliert werden. Der Geschwindigkeitsgewinn durch die parallele Simulation sollte diesen Nachteil jedoch deutlich überwiegen.

7.1.2 Parallele Berechnung der Entlappung

Bei der parallelen Simulation der Fraktionen erfolgt die Generierung der Packung pro Fraktion genau wie bei der bisherigen sequentiellen Simulation. Simulationen von weniger umfangreichen Korngrößenverteilungen, die in nur einer einzigen Fraktion erfolgen können, profitieren nicht davon. Da die Verarbeitung jeder einzelnen Fraktion auch genau so lange dauert wie bei der sequentiellen Simulation, ist es nicht möglich, die Anzahl der Kugeln pro Fraktion zu erhöhen. Eine bessere Strategie ist, die Generierung der Packung per CR-Algorithmus selbst zu parallelisieren. Der Aufwand für die Implementierung ist hier etwas höher, da die bisher benutzte Variante des Algorithmus auf eine schnelle sequentielle Verarbeitung der Kugelpackung ausgerichtet ist. In jedem Iterationsschritt wird jede Kugel der Packung genau einmal verschoben. Dazu werden alle Kugeln sequentiell durchlaufen, wobei für jede Kugel ein Vektor berechnet wird, der ihre Überlappungen mit den jeweiligen Nachbarkugeln minimiert. Jede Kugel wird sofort um diesen Vektor verschoben, noch bevor mit der nächsten Kugel fortgefahren wird. Die sofortige Verschiebung sorgt für ein schnelles Voranschreiten des Entlappungsprozesses, sie birgt aber auch Probleme bei der parallelen Berechnung.

Um den CR-Algorithmus parallel auszuführen, wird die Kugelpackung in mehrere Teilmengen zerlegt, die jeweils innerhalb eines eigenen Threads entlappt werden. Stellen wir uns vor, der Thread 1 hat gerade eine neue Position für die Kugel A berechnet, die ihre aktuellen Überlappungen minimiert. Diese neue Position wurde aber noch nicht gespeichert. Gleichzeitig berechnet der Thread 2 eine neue Position für die Kugel B, die eine Überlappung mit Kugel A an deren alten Position hatte. An der neuen Position der Kugel A besteht diese Überlappung aber nicht mehr. Da der Thread 2 die alte und nicht die neue Position der Kugel A im Speicher findet, verschiebt er die Kugel B unnötigerweise. Solche „Schreib-Lese-Konflikte“ würden häufig auftreten und die unnötigen Verschiebungen würden dafür sorgen, dass Lücken in der Packung entstehen, die in den folgenden Iterationsschritten wieder beseitigt werden müssen. Der Geschwindigkeitsgewinn durch die parallele Berechnung würde dadurch teilweise wieder zunichte gemacht.

Durch eine einfache Anpassung des Algorithmus kann dieses Problem vermieden werden. Dazu wird jeder Iterationsschritt in zwei Teilschritte zerlegt. Im ersten Teilschritt werden alle paarweisen Überlappungen zwischen den Kugeln gesucht. Für jede gefundene Überlappung werden zwei Vektoren berechnet, die den beiden überlappenden Kugeln zugewiesen werden. Die zwei Vektoren beschreiben entgegengesetzte Verschiebungen, durch die die Überlappung der beiden Kugeln beseitigt wird. Die Verschiebungen werden jedoch nicht sofort ausgeführt. Wenn alle Überlappungen gefunden und die entsprechenden Vektoren berechnet wurden, werden im zweiten Teilschritt die Verschiebungen der Kugeln durchgeführt. Da jede Kugel Überlappungen mit mehreren Nachbarkugeln haben kann, können für sie im ersten Teilschritt auch mehrere Vektoren berechnet werden, je ein Vektor pro Überlappung. Diese Vektoren können als Abstoßungskräfte interpretiert werden, die die Kugel in unterschiedliche Richtungen drücken. Durch ein Aufsummieren aller zu einer Kugel gehörenden Vektoren entsteht ein resultierender Vektor, der die Überlappungen der Kugel im aktuellen Iterationsschritt minimiert. Das entspricht wieder genau dem Prinzip, mit dem auch im bisherigen Algorithmus die Entlappung berechnet wird. Da die Berechnung und die Durchführung der Verschiebung nun aber in 2 getrennten Teilschritten erfolgt, können bei der parallelen Berechnung keine „Schreib-Lese-Konflikte“ mehr auftreten.

7.1.3 Parallele Simulation auf der GPU

Neben der Ausführung der Simulation auf einem Computer mit mehreren CPUs (bzw. einer CPU mit mehreren Rechenkernen) oder einem Cluster aus mehreren Computern bietet die Architektur moderner Computer noch eine weitere Möglichkeit der parallelen Berechnung an. Der Grafikprozessor (engl. graphics processing unit, kurz GPU)

hatte ursprünglich die Aufgabe, die CPU zu entlasten indem er die Steuerung der Ausgabe von Text und Grafik auf dem Bildschirm übernahm. Die steigenden Anforderungen an die Hardware durch die Bild- und Videoverarbeitung (unter anderem ausgelöst durch die Entwicklung aufwendiger 3D Computerspiele) haben dazu geführt, dass die GPU um immer mehr Funktionen erweitert wurde. Die Anzahl der Rechenkerne in heutigen GPUs ist in der Regel um ein Vielfaches höher als in CPUs. Sie sind darauf ausgelegt, einfache Berechnungen parallel auf einer großen Anzahl von Daten auszuführen, was eine typische Anforderung in der Bildverarbeitung ist. Diese Leistungsfähigkeit ist jedoch auch für eine Vielzahl anderer Anwendungsgebiete interessant, unter anderem für wissenschaftliche Simulationen. Die Entwickler von Grafikprozessoren stellen mittlerweile neue Technologien zur Verfügung, die eine Ausführung von Programmcode auf der GPU ermöglichen, ohne dafür hardwarenah programmieren zu müssen. Grundsätzlich müssen dazu die Programmteile, die parallel berechnet werden sollen, speziell angepasst werden. Diese werden nach dem Programmstart in den Grafikspeicher übertragen und anschließend von der GPU bearbeitet. Die fertigen Endergebnisse werden dann zurück in den Arbeitsspeicher übertragen, auf den die CPU zugreifen kann. Die Datenübertragung zwischen CPU und GPU ist zeitaufwändig und stellt einen „Flaschenhals“ bei dieser Art der parallelen Berechnung dar. Um die Vorteile einer Berechnung auf der GPU voll ausschöpfen zu können, sollten möglichst nie Teilergebnisse übertragen werden. Konkret bedeutet das: wenn der Entlappungsprozess des CR-Algorithmus parallel auf der GPU berechnet werden soll, dann sollten alle dafür notwendigen Daten und Funktionen - also auch die Nachbarschaftsverwaltung - auf der GPU laufen. Auf diese Weise sind während der Simulation keine weiteren Datenübertragungen zwischen CPU und GPU notwendig. Die dafür notwendigen Anpassungen des Quellcodes sind weitaus umfangreicher als für eine parallele Ausführung der Simulation auf der CPU. Die zu erwartende Steigerung der Geschwindigkeit und die damit verbundene Erweiterung der Möglichkeiten (Einsatz von „RaSim“ innerhalb einer Optimierungsaufgabe, Simulation nicht kugelförmiger Partikel) machen diese Strategie aber sinnvoll.

Anhang A

Eingesetzte Software

A.1 RaSim

Die Packungssimulation „RaSim“ liegt in zwei Versionen vor: einer Konsolenversion und einer Version mit grafischer Oberfläche, die sowohl die Bedienung als auch die Verwaltung der Daten für den Endanwender erleichtert. Beide Versionen des Programms werden grundsätzlich über eine Konfigurationsdatei (Dateiendung .cfg) gesteuert, die alle Parameter und die Pfade zu weiteren benötigten Dateien enthält. Bei der Konsolenversion muss die Konfigurationsdatei dem Programm als Startparameter übergeben werden. Die Einträge in der Datei erfolgen nach folgendem Muster:

```
Parametername = Wert
```

Zeilen die mit einem # beginnen werden als Kommentare gewertet. Als Wert kommen Zeichenketten, Ganz- und Fließkommazahlen sowie boolesche Werte (0 oder 1) in Frage. Im Folgenden sind alle zur Verfügung stehenden Parameter aufgelistet.

Parameter für die Simulation

zufallStartwert (Ganzzahl ≥ 0)

Legt einen Startwert für den Zufallsgenerator fest. Der Standardwert ist 0, bei dieser Einstellung wird der Startwert aus dem Systemdatum generiert. Legt man einen anderen Wert fest, erlaubt dies die genaue Reproduktion eines Simulationslaufs; eine Option, die vor allem zur Fehlersuche während der Entwicklung eingesetzt werden kann.

Fraktionsgroesse (Ganzzahl ≥ 4)

Legt die Anzahl von Partikeln fest, aus denen die Packung einer Fraktion bestehen soll. Die Anzahl der Kugeln in einer Packung kann höher sein als dieser Wert, wenn nicht kugelförmige Partikel als Cluster aus Kugeln dargestellt werden.

simKugelAnteil (Fließkommazahl aus dem Intervall $(0, 1]$)

Gibt den Anteil der aktiven Partikel in der Packung einer Fraktion an, der Rest besteht aus Wandkugeln aus höheren Fraktionen. Besteht die Fraktion aus nur einer Fraktion, ist der Anteil der aktiven Partikel immer 1, egal welcher Wert für diesen Parameter eingestellt wurde.

AnzahlFraktionen (Ganzzahl ≥ 1)

Legt die Anzahl der benötigten Fraktionen für die Simulation fest. Bei Verwendung der Online-Fraktionierung, wird die Anzahl der benötigten Fraktionen von der Simulation selbst ermittelt. Der hier eingestellte Wert wird dann ignoriert.

aequidistanteDurchmesserintervalle (boolescher Wert)

Legt fest, dass die Fraktionen immer ein gleich breites Durchmesserintervall umfassen.

lzFraktionierung (boolescher Wert)

Legt fest, dass die Online-Fraktionierung benutzt wird.

initialePackdichte (Fließkommazahl > 0)

Legt die Packungsdichte zu Beginn der Simulation fest. Die Simulation legt anhand dieses Wertes die Containergröße fest.

maxDichteMinderung (Fließkommazahl > 0)

Legt fest, um wie viel Prozent die Packungsdichte während einer Containervergrößerung maximal gesenkt werden darf.

anzahlVerkleinerungen (Ganzzahl ≥ 0)

Legt fest, wie oft der Container wieder verkleinert werden soll.

Zielueberlappungsrate (Fließkommazahl > 0)

Legt die Überlappungsrate fest, ab der die Packung als überlappungsfrei gilt. Nach unterschreiten dieses Wertes wird der Container wieder verkleinert oder, falls die eingestellte Anzahl Verkleinerungen bereits erreicht ist, die Simulation der Packung beendet.

maxMinKugelRatio (Fließkommazahl ≥ 0)

Dieser Parameter wird nur bei eingeschalteter Online-Fraktionierung benutzt. Er legt einen Faktor fest für den maximalen Größenunterschied zwischen kleinstem und größtem Partikel in einer Packung. Wird der Wert 0 eingetragen, findet keine Begrenzung statt.

Mischungsdateiname (Zeichenkette)

Gibt den Namen (inklusive Pfad) der Datei mit der Korngrößenverteilung an.

Containerform (boolescher Wert)

Gibt an, ob ein würfelförmiger (0) oder ein zylinderförmiger (1) Container verwendet werden soll.

ZylinderDurchmesser (Fließkommazahl > 0)

Falls ein zylinderförmiger Container verwendet wird, muss hier ein fixer Durchmesser für den Zylinder angegeben werden. Bei Größenveränderungen des Containers während der Simulation wird dieser Durchmesser beibehalten, nur die Höhe des Zylinders wird verändert. Es wird dieselbe Einheit wie bei der Korngrößenverteilung benutzt.

periodischerRand (boolescher Wert)

Schaltet periodische Randbedingungen ein oder aus.

alternativePartikelform (boolescher Wert)

Schaltet nicht kugelförmige Partikelformen ein oder aus.

Partikeldateiname (Zeichenkette)

Gibt den Namen (inklusive Pfad) der Datei mit der (als Kugelcluster dargestellten) Partikelform an.

Parameter für die Visualisierung**PovRay** (boolescher Wert)

Schaltet die Ausgabe einer Pov-Ray [POV13] Datei ein oder aus.

QuteMol (boolescher Wert)

Schaltet die Ausgabe einer QuteMol [TCM06, Qut07] Datei ein oder aus.

Falls die Ausgabe für beide Visualisierer ausgeschaltet wurde, dann haben alle folgenden Parameter keine Wirkung. Die folgenden 3 Parameter gelten für beide Visualisierer.

Ausgangsbild (boolescher Wert)

Schaltet die Ausgabe eines Bildes der Packung nach der Initialisierung ein oder aus.

Endbild (boolescher Wert)

Schaltet die Ausgabe eines Bildes der fertig entlappten Packung ein oder aus.

Visdateiname (Zeichenkette)

Gibt den Namen (inklusive Pfad) der auszugebenden Datei an. Die Dateierweiterung für jeden der beiden Visualisierer wird automatisch angehängt und muss deshalb nicht angegeben werden.

Die folgenden Parameter gelten nur für Pov-Ray.

schnittvideo (boolescher Wert)

Schaltet die Ausgabe eines Schnittvideos ein oder aus. Für dieses Video werden eine Reihe einzelner Pov-Ray Dateien ausgegeben, die jeweils einen Schnitt durch die Packung beschreiben. Nach dem Rendern kann aus den einzelnen Bildern, mit Hilfe eines externen Programms zur Videobearbeitung, ein Video zusammengesetzt werden, das einen parallel zur X-Y-Ebene durch die Packung wandernden Schnitt zeigt.

schnittbilder (Ganzzahl ≥ 1)

Legt fest, aus wie vielen Einzelbildern das Schnittvideo bestehen soll.

bewegungsvideo (boolescher Wert)

Schaltet die Ausgabe eines Bewegungsvideos ein oder aus. Das Video wird auf die gleiche Weise erstellt wie das Schnittvideo, es zeigt die Bewegungen der Partikel während des Entlappungsprozesses der Packung.

bewFrameIntervall (Ganzzahl ≥ 1)

Gibt an, nach wie vielen Iterationsschritten jeweils ein Bild für das Bewegungsvideo erzeugt werden soll.

bewFrameAnzahl (Ganzzahl ≥ 0)

Legt fest, aus wie vielen Einzelbildern das Bewegungsvideo bestehen soll. Wenn 0 eingetragen wird, findet keine Begrenzung statt, dann werden Bilder bis zum Ende der Simulation erzeugt.

periodischeAnsicht (boolescher Wert)

Schaltet die periodische Ansicht bei der Darstellung des Ausgangs- und Endbildes sowie des Bewegungsvideos ein oder aus. Dazu werden die Kugeln am oberen, unteren, linken und rechten Containerrand in einer dunkleren Farbe weiter gezeichnet. Ist der periodische Rand aber ausgeschaltet, so hat auch dieser Parameter keine Wirkung.

schnitt (boolescher Wert)

Legt fest, ob die Packung bei der Darstellung im Ausgangs- und Endbild sowie im Bewegungsvideo aufgeschnitten sein soll.

schnittkoordinate (Fließkommazahl)

Legt die Z-Koordinate fest, an der der Schnitt parallel zur X-Y-Ebene durchgeführt wird.

schnittmodus (boolescher Wert)

Legt fest, ob nur ein einfacher Schnitt durchgeführt wird (0) oder ob noch ein zweiter Schnitt dicht hinter dem ersten Schnitt erfolgt (1), so dass nur eine dünne Schicht der Packung sichtbar ist.

xRotation (boolescher Wert)

yRotation (boolescher Wert)

zRotation (boolescher Wert)

Mit POV-Ray ist es möglich, ein Video mit einer Kamerafahrt um die Szene (die Packung) herum zu erzeugen. Mit diesen 3 Parametern kann die Rotationsachse bestimmt werden, um die herum die Kamerafahrt erfolgt. Dies gilt für das Ausgangs- und das Endbild.

titel (Zeichenkette)

Legt einen Titel fest, der im Bild angezeigt wird.

Parameter für weitere Ausgaben

verteilungsdateiAnlegen (boolescher Wert)

Schaltet die Ausgabe einer Datei (pro Fraktion), die die jeweils simulierte Korngrößenverteilung ausgibt, ein oder aus.

verteilungsdateiname (Zeichenkette)

Gibt den Namen (inklusive Pfad) der auszugebenden Verteilungsdatei an.

ablaufdateiAnlegen (boolescher Wert)

Schaltet die Ausgabe einer Datei (pro Fraktion), die die Statusdaten jedes Iterationsschrittes ausgibt, ein oder aus.

ablaufdateiname (Zeichenkette)

Gibt den Namen (inklusive Pfad) der auszugebenden Ablaufdatei an.

PKFDateiAnlegen (boolescher Wert)

Schaltet die Ausgabe einer Datei (pro Fraktion), die die Werte der Paarkorrelationsfunktion ausgibt, ein oder aus.

PKFDateiname (Zeichenkette)

Gibt den Namen (inklusive Pfad) der auszugebenden Datei der Paarkorrelationsfunktion an.

PKFSchrittweite (Fließkommazahl > 0)

Legt die Schrittweite zwischen je zwei Werten fest, für die die Paarkorrelationsfunktion berechnet wird.

PKFMaxWert (Fließkommazahl > 0)

Gibt den maximale Wert an, bis zu dem die Paarkorrelationsfunktion berechnet wird.

koordinatenMittelwertBerechnen (boolescher Wert)

Schaltet die Berechnung und Ausgabe der mittleren Koordinaten aller Kugeln ein oder aus.

koordinatendateiAnlegen (boolescher Wert)

Schaltet die Ausgabe einer Datei, die die Koordinaten und Radien aller Kugeln enthält, ein oder aus.

koordinatendateiname (Zeichenkette)

Gibt den Namen (inklusive Pfad) der auszugebenden Koordinatendatei an.

wandeffektdateiAnlegen (boolescher Wert)

Schaltet die Ausgabe einer Datei, die die lokale Packungsdichte abhängig vom Abstand zur Containerwand enthält, ein oder aus.

wandeffektdateiname (Zeichenkette)

Gibt den Namen (inklusive Pfad) der auszugebenden Wandeffektdatei an.

wandeffektSchrittweite (Fließkommazahl > 0)

Legt die Schrittweite zwischen je zwei Abständen zur Containerwand, in denen die lokale Packungsdichte berechnet wird, fest.

testpunkte (Ganzzahl ≥ 1)

Legt die Anzahl der Testpunkte fest, die für die Bestimmung der lokalen Packungsdichte benutzt werden (Monte-Carlo-Methode).

ergebnisdateiAnlegen (boolescher Wert)

Schaltet die Ausgabe einer Ergebnisdatei ein oder aus. Diese Datei enthält eine Auflistung aller Parametereinstellungen sowie alle Zwischenergebnisse pro Fraktion und die Gesamtpackungsdichte.

ergebnisdateiname (Zeichenkette)

Gibt den Namen (inklusive Pfad) der auszugebenden Ergebnisdatei an.

keineEingaben (boolescher Wert)

Legt fest, ob das Programm nach dem Start ohne weitere Benutzereingaben laufen soll. Das ist insbesondere für den Batchbetrieb wichtig.

A.1.1 RaSim mit grafischer Oberfläche

Um die Bedienung der Packungssimulation auch für Endanwender ohne Kenntnisse der Implementierungsdetails zu ermöglichen, wurde eine grafische Oberfläche entwickelt, die sich an Standardanwendungen orientiert. Die Oberfläche gliedert sich in 3 Ansichten, die in Abbildung A.1 dargestellt sind. Auf der ersten Ansicht wird die Zusammensetzung der Mischung eingetragen. Dazu wird für jede Komponente der Mengenanteil in Prozent, der Name, die Reindichte und der Dateiname (inklusive Pfad) der Korngrößenverteilung angegeben. Die Anwendung berechnet daraus die Gesamtkorngrößenverteilung der Mischung, die für die Simulation der Packung benötigt wird. Auf der zweiten Ansicht werden die in der Konfigurationsdatei zur Verfügung stehenden Parameter angezeigt. Der Endanwender hat hier die Möglichkeit die gewünschten Einstellungen vorzunehmen. Parameter, die für den Endanwender unwichtig sind, können auch auf Standardwerte gesetzt und ausgeblendet werden. Wenn alle Einstellungen vorgenommen wurden, kann die Simulation über die Schaltfläche „Simulation starten“ gestartet werden. Sobald sie fertig gestellt ist, wird der Inhalt der Ergebnisdatei auf der dritten Ansicht der Oberfläche angezeigt.

Neben der Erleichterung der Bedienung bietet die grafische Oberfläche auch Funktionen, die der Verwaltung der Simulationsdaten dienen. Über die Menüpunkte „Datei->Profil hinzufügen...“ und „Datei->Profil auswählen“ können Profile erstellt und zwischen ihnen gewechselt werden. Ein Profil steht für eine Gruppe von zusammengehörenden Simulationen. Es besteht aus einem Hauptverzeichnis mit 3 Unterverzeichnissen, in denen alle Daten der Simulationen gespeichert werden. Das Hauptverzeichnis ist nach dem Profil benannt, die Namen der 3 Unterverzeichnisse sind fix und lauten:

Parametereinstellungen In diesem Verzeichnis werden wiederverwendbare Voreinstellungen für die Simulationen gespeichert.

Rohdaten Hier werden die gemessenen Korngrößenverteilungen für die Mischungskomponenten gespeichert.

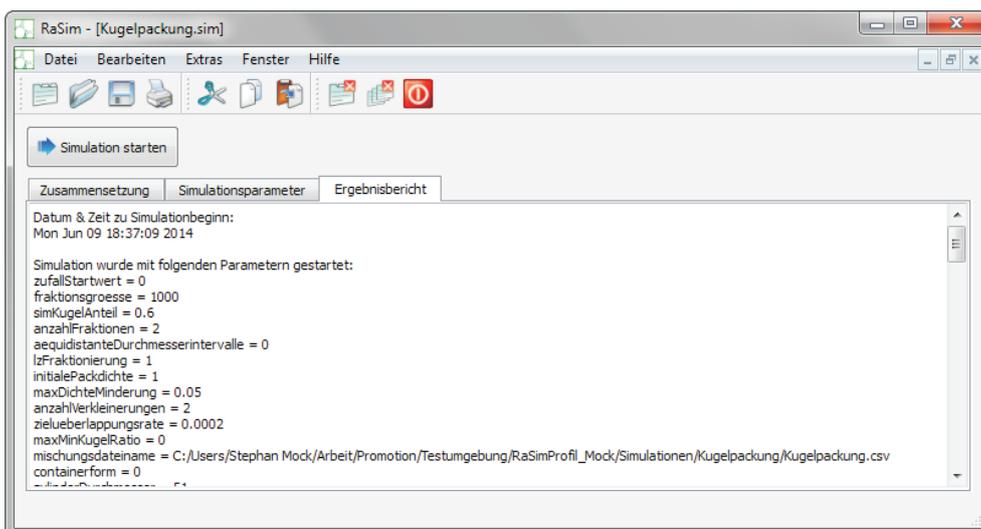
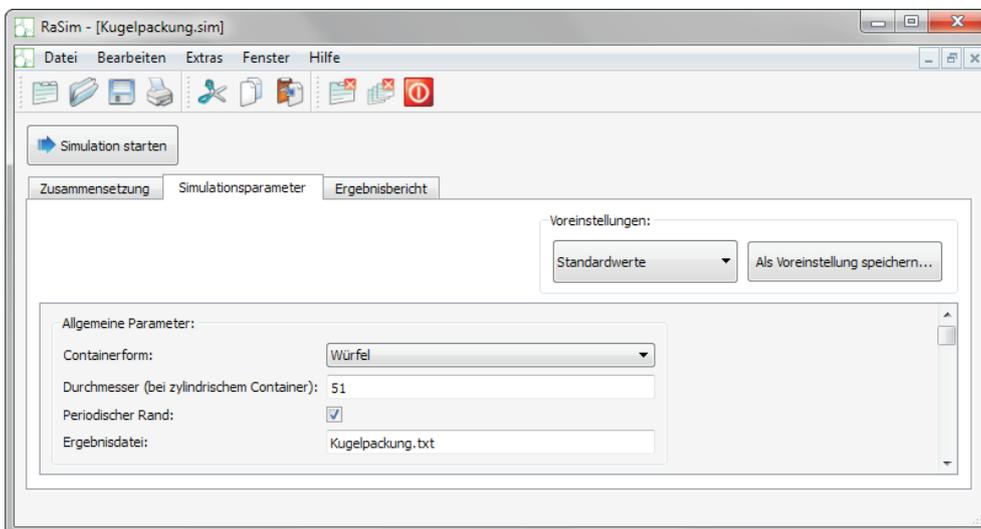
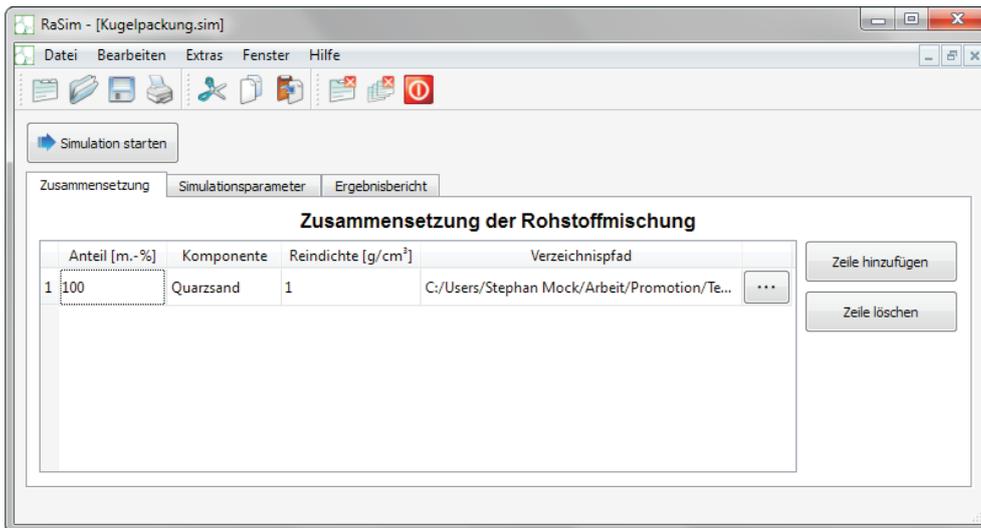


Abbildung A.1: RaSim-Version mit grafischer Oberfläche

Simulationen Dieses Verzeichnis enthält alle Einstellungen, Ergebnisse und Visualisierungen der durchgeführten Simulationen.

A.2 PSD-Morph

Ebenso wie RaSim wird auch die PSD-Morph Simulation über eine Konfigurationsdatei gesteuert. Die zur Verfügung stehenden Parameter sind im Folgenden aufgelistet.

Parameter für den Agglomeratengenerator

zufallStartwert (Ganzzahl ≥ 0)

Legt einen Startwert für den Zufallsgenerator fest. Der Standardwert ist 0, bei dieser Einstellung wird der Startwert aus dem Systemdatum generiert. Legt man einen anderen Wert fest, erlaubt dies die genaue Reproduktion eines Simulationslaufs; eine Option, die vor allem zur Fehlersuche während der Entwicklung eingesetzt werden kann.

Algorithmus (Ganzzahl aus $\{10, 11, 12, 13\}$)

Legt fest, welcher Algorithmus (Versionen des DLA-Algorithmus) zur Erzeugung der Agglomerate verwendet werden soll. Folgende Einstellungen sind möglich:

10: Einfacher DLA-Algorithmus, Kugeln werden in Clusternähe sofort angedockt.

11: RLA-Variante, Kugeln werden in Clusternähe gemäß einer festgelegten Wahrscheinlichkeit angedockt.

12: Genau wie 10, aber der Cluster wird während des Wachstums gedreht. Kristallförmige Cluster werden so verhindert.

13: Genau wie 11, aber der Cluster wird während des Wachstums gedreht. Kristallförmige Cluster werden so verhindert.

haftwahrscheinlichkeit (Fließkommazahl aus dem Intervall $(0, 1]$)

Legt die Wahrscheinlichkeit fest, mit der eine Kugel in Clusternähe angedockt wird. Gilt nur wenn für Algorithmus die Werte 11 oder 13 eingestellt wurden.

anzahlWiederholungen (Ganzzahl ≥ 0)

Legt fest, wie viele Agglomerate pro Generation erzeugt werden sollen. Bei Einstellung 0 legt die Simulation selbst fest, wie viele Agglomerate mindestens benötigt werden.

relativerGroessenfaktor (Fließkommazahl aus dem Intervall $(0, 1]$)

Die erzeugten Agglomerate werden innerhalb der Packungssimulation „RaSim“ durch Kugeln repräsentiert. Um der unregelmäßigen fraktalen Struktur der Agglomerate Rechnung zu tragen, sollten diese Kugeln etwas kleiner sein als die tatsächlich von „PSD-Morph“ erzeugten Cluster. Mit diesem Parameter kann der Skalierungsfaktor festgelegt werden. (In Kapitel 5.4.5 unter „Bestimmung der Größe der Agglomerate“ wird dieser Skalierungsfaktor mit τ bezeichnet.)

Mischungsdateiname (Zeichenkette)

Gibt den Namen (inklusive Pfad) der Datei mit der nicht agglomerierten Korngrößenverteilung ($q_0(x)$) an.

Agglomerationsneigung (Zeichenkette)

Gibt den Namen (inklusive Pfad) der Datei mit der Agglomerationsneigung ($a(x)$) an.

aggloVerteilung (Zeichenkette)

Legt den Namen (inklusive Pfad) der Datei für die Ausgabe der fertig agglomerierten Korngrößenverteilung mit den Agglomerateanteilen ($\hat{q}(x)$ und $h(x)$) fest.

partikeldateiAnlegen (Ganzzahl aus $\{0, 1, 2\}$)

Legt fest, ob die erzeugten Agglomerate als Partikeldateien gespeichert werden sollen. Folgende Einstellungen sind möglich:

0: Es werden keine Agglomerate gespeichert.

1: Alle Agglomerate jeder Generation werden gespeichert.

2: Nur das größte, kleinste und durchschnittlichste Agglomerat jeder Generation wird gespeichert.

PartikelDateiname (Zeichenkette)

Legt den Namen (inklusive Pfad) der auszugebenden Partikeldatei fest. Die Nummer der Generation wird automatisch angehängt, ebenso die Nummer des Agglomerates bzw. die Kürzel `maxGr`, `midGr` und `minGr` für das größte, durchschnittlichste und kleinste Agglomerat der Generation (je nach gewählter Einstellung des Parameters „partikeldateiAnlegen“).

Parameter für die Visualisierung

Genau wie die Partikelpackungen von „RaSim“ können auch die vom Agglomeratengenerator erzeugten Agglomerate visualisiert werden. Die Ausgabe richtet sich danach,

was im Parameter „partikeldateiAnlegen“ eingestellt wurde. Es werden also entweder keine Agglomerate, alle Agglomerate jeder Generation oder nur das kleinste, das größte und das durchschnittlichste Agglomerat jeder Generation ausgegeben.

PovRay (boolescher Wert)

Schaltet die Ausgabe einer Pov-Ray [POV13] Datei ein oder aus.

QuteMol (boolescher Wert)

Schaltet die Ausgabe einer QuteMol [TCM06, Qut07] Datei ein oder aus.

Visdateiname (Zeichenkette)

Legt den Namen (inklusive Pfad) der auszugebenden Visualisierungsdatei fest. Es werden die gleichen Kürzel wie beim Parameter „PartikelDateiname“ angehängt, ebenso die Dateiendungen für Pov-Ray bzw. QuteMol.

Die folgenden Parameter gelten nur für die Visualisierung mit Pov-Ray.

DLAFarbverlauf (boolescher Wert)

Legt fest, ob die Kugeln des Clusters gemäß der Reihenfolge ihres Andockens eingefärbt werden sollen. Zusätzlich wird eine dünne Scheibe aus dem Cluster heraus geschnitten, so dass der Farbverlauf in seinem Inneren sichtbar wird. Einige Beispiele für diese Art der Visualisierung finden sich in Abbildung 6.8.

VonFarbe (Vektor aus Fließkommazahlen)

NachFarbe (Vektor aus Fließkommazahlen)

Diese beiden Parameter legen die erste und die letzte Farbe für den Farbverlauf der Kugeln im Cluster fest. Die Farbe wird als RGB-Wert angegeben, wobei die Anteile für Rot, Grün und Blau jeweils als Fließkommazahl aus dem Intervall $[0, 1]$ angegeben werden. Die drei Werte des Vektors werden durch einen Bindestrich voneinander getrennt, so dass beispielsweise der Eintrag $0.0-0.0-1.0$ die Farbe Blau und der Eintrag $1.0-0.0-0.0$ die Farbe Rot ergibt.

halbeScheibenBreite (Fließkommazahl > 0)

Legt die halbe Breite der Scheibe fest, die aus dem Cluster geschnitten wird wenn der Parameter „DLAFarbverlauf“ auf 1 gesetzt wird.

xRotation (boolescher Wert)

yRotation (boolescher Wert)

zRotation (boolescher Wert)

Mit POV-Ray ist es möglich, ein Video mit einer Kamerafahrt um die Szene (das

Agglomerat) herum zu erzeugen. Mit diesen 3 Parametern kann die Rotationsachse bestimmt werden, um die herum die Kamerafahrt erfolgt.

visualisiereHuellkugeln (boolescher Wert)

Legt fest, ob eine transparente Hüllkugel um den Cluster (das Agglomerat) herum angezeigt werden soll.

maxTraceLevel (Ganzzahl ≥ 1)

Legt die maximale Rekursionstiefe für den Ray-Tracing Algorithmus fest. Der Wert muss nur erhöht werden, wenn im Bild mehrere transparente Objekte übereinander angezeigt werden sollen.

titel (Zeichenkette)

Legt einen Titel fest, der im Bild angezeigt wird.

Weitere Parameter

ergebnisdateiAnlegen (boolescher Wert)

Schaltet die Ausgabe einer Ergebnisdatei ein oder aus. Diese Datei enthält eine Auflistung aller Parametereinstellungen sowie die Ergebnisse jeder erzeugten Agglomeratengeneration (die Abbruchgröße, die Größe der erzeugten Agglomerate, die mittlere Anzahl der enthaltenen Primärpartikel und die Anzahl der erzeugten Agglomerate).

ergebnisdateiname (Zeichenkette)

Gibt den Namen (inklusive Pfad) der auszugebenden Ergebnisdatei an.

keineEingaben (boolescher Wert)

Legt fest, ob das Programm nach dem Start ohne weitere Benutzereingaben laufen soll. Das ist insbesondere für den Batchbetrieb wichtig.

A.3 Ein- und Ausgabedateien

Die gegebenen Korngrößenverteilungen (Mischungsdateien), die als Eingabe für RaSim und PSD-Morph verwendet werden, liegen als Tabellen in Form von CSV-Dateien (Dateiendung .csv) vor. Als Trennzeichen für die Spalten der Tabellen dient ein Tabulatorschritt. In der selben Weise sind auch die Agglomerationsneigung (Eingabe von PSD-Morph) und die agglomerierte Korngrößenverteilung mit den darin enthaltenen Agglomerateanteilen (Ausgabe von PSD-Morph) gespeichert.

Die Mischungsdatei

Die gegebene Korngrößenverteilung für RaSim bzw. PSD-Morph liegt in folgendem Format vor:

```
DICHTE  2.66672
ANZAHL
0.04    1.1277511
0.044   1.1801647
...
1377
```

Die erste Zeile enthält die Reindichte des Stoffes in g/cm^3 . Diese Angabe wird gebraucht, wenn aus den Korngrößenverteilungen mehrerer Komponenten die Gesamtkorngrößenverteilung der Mischung berechnet werden soll. Für die Simulation der agglomerierten Korngrößenverteilung mit PSD-Morph sowie der Packung mit RaSim wird die Reindichte nicht gebraucht. In der zweiten Zeile wird angegeben, ob es sich bei der dargestellte Verteilung um eine Anzahlverteilung $q_0(x)$ (ANZAHL) oder um eine Volumenverteilung $q_3(x)$ (VOLUMEN) handelt. Ab der dritten Zeile folgt dann die Verteilung. Die erste Spalte enthält die Korngrößen in μm , während die zweite Zeile den zugehörigen Anteil in Prozent an der Gesamtmenge (Anzahl Partikel, oder Volumen) enthält. Alle angegebenen Anteile gelten jeweils für ein Korngrößenintervall. Die vor einem Anteil stehende Korngröße bildet die untere Intervallgrenze während die Korngröße in der nächsten Zeile die obere Intervallgrenze angibt. Die letzte Zeile enthält keinen Anteil mehr, dort steht nur noch die obere Grenze des letzten Korngrößenintervalls.

Die Agglomerationsneigung

Die Agglomerationsneigung $a(x)$ wird nach dem gleichen Muster gespeichert wie die Mischungsdatei.

```
0.04    0.800
0.044   0.507
...
```

Hier wird zu jedem Korngrößenintervall eine Agglomerationsneigung als Fließkommazahl aus dem Intervall $[0, 1]$ angegeben.

Die agglomerierte Korngrößenverteilung

Die agglomerierte Korngrößenverteilung, die als Ergebnis einer PSD-Morph Simulation erzeugt wird, liegt in folgendem Format vor:

```
DICHTE  2.66672
ANZAHL
KG       Anz.Anteile  Agglo.Anteile  innere Dichte
0.04    1.1277511     0              0
0.044   1.1801647     0              0
0.048   1.5848345     0              0
0.053   2.5420803     0              0
0.058   4.0200051     0              0
0.064   5.6752869     0.24279921    0.43304424
...
1377
```

Neben den bereits aus der Mischungsdatei bekannten Angaben in den ersten zwei Zeilen, enthält diese Datei in der dritten Zeile die Überschriften für die 4 Spalten der folgenden Tabelle. Die ersten Spalte enthält (wie schon in der Mischungsdatei und der Agglomerationsneigung) die Korngrößen in μm . Die zweite Spalte enthält die zugehörigen Anteile in Prozent, wobei es sich hier um die agglomerierte Verteilung $\hat{q}(x)$ handelt. Die in dieser Verteilung enthaltenen Agglomerateanteile $h(x)$ sind in der dritten Spalte dargestellt. Sowohl $\hat{q}(x)$ als auch $h(x)$ werden hier stets als Anzahlanteile angegeben. Die vierte Spalte enthält die mittlere innere Dichte der für jedes Korngrößenintervall erzeugten Agglomerate.

Die Partikeldatei

Die Eingabe einer nicht kugelförmigen Partikelform in RaSim wie auch die Ausgabe der Agglomerateform aus PSD-Morph erfolgt als Partikeldatei (Dateiendung .par). Sie liegt in folgendem Format vor:

```
OBERFLAECHE  0.0113522
VOLUMEN      6.04287e-005
-0.0173545   -0.0136482     0.00372669     0.0204929
0.0158605    0.0121543     -0.00223276    0.0219868
...
```

In den ersten beiden Zeilen sind die Oberfläche und das Volumen des dargestellten Partikels angegeben. Ab der dritten Zeile sind die Koordinaten und Abmessungen der Kugeln angegeben, aus denen der Partikel zusammengesetzt ist. Die ersten drei Spalten enthalten die X-, Y- und Z-Koordinaten und die vierte Spalte die Radien der Kugeln.

A.4 Sonstige Software

Dieses Dokument sowie viele der enthaltenen Abbildungen wurden mit \LaTeX erstellt. Abbildungen für die dies nicht gilt, wurden mit folgender Software erstellt:

- **QuteMol** [TCM06, Qut07]
 - Abbildung 4.1
 - Abbildung 6.1 Teil (b)
 - Abbildung 6.2
 - Abbildung 6.3 Teil (b)
 - Abbildung 6.10 Teil (a-c)
 - Abbildung 7.1 (Packungsbilder)
- **Blender** [Ble94]
 - Abbildung 2.3
 - Abbildung 2.5 Teil (a)
 - Abbildung 2.7 Teil (a)
 - Abbildung 3.6
- **LibreOffice** [Lib11]
 - Abbildung 2.4
 - Abbildung 2.8
 - Abbildung 3.3
 - Abbildung 3.4
 - Abbildung 5.3
 - Abbildung 5.13
 - Abbildung 6.1 Teil (a)

- **POV-Ray [POV13]**
 - Abbildung 4.4 (Packungsbilder)
 - Abbildung 6.3 Teil (a)
 - Abbildung 6.8

Literaturverzeichnis

- [AM72] ADAMS, D. J. ; MATHESON, A. J.: Computation of Dense Random Packings of Hard Spheres. In: *The Journal of Chemical Physics* 56 (1972), Nr. 5, S. 1989–1994
- [Anh04] ANH, Ho C.: *Modellierung der Partikelagglomeration im Rahmen des Euler/Lagrange-Verfahrens und Anwendung zur Berechnung der Staubabscheidung im Zyklon*, Martin-Luther-Universität Halle-Wittenberg, Diss., 2004
- [AT91] ALLEN, M. P. ; TILDESLEY, D. J.: *Computer Simulation of Liquids*. Oxford University Press, 1991
- [AW59] ALDER, B. J. ; WAINWRIGHT, T. E.: Studies in Molecular Dynamics I. General Method. In: *The Journal of Chemical Physics* 31 (1959), Nr. 2, S. 459–466
- [BAS09] *Merkblatt ELKEM MICROSILICA GRADE 971U*. www.basf-cc.de, 2009. – BASF Construction Polymers GmbH Geschäftsbereich Betonzusatzmittel
- [BB85] BALL, R. C. ; BRADY, R. M.: Large scale lattice effect in diffusion-limited aggregation. In: *Journal of Physics A: Mathematical and General* 18 (1985), S. L809–L813
- [BBS02] BEZRUKOV, Alexander ; BARGIEL, Monika ; STOYAN, Dietrich: Statistical Analysis of Simulated Random Packings of Spheres. In: *Particle & Particle Systems Characterization* 19 (2002), Nr. 2, S. 111–118
- [BDM⁺88] BARNESLEY, Michael F. ; DEVANEY, Robert L. ; MANDELBROT, Benoît B. ; PEITGEN, Heinz-Otto ; SAUPE, Dietmar ; VOSS, Richard F.: *The Science of Fractal Images*. Springer, 1988
- [BH46] BROWN, R. L. ; HAWKSLEY, P. G. W.: Effect of Container Walls on Packing Density of Particles. In: *Nature* 157 (1946), May, S. 585. <http://dx.doi.org/10.1038/157585a0>. – DOI 10.1038/157585a0

- [BJ00] BAISH, James W. ; JAIN, Rakesh K.: Fractals and Cancer. In: *Cancer Research* 60 (2000), S. 3683–3688
- [Übl02] ÜBLER, Wolfgang: *Erhöhung der thermischen Leitfähigkeit elektrisch isolierender Polymerwerkstoffe*, Universität Erlangen-Nürnberg, Diss., 2002
- [Ble94] *Blender*. www.blender.org, 1994. – 3D-Grafiksoftware
- [Blu06] BLUM, Jürgen: Dust agglomeration. In: *Advances in Physics* 55 (2006), Nr. 7-8, S. 881–941
- [BM91] BARGIEŁ, Monika ; MOŚCIŃSKI, Jacek: C-language program for the irregular close packing of hard spheres. In: *Computer Physics Communications* 64 (1991), Nr. 1, S. 183–192
- [BPP94] BORKOVEC, M. ; PARIS, W. de ; PEIKERT, R.: The Fraktal Dimension of the Apollonian Sphere Packing. In: *Fractals* 2 (1994), Nr. 4, S. 521–526
- [BPW⁺91] BORHO, Klaus ; POLKE, Reinhard ; WINTERMANTEL, Klaus ; SCHUBERT, Helmar ; SOMMER, Karl: Produkteigenschaften und Verfahrenstechnik. In: *Chemieingenieurtechnik* 63 (1991), Nr. 8, S. 792–808
- [Bra32] BRADLEY, R.S.: LXXIX. The cohesive force between solid surfaces and the surface energy of solids. In: *Philosophical Magazine Series 7* 13 (1932), Nr. 86, S. 853–862
- [BS06] BEZRUKOV, Alexander ; STOYAN, Dietrich: Simulation and Statistical Analysis of Random Packings of Ellipsoids. In: *Particle & Particle Systems Characterization* 23 (2006), Nr. 5, S. 388–398
- [BSMM99] *Kapitel Krümmung und Windung*. In: BRONSTEIN, I. N. ; SEMENDJAJEW, K. A. ; MUSIOL, G. ; MÜHLIG, H.: *Taschenbuch der Mathematik*. Harri Deutsch, 1999, S. 240–241
- [CS79] CUNDALL, P A. ; STRACK, O D L.: A discrete numerical model for granular assemblies. In: *Géotechnique* 29 (1979), Nr. 1, 47–65. <http://www.icevirtuallibrary.com/content/article/10.1680/geot.1979.29.1.47>
- [DBCW07] DOMINIK, Carsten ; BLUM, Jürgen ; CUZZI, Jeffrey N. ; WURM, Gerhard: Growth of Dust as the Initial Step Toward Planet Formation. In: *Protostars and Planets V* (2007), S. 783–800

- [Ede61] EDEN, Murray: A two-dimensional growth process. In: *Proceedings of Fourth Berkeley Symposium on Mathematics, Statistics, and Probability* Bd. 4, University of California Press, Berkeley, 1961, S. 223–239
- [Els09] ELSNER, Antje: *Computergestützte Simulation und Analyse zufällig dichter Kugelpackungen*, TU Bergakademie Freiberg, Diss., 2009
- [Faz07] FAZEKAS, Sandor: *Distinct Element Simulations of Granular Materials*, Budapest University of Technology and Economics, Department of Theoretical Physics, Diss., 2007
- [FL04] FILBET, Francis ; LAURENÇOT, Philippe: Numerical simulation of the Smoluchowski coagulation equation. In: *SIAM Journal on Scientific Computing* 25 (2004), Nr. 6, S. 2004–2028
- [FT06] FULLER, William B. ; THOMPSON, Sanford E.: The laws of proportioning concrete. In: *Transactions of the American Society of Civil Engineers* 57 (1906), Nr. 2, S. 67–143
- [Fur29] FURNAS, C. C.: Flow of Gases Through Beds of Broken Solids. In: *Bureau of Mines Bulletin* 307 (1929), S. 74–83
- [GDK89] GREST, Gary S. ; DÜNWEIG, Burkhard ; KREMER, Kurt: Vectorized link cell Fortran code for molecular dynamics simulations for a large number of particles. In: *Computer Physics Communications* 55 (1989), Nr. 3, S. 269–285
- [GS87a] GIBSON, K. D. ; SCHERAGA, H. A.: Exact calculation of the volume and surface area of fused hard-sphere molecules with unequal atomic radii. In: *Molecular Physics* 62 (1987), Nr. 5, S. 1247–1265
- [GS87b] GIBSON, K. D. ; SCHERAGA, H. A.: Volume of the Intersection of Three Spheres of Unequal Size - A Simplified Formula. In: *The Journal of Physical Chemistry* 91 (1987), S. 4121–4122
- [GS88] GIBSON, K. D. ; SCHERAGA, H. A.: Surface area of the intersection of three spheres with unequal radii - A simplified analytical formula. In: *Molecular Physics* 64 (1988), Nr. 4, S. 641–644
- [Hal00a] HALES, Thomas C.: Cannonballs and Honeycombs. In: *Notices of the American Mathematical Society* 47 (2000), Nr. 4, S. 440–449
- [Hal00b] HALSEY, Thomas C.: Diffusion-limited aggregation: A model for pattern formation. In: *Physics Today* Nov. 2000 (2000), S. 36–41

- [Hal05] HALES, Thomas C.: A proof of the Kepler conjecture. In: *Annals of Mathematics* 162 (2005), Nr. 3, S. 1065–1185
- [Hal06] HALES, Thomas C.: Historical Overview of the Kepler Conjecture. In: *Discrete and Computational Geometry* 36 (2006), Nr. 1, S. 5–20
- [HAM37] HAMAKER, H. C.: The London - van der Waals attraction between spherical particles. In: *Physica* 4 (1937), Nr. 10, S. 1058–1072
- [HE98] HE, D. ; EKERE, N. N.: Computer simulation of powder compaction of spherical particles. In: *Journal of Materials Science Letters* 17 (1998), Nr. 20, S. 1723–1725
- [HEC99] HE, D. ; EKERE, N. N. ; CAI, L.: Computer simulation of random packing of unequal particles. In: *Physical Review E: Statistical, Nonlinear and Soft Matter Physics* 60 (1999), Nr. 6, S. 7098–7104
- [HF11] HALES, Thomas C. ; FERGUSON, Samuel P.: *The Kepler Conjecture: The Hales-Ferguson Proof*. Springer, 2011
- [HGE74] HOCKNEY, R. W. ; GOEL, S. P. ; EASTWOOD, J. W.: Quiet High-Resolution Computer Models of a Plasma. In: *Journal of Computational Physics* 14 (1974), Nr. 2, S. 148–158
- [HS88] HERRMANN, Hans J. ; STANLEY, H. E.: The fractal dimension of the minimum path in two- and three-dimensional percolation. In: *Journal of Physics A: Mathematical and General* 21 (1988), S. L829–L833
- [HSS99] HUIJSER, R. H. ; SAR, E. G. d. ; SCHELLING, R.: Cosmic Dust Aggregation in Microgravity - Flight Report of the CODAG module on MASER 8. In: *14th ESA Symposium on European Rocket and Balloon Programmes and Related Research*, 1999, S. 511–516
- [JN92] JAEGER, H. M. ; NAGEL, Sidney R.: Physics of Granular States. In: *Science* 255 (1992), S. 1523–1531
- [JT80] JACKINS, Chris L. ; TANIMOTO, Steven L.: Oct-trees and their use in representing three-dimensional objects. In: *Computer Graphics and Image Processing* 14 (1980), Nr. 3, S. 249–270. [http://dx.doi.org/doi:10.1016/0146-664x\(80\)90055-6](http://dx.doi.org/doi:10.1016/0146-664x(80)90055-6). – DOI doi:10.1016/0146-664x(80)90055-6

- [Kje07] KJELDEN, Ane M.: *Consolidation behavior of cement-based systems*, Technical University of Denmark, Department of Civil Engineering, Diss., 2007
- [Kol08] *Kapitel Inversionsmethode*. In: KOLONKO, Michael: *Stochastische Simulation - Grundlagen, Algorithmen und Anwendungen*. 1. Vieweg + Teubner, 2008, S. 85–95
- [KR11] KOLONKO, Michael ; RASCHDORF, Steffen: A comparison of data structures for the simulation of polydisperse particle packings. In: *International Journal for Numerical Methods in Engineering* 85 (2011), S. 625–639. <http://dx.doi.org/10.1002/nme.2988>. – DOI 10.1002/nme.2988
- [KRW10] KOLONKO, Michael ; RASCHDORF, Steffen ; WÄSCH, Dominic: A hierarchical approach to simulate the packing density of particle mixtures on a computer. In: *Granular Matter* 12 (2010), Nr. 6, S. 629–643. <http://dx.doi.org/10.1007/s10035-010-0216-5>. – DOI 10.1007/s10035-010-0216-5
- [Lee70] LEE, Do I.: Packing of Spheres and its effects on the viscosity of suspensions. In: *Journal of Paint Technology* 42 (1970), Nr. 550, S. 579–587
- [Len04] *Kapitel Transforms*. In: LENGYEL, Eric: *Mathematics for 3D Game Programming & Computer Graphics*. 2. Charles River Media, 2004, S. 71–99
- [Lib11] *LibreOffice*. de.libreoffice.org, 2011. – Office-Paket
- [LWD⁺06] LINSENBUHLER, M. ; WERTH, J.H. ; DAMMER, S.M. ; KNUDSEN, H.A. ; HINRICHSSEN, H. ; WIRTH, K.-E. ; WOLF, D.E.: Cluster size distribution of charged nanopowders in suspensions. In: *Powder technology* 167 (2006), Nr. 3, S. 124–133
- [Mea82] MEAGHER, Donald: Geometric modeling using octree encoding. In: *Computer Graphics and Image Processing* 19 (1982), Nr. 2, S. 129–147
- [Mea85] MEAKIN, Paul: The structure of two-dimensional Witten-Sander aggregates. In: *Journal of Physics A: Mathematical and General* 18 (1985), S. L661–L666
- [Mea95] MEAKIN, Paul: Progress in DLA research. In: *Physica D: Nonlinear Phenomena* 86 (1995), S. 104–112
- [MM44] MADOW, William G. ; MADOW, Lillian H.: On the Theory of Systematic Sampling, I. In: *The Annals of Mathematical Statistics* 15 (1944), Nr. 1, S. 1–24

- [Mue04] MUELLER, Chris: Typed DLAs. In: *P700: Fractals and Pattern Formation - Project Study* (2004)
- [MV87] MEAKIN, Paul ; VICSEK, Tamás: Diffusion-limited aggregation with radial bias. In: *Journal of Physics A: Mathematical and General* 20 (1987), S. L171–L175
- [NB97] NEINHUIS, C. ; BARTHLOTT, W.: Characterization and Distribution of Water-repellent, Self-cleaning Plant Surfaces. In: *Annals of Botany* 79 (1997), Nr. 6, S. 667–677
- [Pal09] PALM, Sebastian: *Optimierung der Raumauffüllung und der Komponentenverteilung von Multikompositzementen*, TU Clausthal, Diss., 2009
- [POV13] *POV-Ray*. www.povray.org, 2013. – Ray-Tracer
- [Qut07] *QuteMol*. qutemol.sourceforge.net, 2007. – Molecular Visualization System
- [Ras06] RASCHDORF, Steffen: *Simulation der Raumauffüllung von Beton mit vorgegebener Korngrößenverteilung*. TU Clausthal, Dipl., 2006
- [Ras10] RASCHDORF, Steffen: *Bestimmung der Raumauffüllung von Partikelmischungen - Modelle und Datenstrukturen für die Simulation durch Kugelpackungen*, TU Clausthal, Diss., 2010
- [RSK05] RUNKANA, Venkataramana ; SOMASUNDARAN, P. ; KAPUR, P. C.: Reaction-Limited Aggregation in Presence of Short-Range Structural Forces. In: *AIChE Journal* 51 (2005), Nr. 4, S. 1233–1245
- [Rut62] RUTGERS, R.: Packing of Spheres. In: *Nature* 193 (1962), Feb, S. 465–466. <http://dx.doi.org/10.1038/193465a0>. – DOI 10.1038/193465a0
- [Sco60] SCOTT, David G.: Packing of Spheres: Packing of Equal Spheres. In: *Nature* 188 (1960), Dec, S. 908–909. <http://dx.doi.org/10.1038/188908a0>. – DOI 10.1038/188908a0
- [Smo16] SMOLUCHOWSKI, Marian V.: Drei Vorträge über Diffusion, Brownsche Bewegung und Koagulation von Kolloidteilchen. In: *Zeitschrift für Physik* 17 (1916), S. 557–585
- [Smo17] SMOLUCHOWSKI, Marian V.: Versuch einer mathematischen Theorie der Koagulationskinetik kolloidaler Lösungen. In: *Zeitschrift für physikalische Chemie* 92 (1917), Nr. 2, S. 129–168

- [Spe03] SPENGLER, Georg: *Koagulation von Staubteilchen*. 2003. – Seminarvortrag zum Thema „Urmaterie des Sonnensystems“
- [Sti09] STIESS, Matthias: *Mechanische Verfahrenstechnik - Partikeltechnologie 1*. Springer, 2009
- [SY07] SIIRIÄ, Simo ; YLIRUUSI, Jouko: Particle packing simulations based on Newtonian mechanics. In: *Powder Technology* 174 (2007), Nr. 3, S. 82–92
- [TCM06] TARINI, Marco ; CIGNONI, Paolo ; MONTANI, Claudio: Ambient Occlusion and Edge Cueing to Enhance Real Time Molecular Visualization. In: *IEEE Transactions on Visualization and Computer Graphics* 12 (2006), Nr. 5, S. 1237–1244
- [TCW68] TORY, E. M. ; COCHRANE, N. A. ; WADDELL, S. R.: Anisotropy in Simulated Random Packing of Equal Spheres. In: *Nature* 220 (1968), S. 1023–1024
- [TM89] TOLMAN, Susan ; MEAKIN, Paul: Off-lattice and hypercubic-lattice models for diffusion-limited aggregation in dimensionalities 2 – 8. In: *Phys. Rev. A* 40 (1989), Jul, 428–437. <http://dx.doi.org/10.1103/PhysRevA.40.428>. – DOI 10.1103/PhysRevA.40.428
- [TTD00] TORQUATO, S. ; TRUSKETT, T.M. ; DEBENEDETTI, P.G.: Is Random Close Packing of Spheres Well Defined? In: *Physical Review Letters* 84 (2000), Nr. 10, S. 2064–2067
- [Ulr00] *Kapitel Loose Octrees*. In: ULRICH, Thatcher: *Game Programming Gems*. 1. Charles River Media, 2000, S. 444–453
- [VC06] VALVERDE, Jose M. ; CASTELLANOS, Antonio: Random loose packing of cohesive granular materials. In: *Europhysics Letters* 75 (2006), Nr. 6, S. 985–991. <http://dx.doi.org/10.1209/epl/i2006-10208-4>. – DOI 10.1209/epl/i2006-10208-4
- [VC07] VALVERDE, Jose M. ; CASTELLANOS, Antonio: Compactification of fine powders: from fluidized agglomerates to primary particles. In: *Granular Matter* 9 (2007), Nr. 1, S. 19–24. <http://dx.doi.org/10.1007/s10035-006-0022-2>. – DOI 10.1007/s10035-006-0022-2
- [VDZ08] *Zement-Taschenbuch, 51. Ausgabe*. Verlag Bau + Technik GmbH, 2008. – Verein Deutscher Zementwerke e.V.

- [Ver67] VERLET, Loup: Computer „Experiments“ on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. In: *Physical Review* 159 (1967), Nr. 1, S. 98–103
- [Ver04] VERNET, Christian P.: Ultra-Durable Concretes: Structure at the Micro- and Nanoscale. In: *MRS Bulletin* 29 (2004), Nr. 5, S. 324–327
- [Wac06] WACKENHUT, Martin: *Fast Algorithms to simulate extremely polydisperse media*, Universität Stuttgart, Institut für Computerphysik Fakultät Mathematik und Physik, Diss., 2006
- [WK08a] WONG, Henry H. C. ; KWAN, Albert K. H.: Packing density of cementitious materials: part 1—measurement using a wet packing method. In: *Materials and Structures* 41 (2008), Nr. 4, S. 689–701
- [WK08b] WONG, Henry H. C. ; KWAN, Albert K. H.: Packing density of cementitious materials: part 2—packing and flow of OPC + PFA + CSF. In: *Materials and Structures* 41 (2008), Nr. 4, S. 773–784
- [WLD⁺03] WERTH, J.H. ; LINSENBÜHLER, M. ; DAMMER, S.M. ; FARKAS, Z. ; HINRICHSEN, H. ; WIRTH, K.-E. ; WOLF, D.E.: Agglomeration of charged nanopowders in suspensions. In: *Powder Technology* 133 (2003), Nr. 1, S. 106–112
- [Wol11] WOLTER, Albrecht: *Vorlesung Baustofflehre*. 2011. – TU-Clausthal
- [WS81] WITTEN, T. A. ; SANDER, L. M.: Diffusion-Limited Aggregation, a Kinetic Critical Phenomenon. In: *Physical Review Letters* 47 (1981), Nr. 19, S. 1400–1403
- [WW37] WHITE, H. E. ; WALTON, S. F.: PARTICLE PACKING AND PARTICLE SHAPE. In: *Journal of the American Ceramic Society* 20 (1937), S. 155–166. <http://dx.doi.org/10.1111/j.1151-2916.1937.tb19882.x> – DOI 10.1111/j.1151-2916.1937.tb19882.x
- [Yat48] YATES, F.: Systematic Sampling. In: *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences* 241 (1948), Nr. 834, S. 345–377
- [YMT96] YANG, A. ; MILLER, C. T. ; TURCOLIVER, L. D.: Simulation of correlated and uncorrelated packing of random size spheres. In: *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics* 53 (1996), Nr. 2, S. 1516–1524

- [YMWZ12] YI, Weiyu ; MOCK, Stephan ; WELLER, Rene ; ZACHMANN, Gabriel: Sphere-Spring Systems and their Application to Hand Animation. In: GEIGER, Christian (Hrsg.) ; HERDER, Jens (Hrsg.) ; VIERJAHN, Tom (Hrsg.): *Virtuelle und Erweiterte Realität - 9. Workshop der GI-Fachgruppe VR/AR*, Shaker Verlag, 2012, S. 131–142
- [YS88] YU, A. B. ; STANDISH, N.: An analytical-parametric theory of the random packing of particles. In: *Powder Technology* 55 (1988), Nr. 3, S. 171–186
- [YS91] YU, A. B. ; STANDISH, N.: Estimation of the porosity of particle mixtures by a linear-mixture packing model. In: *Industrial & Engineering Chemistry Research* 30 (1991), Nr. 6, S. 1372–1385
- [YXBI⁺08] YE, Guo ; XIN, Huang ; BAO-LIN, Zhu ; BAO-GUO, Ma ; HONG-BO, Zhu: Method for calculating packing density of powder particles in paste with continuous particle size distribution. In: *Powder Technology* 187 (2008), S. 88–93
- [YZYC06] YANG, R. Y. ; ZOU, R. P. ; YU, A. B. ; CHOI, S. K.: Pore structure of the packing of fine particles. In: *Journal of Colloid and Interface Science* 299 (2006), S. 719–725