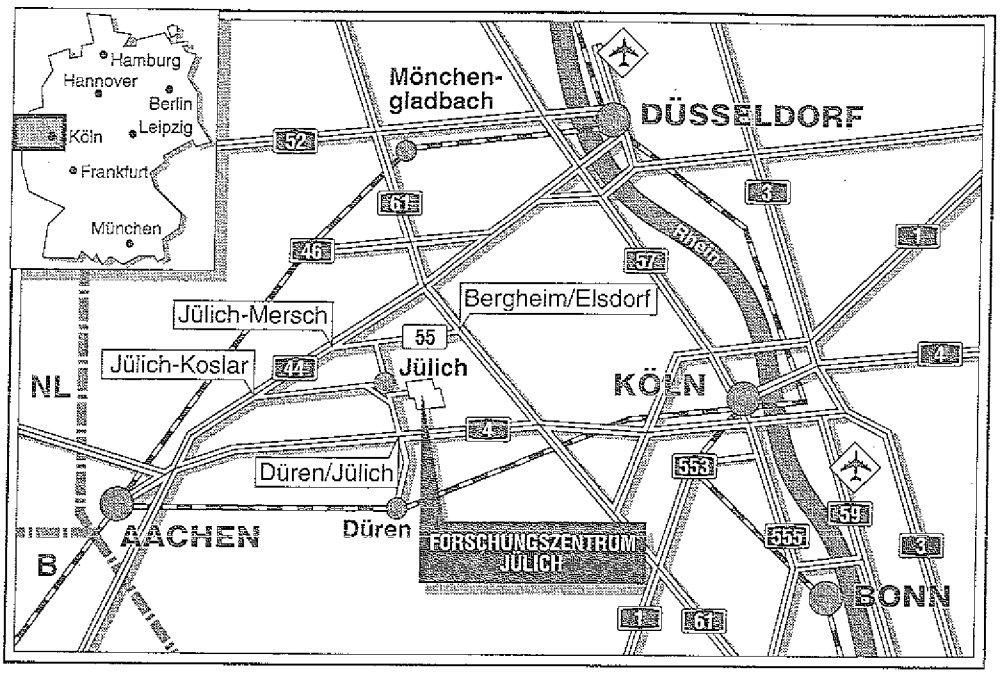
FORSCHUNGSZENTRUM JÜLICH GmbH

Höchstleistungsrechenzentrum

Untersuchungen an stochastischen Modellen zum Laplaceschen und Rißwachstum

Peter Ossadnik



Berichte des Forschungszentrums Jülich ; 2745
ISSN 0366-0885
Höchstleistungsrechenzentrum Jül-2745
D 38 (Diss. Universität Köln)

Zu beziehen durch: Forschungszentrum Jülich GmbH · Zentralbibliothek
Postfach 1913 · D-5170 Jülich · Bundesrepublik Deutschland
Telefon: 02461/61-6102 · Telefax: 02461/61-6103 · Telex: 833556-70 kfa d

Untersuchungen an stochastischen Modellen zum Laplaceschen und Rißwachstum

Peter Ossadnik

Inhaltsverzeichnis

Kapitel 1

Vorwort

Wie zerbricht ein elastisches Material, welches äußeren Kräften unterworfen wird? Welche Bruchmuster werden entwickelt, welche Eigenschaften besitzen diese Risse und welche Parameter können benutzt werden, sie zu kontrollieren? Diese und ähnliche Fragen werden auf dem Gebiet der Bruchmechanik gestellt. Ihre theoretische Beantwortung erweist sich als extrem schwierig, was einerseits dadurch bestimmt wird, daß selbst die einfachsten mathematischen Formulierungen dieser Fragen kaum analytisch gelöst werden können. Andererseits stellt man experimentell eine derartige Fülle von unterschiedlichsten Bruchphänomenen fest, daß selbst die korrekte Formulierung des Problems schwierig erscheint.

In der einfachsten Form beschreibt man ein elastisches Medium im Rahmen der linearen Elastizitätstheorie mit Hilfe der Laméschen Gleichung durch ein von den äußeren Kräften bestimmtes Spannungsfeld $\underline{\underline{\sigma}}$. Ein Riß wird in einer solchen Formulierung als kraftfreie Oberfläche angesehen, die sich unter dem Einfluß des Spannungsfeldes forwärtsbewegt. Die ersten theoretischen Untersuchungen über das Verhalten von Rissen in spröden Körpern stammen von Griffith [1], der als Begründer der quantitativen Riß- und Bruchtheorie angesehen werden kann. Er betrachtete im wesentlichen in einem elastischen Medium unter uniaxialem Zug die Spannungsverteilung um ein elliptisches Loch, dessen kleine Halbachse b gegenüber der großen Halbachse a verschwindet: $b/a \to 0$. Diese Konfiguration bezeichnet man als "Griffith-Riß". Lösungen dieses Problems lassen sich sehr elegant mittels konformer Transformation aus der Lösung für ein kreisrundes Loch gewinnen. Mit Hilfe dieser Lösung sind später auch die Spannungsfelder für kompliziertere Geometrien mit mehreren getrennten Griffith-Rissen errechnet worden. Die komplizierten, realistischen und in der Natur beobachteten Rißgeometrien entziehen sich wegen ihrer Komplexität jedoch allen analytischen Berechnungsbemühungen.

In dieser Formulierung ist ein Riß bisher ein statisches Objekt, aber selbst hier ist die analytische Berechnung der Spannungsfelder nur in Spezialfällen möglich. Allerdings muß man beachten, daß das Wachstum eines Risses ein dynamischer Prozeß ist, der sich als eine Differentialgleichung mit variablen Randbedingungen — ein *Moving-Boundary Problem* — darstellt. Derartige Probleme sind tatsächlich nur noch in extremen Spezialfällen lösbar und meist ist eine lineare Stabilitätsanalyse des dynamischen Problems [2–4] der einzige Lösungsansatz.

Neben dieser prinzipiellen Schwierigkeit, realistische Probleme analytisch nicht handhaben zu können, findet man zusätzlich noch ein weiteres Problem. Realistische Materialien weisen

alle eine strukturelle Unordnung auf, die zu einer stochastischen Verteilung der elastischen Module und Festigkeiten führt. Unter diesem globalen Begriff "Unordnung" verbergen wir hier alle möglichen Effekte, die eine Abweichung von einer idealen Gitterstruktur bedeuten und von mikroskopischen Gitterdefekten — Gitterfehlstellen oder Versetzungen — über mesoskopische Effekte wie Korngrenzen bis hin zu makroskopischen Defekten wie z. B. in porösen Materialien führen können. Dies führt z. B. dazu, daß die Kräfte, die man benötigt um ein reelles Werkstück zu brechen und die man experimentell bestimmen kann, um mehrere Größenordnungen kleiner sind, als die analytischen Vorhersagen, die man für einen idealen Kristall analytisch treffen kann.

Auf den für uns interessanten — also makroskopischen — Längenskalen sind die detaillierte Struktur und die exakten physikalischen Prozesse solcher räumlichen Inhomogenitäten allerdings nicht relevant und können reduziert werden auf eine statistische Verteilung der Materialeigenschaften wie Festigkeit und Module. Daher können über das Rißwachstum Aussagen mit Hilfe von Methoden der statistischen Physik gemacht werden.

Wegen der Komplexität des Problems sind numerische Methoden offensichtlich beinahe unumgänglich. Die angewendeten Methoden reichen von Finite-Elemente-Methoden über Randintegral-Rechnungen hin zu Gittermodellen. Dabei wird das betrachtete Medium an einer endlichen Menge von Punkten diskretisiert, auf der dann eine Lösung der elastischen Gleichungen bei vorgegebenen Randbedingungen berechnet wird. Nach bestimmten Wachstumsregeln werden Bonds bestimmt, die gebrochen werden sollen, wodurch neue Randbedingungen erzeugt werden, die eine vollkommen neue Berechnung der elastischen Felder erfordern. Derartige Methoden erweisen sich als derart rechenintensiv, daß die möglichen Gittergrößen stark eingeschränkt sind und nur kleine Risse mit einigen hundert bis zu wenigen tausend gebrochenen Bonds simuliert werden können.

Darüberhinaus ist nicht klar, welchen Einfluß die zugrundeliegenden Gitter ausüben. In dem skalaren Analogon des Elastizitätsproblems, der einfachen Laplace-Gleichung, ist sehr wohl bekannt, daß die Präsenz eines Gitters einen sehr starken Einfluß ausübt und die gesamte resultierende Struktur dominiert. Eine Möglichkeit, derartige Gittereffekte zu umgehen, besteht darin, Zufallsgitter statt regulärer Gitter zur Diskretisierung zu benutzen. Man kann jene aus regulären Gitter generieren, indem man alle Gitterplätze um eine kleine Strecke — die kleiner als die Gitterkonstante ist — in eine zufällige Richtung verschiebt [5]. Erst kürzlich wurde in Simulationen gezeigt, daß mit dieser Methode Gittereffekte bei Untersuchung von Laplaceschem Wachstum unterdrückt werden können [6]. Zusätzlich wurden derartige Zufallsgitter auch bei der Simulation von Rißwachstumsphänomenen angewendet [7,8]. Zwar kann man mit dieser Methode Gittereffekte vermeiden, allerdings gestattet auch die Benutzung von Zufallsgittern keine Generierung großer Strukturen, da immer noch die numerische Lösung der Laplace- bzw. der Lamé-Gleichung notwendig ist.

Für die Simulation von Laplaceschem Wachstum existiert allerdings eine Methode, die sich als extrem effizient für die Generierung großer Strukturen erwiesen hat. Das hierfür benutzte Modell ist die 1981 von Witten und Sander eingeführte "Diffusion Limited Aggregation" (DLA) [9], mit deren Hilfe Aggregate mit vielen Millionen Teilchen simuliert werden können, ohne explizit die Laplace-Gleichung lösen zu müssen. Der Weltrekord in DLA liegt zur Zeit bei Clustern mit 50 Millionen Teilchen und wurde im Rahmen dieser Arbeit aufgestellt. In der Abbildung 1 ist der Weltrekord-Cluster gezeigt.

Roux zeigte 1987, daß auch für das Elastizitätsproblem ein Random-Walk konstruiert werden kann, der im Grenzfall beliebig kleiner Schrittweite gegen die Lösung der Lamé-Gleichung mit

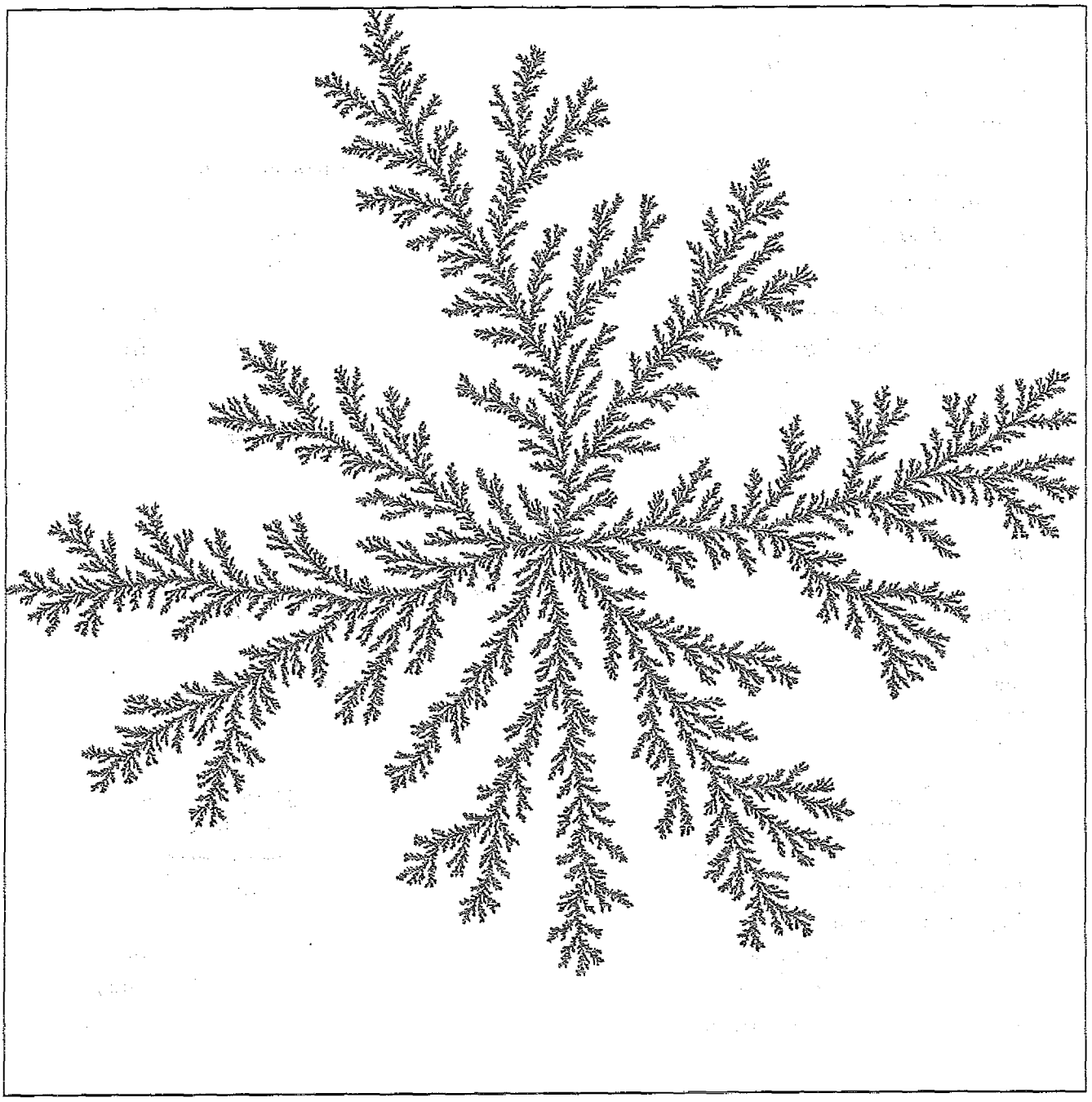


Abbildung 1 Off-Lattice DLA Cluster mit 51.200.000 Teilchen. Dies ist der bisher größte überhaupt generierte Off-Lattice DLA Cluster.

vorgegebenen Randbedingungen konvergiert [10]. Seine Arbeit erweitert somit die klassischen Überlegungen von Chandrasekhar über Random Flights und Brownsche Bewegung [11] und stellt eine Verbindung zwischen Laplaceschem und Laméschem Wachstum her. Allerdings wurden seine Ergebnisse bisher nicht weiter für explizite numerische Simulationen benutzt, so daß deren praktische Konsequenzen unbekannt sind. Allerdings weisen sie einen völlig neuen Weg für die Simulation von Rißphänomenen.

Mit diesem letzten Punkt beschäftigt sich die vorliegende Arbeit, die im einzelnen wie folgt gegliedert ist.

In Kapitel 2 werden die Grundlagen des Laplaceschen und des Rißwachstums wiederholt und die wesentlichen Begriffe eingeführt. In diesem Kapitel soll auch wesentlich detaillierter auf das interessante Problem des Nicht-Gleichgewichts Wachstums eingegangen werden und bekannte Resultate sollen dort skizziert werden.

Die erste Aufgabe in dieser Dissertation besteht darin, einen effizienten Algorithmus für die Simulation von Laplaceschem Wachstum zu entwickeln, da das Lamésche Wachstum im wesentlichen als Erweiterung von DLA modelliert werden soll. Mit Hilfe dieses Programms sollen zunächst einige neue Untersuchungen über DLA angestellt werden.

Als erstes Problem wird in Kapitel 3 die Frage des Multiscaling in DLA untersucht. Dabei gilt es zu untersuchen, ob in großen DLA Clustern eine radiale Abhängigkeit der fraktalen Dimension herrscht, oder ob die fraktale Dimension vom Abstand vom Cluster-Ursprung unabhängig ist. In dieser ersten Anwendung auf DLA soll der benutzte Algorithmus ausführlich diskutiert und beschrieben werden. In diesem Kapitel sollen auch einige Fragen der Selbstähnlichkeit großer DLA Cluster behandelt werden, wobei wir Cluster, deren Massen über drei verschiedene Größenordnungen variieren, vergleichen.

Eine zentrale Größe zur Beschreibung von DLA Clustern ist die Wahrscheinlichkeit $p(i)$, daß DLA Cluster an einer bestimmten Stelle i wachsen. Mit dieser Größe besitzt man eine vollkommene Beschreibung der DLA-Struktur, allerdings ist sie nur mit sehr aufwendigen Methoden und nur für kleine Cluster numerisch bestimmbar. Für sehr große Cluster ist dagegen die radiale Wachstumswahrscheinlichkeit $p(r)$ zu bestimmen, welche in Kapitel 4 untersucht werden soll. Dies Wahrscheinlichkeit $p(r)$ ist direkt verantwortlich für das in Kapitel 3 untersuchte Multiscaling.

Die auffällige Baumstruktur von DLA Clustern wurde zum Anlaß genommen, ein hierarchisches Ordnungsschema für DLA zu untersuchen. Man stellt fest, daß DLA Cluster eine hierarchische Struktur besitzen und daß die Eigenschaften der verschiedenen Hierarchien einfachen Regeln folgen. Diese Frage soll in Kapitel 5 behandelt werden. Dort werden wir die Längen, Breiten und Massen von Zweigen verschiedener Ordnungen studieren. Wir wollen zeigen, daß man eine einfache Skalenbeziehung zwischen der Länge und der Breite von Zweigen messen kann und damit ein für Lattice-DLA wohlbekanntes Resultat verallgemeinern.

In dem darauf folgenden Kapitel 6 soll dann das Wachstum von Rissen in elastischen Medien zum Thema werden. In bisher gebräuchlichen Methoden der numerischen Simulation wird ein elastisches Gitter zur Berechnung von *Gleichgewichtszuständen* in dem repräsentierten Medium benutzt: Man bestimmt numerisch in jedem Zeitschritt die Gleichgewichtskonfiguration des Gitters. Dies entspricht dem Wachstum von Rissen, bei der nur wenig Energie frei wird. Wir wollen dagegen die Rißbildung unter dem Einfluß einer Schockwelle und somit eine starke Nicht-Gleichgewichts Situation auf einem Gitter studieren. Dazu werden wir ein Methode verwenden, die stark an einen Zellular-Automaten erinnert und auch im Spring-Block Modell zum Studium von Erdbebenphänomenen Anwendung findet.

In dem Kapitel 7 soll letztlich die wesentliche und wichtigste Frage dieser Arbeit untersucht werden. Dort werden wir in enger Analogie zu DLA ein Modell zur Bildung von Rissen in elastischen, ungeordneten Materialien mit Hilfe von Random-Walkern entwickeln und untersuchen. Hier sollen verschiedene Varianten dieses Modells diskutiert und die Einflüsse verschiedener Parameter studiert werden. Außerdem sollen die dort gewonnen Ergebnisse mit Resultaten aus Experimenten verglichen werden.

Schließlich sollen in Kapitel 8 die einzelnen Ergebnisse zusammengefaßt und abschließend diskutiert werden.

Kapitel 2

Einleitung und Grundlagen

Diese Arbeit wird sich mit dem Rißwachstum in elastischen Medien beschäftigen. Dieses Problem kann in das hochinteressante Gebiet der Wachstumsphänomene eingeordnet werden, welches wegen seiner Reichhaltigkeit und Strukturfülle in den letzten Jahren einen starken Einfluß hat.

Ein Grund für das Interesse an Wachstumsmodellen sind die unterschiedlichen neuen Skalengesetze, die man finden kann und die eng mit der geometrischen Form der gewachsenen Strukturen verknüpft sind. Die Bildung dieser Strukturen wird häufig von Instabilitäten der zugrundeliegenden Physik gesteuert, welche zu einer verblüffenden Vielfalt von sehr oft fraktalen Formen führen kann. Aber was macht derartige Wachstumsmodelle so kompliziert? Die Antwort ist, daß Wachstum hier untrennbar mit Geschichte verbunden ist: Eine gewachsene Struktur hängt stark von der eigenen Form zu einem frühere Zeitpunkt ab. Verschiedene Konfigurationen zu verschiedenen Zeitpunkten sind nicht unabhängig, sondern in einer sehr komplizierten Weise durch die eigene Geschichte miteinander korreliert. Daher ist es normalerweise nicht möglich, das Problem durch Zustandssummen oder generierende Funktionen zu lösen, sondern neue Ansätze sind notwendig.

Ein Paradebeispiel für ein einfaches, struktur-reiches und immer noch unverstandenes Wachstumsmodell ist "Diffusion Limited Aggregation" (DLA). Es hat sich nicht nur gezeigt, daß DLA wichtige experimentelle Anwendungen besitzt, sondern DLA wird die wesentliche Grundlage für Rißwachstum in dieser Arbeit sein und die essentielle numerische Technik zur Simulation von Rißwachstum bereitstellen. Daher wird DLA auch einen breiten Teil dieser Arbeit einnehmen.

Im folgenden werde ich beide Phänomene, zunächst die "einfache" skalare DLA, später das "komplizierte" vektorielle Rißwachstum, einführen und grundlegende Resultate diskutieren.

2.1 Diffusion Limited Aggregation

Diffusion Limited Aggregation (DLA) wurde 1981 von Witten und Sander [9] als einfaches Modell für Teilchenaggregation eingeführt. Im Laufe der Zeit stellte sich seine bemerkenswerte Anwendbarkeit auf die unerschiedlichsten physikalischen Phänomene heraus. Als Auswahl sollen hier Fluß durch poröse Medien [12], chemische Auflösung, Verdrängung viskoser Flüssigkeiten [13], "Dielectric-Breakdown" [14], Wachstum von Bakterienkolonien [15], Kristallwachstum und Elektrodeposition [16] genannt werden. Insgesamt sind mehr als 100

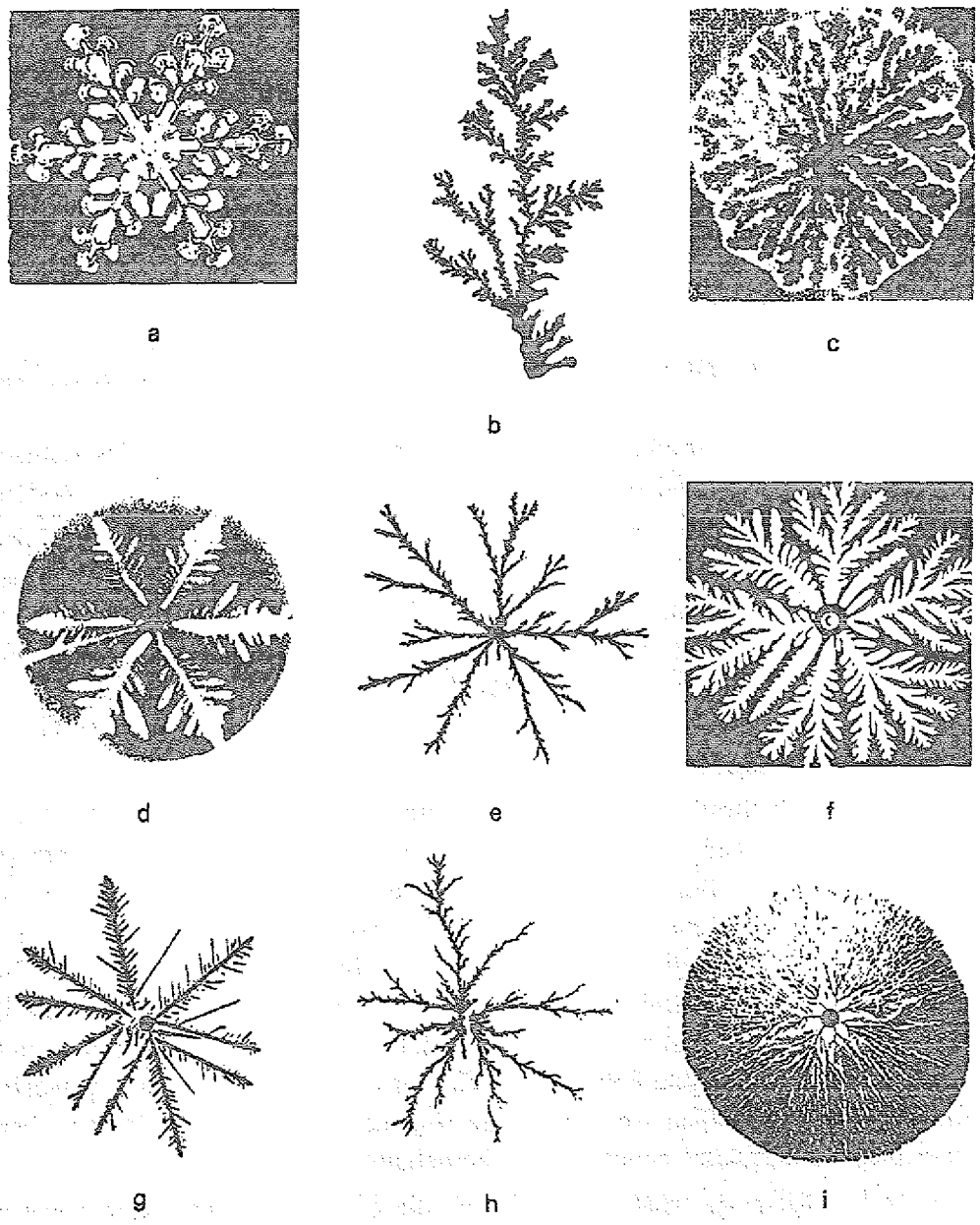


Abbildung 2 Laplacesche Wachstumsstrukturen für verschiedene Experimente. Man kann die drei wesentlichen Morphologien erkennen: (a, d, g) dendritisches Wachstum, (b, e, h) offene Verzweigungsstruktur (DLA), (c, f, i) dichte Zweigstruktur. Die entsprechenden Experimente sind: (a, b, c) Kristallisation, (d, e, f) Verdrängung einer viskosen Flüssigkeit "Viscous Fingering", (g, h, i) Elektrodeposition von Zink (entnommen aus [17]). Die einzelnen Abb. (a)-(i) entsprechen den Arbeiten [18–25].

verschiedene experimentelle Realisationen bekannt. Für einige der Realisationen ist bekannt, daß in ihnen die Strukturbildung von der Laplace-Gleichung bestimmt wird, bei anderen — wie z. B. dem Wachstum von Bakterienkolonien — findet man lediglich experimentell eine Übereinstimmung der Formen, während die Verbindung zur Laplace-Gleichung noch nicht geklärt ist.

Einige dieser Experimente sind in der Abbildung 2 wiedergegeben, andere findet man z. B. in dem Übersichtsartikel von Meakin [26], in dem Buch von Vicsek [27], oder in [28] bzw. [29].

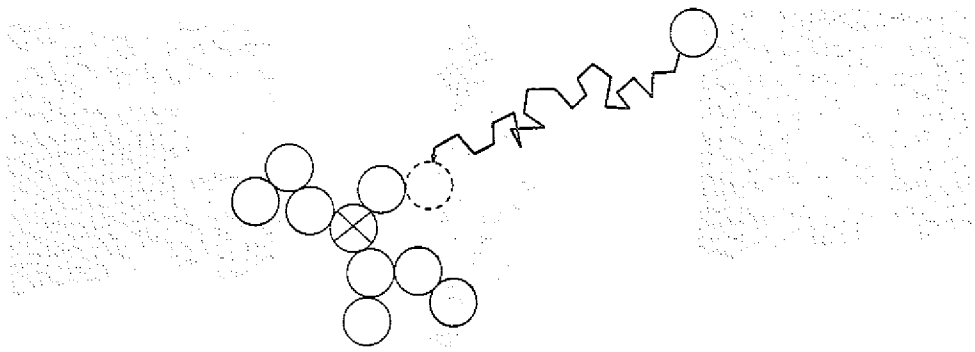


Abbildung 3 Grundlegender DLA Algorithmus. Das Seed-Teilchen ist mit ⊗ gekennzeichnet.

Wegen dieser breiten Anwendbarkeit wird DLA vielfach als der Prototyp des Laplaceschen Wachstums angesehen, manchmal sogar als "3–d Ising-Modell" der Wachstumsmodelle [30].

Wegen ihrer Relevanz für das Rißwachstum soll kurz die Versuchsanordnung zur Verdrängung viskoser Flüssigkeiten — "Viscous Fingering" — in einem Hele-Shaw-Zellen Experiment [31] beschrieben werden. Dabei befindet sich eine Flüssigkeit mit hoher Viskosität zwischen zwei parallelen Glasplatten, die nur einen Abstand von ca. 1 mm haben. An einer bestimmten Stelle preßt man eine Flüssigkeit mit niedrigerer Viskosität zwischen die Platten und verdrängt die andere Flüssigkeit. Je nach Art der Flüssigkeiten, Struktur der Glasplatten und Druck kann man Strukturen wie in Abb. 2d-2f finden.

Alle diese verschiedenen Strukturen können mit verschiedenen Varianten von DLA generiert werden. Die ursprüngliche Simulationsmethode ist in zwei Dimensionen die folgende [9] (vgl. Abb. 3): Man plaziert ein Teilchen in die Mitte eines unendlich großen Kreises. Danach startet man ein weiteres Teilchen von einem zufällig gewählten Punkt auf dem Kreisrand, welches durch den Kreis diffundiert, bis es auf das erste Teilchen trifft und sofort anhält, um mit jenem einen Cluster bestehend aus zwei Teilchen zu bilden. Danach wird ein nächstes Teilchen gestartet, welches wieder solange diffundiert, bis es auf den bisher generierten Cluster trifft. Dieser Vorgang wird wiederholt, bis ein Cluster der benötigten Größe gewachsen ist. Für die diffundierenden Teilchen werden wir im folgenden den üblichen Begriff "Random-Walker" oder einfach "Walker" benutzen und beibehalten.

Mit dieser einfachen Methode erhält man lediglich die offenen Zweigstrukturen wie in Abb. 2b, 2e und 2h. Mit anderen Varianten von DLA, in denen man eine anisotrope Wachstumsrichtung implementiert, kann man dendritische Strukturen wie in den Abbildungen 2a, 2d und 2g erhalten [32]. In einer weiteren Variante von DLA, in der man die Wachstumswahrscheinlichkeit variiert [33], kann man auch dichte Zweigstrukturen wie in Abb. 2i erhalten.

Ursprünglich wurden DLA Simulationen auf einem Gitter durchgeführt [9] — Lattice-DLA —, auf dem wie in einem "Gittergas" mit verschwindender Teilchendichte ein einzelner Walker in jedem Zeitschritt von einem Gitterplatz auf einen der nächsten Nachbarplätze springt. Dies wird wiederholt, bis der Walker einen Platz erreicht, der nächster Nachbar eines besetzten Platzes ist, und dort sitzen bleibt. Man muß beachten, daß hier die Wachstumsregel anisotrop ist. Das führt zu einem Crossover der Clusterstruktur [34]. Kleine Cluster mit wenigen tausend Teilchen sind isotrop während sehr große Cluster die Symmetrie des zugrundeliegenden Gitters wiederspiegeln: Lattice-DLA Cluster auf einem Rechteckgitter besitzen vier Hauptarme, die entlang der Gitterachsen wachsen [32]. Dieses Verhalten ist jedoch nicht nur ein Artefakt der Simulation! Wie man z. B. an der regelmäßigen Struktur eines Eiskristalls deutlich sieht, stabilisiert der Einfluß einer Anisotropie tatsächlich die Clusterspitzen, so daß eine

Tip-Splitting Instabilität unterdrückt wird. Das gleiche Phänomen kann in Hele-Shaw-Zell Experimenten beobachtet werden, in denen die räumliche Isotropie der Zelle künstlich gebrochen wird [35].

In einer wichtigen Variante von DLA [36–38] wird versucht, die statistischen Fluktuationen zu unterdrücken, indem nicht jedes eintreffende Teilchen am Cluster haftet, sondern lediglich jedes m-te an einem bestimmten Gitterplatz eintreffende Teilchen. Alle vorherigen werden ignoriert. Der gesamte Vorgang ist als "Noise-Reduced DLA" bekannt und der Parameter m heißt "Noise-Reduction" Parameter. Ein Wert $m = 1$ entspricht der üblichen DLA, während $N \to \infty$ einer exakten Lösung der Laplace-Gleichung auf dem Gitter entspricht.

Für eine feste Clustermasse M kann ein interessanter morphologischer Übergang der Clusterstruktur für wachsendes m beobachtet werden: Für kleine m erhält man die übliche verzweigte DLA-Struktur, während man für große m ($m = 400$) eine nadelartige Struktur findet, wobei die Anzahl und Richtung der Nadeln vom zugrundeliegenden Gitter bestimmt wird. In zwei Dimensionen findet man also auf einem Quadratgitter vier spitze Nadeln, die entlang der Gitterachsen wachsen. Eckmann et al. [39] analysieren mittels konformer Transformation den Einfluß von großen m auf die Struktur der Cluster und zeigen, daß man für Clustermassen $M < (\ln m)^3$ tatsächlich nadelartige Strukturen erhält, während für größere Massen $M < (\ln m)^3$ das (fraktale) dendritische Wachstum mit Seitenzweigen dominiert. Als Konsequenz erhält man, daß im Fall unendlicher Noise-Reduction für Lattice-DLA keine Seitenzweige auftreten, sondern lediglich zweiglose Nadeln wachsen. Auch dieses Beispiel illustriert das wichtige Wechselspiel zwischen statistischen Fluktuationen und Anisotropie im Laplaceschen Wachstum.

Zur Vermeidung derartiger Gittereffekte in Lattice-DLA betrachtet man daher Off-Lattice DLA. In diesem Modell bewegen sich die Teilchen nicht mehr auf einem Gitter, sondern diffundieren im Kontinuum und beenden ihren Random-Walk , sobald sie während eines Schrittes den Cluster berühren [40,41]. Hier ist offensichtlich die oben erwähnte Anisotropie beseitigt und die mit diesem Off-Lattice DLA generierten Cluster sind isotrop.

Trotz der einfachen mikroskopischen Regeln von DLA, sind die so erzeugten Cluster sehr kompliziert. Sie besitzen eine fraktale Struktur, die durch die fraktale Dimension D_f charakterisiert wird, welche durch die Abhängigkeit der Clustermasse M vom Trägheitsradius R_g

$$M \propto R_g^{D_f} \tag{1}$$

bestimmt wird. Der Trägheitsradius, für den im folgenden der übliche Begriff "Radius of Gyration" benutzt werden soll, ist definiert als der mittlere Abstand der Teilchenpositionen $\vec{r}_i$ vom Clusterschwerpunkt $\vec{r}_{CM}$

$$R_g = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(\vec{r}_i - \vec{r}_{CM})^2}. \tag{2}$$

Für Off-Lattice und kleine Lattice-DLA Cluster findet man eine fraktale Dimension $D_f = 1.715 \pm 0.003$ [42,43], während große Lattice-DLA Cluster wegen der oben beschriebenen Anisotropie eine Dimension von $D_f = 3/2$ [32] aufweisen.

Im folgenden soll die Verbindung zwischen DLA und Laplaceschem Wachstum, wie sie z. B. in [11,9] aufgezeigt wird, skizziert werden. Dazu betrachtet man die Dichte $u(\vec{r}, t)$ von dif-

fundierenden Random-Walkern. Deren Zeitentwicklung wird durch die Diffusionsgleichung

$$\frac{\partial u(\vec{r},t)}{\partial t} \propto \nabla^2 u(\vec{r},t) \tag{3}$$

bestimmt. Die Random-Walker werden am Cluster vernichtet, so daß man dort die Bedingung

$$u(\vec{r},t) = 0 \tag{4}$$

findet, während sie auf dem äußeren Rand mit konstanter Dichte erzeugt werden, so daß hier

$$u(\vec{r},t) = 1 \tag{5}$$

gilt. Die Wachstumswahrscheinlichkeit $v(\vec{r})$ auf der Clusteroberfläche ist bestimmt durch den Fluß $\vec{j}(\vec{r})$ von Walkern in den Cluster hinein und kann daher bestimmt werden als

$$v(\vec{r}) \propto \vec{j}(\vec{r}) \cdot \hat{n} \propto -\nabla u(\vec{r}) \cdot \hat{n} \tag{6}$$

Ist die Zeitskala für die Teilchendiffusion wesentlich größer als die Zeitskala des Wachstums, so kann man eine stationäre Teilchendichte annehmen $\partial u/\partial t = 0$ und die Diffusionsgleichung geht in die Laplace-Gleichung über. DLA-Wachstum wird daher durch das Laplace-Feld der Random-Walker um den Cluster herum bestimmt und ist also ein Prototyp für Laplacesches Wachstum, wodurch sich der Erfolg bei der Simulation Laplacescher Wachstumsphänomene ergibt.

Die Gleichungen (3)-(6) bilden gemeinsam eine Differentialgleichungs-System mit bewegten Rändern — ein *Moving-Boundary Problem*. Diese Art von Problemen ist natürlich schwieriger zu lösen, als das statische Problem selbst. In der Arbeit von Kessler, Koplik und Levine [44] werden eine Vielzahl unterschiedlicher Lösungsansätze diskutiert. Analytische Lösungen dieses Problems bleiben aber im wesentlichen beschränkt auf die Untersuchung eines gleichförmigen Wachstums — d. h. einer gleichförmigen Bewegung des Randes, wie z. B. in der klassischen Arbeit von Saffmann und Taylor [45,46] zum Fluß zweier Flüssigkeiten in einer Hele–Shaw Zelle [31] — bzw. eine lineare Stabilitätsanalyse für den Fall, daß der bewegte Rand Instabilitäten aufweist [44]. Gerade diese Instabilitäten führen zu der Bildung der physikalisch interessanten Muster. In der linearen Stabilitätsanalyse wird eine bekannte stabile Lösung des Moving-Boundary Problems räumlich oder zeitlich durch eine kleine periodische Perturbation gestört und in einer linearen Näherung untersucht, ob die Amplitude einer derartigen Anregung wächst oder schrumpft.

Trotz der einfachen lokalen Regeln in DLA und der Verbindung zur Laplace-Gleichung existiert noch keine befriedigende Theorie, die in der Lage wäre, DLA zu beschreiben. Selbst die Begründung der Fraktalität von DLA Clustern und eine exakte theoretische Bestimmung der fraktalen Dimension D_f fehlt bisher. Lediglich durch Mean-Field artige Rechnungen [47–49] und Ortsraum-Renormierungsmethoden [50] ist bisher eine Näherung der fraktalen Dimension gelungen. Für DLA in d Dimension findet z. B. Muthukumar [47]

$$D_f = \frac{d^2+1}{d+1} \tag{7}$$

Daneben haben Halsey und Leibig [51] ein Modell aufgestellt, in dem sie einen einfachen Zusammenhang zwischen der Masse eines Zweiges M_i und dessen gesamter Wachstumswahrscheinlichkeit P_i annehmen. Hiermit stellen sie einfache Bewegungsgleichungen für M_i und

P_i auf und studieren die Fixpunkte dieser Dynamik. Mit diesem Modell erhalten sie eine fraktale Dimension, die sehr nahe bei der Muthukumar Formel liegt. Wie üblich in Mean-Field artigen Rechnungen wird die Qualität der oben beschriebenen analytischen Resultate in höheren Dimensionen $d > 2$ besser, wie in Simulationen bis $d = 8$ gezeigt wurde [42], allerdings sind alle analytischen Ergebnisse in $d = 2$ unbefriedigend.

Ein wesentliches Problem bei der analytischen Berechnung der fraktalen Dimension insbesondere in $d = 2$ ist das langreichweitige Abschirmverhalten der Laplace-Gleichung. Dadurch sind die Wachstumsraten nicht nur von der lokalen Umgebung, sondern von der globalen Clusterstruktur abhängig.

Eine zentrale Rolle bei der Charakterisierung von DLA spielt daher das "harmonische Maß" p_i, welches die Wahrscheinlichkeit dafür angibt, daß der nächste Random-Walker den Cluster am Platz i berührt, und welches zuerst von Meakin et al. [52] studiert wurde. Die Menge der Wachstumswahrscheinlichkeiten $\{p_i\}$ beschreibt vollkommen die Struktur von DLA. Große Energie [53–55] wurde daher dem Studium der Verteilung der Wahrscheinlichkeiten $n(p, M)$ bei Clustern der Masse M, bzw. deren Momenten

$$Z_q(M) \equiv \sum_{i=1}^{M} p_i^q \tag{8}$$

gewidmet.

Man findet, daß in zwei Dimensionen die verschiedenen Momente für positive q mit jeweils unterschiedlichem Exponenten mit der Clustermasse M skalieren

$$Z_q \propto M^{-F(q)/D_f} \tag{9}$$

Hier ist $F(q)$ eine nichtlineare Funktion von q und damit hat man eine multifraktale Verteilung der $\{p_i\}$. Diese positiven Momente beschreiben die Plätze mit großen Wachstumswahrscheinlichkeiten, die nahe bei den Spitzen der Zweige liegen. Die innere Struktur der DLA Cluster wird dagegen von den negativen Momenten beschrieben, welche gerade die Plätze mit niedrigen Wachstumswahrscheinlichkeiten charakterisieren. Aber gerade diese niedrigen Wachstumswahrscheinlichkeiten sind nur schlecht durch numerische Rechnungen zu bestimmen. Allerdings steht fest, daß man für die negativen Momente q kein wohldefiniertes Skalenverhalten der Form (9) finden kann [56,55].

Statt der negativen Momente wurde das Verhalten der kleinsten aller Wachstumswahrscheinlichkeiten p_{min} als Beschreibung der niedrigen Wahrscheinlichkeiten studiert [57,58,54]. Eine Berechnung von p_{min} ist direkt mit einem Bild der im Clusterinneren liegenden Fjords verknüpft. Drei verschiedene Möglichkeiten der Massenabhängigkeit von p_{min} wurden diskutiert:

1.
$$p_{min}(M) \propto \exp\left(-AM^x\right) \tag{10}$$

   Dieses Verhalten findet man, wenn man sich die Fjords als lange, gerade Tunnel vorstellt [57].

2.
$$p_{min}(M) \propto M^{-\alpha_{max}/D_f} \tag{11}$$

   Ein solches Verhalten findet man für trichterförmige Fjords [58].

3.
$$\ln p_{min}(M) \propto (\ln M)^y \tag{12}$$

Ein solches Verhalten findet man in einem Modell, in dem selbstähnliche Hohlräume durch enge Kanäle verbunden werden [54] ("Void-Neck-Modell"). Dieses Verhalten wurde auch von unabhängigen analytischen Rechnungen bestätigt [59].

Bisher scheint das "Void-Neck-Modell" die beste Beschreibung der $\{p_i\}$ zu liefern.

Im Gegensatz zu zweidimensionalem DLA zeigen neuere Untersuchungen [60], daß in drei Dimensionen auch die negativen Momente ein Skalenverhalten (9) aufweisen. Als Begründung dient hierfür, daß in $2d$ DLA ein Zweig einen gesamten Raumbereich abschirmen kann, während in $3d$ Walker immer noch um den Zweig herumdiffundieren können und man eine "Ebene" benötigt, um einen größeren Raumbereich abzuschirmen.

2.2 Rißwachstum

Nach dieser Beschreibung des Laplaceschem Wachstum werde ich mich nun dem Wachstum von Rissen zuwenden. Rißwachstum in elastischen Materialien ist ein interessantes und sehr wichtiges Forschungsgebiet mit weitreichendem Einfluß sowohl im Ingenieur-Bereich als auch im Bereich der Grundlagenforschung. Gerade wegen seiner Reichhaltigkeit und seinem Einfluß wird dieses Gebiet ausgiebig studiert und selbstverständlich kann nur ein kleiner Teil des vorhandenen Wissens über Rißphänomene in dieser Einleitung dargestellt werden. Detailliertere Informationen findet man in vielen Lehrbüchern, wie z. B. [61], in der klassischen Sammlung von Liebowitz [62] oder in den neuen Artikelsammlungen von Herrmann und Roux [63] bzw. Charmet, Guyon und Roux [64]. Besonders die beiden letzten Sammlungen setzen sich mit dem zentralen Punkt der modernen Rißforschung, der räumlichen Unordnung realer Materialien auseinander. Eine sehr schöne Einführung in das Problem des Rißwachstums findet man auch in [65]. Hier soll im wesentlichen ein Überblick über die bekannten Rißphänomene gegeben werden.

Es ist leicht vorstellbar, daß die genaue Kenntnis der Brucheigenschaften der benutzten Werkstoffe beinahe lebensnotwendig im Bereich der konstruierenden Ingenieur-Tätigkeiten ist — sei es im Fahrzeugbau, im Flugzeugbau, im Schiffbau oder im Zivil-Ingenieur Bereich. Allzuoft hört man von schweren Unfällen, bei denen ein Materialfehler die Ursache war, weil die verwendeten Materialien ermüdet waren oder die Brucheigenschaften einfach nicht korrekt eingeschätzt worden sind. Daher ist auch klar, daß das Studium dieses Themas zusätzlich von großem finanziellem Interesse ist.

Besonders interessant ist in allen diesen Beispielen die große Bandbreite von Materialien, die benutzt werden und deren Verhalten untersucht werden muß. Die benutzten Werkstoffe sind u. a. Metalle, Stein, Beton, Glas, Keramiken und die verschiedensten Arten von Kunststoffen, die alle ein unterschiedliches Bruchverhalten aufweisen.

Ebenso interessant sind die unterschiedlichen Lastzustände, die in den obigen Beispielen auftreten können und zu unterschiedlichem Bruchverhalten führen. Neben einer einfachen statischen Dauerlast, die durch eine konstante Kraft auf ein Material ausgeübt wird, muß man noch eine Reihe anderer Lastformen berücksichtigen. Z. B. können Temperatur- und Feuchtigkeitsschwankungen das elastische Verhalten von Materialien drastisch beeinflussen und zu Spannungen im Material führen. Ein einfaches, aber drastisches Beispiel für diesen Punkt ist die Explosion der amerikanischen "Challenger"-Raumfähre, die ausgelöst wurde durch einen durch Kälte spröde gewordenen Dichtungs-O-Ring, der dann beim Start auftretenden Vibrationen nicht mehr standhielt. Daneben muß man berücksichtigen, daß Umwelteinflüsse

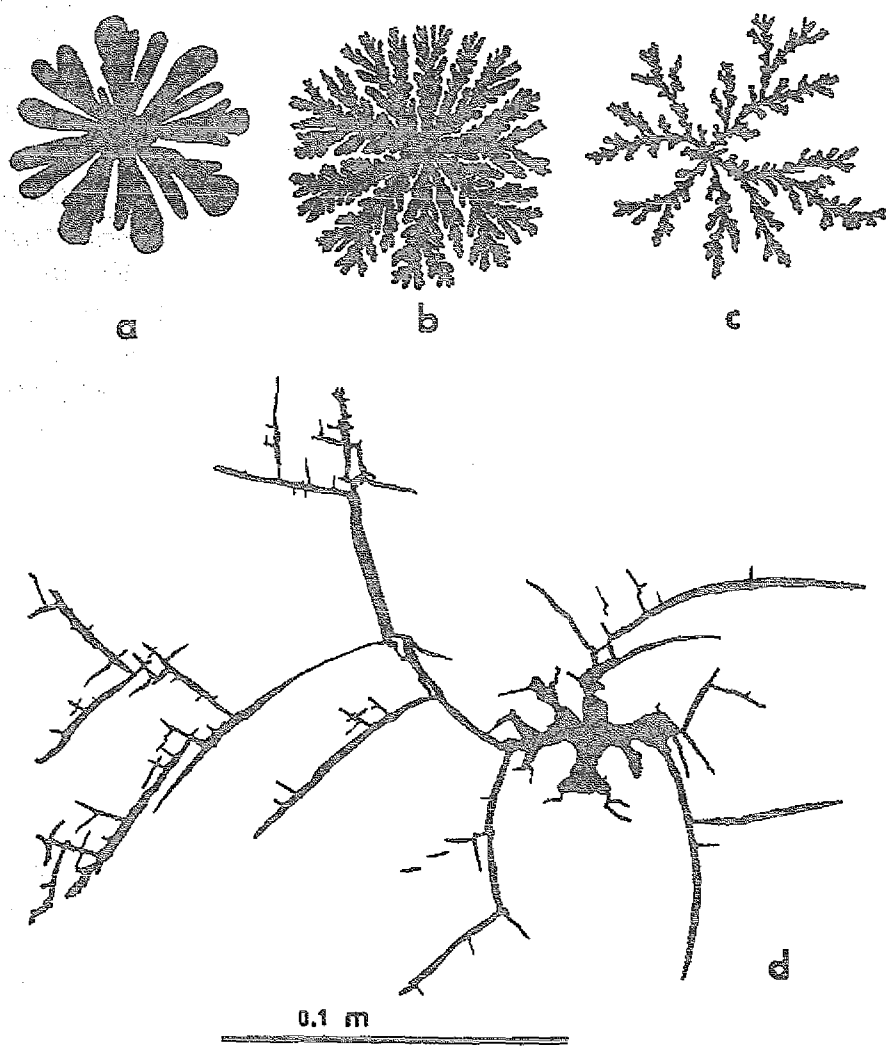


Abbildung 4 Strukturen entstanden in einer radialen Hele-Shaw Zelle durch: a) bzw. b) Verdrängung von Glyzerin durch Luft unter hohem bzw. niedrigem Druck; c) bzw. d) Verdrängung von Ton durch Wasser bei niedriger Tonkonzentration und niedrigem Wasserdruck bzw. bei hoher Konzentration und hohem Druck (entnommen aus [66]). Man beachte den Übergang vom Viscous-Fingering über eine dichte Zweigstruktur und eine offene Zweigstruktur hin zum Rißwachstum.

ein schädigendes Verhalten haben und durch Korrosion Spannungen hervorrufen können. Als weiteres wichtiges Beispiel sei hier auch noch die übermäßige Belastung von Bauteilen durch Vibrationen angefügt, die besonders im Flugzeugbau an Tragflächen extreme Konsequenzen haben kann.

Alle bisher ausgesuchten Beispiele stellten Probleme auf einer makroskopischen Längenskala dar. Das Problem des Rißwachstums ist jedoch auch auf wesentlich kleineren Längenskalen interessant. Im Bereich der Materialwissenschaften werden z. B. in mindestens drei verschiedenen Bereichen Rißphänomene untersucht. Das Rißverhalten von Polymeren und Kunststoffen ist einer der Bereiche, neben dem noch das Bruchverhalten von Metallen und Gesteinen untersucht wird. Unter "Gesteinen" sollen hier verschiedenste Materialien wie Glas, Keramik, Beton oder Ton zusammengefaßt sein, die selbst wieder die unterschiedlichsten Verhalten aufweisen.

Besonders interessant im Rißwachstum ist die scheinbar enge Verwandschaft zum Laplaceschem Wachstum, welche deutlich in den Experimenten von Van Damme et al. hervortritt und die deswegen besonders diskutiert werden sollen [66–69]. In ihnen wird in einer Hele-Shaw Zelle eine verdünnte Tonmischung variabler Konzentration oder Glyzerin von Wasser oder Luft verdrängt. In Abhängigkeit von der Konzentration des Tons bzw. dem Wasser- und Luftdruck kann man einen wichtigen morphologischen Übergang der Strukturen erkennen (Abb. 4). Für niedrige Konzentrationen bzw. Drucke findet man die üblichen Morphologien in Hele-Shaw Zellen: Viscous-Fingering (Abb. 4a), dichtes Zweigwachstum (Abb. 4b) und offenes Zweigwachstum (Abb. 4c), wie es auch für DLA typisch ist (vgl. Abb. 2). Die vierte und interessanteste Morphologie, nämlich eine typische Rißstruktur (Abb. 4d), erhält man, wenn man Wasser unter hohem Druck in eine Paste hoher Tonkonzentration preßt. Man beachte, daß sich in dieser Morphologie wesentliche Eigenschaften von Rißstrukturen herausgebildet haben, die das Rißwachstum deutlich vom Laplaceschen Wachstum unterscheiden [70]:

1. Die Form der Zweigspitzen ist verändert. Eine Reihe von Experimenten zeigt, daß sich mit zunehmendem Wasserdruck bzw. zunehmender Tonkonzentration die Spitzen von einer runden, konvexen Form kontinuierlich in eine für Risse typische scharfe, konkave Form entwickeln.
2. Der mittlere Winkel zwischen Zweigen folgt einer weiteren kontinuierlichen Entwicklung. Er geht kontinuierlich von einer fast parallelen Wachstumsrichtung (Abb. 4a) in einen fast rechten Winkel zwischen Zweigen über.

Quasi-rechte Winkel und konkave Spitzen in konzentrierten Pasten werden von Van Damme et al. [70] als Hinweise auf das Einsetzen der Rißbildung gewertet. Sie schlagen vor, daß ab einer bestimmten Wachstumsgeschwindigkeit des Risses sich Spannungen *hinter* der Rißspitze aufbauen können, die parallel zur Rißoberfläche gerichtet sind. Ab einem bestimmten Punkt kann sich diese Spannung entladen, indem ein senkrechter Seitenriß entsteht. Der dabei entstehende rechte Winkel ist eine Konsequenz der Tatsache, daß nur Spannungen parallel zur Rißoberfläche entstehen können (vgl. auch Gl. (18)).

Diese Vermutungen wurden in weiteren Experimenten bestätigt [68], in denen noch höhere Tonkonzentrationen studiert wurden (vgl. auch Abb. 5). Die hier gezeigte Sequenz des Rißwachstums zeigt deutlich, daß Bifurkationen tatsächlich hinter der Rißspitze stattfinden können. Die Risse sind im allgemeinen gekrümmt und Seitenzweige wachsen systematisch nur aus der konvexen Seite unter einem rechten Winkel heraus.

Ähnliche Resultate finden Zhao und Maher [71] in neuen Hele-Shaw-Zellen Experimenten. Sie studieren die Strukturbildung in Polymerlösungen, in die sie Wasser mit verschiedenen Injektionsraten hineinpressen. In Abhängigkeit von der Polymerkonzentration und der Injektionsrate finden sie einen Übergang der Struktur-Morphologien von einem Viscous-Fingering Verhalten hin zu Rißwachstum. Verschiedene typische Rißformen sind in der Abbildung 6 gezeigt. Im Gegensatz zu den Experimenten von Van Damme et al. findet man hier keinen kontinuierlichen, sondern einen abrupten Übergang zwischen den Morphologien. Allerdings möchte ich auch hier wieder auf die strukturellen Unterschiede zwischen den Rißstrukturen (Abb. 6b-6d) und der Viscous-Fingering Struktur (Abb. 6a) hinweisen. Es sind dies die gleichen Unterschiede, die oben schon beschrieben wurden: Scharfe Rißspitzen statt runder Finger und senkrechte Seitenzweige.

Alle diese Beispiele machen deutlich, wie vielschichtig das untersuchte Problem ist und daß es eine Vielzahl der unterschiedlichsten Bruchphänomene gibt. Ich möchte jedoch noch einmal

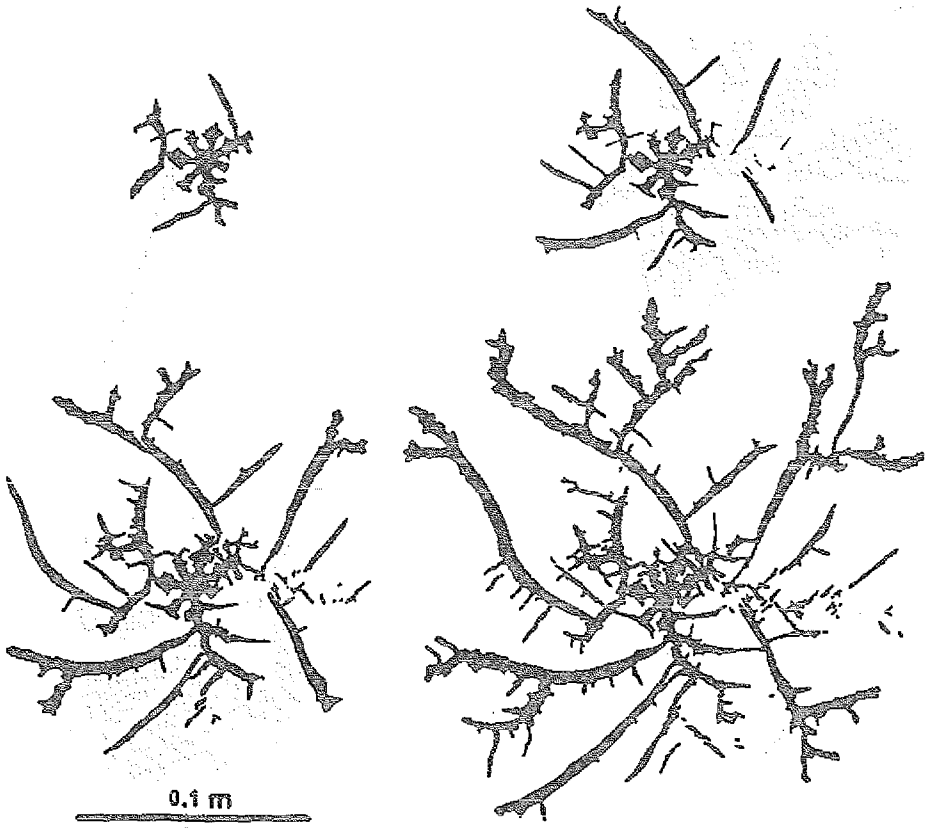


Abbildung 5 Zeitlich aufeinanderfolgende Stadien des Rißwachstums. Man beachte die ausgeprägten 90° Winkel von neu entstehenden Seitenzweigen und das Wachstum von Rissen hinter den Hauptspitzen.

besonders auf die in den letzten beiden Experimenten diskutierten Eigenschaften von Rissen hinweisen! Sie sind für diese Arbeit besonders wichtig und stellen wesentliche Eigenschaften physikalischer Risse dar. Wir werden später in der Lage sein, diese Eigenschaften mit einem einfachen Modell zu reproduzieren.

Es wäre vermessen, alle diese geschilderten Probleme lösen zu wollen. Im Gegenteil, es wäre bereits in großer Fortschritt im Bereich der Rißforschung, Modelle zur Verfügung zu haben, mit denen man wenigstens einen Teil der genannten Phänomene korrekt beschreiben kann. Hier möchten wir uns auf ein einfaches Modell eines linear-elastischen Körpers beschränken, der ein sprödes Bruchverhalten aufweisen soll, d. h. er soll sich linear elastisch verhalten bis die Materialfestigkeit erreicht ist und dann sofort brechen. In diesem Fall ist es sinnvoll, ähnlich zum Laplaceschen Wachstum ein Moving-Boundary Problem für das Rißwachstum zu formulieren.

Im folgenden werden einige Grundlagen der linearen Elastizitätstheorie wiederholt, die man auch in jedem Lehrbuch [72] wiederfinden kann.

Die Deformation des Körpers durch anliegende Kräfte wird durch das Verschiebungsfeld $\vec{u}(\vec{r})$ beschrieben, welches die Auslenkung eines Volumenelements innerhalb dieses Körpers von seiner Gleichgewichtslage — d. h. in Abwesenheit äußerer Kräfte — bezeichnet, und durch die Lamé-Gleichung bestimmt wird

$$(1-\nu(d-1))\nabla^2\vec{u}+\nabla(\nabla\cdot\vec{u})=0 \tag{13}$$

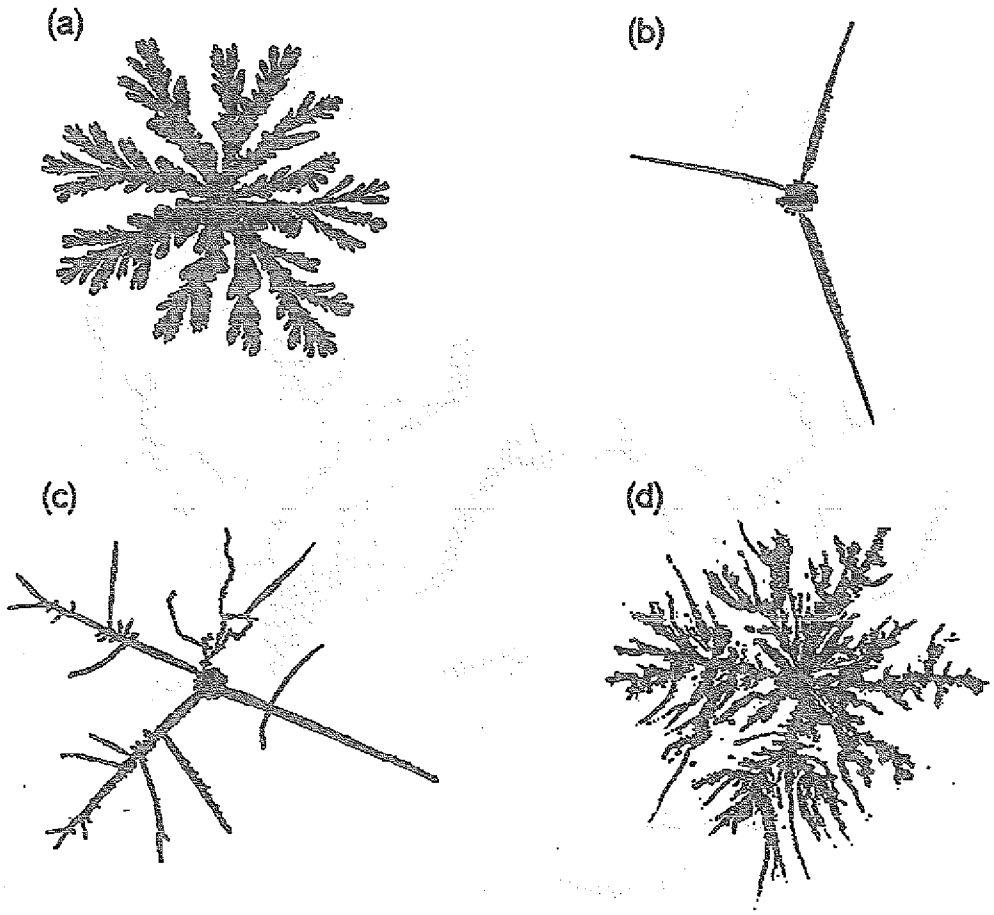


Abbildung 6 Rißmuster erzeugt in einem Hele-Shaw-Zellen Experiment. Man erkennt einen Übergang vom Viscous-Fingering Verhalten (a) zum Rißwachstum (b)-(d) mit zunehmender Wasser-Injektionsrate.

Hierbei ist d die Raumdimension und ν das Poisson-Verhältnis.

Das Poisson-Verhältnis hat physikalisch folgenden Einfluß. Ein Quader der Länge L und Querschnitt $W \times W$ sei einem einachsigen Zug ausgesetzt ("Young-Experiment"). Unter dieser Last wird der Quader nicht nur seine Länge in Richtung der Kraft um ΔL ändern, sondern sich auch senkrecht zur Kraftrichtung um ΔW kontrahieren. Die Poisson-Zahl beschreibt nun das relative Verhältnis

$$\nu = -\frac{\Delta W/W}{\Delta L/L} \tag{14}$$

Aus thermodynamischen Gründen ist das Poisson-Verhältnis beschränkt auf $-1 < \nu < 0.5$. Beton hat typischerweise ein Verhältnis $\nu = 0.2$. Allerdings kann man auch Materialien, wie z. B. granulare Medien, finden, die ein negatives Poisson-Verhältnis aufweisen, d. h. unter Zug in der transversalen Richtung expandieren.

Die Lamé-Gleichung selbst ist ein Ausdruck des Kräftegleichgewichts zwischen benachbarten Volumenelementen in einem elastischen Körper ausgedrückt mit Hilfe des Verschiebungsfeldes. Sie stellt daher die Bewegungsgleichung eines Volumenelements dar. Zur Herleitung muß man allerdings lokale Rotationen im Material vernachlässigen. Möchte man zusätzlich diese Rotationen berücksichtigen, muß man zur Cosserat-Elastizität [73] übergehen, in der neben lokalen Kräften auch lokale Drehmomente berücksichtigt werden.

Aus diesem Verschiebungsfeld kann — unter Vernachlässigung lokaler Rotationen — das tensorielle Verzerrungsfeld $\underline{\underline{\varepsilon}}$ als Ortsableitung der Verschiebungen bestimmt werden

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{15}$$

Thermodynamische Überlegungen, in welchen man die Arbeit betrachtet, die man bei einer bestimmten Materialverzerrung gegen die anliegende Spannung zu leisten hat, führen zu dem zur Verzerrung konjugierten Feld, dem Spannungsfeld $\underline{\underline{\sigma}}$. Das Hookesche Gesetz sagt nun aus, daß das Spannungs- und Verzerrungsfeld linear miteinander verknüpft sind

$$\sigma_{ij} = D_{ijkl}\varepsilon_{kl} \tag{16}$$

Physikalisch beschreibt das Spannungsfeld, welche Kraft $\vec{F}$ auf ein Oberflächenelement $\hat{n} \cdot dS$ innerhalb des Mediums wirkt

$$\vec{F} = \underline{\underline{\sigma}} \cdot \hat{n} \cdot dS \tag{17}$$

Auf den äußeren Rändern ist die Randbedingung durch die vorgegebenen Verschiebungen oder angelegten Kräfte festgelegt. Auf inneren Randflächen, d. h. Rißoberflächen, müssen alle Spannungen $\sigma_\perp$ senkrecht zur Rißoberfläche verschwinden, da am Riß selbst keine Rückstellkräfte wirken

$$\sigma_\perp = 0 \tag{18}$$

Die Rißoberfläche kann in dieser Vorstellung auf natürliche Weise als Oberfläche mit einer speziellen Randbedingung integriert werden. Die Dynamik der Rißpropagation kann beschrieben werden durch eine Geschwindigkeit v_n senkrecht zur Rißoberfläche. Diese Geschwindigkeit hängt natürlich von den mikroskopischen Bruchvorgängen ab, für die kein expliziter Mechanismus bekannt ist. In unserer Ableitung ist die einzige nicht-verschwindende Spannungskomponente $\sigma_\parallel$ parallel zur Rißoberfläche gerichtet. D. h. die Geschwindigkeit v_n wird einerseits von $\sigma_\parallel$ und andererseits von der Materialfestigkeit, d. h. letztendlich von den intermolekularen Kohäsionskräften σ_c abhängen. In dieser Formulierung berücksichtigt man offensichtlich Materialbruch durch Spaltung. Möchte man zusätzlich noch Materialbruch durch Biegen beschreiben, wird man die Cosserat-Elastizität benutzen. Da kein fundamentales Gesetz für v_n bekannt ist, nimmt man ad hoc ein allgemeines Verhalten der Gestalt

$$v_n \propto \left(\sigma_\parallel - \sigma_C\right)^\eta \tag{19}$$

an. Hier ist η ein phänomenologischer Parameter, der oft der Einfachheit halber als Eins angenommen wird. Diese Gleichungen beschreiben das Wachstum von Rissen als Moving boundary Problem.

Man sieht, daß die Propagation einer Rißoberfläche mit Hilfe einer sehr speziellen Regel "von Hand" in dieses Modell eingeführt wurde, um das tatsächliche Verhalten eines Risses zu beschreiben. Besonders im Ingenieurbereich sind dagegen eine Vielfalt von verschiedenen Bruchkriterien bekannt [74], die benutzt werden, um verschiedenste Materialverhalten möglichst realistisch zu beschreiben. Um dagegen Einsicht in das kollektive Phänomen des Rißwachstums zu erhalten, scheint es sinnvoll, möglichst einfache Bruchregeln zu studieren.

Das oben beschriebene Moving-Boundary Problem ist kaum analytisch zu lösen, so daß numerische Lösungsmethoden unerläßlich sind. Besonders im Ingenieurbereich sind kommerzielle

Finite-Elemente-Programme weit verbreitet [75], die allerdings im wesentlichen als einfache Black-Box benutzt werden und kaum Einsicht in die physikalischen Vorgänge des Rißwachstums gestatten. Hier sollen daher statt dessen die Gittermodelle beschrieben werden, die meist in der physikalischen Erforschung von Rißphänomenen Anwendung finden.

In diesen Modellen wird das untersuchte Medium mit Hilfe eines Gitters an einer endlichen Anzahl von Plätzen diskretisiert. An jedem Gitterplatz wird das Medium durch Angabe des Displacement-Vektors $\vec{u}$ und evtl. einer lokalen Rotation ϕ beschrieben. Auf dieser endlichen Menge von Gitterplätzen werden dann mögliche Diskretisierungen der elastischen Gleichungen numerisch gelöst.

Im wesentlichen werden drei Diskretisierungen betrachtet: das Central-Force Modell [76], das Bond-Bending Modell [77] und das Beam-Modell [78].

In dem Central-Force–Modell werden die einzelnen Gitterplätze durch einfache Hookesche Federn, die frei rotieren können, verknüpft. Das Bond-Bending Modell erhält man aus diesem Modell dadurch, daß man das freie Rotieren der elastischen Federn durch eine weitere Feder verhindert. Hier muß man beachten, daß man kein Quadrat-Gitter in einem Central-Force-Modell benutzen darf, da dieses Gitter kein Schermodul besitzt.

In dem Beam-Modell werden die frei rotierenden Hookeschen Federn durch elastische Stäbe ersetzt, die nicht nur Zug- und Druckkräfte, sondern auch Scherkräfte und Drehmomente zwischen den Gitterplätzen übertragen können. Da hier lokale Rotationen berücksichtigt werden, kann man die mit diesem Modell die allgemeinere Cosserat-Elastizität simulieren.

Das Bruchwachstum wird in diesen Modellen dadurch implementiert, daß die elastischen Konstanten von Federn oder Stäben, die über die Materialfestigkeit hinaus belastet werden, zu Null gesetzt werden und somit keine Last mehr tragen können. Verschiedene Varianten dieser Regeln wurden untersucht, die sich alle dadurch unterscheiden, welche Bonds zum Brechen ausgewählt werden. Eine Möglichkeit ist es, den jeweils am stärksten belasteten Bond auszuwählen und zu brechen, wodurch man ein deterministisches Bruchwachstum erhält [79]. Eine andere Möglichkeit besteht darin, die in jedem belasteten Bond gespeicherte elastische Energie E_i zu bestimmen und die Bonds dann alle mit einer Wahrscheinlichkeit p_i

$$p_i \propto \exp(-E_i/k_bT)$$

zu brechen [80]. Dieser Bruchmechanismus gilt in solchen Materialien, in denen Rißwachstum durch thermisch induzierte Unordnung, statt durch räumliche Unordnung, ausgelöst wird. Louis und Guinea [81] betrachten eine Variation, in der in einem Central-Force Modell die Streckung δ_i eines Bonds als Wahrscheinlichkeit p_i interpretiert wird, daß dieser gebrochen wird

$$p_i \propto \frac{\delta_i}{\sum_j \delta_j}.$$

Realistischen Materialien weisen eine räumliche Unordnung auf, die alle am Rißwachstum beteiligten Größen wie Kohäsionskräfte und elastische Moduln betrifft. Diese Unordnung wird in numerischen Simulationen als "quenched disorder" der Modellparameter berücksichtigt, d. h. die elastischen Konstanten und Festigkeiten der einzelnen Gitterbonds werden zu Beginn der Simulation anhand einer vorgegebenen Verteilung zufällig bestimmt. Demgegenüber steht die "annealed disorder" in solchen Modellen, in denen die Wahrscheinlichkeit, ob ein Bond gebrochen werden soll, in jedem Schritt neu berechnet wird.

Diese Modelle werden auf einem Computer auf folgende Weise implementiert: zunächst definiert man ein reguläres zweidimensionales Gitter, welches typischerweise von der Größe

100×100 ist. Dann legt man eine vorbestimmte äußere Last (z. B. uniaxialen Zug) an und equilibriert das System ins Gleichgewicht. Danach bricht man einen Bond entsprechend dem gewählten Bruchkriterium und relaxiert das System wieder ins Gleichgewicht. Dieser Vorgang wird wiederholt, bis im wesentlichen die zur Verfügung stehende Rechenzeit verbraucht und ein Riß mit typischerweise $10^2 - 10^3$ gebrochenen Bonds generiert ist.

Ein wesentliches Ergebnisse der neueren Rißforschung ist die Fraktalität der beobachteten Risse. Obwohl sie numerisch [82,83,79] durch Messung des Radius of Gyration der generierten Cluster und experimentell [84] z. B. durch Messung der Oberflächenrauhigkeit eines Risses bestätigt wurde, ist der Ursprung dieses fraktalen Verhaltens und deren Konsequenzen noch nicht geklärt. Die beobachteten fraktalen Dimensionen variieren sehr stark und reichen von $D_f \approx 1$ in einem Central-Force-Modell unter einachsigem Zug [83] bis hin zu $D_f \approx 1.6$ für ein Central-Force-Modell unter isotropem Zug [81].

Kapitel 3

Multiscaling und Simulationsmethode in DLA+

Ein verbesserter Algorithmus wurde benutzt, Off-Lattice DLA in zwei Dimensionen zu simulieren. Er gestattet uns, Cluster mit 50×10^6 Teilchen zu generieren, welches die bisher größten simulierten DLA Cluster sind. Das Dichteprofile in zwei Dimensionen wird auf die Multiscaling-Form $g(r, R_G) = r^{-d+D(x)} \cdot C(x)$ hin untersucht, wobei $x = r/R_G$ bedeutet, r der Abstand vom Clusterzentrum und R_G der Radius of Gyration des Clusters ist. Wir untersuchen 100 Cluster mit jeweils 1.000.000 Teilchen und können frühere Untersuchungen an wesentlich kleineren Clustern bestätigen. Außerdem finden wir, daß Multiscaling sehr empfindlich auf Anisotropien des Clusters reagiert und durch diese vollkommen zerstört wird.

3.1 Einleitung

Wachstumsphänomene werden zur Zeit wegen ihrer Relevanz für viele verschiedene Felder untersucht. "Diffusion Limited Aggregation" (DLA) [9] ist ein besonders interessantes Modell, welches Phänomene wie Dielectric-Breakdown, Viscous-Fingering oder Elektrodeposition beschreiben kann. Einige dieser Anwendungsgebiete werden z. B. in [27,26] zusammengefaßt.

Eine sehr interessante Frage zu DLA ist im Moment die interne Struktur der entstehenden Cluster. Es ist wohlbekannt, daß Off-Lattice DLA Cluster eine fraktale Struktur mit einer Dimension $D = 1.715 \pm 0.004$ [42] haben, welche durch die Abhängigkeit der Clustermasse M vom Radius of Gyration R_g definiert ist

$$M \propto R_g^D \tag{20}$$

Vor kurzem wurde vorgeschlagen, daß DLA Cluster eine reichere Struktur besitzen. Es wurde vermutet, daß die Teilchendichte in verschiedenen konzentrischen Schalen eines DLA Clusters mit unterschiedlichen Exponenten skalieren, welche durch eine Multiscaling-Dimension $D(x)$ beschrieben werden und die kontinuierlich vom relativen Abstand $x = r/R_g$ vom Clusterursprung abhängen [86]. Multiscaling hat in diesem Zusammenhang nichts mit Multifraktalität

+ Veröffentlicht in Physica A **176**, 454 (1991) [43], erscheint in Physica A (1993) [85]

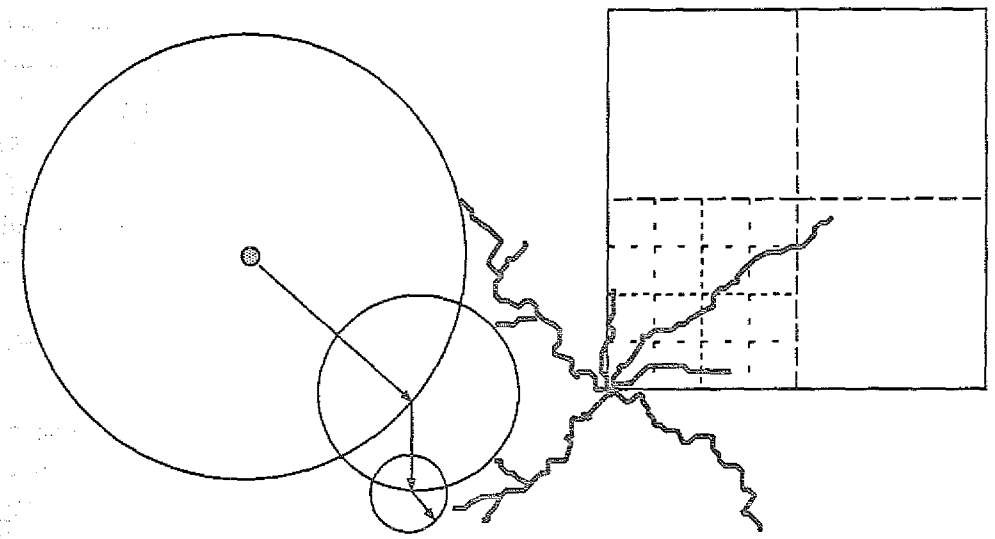


Abbildung 7 DLA mit variabler Schrittlänge. Die verschiedenen Gitter deuten die unterschiedlichen Diskretisierungsstufen an.

von DLA Clustern zu tun. In der vorliegenden Arbeit wollen wir das Verhalten von $D(x)$ für sehr große DLA Cluster mit 10^6 Teilchen studieren.

Im Abschnitt 3.2 präsentieren wir eine detaillierte Beschreibung des Algorithmus, mit dem wir große Cluster von 10^6 Teilchen leicht auf Workstations generieren können. Im Abschnitt 3.3 präsentieren wir einige Gedanken zur statistischen Selbstähnlichkeit von großen DLA Clustern und im Abschnitt 3.4 präsentieren wir unsere Resultate der Multiscaling-Dimensionen.

3.2 Modell und Methode

Der zugrundeliegende Prozeß zur Generierung von Off-Lattice DLA Clustern ist der folgende: Anfangs wird ein Seed-Teilchen an eine feste Stelle — z. B. der Ursprung des Koordinatensystems — plaziert. Danach wird im Unendlichen ein einzelnes Teilchen gestartet, welches einen Random-Walk ausführt indem es um einen Teilchenradius in eine zufällige Richtung springt, bis es auf das erste Teilchen trifft und dann an diesem Teilchen haftet. Dieser Prozeß wird wiederholt, bis ein Cluster der gewünschten Masse generiert ist. Dieser Prozeß ist jedoch nicht sehr effizient, da die Walker immer die gleiche Schrittlänge haben.

In allen neueren Algorithmen [87,42] werden die Walker nicht im Unendlichen, sondern auf einer Hyperkugel, die den bestehenden Cluster gerade umschließt, gestartet. Sie werden angehalten, wenn sie entweder den Cluster berühren, oder sie sich zu weit fort von ihm befinden. In beiden Fällen wird ein neuer Walker an einer zufälligen Stelle auf der einhüllenden Hyperkugel gestartet. Offensichtlich ist es ausreichend, daß diese Hyperkugel nur unwesentlich größer als der Cluster selbst ist.

Der wichtigste Punkt bei der effizienten Generierung ist es, die Schrittlänge während des Random-Walks dynamisch zu verändern [34,42]. In den freien Bereichen zwischen den Clusterarmen oder in Bereichen weit weg vom Cluster ist es nicht notwendig, eine feste, kleine Schrittlänge zu benutzen, sondern es ist vorteilhaft, große Schritte machen zu können, solange man darauf achtet, daß keine Zweige des Clusters übersprungen werden (vgl. Abb. 7). Daher ist es wichtig, die größten freien Bereiche um die jeweilige Position des Walkers herum zu ermitteln, um eine gute Abschätzung für die größtmöglich Schrittlänge zu erhalten.

Um dies zu erreichen wird der Cluster im wesentlichen auf Gittern mit unterschiedlicher Längenskala diskretisiert (vgl. Abb. 7). Wenn der Walker einen Sprung ausführen möchte, betrachtet man zunächst die gröbste Diskretisierung. Wenn man hier findet, daß sich der Walker auf dieser Längenskala in einem freien Raumbereich befindet, wird der Sprung ausgeführt, ansonsten wir die nächst feinere Diskretisierung untersucht. Dieser Prozeß wird wiederholt, bis entweder eine Diskretisierungsstufe erreicht wird, auf der ein Sprung möglich ist, oder eine untere Cutoff-Längenskala erreicht ist. Auf dieser untersten Längenskala enthält jeder Diskretisierungspunkt einen Pointer zu einer verketteten Liste der exakten Koordinaten aller Teilchen, die während des nächsten Sprunges berührt werden könnten (vgl. als Beispiel die Abb. 8 und besonders die Bildunterschrift).

Nach dieser groben Beschreibung der Methode werden wir nun einige Details der tatsächlichen Implementierung beschreiben. Um die Position des Walkers festzustellen, beginnen wir mit einer groben Diskretisierung, die aus einem N_r^d dimensionalen Gitter von Pointern besteht, wobei N_r die Anzahl der Gitterpunkt pro Raumdimension ist. Jedes Gitterelement repräsentiert einen bestimmten Raumbereich der Größe L_0^d. Wenn dieser Pointer einen besonderen Wert `NIL` beinhaltet, bedeutet dies, daß sich kein weiteres Teilchen in oder nahe diesem Raumbereich befindet. Ansonsten zeigt dieser Pointer auf eine Struktur von weiteren 2^d Pointern, von denen jeder einen bestimmten Unterbereich der Größe $(L_0/2)^d$ darstellt. Auf dieser Ebene bedeutete wieder jeder Pointer, der `NIL` enthält, einen leeren Raumbereich, während ein anderer Eintrag auf die nächste, feinere Ebene zeigt. Diese Strukturierung wird bis auf die unterste Cutoff-Ebene fortgesetzt. Auf dieser feinsten Ebene enthält jedes Gitterelement einen Zeiger auf eine verkettete Liste von Koordinaten. Für die Menge von Teilchen, die in jeder Liste enthalten sein sollen, wurden zwei Möglichkeiten vorgeschlagen [42]:

1. Alle Teilchen, deren Mittelpunkt in dem jeweiligen Raumbereich liegen.
2. Alle Teilchen, die von einem Walker berührt werden könnten, dessen Mittelpunkt in dem betreffenden Raumbereich liegt.

Für Simulationen von großen Clustern ($\geq 5 \cdot 10^5$ Teilchen) ist es notwendig, die erste Möglichkeit zu implementieren, um den gesamten Speicherbedarf, der das Hauptproblem bei der Simulation ist, zu reduzieren. Hier muß betont werden, daß das Hauptproblem der DLA Simulation mit dieser Methode nicht die Rechenzeit, sondern der benötigte Speicherplatz ist. Zwar mag die oben vorgestellte Pointer-Struktur sehr aufwendig erscheinen, aber sie ist essentiell wichtig, den gesamten benutzten Speicher auf ein vernünftiges Maß zu reduzieren.

Ein weiterer Parameter — mit dem man das Verhältnis von Rechenzeitbedarf zu Speicherplatzbedarf bestimmen kann — ist das Verhältnis von Teilchendurchmesser d und Cutoff-Länge $l_{\min}$ der feinsten Diskretisierung. Ein kleines Verhältnis — z. B. $d = l_{\min}$ — führt zu schnellen Simulationen mit hohem Speicherbedarf, während ein großes Verhältnis — z. B. $d \simeq l_{\min}/10$ — zu langsamen Simulationen mit niedrigem Speicherbedarf führen.

Mit dieser Implementierung benötigt eine Simulation eines Clusters mit 10^6 Teilchen 14 MByte Hauptspeicherplatz und 9 Stunden Rechenzeit auf einer SPARCstation 1. Wir müssen betonen, daß alle beschriebenen Simulationen auf Workstations durchgeführt werden können und keinerlei Supercomputer benötigt werden. Für die Simulation wesentlich größerer Cluster mit 50×10^6 Teilchen muß man zusätzlich noch eine spezielle Speicherverwaltung benutzen, bei der man ausnutzt, daß nur ein geringer Teil des gesamten DLA Clusters aktiv am Wachstum beteiligt ist [88], da das Wachstum von DLA Clustern im wesentlichen am Rand stattfindet.

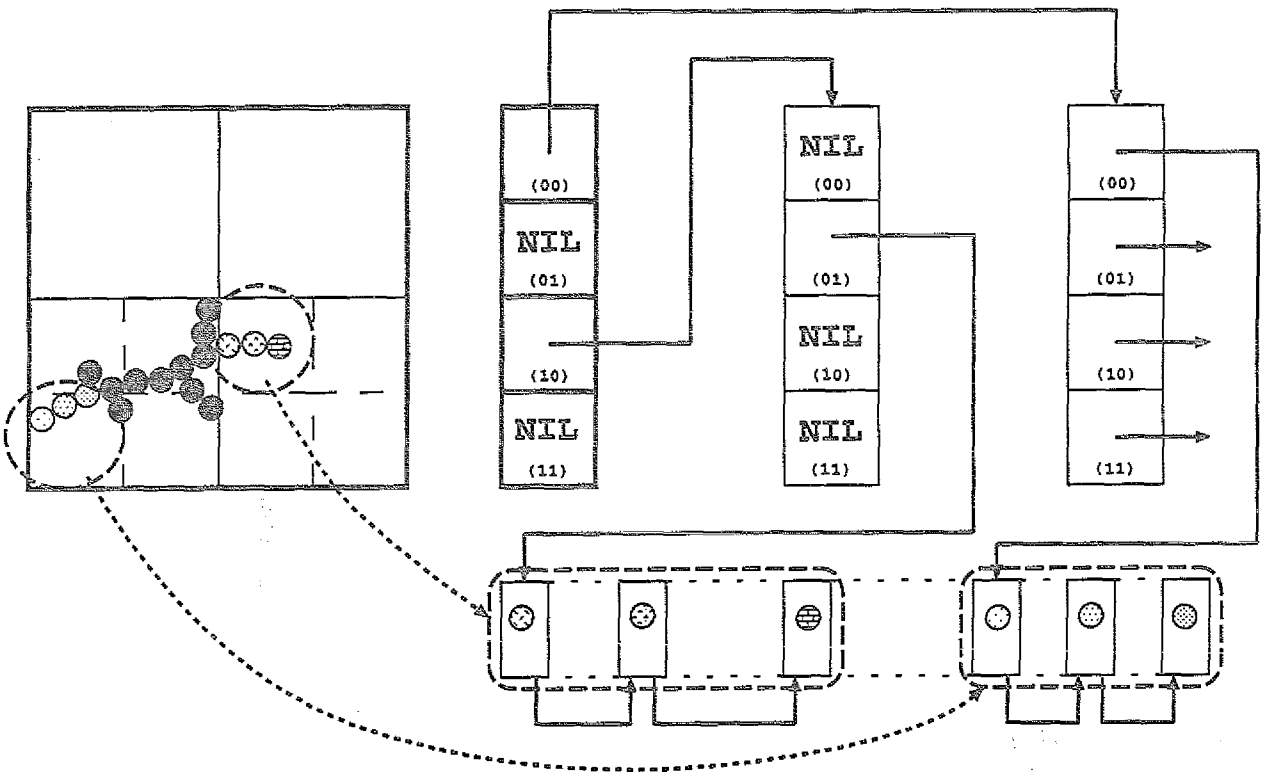


Abbildung 8 Schematische Darstellung der in DLA benutzten Datenstruktur. Ganz links sieht man einen Teil eines Zweiges mit insgesamt 17 Teilchen, der in einem gegebenen Raumelement (großes Quadrat) enthalten ist. Dieses große Quadrat enthält vier Unter-Bereiche: Die oberen beiden Unter-Quadrate — markiert als (01) und (11) — sind leer, während die unteren beiden — (00) und (10) mit Teilchen besetzt sind. Dies spiegelt sich in der Pointer-Struktur wieder, die diesen Raumbereich repräsentiert und die daneben gezeigt ist. Jede Struktur enthält vier Pointer — einen für jeden der vier Unter-Quadrate. Leere Raumbereiche werden durch einen besonderen Wert `NIL` dargestellt. Also enthalten die Einträge (01) und (11) der ersten Pointerstruktur gerade diesen speziellen Wert. Ist ein Unter-Quadrat besetzt, so zeigt der entsprechende Pointer — in diesem speziellen Beispiel (00) und (10) — auf diejenige Pointerstruktur, die diesen entsprechenden Unterbereich repräsentiert. Dieses hierarchische Ordnungsschema wird bis zu einer unteren Cutoff-Länge fortgesetzt. Auf der feinsten Ebene enthält ein Raumelement entweder kein Teilchen und der entsprechende Pointer enthält `NIL`, oder dieses Raumelement ist besetzt und dann enthält der Pointer einen Zeiger auf eine verkettete Liste der Teilchen, die sich in diesem Raumelement befinden. Diese verketteten Listen sind für die Raumbereiche, in denen jeweils drei Teilchen durch einen Kreis markiert sind, im unteren Teil der Figur angedeutet.

Zur Illustration betrachte man z. B. die Abbildung 1. Man sieht dort deutlich, daß DLA Cluster in "Schichten" wachsen und nur eine äußerer aktiver Bereich am Wachstum beteiligt ist. Es reicht daher, lediglich einen Bruchteil der gesamten Koordinaten im Speicher zu halten, was hier sehr wichtig ist, denn der Versuch, alle Teilchenkoordinaten gleichzeitig resident zu halten, würde alleine für die Koordinaten 400MB Speicher verschlingen. Wir wenden daher eine "least-recently-used" Strategie an, bei der nur die letzten benötigten Koordinaten im Speicher gehalten werden.

Dabei betrachten wir einen Stack von Teilchenkoordinaten, in dem insgesamt N_S Koordinaten gespeichert werden können. Sobald die Koordinaten eines bestimmten Teilchens benötigt

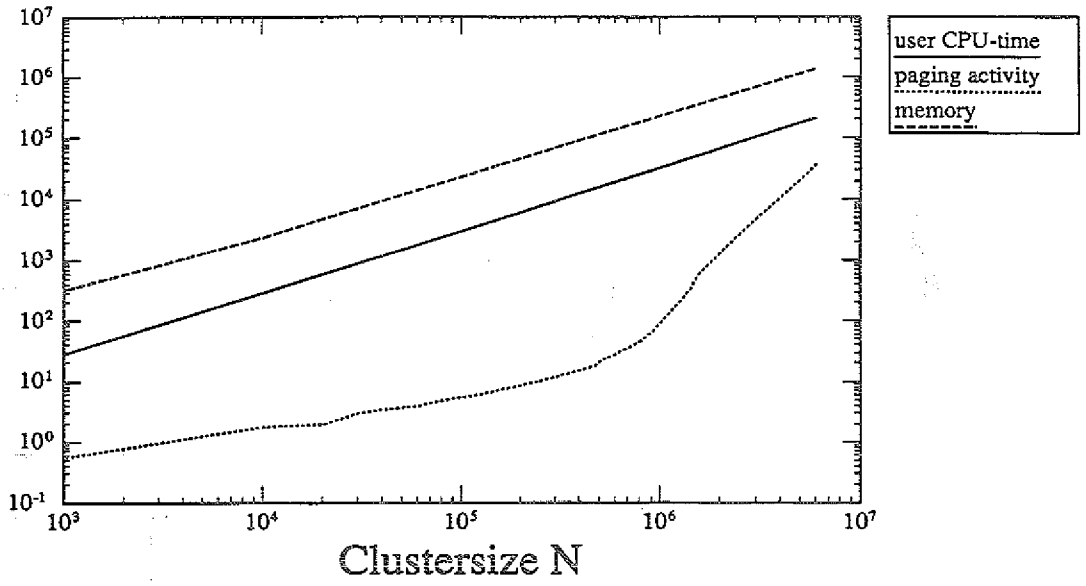


Abbildung 9 Größenabhängigkeit der Laufzeit, des Speicherbedarfs und der Paging Aktivität eines 6×10^6 Teilchen Off-Lattice DLA Clusters, simuliert auf einer SPARCstation 2 mit 8MB Hauptspeicher. Die Benutzer CPU-Zeit ist in Sekunden gemessen, der Hauptspeicher-Bedarf ist für die Speicherung der Diskretisierungs-Karten gemessen. Die Einheit ist hier 12 Byte. Die Paging-Aktivität ist die benutzte System-Zeit und wird in Sekunden gemessen.

werden, werden diese entweder von der Platte gelesen oder von ihrer momentanen Position im Stack entfernt und auf die Spitze des Stack gelegt. Da der Stack nur eine begrenzte Zahl von Koordinaten beinhalten kann, müssen alle anderen Koordinaten im Stack um eine Stelle nach unten rücken und die Koordinaten, die sich ganz am Ende des Stack befindet und somit am seltensten benutzt wurde ("least-recently-used") werden auf die Platte zurückgeschrieben. Zur Generierung eines 50 Millionen Teilchen Clusters werden so z. B. lediglich die Koordinaten von 7 Millionen Teilchen im Hauptspeicher gehalten. Hier ist der gesamte Speicherbedarf 120MB und die Laufzeit beträgt ca. vier Wochen auf einem SPARCserver 670MP.

Eine interessante Eigenschaft unserer Implementierung ist, daß sowohl die benötigte CPU Zeit, als auch der Speicherbedarf linear von der Clustermasse M abhängen. Dies wird in Bild 9 für ein Cluster mit 6×10^6 Teilchen demonstriert. Aus diesen Daten erhalten wir eine Potenz Abhängigkeit der Laufzeit mit einem Exponenten $\nu_T = 1.047 \pm 0.005$ und eine Abhängigkeit des Speicherbedarfs mit einem Exponenten $\nu_m = 0.994 \pm 0.005$ von der Clustermasse. Das Laufzeitverhalten ändert sich drastisch für sehr große Cluster, für die mehr Teilchen aktiv am Wachstum beteiligt sind, als effektiv im Hauptspeicher gehalten werden können. Hier setzt ein starkes System-Paging ein, durch welches sich die gesamte Laufzeit drastisch erhöht (vgl. auch Bild 9). Man erkennt hier deutlich ein Crossover Verhalten vom linearen zum superlinearen Verhalten. Dieses Problem tritt allerdings nur auf Maschinen mit stark begrenztem Hauptspeicher auf. Die Simulation des Clusters mit 50.000.000 konnte auf einer Maschine mit 64 MB Hauptspeicher ohne wesentliche Probleme durchgeführt werden.

Leider existieren bisher weder parallelisierte noch vektorisierte DLA Algorithmen. Alle bisher geschilderten Methoden eignen sich lediglich für skalare Rechner. Trotzdem haben wir unser Programm auf eine Vielzahl von Maschinen portiert, um die dort erzielbaren

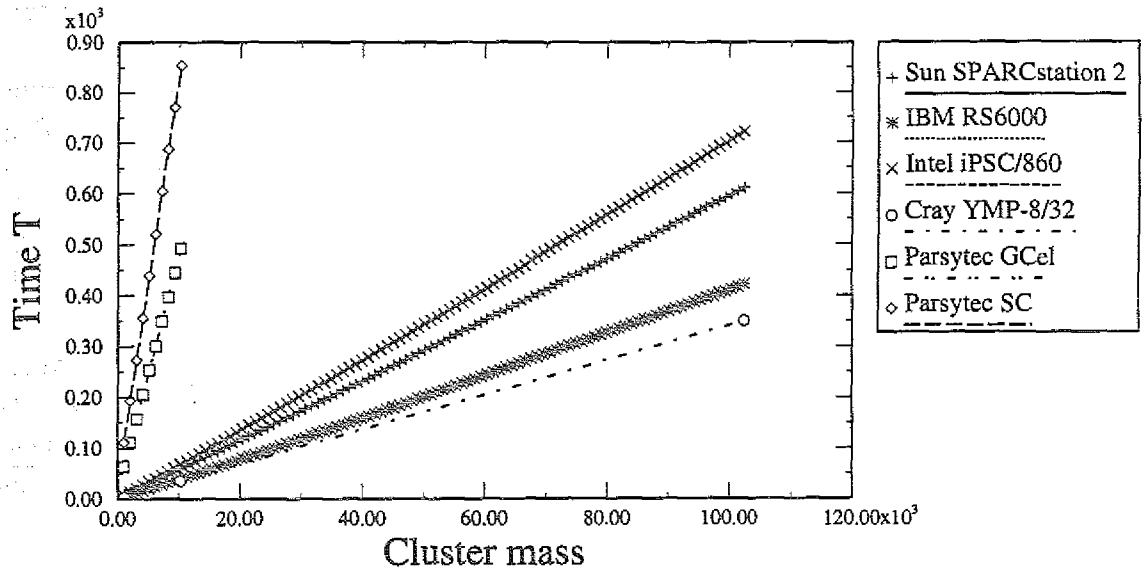


Abbildung 10 Laufzeitverhalten des DLA Algorithmus auf verschiedenen Maschinen.

Maschine	Laufzeit für 1024 Teilchen in Sekunden
Sun SPARCstation 2	5.9
IBM RS6000/330	4.0
CRAY-Y-MP8/832	3.6
Intel iPSC/860	6.9
Parsytec GCel	49.3
Parsytec SC	85.3

Tabelle 1 Laufzeiten für die Anlagerung von 1024 Teilchen.

Leistungen vergleichen zu können. In Figur 10 zeigen wir die Laufzeit in Abhängigkeit von der Clustermasse für einen kleineren Cluster mit 10^5 Teilchen und im folgenden werde ich die Ausführungszeiten mit den auf einer Sun SPARCstation 2 gemessenen vergleichen. In der Tabelle 1 sind die Zeiten, die man zur Anlagerung von 1024 Teilchen benötigt, zusammengefaßt.

Auf einer SPARCstation 2 beträgt die minimale Zeit, die man für die Anlagerung von 1024 Teilchen benötigt ca. 6 Sekunden, so daß ein Cluster mit 102.400 Teilchen leicht in ca. 10 Minuten generiert werden kann. Zusätzlich muß man bedenken, daß ca. 30 Sun-Workstations — nicht alle davon sind SPARCstation 2, sondern auch SPARCstation 1 — im HLRZ-Cluster verfügbar sind, so daß man sehr schnell, sehr viele Cluster generieren kann.

Die schnellsten Ausführungszeiten haben wir auf einer CRAY-Y-MP8/832 erzielt. Da jedoch nur unwesentliche Initialisierungsroutinen vektorisiert werden können, ist die Simulation hier nur weniger als doppelt so schnell, wie auf einer SPARC 2, was einzig auf die größere

Leistung der skalaren Prozessoren zurückzuführen ist[2]. Auch auf einer superskalaren IBM-RS6000/330 ist die Programmgeschwindigkeit lediglich verdoppelt, während man auf einem Prozessor einer Intel iPSC/860 MIMD Maschine sogar langsamere Programme als auf der SPARC 2 erhält. Beide Maschinen besitzen ihre größten Vorteile gegenüber einer Sun-Workstation bei der Floating-point Berechnung, die allerdings in diesem DLA-Programm nur eine untergeordnete Rolle spielt, während ein sehr großer Teil der Rechenzeit für die Verfolgung der einzelnen Pointer in der Karten-Struktur verbraucht wird. Für diese Aufgabe ist nicht nur die Rechenleistung des Prozessors, sondern auch die Geschwindigkeit des Hauptspeichers wesentlich, da die einzelnen Pointer nicht sequentiell, sondern an beinahe zufälligen Stellen im Speicher abgelegt sind.

Die langsamsten Ausführungszeiten wurden auf zwei Parsytec Multiprozessor-Maschinen, die beide auf einem T800–Transputer basieren, gemessen. Im Vergleich zum iPSC/860 steigt die Laufzeit um einen Faktor 7–15 an. Dieser Geschwindigkeitsverlust kann auch in parallelisierten und für MIMD Maschinen optimierten Molekular-Dynamik Simulationen festgestellt werden [89] und repräsentiert die relativ geringe Rechenleistung jedes einzelnen Prozessors. Allerdings sollen spätere Maschinen-Versionen auf einem T9000–Transputer basieren, der um einen Faktor 10 schneller als der Vorgängerprozessor sein soll, so daß hier mit einem iPSC/860 vergleichbare Resultate erzielbar sein sollten.

3.3 Statistische Selbstähnlichkeit

Es ist bekannt, daß DLA auf dem Gitter ein langsames Crossover Verhalten von einer isotropen Struktur in eine stark anisotrope, vom Gitter dominierte Struktur aufweist: Während kleine Cluster von wenigen tausend Teilchen aufgrund der starken Fluktuationen eine verzweigte Struktur aufweisen, ist die Struktur von Clustern mit mehreren Millionen Teilchen dominiert von der Topologie des zugrundeliegenden Gitters [32]. Große Cluster auf einem quadratischen Gitter zeigen z. B. eine vierzählige Symmetrie.

Obwohl nicht zu erwarten ist, daß auch Off-Lattice DLA Cluster ein solches Crossover Verhalten aufweisen, ist diese Frage immer noch nicht vollständig geklärt. Es gibt Anzeichen für ein Crossover in eine vollkommen selbst-ähnliche Struktur, als auch für ein komplexeres Verhalten. So gab es Vorschläge und numerische Hinweise, die auf ein gerichtetes Fraktal [58], auf geometrische Multifraktalität [90] oder Multiscaling [86] hindeuten. Es sollte daher interessant sein, ob man in Off-Lattice DLA ein Crossover in eine selbst-ähnliche Struktur erhält. Zu diesem Zweck wenden wir ein einfaches visuelles Kriterium an. Wir haben Cluster mit Massen in drei verschiedenen Größenordnungen — 500.000, 5.000.000 und 50.000.000 Teilchen — generiert und sie in derselben Auflösung gedruckt. Dabei haben wir nicht nur die Ausdehnung der einzelnen Cluster auf die gleiche Größe skaliert, sondern auch die Teilchendichte renormiert, so daß in allen Bildern dieselbe Anzahl von Punkten gesetzt ist. Dies ist notwendig, damit nicht durch eine unterschiedliche Bildschwärzung ein falscher Eindruck entsteht. Die Cluster sind in den Bildern 11-13 gezeigt.

Zunächst fällt besonders auf, daß die beiden Bilder der größten Cluster bis auf statistische Fluktuationen praktisch ununterscheidbar sind. Das bedeutet, daß geometrische Selbstähnlichkeit über eine gesamte Größenordnung vorhanden ist. Dies ist nicht uneingeschränkt gültig

[2] Selbstverständlich haben wir die CRAY nur für kurze Timing-Läufe benutzt und keinerlei Produktionsläufe durchgeführt.

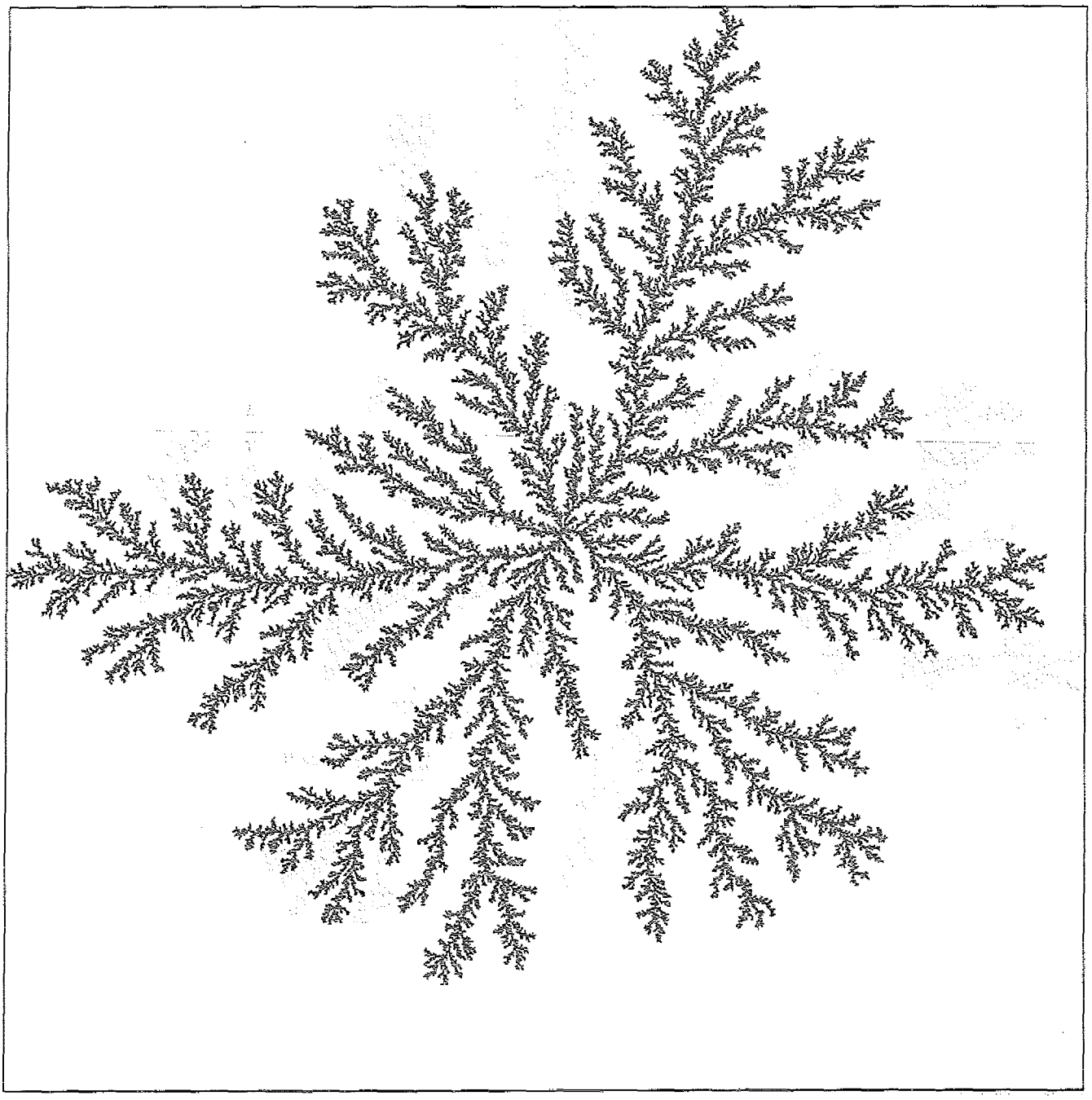


Abbildung 11 Off-Lattice DLA Cluster mit 500.000 Teilchen.

für den kleinsten Cluster mit 500.000 Teilchen. Zwar sind die strukturellen Unterschiede nur sehr gering, aber trotzdem scheinen die einzelnen Zweige mit zunehmender Clustermasse immer glatter und gerader zu werden.

Hieraus kann man drei Schlußfolgerungen ziehen. Zunächst scheinen off-Lattice DLA Cluster tatsächlich für unendliche Masse geometrisch selbst-ähnlich zu sein. Als zweites erfolgt offensichtlich kein Crossover wie in Lattice-DLA in eine anisotrope Struktur und als letztes muß man sehr große Cluster mit wenigstens mindestens 10^5 Teilchen generieren, um nicht mehr von Finite size Effekten beeinflußt zu werden. Daher ist high-quality-computing notwendig, um die Selbstähnlichkeit von DLA zu testen.

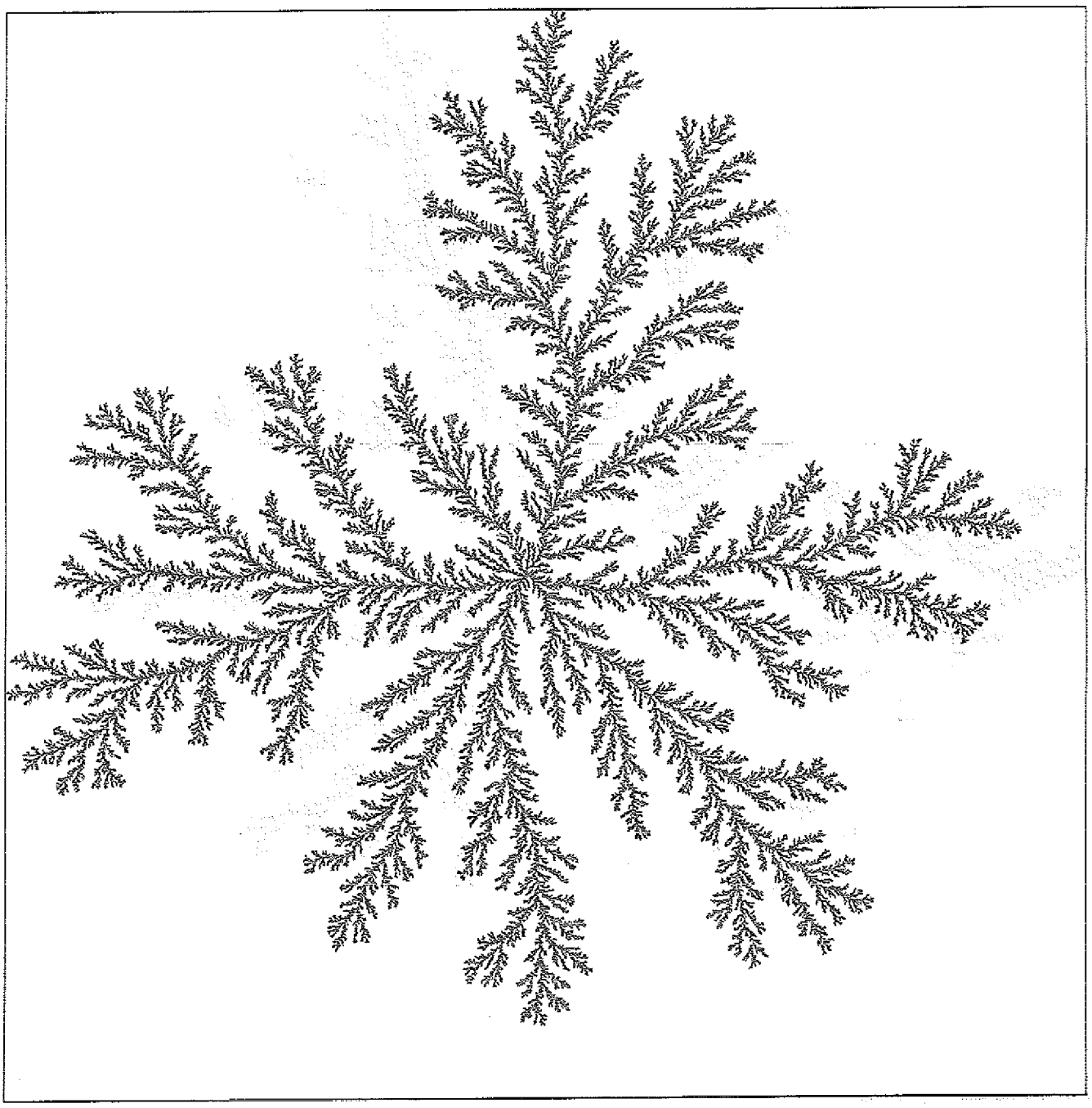


Abbildung 12 Off-Lattice DLA Cluster mit 5.000.000 Teilchen.

3.4 Multiscaling

Eine Eigenschaft von DLA Clustern mittlerer Größe scheint die Existenz von verschiedenen Skalen-Exponenten für unterschiedliche Größen zu sein. Bei einer Untersuchung der Breite der aktiven Zone in DLA fanden Plischke und Rácz [88] einen Exponenten, der vom üblichen Exponenten, der über den Radius of Gyration bestimmt wird, abweicht. Coniglio und Zannetti [86] schlugen vor, die fraktale Dimension in DLA in Abhängigkeit vom Abstand vom Clusterzentrum zu bestimmen.

Zu diesem Zweck messen wir die radiale Massendichte $g(r, R)$, welche definiert ist als

$$g(r, R)\, d^d r = dN \tag{21}$$

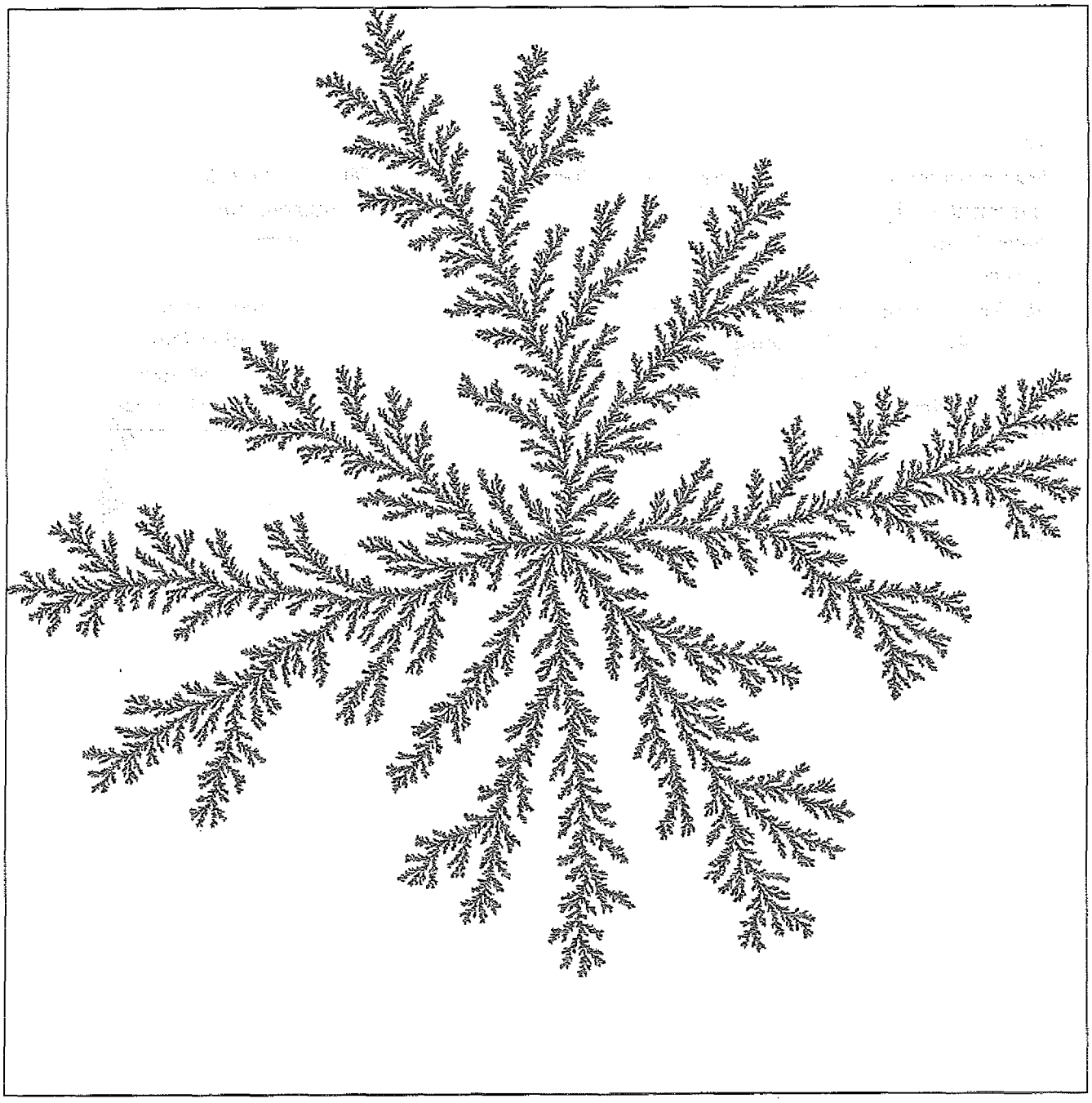


Abbildung 13 Off-Lattice DLA Cluster mit 50.000.000 Teilchen.

wobei dN die Anzahl von Teilchen in einem Volumenelement $d^d r$ im Abstand r vom Clusterzentrum ist. Die Masse des Clusters N ist implizit durch den Radius of Gyration $R = R(N)$ gegeben.

In Analogie mit kritischen Phänomenen nimmt man üblicherweise ein Skalenverhalten der Form

$$g(r, R) = r^{-d+D} f\left(\frac{r}{R}\right) \tag{22}$$

mit einer Skalenfunktion $f(x)$ und einer globalen fraktalen Dimension D an. Coniglio und Zannetti [86] schlugen vor, die globale fraktale Dimension durch eine strukturierte Funktion $D = D(r/R)$ zu ersetzen, womit die fraktale Dimension in verschiedenen Schalen bestimmt

wird. Sie schlugen daher ein Skalenverhalten

$$g(r,R) = r^{-d+D(x)} f(x), \quad x = \frac{r}{R} \tag{23}$$

vor.

Man erwartet, daß tief in der eingefrorenen Region des Clusters für $x \to 0$ die Multiscaling-Exponenten $D(0)$ mit der globalen fraktalen Dimension D übereinstimmen. Auf der anderen Seite ist es nicht möglich, a priori das Verhalten von $D(x)$ in den äußeren Randbereichen vorherzusagen.

Vor kurzem wurde dieses Multiscaling-Verhalten mit Clustern von 10^5 Teilchen von Amitrano et al. [91] numerisch untersucht. Es wurde dort gefunden, daß die Multiscaling-Dimension $D(x)$ nicht konstant ist, sondern kontinuierlich von x abhängt. Für kleine x stimmen die fraktale Dimension D und $D(x)$ wie erwartet überein. Mit zunehmendem x erhält man ein kleines Maximum bei $x \simeq 1.4$ und danach einen kontinuierlichen Abfall zwischen $x \simeq 1.7$ und $x \simeq 2.1$.

In unserer Untersuchung des Multiscaling haben wir 100 Off-Lattice DLA Cluster mit jeweils 10^6 Teilchen generiert. Das Wachstum wurde in 10 Stufen unterteilt, in denen jeweils der Radius of Gyration gemessen wurde. Von Stufe zu Stufe wurde die Masse der Cluster um einen Faktor 1.25 erhöht. In jeder Stufe wurden die Cluster in 30 Schalen unterteilt, in denen die Masse $M(r_n, R)$, die direkt mit der radialen Dichteverteilung

$$M(r_n, R) = 2\pi r_n g(r_n, R)\Delta r. \tag{24}$$

zusammenhängt, gemessen wurde. Ein Vergleich der Radii of Gyration von aufeinanderfolgenden Stufen liefert einen Skalenfaktor l, der durch

$$l = \frac{R_{n-1}}{R_n}. \tag{25}$$

gegeben ist. Durch skalieren des Radius und der Dicke der Schalen mit diesem Faktor erhält man die fraktale Dimension innerhalb dieser Schale als

$$\begin{aligned} \Delta r_{n+1} &= l \cdot \Delta r_n \\ r_{n+1} &= l \cdot r_n \\ D\left(\frac{r_{n+1}}{R_{n+1}}\right) &= d + \frac{\ln\left(g(r_{n+1}, R_{n+1})/g(r_n, R_n)\right)}{\ln(l)} \end{aligned} \tag{26}$$

Danach mitteln wir über die verschiedenen Wachstums-Stufen und die verschiedenen Realisationen. Die erhaltenen Resultate — für 100 Cluster mit je 10^6 Teilchen und für einen Cluster mit 50×10^6 Teilchen sind in Bild 14 zusammen mit den Daten von Amitrano et al. gezeigt. Für kleine x erhalten wir für $D(x)$ sowohl bei $M = 10^6$ als auch bei $M = 50 * 10^6$ einen Wert, der innerhalb der Fehler nahe der fraktalen Dimension D ist: $D(0) = 1.69 \pm 0.02$ und $D = 1.712 \pm 0.003$. Hierbei haben wir $D(0)$ erhalten, indem wir einen linearen least-squares-fit an die Werte von $D(x)$ für kleine x durchführen und nach $x = 0$ hin extrapolieren. Für steigende x bleibt die Multiscaling-Dimension ziemlich konstant. Amitrano et al. finden für mittlere Abstände $x \simeq 1.1$ einen kleinen Buckel in $D(x)$. Dieser verschwindet in unseren Messungen und ist daher auf Finite-size Effekte zurückzuführen. Für große x muß man die äußeren Clusterregionen messen, in denen die Dichte nur sehr ungenau zu bestimmen ist. Daraus resultieren große statistische Fluktuationen für $D(x)$. Trotzdem können wir den

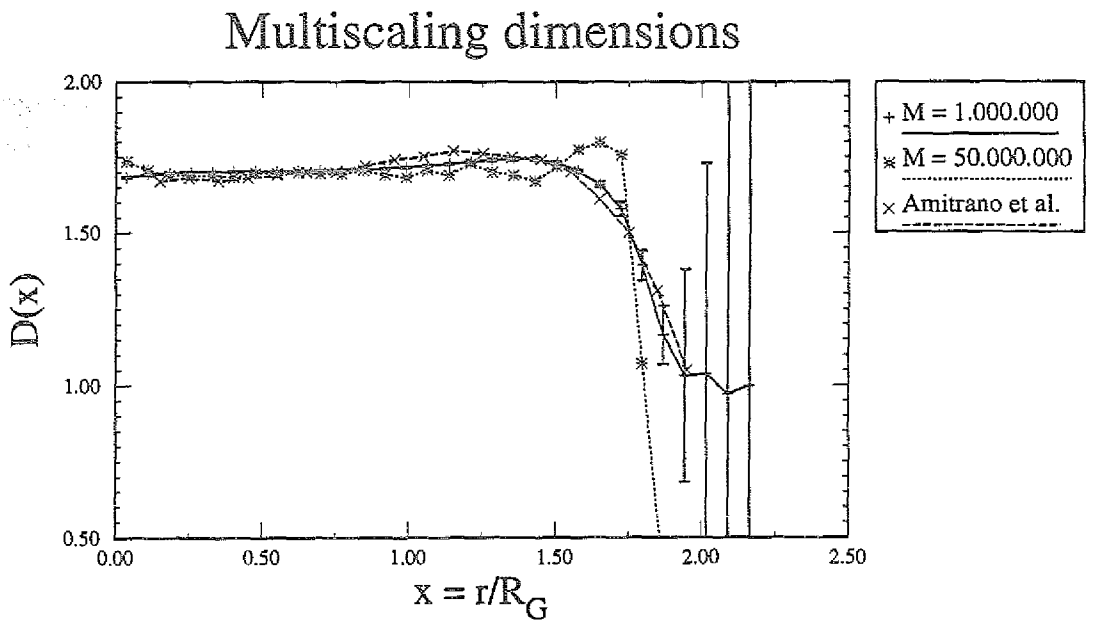


Abbildung 14 Plot der Multiscaling-Dimensionen $D(x)$ für 100 Off-Lattice DLA Cluster mit 10^6 Teilchen (+). Die Daten von Amitrano et al. sind mit × gekennzeichnet. Zusätzlich haben wir das Multiscaling des Clusters mit 50.000.000 Teilchen bestimmt. Diese Daten sind mit $*$ markiert.

systematischen Abfall von $D(x)$ für größere x reproduzieren. Innerhalb der statistischen Fehlerbalken stimmen die Daten für $M = 10^6$ mit den Ergebnissen von Amitrano et al. überein.

Für den sehr großen Cluster mit $M = 50.000.000$ können wir das Multiscaling-Verhalten nicht nachweisen. Statt dessen finden wir lediglich statistische Fluktuationen um die globale fraktale Dimension D. Da hier jedoch nur ein einzelner Cluster zur Verfügung steht, kann dies nicht als Gegenbeweis für Multiscaling gewertet werden, da die statistische Aussagekraft zu gering ist. Allerdings ist zu beachten, daß man selbst mit diesem einzelnen Cluster sehr genau die fraktale Dimension des inneren Clusterbereichs bestimmen kann. Das bedeutet, daß die fraktale Dimension an sich eine recht stabile und selbstmittelnde Größe ist, daß allerdings die Messung von Multiscaling sehr empfindlich sein kann.

Um eine gewisse Vorstellung von der Größe der Fluktuationen der $D(x)$ bei $M = 50.000.000$ zu bekommen, haben wir $D(x)$ an dem gleichen Cluster für $M = 25.000.000$ und $M = 1.000.000$ gemessen. Das Ergebnis ist in der Abbildung 15 gezeigt. Man findet, daß die $D(x)$ für alle drei Clustermassen im inneren Clusterbereich $x < 1.5$ übereinstimmen, allerdings für $x > 1.5$ sehr stark fluktuieren. Man kann daher schließen, daß tatsächlich der interessante Bereich für große x von statistischen Fluktuationen überdeckt ist.

In ersten Messungen konnten wir den Abfall von $D(x)$ bei großen Clustermassen nicht reproduzieren. Statt dessen erhielt man lediglich eine Fluktuation von $D(x)$ um einen mittleren Wert $D(x) \approx 1.69$. Allerdings mußte dieses Verhalten auf einen numerischen Fehler zurückgeführt werden: Die Qualität der von uns benutzten Zufallszahlen war nicht ausreichend, so daß der Schwerpunkt der generierten Cluster langsam in die x-Richtung driftete. Das resultierte in einer Verzerrung, die das Auftreten von Multiscaling überdeckte. Allerdings machte sich dieser Fehler erst für große Clustermassen bemerkbar: An Clustern

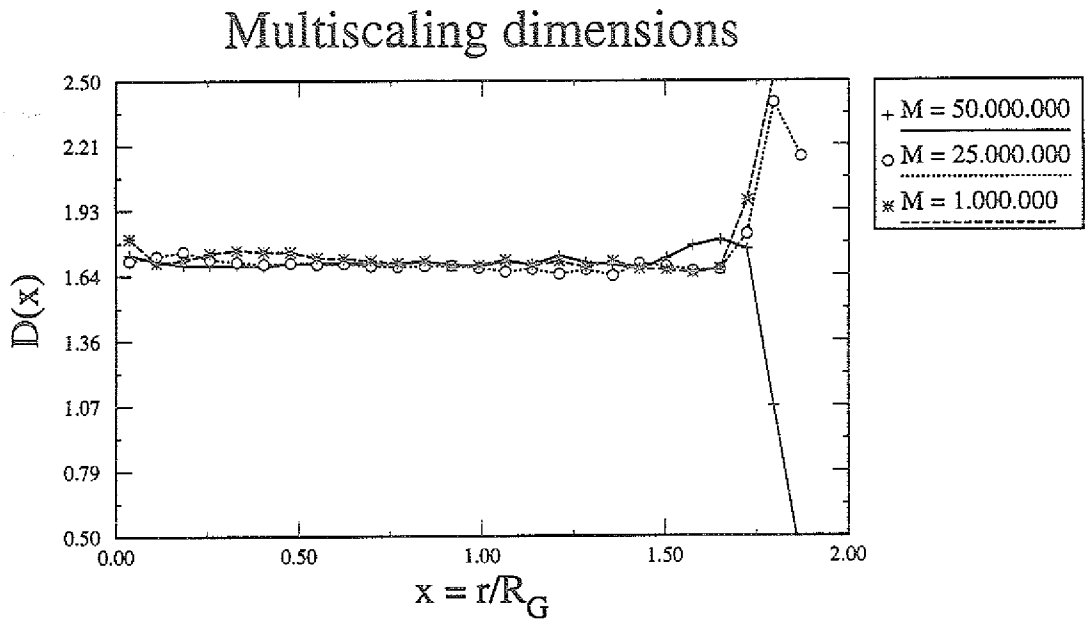


Abbildung 15 Messung der Multiscaling-Dimensionen an einem einzelnen Cluster bei verschiedenen Clustermassen.

mit 10^5 Teilchen konnten wir das Auftreten von Multiscaling exakt reproduzieren. Das bedeutet, daß Multiscaling eine sehr empfindliche Größe ist, die durch geringe Anisotropien zerstört werden kann. Besonders die Generierung sehr großer DLA Cluster stellt daher hohe Anforderungen an die benutzten Zufallszahlen.

3.5 Zusammenfassung

In dieser Arbeit haben wir eine detaillierte Beschreibung eines effizienten Algorithmus zur Generierung großer DLA Cluster gegeben. Mit Hilfe dieses Algorithmus haben wir 100 Off-Lattice DLA Cluster mit 10^6 Teilchen generiert und den kürzlich vorgeschlagenen Multiscaling-Ansatz geprüft [86]. Wir haben auch für große DLA Cluster den systematischen Abfall der Multiscaling-Dimension $D(x)$ bei großen x gefunden, wie er bereits von Amitrano et al. gemessen und von Coniglio und Zannetti vorhergesagt wurde. Daneben haben wir versucht, Multiscaling an einem Cluster mit 50.000.000 Teilchen nachzuweisen. Hier haben wir allerdings nur eine statistische Fluktuation um die globale fraktale Dimension $D \approx 1.69$ gefunden. Obwohl wir selbst an diesem einzelnen Cluster recht zuverlässig die fraktale Dimension im Clusterinneren bestimmen, kann dies nicht als ein Nachweis für das Verschwinden von Multiscaling bei großen Clustermassen gewertet werden, da nur eine unzureichende Statistik vorliegt.

Darüberhinaus haben wir die statistische Selbstähnlichkeit großer DLA Cluster betrachtet. Wir haben gezeigt, daß Off-Lattice DLA Cluster selbst für extrem große Massen keinerlei Tendenz zu einer Anisotropie aufweisen und man daher von einer geometrischen Selbstähnlichkeit großer Cluster ausgehen kann.

Kapitel 4

Wachstumswahrscheinlichkeiten in DLA[+]

Sowohl mit analytischen als auch mit numerische Methoden untersuchen wir in diesem Kapitel die radialen Wachstumswahrscheinlichkeiten $P(r, M)$ für große Off-Lattice DLA Cluster. Für den Fall, daß $P(r, M)$ eine Gauß-Verteilung ist, zeigen wir analytisch, daß die Breite $\xi(M)$ der Verteilung *nicht* wie der Radius of Gyration R_G skalieren kann. Wir generieren weiter 1750 Cluster mit jeweils bis zu 500,000 Teilchen und messen die Verteilung $P(r, M)$ mit weiteren 10^6 Random-Walkern für jeden Cluster. Wir zeigen, daß die gemessene Verteilung ein Potenz-Verhalten im Inneren ($r \sim 0$) des Clusters besitzt und beschrieben werden kann durch den Skalen-Ansatz $P(r, M) \propto \frac{r^\alpha}{\xi} \cdot g\left(\frac{r-r_0}{\xi}\right)$, wobei $g(x)$ eine Skalenfunktion bedeutet, die um Null herum lokalisiert ist und eine Breite Eins besitzt. Der Exponent α wird bestimmt als ≈ 2. Indem wir diese Potenzabhängigkeit einbeziehen, zeigen wir, daß es keinen Widerspruch mehr impliziert, wenn die Breite doch wie R_G skaliert, solange $\alpha > D_f - 1$ ist.

4.1 Einleitung

Das Wachstum von DLA [9] Clustern kann beschrieben werden durch eine Menge von Wachstumswahrscheinlichkeiten $\{p_i\}$, von denen jede die Wahrscheinlichkeit angibt, daß das Teilchen i von dem nächsten eintreffenden Random-Walker erreicht wird. Die Bestimmung der p_i und deren Verteilung hat in letzter Zeit große Beachtung gefunden und eine sehr kontroverse Diskussion ausgelöst [52,93,94,54]. Numerische Berechnungen in diesem Gebiet werden üblicherweise durchgeführt, indem man die Laplace Gleichung exakt für einen bestimmten Cluster löst [55,53] und die Wachstumswahrscheinlichkeiten dann über das elektrische Feld erhält [11]. Allerdings beschränkt hier die zur Verfügung stehende Computer-Leistung die Clustermassen auf 50.000 Teilchen. Eine weitere Größe, die allerdings wesentlich einfacher in Simulationen großer Cluster zugänglich ist, sind die integrierten radialen Wachstumswahrscheinlichkeiten $P(r, M)$, die die Wahrscheinlichkeit angeben, daß das nächste ankommende Teilchen den Cluster der Masse M in einem Abstand r vom Zentrum berührt. Diese Größe

[+] Eingereicht bei J. Phys. A: Mathematical and General [92]

wurde zunächst von Plischke und Rácz [88] betrachtet, die die ersten beiden Momente dieser Verteilung bestimmten und ein Gaußsches Verhalten feststellten

$$P(r, M) \propto \exp\left(-\frac{(r-r_0)^2}{2\xi^2}\right). \tag{27}$$

Mit Hilfe von 4000 Clustern mit bis zu 2.500 Teilchen erhalten sie ein Potenzverhalten des Mittelwerts r_0 und der Breite ξ dieser Verteilung

$$r_0 \propto R_G \propto M^{\nu},\ \xi \propto M^{\nu'} \tag{28}$$

mit Exponenten $\nu = 1/D_f \approx 0.585$, $\nu' \approx 0.48$, wobei D_f die fraktale Dimension und R_G den Radius of Gyration des Clusters bedeutet. Später zeigten Meakin und Sander mit Simulationen von wesentlich größeren Clustern [95], daß der Exponent ν' sich mit wachsender Clustermasse ν nähert und warfen dadurch die Frage des tatsächlichen Verhaltens der Breite ξ auf. Momentan werden drei verschiedene Möglichkeiten in Betracht gezogen

1. Die Breite skaliert mit dem gleichen Exponenten ν wie auch der Radius of Gyration R_G: $\xi(M) \propto M^{\nu}$.
2. Die Breite skaliert mit einem kleineren Exponenten ν': $\xi(M) \propto M^{\nu'}$, $\nu' < \nu$
3. Die Breite skaliert mit ν, hat aber zusätzlich logarithmische Korrekturen $\xi(M) \propto M^{\nu}/(\ln(M))^{\beta}$

Besonders der letzte Fall (3) hat einiges Interesse ausgelöst [86,43], da er für $\beta = 1/2$ ein Multiscaling Verhalten der Massendichte impliziert $M(x) \propto r^{D(x)-1}$.

Im folgenden werden wir zeigen, daß der Fall (1) zusammen mit der Annahme einer Gauß-Verteilung zu einer unphysikalischen Singularität der Massendichte im Grenzfall unendlicher Clustergröße führt. Dieses Resultat ist eine Konsequenz der Tatsache, daß die Gauß-Verteilung hier nicht schnell genug zum Clusterzentrum hin abfällt, was schließlich dazu führt, daß $\nu' = \nu$ nicht korrekt sein kann.

Andererseits finden wir numerisch, daß die Gauß-Verteilung selbst keine gute Beschreibung der Wachstumswahrscheinlichkeiten nahe dem Clusterzentrum ist, denn nahe dem Clusterzentrum hängen sie mit einem Potenzgesetz vom Radius r ab [96]. Des weiteren muß man sich klar machen, daß $P(r, M)$ für große r nur deswegen verschwindet, weil der Cluster eine endliche Masse und deshalb nur eine endliche Ausdehnung besitzt. Solche Effekte werden aber üblicherweise durch einen Finite-Size Cutoff einer universellen Funktion berücksichtigt.

Hier untersuchen wir die Wachstumswahrscheinlichkeiten $P(r, M)$. Wir werden zeigen, daß die numerischen Daten lediglich im Bereich um das Verteilungsmaximum herum mit einer Gauß-Verteilung konsistent sind, aber daß diese Verteilung im Bereich für kleine r nicht korrekt ist. Wir werden zeigen, daß ein anderes Verhalten $P(r, M) \propto \frac{r^{\alpha}}{\xi} \cdot g\left(\frac{r-r_0}{\xi}\right)$ — wobei $g(x)$ eine Skalenfunktion ist — sowohl das Maximum, als auch den inneren Abfall für kleine Abstände gut beschreibt. Eine Berücksichtigung des Potenzgesetzes entfernt die eben beschriebene Singularität.

Die Aufteilung dieses Kapitels ist daher die folgende. Im Abschnitt 4.2 analysieren wir die Konsequenzen für die verschiedenen Verhaltensweisen von $\xi(M)$ auf die Massendichte. Im Abschnitt 4.3 untersuchen wir die numerischen Resultate für die Wachstumswahrscheinlichkeiten und im Abschnitt 4.4 untersuchen wir die Konsequenzen dieser numerischen Resultate für die Massendichte.

4.2 Analytische Ergebnisse

Analog zu [53] definieren wir zwei Längenskalen, einen Radius $r_0(M)$ und eine Breite $\xi(M)$, als $r_0(M) \equiv \langle r \rangle$, $\xi^2(M) \equiv \langle r^2 \rangle - \langle r \rangle^2$ und $r = |\vec{r}|$. Hierbei ist$\langle f(r) \rangle \equiv \int f(r) P(r, M) dr$. Im folgenden nehmen wir an, daß $P(r, M)$ durch eine Gauß-Verteilung beschrieben werden kann,

$$P(r, M) = \frac{1}{\sqrt{2\pi\xi^2(M)}} \exp\left(\frac{-(r - r_0(M))^2}{2\xi^2(M)}\right). \tag{29}$$

Nun untersuchen wir, ob diese Form von $P(r, M)$ konsistent mit einem fraktalen Wachstum ist. Wir beginnen mit der Beziehung

$$N(r, M) = \int_1^M P(r, M')\, dM' \tag{30}$$

wobei $N(r, M)dr$ die Anzahl von Teilchen im Intervall $[r, r + dr]$ in Clustern der Masse $< M$ ist. Gleichung (30) kann hergeleitet werden aus der Tatsache, daß jedes Teilchen, das innerhalb des Intervalls $[r, r + dr]$ in einem Cluster der Masse M liegt, in einem früheren Wachstumstadium ($1 < M' < M$) dem Cluster zugefügt wurde [95].

Fall (1): Nehmen wir nun an, daß $r_0(M) \equiv AM^\nu$ und $\xi(M) \equiv BM^\nu$ gilt, wobei A, B Konstanten sind und höhere Korrekturen vernachlässigt werden können. Mit Hilfe von Gl. (30) erhalten wir

$$\begin{aligned} N(r, M) &= \frac{1}{\sqrt{2\pi B^2}} \int_1^M \frac{1}{M'^\nu} \exp\left(-\frac{(r - AM'^\nu)^2}{2B^2 M'^{2\nu}}\right) dM' \\ &= \frac{1}{\sqrt{2\pi B^2}} \exp\left(-\frac{A^2}{2B^2}\right) \int_1^M \frac{1}{M'^\nu} \exp\left(-\frac{r^2}{2B^2 M'^{2\nu}} + \frac{rA}{B^2}\right) dM'. \end{aligned} \tag{31}$$

Durch eine Variablensubstitution zu $x \equiv M'^{-\nu}$ erhält man hieraus

$$N(r, M) = \frac{1}{\nu\sqrt{2\pi B^2}} \exp\left(-\frac{A^2}{2B^2}\right) \int_{M^{-\nu}}^1 \frac{1}{x^{1/\nu}} \exp\left(-\frac{r^2 x^2}{2B^2} + \frac{rAx}{B^2}\right) dx. \tag{32}$$

Die gesamte Masse, die in der Schale $[r, r + dr]$ enthalten ist, ist per definitionem $N(r, M)dr$. Für einen festen Wert von r wächst $N(r, M)$ für größere Werte von M. Besonders im Grenzfall $M \to \infty$ divergiert dieses Integral wegen der Singularität bei $x = 0$, denn $1/\nu$ ist größer als Eins. Diese Divergenz ist inkonsistent mit der akzeptierten fraktalen Struktur von DLA.

Fall (2): Betrachte nun den Fall, daß $\xi \equiv BM^{y\nu}$ mit $0 < y < 1$ ist. Das Gl. (31) entsprechende Integral ist nun

$$N(r, M) = \frac{1}{\sqrt{2\pi B^2}} \int_1^M \frac{1}{M'^{y\nu}} \exp\left(-\frac{(r - AM'^\nu)^2}{2B^2 M'^{2y\nu}}\right) dM'. \tag{33}$$

Substitution der Variablen zu $x \equiv M'^{-\nu}$ liefert nun

$$N(r,M) = \frac{1}{\nu\sqrt{2\pi B^2}} \int\limits_{M^{-\nu}}^{1} x^{y-1-1/\nu} \exp\left(-\frac{A^2}{2B^2}x^{2(y-1)} - \frac{r^2}{2B^2}x^{2y} + \frac{rA}{B^2}x^{2y-1}\right) dx. \quad (34)$$

Im Grenzfall $M \to \infty$ wird dies zu

$$N(r,M) = \frac{1}{\nu\sqrt{2\pi B^2}} \int\limits_{0}^{1} x^{y-1-1/\nu} \exp\left(-\frac{(rx-A)^2}{2B^2}x^{2(y-1)}\right) dx \quad (35)$$
$$\propto r^{1/\nu-1}.$$

Da $D_f \equiv 1/\nu$ liefert Gl. (35) $N(r,M) \sim r^{D_f-1}$, was konsistent ist mit der Tatsache, daß die Cluster fraktal sind mit einer Dimension D_f.

Fall (3): Betrachte schließlich den Fall $\xi \equiv BM^\nu/\sqrt{\ln M}$. Hier erhält man analog

$$N(r,M) = \frac{1}{\sqrt{2\pi B^2}} \int\limits_{1}^{M} \frac{\sqrt{\ln M'}}{M'\nu} M'^{-(rM'^{-\nu}-A)^2/2B^2}\, dM'. \quad (36)$$

Substitution der Variablen $x \equiv M'^{-\nu}$ liefert nun hieraus

$$N(r,M) = \frac{1}{\sqrt{2\pi\nu^3 B^2}} \int\limits_{M^{-\nu}}^{1} \sqrt{-\ln x}\; x^{-1/\nu+(rx-A)^2/(2\nu B^2)}\, dx. \quad (37)$$

Da die logarithmischen Korrekturen der Breite nicht nur in der Normierung der Gauß-Verteilung, sondern auch im Exponenten erscheinen, ist das Verhalten hier gegenüber dem Fall (1) entscheidend verändert. Dort findet man ein Verhalten $x^{-1/\nu}\cdot\exp(-ax^2+bx+c)$ (Gl. (32)), während man hier $x^{-1/\nu+ax^2+bx+c}$ erhält. Das Integral in Gl. (37) kann nicht in geschlossener Form ausgewertet werden, aber trotzdem kann man einige Eigenschaften bestimmen. Zunächst betrachte den Grenzfall $M \to \infty$. Der logarithmische Term im Integranden divergiert bei $x = 0$ wenn der Exponent des Potenz-Terms $(A^2 - 2B^2)/(2\nu B^2)$ kleiner als -1 ist. Daher konvergiert das gesamte Integral nur dann, wenn $A^2 > 2(1-\nu)B^2$ gilt. Wenn das Integral in (37) konvergiert, kann der Wert bestimmt werden, indem der $\ln x$ Term als konstant betrachtet wird. Die resultierende Gestalt ist konsistent mit der Idee von Multiscaling.

Die hier dargestellte Analyse zeigt, daß falls $P(r,M)$ gaußförmig ist, die Breite $\xi(M)$ von $P(r,M)$ nicht wie R_G skalieren kann. Die Divergenz im Fall (1) deutet darauf hin, daß das Innere des Clusters nicht stark genug von den Randzonen abgeschirmt wird. Daher sollte $\xi(M)$ langsamer anwachsen als R_G wenn M gesteigert wird.

Im nächste Abschnitte wollen wir numerisch untersuchen, ob die Annahme einer Gauß-Verteilung für $P(r,M)$ gerechtfertigt ist.

4.3 Numerische Ergebnisse

Um die Wahrscheinlichkeiten $P(x,M)$ (hier haben wir skaliert $x = r/R_G$) numerisch zu bestimmen, haben wir jeweils 150 verschiedene Cluster mit 10.000, 20.000, 50.000, 200.000 und 500.000 Teilchen generiert und zusätzlich weitere 1000 Cluster mit jeweils

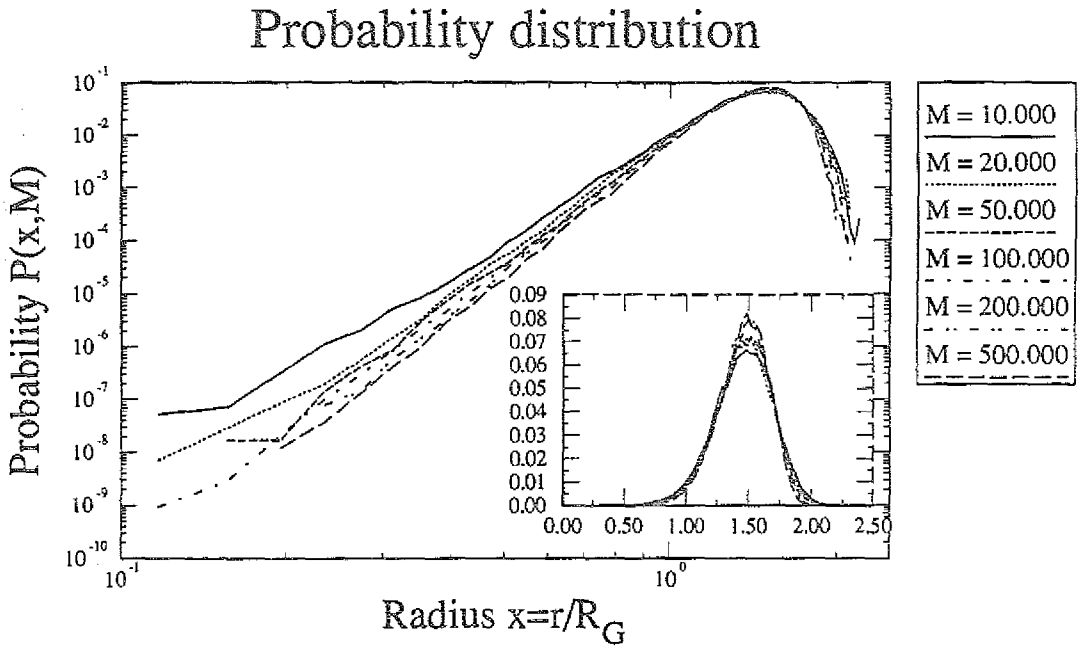


Abbildung 16 Radiale Wachstumswahrscheinlichkeit für verschiedene Clustermassen in linearer und doppelt logarithmischer Skala.

100.000 Teilchen. Die Wachstumswahrscheinlichkeiten sind in einer statischen Messung bestimmt worden, in der pro Cluster weitere 10^6 Random-Walker gestartet werden und der Radius r des Auftreffpunktes gemessen wurde, ohne jedoch den Cluster wachsen zu lassen. Mit Hilfe des vorhin beschriebenen schnellen Algorithmus [43], kann die Generierung eines 100.000 Teilchen Clusters und die Messung von $P(x, M)$ innerhalb von 90 Minuten auf einer SPARCstation 2 erfolgen. Um die $P(x, M)$ zu bestimmen, diskretisieren wir diese Größe für $x \in [0, 2.5]$ an 64 äquidistanten Punkten und zählen die Anzahl von Teilchen, die den Cluster in einem Abstand zwischen x und $x + dx$ vom Zentrum berühren. Abbildung 16 zeigt die Wachstumswahrscheinlichkeiten für alle Clustermassen in linearer und logarithmischer Darstellung. Man beachte das ausgeprägte Potenz-Verhalten von $P(x, M)$ für kleine x — eine Eigenschaft, die bei Gauß-Verteilungen nicht gefunden werden kann. Um die statistischen Fluktuationen abzuschwächen zeigen wir in Abbildung 17 die integrierte Verteilung $IP(x, M) = \int_0^x P(x', M)\, dx'$. Bei kleinen Clustermassen — bis zu 100.000 Teilchen — sieht man deutlich eine systematische Abnahme der Breite der Verteilung. Für die größten Massen kann solch ein systematisches Verhalten nicht direkt beobachtet werden. Die Verteilungen für 200.000 und 500.000 Teilchen liegen direkt aufeinander.

Ein quantitative Messung der Breite ξ erfolgt auf zwei Methoden. Bei der ersten bestimmt man die Breite durch eine direkte Messung des ersten und des zweiten Moments der Verteilung und bestimmt die Breite als $\xi/R_G = \sqrt{\langle x^2 \rangle - \langle x \rangle^2}$ und bei der zweiten Methode benutzen wir eine nichtlineare Fit-Routine, um direkt eine Gauß-Verteilung an unsere Daten anzupassen. Mit beiden Methoden erhält man ähnliche Resultate für das Verhalten der Breite

$$\begin{aligned} \xi_{\text{Moments}} &\propto R_G \cdot M^{-0.044 \pm 0.003} \\ \xi_{\text{Gauss}} &\propto R_G \cdot M^{-0.050 \pm 0.003} \end{aligned} \tag{38}$$

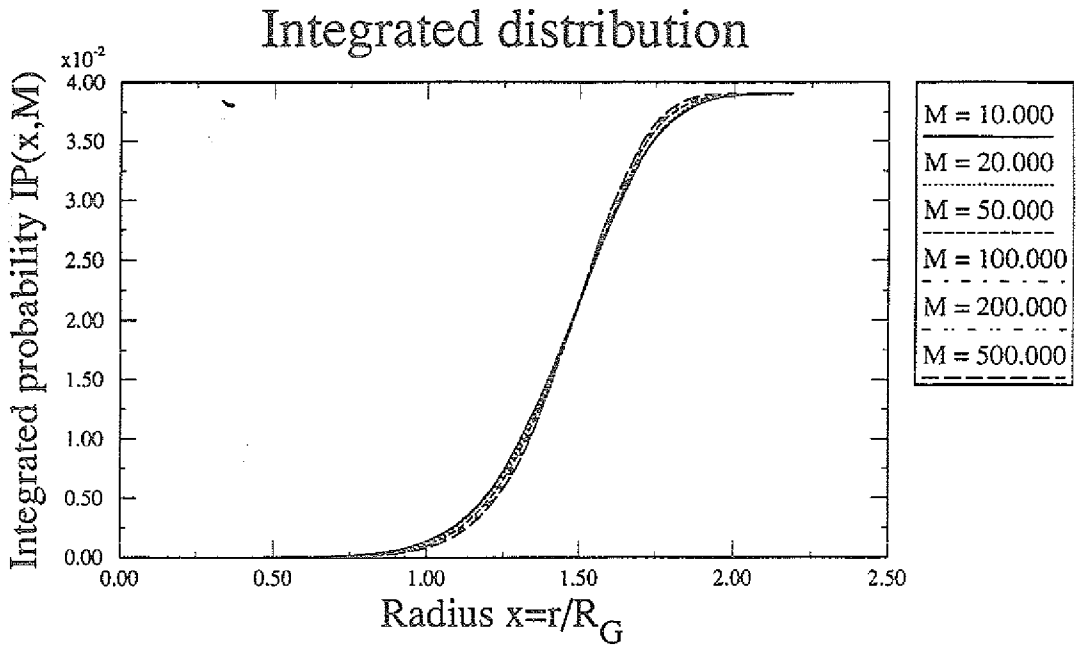


Abbildung 17 Integrierte Wachstumswahrscheinlichkeit.

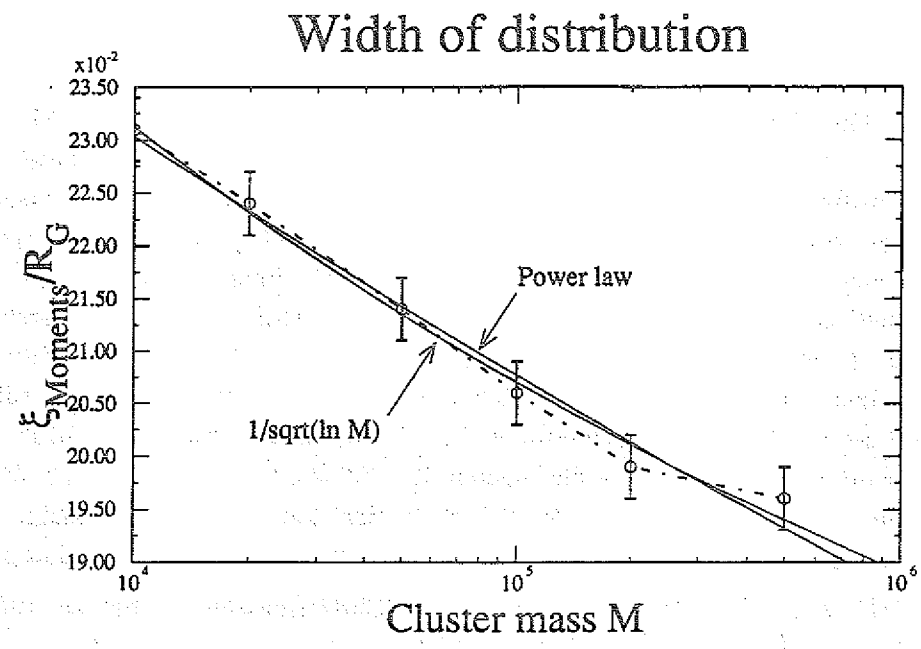


Abbildung 18 Skalenverhalten der Breite der Verteilung (bestimmt durch direkte Messung der beiden ersten Momente). Die durchgezogenen Linien bezeichnen die Fits eines Potenzgesetzes und einer Abhängigkeit $1/\sqrt{\ln M}$ an unsere Daten. Diese beiden Möglichkeiten werden durch die Fälle (2) und (3) vorgeschlagen.

Wir prüfen auch, ob unsere Daten konsistent mit dem Fall (3) sind und finden

$$\xi_{\text{Moments}} \propto \frac{R_G}{\sqrt{\ln\,(1.07 \cdot M)}}. \tag{39}$$

In Bild 18 zeigen wir die Daten für die Breite ξ_{Moments}/R_G in Abhängigkeit von der Cluster Masse. Die durchgezogenen Linien sind die Resultate eines Fits eines Potenz Gesetzes und

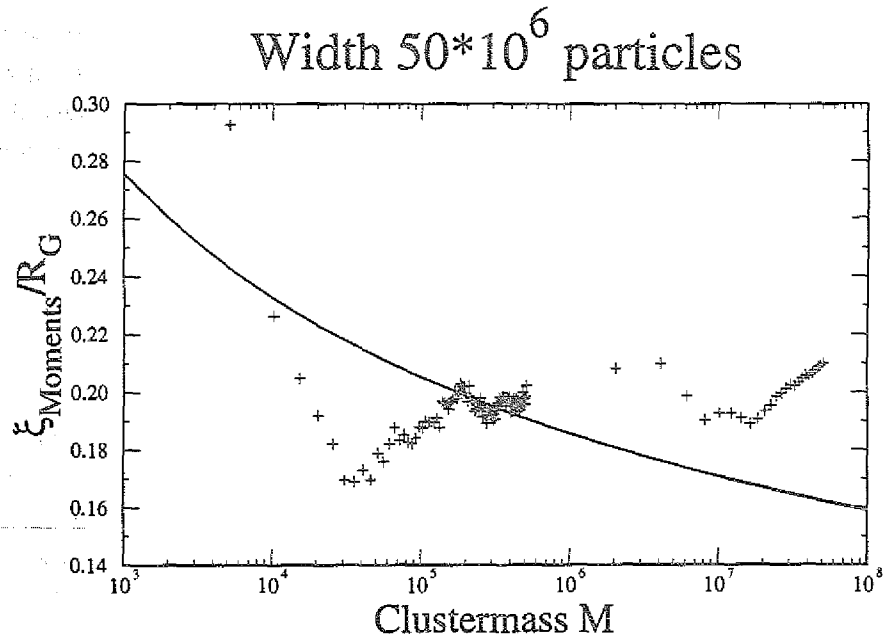


Abbildung 19 Breite der aktiven Zone in einem einzelnen Cluster mit $50 \cdot 10^6$ Teilchen. Die durchgezogene Linie zeigt den Fit einer Funktion $\xi/R_G \propto 1/\sqrt{\ln M}$ an die Daten kleinerer Cluster.

eines $1/\sqrt{\ln M}$ Verhaltens an unsere Daten. Dies entspricht den oben vorgeschlagenen Fällen (2) und (3). Für Cluster mit bis zu 200.000 Teilchen sieht man eine gute Übereinstimmung mit der Annahme einer logarithmischen Korrektur. Unsere Daten scheinen daher nicht mit dem Fall (1) $\nu' = \nu$ konsistent zu sein, sondern scheinen mit den Fällen (2) und (3) übereinzustimmen.

Neben diesen Daten haben wir zusätzlich die Breite ξ in einem einzelnen Cluster mit $50 \cdot 10^6$ Teilchen gemessen. Allerdings konnten wir hier keine statische Messung vornehmen, sondern haben ξ während des Wachstums bestimmt. Dazu wurde für jeweils 5.000 (bei $M \leq 500.000$) bzw. 2.000.000 (bei $500.000 < M \leq 50.00.000$) eintreffende Teilchen die Größe $x = r/R_g$ bestimmt und ξ/R_G durch die Mittelwerte $< x >$ bzw. $< x^2 >$ bestimmt als $\xi/R_G = \sqrt{\langle x^2 \rangle - \langle x \rangle^2}$. Das Resultat ist in der Abb. 19 in Abhängigkeit von der Clustermasse M gezeigt. Die durchgezogene Linie ist dort der eben bestimmte Fit der Funktion $1/\sqrt{\ln M}$. Den interessanten Bereich dieser Daten findet man bei großen M. Hier scheint die relative Breite ξ/R_G nicht weiter zu fallen, sondern um einen konstanten Wert zu fluktuieren. Allerdings können diese Fluktuationen nicht zur Bestimmung eines statistischen Fehlers benutzt werden, da sie von korrelierten Daten stammen: Sie stammen alle von einem einzelnen Cluster, dessen Gestalt für verschiedene Massen stark korreliert ist. Allerdings kann man diese Daten als einen Hinweis darauf werten, daß die relative Breite für sehr große Cluster doch möglicherweise gegen einen konstanten Wert ≈ 0.2 strebt.

Als eine Konsequenz dieser Ergebnisse kann man nun fragen, ob die Annahme einer gaußförmigen Verteilung gerechtfertigt ist. Diese Annahme basiert völlig auf den frühen Messungen von r_0 und ξ von Plischke und Rácz [88] und wird nicht notwendig durch analytische Argumente gefordert. Daher nehmen wir ein Skalenverhalten der Form

$$P(x, M) \propto \frac{x^\alpha}{\xi/R_G} \cdot f\left(\frac{x - x_0}{\xi/R_G}\right) \tag{40}$$

an, mit einer Skalenfunktion $f(x)$. Das hier angenommene Verhalten legt ein größeres Gewicht auf den Potenzabfall für kleine x im Vergleich zum Gaußschen Ansatz, der ein großes Gewicht auf das Maximum der Verteilung legt. Dieser Ansatz wird dadurch motiviert, daß der Abfall von $P(x)$ für große x lediglich auf die endliche Größe des Clusters zurückzuführen ist. Dieser Effekt wird üblicherweise durch einen Skalenansatz der obigen Form berücksichtigt.

Um unsere Daten anzupassen benutzen wir zunächst einen einfachen, ad hoc gemachten Ansatz für die Skalenfunktion

$$f(x) = \tanh\left(-\frac{x^2}{2} \cdot \mathrm{sign}(x)\right) + 1 \tag{41}$$

welcher die wesentlichen Züge einer Cutoff Funktion aufweist — $f(x) \to$ const. für $x \ll 1$ aber $f(x) \to 0$ für $x \gg 1$ — aber zusätzlich ein gaußförmiges Verhalten in den Flanken besitzt[4]. In Bild 20 vergleichen wir die Fits der beiden Test-Funktionen — Gauß-Verteilung und Power-Law $\times$ cutoff — für die 100.000–Teilchen Daten. Für Gl. (40) haben wir einen gewichteten Fit durchgeführt, während der Fit der Gauß-Verteilung sowohl gewichtet als auch ungewichtet durchgeführt wurde.

In einem ungewichteten Fit liefert die Gauß-Verteilung eine gute Beschreibung für das Maximum der Verteilung, aber auf der logarithmischen Skala findet man deutlich, daß der innere Bereich für kleine x nicht beschrieben wird. Man findet eine starke Aufwärtskrümmung, die ein Resultat der Tatsache ist, daß die Gauß-Verteilung bei $x = 0$ im Gegensatz zum Potenzgesetz nicht verschwindet. In einem gewichteten Fit sind dagegen weder dieser Schwanz, noch das Maximum der Verteilung korrekt beschrieben. Unsere ad hoc Annahme (41) dagegen liefert eine gute Beschreibung der Daten sowohl im Maximum, als auch in der Flanke für kleine x.

In den Tabellen 2-3 zeigen wir die Ergebnisse für die Parameter x_0, ξ/R_G und α für unsere verschiedenen Ansätze und die verschiedenen Fit-Verfahren. Der Fit der Gauß-Verteilung zeigt deutlich die wohlbekannte Tatsache, daß die Position des Maximums proportional zum Radius of Gyration ist, während die relative Breite der Verteilung langsam abnimmt. Selbst für unsere ad hoc Annahme (41) nimmt die relative Breite ξ mit der Clustermasse ab. Es ist jedoch erstaunlich, daß man für die Flanke des Potenzabfalls einen relativ großen Exponenten α findet, der logarithmisch mit der gesamten Clustermasse zunimmt (Bild 21). Das bedeutet, daß das Innere des Clusters sehr stark durch die Randbereiche abgeschirmt wird. Analytische Rechnungen für N-armige Sterne und Simulationen von kleinen Clustern bestätigen dieses Ergebnis [96]. Andererseits zeigt der langsame Anstieg von α mit wachsender Clustermasse, daß die ad hoc gewählte Skalenfunktion (41) den Exponenten beeinflußt und daher modifiziert werden muß.

Um dies zu berücksichtigen führen wir nun einen vollständigen Daten-Kollaps für die Skalenform (40) durch. Wir benutzen hier für die relative Breite ξ/R_G und die Position des Zentrums x_0 die Werte aus der Tabelle 2 und variieren den Exponenten α. Den besten Daten-Kollaps, der in Bild (22) gezeigt ist, erhalten wir für einen Wert $\alpha = 2.0 \pm 0.4$, welcher wesentlich kleiner ist als der vorher gemessene und auch wesentlich kleiner als der in [52] gefundene. Der vorher bestimmte Wert war daher nur ein "offensichtlicher" Wert, der einen Anteil aus der Potenz-Komponente der Skalenfunktion beinhaltet.

---

[4] Die exakte Form von $f(x)$ ist nicht wichtig, solange die ersten beiden Momente endlich sind.

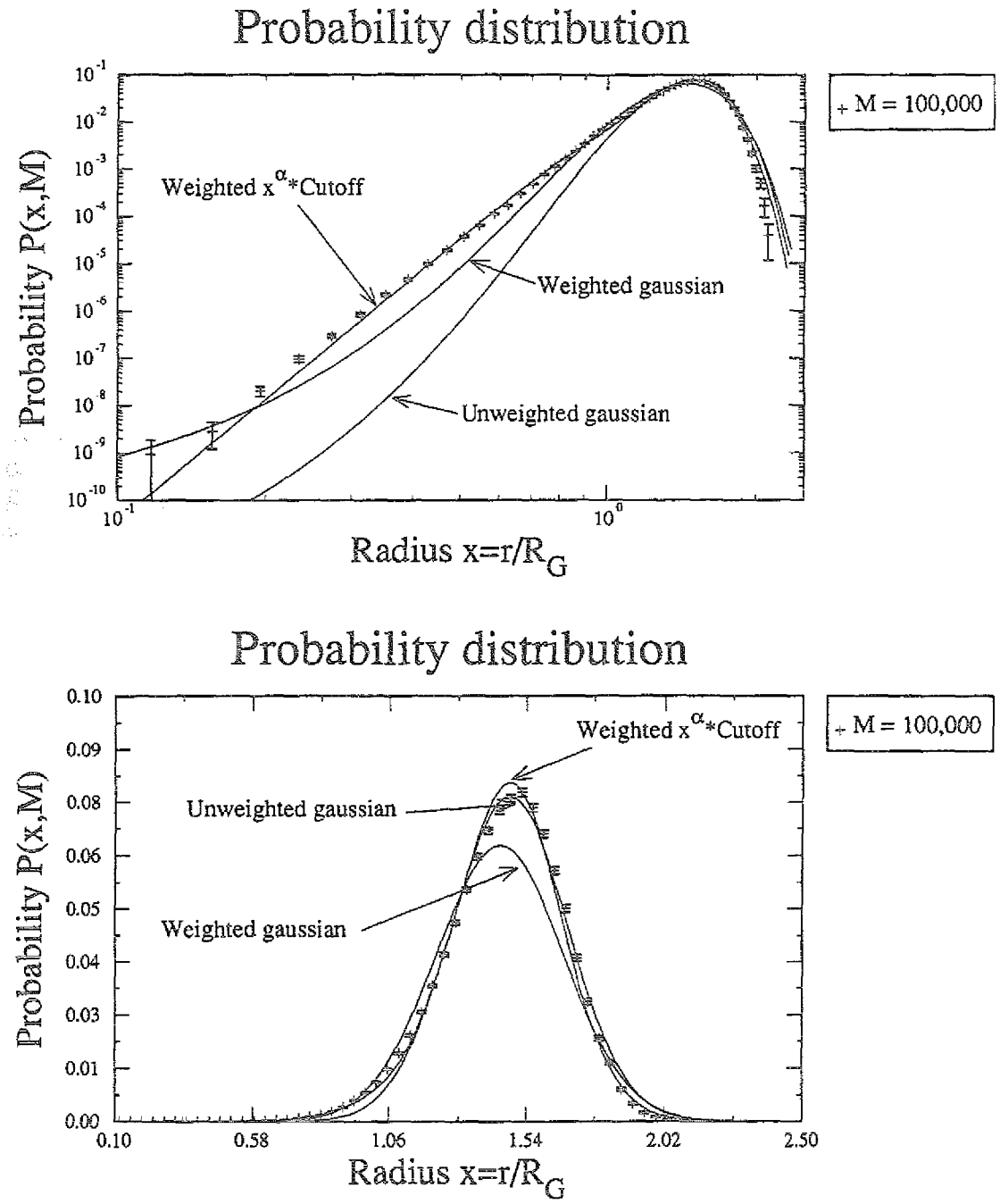


Abbildung 20 Vergleich der drei durchgeführten Fit-Prozeduren: Ungewichteter Fit einer Gauß-Funktion, gewichteter Fit einer Gauß-Funktion und gewichteter Fit des Ansatzes (41). Figur 20a zeigt die Fits in doppelt logarithmischer Skala, während 20b die Fits in linearer Skala zeigt.

4.4 Konsequenzen

Nachdem die Existenz des Potenzabfalls der Verteilung $P(r, M)$ feststeht, wollen wir den Einfluß auf die Resultate aus Abschnitt 4.2 untersuchen. Wir nehmen daher an, daß $P(r, M)$ durch Gl. (40) mit einem Normierungsfaktor N_0 beschrieben werden kann. Im folgenden möge der Mittelwert von $f(x)$ Null und deren Breite eine endliche Konstante sein. Wir

mass	x_0	ξ/R_G
10.000	1.474±0.003	0.232±0.003
20.000	1.475±0.003	0.223±0.003
50.000	1.480±0.003	0.215±0.003
100.000	1.481±0.003	0.203±0.003
200.000	1.480±0.003	0.197±0.003
500.000	1.485±0.003	0.192±0.003

Tabelle 2 Breite und Zentrum der Gauß-Verteilung für einen ungewichteten Fit.

mass	x_0	ξ/R_G	α
10.000	1.08±0.01	0.348±0.005	7.45±0.04
20.000	1.02±0.01	0.366±0.005	8.05±0.04
50.000	1.06±0.01	0.345±0.004	8.44±0.05
100.000	1.134±0.004	0.307±0.001	8.56±0.02
200.000	1.16±0.01	0.292±0.004	8.83±0.06
500.000	1.16±0.01	0.285±0.006	9.10±0.08

Tabelle 3 Breite, Zentrum und Exponent α eines gewichteten Fits mit dem einfachen Skalen-Ansatz (41).

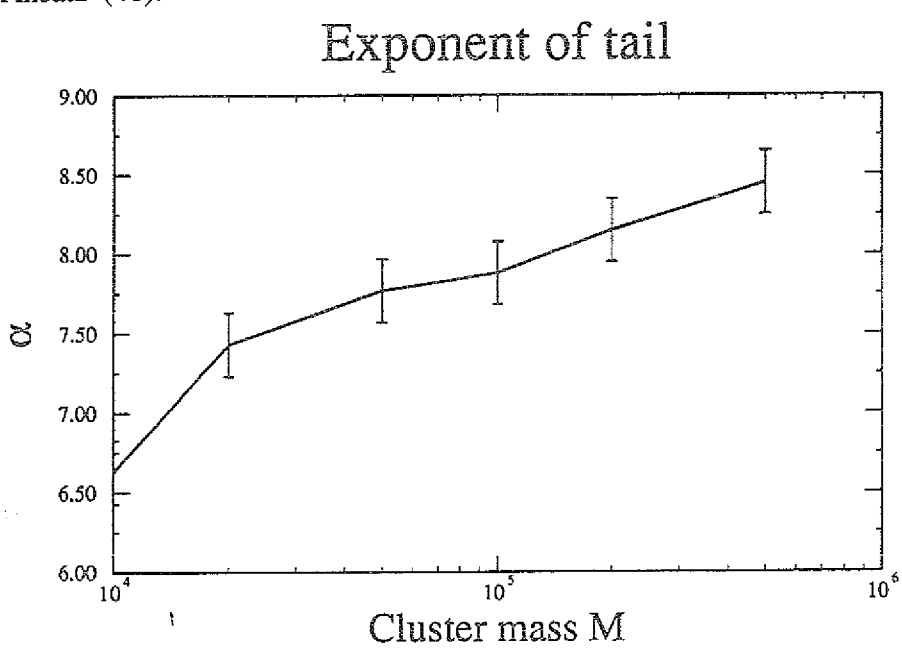


Abbildung 21 Verhalten des Skalen-Exponenten α mit wachsender Clustermasse.

betrachten zunächst Fall (1). Da $P(r, M)$ normiert sein muß, gilt

$$\int_0^\infty \frac{N_0}{\xi}\left(\frac{r}{R_G}\right)^\alpha f\left(\frac{r-r_0}{\xi}\right) dr = 1 \tag{42}$$

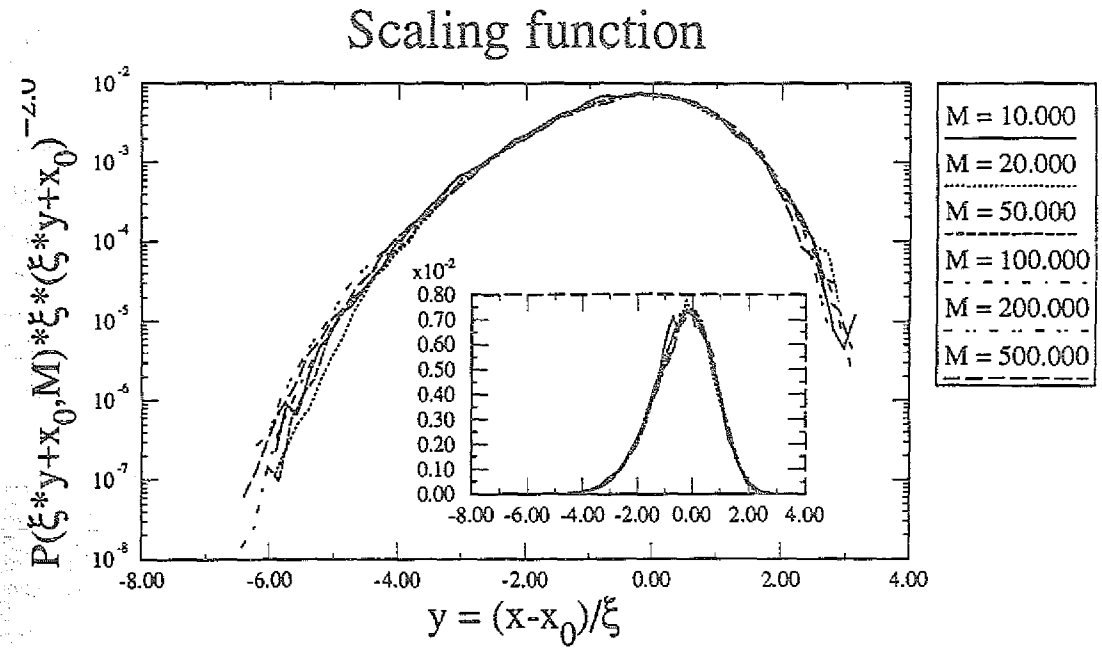


Abbildung 22 Plot der Skalenfunktion $g(y) \propto P(\xi \cdot y + x_0, M) \cdot \xi \cdot (\xi \cdot y + x_0)^{\alpha}$. Sowohl das Maximum der Verteilung, als auch die innere Flanke fallen gut aufeinander.

Nach Substitution der Integrationsvariablen zu $x = r/R_G$ liefert Gl. (42)

$$N_0 \int_0^{\infty} x^{\alpha} f(x - x_0)\, dx = 1, \tag{43}$$

mit $x_0 \equiv r_0/R_G$. Da dieses Integral konstant ist, sollte N_0 wie ξ/R_G skalieren. Die Gleichung, die Gl. (31) entspricht, ist

$$\begin{aligned} N(r, M) &\propto \int_1^M \frac{1}{R_G}\left(\frac{r}{R_G}\right)^{\alpha} f\left(\frac{r - r_0}{\xi}\right) dM \\ &\propto \int_1^M M^{-(1+\alpha)/D_f} r^{\alpha} f\left(\frac{r - r_0}{\xi}\right) dM \end{aligned} \tag{44}$$

Der eine Divergenz erzeugende Term in (44) ist $M^{-(1+\alpha)/D_f}$, da alle anderen Terme für festes r konstant werden. Daher divergiert das Integral (44) nur, wenn der Exponent $(1+\alpha)/D_f$ kleiner als Eins ist, was $\alpha > D_f - 1$ entspricht. Da der numerische Wert von α aber offensichtlich größer als $\alpha > D_f - 1 \simeq 0.71$ ist, wird unser Ansatz nicht länger durch ein Selbst-Konsistenz Argument ausgeschlossen. Auf ähnliche Weise zeigt man, daß die Existenz eines Potenz-Abfalls konsistent mit den beiden anderen Fällen (2) und (3) ist.

4.5 Zusammenfassung

Wir haben die radialen Wachstumswahrscheinlichkeiten $P(r, M)$ für große Off-Lattice DLA Cluster untersucht und gezeigt, daß die übliche Annahme einer Gauß-Verteilung zu unphysikalischen Divergenzen führt, wenn man annimmt, daß die Breite ξ und das Zentrum r_0 der Verteilung beide mit demselben Exponenten skalieren. Wir haben in umfangreichen numerischen

Simulationen gezeigt, daß tatsächlich die Annahme $\nu' = \nu$ mit unseren Daten inkonsistent ist und daß die Fälle (2) und (3) bevorzugt werden müssen. Weiter haben wir gezeigt, daß die Verteilung $P(r, M)$ einem Potenzgesetz für kleine r gehorcht, welches inkonsistent mit der üblichen Annahme einer Gauß-Verteilung ist. Wir haben daher ein Skalenverhalten der Form $P(r, M) \propto \frac{r^\alpha}{\xi} f\left(\frac{r-r_0}{\xi}\right)$ vorgeschlagen, welches die oben beschriebene Singularität entfernt. Dieser Ansatz beschreibt sowohl den Bereich des Maximums als auch das Potenz-Verhalten. Wir haben gezeigt, daß man explizit die Massenabhängigkeit der Breite der Verteilung in der Skalenfunktion $f(x)$ berücksichtigen muß, um den korrekten Exponenten $\alpha \approx 2.$ bestimmen zu können. Ein einfacher ad hoc Ansatz bzw. eine einfache Messung des Potenz-Verhaltens liefert lediglich einen effektiven Exponenten, in dem der tatsächliche Anteil und ein Beitrag der Skalenfunktion vermischt sind.

Leider gestatten unsere Daten nicht, Schlußfolgerungen über die kleinsten Wachstumswahrscheinlichkeiten, die z. B. in [54] studiert werden, zu ziehen. Unsere Daten sind bereits Mittelwerte der lokalen Wachstumswahrscheinlichkeiten, gemittelt über eine ringförmige Schale des Clusters. Wir besitzen daher nur Informationen über die räumliche Verteilung der Wachstumswahrscheinlichkeiten, und nicht aber über die Verteilung $n(P)$ der einzelnen Wahrscheinlichkeiten selbst. Eine direkte Messung der interessanten Eigenschaften dieser Verteilung ist mit Hilfe von Random-Walkern nicht möglich. Dazu muß man das Verhalten der kleinsten Wachstumswahrscheinlichkeiten studieren, welche im Bereich $p_{\text{min}} \approx 10^{-14}$ liegt. Ein derartiger Bereich ist selbst mit hochoptimierten Random-Walker Methoden nicht zugänglich.

Kapitel 5

Zweigordnungen in DLA[+]

Wir benutzen ein hierarchisches Ordnungsschema, um DLA Cluster mit 10^6 Teilchen systematisch in Zweige verschiedener Ordnung aufzuteilen. In jeder Ordnung bestimmen wir die Anzahl, Masse, Länge und Breite der Zweige. Wir finden, daß alle diese Größen exponentiell von der Zweigordnung abhängen. Das bedeutet, daß das Verhältnis aller Größen untereinander konstant ist. Indem wir die Längen, Breiten und Massen der verschiedenen Ordnungen miteinander vergleichen finden wir, daß sowohl die Länge, als auch die Breite der Zweige mit dem gleichen Exponenten $\nu_{\parallel} = \nu_{\perp} = 0.60 \simeq \beta \simeq 1/D$ von der Masse der Zweige abhängen. Durch Betrachtung der Verzweigungsmatrix R_{ik} finden wir, daß die einzelnen Zweigordnungen in einer selbstähnlichen Art verteilt sind. Darüberhinaus messen wir die Winkel, die Zweige verschiedener Ordnung miteinander einschließen. Wir finden, daß diese Winkel gegen einen Grenzwinkel von ungefähr 38° streben. Alle diese Ergebnisse deuten eine statistische Selbstähnlichkeit zwischen Zweigen verschiedener Ordnungen an.

5.1 Einleitung

Vor kurzem wurden verschiedene Ansätze vorgeschlagen, mit denen die Morphologie von DLA Clustern beschrieben werden soll: Zu nennen sind hierbei eine Analyse der Verzweigungsstruktur [98], eine Analyse der Zweig-Ordnungen [99], ein gerichtetes Fraktal [58] oder Untersuchungen der zeitlichen Korrelationen [100]. Wegen der auffallenden baumartigen Struktur der DLA Cluster ist insbesondere der Zweig-Ordnungs Zugang besonders interessant. Bei diesem Ansatz wird der gesamte Cluster in Teile zerschnitten, die Blättern, Zweigen und Ästen eines Baums entsprechen. Hinrichsen et al. [99] fanden z. B. daß das Bifurkationsverhältnis r_N, welches das Verhältnis der Anzahl von Zweigen aufeinanderfolgender Ordnung ist, konstant ist. Weiter weiß man, daß ebenso das Verhältnis der Längen von Zweigen aufeinanderfolgender Ordnungen r_L konstant ist. Dies führt sofort auf eine einfache exponentielle Abhängigkeit der Längen und Anzahlen von der Zweigordnung. Gerade wegen dieser Einfachheit der Resultate scheint das Konzept der Zweigordnungen besonders interessant zu sein.

In dem hier benutzten Formalismus wird jeder Zweig bestimmt durch einen eindeutigen Pfad, der an der Spitze eines Zweiges — d. h. einem Teilchen ohne Nachfolger — beginnt und an dem nächsten Zweig einer höheren Ordnung endet. Die Klassifizierung ist dabei die folgende:

[+] Veröffentlicht in Phys. Rev. A 45, 1058 (1992) [97]

Jeder Zweig, der keine Seitenzweige besitzt — wie ein Blatt eines Baumes — ist von der Ordnung Null. Jeder Zweig, der lediglich Seitenzweige der Ordnung Null besitzt, ist von der Ordnung Eins. Wenn nun N Zweige der Ordnungen $n_1 \ldots n_N$ zusammentreffen, wird der längste von ihnen als Hauptzweig interpretiert und alle anderen als dessen Seitenzweige[6]. Dieser Hauptzweig erhält dann eine neue Zweigordnung $n_1 = \max(n_1, n_2 + 1 \ldots n_N + 1)$, während die Zweigordnung der Seitenzweige nicht geändert wird. Dementsprechend hat jeder Zweig der Ordnung n lediglich Seitenzweige mit einer kleineren Ordnung.

Auf einem Computer kann dies sehr effizient mit einem Burning-Algorithmus [101] implementiert werden: Da Off-Lattice DLA immer schleifenfreie Cluster generiert, kann jedem Teilchen ein eindeutiger Vorgänger zugeordnet werden. Daher ist es möglich, für jeden Zweig einen eindeutigen Pfad von seiner Spitze bis zu seiner Wurzel zu finden. Methodisch geschieht das auf folgende Weise: Man markiert schrittweise alle Teilchen, die zu demselben Zweig gehören. Im ersten Zeitschritt markiert man alle Teilchen an den Spitzen des Clusters, im zweiten Schritt alle Vorgänger der Spitzen und so fort. In jedem Zeitschritt beachtet man den Abstand von der Zweigspitze und die Ordnung des jeweiligen Zweigs. An jedem Verzweigungspunkt — welches alle Teilchen mit mehr als einem Nachfolger sind — wird das Markieren angehalten, bis der letzte und damit längste aller Zweige, die hier beginnen, in diesem Punkt eintrifft. Die Zweigordnung dieses Zweigs wird dann entsprechend den vorigen Regeln geändert und das Burning wird an diesem Punkt fortgesetzt.

Dieses Ordnungsschema führt nur für Zweige von niedriger Ordnung zu stabilen Resultaten, denn nur hier sind die statistischen Fluktuationen der einzelnen Zweigeigenschaften — wie Masse und Länge — relativ gering. Für Zweige mit höherer Ordnung ist dieses Schema weniger scharf. Dies hat zwei Gründe. Zunächst hängt die tatsächliche Zweigordnung der größten Zweige sehr empfindlich von der Struktur der Cluster nahe am Zentrum ab: Hier müssen notwendigerweise alle großen Zweige zusammenlaufen und dadurch kann die Zweigordnung stark verändert werden. Daher ist die Ordnung der großen Zweige nicht sehr gut definiert. Um dieses Problem zu vermeiden, werden die innersten 1024 Teilchen eines jeden Clusters entfernt, so daß die größten Zweige nicht mehr zusammentreffen. Der zweite Grund ist, daß man nur wenige Zweige hoher Ordnung findet und daher starke Fluktuationen und eine nicht ausreichende Statistik zu schlechten Resultaten für die großen Ordnungen führen. Trotz allem muß man beachten, daß die größte vorkommende Zweigordnung in einem Cluster mit 10^6 Teilchen typischerweise lediglich 10 beträgt. Das bedeutet, daß diese Definition der Zweigordnungen sehr stabil gegenüber Fluktuationen an den Spitzen ist.

Für diese Arbeit haben wir 109 Off-Lattice DLA Cluster mit jeweils 10^6 Teilchen generiert. Im Abschnitt 5.2 studieren wir die Anzahl, Masse und geometrische Ausdehnung der verschiedenen Zweigordnungen. Im Abschnitt 5.3 betrachten wir die Winkel zwischen den Zweigen verschiedener Ordnung und im Abschnitt 5.4 untersuchen wir den Einfluß der Definition der Zweigordnungen auf die Skalenergebnisse.

5.2 Anzahl, Masse, Länge und Breite

Im folgenden präsentieren wir die Ergebnisse, die wir für insgesamt 47 DLA Cluster mit jeweils 10^6 Teilchen gewonnen haben. Die am einfachsten zu bestimmende Größe ist die

[6] Als Längenmaß benutzen wir die "chemische Länge", also eigentlich die Masse des Zweiges.

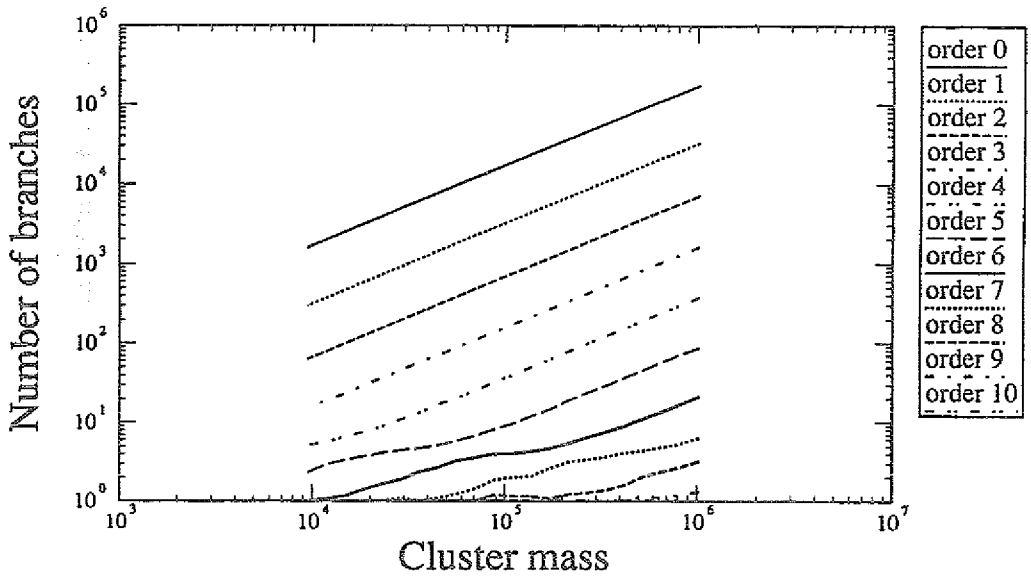


Abbildung 23 Anzahl von Zweigen einer bestimmten Ordnung in Abhängigkeit von der Clustermasse. Der asymptotische Bereich entspricht einer linearen Abhängigkeit.

Anzahl von Zweigen einer bestimmten Ordnung. Hinrichsen et al. haben dies bereits für kleinere Cluster mit bis zu 2.5×10^5 [99] Teilchen gemessen. Die Ergebnisse unserer Messungen der Anzahl in Abhängigkeit von der totalen Masse sind in Bild 23 gezeigt[7]. Man sieht deutlich ein Verhalten, daß man auch bei allen anderen gemessenen Größen wiederfindet: Man kann deutlich für jede Zweigordnung zwischen einem mittleren Regime und einem asymptotischen Regime unterscheiden. Letzteres bezeichnet den Bereich, in dem die gemessenen Größen stabil gegenüber Änderungen der Clustermasse und nur kleinen statistischen Fluktuationen unterworfen sind. Ersteres bezeichnet den Crossover Bereich, in dem Fluktuationen noch eine wichtige Rolle spielen.

Im Falle der Anzahl der Zweige ist der asymptotische Bereich gekennzeichnet durch eine lineare Abhängigkeit der Zweigzahl N_n einer bestimmten Ordnung n von der Clustermasse$M$

$$N_n = a_n * M. \tag{45}$$

Die Steigungen der jeweiligen Geraden a_n hängen exponentiell von der Zweigordnung n ab

$$a_n \propto (0.234 \pm 0.002)^n. \tag{46}$$

Das Bifurkations-Verhältnis r_N ist demnach gegeben durch $r_N = N_{n+1}/N_n = a_{n+1}/a_n = 0.234 \pm 0.002$.

Eine detailliertere Beschreibung der Verhältnisse erhält man durch die Verzweigungs-Matrix R_{ik}, welche für DLA und Viscous-Fingering von Vannimenus und Viennot [98] eingeführt wurde. Wenn N_{ik} die Anzahl von Seitenzweigen der Ordnung i ist, welche auf einem Hauptzweig der Ordnung k sitzen (beachte, daß immer $i < k$ sein muß), dann definieren

[7] Die doppelt logarithmische Skala ist hier gewählt, um die N_n über fünf Größenordnungen deutlich zeigen zu können.

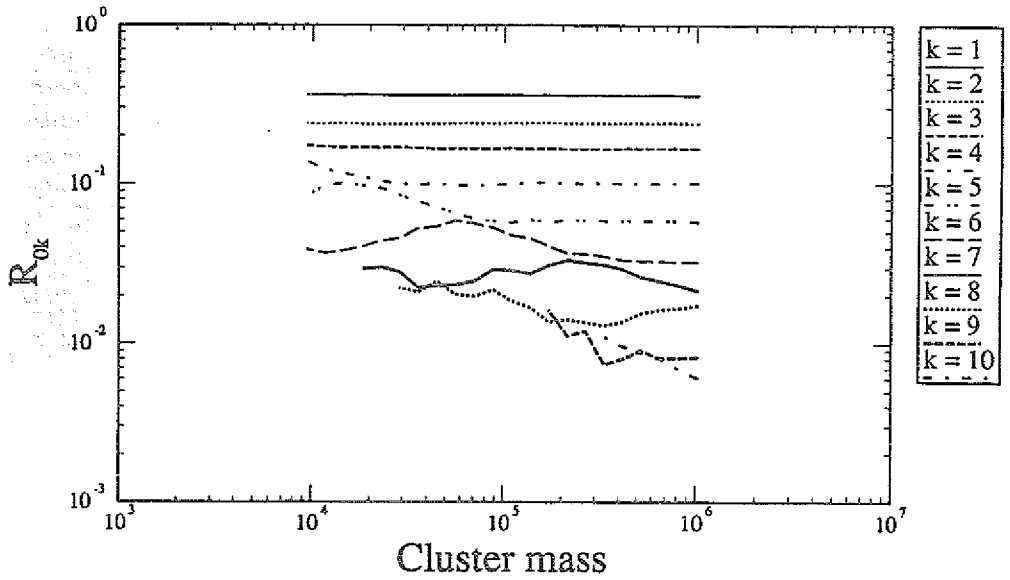


Abbildung 24 Wahrscheinlichkeiten R_{0k}, daß ein Zweig der Ordnung 0 auf einem Zweig der Ordnung k sitzt. Im asymptotischen Grenzfall streben alle R_{0k} einer für die Zweigordnung charakteristischen Konstanten zu.

wir

$$R_{ik} = \frac{N_{ik}}{\sum\limits_{j,j>i} N_{ij}}$$
$$R_{ki} = \frac{N_{ik}}{\sum\limits_{j,j<k} N_{jk}}. \tag{47}$$

Da Seitenzweige notwendigerweise eine kleinere Ordnung besitzen als deren Hauptzweige, muß notwendigerweise $N_{ik} = 0$ für $i \geq k$ gelten. Mit dieser Definition ist die Verzweigungsmatrix R_{ik} bereits korrekt normiert

$$\sum_{j,j>i} R_{ij} = \sum_{j,j<k} R_{ij} = 1 \tag{48}$$

und hat eine einfache geometrische Interpretation. Für $i < k$ beschreibt sie die Wahrscheinlichkeit, daß ein Zweig einer bestimmten Ordnung i auf einem Hauptzweig der Ordnung k sitzt, während sie für $i > k$ die Wahrscheinlichkeit beschreibt, daß ein Hauptzweig der Ordnung k Seitenzweige der Ordnung i besitzt. In Bild 24 zeigen wir exemplarisch die Wahrscheinlichkeiten R_{0k} für verschiedene Zweigordnungen k in Abhängigkeit von der Clustermasse. Auch hier können wir deutlich zwischen dem asymptotischen Bereich, in dem R_{0k} gegen einen konstanten Wert strebt, und einem mittleren Bereich unterscheiden. Ein ähnliches Verhalten findet man auch für alle anderen R_{ik}.

In den Bildern 25 und 26 fassen wir in halb-logarithmischer Darstellung die Verzweigungsmatrizen für feste Stamm Ordnungen R_{ki} bzw. feste Zweig Ordnungen R_{ik} zusammen. In beiden Graphen ist $i < k$ und alle Werte sind für eine feste Masse $M = 10^6$ genommen. Die Wahrscheinlichkeiten R_{ik} bzw. R_{ki} hängen ebenfalls exponentiell von den Ordnungen

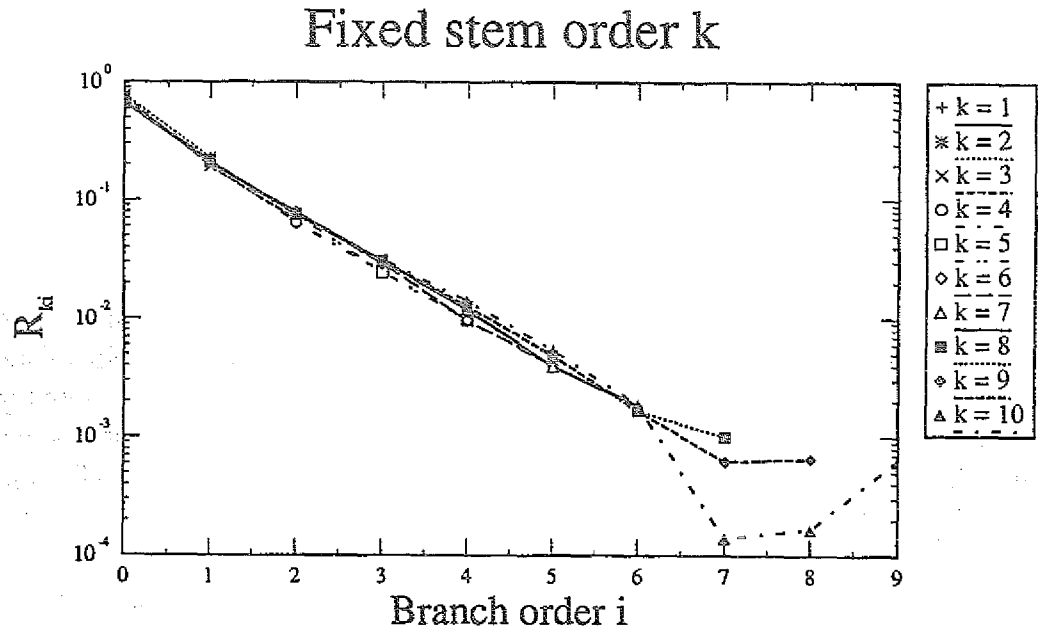


Abbildung 25 Zusammenfassung der R_{ki} für feste Stamm-Ordnung k und Cluster der Masse 10^6. Da die Ordnung eines Seitenzweigs stets kleiner als die Ordnung des entsprechenden Hauptzweigs ist, findet man für $k = 1$ lediglich einen einzelnen Punkt $R_{01} = 1$.

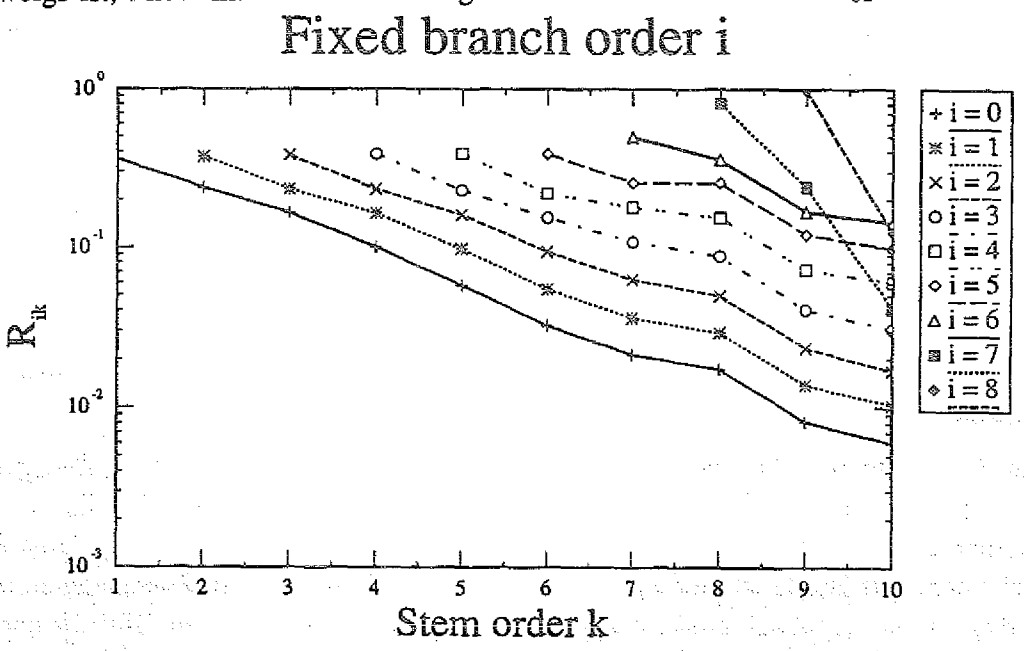


Abbildung 26 Zusammenfassung der R_{ik} für feste Zweigordnung i. Da immer $k > i$ gelten muß, fallen die einzelnen Kurven nicht zusammen.

i bzw. k ab

$$\begin{aligned} R_{ik} &\propto (0.63 \pm 0.02)^{k-i} \text{ bei fester Zweigordnung i} \\ R_{ki} &\propto (0.37 \pm 0.03)^{i} \text{ bei fester Stammordnung k} \end{aligned} \tag{49}$$

Das bedeutet, daß man ein selbstähnliches Verhalten der verschiedenen Zweigordnungen erhält.

Bei DLA auf dem Gitter gab es Untersuchungen der Abhängigkeit der Länge L und der Breite W der Hauptzweige von der Gesamtmasse des Clusters. Man hat gefunden, daß beide mit unterschiedlichen Exponenten skalieren

$$\begin{aligned} L &\propto M^{\nu_\parallel} \\ W &\propto M^{\nu_\perp} \end{aligned} \tag{50}$$

wobei $\nu_\parallel \simeq \frac{2}{3}$ und $\nu_\perp \simeq \frac{1}{2}$ [32] gemessen wurde. Ähnliche Exponenten wurden für Diffusion-Limited-Deposition [102] und bei Dielectric-Breakdown mit $\eta = 1$ in einer Streifen-Geometrie [103] gefunden. Das zeigt, daß diese Wachstumsmodell eher selbst-affine Strukturen mit unterschiedlichen Skalen-Exponenten statt vollkommen selbst-ähnlicher Strukturen generieren. Allerdings haben Simulationen von Noise-reduzierten DLA Clustern auf dem Gitter gezeigt, daß dieses Verhalten nur ein Crossover Effekt ist, der für große Cluster verschwindet. [38]. Hier findet man $\nu_\parallel \simeq \nu_\perp \simeq \frac{2}{3}$.

In allen diesen Simulationen auf dem Gitter existiert jedoch eine ausgezeichnete Wachstumsrichtung, die zu einer starken Anisotropie der Cluster führt, wodurch eine Bestimmung der Hauptzweige sehr einfach wird. Für Off-Lattice DLA existiert eine solche Anisotropie jedoch nicht, so daß bisher noch keine systematische Messung der Breite und Länge einzelner Zweige existiert. Wir haben daher die Breite, Länge und Masse der Zweige einer jeden Ordnung getrennt bestimmt. Diese Größen werden sowohl für einen gesamten Zweig — inklusive aller seiner Seitenzweige — als auch nur für den Backbone eines Zweiges, der lediglich vom nackten Stamm ohne Seitenzweige gebildet wird, gemessen. Länge und Breite werden über einen "skalierten" Trägheitstensor I_{ik} bestimmt, der nicht mehr von der Masse N der Zweige abhängt.

$$I_{ik} = \frac{1}{N}\sum_{j=1}^{N} \delta_{ik} r^2 - x_i x_k. \tag{51}$$

Die Wurzel des jeweils größeren bzw. kleineren Eigenwerts bestimmt die Länge bzw. Breite des Zweigs.

In Bild 27 zeigen wir die Daten für die gemittelte Breite eines gesamten Zweiges einer bestimmten Ordnung in Abhängigkeit von der gesamten Clustermasse. Wieder kann man den asymptotischen Bereich, in dem die Daten konstant werden, von einem mittleren Bereich unterscheiden. Die Breite ist hier exemplarisch gezeigt, da sowohl die Zweiglängen, als auch die Zweigmassen qualitativ exakt das gleiche Verhalten aufweisen. In Bild 28 haben wir daher die Werte für die Länge, Breite und Masse der Zweige für eine gesamte Clustermasse von 10^6 Teilchen in Abhängigkeit von der Zweigordnung bestimmt. Für jede der Größen erhält man wieder ein Verhalten $y_n \propto r_y^n$. Lediglich die Zweige mit der größten Ordnung gehorchen aus den oben genannten Gründen nicht dieser Regel.

Dieses einfache Gesetz bedeutet, daß nicht nur die Längen- und Bifurkations-Verhältnisse, r_L und r_N, sondern auch die Breiten- und Massen-Verhältnisse, r_W und r_M, konstant sind, denn sie werden für jede Größe y_n bestimmt als

$$r_y = \frac{y_{n+1}}{y_n} = \text{const.} \tag{52}$$

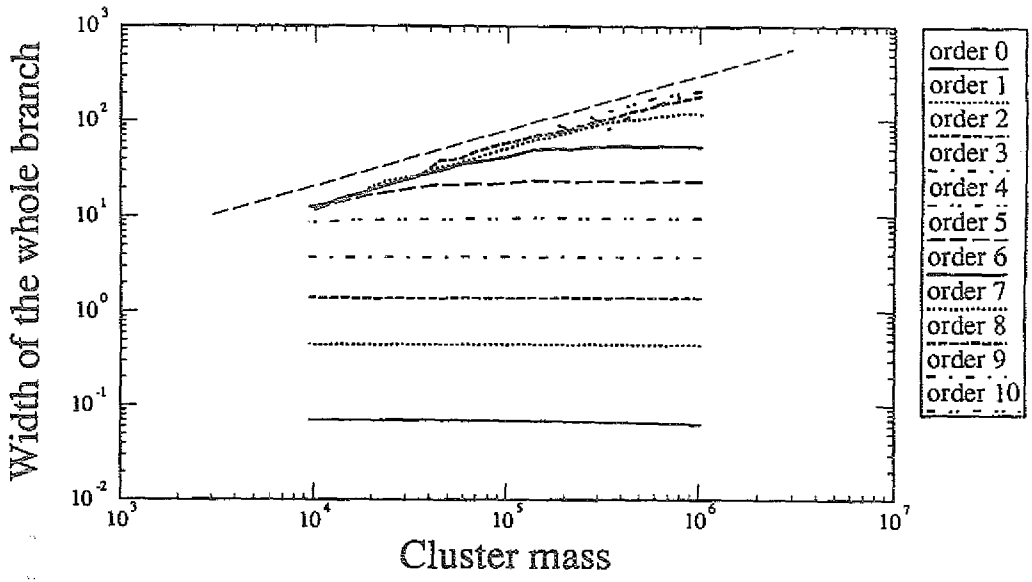


Abbildung 27 Breite der gesamten Zweige als Funktion der Clustermasse. Der asymptotische Bereich entspricht konstanter Breite. Im mittleren Bereich findet man ein Skalenverhalten der jeweils größten Zweigbreite mit einem Exponenten $\beta = 0.585$, welches durch die gestrichelte Linie angedeutet ist. Die Zweigbreite ist hier exemplarisch gezeigt, alle anderen Größen zeigen exakt das gleich Verhalten.

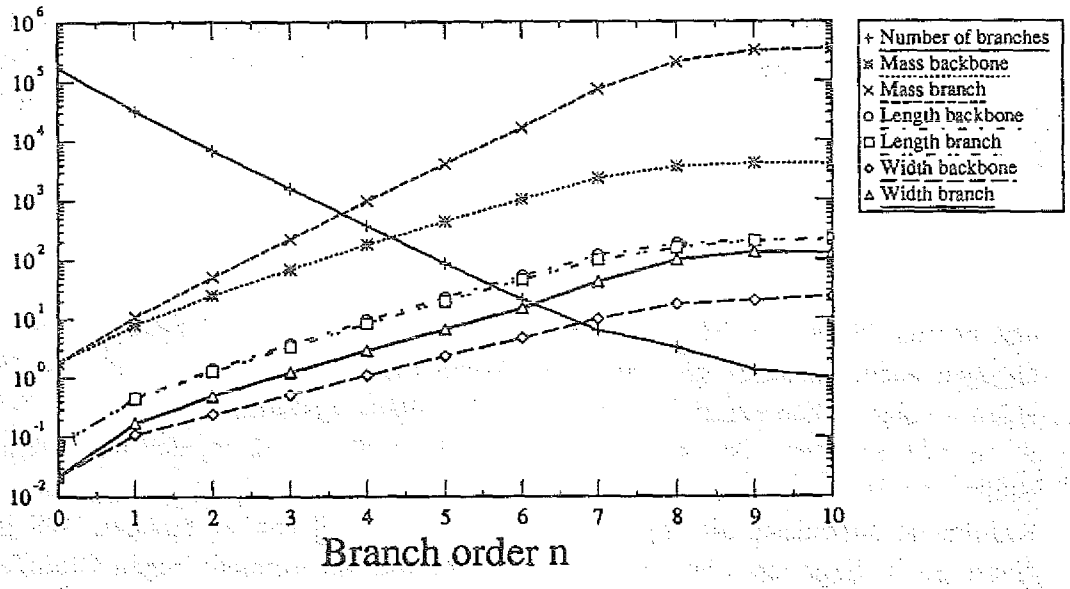


Abbildung 28 Abhängigkeit der Anzahl, Masse, Länge und Breite der Zweige — getrennt nach kompletten Zweigen und Backbones — von der Zweigordnung.

Die verschiedenen Verhältnisse — jeweils für den Backbone und den Gesamtzweig — sind

in der folgenden Tabelle zusammengefaßt:

$$\begin{aligned}
&\text{Backbonemasse: } r_m = 2.46 \pm 0.07 \\
&\text{Backbonelaenge: } r_l = 2.43 \pm 0.07 \\
&\text{Backbonebreite: } r_w = 2.10 \pm 0.08 \\
&\text{Zweigmasse: } r_M = 4.22 \pm 0.04 \\
&\text{Zweiglaenge: } r_L = 2.39 \pm 0.07 \\
&\text{Zweigbreite: } r_W = 2.39 \pm 0.07
\end{aligned} \tag{53}$$

Aus diesen Verhältnissen haben wir verschiedene Skalen-Exponenten bestimmt, die die Länge, Breite und Masse der Zweige verknüpfen. Für die gesamten Zweige erhalten wir:

$$\begin{aligned}
L &\propto M^{\nu_\parallel} \text{ mit } \nu_\parallel \simeq 0.60 \pm 0.04 \\
W &\propto M^{\nu_\perp} \text{ mit } \nu_\perp \simeq 0.60 \pm 0.04 \\
W &\propto L^{\xi} \text{ mit } \xi \simeq 1.00 \pm 0.06
\end{aligned} \tag{54}$$

Man findet also, daß die Länge und Breite der Zweige beide mit demselben Exponenten skalieren, der innerhalb der Fehler mit $\beta = 1/D \simeq 0.585$ — also der fraktalen Dimension des gesamten Clusters [42,43] — übereinstimmt. Das heißt, daß hier die individuellen Zweige selbst-ähnlich und nicht selbst-affin sind.
Die Analyse der Backbones der Zweige liefert folgendes Skalenverhalten:

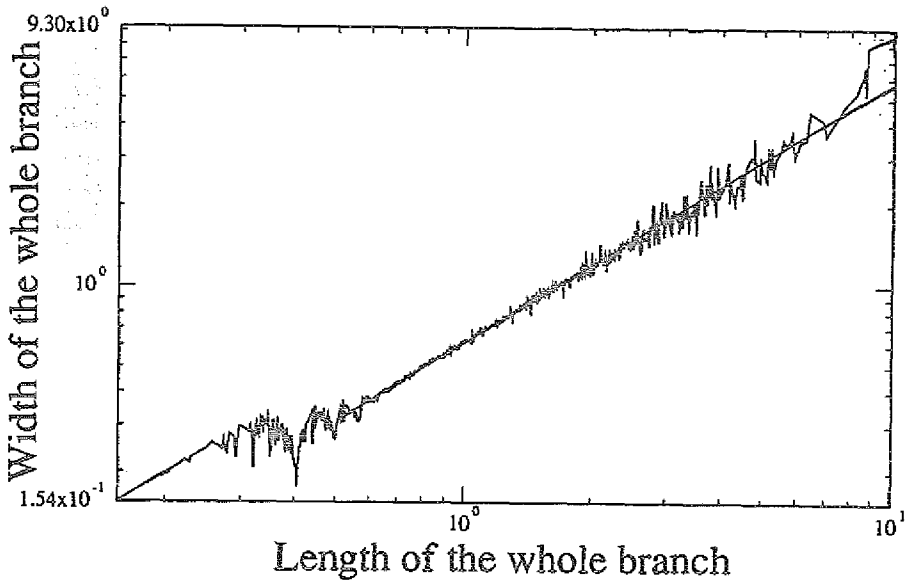
$$\begin{aligned}
l &\propto m^{\nu_\parallel} \text{ mit } \nu_\parallel \simeq 0.98 \pm 0.04 \\
w &\propto m^{\nu_\perp} \text{ mit } \nu_\perp \simeq 0.82 \pm 0.05 \\
w &\propto l^{\xi} \text{ mit } \xi \simeq 0.83 \pm 0.05
\end{aligned} \tag{55}$$

Hier findet man also folgendes Verhalten: Längen und Breiten der Backbones sind noch nicht proportional zueinander. Da die Länge der Backbones proportional zu ihrer Masse ist, sind sie im wesentlichen eindimensionale Objekte. Dies stimmt überein mit einem Exponenten $d_{\text{min}} = 1$, der den "chemischen" l_{C} und den euklidischen l_{E} Abstand zwischen zwischen zwei verschiedenen Cluster Plätzen verknüpft

$$l_c \propto l_E^{d_{\text{min}}}. \tag{56}$$

und bereits früher von Meakin et al. [104] bestimmt wurde. Der Exponent $\nu_\perp$ scheint dagegen außerordentlich groß zu sein. Wenn man z. B. annimmt, daß die Spitze eines jeden Zweiges einen eindimensionalen Random-Walk ausführt, sollte man einen Exponenten $\nu_\perp \simeq 1/2$ erwarten. Allerdings muß man hier berücksichtigen, daß die Daten für Länge, Breite und Masse der Backbones kein exakt exponentielles Verhalten, sondern eine leichte Krümmung aufweisen, die sogar eine weitere Erhöhung von $\nu_\perp$ bewirkt. Wir glauben, daß dieser große Exponent ein Hinweis darauf ist, daß im asymptotischen Grenzfall nicht nur die Länge, sondern auch die Breite der Backbones proportional zur Backbonemasse ist: $\nu_\parallel = \nu_\perp \simeq 1$.
Wie man deutlich in Bild 27 erkennt, gehorcht die Breite der Zweige der jeweils größten Ordnung einem Potenzgesetz

$$w \propto m^{\beta}. \tag{57}$$



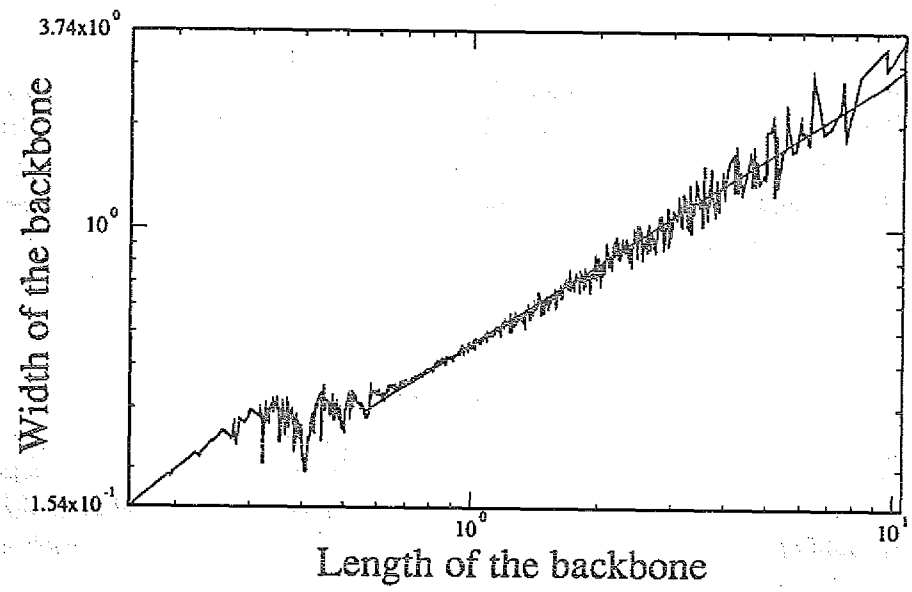


Abbildung 29 Direktes Skalenverhalten von Länge und Breite der Zweige. a) zeigt die Daten für den gesamten Zweig, während b) die Daten für die Backbones zeigt. Die Plateaus bei -1.00 sind ein Resultat der Binning Methode. Die systematische Krümmung der Daten für die Backbones ließe auch einen Exponenten $\xi = 1$ zu.

welches durch die gestrichelte Linie angedeutet ist. Das gleiche Verhalten findet man auch für die Masse, Länge und Breite der Backbones und für die Länge und Breite der gesamten Zweige. Eine Ausnahme bildet lediglich die Masse der gesamten Zweige: Statt eines Exponenten β findet man hier, daß die Masse der größten Zweige direkt proportional zur Masse des gesamten Clusters ist. Wir möchten hier betonen, daß der Exponent wieder durch $\beta = 1/D_f$ gegeben ist, d. h. auch die geometrischen Eigenschaften der größten Zweige skalieren mit der fraktalen Dimension von DLA.

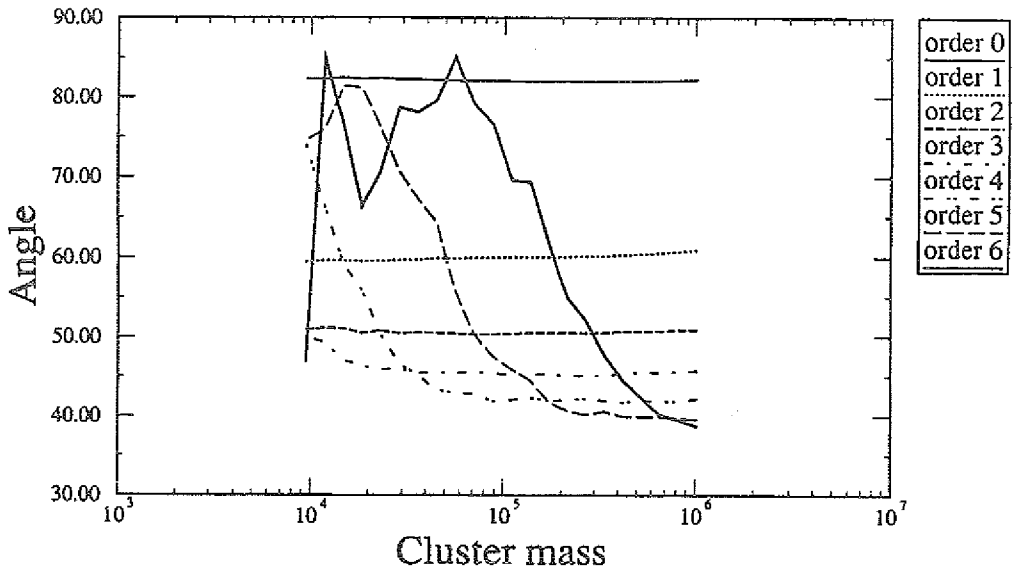


Abbildung 30 Mittlerer Winkel zwischen Haupt- und Seitenzweigen für verschieden Zweigordnungen.

Um unsere Resultate zu überprüfen, vergleichen wir nun die Längen und Breiten einzelner Zweige direkt miteinander, ohne die jeweiligen Zweigordnungen zu betrachten. In Bild 29 zeigen wir die Breite in Abhängigkeit von der Zweiglänge sowohl für die gesamten Zweige, als auch für die Backbones. Für kleine Zweiggrößen findet man ein kleines Plateau mit starken Fluktuationen, welches ein Artefakt der benutzten Binning Methode ist. Für mittlere Zweiggrößen findet man allerdings ein Skalenverhalten $w \propto l^{\xi}$ mit Exponenten

$$\begin{aligned} \xi &= 0.79 \pm 0.01 \text{ bei den Backbones} \\ \xi &= 0.98 \pm 0.01 \text{ bei den gesamten Zweigen} \end{aligned} \tag{58}$$

Dies ist in guter Übereinstimmung mit den vorhin erhaltenen Resultaten. Allerdings zeigen die Daten für die Breite der Backbones wieder eine Abweichung vom exakten Skalenverhalten, so daß wieder ein größerer Exponent als der gemessene möglich ist. Im Bereich großer Zweige ist sogar ein Exponent Eins möglich, was mit der oben vermuteten Selbstähnlichkeit der Backbones übereinstimmt.

5.3 Winkel

Die Messung des Trägheitstensors liefert neben Länge und Breite auch Informationen über die Richtung der Zweige und damit über die Winkel zwischen Zweigen verschiedener Ordnung. Wir haben daher für jeden Zweig der Ordnung n den Winkel gemessen, den dieser mit seinem Hauptzweig einschließt. Die gemittelten Winkel φ_n sind in Bild 30 in Abhängigkeit von der Zweigordnung gezeigt. Deren Analyse liefert im asymptotischen Bereich einen exponentiellen Abfall zu einem Grenzwinkel von $\varphi_\infty = 38 \pm 3$ Grad

$$\varphi_n - \varphi_\infty \propto (0.53 \pm 0.01)^n. \tag{59}$$

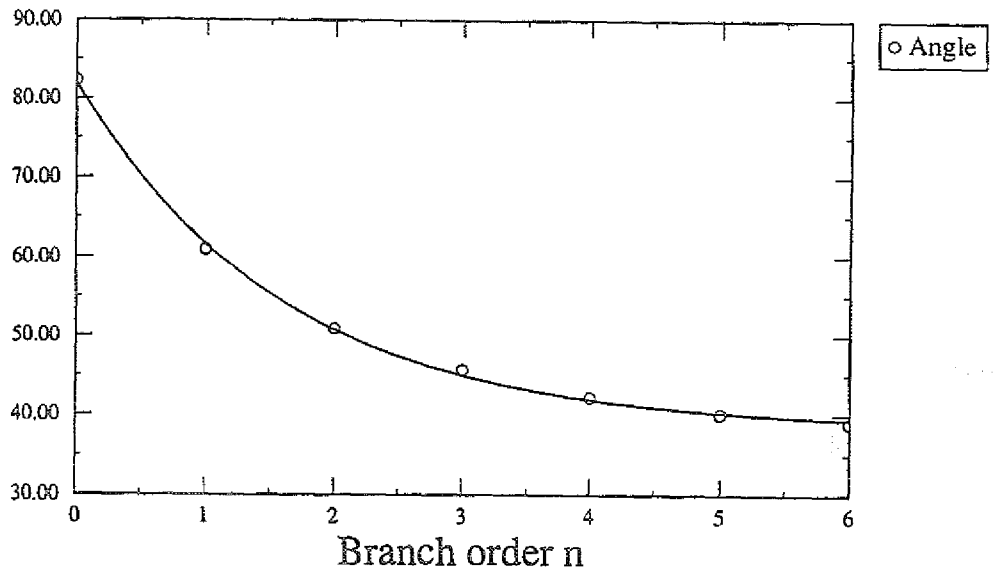


Abbildung 31 Analyse des Winkels φ_n zwischen Seitenzweigen gegebener Ordnung n und deren Hauptzweigen im asymptotischen Bereich. Die durchgezogene Linie entspricht $\varphi_n - 38.4 \propto 0.53^n$.

Dieser Abfall ist in Bild 31 gezeigt.

Darüberhinaus findet man den folgenden globalen Trend: die Blätter eines jeden Zweiges bilden ungefähr einen rechten Winkel mit dem jeweiligen Hauptast, da $\phi_0 \approx 90°$. Eine Abnahme der Winkel bedeutet, daß die einzelnen Zweige mehr und mehr parallel werden, bis der Winkel bei ungefähr 38° sättigt. Es gab Spekulationen, ob man im asymptotischen Grenzfall unendlicher Clustermasse möglicherweise einen Übergang in ein radiales Wachstum finden könnte, was einem Grenzwinkel von 0° entspricht [105]: Alle Zweige wachsen parallel. Unsere Daten widersprechen jedoch einer solchen Vermutung.

Trotz allem befinden sich alle Winkel auf jeden Fall in einem Bereich zwischen 0° und 90°, was auf jeden Fall einem gerichteten Wachstum entspricht: Seiten- und deren Hauptzweige wachsen im wesentlichen in die gleiche Richtung. Diese Beobachtung unterstützt ein Modell für DLA von Mandelbrot und Vicsek [58], in dem ein gerichtetes Fraktal zugrunde gelegt wird. Wir glauben daher, daß eine Beschreibung von DLA ähnlich zu der eines gerichteten Fraktals am erfolgversprechendsten ist.

Im folgenden möchten wir zwei weitere Untersuchungen, die sich mit der Frage eines ausgezeichneten Winkels und einer dadurch bedingten Symmetriebrechung beschäftigen, diskutieren.

Derrida und Hakim [106] studieren ein Modell eines N-armigen Sterns, in dem N Nadeln — die alle den gleichen Winkel $\theta = 2\pi/N$ einschließen — von einem gemeinsame Ursprung ausgehend radial nach außen wachsen (vgl. Abbildung 32). Sie lösen analytisch die Laplace Gleichung $\Delta\Phi = 0$ mit den entsprechenden Randbedingungen — $\Phi = 0$ auf dem Stern und $\Phi(r) \propto \log r$ für $r \to \infty$ — mittels konformer Transformation. Mit einer linearen Stabilitätsanalyse finden sie, daß für $N < 6$ kleine periodische Störungen der Amplitude ε

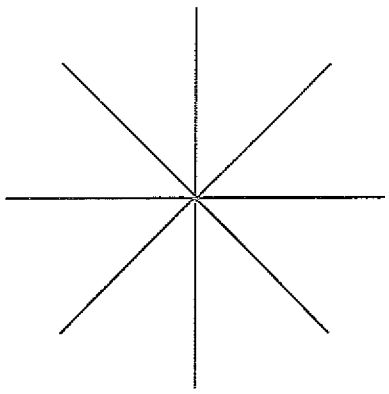


Abbildung 32 Schematische Darstellung eines in [106] benutzten N-armigen Sterns. In dieser Abbildung ($N = 8$) haben alle Arme die gleiche Länge L. In der linearen Stabilitätsanalyse werden diese Längen leicht gestört $L \to L + \varepsilon$.

der Zweiglängen gedämpft werden

$$\frac{d\varepsilon}{dt} \propto -\varepsilon \tag{60}$$

während für $N > 6$ diese Geometrie instabil ist und man eine spontane Brechung der Symmetrie findet: Einige Zweige beginnen, wesentlich stärker als andere zu wachsen. Der Fall $N = 6$ stellt gerade einen marginalen Fall dar, in dem Geometrien mit unterschiedlichen Nadellängen stabil sein können. Das bedeutet aber, daß der Fall $N = 5$ die stabile Geometrie mit den meisten Seitenarmen ist und daß hier der Winkel zwischen den Armen gerade $\theta = 2\pi/5 \approx 2 \cdot 36°$ beträgt. Es ist aber nicht klar, ob ein derartiges Modell zur Erklärung der hier gefundenen 38° Winkel benutzt werden kann, da dieses Modell erstens sehr idealisiert ist und zweitens keinerlei Unordnung enthält, die üblicherweise sehr stark die Modelleigenschaften beeinflußt.

Neben diesem Modell kommen Arneodo et al. [107] sogar zu dem Resultat, direkte Hinweise auf eine fünfzählige Symmetrie in Off-Lattice DLA gefunden zu haben. Dies stützt sich auf drei Beobachtungen, die erst dargestellt und dann kommentiert werden sollen.

1. Eine Wavelet-Transformation [108] wird dazu benutzt, systematisch die feinen Details großer Off-Lattice DLA Cluster zu 'verschmieren', damit die groben Strukturen und die Hauptzweige deutlich sichtbar werden [109]. An dieser Struktur wird die Verteilung der Winkel zwischen benachbarten Zweigen gemessen. Man erhält eine breite Verteilung mit einem Mittelwert bei $\theta \approx 36° = 2\pi/10$. Dieses Resultat stimmt mit der von uns gemachten Beobachtung überein. Nun wird geschlossen, daß sich dieser lokale Winkel global als Anisotropie bemerkbar macht. Das bestimmte Verteilungsmaximum von 36° hat den gleichen Ursprung, wie die von uns gemessenen Winkel. Allerdings kann man aus der lokalen Anisotropie keine Schlußfolgerungen über das globale Verhalten ziehen kann, denn das globale Verhalten wir sehr stark davon beeinflußt, daß alle großen Zweige von einem gemeinsamen Zentrum radial nach außen wachsen und daß die Random-Walker isotrop aus dem Unendlichen herandiffundieren. Zusätzlich muß bezweifelt werden, daß eine derartige Anisotropie die von den Random-Walkern induzierte Unordnung überdeckt und die globale Struktur dominiert.
2. Bei der Simulation von Saffmann-Taylor Fingern durch Lattice-DLA in einer Streifen Geometrie findet man neben dem dominierenden Finger noch einen abgeschirmten Untergrund, dessen Einhüllende nach einer Mittelung über viele Simulationen näherungsweise

einen Winkel $\theta \approx 72° = 2\pi/5$ mit der Hauptwachstumsrichtung einschließt. Hieraus wird auf einen charakteristischen Abschirmwinkel für Laplacesches Wachstum geschlossen, was sich wiederum als fünf-zählige Symmetrie niederschlägt. Bei diesem Resultat muß man allerdings beachten, daß es gewonnen wird durch Überlagerung vieler auf dem Gitter erzeugter Cluster. Mit exakt der gleichen Methode — allerdings nicht in einer Streifen-Geometrie — zeigten Meakin et al. [32] überzeugend, daß man für Lattice-DLA bereits an kleinen Clustern die vom Gitter hervorgerufene Anisotropie nachweisen kann. Daher ist es nicht überzeugend, wenn ein ähnliches Argument hier benutzt wird, eine fünfzählige Geometrie in Off-Lattice DLA zu begründen.

3. An einer Wavelet Transformation großer DLA Cluster findet man, daß die einzelnen Zweigordnungen im Mittel einer Fibonacci-Vorschrift gehorchen: Aus jedem Verzweigungspunkt wachsen zwei Zweigarten heraus, die A- und B-Zweige genannt werden. Während die A-Zweige instabil sind und sich spalten, sind die B-Zweige zunächst stabil und spalten sich erst später. Dies kann man formell durch die Abbildung

$$A \to AB, \quad B \to A \tag{61}$$

beschreiben, was nichts anderes als eine Fibonacci-Abbildung ist. Von der Fibonacci-Reihe

$$F_n = F_{n-1} + F_{n-2}, \quad F_0 = F_1 = 1 \tag{62}$$

weiß man aber, daß das Verhältnis aufeinanderfolgender Werte gegen den "Goldenen Schnitt" konvergiert

$$\lim_{n\to\infty} \frac{F_n}{F_{n-1}} = 2\cos(\pi/5) \simeq 1.618 \tag{63}$$

welcher mit den Abmessungen eines Pentagons in Verbindung steht. Aus der Verbindung von DLA und der Fibonacci-Reihe wird wieder auf die fünf-zählige Symmetrie geschlossen. Hier muß man mehrere Punkte entgegenhalten: Die Verbindung, die zwischen DLA und der Fibonacci-Reihe gezogen wird, ist sehr vom subjektiven Standpunkt abhängig und kann nicht quantisiert oder systematisiert werden. Außerdem basiert die Argumentation auf einer Messung der fraktalen Dimension [110], die zumindest umstritten ist [111].

Abschließend möchte ich die eben beschriebenen Resultate so bewerten, daß man eine Instabilität der Laplace Gleichung beobachtet, die zu einer Symmetriebrechung führen kann, daß aber die numerischen Hinweise darauf keinesfalls überzeugend sind. Wir glauben daher, daß die hier gemessenen Winkel nicht als Nachweis für eine mögliche fünfzählige Symmetrie benutzt werden können. Außerdem ergeben umfangreiche numerische Simulationen (vgl. Kapitel 3) keinerlei Hinweis auf ein derartiges Verhalten.

5.4 Stabilität der Definition der Ordnungen

Die hier benutzte Definition der Zweigordnungen weicht leicht von der Definition ab, die in einer anderen Arbeit benutzt wird [99]. Um unsere Ergebnisse direkt vergleichen zu können und um die Stabilität der Ergebnisse gegenüber Änderungen der Definition zu prüfen, haben wir weitere 62 Cluster mit jeweils 10^6 Teilchen generiert und analysiert. Dabei haben wir die Definition der Zweigordnung leicht verändert: Wenn sich mehrere Zweige treffen wird nicht notwendigerweise der längste Zweig als Hauptzweig angesehen, sondern derjenige mit der bis

dahin höchsten Ordnung. Lediglich wenn Zweige der gleichen Ordnung n aufeinandertreffen, wird der längere als Hauptzweig angesehen und erhält die Zweigordnung $n+1$. Mit dieser Definition, die mit der in [99] benutzten übereinstimmt, erhalten wir keine implizite Sortierung der Zweigordnungen nach Zweiglängen, wie in der ersten Definition.

Die hiermit erzielten Resultate unterscheiden sich nur quantitativ von den vorher erhaltenen. Das führt im wesentlichen nur zu veränderte Größenverhältnissen, die in der folgenden Tabelle zusammengefaßt sind.

$$\begin{aligned}
&\text{Anzahl von Zweigen: } r_N = 0.194 \pm 0.002\\
&\text{Backbonemasse: } r_m = 3.00 \pm 0.06\\
&\text{Backbonelaenge: } r_l = 2.94 \pm 0.06\\
&\text{Backbonebreite: } r_w = 2.41 \pm 0.01\\
&\text{Zweigmasse: } r_M = 5.36 \pm 0.03\\
&\text{Zweiglaenge: } r_L = 2.86 \pm 0.05\\
&\text{Zweigbreite: } r_W = 2.80 \pm 0.06
\end{aligned} \tag{64}$$

Das Branching- und Längenverhältnis r_N und r_L stimmen nun mit den Werten in [99] überein. Der Unterschied zwischen den beiden Definitionen beträgt ungefähr 10–20 Prozent

Wir berechnen mit diesen Verhältnissen nun wieder die entsprechenden Exponenten, die Längen, Breiten und Massen verknüpfen. Für die gesamten Zweige finden wir:

$$\begin{aligned}
L &\propto M^{\nu_\parallel} \text{ mit } \nu_\parallel \simeq 0.63 \pm 0.01\\
W &\propto M^{\nu_\perp} \text{ mit } \nu_\perp \simeq 0.61 \pm 0.01
\end{aligned} \tag{65}$$

und für die Backbones erhalten wir:

$$\begin{aligned}
l &\propto m^{\nu_\parallel} \text{ mit } \nu_\parallel \simeq 0.98 \pm 0.02\\
w &\propto m^{\nu_\perp} \text{ mit } \nu_\perp \simeq 0.80 \pm 0.03\\
w &\propto l^{\xi} \text{ mit } \xi \simeq 0.81 \pm 0.05
\end{aligned} \tag{66}$$

Obwohl also die erhaltenen Verhältnisse unterschiedlich sind, ändert dies nichts an dem Skalenverhalten der Längen und Breiten. Das deutet an, daß diese Skalenrelationen unabhängig von den Details der Definition sind.

5.5 Zusammenfassung

In diesem Kapitel haben wir die Baum-Struktur von Off-Lattice DLA Clustern untersucht. Wir haben gezeigt, daß nicht nur die Bifurkations- und Längen-Verhältnisse r_N und r_L konstant sind, sondern auch die Massen- und Breiten-Verhältnisse r_W und r_M. Mit Hilfe dieser Ergebnisse haben wir das Skalenverhalten der Längen und Breiten der Zweige betrachtet und gefunden, daß in Off-Lattice DLA die Exponenten $\nu_\parallel$ und $\nu_\perp$ übereinstimmen. Durch Messung der Verzweigungsmatrix R_{ik} haben wir das Wechselspiel zwischen Haupt- und Nebenzweigen untersucht und gezeigt, daß auch R_{ik} einfachen Exponential-Gesetzen gehorcht. Weiter haben wir die Winkel zwischen Haupt- und Nebenzweigen gemessen und Hinweise dafür gefunden, daß diese Winkel gegen einen asymptotischen Wert von ca. 38° streben. Alle diese Beobachtungen deuten darauf hin, daß man bei Off-Lattice DLA für den asymptotischen Grenzfall unendlicher Clustermasse einen Crossover in eine selbst-ähnliche Struktur erhält.

Kapitel 6

Schockwellen auf Gittern[+]

Wir untersuchen numerisch das Wachstum von Rissen in elastischem Materialien unter dem Einfluß einer wandernden Schockwelle. Wir beschreiben die Implementierung eines schnellen Algorithmus, der perfekt auf Daten-Parallele Multiprozessor Computer zugeschnitten ist. Mit Hilfe von umfangreichen Simulationen auf der Connection Machine CM2 generieren wir Risse mit mehr als 10000 gebrochenen Bonds auf einem 1024×1024 Gitter. Wir zeigen, daß die resultierenden Strukturen fraktal sind mit einer Dimension, die je nach Wahl des Bruchkriteriums zwischen 1. und 2. liegt

6.1 Einleitung

Wie bricht ein Festkörper unter einer angelegten Last? Das ist in vielen Bereichen eine wichtige Frage und hat starke Einflüsse z. B. in Materialforschung, Ingenieurwesen oder Geophysik. Sie wird seit langer Zeit extensiv sowohl mit numerischen, experimentellen und analytischen Methoden studiert [63–65]

Wenn man den Festkörper als linear-elastisches Medium ansieht, kann die Formulierung dieser Frage mit Hilfe der Lamé-Gleichung ausgedrückt werden

$$(1-2\nu)\nabla^2\vec{u} + \nabla\nabla\cdot\vec{u} = 0 \tag{67}$$

welche die Verschiebung $\vec{u}$ eines kleinen Volumen-Elements von seiner Gleichgewichtsposition beschreibt. Die Poisson-Zahl ν ist eine Material-Konstante, welche folgende Bedeutung besitzt: Wenn man eine uniaxiale Kraft auf einen elastischen Balken der Länge L und Querschnitt $W \times W$ anwendet, wird sich nicht nur dir Länge um ΔL in Richtung der Kraft, sondern auch die Breite um ΔW senkrecht zur Kraftrichtung ändern. Die Poisson-Zahl erhält man dann als Verhältnis der relativen Längenänderungen

$$\nu = -\frac{\Delta W/W}{\Delta L/L}. \tag{68}$$

Aufgrund thermodynamischer Argumente muß ν in dem Bereich $-1 \leq \nu < 0.5$ liegen. Beton hat zum Beispiel eine Poisson-Zahl $\nu \approx 0.2$.

[+] Wird veröffentlicht in Int. J. Mod. Phys. C [112]

Ein Riß kann dann in einem solchen System als zusätzliche kraftfreie Oberfläche angesehen werden, was mit Hilfe des Spannungstensors $\underline{\underline{\sigma}}$ ausgedrückt wird als zusätzliche Randbedingung

$$\underline{\underline{\sigma}} \cdot \vec{n} = 0 \tag{69}$$

wobei $\vec{n}$ ein Oberflächenvektor ist. Ein Riß wächst nun, wenn die Spannung parallel zur Rißoberfläche $\sigma_{\|}$ eine bestimmte, materialabhängige Festigkeit σ_C überschreitet. Da kein first-principles Argument für die Wachstumsgeschwindigkeit bekannt ist, nimmt man oft heuristisch folgendes Verhalten der Wachstumsgeschwindigkeit an

$$v_n \propto \left(\sigma_{\|} - \sigma_C\right)^{\eta} \tag{70}$$

wobei η oft einfach auf Eins gesetzt wird. Diese Gleichungen (79), (80) und (70) formulieren das Rißwachstum als nicht-lineares Moving-Boundary Problem, welches wesentlich schwieriger zu lösen ist, als die Lamé-Gleichung bei vorgegebenen Randbedingungen selbst. Allerdings fehlt hier immer noch ein wesentlicher Faktor, welcher im folgenden einfach als "Unordnung" beschrieben werden soll. Mikroskopisch sind damit Abweichungen von der perfekten Gitterstruktur gemeint, wie Leerstellen, Versetzungen oder Korn-Grenzen. Auf makroskopischer Längenskala werden diese Fehlstellen einfach auf eine räumliche, stochastische Verteilung der Material-Parameter wie Poisson-Zahl oder Festigkeit reduziert.

In numerischen Simulationen des Rißwachstums diskretisiert man oft das elastische Medium auf einem Gitter und löst die entstandenen Gleichungen numerisch [113,76,78]. Entsprechend den gewählten Randbedingungen und den angelegten Kräften relaxiert man das System ins Gleichgewicht. Danach wird ein oder mehrere Bonds entsprechend der gewählten Bruchregel — zum Beispiel der Bond mit der größten Last — ausgewählt und gebrochen. Dadurch entsteht ein neuer Rand, der wieder eine Relaxation ins Gleichgewicht und damit eine komplette Neulösung des Problems erfordert. Dieser Prozeß wird iteriert, bis ein Riß der gewünschten Größe generiert ist. Die Unordnung wird in solchen Simulationen implementiert, indem die elastischen Parameter oder die Festigkeit der einzelnen Bonds variiert werden. Da man mit dieser Methode mit jedem Zeitschritt immer nur zu Gleichgewichtspositionen gelangt, ist eine solche Methode nur in der Lage, quasi-statische Prozesse oder "Slow-Cracking" zu beschreiben.

Allerdings gibt es andererseits Phänomene — wie Explosionen oder Schockwellen — welche nicht in diesem quasi-statischen Grenzfall betrachtet werden sollten und in welchen die Relaxationszeit ins Gleichgewicht vergleichbar mit oder länger als die Wachstumsgeschwindigkeit eines Risses ist. Dies wirft das Problem der sehr unterschiedlichen Zeitskalen auf, die im Rißwachstum auftreten können.

Zunächst muß man beachten, daß Viskosität ein Phänomen ist, daß von einer charakteristischen Zeitskala beherrscht wird. Das wird ganz deutlich in den Versuchen von Van Damme et al. [68]: Je nach Druck des Wassers, welches in den Ton hineingepreßt wird, verhält sich dieser wie eine viskose Flüssigkeit oder wie ein Festkörper. Außerdem sei erwähnt, daß in Geophysikalischen Modellen auf hinreichend großer Zeitskala auch die Erdkruste als Material mit endlicher Viskosität betrachtet wird.

Daher ist es offensichtlich wichtig, auf welchen Zeitskalen man ein Rißphänomen studieren will. Bei einer schockartigen Anregung — z. B. wenn man mit einem Hammer auf eine Glasplatte schlägt — wird dem elastischen System wesentlich mehr Energie zugeführt, als für das Wachstum eines Risses benötigt wird. Die überschüssige Energie geht teilweise in die

kinetische Energie des propagierenden Risses, der sich deshalb sehr schnell forwärtsbewegt. Allerdings muß man beachten, daß beim Rißwachstum selbst elastische Energie frei wird, die hauptsächlich durch Rayleigh-Wellen entlang der Rißoberfläche zur Rißspitze transportiert wird. Diese Wellen haben eine endliche Ausbreitungsgeschwindigkeit $v \approx (1.5-4)\times 10^3 m/s$. Wenn die Wachstumsgeschwindigkeit des Risses diese Geschwindigkeit überschreitet, kann die freiwerdende Energie die Rißspitze nicht mehr erreichen und akkumuliert an den Seiten des Risses. Diese akkumulierte Energie führt dazu, das weitere Rißoberfläche gebildet wird und Seitenzweige entstehen, die üblicherweise kurz sind und daher "Micro-Zweige" genannt werden. Wenn die Rißgeschwindigkeit den Wert $\sqrt{2}v$ überschreitet, kann sich die Rißspitze spontan spalten [114] und zwei Risse propagieren gleichzeitig ("Macro-Branching"). Solch eine Situation resultiert also in Rißstrukturen mit vielen Seitenzweigen, die auch hinter der Rißspitze entstehen, weil die freiwerdende Energie nicht schnell genug dissipiert werden kann [114,115]. Im folgenden wollen wir numerisch solch einen Prozeß, in dem man offensichtlich die dynamischen Aspekte berücksichtigen muß, untersuchen.

Die Einteilung dieses Kapitels ist daher die folgende: Im Abschnitt 6.2 wollen wir das benutzte Modell vorstellen, dessen Realisierung auf der Connection Machine im Abschnitt 6.3 dargestellt wird. Im Abschnitt 6.4 beschreiben wir Details der Simulationen und im Abschnitt 6.5 werden wir die Ergebnisse darstellen.

6.2 Das Modell

Als Modell des elastischen Materials betrachten wir ein Dreiecksgitter masseloser Hookescher Federn, welche Massenpunkte der Masse m verbinden. Ein Dreiecksgitter ist hier für die Simulation notwendig, da ein einfaches Quadratgitter kein Scher-Modul besitzt und damit beliebig unter Scherung verformt werden kann. Dieses Modell ist als "Central-Force" Modell bekannt [76]. Die Ränder dieses Gitters werden festgehalten. Auf jedem Gitterplatz hat man zwei kontinuierliche Freiheitsgrade, die Verschiebungen u_x und u_y des Massepunktes von der Gleichgewichtsposition $\vec{r}_0$. Da wir das dynamische Verhalten dieses Gitters studieren wollen, benötigen wir die Zeitentwicklung der Verschiebungen, welche wir durch Newton's Gleichung erhalten. Entsprechend einem Vorschlag von Chopard [116,117] benutzen wir einen diskreten Hamilton-Formalismus, um die Dynamik auszudrücken. Der Hamiltonian dieses Systems ist gegeben durch

$$H(\vec{p}_1 \dots \vec{p}_N, \vec{r}_1 \dots \vec{r}_N) = \sum_{i=1}^{N} \frac{\vec{p}_i^2}{2m_i} + \frac{1}{2}\sum_{i,j=1}^{N} U_{ij}(\vec{r}_i, \vec{r}_j) \tag{71}$$

wobei $\vec{p}_i$ und $\vec{r}_i$ Ort und Impuls des Gitterplatzes i sind und U_{ij} die elastische Wechselwirkung zwischen den Nächsten-Nachbar-Plätzen i und j beschreibt. Da wir Hookesche Federn haben, ist

$$\begin{aligned} U_{ij} &= \frac{k_{ij}}{2}(|\vec{r}_i - \vec{r}_j| - a)^2 = \frac{k_{ij}}{2}(|\vec{u}_i + \vec{r}_{0i} - \vec{u}_j - \vec{r}_{0j}| - a)^2 \\ &= \frac{k_{ij}}{2}\left(\left|\vec{u}_i - \vec{u}_j + \vec{dr}_{ij}\right| - a\right)^2 \end{aligned} \tag{72}$$

mit Kopplungskonstanten k_{ij} zwischen den Plätzen i und j. $\vec{dr}_{ij} = \vec{r}_{0i} - \vec{r}_{0j}$ ist der Gittervektor zwischen Gleichgewichtspositionen und a ist die Gleichgewichtslänge der verbindenden Federn (man beachte, daß $\vec{dr}_{ij}$ hier keine Differentiale, sondern den Differenzenvektor bedeutet). In unseren Simulationen werden wir den Fall $a = 0$ betrachten, während wir $\vec{dr}_{ij} = 1$

festhalten. Das entspricht dem Fall eines vorgespannten Netzwerkes, wie z. B. einem gespannten Fell über einer Trommel.

Die diskretisierten Hamiltonschen Gleichungen sind dann

$$\begin{aligned} \vec{r}_i(t+1) - \vec{r}_i(t) &= \frac{\partial H}{\partial \vec{p}_i} = \frac{\vec{p}_i(t)}{m_i} \\ \vec{p}_i(t) - \vec{p}_i(t-1) &= -\frac{\partial H}{\partial \vec{r}_i} \end{aligned} \tag{73}$$

wobei wir vektorielle Ableitungen als Symbole für die entsprechen Gradienten benutzen. Die Zeitentwicklung des Verschiebungsfeldes kann schließlich geschrieben werden als

$$\begin{aligned} \vec{u}_i(t+1) - 2\vec{u}_i(t) + \vec{u}_i(t-1) &= -\frac{1}{m_i}\frac{\partial H}{\partial \vec{u}_i(t)} \\ &= \sum_{j=NN(i)} k_{ij} \cdot \vec{\delta}_{ij} \cdot \left(1 - a/\left|\vec{\delta}_{ij}\right|\right) \end{aligned} \tag{74}$$

mit $\vec{\delta}_{ij} \equiv \vec{u}_i(t) - \vec{u}_j(t) + \vec{dr}_{ij}$. Diese Gleichung definiert eine einfache Update-Regel, welche die Verschiebungen zur Zeit $t+1$ mit den Verschiebungen bei t und $t-1$ verbindet. Obwohl diese gewählte Dynamik recht grob im Vergleich zu z. B. Molekular Dynamik erscheint, kann man tatsächlich zeigen, daß sie den Gesamtimpuls und eine Art Gesamt-Energie erhält. Darüber hinaus erhält sie alle Eigenschaften der Schockwelle, die wir zur Generierung von Rissen benötigen: Ein aufgeprägtes Wellenpaket wandert mit nur kleinen Formänderungen mit einer definierten Geschwindigkeit durch das Gitter. Außerdem gestattet sie, große Rißstrukturen relativ schnell zu simulieren.

6.3 Das Modell auf der CM

Da die Update-Regel (74) gleichzeitig in jedem Zeitschritt auf alle Gitterplätze angewendet wird, und da die Topologie des zugrundeliegenden Gitters — im Gegensatz zur Molekular Dynamik — nicht durch Teilchen Umordnung verändert wird, bietet eine single-instruction-multiple-data (SIMD) Maschine wie die Connection Machine eine hervorragende Plattform zur Implementierung dieses Problems: Jeder Gitterplatz wird auf einen virtuellen Prozessor abgebildet und alle Gitterplätze werden parallel neu berechnet. Inter-Prozessor Kommunikation wird lediglich für die Kraftberechnung in (74) benötigt. Da jedoch ein Dreiecksgitter auf ein Quadrat-Gitter mit Übernächsten-Nachbar Verbindungen abgebildet werden kann, ist eine einfache Gitterkommunikation mit Hilfe von `CSHIFT` Kommandos ausreichend und man benötigt keine langsame Router-Kommunikation, was dieses Gitter-Modell schnell und effizient macht.

In der tatsächlichen Implementierung benutzen wir die folgende Datenstruktur der Haupt-Variablen: Jeder Prozessor speichert die Verschiebungen zur Zeit t und $t-1$, `UT` und `UTM1`, welche beide als komplexe Zahlen repräsentiert werden. Da wir ein ungeordnetes System mit quenched disorder simulieren wollen, weisen wir jedem Bond eine unterschiedliche, zufällige Kopplungskonstante zu. Daher müssen wir auf jedem Gitterplatz die sechs Kopplungen `K` zu allen Nachbarn speichern. Dadurch benutzen wir zwar zu viel Speicher, da

jedes k_{ij} zweimal gespeichert wird – am Platz i und am Platz j – allerdings sparen wir so unnötige Kommunikation. Die Datenstruktur der Hauptvariablen ist daher

```
      COMPLEX , ARRAY (NXY, NXY)     :: UT, UTM1
      REAL    , ARRAY (Z, NXY, NXY) :: K
CMF$  LAYOUT UT(:NEWS,:NEWS)
CMF$  LAYOUT UTM1(:NEWS,:NEWS)
CMF$  LAYOUT K(:SERIAL,:NEWS,:NEWS)
```

Die LAYOUT Direktive für K wird hier benutzt, um die Kopplungen eines Gitterplatzes auf einer seriellen Dimension auf einem einzigen Prozessor zu speichern. Das ist notwendig, da sonst der Compiler alle Kopplungen auf verschiedenen Prozessoren verteilt, was zu unnötiger Kommunikation führt. Mit diesem Layout kann ein Schritt von (74) elegant in CM FORTRAN implementiert werden

```
C     GET VECTORS TO NEIGHBORS
C     COMPUTE NEW PARTICLE POSITION AND MOMENTUM
C     NN is a help field to store r_i-r_j. It has the same layout as
C     ut and utm1.
      NN(1,:,:) = CSHIFT(UT,1,1)
      NN(2,:,:) = CSHIFT(UT,2,1)
      NN(4,:,:) = CSHIFT(UT,1,-1)            !get displacements
      NN(5,:,:) = CSHIFT(UT,2,-1)            !of nearest neighbors
      NN(3,:,:) = CSHIFT(NN(4,:,:),2,1)
      NN(6,:,:) = CSHIFT(NN(1,:,:),2,-1)

      NN(1,:,:) = NN(1,:,:)-UT+DR_1
      NN(2,:,:) = NN(2,:,:)-UT+DR_2
      NN(3,:,:) = NN(3,:,:)-UT+DR_3
      NN(4,:,:) = NN(4,:,:)-UT+DR_4          !calculate difference vector
      NN(5,:,:) = NN(5,:,:)-UT+DR_5
      NN(6,:,:) = NN(6,:,:)-UT+DR_6

      UTM1 = SUM(NN*(K*(1.-A1/ABS(NN))),DIM=1)/M + 2.*UT - UTM1
```

Die CSHIFT Kommandos werden benutzt, um die Verschiebungen zwischen benachbarten Prozessoren zu kommunizieren. Dadurch, daß bereits kommunizierte Daten weiter benutzt werden, kann die Anzahl der Kommunikationen auf insgesamt sechs reduziert werden. Da wir hier komplexe Zahlen und keine volle Nächste-Nachbar Kommunikation benutzen, ist eine Benutzung von `stencil` Operationen, die speziell für Finite-Differenzen Methoden für die Connection Machine entwickelt wurden, nicht angeraten. Die globale Summe entlang der ersten Feld-Dimension berechnet die von den Nachbarn auf jeden Gitterplatz ausgeübte Kraft. Da die ersten Dimensionen des Arrays k und der Vektoren zu den nächsten Nachbarn NN alle als `:serial` Dimension auf denselben Prozessor gelegt sind, erfordert die Berechnung der SUM keinerlei Kommunikation.

Leider liefert diese einfache Implementierung noch nicht die optimale Performance. Der Compiler alloziert und dealloziert unnötig viele temporäre Felder und generiert sogar globale `CM_send` Router Kommunikation, welche tatsächlich 43% der gesamten Rechenzeit benötigt! Um eine bessere Performance zu erhalten, haben wir das gesamte obige Programm-Fragment in PARIS (PARallel Instruction Set) codiert. Dadurch haben wir eine vollständige Kontrolle über die Speicherzuteilung, die Kommunikation und können explizit durch eine Pipeline verknüpfte Kommandos wie `CM_f_sub_const_mult_always` benutzen, die die Connection Machine Hardware besonders gut ausnutzen. Nun wird tatsächlich die meiste Rechenzeit — 38% — für die Berechnung von $|\vec{r}_i - \vec{r}_j|$ aufgewendet. Der Kommunikationsanteil kann nun vernachlässigt werden und ergibt lediglich 5.8% der Rechenzeit.

Diese Verbesserungen resultieren in einer großen Beschleunigung um einen Faktor fünf. Man erhält auf 8K Prozessoren einer CM2 eine Update Rate von 1.1 "Millionen von Updates pro Sekunde" (MUPS) und eine Geschwindigkeit von 110 MFLOPS. Im Vergleich dazu erhält man für einen typischen Spin Zellularautomaten mit Multispin-Coding mehr als 1000 MUPS auf einem Prozessor einer NEC-SX3 [118] und auf einer CM2-32K läuft ein Q2R Zellular

Automat mit 1600 MUPS [119]. Für eine komplette Molekular Dynamik Simulation auf einer CM200-8K erreichen Hedman and Laaksonen 0.2 MUPS [120] und für eine MD Simulation auf einer CM2-16K erreichen Mel'čuk et al. eine Update Rate von 4.5 KUPS [121]

6.4 Die Simulation

Alle Simulationen wurden auf einem 1024 × 1024 Gitter durchgeführt und wir benutzten 3000 und 5000 Zeitschritte. Zu Beginn der Rechnung wurden die Kopplungen zufällig aus einer uniformen Verteilung mit einem Mittelwert $k_0 = 0.005$ und einer typischen Breite von 50% gezogen. Wir brechen anfangs alle Bonds zwischen dem zentralen Gitterplatz $\vec{r}_0$ und seinen Nachbarn. Danach wird dem Gitter ein Anfangs-Impuls aufgeprägt: Wenn $\vec{r}_0$ der zentrale Gitterplatz ist, dann werden alle seine nächsten Nachbarn $\vec{r}_1 \ldots \vec{r}_6$ radial um 100 Gitterkonstanten nach außen verschoben, während alle anderen Gitterplätze festgehalten werden. Diese Verschiebung scheint zunächst sehr groß, aber da wir ein exakt harmonisches Potential benutzen, ist die tatsächliche Größe dieser Verschiebung nicht relevant. Wir wollen nur sicherstellen, daß die aufgeprägte Welle stärker als das durch die zufälligen Kopplungen aufgeprägte Rauschen ist. Zur Zeit $t = 0$ werden alle Gitterplätze freigelassen und das System kann sich frei entwickeln. Danach wird nach jedem zweiten Zeitschritt der Bond mit der größten Auslenkung l_m bestimmt und alle Bonds gebrochen, deren Auslenkung größer als $\alpha \cdot l_m$ ist — wobei α ein freier Parameter ist — und welche auf der Oberfläche des bereits generierten Risses liegen. Alle diese Bonds werden gebrochen, indem man die entsprechenden Kopplungen k_{ij} auf Null setzt. Man muß beachten, daß diese Wachstumsregel keinerlei Kommunikation benötigt und vollkommen parallel auf den Plätzen i und j durchgeführt werden kann, da jeder der beiden beteiligten Gitterplätze eine lokale Kopie von k_{ij} besitzt.

Wir betrachten also ein relatives Bruchkriterium statt eines absoluten, was folgende Gründe hat: Bei einem absoluten Kriterium würde man alle Bonds brechen, deren Ausdehnung einen kritischen Wert l_c überschreitet. Andererseits nimmt die Amplitude der nach außen laufenden Welle mit zunehmendem Abstand ab. Wenn also die Welle ursprünglich eine Amplitude oberhalb des kritischen Wertes hat, bricht die Welle alle Bonds, die sie erreicht, und nachdem die Amplitude unter den kritischen Wert gefallen ist, werden keine weiteren Bonds gebrochen. Man erzeugt auf diese Weise lediglich ein strukturloses Loch im Medium.

Ein ähnliches Modell wurde bereits von Louis und Guinea [81] studiert. Sie lösen ein quasi-statisches Problem, führen dabei allerdings nur ein paar Relaxationsschritte durch, um das Gleichgewicht in deren Gittersystem zu ermitteln. Da sie jedoch eine Overrelaxation-Methode benutzen, hat in ihrer Rechnung jeder Zeitschritt eine andere dynamische Bedeutung, als in dieser Methode und kann nicht mit einer tatsächliche Zeitentwicklung in Verbindung gebracht werden. Eine weitere Parallele findet man bei der Wachstumsregel. Louis et al. wählen Bonds, die gebrochen werden sollen, mit einer Wahrscheinlichkeit aus, die proportional zur Bondlänge ist $P_{ij} \propto |\vec{r}_i - \vec{r}_j|$. Dies entspricht einer annealed disorder, die als thermisch angeregtes Rißwachstum interpretiert werden kann.

6.5 Resultate

Um zu demonstrieren, daß die gewählte Dynamik (74) vernünftige Resultate liefert, betrachten wir zunächst den Fall eines kontinuierlichen Wellenpaketes, daß durch ein System ohne

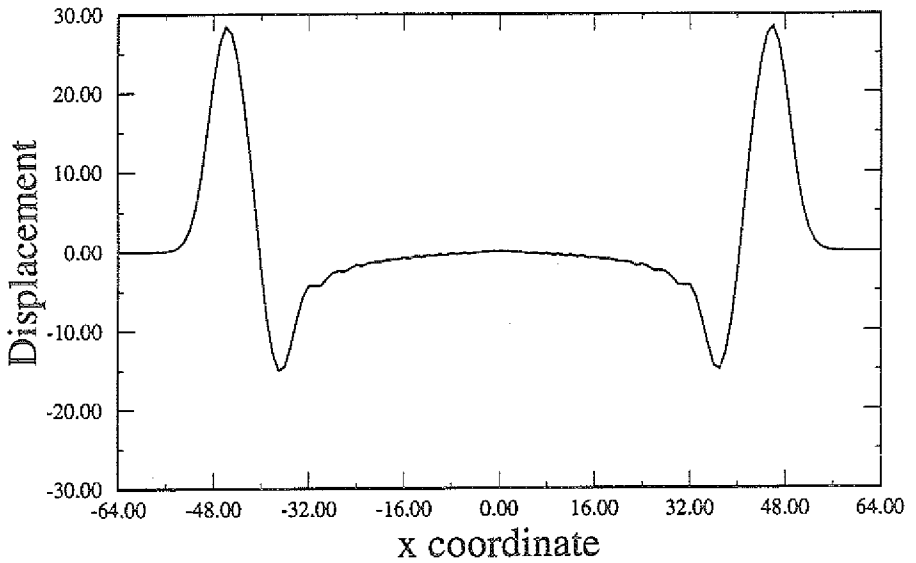


Abbildung 33 Schnitt durch eine Schockwelle nach 200 Zeitschritten.

Unordnung und ohne Risse wandert. Dazu wählen wir einen kontinuierlichen initialen Puls der Form

$$\vec{u}(t) = \vec{u}_0 \cdot \left(1 - \cos\left(2\pi \cdot \frac{t}{\tau}\right)\right) \tag{75}$$

mit $0 \leq t \leq \tau$ und $\vec{u}(t) = 0$ bei $t > \tau$ während sich alle anderen Gitterplätze frei bewegen können. Die Periode ist gewählt als $\tau = 100$.

Fig. (33) zeigt einen Schnitt entlang der x-Achse durch die aufgeprägte Welle zur Zeit $t = 200$. Man findet tatsächlich den initialen Puls wieder, der von einer weiteren kleinen Welle begleitet wird, die durch die Trägheit des ausgelenkten, zentralen Platzes verursacht wird. Bei einer Messung der Geschwindigkeit des Wellen-Maximums findet man die Gruppengeschwindigkeit einer radialen Welle mit Frequenz $\omega = 2\pi/\tau$ auf einem Dreiecksgitter.

Nachdem wir überprüft haben, daß diese Welle das erforderliche Verhalten aufweist, prägen wir nun einen singulären Puls auf, der dem Fall $\tau = 2$ entspricht.

In den Bildern (34-36) zeigen wir verschiedene Beispiele von Rissen, die für verschiedene Bruch-Schwellen generiert wurden: $\alpha = 1$ (10934 sites) — hier wurde nur genau der jeweils am stärksten belastete Bond gebrochen —, $\alpha = 0.98$ (16038 sites) und $\alpha = 0.95$ (35837 sites). In Klammern steht die Anzahl der Gitterplätze, die zu gebrochenen Bonds gehören. Alle Risse zeigen eine sternartige, fraktale Struktur. Für abnehmendes α sind die Strukturen mehr und mehr verzweigt. Das ist einfach zu verstehen, denn mit abnehmendem α liegt die Last von wesentlich mehr Bonds oberhalb der Materialfestigkeit und können gebrochen werden. Außerdem muß man beachten, daß wir ein vorgespanntes Netzwerk simulieren: Jeder Bond, der gebrochen wird, ist Ausgangspunkt einer weiteren Welle. Für kleines α können auch diese Streuwellen wieder zum Wachstum von Rissen beitragen und somit ein lawinenartiges Anwachsen der gebrochenen Bonds nach sich ziehen. Man kann daher zwei verschiedene Bereiche unterscheiden: Für α nahe bei Eins wächst der Riß im wesentlichen an den Rißspitzen, welche im wesentlichen mit der Front der aufgeprägten Welle übereinstimmen.

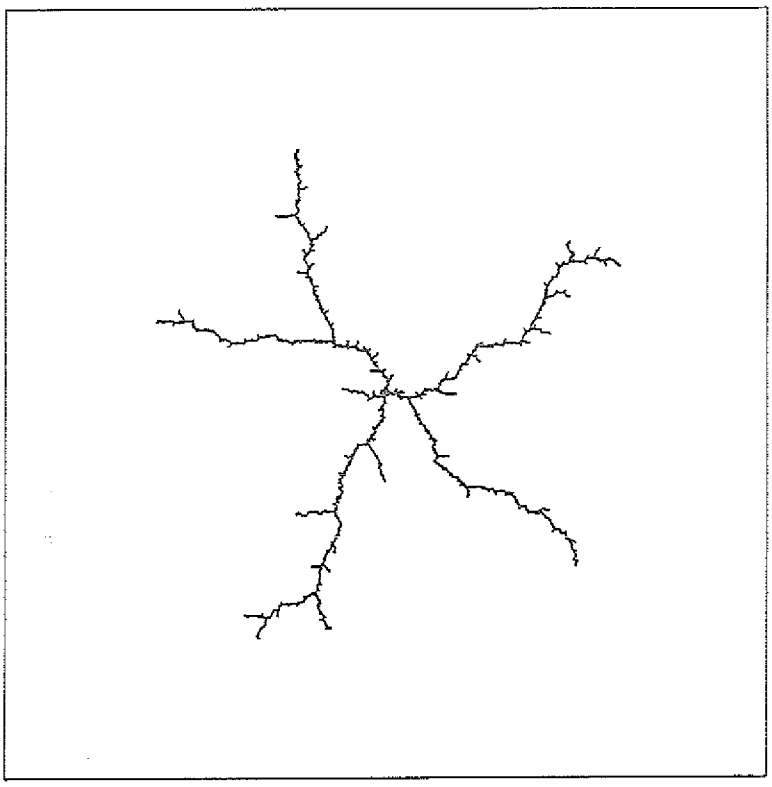


Abbildung 34 Rißstruktur generiert für $\alpha = 1$.

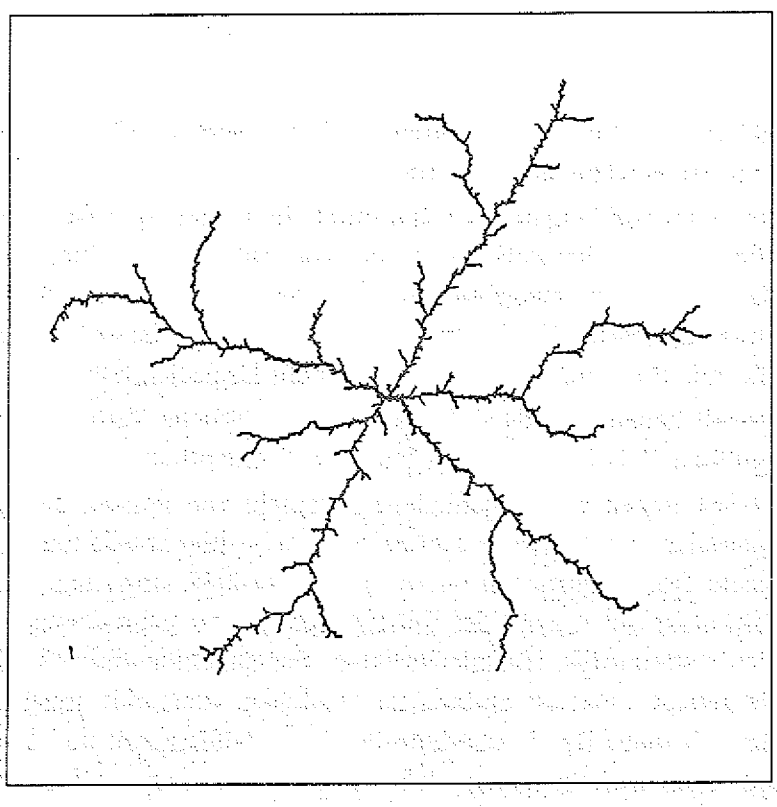


Abbildung 35 Rißstruktur generiert für $\alpha = 0.98$.

Die Seitenzweige hinter der Front bleiben inaktiv. Für kleineres α beginnen auch die Risse hinter der Schockfront zu wachsen und spalten sich auf, und letztendlich wir der gesamte Riß raumfüllend.

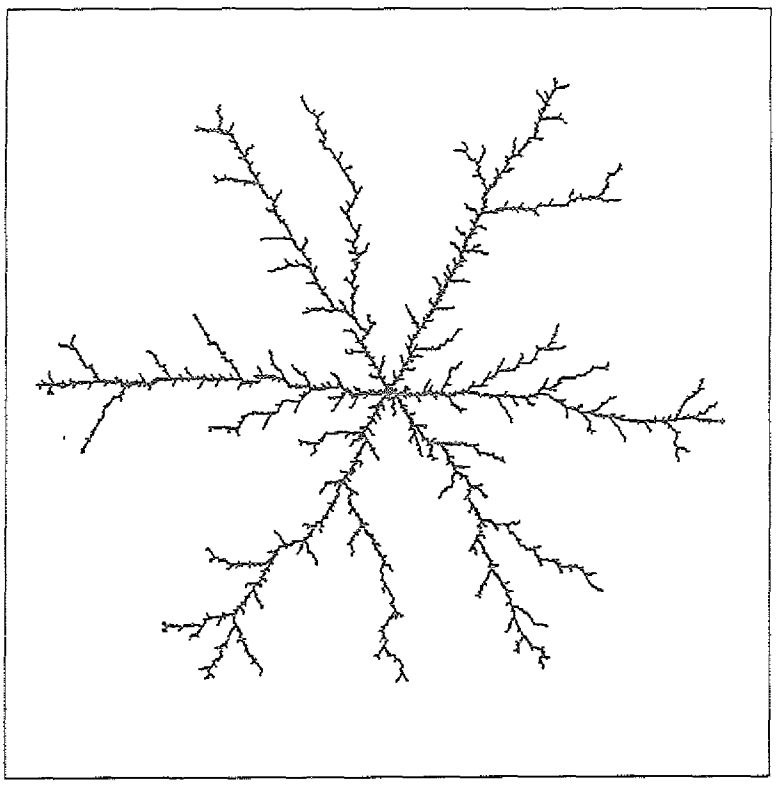


Abbildung 36 Rißstruktur generiert für $\alpha = 0.95$.

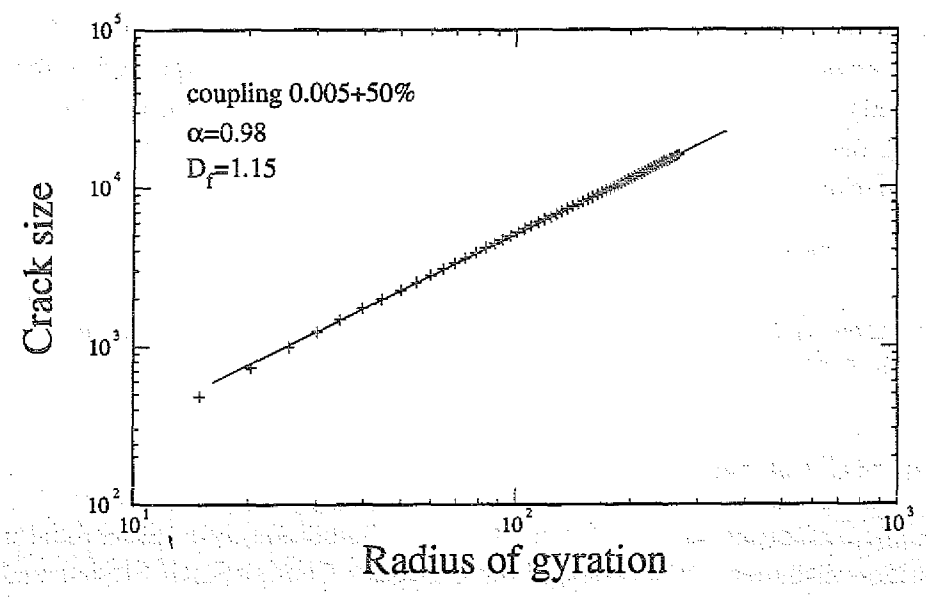


Abbildung 37 Scaling der Anzahl der gebrochenen Bonds mit dem Radius of Gyration.

Darüberhinaus stellt man fest, daß mit kleinerem α die Struktur des verwendeten Gitters immer mehr an Einfluß gewinnt und der Riß letztendlich eine sechszählige Symmetrie aufweist.

Eine quantitative Beschreibung erreicht man durch das Studium der fraktalen Dimension, die hier über den Radius of Gyration R_G bestimmt wird. In Fig. (37) zeigen wir das typische Abhängigkeit der Anzahl der gebrochenen Bonds vom Radius of Gyration. Exemplarisch

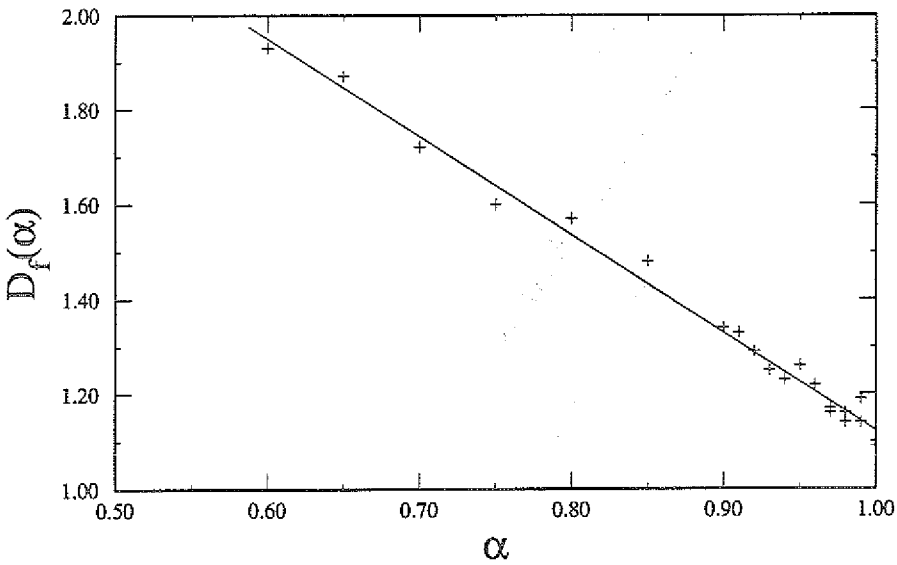


Abbildung 38 Abhängigkeit der fraktalen Dimension vom Bruch-Schwellwert .

zeigen wir die Daten für $\alpha = 0.98$, für die über vier verschiedene Risse gemittelt wurde. Man sieht sehr schön das Skalenverhalten

$$N \propto R_G^{D_f} \tag{76}$$

und für dieses spezielle Beispiel erhält man eine fraktale Dimension $D_f = 1.15$. Wie oben angesprochen variiert diese Dimension mit dem α. In Bild (38) zeigen wir daher die Abhängigkeit der fraktalen Dimension für verschiedene α. Auch in diesem Graphen haben wir zur Bestimmung der fraktalen Dimension jeweils über vier Cluster gemittelt. Man findet eine lineare Abhängigkeit

$$D_f(\alpha) = -(2.06 \pm 0.05) \cdot \alpha + (3.19 \pm 0.05) \tag{77}$$

D. h. mit abnehmendem α nähert man sich einer raumfüllenden Struktur, die man für einen kritischen Wert $\alpha \approx 0.58$ erreicht.

6.6 Zusammenfassung

Wir haben die Implementierung und die Resultate einer Simulation von Rißwachstum mit diskreten Zeitschritten diskutiert. Wir benutzen ein einfaches Central-Force Modell und untersuchen die schnellen Rißprozesse ("Fast-Cracking") anstatt der üblicherweise betrachteten quasi-statischen Vorgänge ("Slow-Cracking"). Mit Hilfe einer vereinfachten Dynamik, die trotzdem die wesentlichen Eigenschaften zur Rißerzeugung erhält, sind wir in der Lage, Risse mit mehr als 10000 Plätzen auf einem 1024×1024 Gitter zu generieren. Wir erhalten fraktale Rißmuster mit einer fraktalen Dimension, die linear von der Materialfestigkeit α abhängt und einen Bereich zwischen 1.1 and 2.0 abdeckt.

Kapitel 7

Rißwachstum mit Random-Walkern+

In enger Analogie zu Diffusion Limited Aggregation (DLA) und angeregt durch eine Arbeit von Roux, konstruieren wir einen Random-Walk Algorithmus, mit dem das Rißwachstum in elastischen Medien simuliert werden soll. Im Gegensatz zu den üblichen Gittermodellen wird hier das Spannungsfeld nicht im gesamten Medium bestimmt, sondern wir benutzen Random-Walker, um diejenigen Punkte auf der Rißoberfläche zu bestimmen, an denen die größte Spannung herrscht. An dieser Stelle wird das Spannungsfeld mit Hilfe einer analytisch berechneten Green-Funktion bestimmt. Die komplizierte Randbedingung an der Rißoberfläche wird berücksichtigt, indem Walker immer wieder in das System zurückgeschickt werden. Mit dieser Methode können wir Risse generieren, die bis zu 20.000 Teilchen enthalten. Alle Simulationen können auf konventionellen Workstations innerhalb relativ kurzer Zeit durchgeführt werden. Wir simulieren mehrere verschiedene Randbedingungen wie einachsigen Zug, reine Scherung oder isotrope Dilatation. Darüberhinaus studieren wir den Einfluß der verschiedenen freien Modell-Parameter wie Noise-Reduction und Ermüdungs-Effekte. Wir untersuchen weiter den Einfluß eines anisotropen Random-Walk, bei dem ein Bias aus dem lokalen Spannungsfeld bestimmt wird. Mit diesem neuen Modell gelingt es uns, typische Rißstrukturen aus Experimenten und anderen numerischen Rechnungen zu reproduzieren und die wesentlichen Eigenschaften von realistischen Cracks zu simulieren.

7.1 Einleitung

Das Studium von Nicht-Gleichgewichts Wachstumsmodellen ist in den letzten Jahren extrem populär und erfolgreich gewesen. Besonders das durch die Laplace-Gleichung bestimmte Wachstum wurde eingehend untersucht und eine Vielzahl von Anwendungen kann auf den verschiedensten Gebieten gefunden werden. Als Beispiele sollen hier Elektrodeposition [16], Verdrängung viskoser Flüssigkeiten Displacement [13] oder das Wachstum von Bakterienkolonien[15] angeführt werden. Ein Grund für die Popularität ist, daß eine numerische Simulation sehr effizient mit Hilfe von Diffusion Limited Aggregation (DLA) [9] möglich ist. Mit Hilfe

+ HLRZ-preprint[122]

von optimierten Programmen [42,43] ist es möglich, sehr große Cluster mit 30 oder sogar 50 Millionen Teilchen zu generieren [123,124].

Dieses Kapitel wird sich jedoch nicht mit DLA beschäftigen. Statt dessen werden wir — angeregt durch einen Artikel von Roux [10]— in enger Analogie zu DLA einen Algorithmus konstruieren, mit dem das Problem des Rißwachstums in elastischen Medien studiert werden soll [63,65,64]. Bei diesem Problem wird eine bestimmte Kraft auf ein elastisches Medium angewendet, in dem sich daraufhin Risse ausbilden, so daß das Material in mehrere Teile auseinander brechen kann. Obwohl bereits viele optimierte und wohlbekannte Methoden — die im wesentlichen auf Gittermodellen basieren — existieren [125,83], bleibt die numerische Simulation des Rißwachstums selbst auf modernsten Supercomputern ein extrem zeitaufwendige Aufgabe, so daß immer noch nur kleine Systeme mit wenigen tausend Gitterplätzen und dementsprechend nur Risse mit einigen hundert bis zu wenigen tausend gebrochenen Bonds simuliert werden können. Deswegen werden neue Algorithmen und Methoden, die vielleicht sogar die Simulation von Rißwachstum auf Workstations ermöglichen, dringend benötigt.

Wie bereits von mehreren Wissenschaftlern aufgezeigt wurde [126,127], findet man einige bemerkenswerte Parallelen zwischen dem vektoriellen Elastizitätsproblem — das durch die Lamé-Gleichung bestimmt wird — und dem skalaren Problem der Elektrostatik — welches von der Laplace-Gleichung bestimmt wird. Roux hat sogar exakt gezeigt [10], daß es formal möglich ist, die Green-Funktion der Lamé-Gleichung mit Hilfe eines vektoriellen Random-Walks zu konstruieren, und damit die wohlbekannte Methode zur Bestimmung des elektrischen Potentials mit Hilfe von Random-Walkern erweitert. Allerdings ist die tatsächliche Implementierung und Interpretation eines Random-Walk Algorithmus im Falle der Laplace-Gleichung wesentlich einfacher als für das Elastizitätsproblem. Obwohl Roux's Arbeit [10] den Weg einer konkreten, praktischen Realisation seiner Ideen deutlich aufzeigt, ist bisher keine Umsetzung seiner Arbeit in eine numerische Simulation erfolgt, und es gibt auch keinerlei Überprüfung, ob ein derartiger Algorithmus überhaupt praktikabel ist.

In der vorliegenden Arbeit werden wir Roux's Gedanken und Ergebnisse diskutieren und erweitern und einen darauf basierenden neuen Algorithmus vorstellen, mit dessen Hilfe wir das Rißwachstum in elastischen Medien angehen wollen. Zu diesem Zweck werden wir eine DLA-artige Random-Walker Methode einführen, in der die Walker dazu benutzt werden, diejenigen Punkte auf der Rißoberfläche zu bestimmen, an denen die größte Spannung herrscht und an denen somit ein Riß weiterwachsen wird. "DLA-artig" ist hier nicht in dem Sinne von "probabilistisch" zu verstehen, wie er in [81,82,128,129] benutzt wird. Dort ist im wesentlichen ein Wachstum analog zum DBM Modell von Niemeyer et al. [14] gemeint. Wir verstehen unter DLA-artig die Benutzung von Random-Walkern, um Informationen über angelegte Kräfte an die Rißoberfläche zu transportieren.

Der Aufbau dieses Kapitels ist der folgende. Im Abschnitt 7.2 werden wir eine systematische Einführung in das Problem des Rißwachstums geben. Wir werden kurz die Ideen und Resultate von Roux [10] wiedergeben und erweitern, und wir werden die Gründe diskutieren, warum eine direkte Umsetzung in eine numerische Methode nicht praktikabel ist. Im Abschnitt 7.3 werden wir ein neues Modell für das Rißwachstum vorstellen, bei dem wir uns von Roux's ursprünglichen Ideen und dem DLA Algorithmus leiten lassen werden. Wir werden auch die besondere Behandlung der Rißoberfläche diskutieren, was sich als das wesentliche Problem dieser Methode herausstellen wird. Danach werden wir im Abschnitt 7.4 die mit diesem Modell erzielten Resultate vorstellen. Wir werden verschiedene Variationen studieren, um den Einfluß der verschiedenen Parameter zu verstehen und die Anwendbarkeit

auf realistische Fälle zu untersuchen. Diese Ergebnisse werden wir mit verschiedenen analytischen, experimentellen und anderen numerischen Resultaten vergleichen. Im Abschnitt 7.5 werden wir dann alle Vorteile und Probleme dieser Methode kritisch diskutieren und unsere Ergebnisse zusammenfassen.

7.2 Zusammenhang zwischen dem Rißwachstum und Random-Walkern

Betrachte ein d dimensionales unendliches elastisches Medium mit einer Poisson-Zahl ν. Dieses System kann beschrieben werden durch ein Spannungsfeld $\underline{\underline{\sigma}}$, welches im Gleichgewicht

$$\mathrm{Div}\underline{\underline{\sigma}} = 0 \tag{78}$$

erfüllt. In Verbindung mit dem Hookeschen Gesetz kann dies ausgedrückt werden durch das Verschiebungsfeld $\vec{u}$ und man erhält die Lamé-Gleichung

$$(1-\nu(d-1))\nabla^2\vec{u} + \nabla(\nabla\cdot\vec{u}) = 0. \tag{79}$$

Ein Riß kann in einem solchen System als eine zusätzliche kräftefreie Oberfläche beschrieben werden und daher hat man an ihm die zusätzliche Randbedingung

$$\underline{\underline{\sigma}}\cdot\vec{n} = 0. \tag{80}$$

Hier muß betont werden, daß diese Randbedingung durch das Spannungsfeld ausgedrückt wird — dies entspricht im DLA Fall dem elektrischen Feld — und nicht durch das Verschiebungsfeld — was im DLA Fall dem elektrischen Potential entspricht. Dieser Unterschied macht es erforderlich, daß die Rißoberfläche anders behandelt werden muß, als bei DLA.

Neben diesen allgemeinen Gleichgewichtsbedingungen, kann die Bedingung für das Rißwachstum mit Hilfe der Komponente des Spannungstensors parallel zur Rißoberfläche $\sigma_{\|}$ formuliert werden. Diese hat einen kritischen Materialparameter σ_c zu überschreiten, welcher direkt mit den intermolekularen Kohäsionskräften verknüpft ist, und die Wachstumsgeschwindigkeit v_n eines Risses ist in dieser Darstellung bestimmt durch

$$v_n \propto \left(\sigma_{\|} - \sigma_c\right)^{\eta}. \tag{81}$$

Die Größe η ist ein heuristischer Parameter, der oft der Einfachheit halber auf Eins gesetzt wird. Die Gleichungen (79, 80, 81) formulieren zusammen das Rißwachstum als ein nichtlineares Moving-Boundary Problem.

Das Bruchkriterium (81) entspricht einem erweiterten Griffith-Kriterium, welches für spröde Materialien gültig ist. Neben dieser einfachen Regel existieren noch weitere Kriterien, die z. B. die Biegung eines Materials berücksichtigen. Besonders im Ingenieurbereich sind noch eine Vielzahl weiterer Bruchkriterien heuristisch bestimmt worden, die alle für bestimmte Materialklassen gültig sind (vgl. [74]). Von diesen sei hier nur noch das von-Mises-Kriterium erwähnt, welches das Einsetzen von plastischer Deformation in polykristallinen Körpern beschreibt. Es kann formuliert werden als

$$\frac{1}{16}\left[(S_1-S_2)^2 + (S_2-S_3)^2 + (S_1-S_3)^2\right] \geq k^2 \tag{82}$$

wobei S_1, S_2 und S_3 die Hauptschubspannungen — also die Eigenwerte des Spannungstensors — sind und k eine kritische Materialfestigkeit bedeutet, die in einem uniaxialen Zugversuch bestimmt werden kann.

Um ein vollständige Einführung in das Problem zu geben, werden wir nun kurz die wesentlichen Gedanken von Roux [10] zusammenfassen, um sie danach erweitern und kommentieren zu können. Der grundlegende Gedanke seiner Arbeit besteht darin, einen Markov Prozeß zu formulieren, der einen vektoriellen Charakter besitzt und letztendlich auf eine Green-Funktion der Lamé-Gleichung führt. Dies wird in enger Analogie zu der einfachen skalaren Diffusion durchgeführt, die man letztendlich als Grenzfall in der folgenden Herleitung wiederfinden wird.

Betrachte daher ein elastisches Material in dessen Innerem ein (unbekanntes) Verschiebungsfeld $\vec{u}$ herrscht, welches ein Ergebnis der am Rand angelegten Verschiebungen ist. Dieses Verschiebungsfeld soll am Punkt $\vec{x}$ bestimmt werden. Zu diesem Zweck betrachte den folgenden Markov Prozeß: Ein Random-Walker startet an einem Punkt $\vec{x}$ und führt einen ersten, elementaren Schritt der Länge r in die Richtung $\vec{e}_1$ durch. Nach einer Mittelwertbildung über alle möglichen Richtungen $\vec{e}_1$ definiert man in $\vec{x}$ einen neuen Vektor

$$(Q_r\vec{u})(\vec{x}) = \left\langle \underline{\underline{Q}}(\vec{e}_1)\cdot\vec{u}(\vec{x}+r\vec{e}_1)\right\rangle_{\vec{e}_1} \tag{83}$$

wobei die elementare Schrittmatrix $\underline{\underline{Q}}(\vec{e})$ definiert ist als

$$\underline{\underline{Q}}(\vec{e})_{ij} = \left(\alpha\cdot\underline{\underline{1}}+(1-\alpha)\cdot d\cdot\vec{e}\otimes\vec{e}\right)_{ij} = \left(\alpha\cdot\delta_{ij}+(1-\alpha)\cdot d\cdot e_ie_j\right) \tag{84}$$

und α ein Parameter ist, der später noch definiert wird. Hierbei ist $\vec{e}\otimes\vec{e}$ die Projektionsmatrix $(\vec{e}\otimes\vec{e})_{i,j} \equiv e_i\cdot e_j$.

Wie üblich in Markov Prozessen wird dieser elementare Schritt iteriert. Daher definiert man nach N unabhängigen Schritten — wobei jeder von der Länge r ist — an der Stelle $\vec{x}$ den neuen Vektor

$$\left(Q_r^N\vec{u}\right)(\vec{x}) = \left\langle \underline{\underline{Q}}(\vec{e}_N)\cdots\underline{\underline{Q}}(\vec{e}_1)\cdot\vec{u}(\vec{x}+r\vec{e}_1+\cdots+r\vec{e}_N)\right\rangle_{\vec{e}_1\ldots\vec{e}_N}. \tag{85}$$

Durch Fourier Transformation in den Impuls-Raum

$$\left(Q_r^N\vec{u}\right)\left(\vec{k}\right) = \int\left(Q_r^N\vec{u}\right)(\vec{x})\exp\left(-i\vec{k}\vec{x}\right)d^dx = \underline{\underline{Q}}_r^N\left(\vec{k}\right)\cdot\vec{u}\left(\vec{k}\right) \tag{86}$$

kann man den Grenzfall verschwindender Schrittweite $r\to 0$ und divergierender Anzahl von Schritten $N\to\infty$ bestimmen, wobei $t = Nr^2$ festgehalten wird. Mit Hilfe des zentralen Grenzwert Satzes erhält man

$$\begin{aligned}\lim_{\substack{N\to\infty\\ r\to 0}}\underline{\underline{Q}}_r^N\left(\vec{k}\right) &= \underline{\underline{Q}}^t\left(\vec{k}\right)\\ &= \exp\left(-\frac{t}{2d(d+2)}\left[(2\alpha+d)\vec{k}^2\underline{\underline{1}}+2d(1-\alpha)\vec{k}\otimes\vec{k}\right]\right) = \exp\left(\underline{\underline{M}}\right).\end{aligned} \tag{87}$$

Man sieht leicht nach der Rücktransformation in den Ortsraum, daß die Funktion

$$\left(Q^t\vec{u}\right)(\vec{x}) = \int\underline{\underline{Q}}^t(\vec{r})\cdot\vec{u}(\vec{x}+\vec{r})\,d^dr \tag{88}$$

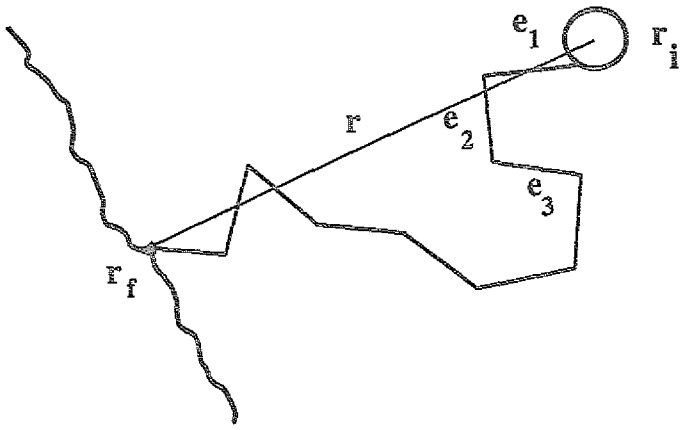


Abbildung 39 Illustration des einfachsten Algorithmus: Ein Walker startet am Rand bei $\vec{r}_i$ erreicht einen Riß bei $\vec{r}_f$. Während des Walks muß man das Produkt der Schrittmatrizen $\underline{\underline{Q}}(\vec{e}_N)\cdots\underline{\underline{Q}}(\vec{e}_1)$ bestimmen.

eine Lösung einer "zeitabhängigen" Lamé-Gleichung ist

$$\frac{\partial\big(Q^t\vec{u}\big)(\vec{x})}{\partial t} = \frac{(2\alpha+d)\nabla^2\big(Q^t\vec{u}\big)(\vec{x}) + 2d(1-\alpha)\nabla\big(\nabla\cdot\big(Q^t\vec{u}\big)(\vec{x})\big)}{2d(d+2)}. \tag{89}$$

Im stationären Grenzfall $\partial\big(Q^t\vec{u}\big)(\vec{x})/\partial t = 0$ erhält man offensichtlich ein Lösung der Lamé Gleichung, wenn α korrekt gewählt wird. Durch einfachen Koeffizientenvergleich findet man auch eine Bestimmungsgleichung für den Parameter α

$$\alpha = \frac{d(1-2\nu(d-1))}{2(d+1-d\nu(d-1))}. \tag{90}$$

Zusammenfassend kann man festhalten, daß in dieser Formulierung die Bestimmung einer zeitabhängigen Green-Funktion $\underline{\underline{Q}}^t(\vec{r})$, die in (88) benötigt wird, mit Hilfe von vektoriellen Random-Walkern ausgedrückt ist. Offensichtlich zeigt Gl. (85) den Markov Prozeß, den man dazu durchzuführen hat. Im folgenden werden wir aber zeigen, daß es nicht praktikabel ist, einen tatsächlichen Algorithmus direkt auf (85) aufzubauen.

Eine direkte und naive Implementierung von (85) würde die folgenden Schritte beinhalten: Ein Random-Walker würde am Rand an einer Stelle $\vec{r}_i = \vec{x}+\vec{r}$, an der eine bekannte Verschiebung $\vec{u}$ herrscht, gestartet werden. Dieser Walker möge nach einem isotropen Random-Walk einen Riß an der Stelle $\vec{r}_f = \vec{x}$ berühren. Während des Walks berechnet man gleichzeitig das Produkt der elementaren Schrittmatrizen $\underline{\underline{Q}}(\vec{e}_N)\cdots\underline{\underline{Q}}(\vec{e}_1)$ und schließlich mittelt man in $\vec{x}$ über alle eintreffenden Vektoren $\underline{\underline{Q}}(\vec{e}_N)\cdots\underline{\underline{Q}}(\vec{e}_1)\cdot\vec{u}(\vec{r}_i)$. Die erhaltenen Daten müssen extrapoliert werden zum Grenzfall $r\to 0$ — was wenigstens in dieser Implementierung automatisch den Grenzfall $N\to\infty$ nach sich zieht — und entsprechend der Gl. (88) ergibt dieser Prozeß schließlich die benötigte Lösung.

Aber obwohl dieser Prozeß stark an die Konstruktion einer Green-Funktion der Laplace-Gleichung erinnert, existieren mehrere Unterschiede, die letztendlich einige Schwierigkeiten nach sich ziehen. Im Gegensatz zu DLA ist hier die Geschichte der einzelnen Walks extrem wichtig. Die Produktmatrix hängt explizit von jedem einzelnen Schritt $\vec{e}_i$ dieses Walks ab und daher ist der tatsächliche Weg, auf dem der Walker von seinem Start- zu seinem

Endpunkt gelangt, essentiell wichtig. Da aber exponentiell viele mögliche Wege von $\vec{r}_i$ nach $\vec{r}_f$ existieren, muß der Beitrag eines einzelnen Walkers zum Mittelwert (85) beliebig klein werden.

Andererseits ist es notwendig, die entsprechenden Grenzübergänge $N \to \infty$ durchzuführen und daher wächst die Anzahl der Faktoren in der Produktmatrix schnell an. Da jedoch die Eigenwerte von $\underline{\underline{Q}}(\vec{e})$ ungleich von Eins sind, hat die Produktmatrix $\underline{\underline{Q}}(\vec{e}_N)\cdots\underline{\underline{Q}}(\vec{e}_1)$ eine unkontrollierbare Norm, die mit der Schrittzahl exponentiell divergiert oder verschwindet. Aus diesen Argumenten folgt, daß der Betrag des Vektors $\underline{\underline{Q}}(\vec{e}_N)\cdots\underline{\underline{Q}}(\vec{e}_1)\cdot\vec{u}(\vec{r}_i)$ nicht kontrollierbar ist. Ähnliche Argumente führen sofort zu dem Schluß, daß nicht nur die Norm, sondern auch die Richtung des resultierenden Vektors exponentiell stark fluktuiert. Das Resultat dieser Überlegungen ist also, daß eine naive Implementierung eines Algorithmus, wie er durch Gl. (85) vorgeschlagen wird, enorme statistische Fluktuationen sowohl des Betrages als auch der Richtung von $\left(Q_r^N\vec{u}\right)(\vec{x})$ nach sich zieht, die in numerischen Simulationen nicht unterdrückt werden können.

Eine weitere wichtige Schwierigkeit ist, daß der eben beschriebene Algorithmus die Gegenwart eines Risses, also eines kraftfreien Randes, der das Spannungsfeld stark beeinflußt, völlig vernachlässigt. Darüberhinaus ist es auch nicht leicht möglich, die Randbedingung (80) einzubeziehen, denn sie ist mit Hilfe des Spannungsfeldes $\underline{\underline{\sigma}}$ formuliert, welches nicht in der vorhergehenden Betrachtung berücksichtigt ist. Zwar ist es möglich, diese Randbedingung mit Hilfe des Verschiebungsfeldes $\vec{u}$ auszudrücken, aber dies zieht die Bestimmung von räumlichen Ableitungen des Verschiebungsfeldes nach sich, welche noch weniger zugänglich sind, als die Verschiebungen selbst.

Trotzdem zeigten die vorhergehenden Überlegungen, daß ein Zweck der Random-Walker darin besteht, die Matrix $\underline{\underline{Q}}^t(\vec{r})$, die die Grundlage von (88) ist, zu konstruieren. In einem unendlichen System kann diese Matrix, die bereits den Mittelwert über alle Produktmatrizen von Walks, die im Ursprung starten und in $\vec{r}$ enden, analytisch bestimmt werden. Somit kann man (88) direkt mit Hilfe von Random-Walks auswerten, wobei die Walks jetzt lediglich dazu benutzt werden, einen Start- und Endpunkt $\vec{x}$ und $\vec{x}+\vec{r}$ zu bestimmen. Hierbei ist der explizite Weg eines jeden Walks nicht mehr relevant und daher hat man nicht die vorhin beschriebenen großen Fluktuationen zu erwarten. Deswegen werden wir jetzt kurz die Rücktransformation von (87) in den Ortsraum und die Bestimmung von $\underline{\underline{Q}}^t(\vec{r})$ darstellen, um die daraus entstehenden Probleme diskutieren zu können

Da die Matrix $\underline{\underline{M}}$ im Exponenten von (87) symmetrisch ist, kann sie leicht diagonalisiert werden, und daher kann $\underline{\underline{Q}}^t\left(\vec{k}\right)$ in geschlossener Form berechnet werden. Man erhält in zwei Dimensionen

$$\underline{\underline{Q}}^t\left(\vec{k}\right) = \frac{1}{\vec{k}^2}\begin{pmatrix} k_1^2\exp_2 + k_2^2\exp_1 & k_1k_2\{\exp_2-\exp_1\} \\ k_1k_2\{\exp_2-\exp_1\} & k_1^2\exp_1 + k_2^2\exp_2 \end{pmatrix} \tag{91}$$

wobei $\exp_i = \exp\left(-t\lambda_i\vec{k}^2\right)$ gilt und λ_1 und λ_2 die Eigenwerte

$$\begin{aligned} \lambda_1 &= \frac{2\alpha+d}{2d(d+2)} \\ \lambda_2 &= \frac{3d+2\alpha(1-d)}{2d(d+2)} \end{aligned} \tag{92}$$

von $\underline{\underline{M}}$ sind. Die Transformation in den Ortsraum

$$\underline{\underline{Q}}^t(\vec{r}) = \int \underline{\underline{Q}}^t\left(\vec{k}\right) \exp\left(i\vec{k}\vec{r}\right) \frac{d^2k}{(2\pi)^2} \tag{93}$$

erfordert dann die Berechnung von Integralen der Form

$$\int \frac{k_i k_j}{\vec{k}^2} \exp\left(-t\lambda\vec{k}^2\right) \exp\left(i\vec{k}\vec{r}\right) \frac{d^2k}{(2\pi)^2}. \tag{94}$$

Mit Hilfe von

$$\vec{k} = k \cdot \begin{pmatrix} \cos\varphi \\ \sin\varphi \end{pmatrix} \text{ and } \vec{r} = r \cdot \begin{pmatrix} \cos\varphi_r \\ \sin\varphi_r \end{pmatrix} \tag{95}$$

liefert die Berechnung der Winkel-Integration Bessel Funktionen der ersten und zweiten Art, $J_0(x)$ und $J_2(x)$

$$\begin{aligned}
&\int_0^{2\pi} \cos^2\varphi \cdot \exp\left(ikr \cdot \cos\left(\varphi - \varphi_r\right)\right) d\varphi = \pi[J_0(kr) - \cos\left(2\varphi_r\right) J_2(kr)] \\
&\int_0^{2\pi} \sin^2\varphi \cdot \exp\left(ikr \cdot \cos\left(\varphi - \varphi_r\right)\right) d\varphi = \pi[J_0(kr) + \cos\left(2\varphi_r\right) J_2(kr)] \\
&\int_0^{2\pi} \sin\varphi\cos\varphi \cdot \exp\left(ikr \cdot \cos\left(\varphi - \varphi_r\right)\right) d\varphi = \pi[-\sin\left(2\varphi_r\right) J_2(kr)]
\end{aligned} \tag{96}$$

Die verbleibende r-Integration erfordert Integrale von Bessel und Gauß Funktionen und schließlich erhält man [130]

$$\underline{\underline{Q}}^t(\vec{r}) = \frac{1}{2\pi}[f_s(r,t) + f_a(r,t)] \cdot \underline{\underline{1}} - \frac{1}{2\pi} f_a(r,t) \cdot 2\frac{\vec{r} \otimes \vec{r}}{\vec{r}^2} \tag{97}$$

wobei gilt

$$\begin{aligned}
f_s(r,t) &= \frac{\exp\left(-r^2/4\lambda_1 t\right)}{4\lambda_1 t} + \frac{\exp\left(-r^2/4\lambda_2 t\right)}{4\lambda_2 t} \\
f_a(r,t) &= \left(1 + \frac{4\lambda_1 t}{r^2}\right) \frac{\exp\left(-r^2/4\lambda_1 t\right)}{4\lambda_1 t} - \left(1 + \frac{4\lambda_2 t}{r^2}\right) \frac{\exp\left(-r^2/4\lambda_2 t\right)}{4\lambda_2 t}
\end{aligned} \tag{98}$$

Dieses Resultat kann unter verschiedenen Gesichtspunkten diskutiert werden.

Man kann an der Struktur von $f_s(r,t)$ und $f_a(r,t)$ erkennen, daß der zugrundeliegende elementare Prozeß ein Random-Walk ist. Sie enthalten Gauß Funktionen, die das Resultat der freien Diffusion von Teilchen in einem unendlichen Gebiet sind. Den Grenzfall der reinen Laplace-Gleichung erhält man durch $\nu \to -\infty$ und in diesem Fall reduziert sich $\underline{\underline{Q}}^t(\vec{r})$ zur Green-Funktion des entsprechenden Laplace-Problems, welches ein einfache Gauß-Funktion ist.

Da man fordert, daß $t = Nr^2$ endlich bleiben soll, führt man eine neue Variable ein, die natürlich als Zeit interpretiert werden kann. Allerdings basieren alle neuen und effizienten DLA Algorithmen darauf, daß die Schrittweite eines Random-Walks dynamisch variiert wird

und somit ist es nicht möglich, in diesen Algorithmen eine sinnvolle Zeitskala einzuführen. Dies ist allerdings kein Nachteil, da für DLA Zeit irrelevant ist. Hier ist es offensichtlich nicht so, denn man findet eine explizite Zeitabhängigkeit! Darüberhinaus muß man beachten, daß Gl. (88) erst im Grenzfall $t \to \infty$ eine Lösung der Lamé-Gleichung liefert. In diesem Grenzfall verschwindet jedoch $\underline{\underline{Q}}^t(\vec{r})$ identisch, so daß man die triviale Lösung $\vec{u}(\vec{r}) \equiv 0$ erhält. Dies kann man in Analogie zur einfachen Diffusion leicht so interpretieren, daß in diesem Grenzfall eine ursprünglich vorhandene Verschiebung komplett "wegdiffundiert" ist.

Das eben bestimmte $\underline{\underline{Q}}^t(\vec{r})$ gilt lediglich in einem unendlichen Gebiet ohne Ränder, denn in (85) mittelt man im wesentlichen über alle Walks, die einen gegebenen Start- und Endpunkt verbinden. Man vernachlässigt, daß in der Gegenwart eines zusätzlichen Randes solche Walks von der Mittelung ausgenommen werden müssen, die diesen Rand berühren, bevor sie den Endpunkt erreichen. Prinzipiell kann man dies korrigieren, indem man den Mittelwert (85) nur aus jenen Walks bestimmt, die einen bestimmten Raumbereich nicht verlassen ("First-passage"). Allerdings ist dann eine geschlossene Bestimmung von $\underline{\underline{Q}}^t(\vec{r})$ nicht mehr möglich.

Zusammenfassend kann hier gesagt werden, daß ebenfalls eine numerische Simulation von (88) — obwohl technisch möglich — nicht praktikabel ist, da im wesentlichen die Einführung der neuen Zeitvariablen t nicht interpretiert werden kann. Zum Verständnis der folgenden Überlegungen muß betont werden, daß die Matrix $\underline{\underline{Q}}^t(\vec{r})$ in (89) die Rolle einer zeitabhängigen Green-Funktion spielt. Demnach führt der gesamte ursprüngliche Algorithmus auf die Berechnung einer zeitabhängigen Green-Funktion in einem unendlichen Medium mit Hilfe von Random-Walkern! Diese Green-Funktion ist aber analytisch direkt zugänglich, so daß deren Konstruktion nicht mit Hilfe von Random-Walks erfolgen muß. Auf diesem Resultat werden wir nun einen neuen Algorithmus aufbauen

7.3 Algorithmus

Aus den vorhergehenden Abschnitten ist klar geworden, daß die ursprüngliche Methode immer auf die Berechnung einer Green-Funktion des Verschiebungsfeldes mit Hilfe von Random-Walkern hinausläuft. Da allerdings sowohl die Randbedingung, als auch die Bedingung für das Rißwachstum mit Hilfe des Spannungsfeldes ausgedrückt ist, ist es natürlich, direkt eine Green-Funktion für das Spannungsfeld zu benutzen und daraus einen Algorithmus ähnlich den oben vorgestellten zu konstruieren.

Betrachte daher eine unendlichen elastischen zweidimensionalen Körper, der im Unendlichen festgehalten werden soll $\vec{u}(\infty) = 0$. Im Ursprung wird eine punktförmige Kraft $\vec{F}$ angelegt, welche ein Spannungsfeld $\underline{\underline{\sigma}}^*(\vec{r})$ erzeugt, das durch $\mathrm{Div}\,\underline{\underline{\sigma}}^* = \vec{F} \cdot \delta(\vec{r})$ bestimmt ist. Dieses Feld kann mit Hilfe der Methode von Kolossov-Mushkelishvili [131,132] bestimmt werden. Diese Methode ist im Anhang A skizziert. Man erhält so folgendes Spannungsfeld

$$\sigma^* = \begin{pmatrix} \sigma^*_{xx} & \tau^* \\ \tau^* & \sigma^*_{yy} \end{pmatrix} \tag{99}$$

mit den Komponenten

$$\begin{aligned}
\sigma^*_{xx} &= 2\Re\left(\frac{c}{z}\right) + \Re\left(\frac{\overline{z}c}{z^2} + \kappa\frac{\overline{c}}{z}\right) \\
\sigma^*_{yy} &= 2\Re\left(\frac{c}{z}\right) - \Re\left(\frac{\overline{z}c}{z^2} + \kappa\frac{\overline{c}}{z}\right) \\
\tau^* &= -\Im\left(\frac{\overline{z}c}{z^2} + \kappa\frac{\overline{c}}{z}\right).
\end{aligned} \tag{100}$$

Hierbei sind alle Vektoren als komplexe Größen dargestellt

$$\begin{aligned}
z &= x + iy \\
c &= \frac{F_x + iF_y}{2\pi(1+\kappa)} \\
\kappa &= 3 - 4\nu
\end{aligned} \tag{101}$$

und $\Re(z)$ bzw. $\Im(z)$ bezeichnen den Real- bzw. Imaginärteil von z.

Zusätzlich muß man beachten, daß die Wahrscheinlichkeit, daß ein Random-Walker, der im Zentrum eines Kreises mit Radius r gestartet wurde, ein bestimmtes Flächenelement $r \cdot d\varphi$ auf dem Rand erreicht, gerade durch $p\,d\varphi = d\varphi/2\pi r$ bestimmt ist. Dieser Faktor muß aus (100) herausdividiert werden, da wir dieses Spannungsfeld mit Hilfe von Random-Walkern auswerten wollen und so diesen Term automatisch durch die Random-Walker integrieren. Wir erhalten somit eine Green-Funktion

$$\underline{\underline{\sigma}} = 2\pi|z| \cdot \underline{\underline{\sigma}}^*, \tag{102}$$

die wir im folgenden in unserer Simulation benutzen werden.

Mit Hilfe dieser Green-Funktion wollen wir das Rißwachstum mit folgendem Algorithmus simulieren:

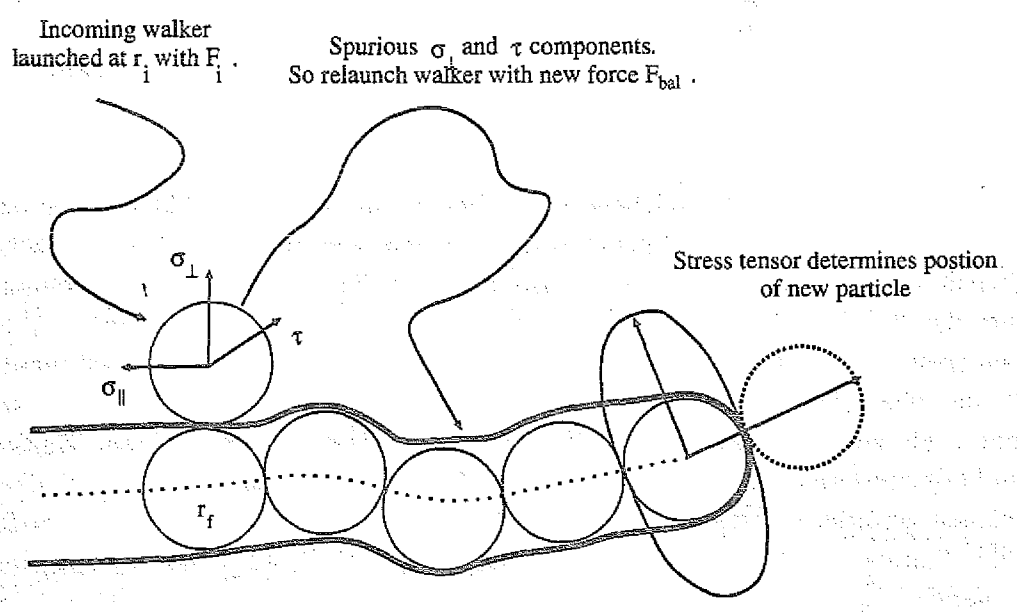


Abbildung 40 Schematische Darstellung des benutzten Algorithmus.

1. Da (78) linear ist, erhält man die Lösung für mehrere angelegte Randkräfte durch Superposition der elementaren Lösung (102). Wie in DLA, wo jeder Punkt, der auf einem nicht-verschwindenden Potential festgehalten wird, eine Quelle für Random-Walker darstellt, wird auch hier von jedem Punkt, an dem eine äußere Kraft anliegt, ein Random-Walker gestartet. Von jedem Punkt $\vec{r}_i$, an dem eine Kraft $\vec{F}_i$ anliegt, wird mit gleicher Wahrscheinlichkeit jeweils ein einzelner Walker gestartet, der einen isotropen Walk ausführt, bis er auf ein Teilchen eines Risses bei $\vec{r}_f$ trifft. Danach berechnet man nach (100–102) den Spannungstensor $\underline{\underline{\sigma}}\left(\vec{r}_f - \vec{r}_i, \vec{F}_i\right)$ und akkumuliert diese Spannung in einem Zähler des entsprechenden Oberflächenelementes
2. Ein Riß wird dargestellt als ein Cluster von Teilchen mit Radius Eins. Wie in DLA wird das Wachstum eines Clusters initiiert durch Anfangs Teilchen, die als Micro-Cracks fungieren. Da wir starke Fluktuationen während des Rißwachstums erwarten, müssen wir einen Mechanismus einführen, mit dem wir diese Fluktuationen unterdrücken können, damit wir eine bessere Abschätzung für das tatsächliche Spannungsfeld erhalten. Da allerdings aus DLA bekannt ist (vgl. auch Kapitel 2), daß jegliches Brechen der Isotropie des Raumes, sei es durch eine Antennenmethode [39,100] oder auch durch Einführung eines Gitters [32], die Struktur der entstehenden Cluster stark verändert, müßen wir sorgfältig darauf achten, keine Raumrichtung zu bevorzugen. Außerdem müssen wir betonen, daß gerade im Laplaceschen Wachstum auf einem Gitter eine Noise Reduction zur Unterdrückung der statistischen Fluktuationen zu extremen Anisotropien führt [36,37,39]! Zwar ergeben Untersuchungen von Rißwachstum, bei denen eine Noise-Reduction Prozedur benutzt wird, keinen Hinweis auf starke Gittereffekte [133], allerdings können wir den Einfluß der Random-Walks nicht abschätzen. Um mögliche Gittereffekte zu vermeiden, formulieren wir daher unsere Methode zur Unterdrückung statistischer Fluktuationen richtungsunabhängig. Daher wird der Crack wie in Bild 40 angedeutet aufgebaut: Jedes Teilchen repräsentiert zwei Zähler — jeweils einer für eine Rißoberfläche —, in denen die Spannungen akkumuliert werden, die eintreffende Walker mitbringen. Lediglich ein Teilchen an einer Rißspitze repräsentiert nur einen Zähler, da hier eine Definition zweier Rißseiten nicht sinnvoll erscheint.
3. Die korrekte Randbedingung (80) wird durch die eben berechnete Spannung (102) im allgemeinen offensichtlich nicht erfüllt, sondern eine unkompensierte Kraft $\vec{F}_{unbal} = \underline{\underline{\sigma}} \cdot \vec{n}$ wirkt auf die Oberfläche. Daher legen wir eine virtuelle, kompensierende Kraft

$$\vec{F}_{bal} = -\underline{\underline{\sigma}} \cdot \vec{n} \tag{103}$$

an dem entsprechenden Oberflächenelement bei $\vec{r}_f$ an, welches exakt die überflüssige Scherkomponente τ und die Spannungskomponente senkrecht zur Rißoberfläche $\sigma_\perp$ kompensiert. Daher ist es prinzipiell ausreichend, lediglich die Spannungskomponente parallel zur Rißoberfläche $\sigma_\parallel$ in dem Oberflächen-Zähler zu akkumulieren. Aber wir werden später zeigen, daß es notwendig ist, auch die Scherkomponente zu akkumulieren, damit die oben beschriebenen Anisotropie-Effekte vermieden werden. Also werden sowohl τ als auch $\sigma_\parallel$ akkumuliert. Wie wir oben gesagt haben, soll ein Walker von jedem Punkt gestartet werden, an dem eine Kraft angelegt ist. Somit muß der Walker von $\vec{r}_f$ aus neu gestartet werden, diesmal allerdings mit der neuen Kraft $\vec{F}_{bal}$. Dieser Prozeß des Neu-Startens wird wiederholt, bis der Betrag der Kraft $\vec{F}_{bal}$ unter einen kritischen Wert gefallen ist und nur noch einen vernachlässigbaren Beitrag zur Gesamtsumme liefert. Dieser Wert ist in unseren Simulationen typischerweise auf 10^{-8} gesetzt. Danach

wird dieser Walker vernichtet und eine neuer Walker vom Rand aus gestartet. Dieser wiederholte Vorgang kann interpretiert werden als eine Methode, die Spannungen auf der Rißoberfläche zu relaxieren. Er kann am einfachsten mit Hilfe einer Rand-Integral Methode verstanden werden [44]. Eine ähnliche Situation kommt auch in DLA bei Simulationen von Viscous Fingering mit nicht-verschwindender Oberflächenspannung [134–136] oder in Simulationen von Elektrodeposition mit endlichem Anodenwiderstand [137] vor. Beide Fälle können auf eine Sticking-Probability-DLA abgebildet werden, bei der die Wahrscheinlichkeit, daß ein Walker nicht an der Clusteroberfläche haftet, sondern wieder in das System zurückgeschickt wird, durch den Krümmungsradius bzw. die Oberflächenladungsdichte bestimmt wird. Physikalisch kann man diesen Vorgang als eine Mehrfachstreuung der Walker an der Rißoberfläche interpretieren, wobei jeder Streuvorgang einem Term höherer Ordnung in der Randintegral-Methode entspricht.

4. Als nächstes haben wir die Wachstumsregel zu definieren. Es muß betont werden, daß diese Regel nicht davon bestimmt werden darf, wie oft ein bestimmter Zähler von einem Walker erreicht wird, sondern alleine die akkumulierte Spannung soll bestimmen, wann der Crack wachsen soll. Da der Spannungstensor symmetrisch ist, kann man reelle Eigenwerte und orthogonale Eigenvektoren bestimmen, welche die Hauptspannungen S_I und S_{II} und die Hauptspannungsrichtungen definieren. Die Eigenwerte erlauben es auch, Druck- und Zugspannungen zu unterscheiden, die ein entgegengesetztes Vorzeichen haben. Diese beiden Hauptspannungen werden nun mit kritischen Spannungen für Zug σ_t und für Druck σ_c verglichen. Diese Spannungen repräsentieren die Materialfestigkeit gegenüber den beiden Belastungsarten, die wie oben bereits erwähnt direkt mit den Kohäsionskräften verbunden sind. Wenn einer der beiden kritischen Spannungen überschritten wird, wächst der Riß indem ein Teilchen zu dem Crack in der Richtung des anderen Eigenwertes hinzugefügt wird[10]. Diese Bruchregel ist so gewählt, daß ein Riß unter uniaxialer Last in der Richtung senkrecht zu dieser Last wächst. Man muß anmerken, daß in dieser Regel lediglich die Hauptspannungen berücksichtigt werden. Diese Situation ist vergleichbar mit dem Central-Force Modell [76], in dem auch nur die reine Zug- oder Druckbelastung eines Bonds relevant ist und ebenfalls Scherkräfte erst durch Einführung eines Dreiecksgitters in Zug- oder Drucklasten übersetzt werden müssen. Diese Bruchregel stimmt im wesentlichen mit einem erweiterten Griffith-Kriterium [74] überein. Allerdings haben wir die Mohr-Coulomb-Regel

$$\frac{S_I}{\sigma_t} - \frac{S_{II}}{\sigma_c} \geq 1, \tag{104}$$

die ein Bruchkriterium für solche Volumenelemente darstellt, die in einer Richtung unter Drucklast und in der dazu senkrechten Richtung unter Zuglast stehen, nicht berücksichtigt. Der Grund hierfür ist, daß zunächst die prinzipiellen kollektiven Eigenschaften dieses neuen Modells studiert werden sollen, bevor in einer weiteren Entwicklung feinere Details implementiert werden. In der Bruchregel benutzen wir die Eigenvektoren des Spannungstensors zur Bestimmung der Wachstumsrichtung. Diese Eigenvektoren bestimmen aber gerade jene Richtungen, in denen die Außerdiagonalelemente des Spannungstensors, also die Scherkomponenten, verschwinden. Physikalisch bedeutet dies, daß bereits

[10] Größere Festigkeiten implizieren, daß mehr Walker ein Teilchen erreichen müssen, bevor der Riß wachsen kann. Damit sind die Festigkeiten in einer DLA-nahen Sprache mit dem "Noise-Reduction" Parameter m verknüpft.

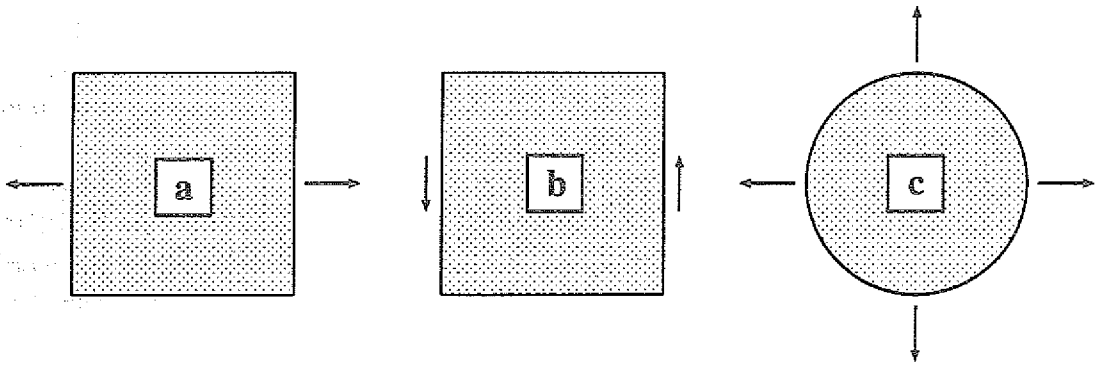


Abbildung 41 Simulierte Geometrien: a) uniaxialer Zug, b) Scherung, c) isotrope Dilatation.

beim Rißwachstum lokal das Auftreten von Scherlast auf der Rißoberfläche verhindert wird. Hier wird nun klar, warum es essentiell wichtig ist, auch die Scherkomponenten zur Bestimmung der Wachstumsrichtung zu benutzen. Würde lediglich $\sigma_{\|}$ benutzt und τ vernachlässigt, so würde die Wachstumsrichtung vollkommen durch die Oberflächen des bereits existierenden Risses bestimmt und die Isotropie des Raumes wäre massiv gebrochen.

5. Sobald eine der Hauptspannungen S_I oder S_{II} die Festigkeit überschreitet, wird dem Cluster ein Teilchen hinzugefügt. Allerdings muß durch diese Rißpropagation lokal Spannung relaxiert werden, was durch Löschen von Spannungzählern geschehen soll. Wir werden zwei Simulations-Varianten untersuchen.

   - In der ersten Variante wird die Spannung nur lokal relaxiert, d. h. nur genau jener Zähler gelöscht, der die Festigkeit überschritten hat. Alle anderen Spannungen werden nicht relaxiert. Man beachte, daß in dieser Variante unser Simuliertes Medium ein sehr starkes "Spannungs-Gedächtnis" hat, so daß selbst kleine Spannungen, die über einen langen Zeitraum wirken, das Material brechen können. Physikalisch kann dies als ein irreversibler Korrosionsvorgang interpretiert werden, bei dem ein Material durch eine andauernde (auch kleine) Spannung ermüdet.
   - In der zweiten Variante wird neben dieser lokalen Relaxation, die in jedem Bruchvorgang durchgeführt wird, nach jeweils M angelagerten Teilchen der gesamte Riß global relaxiert, indem alle Zähler gelöscht werden und das gesamte Spannungsfeld neu bestimmt wird. Der Ermüdungs-Parameter M wird als freier Parameter benutzt. Man beachte, daß sich die erste Variante auf natürliche Weise als $M = \infty$ darstellen läßt.

7.4 Resultate

Im folgenden werden wir die Resultate vorstellen, die wir in Simulationen mit der eben beschriebenen Methode erhalten haben. Wir haben dabei mehrere verschiedene Randbedingungen studiert: uniaxialen Zug, Scherung und isotrope Dilatation (Abb. 41). Um unsere Methode mit experimentellen Ergebnissen vergleichen zu können, haben wir zusätzlich zwei in Experimenten übliche Geometrien simuliert [75,138]. Darüber hinaus haben wir den Einfluß mehrerer Modell-Parameter untersucht.

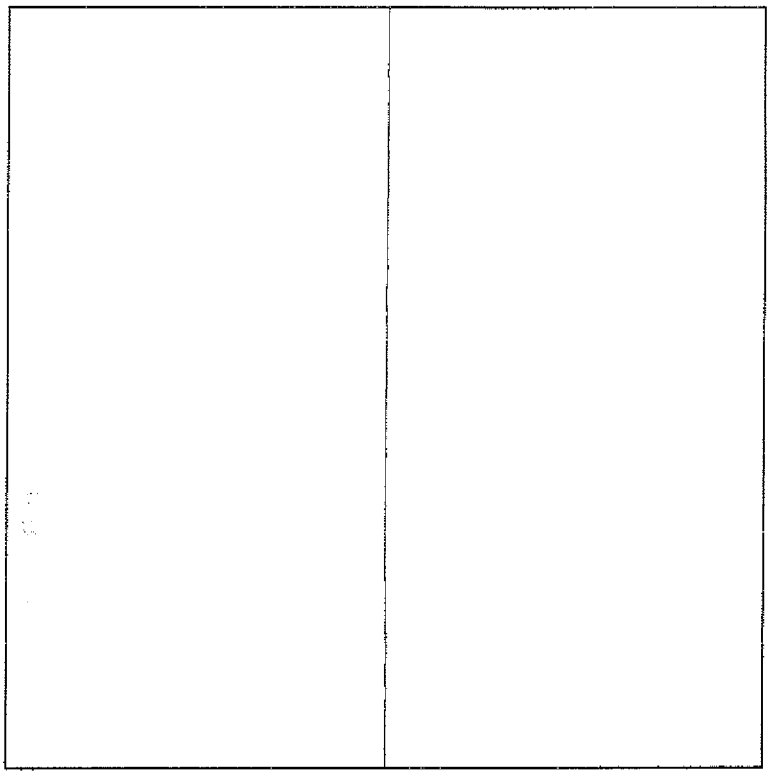


Abbildung 42 Uniaxialer Zug. Zähler werden nach jeweils 10 Teilchen gelöscht. Der Riß hat hier eine einfache, eindimensionale und faden-artige Struktur.

Grundlegende Resultate

Im folgenden werden wir die Resultate zeigen, die wir in Simulationen von uniaxialem Zug, Scherung und isotroper Dilatation erhalten haben. Die kritischen Spannung sind dabei gewählt als $\sigma_t = 20$ und $\sigma_c = -20$. Wir simulieren hier also ein Material, daß gleichermaßen unter Druck wie unter Zug bricht. Da die Spannungs-Komponenten parallel zur Rißoberfläche, die ein Walker zum gesamten Mittelwert beiträgt, typischerweise in der Größe von 0.5 ist, müssen wenigstens $N = 40$ Walker einen bestimmten Platz erreichen, bevor der Riß dort wachsen kann. Als Quelle für das Rißwachstum werden zwei Teilchen bei $\vec{r} = (0, 0.5)$ und $\vec{r} = (0, -0.5)$ als Micro-Cracks benutzt. Die erzielten Resultate hängen jedoch nicht von dieser speziell gewählten Anfangsgeometrie ab. Man kann eine beliebige andere Geometrie wählen, ohne die Ergebnisse zu ändern. Die Poisson-Zahl ist in diesen Simulationen gewählt als $\nu = 0.2$, was der realen Poisson-Zahl von Beton entspricht. Den Ermüdungs-Parameter haben wir als $M = 10$ gewählt, was einen Kompromiß darstellt, da ein kleines M die benötigte Simulationszeit rapide ansteigen läßt. Alle Risse wurden bis zu einer Größe von 6.000 Teilchen gewachsen.

In Bild 42 zeigen wir ein typisches Resultat der Simulation eines einachsigen Zuges. Die externen Kräfte sind dabei entlang der beiden Linien $\vec{r} = (\pm 2000, y)$, $y \in [-2000, 2000]$ angeordnet und zeigen alle in die positive bzw. negative x Richtung. Man erhält einen schmalen geraden Riß, der senkrecht zur Lastrichtung wächst. Dieses Resultat entspricht der physikalischen Erwartung und kann leicht in anderen Simulationen und Experimenten beobachtet werden [128,138]. Im Gegensatz zu DLA Simulationen, ist die Bildung von Seitenzweigen stark unterdrückt — d. h. in dieser Geometrie und in dieser Rißgröße ist sie vollkommen verhindert.

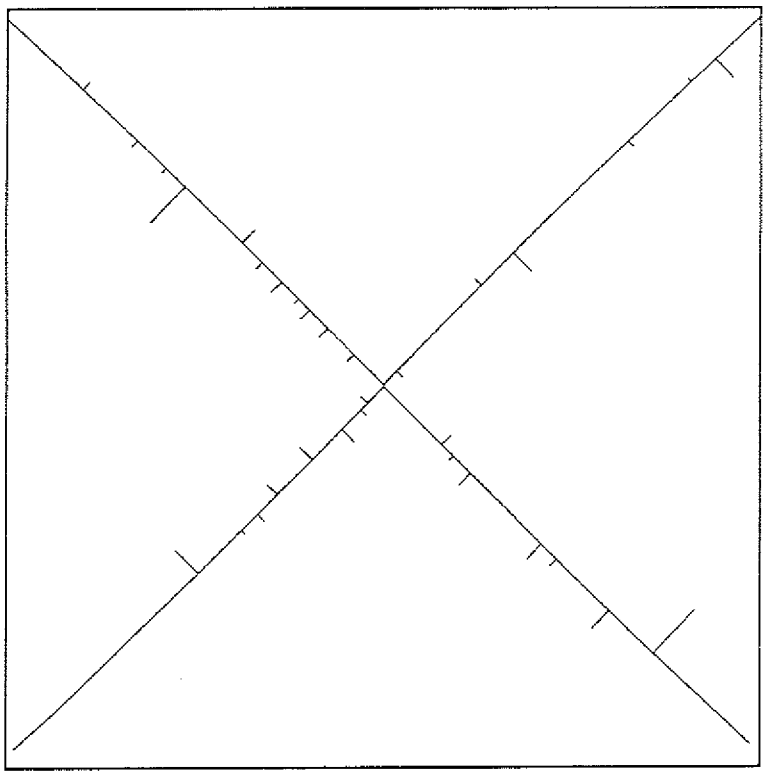


Abbildung 43 Scherbelastung. Zähler werden nach jeweils 10 Teilchen gelöscht.

In Bild 43 zeigen wir ein typisches Ergebnis einer Simulation mit reiner Scherlast. In dieser Rechnung sind alle Kräfte wieder entlang der Linie $\vec{r} = (\pm 2000, y)$ angeordnet, aber die Kraftrichtung ist jetzt die positive bzw. negative y Richtung. Obwohl wir eine Anfangsgeometrie wie in der vorherigen Simulation benutzen, ist die Form des Risses hier völlig verändert, was eine Folge des vektoriellen Charakters der Lamé-Gleichung ist. Der entstandene Riß zeigt im wesentlichen zwei senkrecht aufeinander stehende gekreuzte Hauptrisse, die im Winkel von 45° zur Lastrichtung wachsen. Hier erhält man eine symmetrische Struktur der beiden gekreuzten Hauptrisse, da wir symmetrische Druck- und Zugfestigkeiten gewählt haben $\sigma_c = -\sigma_t$. Diese Struktur der gekreuzten Risse bei symmetrischen Festigkeiten ist auch in anderen Simulationen von kleineren Gittermodellen beobachtet worden [82,83]. Experimentell beobachtet man hauptsächlich immer nur ein diagonales Scherband, da in realen Materialien die Druckfestigkeit meist wesentlich größer als die Zugfestigkeit ist. Auf diesen Fall wird später noch eingegangen.

Neben den gekreuzten Hauptrissen beobachtet man zusätzlich noch kleine Seitenrisse, die senkrecht auf den Hauptrissen stehen. Wir müssen hier betonen, daß dieses Resultat die erste bekannte Simulation ist, die kein Gitter benutzt und trotzdem ein derart ausgeprägtes Verhalten zeigt. In früheren Simulationen von Rißwachstum auf dem Gitter ist der Effekt des Side-Branching zwar zu beobachten, allerdings sind die hier auftretenden Winkel nicht sehr genau definiert: Die simulierten Rißgrößen sind viel zu klein, um wohldefinierte Winkel beobachten zu können.

Außerdem muß darauf hingewiesen werden, daß die hier beobachteten Winkel nicht auf Gittereffekte zurückgeführt werden können. Die benutzte Wachstumsregel ist isotrop mit Hilfe der Hauptspannungen formuliert, und auch für die Bestimmung der Wachstumsrichtung werden nicht nur die Spannungen $\sigma_{\|}$ parallel zur Rißoberfläche, sondern auch die Scherspannungen

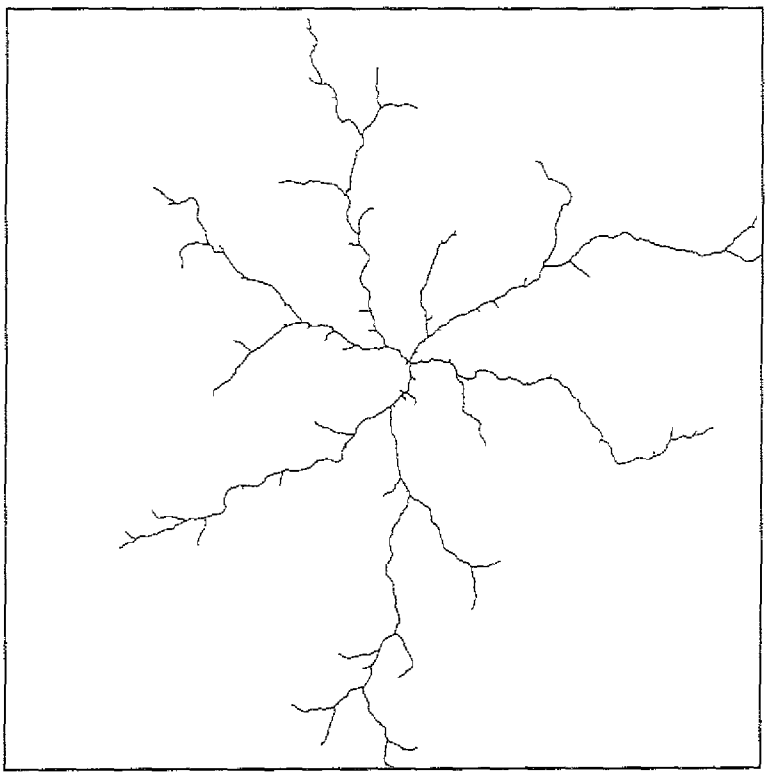


Abbildung 44 Isotroper Zug. Zähler werden nach jeweils 10 Teilchen gelöscht.

τ benutzt. Dadurch wird ein triviales Wachstum senkrecht zur ursprünglichen Rißrichtung vermieden.

Die Ausbildung rechtwinkliger Seitenzweige ist aus physikalischer Sicht wohlbegründet und wird von der Randbedingung (80) gefordert: Da keinerlei Rückstellkräfte auf die Rißoberfläche wirken, sind die einzigen Spannungen, die sich entlang der Rißoberfläche ausbilden können, parallel zu dieser gerichtet. Daher kann ein Seitenriß anfangs auch lediglich senkrecht aus einem Hauptriß herauswachsen. Erst ab einem gewissen Abstand zu der Oberfläche des Hauptrisses macht sich dessen relaxierende Wirkung nicht mehr bemerkbar, und der Seitenriß kann seine Richtung ändern oder selbst wieder Seitenrisse entwickeln. Dieses Verhalten kann sehr deutlich in Experimenten von Van Damme et al. [68] beobachtet werden.

In der Abbildung 44 zeigen wir ein typisches Resultat einer Simulation unter isotroper Dilatation. Dieser Fall ist besonders hervorzuheben, denn hier wird durch die Anordnung der Kräfte keine zusätzliche Anisotropie eingeführt, die möglicherweise die resultierende Struktur beeinflussen könnte. Die Kräfte sind hier entlang eines Kreises mit Radius $R = 2000$ angeordnet und weisen radial nach außen. Ein typisches Resultat dieser Simulation ist in Bild 47 gezeigt. Man erhält eine verzweigte, isotrope Struktur, die Eigenschaften aufweist, die sie klar von DLA abgrenzt. Man findet keine für DLA typische Tip-Splitting Instabilität und die Bildung von Seitenzweigen ist stark unterdrückt. Die Seitenzweige scheinen auch hier zunächst senkrecht aus den Hauptzweigen herauszuwachsen. Erst nachdem sie eine gewisse Länge erreicht haben ist der Einfluß der Randbedingung (80) nicht mehr dominierend, und sie können die Richtung ändern. Allerdings kann kein derart ausgeprägtes Verhalten zur Bildung rechter Winkel wie im Fall der Scherung beobachtet werden. Simulationen unter isotropem Zug wurden bisher nur mit wesentlich kleineren Systemen durchgeführt [82,83].

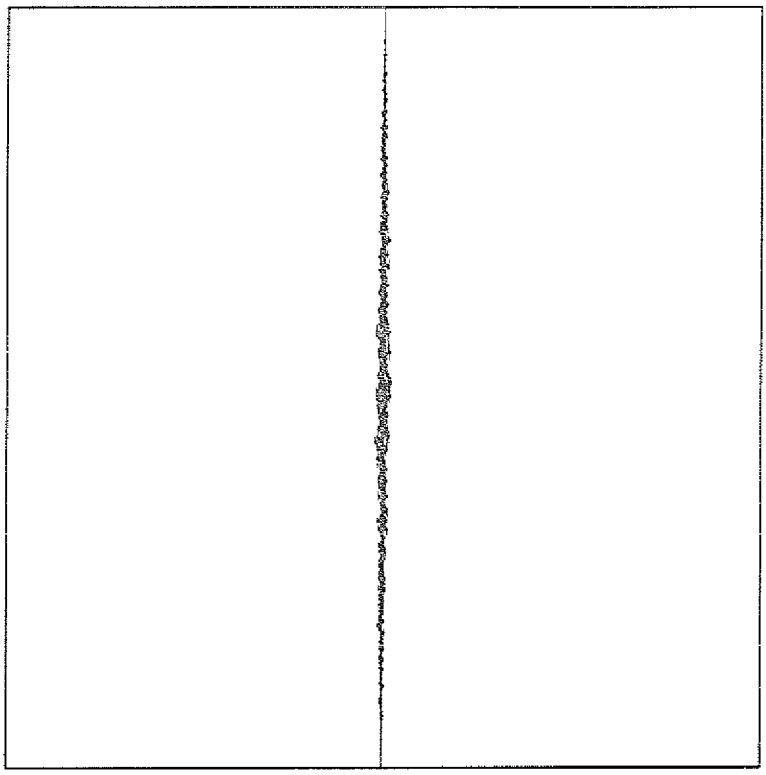


Abbildung 45 Ergebnis unter uniaxialem Zug.

Die dort generierten Strukturen zeigen gewisse Ähnlichkeiten mit unseren Rissen. Ein direkter Vergleich scheint wegen der Größenunterschiede allerdings nicht sinnvoll.

Ermüdungs Effekte

Die eben geschilderten Resultate wurden für einen Ermüdungs-Parameter $M = 10$ durchgeführt und in diesem Abschnitt wollen wir dessen Einfluß untersuchen. Dazu setzen wir hier, wie auch in allen folgenden Abschnitten $M = \infty$, halten aber alle anderen Parameter fest, d. h. $\sigma_t = 20$ und $\sigma_c = -20$. Aufgrund des größeren Ermüdungs-Parameters können wir nun wesentlich größere Risse simulieren. Eine Erhöhung des Ermüdungs-Parameters bei gleichbleibender Festigkeit bedeutet physikalisch, daß sich im Material im Laufe der Zeit Ermüdungsbrüche entwickeln können, da auch kleinere Lasten, die das Material nicht sofort brechen, es aber trotzdem schädigen, sich zu Rissen akkumulieren können. Aus rein technischen Gründen — die Simulation mit $M = \infty$ benötigt wesentlich weniger Rechenzeit als $M = 10$ — können wir hier größere Risse generieren und zeigen im folgenden Risse mit 20.000 gebrochenen Bonds. Für den Fall des isotropen Zuges (Abb. 47) findet man wieder einen geraden schmalen Hauptriß, der senkrecht zur Richtung der angelegten Kraft wächst. Neben dem dominierenden Hauptriß beobachtet man noch viele kleine Seitenrisse, die alle in einem ziemlich genauen rechten Winkel aus dem Hauptriß heraus wachsen. Hier muß wieder betont werden, daß diese stabilen rechten Winkel nicht hervorgerufen werden durch irgendeine zugrundeliegende Anisotropie, sondern wichtiger Bestandteil des benutzten Modells sind. Allerdings werden sie wie eben gezeigt durch Ermüdungs-Effekte hervorgerufen. Weiter fällt auf, daß selbst die Seitenrisse wieder weitere Seitenrisse erzeugen können, die parallel zum Hauptriß wachsen.

Unter Scherlast (Abb. 46) findet man wieder zwei gekreuzte, im Winkel von 45° zur Lastrichtung stehende Hauptrisse. Wie bereits in der vorigen Geometrie führt auch hier die

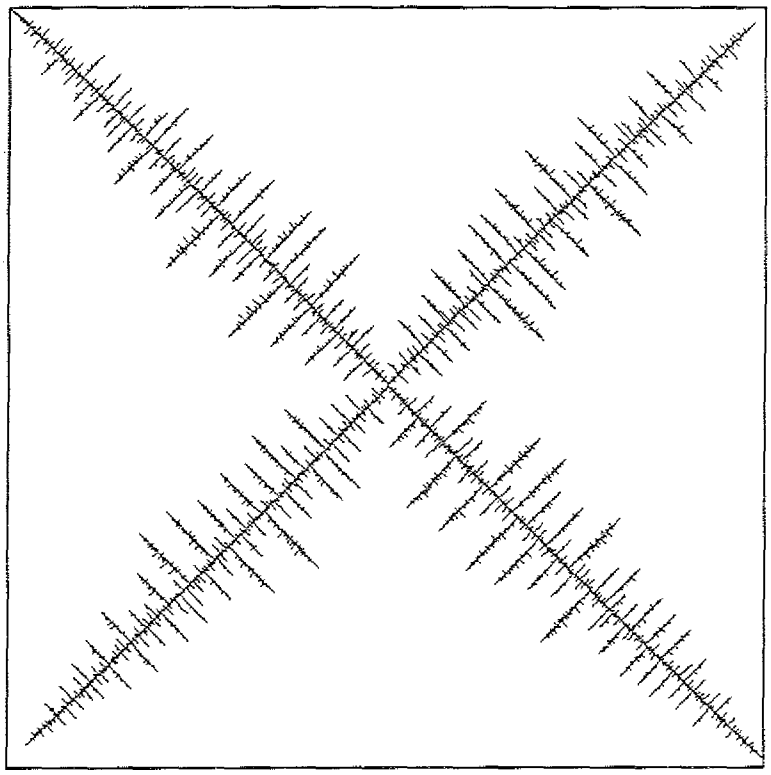


Abbildung 46 Ergebnis der Scherbelastung.

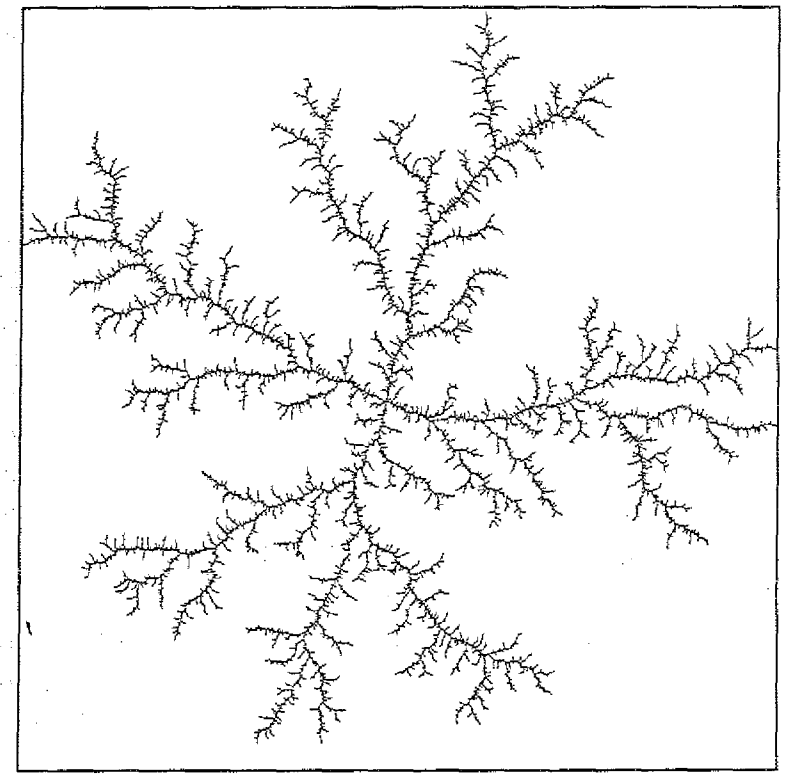


Abbildung 47 Ergebnis der isotropen Dilatation.

Einführung von Materialermüdung zur Ausbildung vieler kleiner Seitenrisse, die sehr stabil im rechten Winkel auf den Hauptrissen stehen. Besonders interessant ist die Stabilität der Seitenrisse, die der Gesamtstruktur ein dendritisches Aussehen geben.

Sehr interessant ist das Ergebnis unter isotropem Zug (Abb. 47). Wie in den beiden vorigen Geometrien findet man auch hier einen Hauptriß, der mit einer Vielzahl von kleinen Seitenrissen übersät ist. Allerdings muß man beachten, daß einige Details auch diesen Riß deutlich von einem DLA Cluster gleicher Größe unterscheiden. Zunächst fehlt die für DLA typische Tip-Splitting Instabilität, sondern alle Zweige scheinen erst zu einer bestimmten Länge angewachsen zu sein, bevor Seitenzweige *hinter* der Rißspitze auftauchen. Diese Seitenzweige wachsen auch hier senkrecht aus den Hauptzweigen heraus. Erst nachdem sie eine gewisse Länge erreicht haben ist der Einfluß der Randbedingung (80) nicht mehr dominierend und sie können die Richtung ändern. Zusätzlich scheint es, daß die Seitenzweige von stark gekrümmten Hauptzweigen im wesentlichen auf der Außenseite der Krümmung auftauchen. Alle drei Beobachtungen sind in Übereinstimmung mit experimentellen Resultaten. Wir halten es für sehr wichtig, daß unser Modell diese Eigenschaften, die Grundeigenschaften von realen Cracks sind, aber in dieser Form in bisherigen Gittermodellen noch nicht beobachtet wurden, reproduziert.

Variation der Festigkeiten

In diesem Abschnitt wollen wir den Einfluß der Materialfestigkeiten σ_t und σ_c auf die Resultate der Simulationen studieren. Eine erhöhte Festigkeit führt — in unserer Random-Walker Interpretation — dazu, daß eine größere Anzahl von Walkern ein bestimmtes Clusterteilchen erreichen muß, bevor der Riß dort wachsen kann. In einer DLA-nahen Sprache heißt das, daß der "Noise-Reduction[11] Parameter" erhöht wird. Als Beispiele diskutieren wir die extremen Fälle $\sigma_c = \sigma_t = 0$ bzw. $\sigma_c = -\sigma_t = -100$. Physikalisch bedeutet eine Variation der Festigkeiten, daß das simulierte Material erst ab einer veränderten Gesamtlast bricht. Da wir wieder $M = \infty$ benutzen, muß man beachten, daß sich auch kleinere Lasten zu Rissen akkumulieren können.

In den Bildern 48–50 zeigen wir die Ergebnisse für $\sigma_c = \sigma_t = 0$. Wenn beide Festigkeiten verschwinden, wächst der Riß sobald er von einem Random-Walker berührt wird. Diesen Fall kann man als DLA-Grenzfall des Rißwachstums ansehen, da das Rißwachstum vom Diffusionsfeld der Walker bestimmt wird. Man muß aber beachten, daß die Richtung des Wachstums immer noch von den extern angelegten Kräften bestimmt wird und daher eine Anisotropie beim Wachstum vorhanden sein muß. Unter uniaxialem Zug (Abb. 48) erhält man im wesentlichen wieder die schmale langgestreckte Rißform der oben dargestellten Simulationen, aber die feine Struktur und Anzahl der Seitenzweige ist verändert. Sowohl die gerade und langgestreckte Form der Seitenrisse, als auch die bemerkenswerten rechten Winkel zwischen Haupt- und Seitenrissen sind verschwunden. Die Risse sehen jetzt DLA Clustern mit anisotroper Wachstumswahrscheinlichkeit sehr ähnlich [139].

In Simulationen unter Scherspannung (Abb. 49) bleibt die globale Form der Risse mit zwei gekreuzten Hauptzweigen und dendritischen Nebenzweigen immer noch vorhanden. Im Gegensatz zur vorherigen Simulation sind die beobachteten rechten Winkel sehr stabil und verschwinden auch im Grenzfall verschwindender Festigkeit nicht. Erstaunlich ist, daß auch die dendritische Form selbst in diesem Grenzfall stabil ist und nicht zerstört wird.

---

[11] Der Begriff Noise Reduction wird hier so wenig wie möglich verwendet, da er aus DLA Rechnungen vorbelastet ist und meist mit Gittereffekten in Verbindung gebracht wird. Es muß daher betont werden, daß in unseren Simulationen keine Gittereffekte auftreten können.

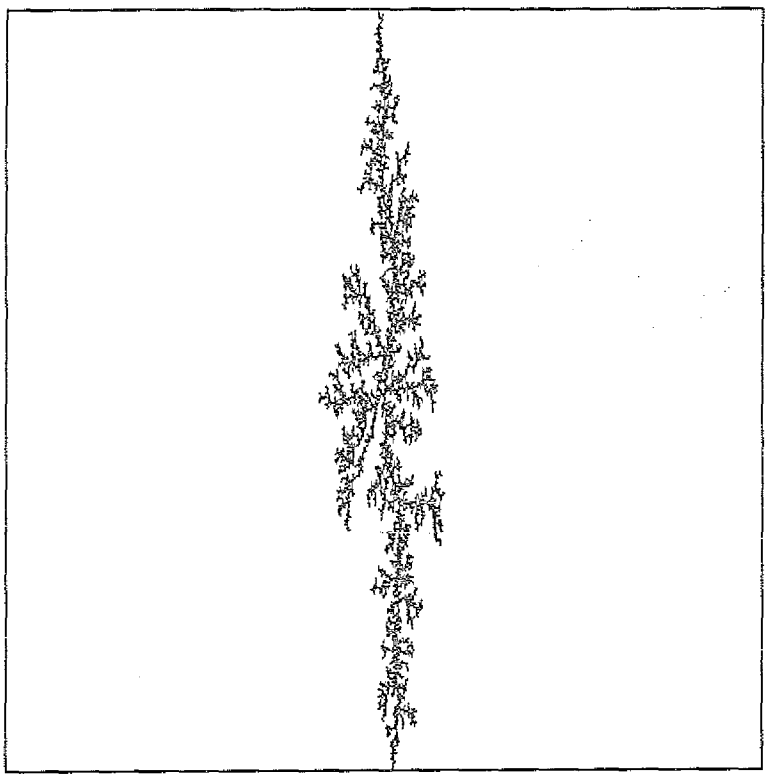


Abbildung 48 Uniaxialer Zug bei verschwindender Festigkeit.

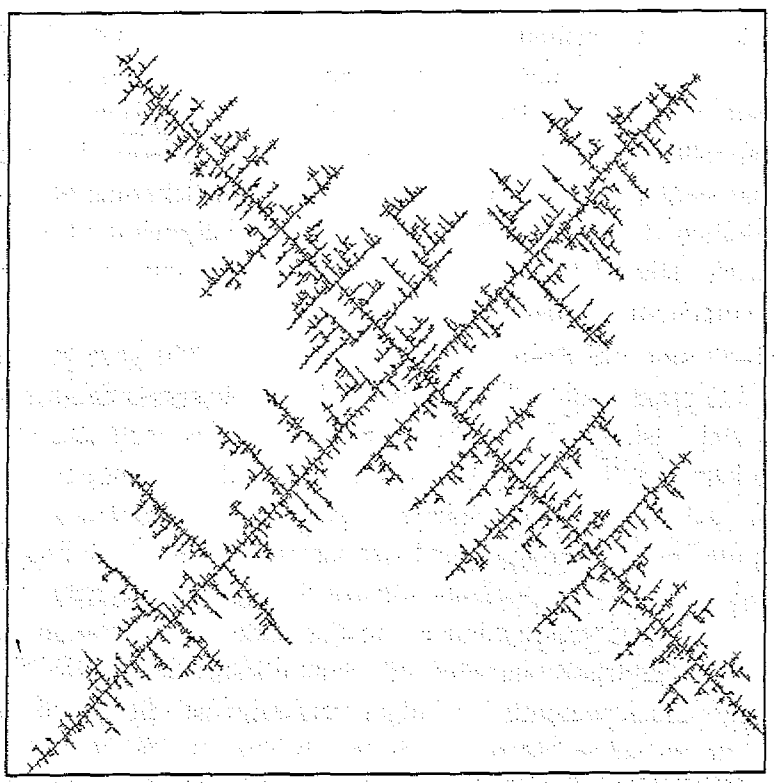


Abbildung 49 Scherspannung bei verschwindender Festigkeit.

Im Fall isotroper Dilatation (Abb. 50) sind alle Rißeigenschaften, die wir vorher so betont haben, verschwunden. Man findet weder rechte Winkel, noch Side Branching, noch ein Abschirmen von gekrümmten Rissen. Der entstandene Riß ähnelt lediglich einem typischen DLA Cluster.

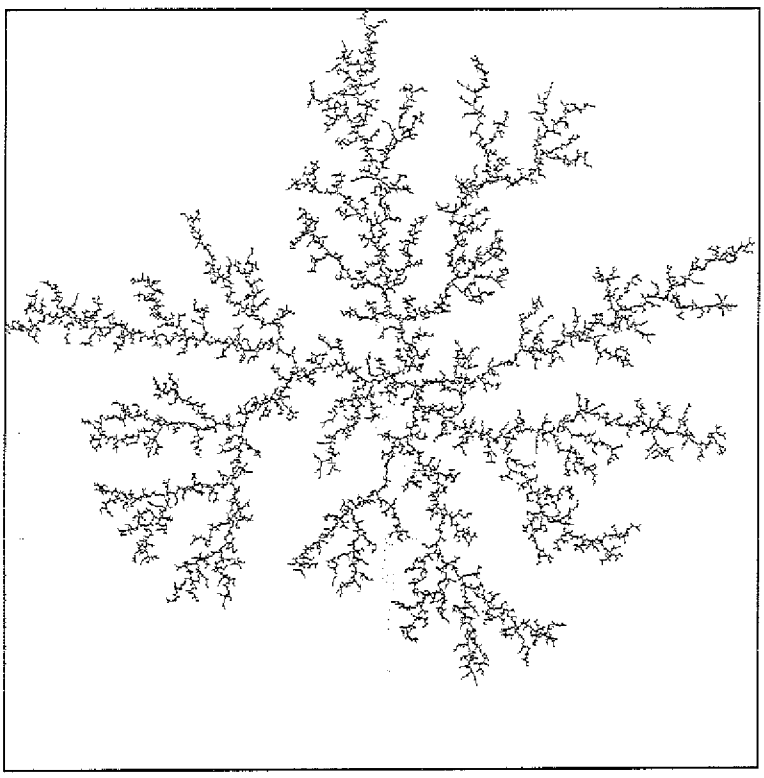


Abbildung 50 Isotrope Dilatation bei verschwindender Festigkeit.

Zusammenfassend kann man festhalten, daß im Grenzfall verschwindender Festigkeit unser Modell nicht mehr nur durch Material-Spannungen, sondern stark durch die Diffusion der Random-Walker bestimmt wird und damit im wesentlichen eine anisotrope Version der wohlbekannten DLA darstellt. Allerdings bleibt der vektorielle Charakter der zugrundeliegenden Physik erhalten. Das sieht man deutlich an den Resultaten für Scherung bzw. uniaxialen Zug: Man erhält grundsätzlich verschiedene Strukturen, obwohl lediglich die Richtung der äußeren Kräfte verschieden ist. Hier sieht man, daß man immer noch eine wesentliche Erweiterung der skalare DLA vorgenommen hat.

In den Bildern 51–53 sind die Resultate für $|\sigma_c| = |\sigma_t| = 100$ gezeigt. In der Scherlast-Simulation (Abb. 51) erhält man auch hier die früher beobachtete dendritische Gestalt, die offensichtlich sehr stabil ist, da sie sowohl für sehr kleine, als auch sehr große Festigkeiten beobachtet werden kann. Weder die Dichte noch die Form der Nebenrisse hat sich gegenüber den Resultaten für $|\sigma_c| = |\sigma_t| = 20$ geändert. Auch für den Fall des einachsigen Zuges (Abb. 52) hat sich die gesamte Struktur der Risse kaum geändert. Man findet selbst für diese hohe Festigkeit immer noch eine Vielzahl kleiner Seitenrisse und auch die Risse parallel zum Haupt Crack sind nicht verschwunden. In der isotropen Dilatation (Abb. 53) findet man, daß sowohl die Hauptrisse, als auch die Seitenzweige wesentlich gerader geworden sind und der gesamte Crack wesentlich weniger verzweigt ist. Ebenso findet man, daß die rechtwinklige Struktur zwischen Haupt und Seitenzweigen wesentlich ausgeprägter ist. Wir werden uns später noch einmal quantitativ mit dieser "rechtwinkligen" Struktur beschäftigen. Obwohl die einzelnen Zweige weniger stark verzweigt sind, hat die Anzahl der Seitenzweige allerdings nicht abgenommen. Statt dessen scheinen nun alle größeren Zweige mit feinen Haaren überzogen zu sein.

Insgesamt findet man, daß die Struktur der Resultate besonders für die Fälle Scherung und uniaxialer Zug kaum von der Materialfestigkeit beeinflußt werden. Das Auftreten von

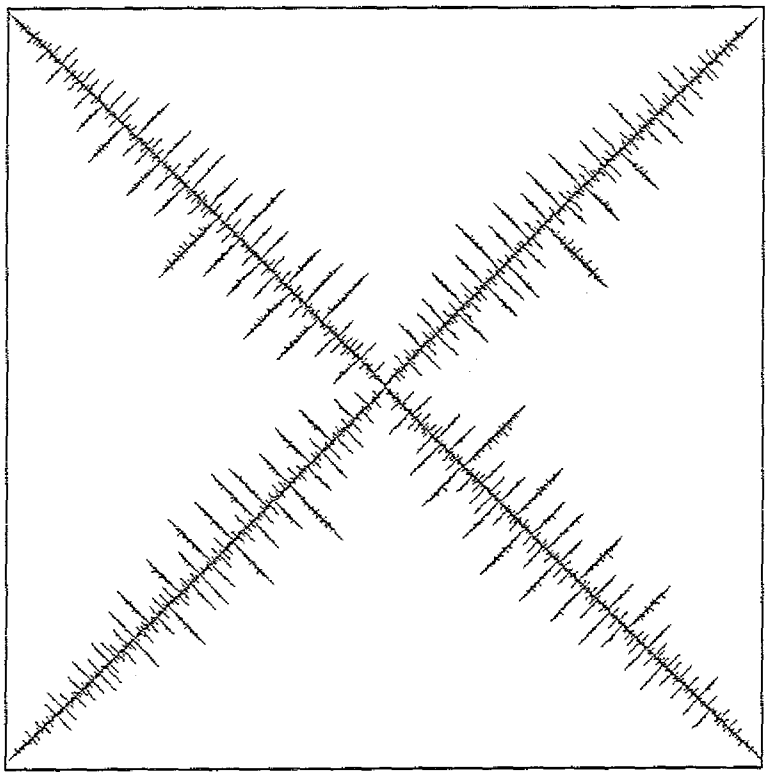


Abbildung 51 Scherlast bei hoher Festigkeit.

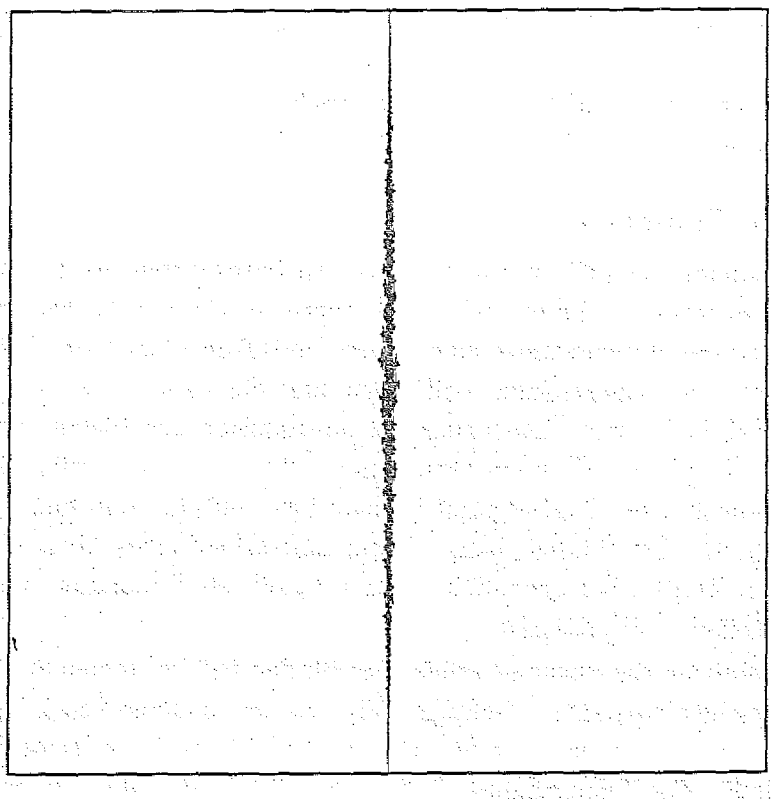


Abbildung 52 Uniaxialer Zug bei hoher Festigkeit.

unphysikalischen Seitenzweigen kann nicht durch eine vergrößerte Festigkeit unterdrückt werden. Dieser Effekt ist jedoch dadurch dominiert, daß das simulierte Material stark von Ermüdungsprozessen geprägt ist, die dazu führen, daß auch kleine Lasten sich zu Rissen akkumulieren können. Daher muß man festhalten und betonen, daß unter dem Einfluß

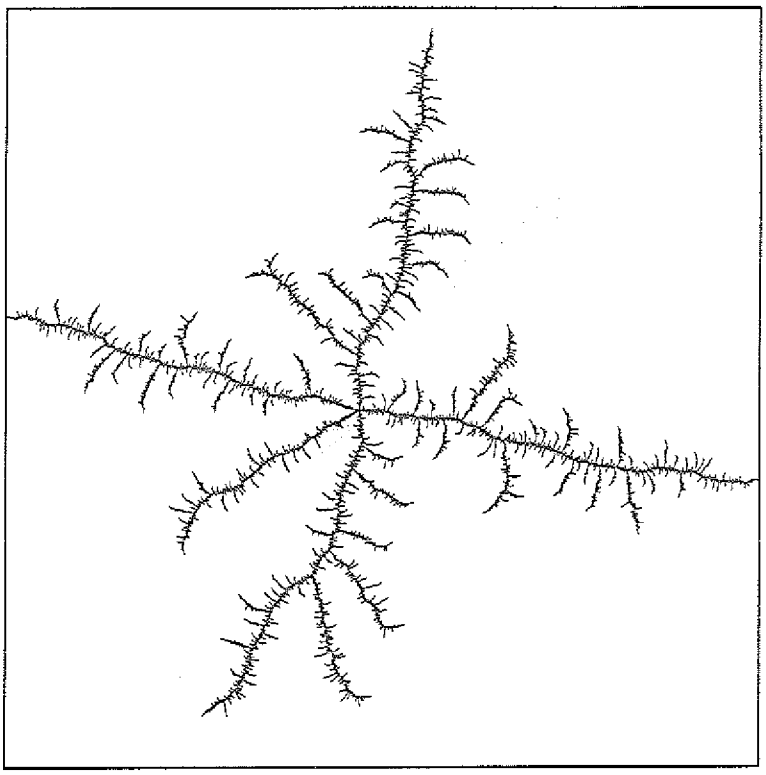


Abbildung 53 Isotrope Dilatation bei hoher Festigkeit.

von Ermüdungsprozessen die Materialfestigkeit in unserem Modell lediglich eine Zeitskala bestimmt, wann ein Material unter einer angelegten Last bricht und keine Spannungsskala definiert, unter welcher Mindestlast das Material bricht: Das simulierte Material wird wegen Materialermüdung selbst unter der kleinsten Last brechen.

Asymmetrische Festigkeit

Nach der Untersuchung der Fälle mit symmetrischen Festigkeiten, wollen wir nun den Fall asymmetrischer Festigkeit studieren, wie er in typischerweise in realistischen Materialien auftritt. Hier findet man üblicherweise eine höhere Festigkeit unter Druckbelastung als unter Zuglast $|\sigma_c| > |\sigma_t|$. Als Spezialfälle wollen wir hier die Fälle $\sigma_c = -\infty$, $\sigma_t = 20$ bzw. $\sigma_t = \infty$, $\sigma_c = -20$ betrachten. Das entspricht physikalisch den Fällen, daß das simulierte Medium nur unter Zug bzw. nur unter Druck und nicht unter der jeweils anderen Belastung bricht. Der Fall unendlicher Zugfestigkeit ist natürlich zunächst rein numerisch interessant. Da hier aber Zug- und Drucklasten lediglich ein unterschiedliches Vorzeichen haben, kann man den Fall einer Zuglast bei unendlicher Zugfestigkeit als Simulation einer Drucklast bei endlicher Zugfestigkeit interpretieren.

Den stärksten Einfluß auf die Resultate erhält man für den Fall der reinen Scherung (Abb. 54). Sobald hier ein Festigkeitsparameter betragsmäßig von dem anderen verschieden ist — bereits eine geringe Asymmetrie genügt — erhält man sofort eine asymmetrische Rißform, und im Grenzfall unendlicher Festigkeit bleibt lediglich einer der beiden Hauptzweige übrig und alle Nebenzweige verschwinden. Konsequenterweise ist nun auch die ausgeprägte dendritische Struktur verschwunden. Welcher der beiden Hauptzweige erhalten bleibt, wird eindeutig durch die Festigkeiten festgelegt und stimmt mit anderen numerischen Simulationen [82] überein.

Ähnliche Beobachtungen kann man bei uniaxialem Zug durchführen (Abb. 55). Bei unendlicher Druckfestigkeit verschwinden alle Seitenzweige und man erhält lediglich einen einzelnen

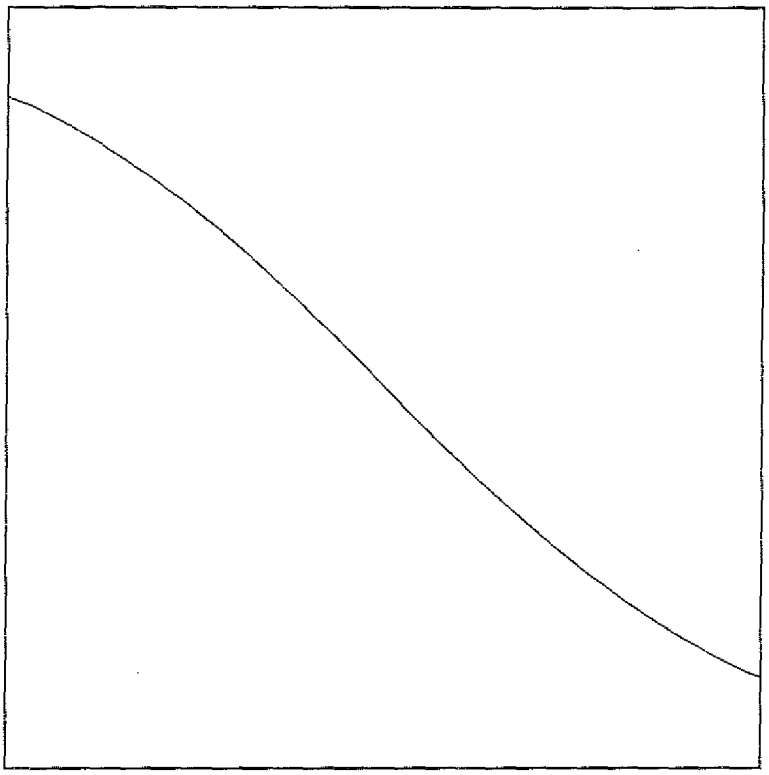


Abbildung 54 Scher-Simulation bei unendlicher Druckfestigkeit. Die systematische Rißkrümmung ist eine Folge von Randeffekten, da dieser Riß bis in die Nähe der angelegten Kräfte gewachsen ist.

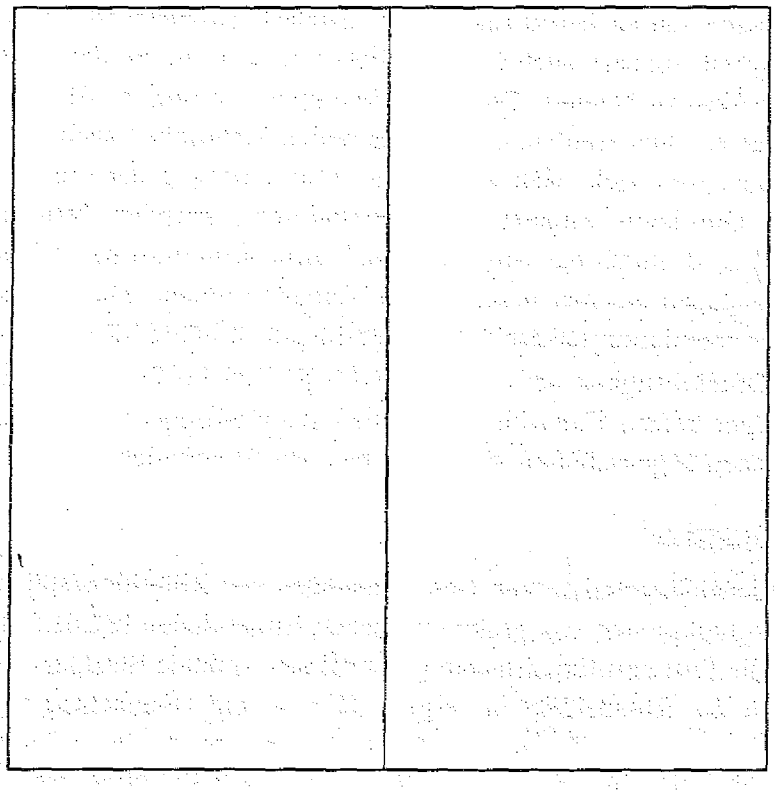


Abbildung 55 Uniaxialer Zug bei unendlicher Druckfestigkeit. Der Riß hat hier eine glatte, eindimensionale Gestalt.

geraden Riß. Im anderen Fall — der unendlichen Zugfestigkeit — findet man einen einzelnen,

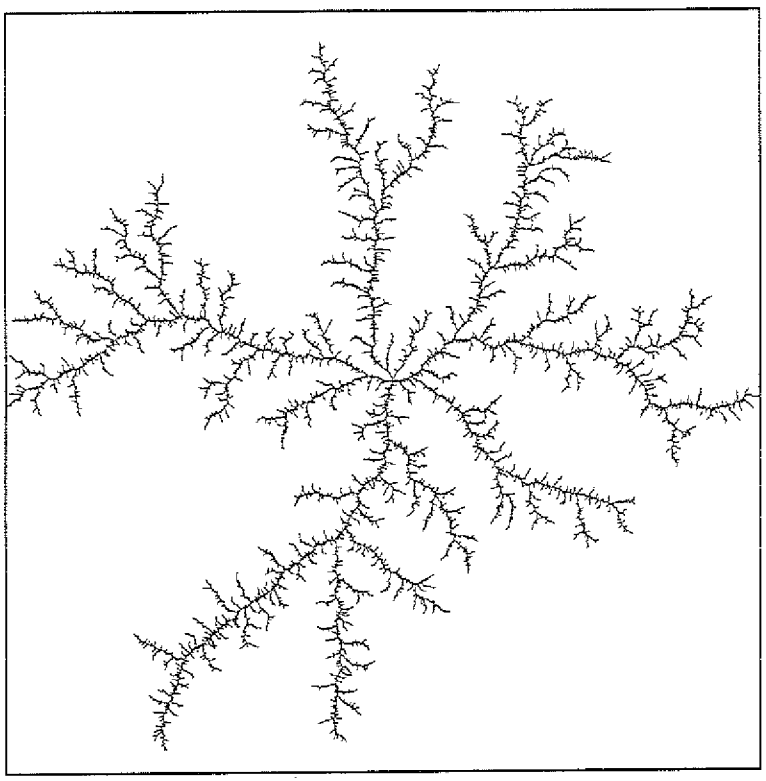


Abbildung 56 Isotrope Dilatation bei unendlicher Druckfestigkeit.

geraden, unverzweigten Riß, der vom gewählten Mikroriß ausgehend parallel zur Lastrichtung wächst. Dieses Ergebnis kann sofort mit Hilfe des Young-Experiments erklärt werden: Ein langgestreckter Balken, der an seinen Enden einer Zuglast ausgesetzt ist, ist in seinem Inneren nicht nur einer Zuglast, sondern auch einer Drucklast ausgesetzt, die dazu führt, daß sich der Querschnitt des Balkens verkleinert. Diese Druckkomponente führt zu dem beobachteten Riß. Hier muß allerdings erwähnt werden, daß man in realen Materialien auch ein anderes Verhalten beobachten kann, weil viele Materialien unter Druck entlang der Linien mit der größten Scherlast brechen. Dies kann besonders in Materialien mit plastischer Deformierbarkeit beobachtet werden und z. B. durch das Mohr-Coulomb Bruchkriterium beschrieben werden. Für Beton kann man dagegen das hier beschriebene Verhalten finden. Diese Bruchmode werden wir später in einer Simulation tatsächlicher Experimente wiederfinden.

Eine unendliche Druckfestigkeit hat den geringsten Einfluß im Fall der isotropen Dilatation (56). Man findet hier keinen Unterschied zum Fall symmetrischer Festigkeiten (45) und alle dort hervorgehobenen Eigenschaften sind auch hier wiederzufinden.

Quantitative Analyse

Eines der wesentlichen Ergebnisse von Untersuchungen von Rißbildungsphänomenen ist, daß sowohl Computersimulationen von probabilistischen Rißmodellen [82,81,83], als auch deterministische Modelle [79] und Experimente [84,140] eine fraktale Struktur sowohl der gesamten Risse als auch der Rißoberflächen zeigen. Hier ist ein Unterschied zum Laplaceschen Wachstum besonders hervorzuheben: Im Grenzfall unendlicher Noise-Reduction liefert das skalare Laplacesche Wachstum auf dem Gitter keine fraktalen Strukturen, während man in dem entsprechenden Elastizitätsproblem tatsächlich fraktale Risse findet [133].

Auch in unseren Simulationen erhalten wir fraktale Strukturen, die wir mit Hilfe der üblichen und auf DLA angewendeten Methoden charakterisieren wollen. Dazu betrachten wir zunächst

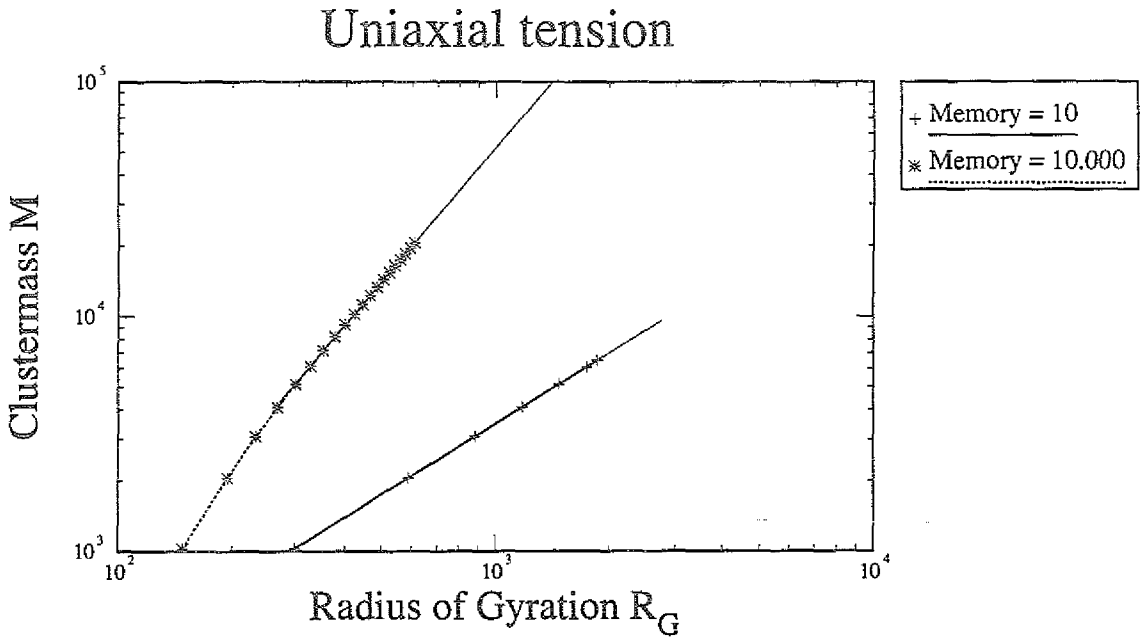


Abbildung 57 Abhängigkeit des Radius of Gyration von der Clustermasse bei uniaxialem Zug. Die Symbole + markieren den Fall 1., während die Symbole * den Fall 2. markieren.

den Radius of Gyration der generierten Risse. Wir haben ihn in allen drei simulierten Geometrien — uniaxialer Zug, Scherung und isotrope Dilatation — gemessen, wobei jeweils über 5 verschiedene Cluster gemittelt wurde. Dabei wurden für jede Geometrie zwei verschiedene Situationen simuliert, bei denen jeweils die Material-Festigkeiten und der Ermüdungs-Parameter unterschiedlich sind:

1. hohe Materialfestigkeit $\sigma_{ct} = \pm 100.$, aber Materialermüdung $M = \infty$.
2. niedrige Festigkeit $\sigma_{ct} = \pm 30.$, aber keine Materialermüdung $M = 10$.

In den folgenden Abbildungen 57-59 ist für jeden der Fälle die Abhängigkeit des Radius of Gyration von der Clustermasse gezeigt. Man findet in allen Fällen ein Potenzverhalten

$$R_G \propto M^{1/\beta} \tag{105}$$

mit $\beta = 1/D_f$. Die fraktalen Dimensionen D_f der verschiedenen Fälle sind in der folgenden Tabelle zusammengefaßt. Zusätzlich zeigen wir hier die in [82] gemessenen Dimensionen für die entsprechenden Fälle. Wie man am isotropen Zug erkennt, produziert der Fall 1. Risse,

Geometrie	Fall 1.	Fall 2.	Dimension aus [82]
uniaxialer Zug	1.89±0.02	1.00±0.01	
Scherung	1.51±0.01	0.99±0.02	1.62
isotroper Zug	1.56±0.02	1.10±0.01	1.51

deren fraktale Dimension mit bisherigen Ergebnissen, aber innerhalb der Fehlergrenzen nicht mit der fraktalen Dimension von DLA übereinstimmt. Eine ähnliche fraktale Dimension findet man auch für den Fall der Scherung. Für den uniaxialen Zug erhält man eine erstaunlich große fraktale Dimension, was dadurch begründet ist, daß man hier langgestreckte Risse erhält, die nicht gut durch eine isotrope Größe wie den Radius of Gyration beschrieben werden können.

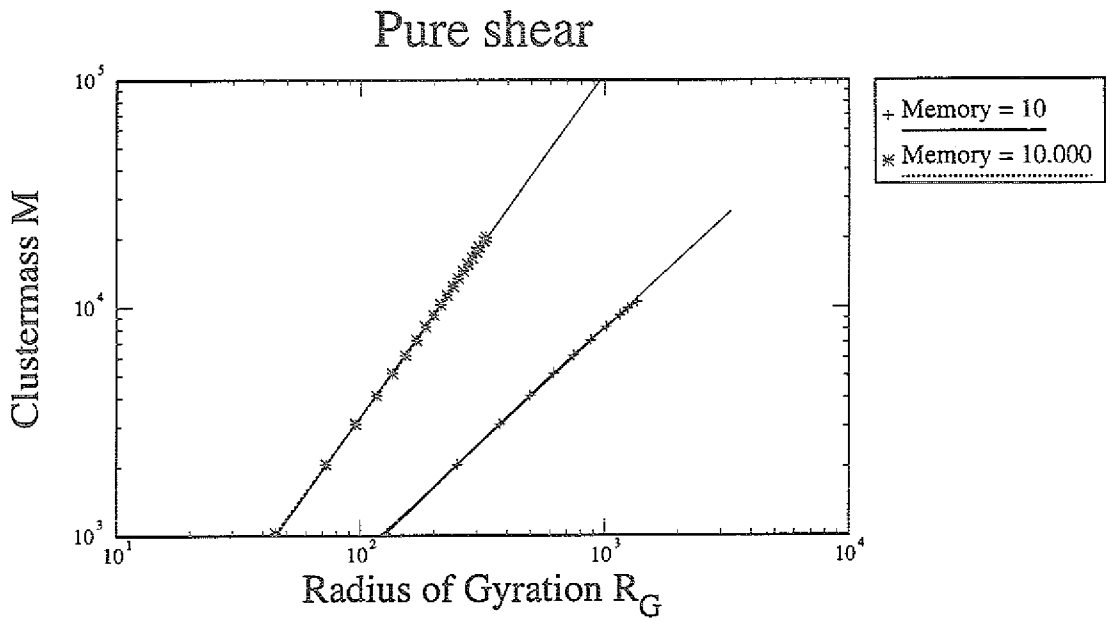


Abbildung 58 Abhängigkeit des Radius of Gyration von der Clustermasse bei Scherung. Die Symbole + markieren den Fall 1., während die Symbole ⊗ den Fall 2. markieren.

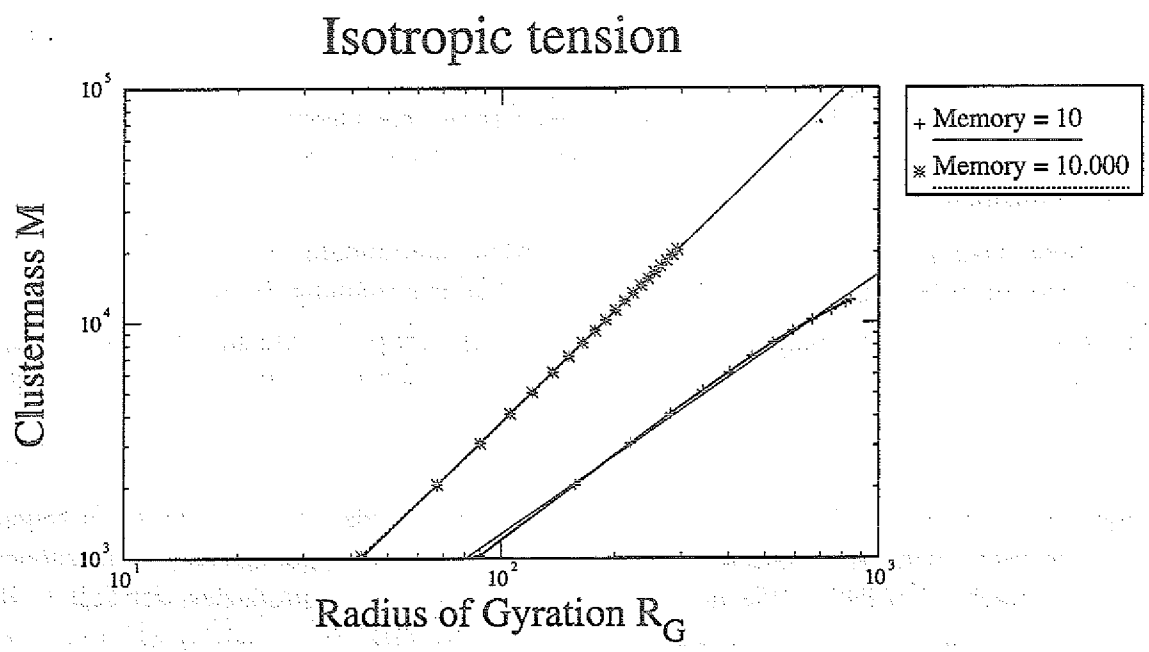


Abbildung 59 Abhängigkeit des Radius of Gyration von der Clustermasse bei isotropem Zug. Die Symbole + markieren den Fall 1., während die Symbole ⊗ den Fall 2. markieren.

Wie man anhand der vorhin gezeigten Bilder bei Reduktion des Ermüdungs-Effekts schon qualitativ feststellen konnte, unterdrückt eine niedriger Ermüdungs-Parameter M sehr stark die Bildung von Seitenzweigen, was sich in einer wesentlich niedrigeren fraktalen Dimension niederschlägt. Alle Risse mit $M = 10$ sind im wesentlichen eindimensionale Strukturen ohne Seitenzweige.

Für den Fall $M = \infty$ kann man einen wesentlichen Unterschied zwischen DLA und dem hier untersuchten Rißmodell feststellen. Dies ist die ausgeprägte Eigenschaft von Seitenzweigen,

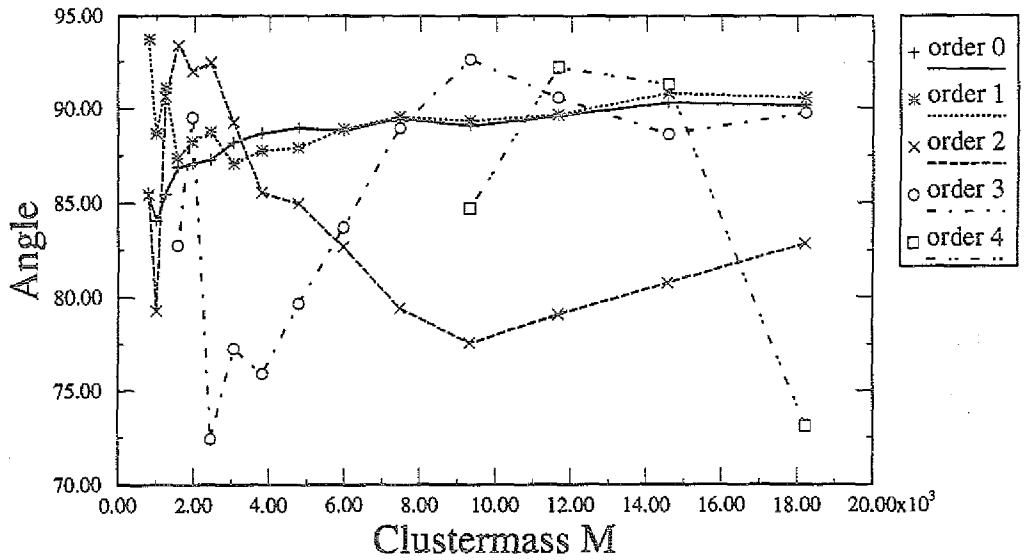


Abbildung 60 Mittlerer Winkel zwischen einem Seitenzweig der Ordnung n und seinem Hauptzweig.

senkrecht zu ihren Hauptzweigen wachsen. In der Abbildung 60 zeigen daher wir analog zur Abbildung 30 den mittleren Winkel zwischen einem Seitenzweig und seinem Hauptzweig in Abhängigkeit von der jeweiligen Zweigordnung. Die strenge Struktur abnehmender Winkel, die bei DLA festgestellt werden konnte, ist hier verschwunden. Außerdem sind die einzelnen Winkel größer als bei DLA und fluktuieren um 90°. Darüberhinaus erkennt man, daß die Anzahl der Zweigordnungen im Vergleich zu DLA wesentlich kleiner ist, d. h. die Bildung von Seitenzweigen ist sehr stark unterdrückt.

Vergleich mit Experimenten

Neben den Simulationen einfacher Geometrien, die wir oben beschrieben haben, haben wir auch versucht Rißstrukturen aus realistischen Experimenten zu reproduzieren. Dazu werden wir unsere Resultate des "Vier-Punkt Scher-Experiments" [75] und des sogenannten "Last Weg 2" Experiments [138] mit den experimentellen Ergebnissen vergleichen.

Im "Vier-Punkt Scher-Test" wird ein Beton-Teststück mit zwei Schlitzen als definierten Sollbruchstellen an vier Punkten — wie in Bild 61 angedeutet — belastet. Hier muß erwähnt werden, daß im Experiment natürlich endliche Teststücke benutzt werden, welche neue Randbedingungen aufprägen, während wir in der Simulation mit vier Kräften in einer unendlichen Ebene rechnen. Allerdings können wir zeigen, daß diese Vereinfachung keine wesentlichen Unterschiede hervorrufen. Zur weiteren Vereinfachung beschreiben wir die beiden Schlitze, die im Experiment als Sollbruchstellen dienen, durch zwei Teilchen an den entsprechen Stellen des Materials (Abb. 61). Die geometrische Abmessungen und Relationen zwischen Kantenlängen, Position und Größe der angreifenden Kräfte stimmen mit den Experimenten überein.

Der Vier-Punkt Scher-Test ist einer von mehreren Experimenten, die im Ingenieurbereich standardisiert sind, damit Experimente unterschiedlicher Gruppen miteinander und mit Computersimulationen verglichen werden können. Kommerzielle Finite-Elemente Programme müssen

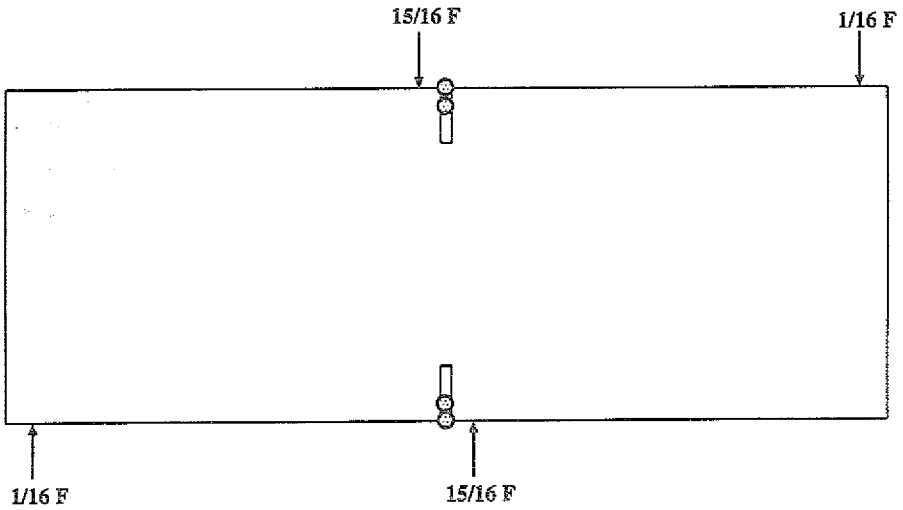


Abbildung 61 Geometrie des "Vier-Punkt Scher-Tests". $\vec{F}$ entspricht der insgesamt am Teststück angelegten Kraft.

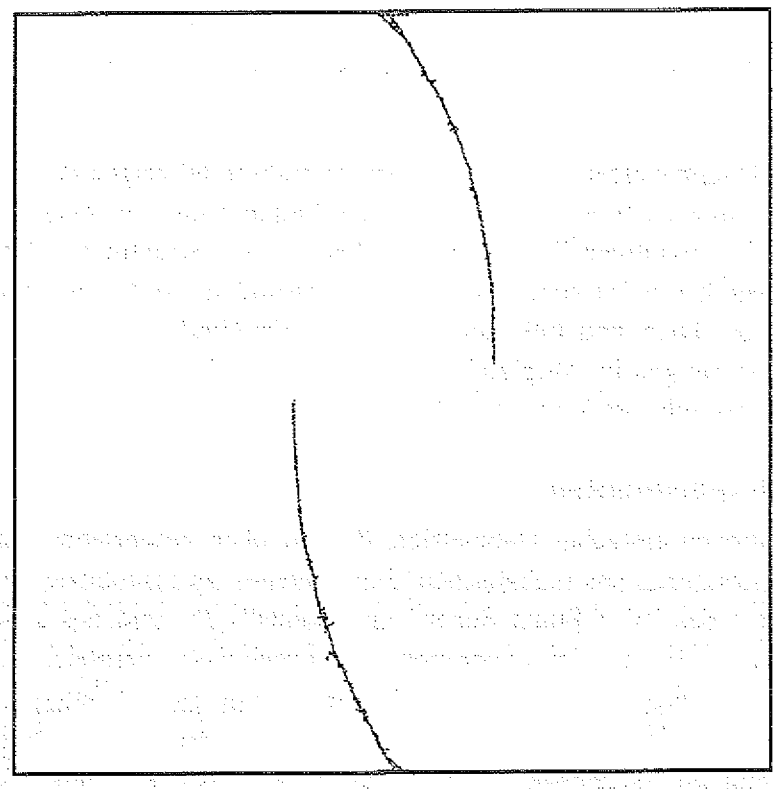


Abbildung 62 Simulation des "Vier-Punkt Scher-Tests".

in der Lage sein, diesen Vier-Punkt Scher-Test reproduzieren zu können, um akzeptiert werden zu können.

Das Ergebnis des "Vier-Punkt Scher-Tests" ist in Bild 62 wiedergegeben, während ein experimentelles Resultat in Bild 63 gezeigt ist. Qualitativ zeigen beide Resultate das gleiche Verhalten: Man erhält einen gekrümmten Riß, der zunächst ungefähr mit einem 45° Winkel beginnt und später in einen geraden Riß übergeht. Dieses Resultat kann kommentiert werden, indem man die verschiedenen Bruchstadien interpretiert. Im Anfangsstadium wird der Riß durch ein Scherfeld bestimmt, welches von den nahe den Schlitzen gelegenen Kräften erzeugt

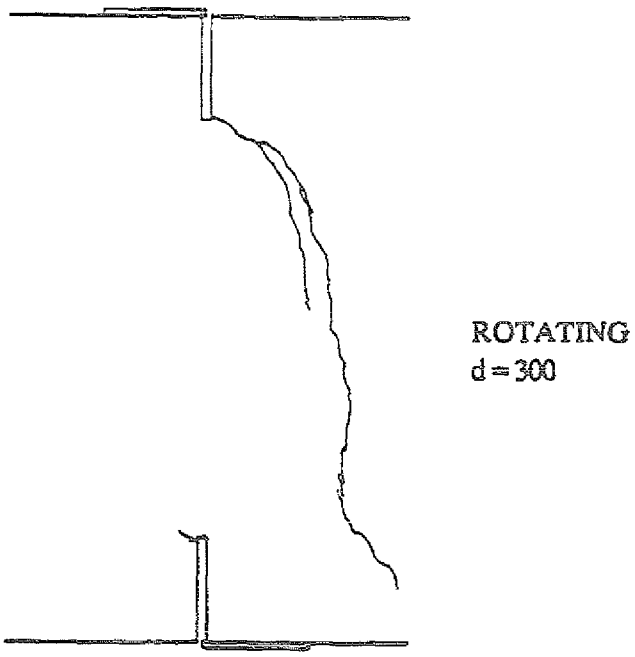


Abbildung 63 Experimentelle Ergebnisse des "Vier-Punkt Scher-Tests" (reproduziert aus [75]).

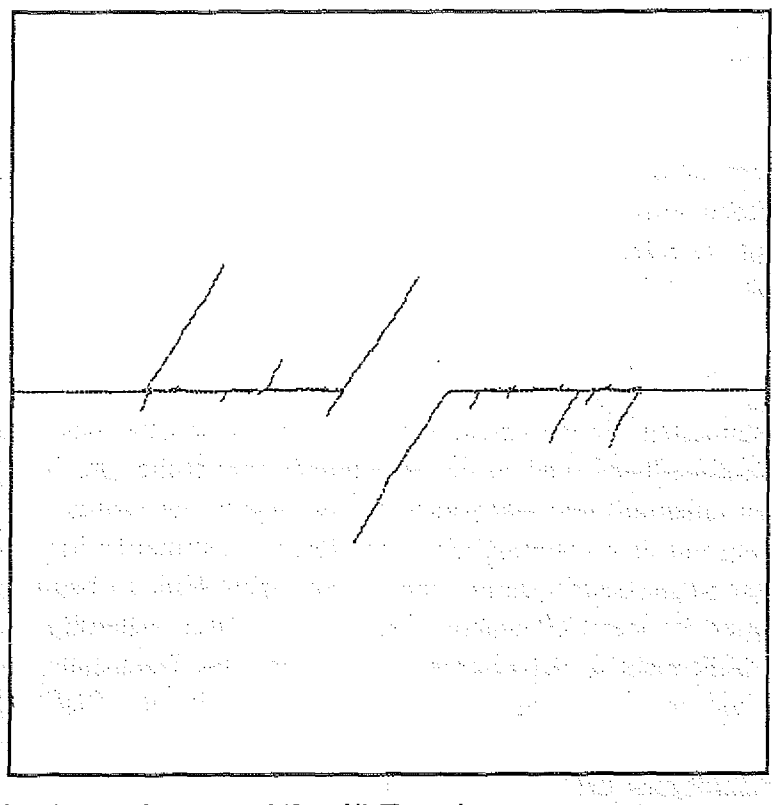


Abbildung 64 Simulation des "Last Weg 2" Experiments.

wird zu einem 45° Winkel führt. Wenn der Riß wächst, erhält man einen Übergang in ein reines Druckfeld, welches dann für das gerade Rißwachstum verantwortlich ist.

Eine ähnlich gutes Resultat erhält man im "Last Weg 2" Experiment. Bei ihm wird ein Beton-Teststück, welches wieder mit zwei Schlitzen versehen ist, zunächst einer Zug-Last ausgesetzt, bis ein Riß einer bestimmten Größe erzeugt ist und danach einer Scherlast ausgesetzt. Ein

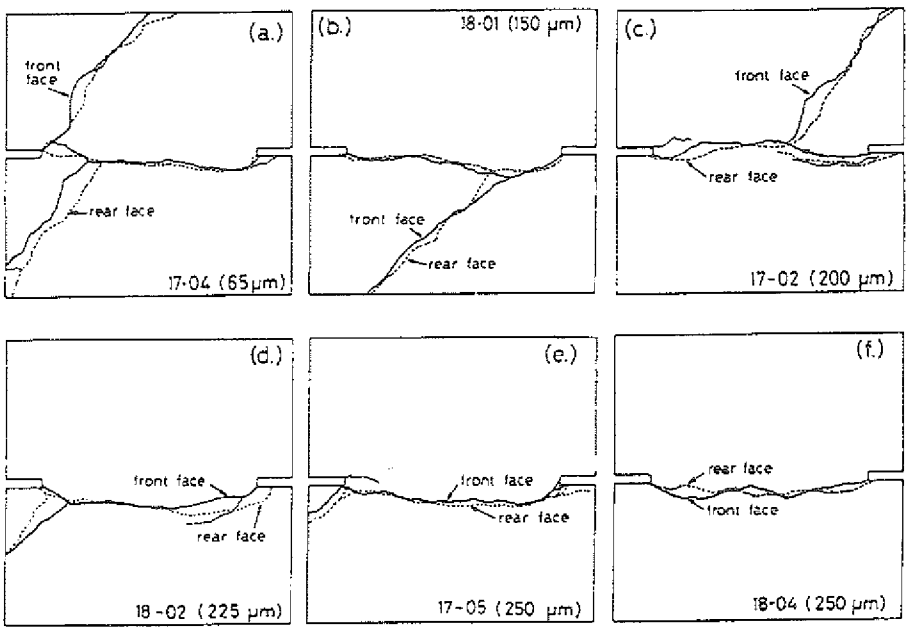


Abbildung 65 Experimentelle Ergebnisse des "Last Weg 2" Experiments" (reproduziert aus[138]). Die Teststücke wurden zunächst uniaxial belastet, bis sich ein Riß mit einer bestimmten Breite gebildete hat. Die Öffnungsweite ist in der verschiedenen Bildern angegeben und variiert zwischen $65\mu m$ (a) bis $250\mu m$ (f). Die beiden Kurven "front face" und "rear face" zeigen die Risse, die auf den beiden Stirnseiten des Teststücks beobachtet werden können.

Resultat unserer Simulation ist in Bild 64 gezeigt und kann mit den experimentellen Resultaten in Bild 65 verglichen werden. Man beobachtete zunächst einen langen, geraden Riß, der während der Zugphase erzeugt wird. Während der zweiten Phase, der Scherphase entstehen dann die diagonalen Scherrisse.

Anisotrope Random-Walks

In dem bisher diskutierten Modell spielte der vektorielle Charakter des Elastizitätsproblems lediglich an der Rißoberfläche und in der Bruchregel eine Rolle, der Random-Walk selbst blieb jedoch davon unberührt und war weiter isotrop, wie in der skalaren Diffusion. Dies ist in Übereinstimmung mit dem ursprünglichen von Roux konstruierten Modell. Die natürliche Modifikation dieser Regel besteht darin, einen anisotropen Walk zu betrachten, bei dem der vektorielle Charakter der Lamé-Gleichung einen Bias einführt. Allerdings wurde dieser anisotrope Random-Walk nicht in der originalen Definition des Wachstumsmodells eingeführt, sondern als Erweiterung dessen, da im Gegensatz zu dem ursprünglichen Modell die physikalische Interpretation des anisotropen Walks noch nicht klar ist. Allerdings muß betont werden, daß die Einführung des anisotropen Walk ein letzter Schritt zur Generierung realistischer Risse zu sein scheint und der benutzte Anisotropie-Parameter der wesentliche Parameter zur Generierung unterschiedlicher, physikalischer Strukturen ist.

Die Einführung eines anisotropen Walks kann durch mehrere Überlegungen begründet werden:

- Die Laplace- und die Lamé-Gleichung weisen ein unterschiedliches Abschirmverhalten auf. Die Wahrscheinlichkeit, daß ein reiner Random-Walker in einen vorgegebenen Raumbereich hineindiffundiert, wird durch die isotrope Laplace-Gleichung gegeben. Durch

Einführung dieses Bias kann der anisotrope Charakter der Lamé-Gleichung berücksichtigt werden.

- In Roux's ursprünglichem Algorithmus spielte die Schrittmatrix $\underline{\underline{Q}}(\vec{e}) = (\alpha \cdot \underline{\underline{1}} + (1-\alpha) \cdot d \cdot \vec{e} \otimes \vec{e})$ eine zentrale Rolle. Sie verändert sowohl die Richtung, als auch den Betrag eines initialen Vektors (vgl. 7.2). Stellt man $\underline{\underline{Q}}(\vec{e})$ dar als ein Produkt aus einer längenerhaltenden Drehmatrix $\underline{\underline{D}}(\vec{e})$ und einer richtungserhaltenden Skalierung $s(\vec{e})$, dann kann die Skalierung bei korrekter Normierung als richtungsabhängige Sprung-Wahrscheinlichkeit interpretiert und abgespalten werden (vgl. Gl. 85):

$$\begin{aligned}\left(Q_r^N \vec{u}\right)(\vec{x}) &= \left\langle \underline{\underline{Q}}(\vec{e}_N) \cdots \underline{\underline{Q}}(\vec{e}_1) \cdot \vec{u}(\vec{x} + r\vec{e}_1 + \cdots + r\vec{e}_N) \right\rangle_{\vec{e}_1 \cdots \vec{e}_N} \\ &= \langle s(\vec{e}_N) \cdots s(\vec{e}_1) \rangle_{\vec{e}_1 \cdots \vec{e}_N} \left[\underline{\underline{D}}(\vec{e}_N) \cdots \underline{\underline{D}}(\vec{e}_1) \cdot \vec{u}(\vec{x} + r\vec{e}_1 + \cdots + r\vec{e}_N) \right]_{\vec{e}_1 \cdots \vec{e}_N}\end{aligned} \tag{106}$$

Um den Mittelwert $[\ldots]_{\vec{e}_1 \cdots \vec{e}_N}$ zu bilden, wird die Wahrscheinlichkeit, in die Richtung $\vec{e}$ zu springen, nicht mehr isotrop, sondern gemäß $p(\vec{e}) \propto s(\vec{e})$ gewählt. Dies läßt sich leicht berechnen, denn

$$s^2(\vec{e}) \propto \left| \underline{\underline{Q}}(\vec{e}) \vec{u} \right|^2 = 1 + 8 \frac{3-2\nu}{(1-2\nu)^2} \cos^2(\varphi) \tag{107}$$

wobei φ den Winkel zwischen der Sprungrichtung $\vec{e}$ und $\vec{u}$ darstellt. Man beachte, daß diese Schreibweise lediglich eine Umformulierung von Gl. (85) darstellt und daß man daher für diesen anisotropen Walk immer noch exakt zeigen kann, daß er gegen die Lamé-Gleichung konvergiert. Allerdings bietet dieser Walk den Vorteil, daß durch den Bias ein "Importance Sampling" durchgeführt wird, in dem hauptsächlich solche Walks berücksichtigt werden, die einen großen Beitrag zu $(Q_r^N \vec{u})(\vec{x})$ liefern.

- Diskretisiert man die Lamé-Gleichung auf einem Quadratgitter mit Hilfe der üblichen Differenzen-Ausdrücken [141] der Differential-Operatoren, so erhält man

$$\begin{aligned}2(3-2\nu)\vec{u}_{00} = &\sum_{ij=NN(00)} \left\{ (1-\nu)\underline{\underline{1}} + \vec{e}_{ij} \otimes \vec{e}_{ij} \right\} \cdot \vec{u}_{ij} \\ &- \frac{1}{4} \sum_{ij=NNN(00)} \left\{ \underline{\underline{1}} - \vec{e}_{ij} \otimes \vec{e}_{ij} \right\} \cdot \vec{u}_{ij}\end{aligned} \tag{108}$$

wobei $NN(00)$ bzw. $NNN(00)$ die nächsten bzw. übernächsten Nachbarn des Gitterplatzes (00) und die $\vec{e}_{ij}$ die Einheitsvektoren zum Gitterplatz (ij) darstellen. Interpretiert man den Betrag der einzelnen Beiträge der nächsten bzw. übernächsten Nachbarplätze als Sprungwahrscheinlichkeiten $s(\vec{e})$, so erhält man

$$s^2(\vec{e}) \propto \begin{cases} \left(1 + \frac{3-2\nu}{(1-\nu)^2} \cos^2(\varphi)\right) & \text{bei naechsten Nachbarn} \\ \left(1 + \cos^2(\varphi)\right) & \text{bei uebernaechsten Nachbarn} \end{cases} \tag{109}$$

wobei φ wieder der Winkel zwischen der Verschiebung $\vec{u}_{ij}$ und $\vec{e}_{ij}$ darstellen. Man beachte, daß die Sprungwahrscheinlichkeiten in dieser Diskretisierung die gleiche Struktur wie in Roux's Kontinuum-Rechnung besitzen. Man findet also richtungsabhängige Sprungwahrscheinlichkeiten, wobei Sprünge zu nächsten- bzw. übernächsten Nachbarn unterschiedlich gewichtet werden. Allerdings muß man berücksichtigen, daß nächste- und übernächste Nachbarn nicht äquivalent sind, da sie ja einen unterschiedlichen Abstand von dem zentralen Platz haben. Ich möchte nur darauf aufmerksam machen, daß man für beide Nachbarn eine ähnliche Struktur der Anisotropie vorfindet. Diese gemeinsame Struktur werden wir später nutzen.

Dies zeigt, daß die Einführung eines anisotropen Random-Walk mit Hilfe des Verschiebungsfeldes eine Vereinfachung und Beschleunigung der Berechnung liefern, da man sich auf die wichtigen Walks mit großem Gewicht beschränkt.

Für unsere Simulationen sind diese Rechnungen noch kein großer Gewinn, da hier nicht das Verschiebungs-, sondern das Spannungsfeld die zentrale Rolle spielt. Allerdings läßt sich auch für das Spannungsfeld ein physikalischer Bias motivieren, welcher aus dessen Norm abgeleitet werden soll. In dem vorigen Fall bot sich natürlich die euklidische Länge des Verschiebungsfeldes als Norm an. Für eine Matrixnorm hat man dagegen größere Freiheiten und deswegen wollen wir uns auf zwei physikalische Argumente stützen, um sinnvolle Ausdrücke für einen Bias zu begründen.

- Die in einem gespannten System gespeicherte potentielle Energie ist zwar eine globale Größe, allerdings kann man formal wie in der Elektrostatik eine Feldenergiedichte einführen, die von dem lokalen Spannungsfeld $\underline{\underline{\sigma}}$ bestimmt wird und angibt, welche Arbeit ein lokales Volumenelement bei einer bestimmten Verzerrung gegen das Spannungsfeld verrichten muß (Clapeyronsche Formel)

$$U = \frac{1}{2}\varepsilon_{ij}\sigma_{ij} \tag{110}$$

  Für linear elastische, isotrope Medien, in denen man eine Punktkraft $\vec{F}$ im Ursprung anwendet, reduziert sich diese Gleichung auf

$$U \propto 1 + 2\frac{(2-\nu)(1+\nu)}{(1-\nu)^2}\cos^2(\varphi) \tag{111}$$

  wobei φ der Winkel zwischen der angelegten Kraft und der Sprungrichtung ist.
- Das von uns gewählte Bruchkriterium beruht auf den beiden Hauptspannungen S_I und S_{II}. Mit deren Hilfe bestimmen wir den Bereich mit großer Gesamtspannung, den wir durch $S_I^2 + S_{II}^2$ definieren, und wählen dies als Basis für unseren Bias. Man erhält

$$S_I^2 + S_{II}^2 \propto 1 + 4\frac{1+\nu}{(1-\nu)^2}\cos^2(\varphi) \tag{112}$$

Auch in diesen Überlegungen findet man eine Anisotropie, welche die Struktur

$$p(\varphi) \propto 1 + \alpha \cdot \cos^2(\varphi) \tag{113}$$

besitzt. Wir wollen daher im folgenden eine Modellvariante untersuchen, bei der die Walker keinen isotropen, sondern einen Biased Random-Walk ausführen, bei dem die Wahrscheinlichkeit, in eine bestimmte Richtung zu springen durch (113) gegeben ist. Setzt man in die oben vorgeschlagenen Formeln ein Poisson-Verhältnis $\nu = 0.2$ ein, erhält man immer einen Wert $\alpha \approx 10$. Dieser Wert bietet sich also zunächst an und soll in den folgenden Simulationen benutzt werden. Es scheint jedoch sehr interessant zu sein, diesen Parameter als freien Parameter zu betrachten und seinen Einfluß auf das hier vorgeschlagene Rißmodell zu studieren.

Einen großen Einfluß auf die Simulationsergebnisse hat zusätzlich eine strukturelle Veränderung der vorgeschlagenen Anisotropie $p(\varphi)$. Mit der oben vorgeschlagenen Form eines Bias wird ein Random-Walker bevorzugt in Richtung einer angelegten Kraft diffundieren. Wie wir gleich erläutern werden, ist dies von Bedeutung für Walker, die an den von außen

angelegten Kräften starten. Für Walker, die allerdings an einem Riß neu gestartet wurden, kann die Form (113) jedoch zu einem absolut unerwünschten Verhalten führen. Da außerdem der exakte Einfluß des Bias noch nicht vollkommen verstanden ist, möchten und deswegen wollen wir auch die Variante

$$p(\varphi) \propto 1 + \alpha \cdot \sin^2(\varphi) \tag{114}$$

betrachten.

Das bisher studierte Modell soll daher derart modifiziert werden, daß ein Walker an der Stelle $\vec{r}_i$ startet und einen Biased Walk ausführt, bei dem er mit der Wahrscheinlichkeit $p(\phi)$ in die Richtung ϕ springt und an der Stelle $\vec{r}_f$ auf den Riß trifft. Dort wird entsprechend Gl. (102) die herrschende Spannung bestimmt. Die im Walk benutzten Sprungwahrscheinlichkeiten dürfen nun nicht doppelt gezählt werden und daher müssen wir an der Rißoberfläche nicht $\underline{\underline{\sigma}}$, sondern

$$\underline{\underline{\sigma}}_{\text{skaliert}} = \underline{\underline{\sigma}}/p(\Phi)$$

akkumulieren, wobei Φ der Winkel zwischen der angelegten Kraft und $\vec{r}_i - \vec{r}_f$ ist.

Ein anisotroper Walk schlägt sich im wesentlichen in zwei Punkten nieder. Der erste offensichtliche Effekt ist, daß Walker, die am Rand starten bevorzugt in bestimmte Raumrichtungen diffundieren werden. Allerdings findet man noch einen zweiten Effekt, welcher für Walker auftritt, die an der Rißoberfläche neu mit einer Kraft $\vec{F}_{\text{bal}}$ gestartet werden. Diese Kraft weist im wesentlichen senkrecht zur Rißoberfläche. Der grundlegende Prozeß ist folgender: Der Walker muß sich — entsprechend dem Bias — solange durch den Raum bewegen, bis er wieder auf die Rißoberfläche trifft. Betrachtet man nun exemplarisch den Fall eines geraden Risses, wie er unter uniaxialem Zug auftritt, so muß der Walker trotz seines Bias an einer Stelle den Riß wieder berühren, die senkrecht zur ursprünglichen Kraftrichtung liegt

$$\vec{r}_i - \vec{r}_f \perp \vec{F}_i \tag{115}$$

Betrachtet man nun einen Cosinus-artigen Bias liefert dies allerdings

$$p(\Phi) \to 0 \tag{116}$$

und damit

$$\underline{\underline{\sigma}}_{\text{skaliert}} \to \infty \tag{117}$$

Diese Argumente machen klar, daß ein zweiter Effekt des Bias darin besteht, das Gleichgewicht zwischen "externen" Walks, die am Rand gestartet werden, und "internen" Walks, die am Riß neu gestartet werden, zu verschieben.

In den Abbildungen 66 und 67 zeigen wir zwei typische Ergebnisse für Simulationen unter isotropem Zug, bei denen wir die Spannungs-Zähler nach jeweils zehn Teilchen löschen.

Die Simulation mit Cosinus-artigem Bias liefert sehr scharfe, Nabel-artige Cluster. Die Zweige sind im Gegensatz zu Simulationen mit hoher Materialfestigkeit extrem gerade und zeigen keine Tendenz zu Krümmungen. Derartige Risse sind durchaus schon experimentell realisiert worden.

Mit einem Bias senkrecht zur Kraftrichtung verändert sich die Struktur der Risse dagegen vollständig! Man erhält ein verwobenes Netzwerk von stark gekrümmten Zweigen, die immer

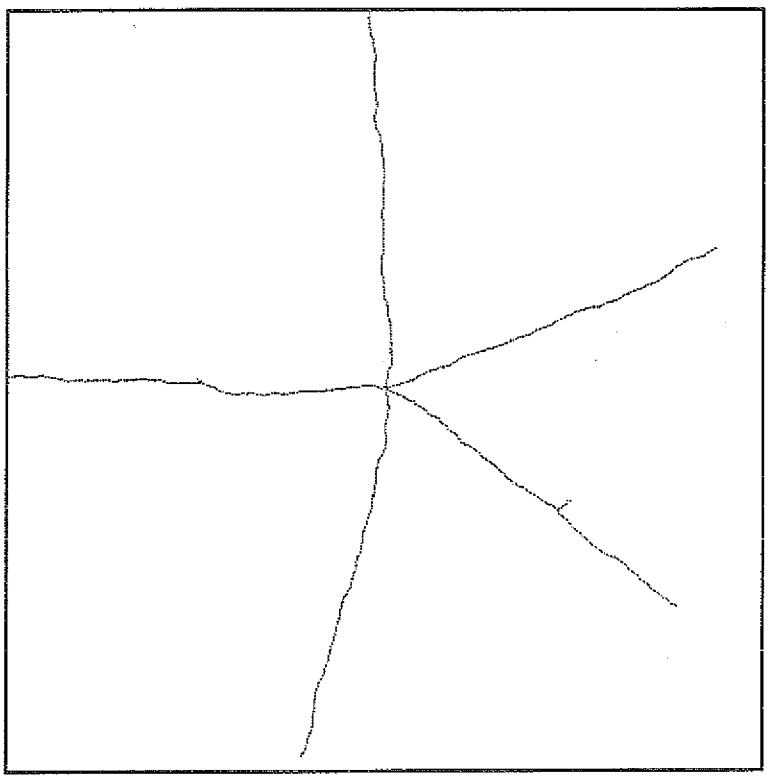


Abbildung 66 Rißstruktur bei isotroper Dilatation und einem Bias $p(\phi) \propto 1 + 10 \cdot cos^2(\phi)$. Der Riß enthält ca. 2000 Teilchen.

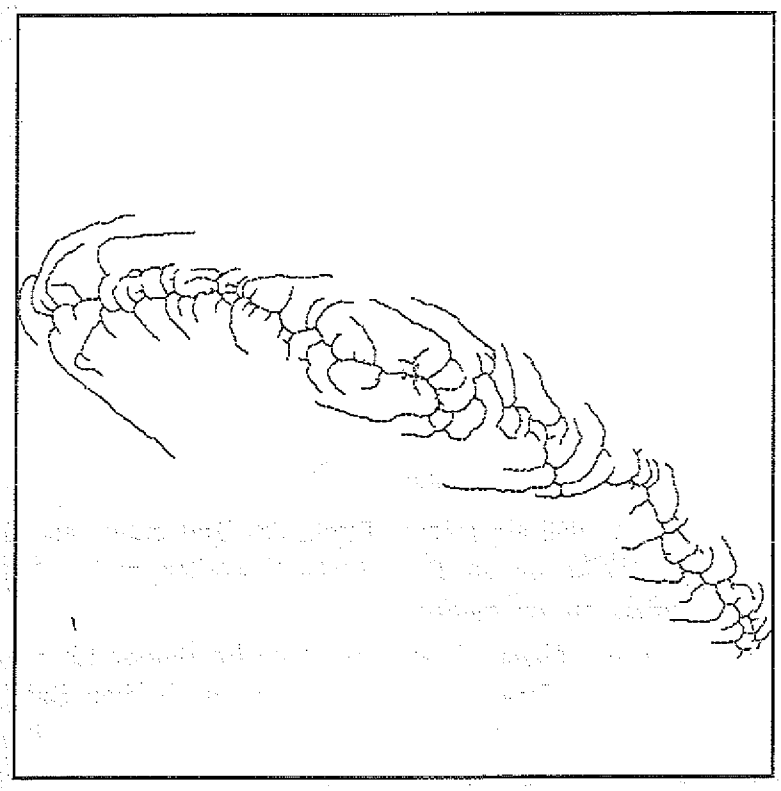


Abbildung 67 Rißstruktur bei isotroper Dilatation und einem Bias $p(\phi) \propto 1 + 10 \cdot sin^2(\phi)$. Der Riß enthält ca. 6000 Teilchen.

neue Seitenzweige produzieren. Besonders interessant ist hier, daß man sehr ausgeprägt

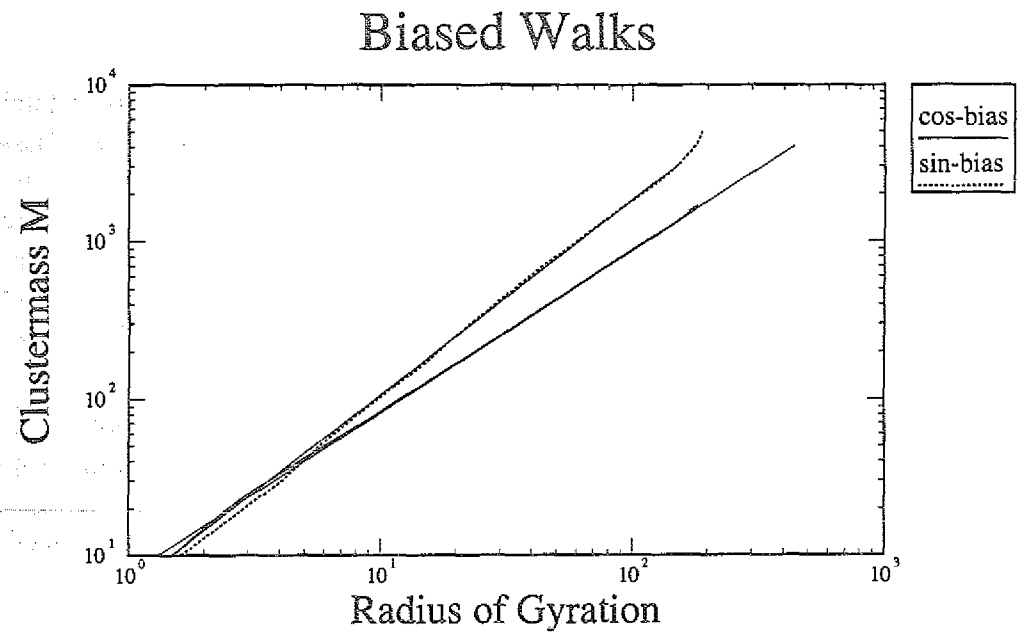


Abbildung 68 Abhängigkeit der Clustermasse vom Radius of Gyration bei Biased-Walks. Die fraktalen Dimensionen sind: $D_f \approx 1.03$ für Cosinus-Bias und $D_f \approx 1.23$ für Sinus-Bias.

wesentliche Merkmale der Rißbildung findet: Alle Seitenzweige wachsen zunächst senkrecht aus den Hauptzweigen heraus, um sich erst später zu krümmen und aufeinander zuzuwachsen. Noch interessanter ist dagegen die Eigenschaft, daß Seitenrisse offensichtlich nur auf der Außenseite von gekrümmten Armen entstehen. Dies ist vollkommen in Übereinstimmung mit experimentellen Ergebnissen von Lemaire und van Damme [68] und kann direkt mit den in der Einleitung gezeigten Rissen verglichen werden.

Auch von diesen Rissen soll wieder die fraktale Dimension bestimmt werden. In der Abbildung 68 zeigen wir die Abhängigkeit der Rißmasse von dem Radius of Gyration, jeweils für einen Cosinus- bzw. Sinus-artigen Bias. Die entsprechenden fraktalen Dimensionen sind

$$\begin{aligned} D_f &= 1.03 \pm 0.05 \quad \text{Cosinus-artiger Bias} \\ D_f &= 1.23 \pm 0.04 \quad \text{Sinus-artiger Bias} \end{aligned} \tag{118}$$

Ein Cosinus-artiger Bias führt also wieder auf eindimensionale, nadelartige Strukturen, während ein Sinus-artiger Bias verzweigte und fraktale Risse ergibt.

7.5 Zusammenfassung und Diskussion

In diesem Kapitel haben wir die Möglichkeiten studiert, Rißwachstum mit Hilfe von Random-Walkern zu simulieren. Ausgehend von einer Arbeit von Roux haben wir mehrere mögliche stochastische Algorithmen diskutiert und dabei gezeigt, daß eine direkte numerische Simulation der vorgeschlagenen Methoden nicht praktikabel ist, weil es nicht möglich ist, die starken numerischen Fluktuationen zu unterdrücken. Allerdings wurde dort gezeigt, daß es tatsächlich möglich ist, das Elastizitätsproblem auf Random-Walker abzubilden. Angeregt durch den verwandten DLA Algorithmus — welcher sehr erfolgreich bei der Simulation Laplacescher Wachstumsprozesse benutzt wird — haben wir eine neue Methode konstruiert, Rißwachstum in elastischen Medien zu simulieren. Die Benutzung eines Random-Walk Algorithmus führt

darauf, daß diese neue Methode auf solche Materialien angewendet werden kann, die eine "quenched disorder" aufweisen.

In dieser Methode werden die Random-Walker nicht länger dazu benutzt, eine Green-Funktion zu konstruieren, sondern werden dazu benutzt, die stark belasteten Spitzen eines Risses zu finden. Dabei sind alle wesentlichen Einzelheiten des Prozesses — der Walk und die Wachstumsregel — vollständig im Kontinuum formuliert, um alle unerwünschten Gittereffekte, die die Struktur der Risse beeinflussen könnten, zu vermeiden. Mit dieser neuen Methode können bekannte numerische und experimentelle Resultate reproduziert werden. Alle Simulationen können auf konventionellen Workstations durchgeführt werden und keine Supercomputer werden benötigt.

Zunächst muß als wesentliches Resultat und als wesentlicher Vorteil hervorgehoben werden, daß mit dieser Methode realistische Rißstrukturen von einer Größe bis zu mehreren zehntausend Teilchen simuliert werden können, was bisher nicht möglich war. Wir haben verschiedene Geometrien simuliert und die dort generierten Rißstrukturen qualitativ und quantitativ mit anderen Simulationen verglichen. Wir konnten bereits bekannte Strukturen reproduzieren und zeigen, daß die resultierenden Risse fraktal sind.

Wir haben streng darauf geachtet, mögliche Gitteranisotropien, welche wenigstens in DBM bzw. DLA eine extreme Auswirkung auf die Resultate haben, zu vermeiden: Sowohl der Walk, als auch das Bruchkriterium und die Wachstumsregel sind ohne Benutzung eines Gitters formuliert. Daher können wir davon ausgehen, daß die von uns beobachteten Anisotropieeffekte nicht auf Symmetriebrechungen zurückzuführen sind, sondern einen tatsächlichen physikalischen Ursprung haben. Tatsächlich sind ja z. B. die rechten Winkel zwischen Haupt- und Nebenzweigen wichtiger Bestandteil der Physik dieses Problems und es ist wichtig, daß dies richtig beschrieben wird. Dies steht im Gegensatz zu z.B. Noise-Reduced DLA Clustern auf dem Gitter [38], die eine Anisotropie aufweisen, die den hier generierten Scherrissen sehr ähnlich ist. Dort ist allerdings die Anisotropie ein Artefakt der Methode und ist vollkommen bestimmt durch das zugrundeliegende Gitter, während Anisotropien in diesem Modell einzig eine Folge des vektoriellen Charakters der Problems und vollkommen physikalisch sind.

Darüberhinaus muß besonders hervorgehoben werden, daß hier ein einziges Modell ausreicht, eine große Vielfalt experimentell realisierte Rißeigenschaften zu reproduzieren. Durch Einführung eines anisotropen Random-Walks ist es zusätzlich möglich, noch eine weitere Klasse von Rißstrukturen einzubeziehen.

Besonders muß hervorgehoben werden, daß man mit dieser Methode in der Lage ist, experimentell beobachtbare Risse zu simulieren, wie im Absatz 6 "Vergleich mit Experimenten" des vorherigen Abschnitts gezeigt wurde. Besonders der "Vier-Punkt-Scher Test" ist hervorzuheben, da er einen Standard-Test im Ingenieur-Bereich darstellt. Ebenfalls müssen wir die Simulationsergebnisse des anisotropen Walks hervorheben. Hier können wir mit einem reinen Off-Lattice Modell sehr komplexe realistische Strukturen simulieren.

Als weiteres muß erwähnt werden, daß alle Rechnungen auf Workstations durchgeführt wurden und keine Supercomputer benötigt werden. Das hier benutzte Programm ist eine Weiterentwicklung des bisher benutzten DLA Programms, das um einige vektorielle Komponenten ergänzt wurde. Eine typische Rechnung für einen Riß mit 20.000 Teilchen benötigt nicht mehr als 5MB Hauptspeicher und ca. 10–15 Stunden Rechenzeit auf einer SPARCstation 2. Je nach Wahl der Festigkeits- und Ermüdungs-Parameter kann sich diese Zeit jedoch stark verlängern oder auch verkürzen.

Ein Nachteil unserer Methode besteht bisher darin, daß lediglich Kräfte und Spannungen, aber keine Verschiebungen $\vec{u}$ berücksichtigt werden können. Deswegen ist es bisher nicht möglich, Spannungs-Verzerrungs Relationen oder Eigenschaften der freiwerdenden Spannungsenergie, die natürlich wichtige physikalische Eigenschaften beschreiben, zu messen. Aus dem gleichen Grund ist es auch nicht möglich, experimentelle Situationen zu simulieren, in denen nicht die Last, sondern die Verschiebung kontrolliert wird.

Ein Problem ergibt sich durch die Randbedingung an der Rißoberfläche (80). Hier kann man keine Analogie mehr zu DLA ziehen, wo die übliche Bedingung $\phi = 0$ für die Crack Oberfläche leicht als absorbierende Oberfläche interpretiert werden kann. Hier erfordert die Randbedingung (80) eine Mehrfachstreuung der Walker an der Rißoberfläche, wobei jede Streuung einen Effekt höherer Ordnung repräsentiert. Formal läßt sich jeder Streuvorgang als einzelner Relaxationsschritt in einer Randintegral-Rechnung interpretieren. Da hier die Relaxation nicht systematisch, sondern stochastisch durchgeführt wird, findet man zusätzlich zum Rauschen durch die Walker noch ein Rauschen in der Randbedingung, dessen Unterdrückung viel Rechenzeit erfordert. Hier könnte die Einführung des Bias einen Lösungsweg aufzeigen, und es ist sicherlich interessant, dies eingehender zu studieren.

Der letzte Punkt, der diskutiert werden soll, ist die Tatsache, daß der Walker einen einfachen Random-Walk ausführt, in dem sich Details der Lamé-Gleichung nicht wiederfinden. Dies ist in Übereinstimmung mit dem ursprünglichen Algorithmus von Roux. Der vektorielle Charakter des Elastizitätsproblems wird bei einem isotropen Random-Walk erst an der Rißoberfläche sichtbar. Der vektorielle Charakter und die spezielle Randbedinung an der Rißoberfläche führen dazu, daß der Walker ständig an der Oberfläche gestreut werden muß. Jeder Streuprozeß reduziert jedoch das statistische Gewicht, so daß eine Relaxation der Oberflächenspannung nur sehr langsam erfolgt. Daher haben wir als Variante einen anisotropen Walk betrachtet, in dem in jedem Schritt das erzeugte Spannungsfeld berücksichtigt wird. Zunächst war diese Variante dazu gedacht, die Simulationsgeschwindigkeit zu erhöhen. Es stellte sich jedoch heraus, daß in Abhängigkeit von der gewählten Anisotropie die gesamte Rißstruktur geändert wird. Damit wird der Anisotropieparameter α zum wichtigsten und interessantesten Parameter des Problems, dessen physikalische Relevanz noch untersucht werden sollte.

Aber selbst eine einfache Diffusion führt nicht zu unphysikalischem Verhalten. Man muß beachten, daß Risse wegen des singulären Verhaltens der Spannungen bevorzugt an der Rißspitze wachsen. Dieses singuläre Verhalten findet man jedoch auch im Elektrostatischen Fall: Das elektrische Feld divergiert an Polspitzen. Also werden Random-Walker im Mittel bevorzugt Rißspitzen erreichen und sind deswegen geeignet, die "hot sites" eines Clusters, an denen eine große Spannung herrscht, herauszufinden und abzutasten.

Kapitel 8

Zusammenfassung und Diskussion

In dieser Dissertation wurde ein neues stochastisches Modell zum Wachstum von Rissen in elastischen, ungeordneten Medien konstruiert und numerisch studiert. Der Grundgedanke dabei war, den engen Zusammenhang zwischen der Laplace-Gleichung und der Lamé-Gleichung auszunutzen und in Analogie zum erfolgreichen DLA Modell für das Laplacesche Wachstum Random-Walker zu benutzen, um Informationen über die an den äußeren Rändern angelegten Kräfte zu einer Rißoberfläche zu transportieren. Durch die Verwendung von Random-Walkern ist in diesem Modell bereits eine "annealed disorder" enthalten, so daß dieses Modell für ungeordnete Materialien gültig ist. Das bedeutet keinerlei Einschränkung sondern im Gegenteil einen Vorteil, da Unordnung ein wesentliches Charakteristikum realer Materialien ist und experimentell beobachtete Rißvorgänge von dieser dominiert werden.

Dieses neue Modell verzichtet vollkommen auf die Benutzung eines Gitters, welches für andere numerische Simulationsmethoden typisch ist. Dadurch können typische Gittereffekte, wie sie besonders im Laplaceschen Wachstum bekannt sind, ausgeschlossen werden. Durch die Verwendung eines erweiterten DLA Algorithmus können sehr effiziente Simulationen auf normalen Workstations durchgeführt werden. Es ist uns so gelungen, realistische Rißstrukturen in einer Größe von mehreren tausend bis wenigen zehntausend Teilchen durchzuführen. Die so generierten Strukturen sind etwa um eine Größenordnung größer, als bisher bekannte Rißsimulationen.

Wegen der wesentlich größeren Anzahl von aggregierten Teilchen, ist es schwierig, unsere Simulationen mit den bisher durchgeführten numerischen Rechnungen mit wesentlich weniger gebrochenen Bonds zu vergleichen. Man findet in allen simulierten Geometrien eine Übereinstimmung zwischen unseren Resultaten und bisher publizierten, numerisch generierten Rißstrukturen. Darüberhinaus weisen unsere Simulationen alle wesentlichen Eigenschaften auf, die man bei realen Rissen beobachten kann: In allen Simulationen findet man eine ausgeprägte Tendenz, daß Seitenzweige senkrecht aus den Hauptzweigen herauswachsen, wie es die Randbedingung der Lamé-Gleichung fordert. Dies ist besonders bemerkenswert und wichtig, da unsere Methode vollkommen "Off-Lattice" ist und dieses Verhalten durch keinen Gittereffekt ausgelöst wird.

Wir finden weiter, daß die Rißspitzen nicht mehr die ausgeprägte Tip-Splitting Instabilität von DLA aufweisen, sondern erst eine gewisse Länge erreicht haben müssen, bevor sie Seitenzweige ausbilden können. Dies stimmt mit experimentellen Beobachtungen überein, bei der man auch Side-Branching als dominierende Verzweigungsart beobachten kann. Zusätzlich

finden wir, daß Seitenzweige auf der Außenseite von gekrümmten Zweigen erscheinen, was ebenfalls mit experimentellen Beobachtungen übereinstimmt und besonders bei Van Damme und Lemaire [68] beobachtet werden kann. Aus der Unterdrückung des Tip-Splitting resultiert schließlich auch, daß die fraktale Dimension der erzeugten Cluster geringer als die fraktale Dimension von DLA ist, was ebenfalls mit anderen Beobachtungen übereinstimmt.

An den generierten Rissen wurden mit Hilfe des Radius of Gyration die fraktalen Eigenschaften bestimmt. Es konnte gezeigt werden, daß die generierten Rißstrukturen fraktal sind mit einer fraktalen Dimension, die deutlich unter der Dimension von DLA liegt. In einigen Simulationen werden sogar nur eindimensionale Strukturen generiert.

In diesem Modell werden im wesentlichen vier freie Parameter benutzt, von denen sich besonders zwei als wesentlich für die Generierung von Rissen mit korrekten Eigenschaften herausgestellt haben. Die ersten beiden Parameter σ_c und σ_t repräsentieren die Materialfestigkeit unter Druck und unter Zug und sind direkt mit den physikalischen Kohäsionskräften verknüpft.

Die beiden wesentlichen Parameter sind dagegen der Ermüdungs-Parameter M und der Bias-Parameter α. Der Ermüdungs-Parameter kontrolliert direkt die Ausbildung von Seitenzweigen, da er bestimmt in welchem Ausmaße eine kleine Last durch Materialermüdung zur Bildung eines großen Risses führen kann.

Den wesentlichsten Einfluß auf die globale Struktur der entstehenden Risse hat der Bias-Parameter α, mit dem die Anisotropie der Random-Walks gesteuert wird. Hier ist allerdings die physikalische Interpretation noch nicht klar. Ursprünglich wurde er eingeführt, um die Simulationen im Sinne eines Importance-Sampling zu beschleunigen. Allerdings stellte sich heraus, daß er sehr stark das relative Gewicht der äußeren und der inneren Randbedingungen verschiebt. Der Bias stellt eigentlich den einzigen freien Parameter dieser Methode dar, da nur er in der Lage ist, eine wesentliche strukturelle Veränderung der Risse hervorzurufen. Hier sind sicherlich weitere Studien interessant und sinnvoll, damit seine physikalische Bedeutung exploriert werden kann.

Besonders muß hervorgehoben werden, daß man mit diesem Modell in der Lage ist, experimentell beobachtbare Risse zu simulieren, wie im Absatz 6 "Vergleich mit Experimenten" des Kapitels 7 gezeigt wurde. Besonders der "Vier-Punk Scher-Test" ist hervorzuheben, da er einen Standard-Test im Ingenieur-Bereich darstellt. Ebenfalls müssen wir die Simulationsergebnisse des anisotropen Walks hervorheben. Hier können wir mit einem reinen Off-Lattice Modell sehr komplexe realistische Strukturen simulieren.

Neben diesem Rißmodell haben wir in Kapitel 6 ein Gittermodell studiert, welches das Wachstum von Rissen unter dem Einfluß einer Schockwelle beschreibt. Dazu benutzen wir eine Methode, die wegen der diskret gewählten Zeitschritte stark an Coupled-Maps erinnert. Dieses Modell ist interessant, da trotz seiner Einfachheit nicht-triviale und realistische Resultate erzielt werden können. Wir haben auf einem 1024×1024 Gitter eine singuläre Schockwelle im Gitterzentrum angeregt, worauf sich eine Schockwelle ausbildet, die radial nach außen wandert. In jedem Zeitschritt werden die jeweils am stärksten belasteten Bonds gebrochen. In umfangreichen Simulationen auf einer Connection Machine CM2–16k haben wir mit dieser Methode Risse mit mehr als zehntausend gebrochenen Bonds generiert und gezeigt, daß die so erzeugten Risse fraktal sind, deren fraktale Dimension linear von der Festigkeit der Bonds abhängt.

Die Untersuchung von Rißstrukturen mit Hilfe eines stochastischen Algorithmus erfordert sehr effiziente Methoden zur Generierung großer DLA Cluster. Daher wurde ein Programm entwickelt, welches die Generierung extrem großer DLA Cluster auf Workstations erlaubt. Mit diesem Programm wurden DLA Cluster mit 50 Millionen Teilchen simuliert. Dies ist ein Rekord für die Simulation von Off-Lattice DLA Clustern. In den Kapiteln 3–5 haben wir diese Möglichkeiten ausgenutzt uns daher mit drei Fragestellungen über DLA beschäftigt.

Im Kapitel 3 wurde die Frage des Multiscaling in DLA untersucht. Es wurde gezeigt, daß in DLA Clustern die in Schalen enthaltenen Massen mit unterschiedlichen Exponenten, die mit zunehmendem Abstand von Clusterursprung abnehmen, skalieren. Damit haben wir gezeigt, daß die bisher an kleinen DLA Clustern mit 100.000 Teilchen gewonnenen Ergebnisse auch für große Cluster gültig sind. Zusätzlich haben wir Multiscaling an einem sehr großen Cluster mit 50 Millionen Teilchen gemessen. Hier haben wir einen Hinweis gefunden, daß für sehr große Cluster der Übergang der Multiscaling-Dimensionen wesentlich schärfer als bei kleiner Clustern ist. Wir haben bemerkt, daß die Messung von Multiscaling sehr empfindlich auf die Isotropie von DLA Clustern reagiert: Ein schlechter Zufallszahlengenerator kann Multiscaling zerstören.

Multiscaling ist eine Konsequenz des besonderen Skalenverhaltens der radialen Wachstumswahrscheinlichkeiten $p(r)$. Plischke und Rácz [88]haben diese Verteilung an kleinen Clustern gemessen und eine gaußförmige Verteilung gefunden. Wir haben im Kapitel 4 bestätigt, daß das Maximum der Verteilung gut durch eine Gauß-Form beschrieben wird und mit umfangreichen Messungen and großen Clustern gefunden, daß die Breite ξ dieser Gauß-Kurve ein Verhalten der Form $\xi/R_g \propto 1/\sqrt{\ln M}$ oder $\xi/R_g \propto M^{\nu'}$ aufweist. Eine Unterscheidung dieser beiden Möglichkeiten ist mit unseren Daten nicht möglich. Durch Messung der relativen Breite an einem Cluster mit 50 Millionen Teilchen haben wir einen Hinweis darauf gefunden, daß die relative Breite für sehr große Clustermassen gegen einen konstanten Wert strebt. Wir haben weiter $p(r)$ im Clusterinneren — also für $r \to 0$ — untersucht und bestätigt, daß man im Inneren ein Potenzverhalten findet. Wir haben daher zwei Skalenansätze für $p(r)$ getestet und so gezeigt, daß der Exponent des inneren Teils wesentlich kleiner ist, als in bisherigen Messungen: Dort wurden lediglich effektive Exponenten gemessen.

Im Kapitel 5 haben wir die auffällige, baumartige Struktur von DLA Clustern mit einem hierarchischen Ordnungsschema analysiert. In diesem Schema werden DLA Cluster unterteilt in Zweige verschiedener Ordnung, die man mit Blättern, Zweigen, Ästen und Stämmen in Bäumen vergleichen kann. Wir haben bestätigt, daß sowohl die Länge, als auch die Anzahl von Zweigen einer bestimmten Ordnung n einer einfachen Gesetzmäßigkeit $X \propto r^n$ gehorchen. Zusätzlich haben wir die Breiten und Massen der Zweige studiert und das gleiche Ordnungsschema gefunden. Wir konnten so zeigen, daß in Off-Lattice DLA selbstähnliche Zweige existieren, deren Länge und Breite mit demselben Exponenten skalieren. Bisher war dieses Ergebnis lediglich für Lattice-DLA bekannt. Besonders interessant ist die Untersuchung der Winkel zwischen Zweigen verschiedener Ordnung. Wir konnten zeigen, daß auch diese Winkel einem strengen Ordnungsschema gehorchen und im Grenzfall unendlicher Clustermasse gegen einen asymptotischen Grenzwinkel von ca. 36° konvergieren. Dieses Verhalten unterscheidet sich völlig von dem Verhalten der Rißstrukturen, bei denen 90° Winkel zwischen Zweigen verschiedener Ordnung dominieren.

Anhang A

Die Methode der komplexen Spannungsfunktionen

A.1 Der allgemeine Formalismus

Die Methode der komplexen Spannungsfunktionen beruht auf der Eigenschaft, daß alle Verschiebungsfelder $\vec{u}$, die Lösungen der Lamé-Gleichung sind, auch Lösungen der Bipotentialgleichung

$$\Delta\Delta\vec{u} = 0 \tag{119}$$

sind. Das kann man leicht durch Bildung der Divergenz und durch Anwendung des Laplace-Operators auf die Lamé-Gleichung (79) verifizieren. Da das Verzerrungsfeld $\underline{\underline{\epsilon}}$ und das Spannungsfeld $\underline{\underline{\sigma}}$ durch Linearkombination von Ortsableitungen des Verschiebungsfeldes gebildet werden, erfüllen sie notwendigerweise auch eine Bipotentialgleichung. Die Bipotentialgleichung spielt somit eine fundamentale Rolle in der Elastizitätstheorie.

Um den Zusammenhang zwischen den Spannungs- bzw. Verschiebungskomponenten und den komplexen Spannungsfunktionen herzustellen, kann man von der Formel von Goursat ausgehen. Sie besagt, daß sich eine beliebige reelle biharmonische Funktion $F(z,\overline{z})$ in der Ebene durch zwei analytische komplexe Funktionen $\phi(z)$ und $\chi(z)$ darstellen läßt und lautet

$$F(z,\bar{z}) = \Re\{\bar{z}\cdot\phi(z) + \chi(z)\} \tag{120}$$

Die Menge der möglichen Spannungsfunktionen $F(z,\overline{z})$ kann man erheblich einschränken, denn man weiß, daß das Spannungsfeld in Abwesenheit von äußeren Volumenkräften die Gleichgewichtsbedingung $\mathrm{Div}\,\underline{\underline{\sigma}} = 0$ zu erfüllen hat. Diese Bedingung kann identisch erfüllt werden durch die sogenannte Airysche Spannungsfunktion $F(x,y)$, die man gemäß

$$\sigma_{xx} = \frac{\partial^2 F}{\partial y^2} \qquad \sigma_{yy} = \frac{\partial^2 F}{\partial x^2} \qquad \tau_{xy} = -\frac{\partial^2 F}{\partial x \partial y} \tag{121}$$

definiert. Mit Hilfe des Hookeschen Gesetzes und der Definition des Verzerrungsfeldes $\underline{\underline{\epsilon}}$ kann man nun zeigen, daß in zwei Dimensionen das Spannungsfeld selbst einer Potentialgleichung genügen muß

$$\Delta(\sigma_{xx} + \sigma_{yy}) = 0 \tag{122}$$

und daß daher auch die Airysche Spannungsfunktion einer Bipotentialgleichung genügt

$$\Delta\Delta F = \frac{\partial^4 F}{\partial x^4} + 2\frac{\partial^4 F}{\partial x^2 \partial y^2} + \frac{\partial^4 F}{\partial y^4} = 0. \tag{123}$$

Also kann man die Airysche Spannungsfunktion mit dem oben definierten $F(z,\overline{z})$ identifizieren und erhält aus (120) und (121) die ersten beiden Kolossovschen Formeln

$$\begin{aligned} \sigma_{xx} + \sigma_{yy} &= 4\Re\big(\phi'(z)\big) \\ \sigma_{xx} - \sigma_{yy} + 2i\tau_{xy} &= -2\Big(z\overline{\phi''(z)} + \overline{\psi'(z)}\Big) \end{aligned} \tag{124}$$

mit den komplexen Spannungsfunktionen $\phi(z)$ und $\psi(z) = \chi'(z)$. Mit Hilfe dieser Formeln und dem Hookeschen Gesetz kann man nun die Verbindung zwischen den Verschiebungen und den komplexen Spannungsfunktionen herstellen und erhält die dritte Kolossovsche Formel

$$2\mu(u_x + iu_y) = \kappa\phi(z) - z\overline{\phi'(z)} - \overline{\psi(z)} \tag{125}$$

Die Konstante κ unterscheidet hier zwischen ebenen Spannungs- (ESZ) und ebenen Verzerrungszuständen (EVZ)

$$\kappa = \begin{cases} \frac{3-\nu}{1+\nu} & \text{bei ESZ} \\ 3 - 4\nu & \text{bei EVZ} \end{cases} \tag{126}$$

Die Formeln (124) und (125) bilden die Grundlagen der Methode der komplexen Spannungsfunktionen.

Man erkennt, daß ebene Probleme der Elastizitätstheorie auf die Ermittlung zweier analytischer komplexer Funktionen $\phi(z)$ und $\psi(z)$ zurückgeführt worden sind. Die Möglichkeiten bei der Wahl dieser beiden Funktionen wird durch die Angabe der Randbedingungen eingeschränkt. Aus (125) erhält man bei vorgegebenen Randverschiebungen $\vec{u}(s) = (u_x, u_y) = (g_1(s), g_2(s))$ sofort die Bedingung

$$\kappa\phi(z) - z\overline{\phi'(z)} - \overline{\psi(z)} = 2\mu(g_1 + ig_2) \tag{127}$$

Für vorgegebene Randspannungen $\vec{p} = (p_x, p_y)$ pro Linienelement ds erhält man aus (121) und dem Kräftegleichgewicht $\underline{\underline{\sigma}} \cdot \vec{n} = p \cdot ds$ die Beziehung

$$-i\Big[\phi(z) + z\overline{\phi'(z)} + \overline{\psi(z)}\Big] = \int (p_x + ip_y)\, ds \tag{128}$$

Für den Kraftfluß, d. h. die Resultierende $\vec{F}_{AB}$ der äußeren Kräfte an einem endlichen Randstück AB, lautet die Randbedingung dementsprechend

$$-i\Big[\phi(z) + z\overline{\phi'(z)} + \overline{\psi(z)}\Big]_A^B = \int_A^B (p_x + ip_y)\, ds = \vec{F}_{AB} \tag{129}$$

Den eben hergeleiteten Formalismus werden wir nun dazu benutzen, das Spannungsfeld zu bestimmen, daß in einem elastischen Kreis vom Radius R von einer Punktkraft $\vec{F}(0) = (F_x, F_y)$ hervorgerufen wird. Der Rand des Kreises möge dabei festgehalten sein: $\vec{u}(R) = 0$.

A.2 Punktkraft im elastischen Kreis

Den eben beschriebenen Formalismus wollen wir nun dazu benutzen, das Spannungsfeld zu bestimmen, daß in einer elastischen Kreisfläche vom Radius R von einer Punktkraft $\vec{F}(0) = (F_x, F_y)$ im Kreismittelpunkt erzeugt wird. Der Rand des Kreises möge dabei festgehalten werden $\vec{u}(R) = 0$.

Da man durch die Punktkraft einen singulären Punkt im Ursprung erhält, ist das betrachtete Gebiet zweifach zusammenhängend und damit lauten die allgemeinen Ansätze für die komplexen Potentiale $\phi(z)$ und $\psi(z)$

$$\begin{aligned} \phi(z) &= c\ln(z) + \phi_0(z) \\ \psi(z) &= -\kappa\overline{c}\ln(z) + \psi_0(z) \end{aligned} \tag{130}$$

mit zwei analytischen Funktionen $\phi_0(z)$ und $\psi_0(z)$. Aus der Randbedingung (129) kann die Konstante c bestimmt werden zu

$$c = \frac{(F_x + iF_y)}{2\pi(1+\kappa)}. \tag{131}$$

Aus der Randbedingung (127) erhält man für $\phi_0(z)$ und $\psi_0(z)$

$$\begin{aligned} \phi_0(z) &= -2c\ln(R) + \frac{\overline{c}}{\kappa}\frac{z^2}{R^2} \\ \psi_0(z) &= -2\frac{\overline{c}}{\kappa} \end{aligned} \tag{132}$$

woraus schließlich die beiden komplexen Potentiale folgen

$$\begin{aligned} \phi(z) &= c\ln(z) - 2c\ln(R) + \frac{\overline{c}}{\kappa}\frac{z^2}{R^2} \\ \psi(z) &= -\overline{c}\left(\kappa\ln(z) - \frac{2}{\kappa}\right) \end{aligned} \tag{133}$$

Mit Hilfe der ersten beiden Kolossovschen Formeln (124) kann man nun leicht das Spannungsfeld bestimmen.

Anhang B

DLA Programm

Im folgenden soll das komplette DLA Programm gezeigt werden, mit dem eine Simulation von Off-Lattice DLA Clustern mit bis zu 50.000.000 Teilchen möglich ist. Obwohl das Programm sehr umfangreich ist, habe ich mich entschlossen, nicht nur Auszüge abzudrucken, da der Code wegen der benutzten Datenstruktur sehr komplex ist. Leider gibt es keine Möglichkeit, Off-Lattice DLA in dieser Größenordnung in einem kleineren und einfacheren Programm zu implementieren.

Das Programm ist komplett für Simulationen auf SPARCstations unter SunOS ausgelegt und einige System-Calls werden benutzt, die nicht unbedingt mit Standard-UNIX kompatibel sind. Allerdings sind dies kleinere Calls, die nicht wesentlich sind, sondern Timing und Interrupt-Behandlung durchführen.

Der wesentliche Punkt zum Verständnis des Programms ist die Kenntnis der benutzten Datenstruktur, wie sie im Kapitel (3) beschrieben ist. Das Feld `map_` enthält die Kartenstruktur, in der der Cluster in immer feinerer Auflösung beschrieben wird. Das Feld `site_` enthält die verkettete Liste aller jener Teilchen, die sich innerhalb einer bestimmten Box befinden. Die Koordinaten der Teilchen sind im Feld `cluster_` gespeichert. Allerdings werden nicht alle Koordinaten im Speicher gehalten, sondern mit einer "least-recently-used" Strategie verwaltet. Die Verwaltung der residenten Speicherseiten erfolgt mit Hilfe des Feldes `stack_`.

In der gezeigten Programmversion ist zweidimensionale DLA mit Teilchen mit konstantem Radius gezeigt. Allerdings kann das Programm leicht auf höhere Dimensionen erweitert werden. Ebenso sollte die Simulation von DLA mit Teilchen mit variablem Radius möglich sein.

Das gesamte Programm ist auf verschiedene Quell-Files aufgeteilt, die mit einem `Makefile` zusammengebaut werden. Für jedes File ist daher der im `Makefile` benutzte Name angegeben.

Dies ist `common.dflt`. Hier werden alle globalen Variablen deklariert. Diese Datei muß erst mit dem `cpp` Präprozessor übersetzt werden, bevor sie benutzt werden kann.

```
/* pipe through cpp to install defaults */
/* cpp -P common.dflt > common.f */
/* this is automatically done in the makefile */

/* supply cpp defaults */
#ifndef M_CLUSTER
#define M_CLUSTER 1000
#endif
#ifndef M_MAP
#define M_MAP 200
#endif
#ifndef BUFFERED_PAGES
```

```
#define BUFFERED_PAGES 500
#endif
#ifndef D_PARTICLE
#define D_PARTICLE 1.d0/4.d0*length_0
#endif
/* defaults stripped by cpp */

c*********************************************************************
c
c     this is common.dflt
c
c     it contains a lot of global definitions
c     concerning parameters and arrays
c     it has to be included into every function or subroutine via
c
c     include 'common.f'
c
c     the resulting code is not compatible with normal fortran-77 code
c     due to the include command above.
c
c*********************************************************************

c*********************************************************************
c
c     general parameters
c
c*********************************************************************

      character version*(*)
      parameter (version='15')

      character rng_name*(*)
      parameter (rng_name='irand(3F) with shuffled table')

      character path*(*)
      parameter (path='data/')

      integer stdin, stdout, stderr
      parameter (stdin=5, stdout=6, stderr=0)

      integer cluster_unit, site_unit, stat_unit, next_unit, count_unit
      parameter (cluster_unit=20,
     &           site_unit=21,
     &           stat_unit=22,
     &           next_unit=23,
     &           count_unit=24)
      integer fd_cluster, fd_site, fd_count

      integer kbyte
      parameter (kbyte=1024)

c     this are the allowed signals recognized by 'output'
      integer
     &        statistics,
     &        update,
     &        initialize,
     &        end_of_simulation,
     &        interrupt,
     &        info_request
      parameter (
     &           statistics=-3,
     &           update=-2,
     &           initialize=-1,
     &           end_of_simulation=0,
     &           interrupt=2,
     &           info_request=30)

      integer x0fff,x0f
      parameter (x0f=2**8-1,x0fff=2**30-1)

      double precision pi, zweipi
      parameter (pi=3.14159265358979323846264 33d0, zweipi=2.d0*pi)

c*********************************************************************
c
c     simulation parameters
c
c*********************************************************************

c     dimension
      integer d
      parameter (d=2)

c     intended cluster mass
      integer m_cluster
      parameter (m_cluster=M_CLUSTER*kbyte)

c     maximum number of occupied map sites
      integer m_map
      parameter (m_map=M_MAP*kbyte)

c     size of paging space
      integer page_size, buffered_pages
      parameter (page_size=kbyte/2, buffered_pages=BUFFERED_PAGES*2)

c     number of map levels
c     if you change this you have to change the unrolled do loop
c     'touch_particle'
```

```
      integer maplevels
      parameter (maplevels=5)

c     dim_0 number of lattice sites on the lowest map level
      integer dim_0
      parameter (dim_0=2**((maplevels+1)+7))

c     length_0 lattice constant on map level 0
      double precision length_0
      parameter (length_0=1.d0)

c     particle radius and diameter
      double precision rparticle, dparticle
      parameter (dparticle=D_PARTICLE,
     &           rparticle=0.5d0*dparticle)

c     size of sin/cos-table
      integer sc_size, sc_bits
      parameter (sc_bits=14, sc_size=2**sc_bits-1)

c*********************************************************************
c
c     workspace declaration
c
c*********************************************************************

c     seed of random number generator
      integer*4 seed

c     n_cluster is the current size of the cluster
c     cluster_ contains the exact particle positions
c     site_    contains the index of the next particle in a list
c     next_    contains the index of each particles parent particle
      integer n_cluster
      real    cluster_(d,0:page_size-1,buffered_pages)
      integer*4 site_(0:page_size-1,buffered_pages)
      integer*4  next_(0:page_size-1)

c     translation tables for paging management
c     only a finite number of particle has to be kept in
c     memory. The stack is organized as a linked list!
c     Therefore it is very easy to remove or insert pages
      integer page_map(0:(m_cluster-1)/page_size)
      integer root, tail
      parameter (root=0, tail=buffered_pages+1)
      integer number, flag, previous, next
      parameter (number=0, flag=1, previous=2, next=3)
      integer stack_(number:next,root:tail)
      integer unchanged, changed
      parameter (unchanged=0, changed=1)
      integer on_disk, never_used
      parameter (on_disk=-2, never_used=-1)

c     n_map is the current size of the map-table
c     map_  contains the map-wood
c     base_ contains the roots of the trees in the map-wood
      integer   n_map
      integer*4 map_(0:1,0:1,0:m_map)
      integer   dim_base
      parameter (dim_base=dim_0/2**(maplevels+1))
      integer   base_(0:dim_base-1,0:dim_base-1)

c     length_(...) is the lattice constant on a given map level
      double precision length_(0:maplevels)

c     location of the seed of the cluster
      double precision r0(d)

c     table of sine and cosine
      double precision sin_(0:sc_size),cos_(0:sc_size)
      integer          phi, theta

c     measurement variables
      real             etime, dtime, etarray(2), dtarray(2), etme, dtme
      integer          access_faults
      double precision avg_site, avg_site_2, n_avg
      double precision rcluster, xymin, xymax

      common
     &       sin_, cos_,
     &       base_, next_,
     &       cluster_,
     &       page_map, stack_,
     &       r0,
     &       length_,
     &       rcluster, xymin, xymax,
     &       avg_site, avg_site_2, n_avg,
     &       etarray, dtarray, etme, dtme,
     &       phi,theta,
     &       seed,
     &       access_faults,
     &       fd_cluster, fd_site, fd_count,
     &       n_cluster, n_map,
     &       map_, site_
      external etime, dtime
```

Dies ist `dla.f`. Im wesentlichen enthält es nur Initialisierungsroutinen und eine Loop über

alle Teilchen, die aggregiert werden sollen.

```
c*********************************************************************
c
c     simulation of off-lattice dla
c     in d dimensions
c     currently only d=2 is implemented
c
c     the common declarations are in file
c          common.dflt
c     and are processed by c preprocessor
c
c*********************************************************************

      program dla
      include 'common.f'

      double precision pihalbe
      parameter (pihalbe = 0.5d0*pi)

      double precision r(d), drand
      integer touch_index, n_old, i, iargc

c     do some intialization stuff
      call output(initialize)
      call init()

c     store the seed
c     you can also do multi-seed dla by specifying the number of seeds
c     and specifying the range, over which the seeds should be distributed
c     ( see the 0*dparticle*(...) )
      do i = 1, 1
         r(1) = r0(1)+0*dparticle*(drand(0)-0.5d0)
         r(2) = r0(2)+0*dparticle*(drand(0)-0.5d0)
         call store(r,0)
      enddo

c     Here comes the whole DLA stuff :-) As easy as this!
c     do the walk and store the particle
      n_old = -1
      do while(n_cluster-1.lt.m_cluster)
         call walk(r,touch_index)
         call store(r,touch_index)
         if (mod(n_cluster-1,page_size).eq.0.and.n_old.ne.n_cluster) then
            call output(statistics)
            n_old = n_cluster
         endif
      enddo
      if (mod(n_cluster-1,page_size).ne.0) then
         call output(statistics)
      endif

      call output(end_of_simulation)
      stop
      end
```

Dies ist `init.f`. Hier werden die benutzten Variablen und der Zufallszahlengenerator initialisiert und die Interrupt-Handler aufgesetzt.

```
c*********************************************************************
c
c     initialise fields and parameters
c
c*********************************************************************

      subroutine init()
c     include common variables
      include 'common.f'

      double precision x, dx
      integer irand, lnblnk, hostnm
      integer i, j, signal, access, time, getpidh, getfd
      integer irand, zufall
      character fdate*24, name*80
      external output, fpe
      integer is, id
      double precision sinhelp, coshelp
      zufall()=rshift(irand(0),31-sc_bits)


c     startup message
      i=hostnm(name)
      write(stderr,*)
      write(stderr,*) '*** DLA-',version,' started at ',fdate(),
     &                ' on ',name(:lnblnk(name))
      write(stderr,'(a,i5.5)') '  Process-id is ',getpidh()
      write(stderr,*) '  The cluster size will be ',m_cluster

      if (m_cluster.gt.x0fff) then
         write(stderr,*) '  Sorry, cluster size too large!'
         stop
      endif

c     look for data subdirectory
c     all the data is written to a subdirectory called data
      if (access(path,' ').ne.0) then
```

```
         write(stderr,*) '  Sorry, can''t access the ',
     &                   path,' subdirectory.'
         stop
      endif


c     install output() as ctrl-c handler
c     the program can be interrupted with a 'kill 2' command
c     the data already generated is not lost but written to disk
      if (signal(interrupt,output,-1).lt.0) then
         write(stderr,*) '  Sorry, can''t install signal',interrupt
         stop
      else
         write(stderr,*) '  Interrupt installed as signal ',interrupt
      endif

c     install 'kill -30' as information handler
c     during a program run the program can be checked using
c     'kill -30 program-id'
      if (signal(info_request,output,-1).lt.0) then
         write(stderr,*) ' Sorry, can''t install signal',info_request
         stop
      else
         write(stderr,*) '  Info installed as signal ',info_request
      endif

c     init parameters
c     get file descriptors
c     they are needed since I write some of the output
c     using operating system calls (only in C-libraries)
      fd_cluster=getfd(cluster_unit)
      fd_site=getfd(site_unit)
c     initialse timing
      etme=etime(etarray)
      dtme=dtime(dtarray)
c     initialise parameters like map- and clustersize
      n_cluster=1
      n_map=1
      avg_site=0.d0
      avg_site_2=0.d0
      n_avg=0.d0
      xymin=1.d30
      xymax=-1.d30
      rcluster=0.d0
      access_faults=0

c     random number generator started with some 'random' number
c     consisting of the time and the process-id
c     irand is supposed to produce a 31 bit integer number
c     this has to be considered in 'zufall' where I reduce this
c     31 bit number to a random number of appropriate size
c           sc_size = 2**sc_bits-1
      seed=and(time()*getpidh(),2**31-1)
      i=irand(seed)
c     do some warmup
      do j = 1, 10000
         i = zufall()
      enddo

c     I do not want to calculate trigonometric functions all the time
c     so use table of trigonometric functions
      dx=zweipi/(sc_size+1)
      do i=0,sc_size
         x = dble(i)*dx
         call d_sincos(x,sin_(i),cos_(i))
      enddo

c     the built in RNG might produce bad numbers
c     make sure that the center of mass of the cluster does not move
c     this problem is removed by shuffling the table
      do i=0,10*sc_size
         id = zufall()
         is = zufall()
         coshelp = cos_(id)
         cos_(id) = cos_(is)
         cos_(is) = coshelp
         sinhelp = sin_(id)
         sin_(id) = sin_(is)
         sin_(is) = sinhelp
      enddo

c     init paging management
c     all pages in memory are empty
      do i=root,tail
         stack_(number,i)=-1
         stack_(flag,i)=unchanged
         stack_(previous,i)=i-1
         stack_(next,i)=i+1
      enddo
      do i=0,(m_cluster-1)/page_size
         page_map(i)=never_used
      enddo

c     init lattice constants of each map_level
      length_(0)=length_0
      do i=1,maplevels
         length_(i)=2.d0*length_(i-1)
      enddo
```

```
c     init seed position
      do i=1,d
         r0(i)=length_0*dim_0/2.d0
      enddo

      return
      end
```

Dies ist `output.f`. Hier werden Meldungen geschrieben, Interrupts behandelt und Daten geschrieben.

```
c*********************************************************************
c
c     output routine
c     open and update the necessary output files
c     this routine is called if you send a kill or info signal
c     to the program
c
c*********************************************************************

      subroutine output(signal)
      include 'common.f'
      integer signal

      integer this_page, page, offset, particle
      integer getpidh
      integer wr_, fseek, r
      character pid*5, fdate*24
      integer BOF, EOF
      parameter (BOF=0, EOF=2)
      double precision sqr, x
      sqr(x)=x*x

c     get the process-id to identify the different jobs
c     the process-id is used to identify the correct data
c          clusters are grown in parallel
      write(pid,'(i5.5)') getpidh()

c     print some statistics
      if (signal.eq.statistics.or.signal.eq.interrupt) then
         etme=etime(etarray)
         dtme=dtime(dtarray)
         write(stat_unit,999) n_cluster-1,
     &        access_faults,
     &        etarray(1),
     &        etarray(2),
     &        avg_site/n_avg,
     &        sqrt(avg_site_2/n_avg-sqr(avg_site/n_avg)),
     &        n_map
c        write(next_unit,
c     &         rec=(n_cluster-2)/page_size+1)
c     &         (next_(i),i=0,page_size-1)

c     write newest data...
c        particle = (n_cluster-page_size/2)
c        page = particle/page_size
c        call get_page(particle, offset, this_page)
c        r = fseek(cluster_unit,page*(page_size*d*4),BOF)
c        r=wr_(fd_cluster, cluster_(1,0,this_page), page_size*d*4)

         avg_site=0.d0
         avg_site_2=0.d0
         n_avg=0.d0
c        flush buffer
         call flush(stat_unit)
 999  format(1x,2i8,1p,4g13.7,i8)
      endif

c     initialize and open output files
      if (signal.eq.initialize) then
c        first setup paging strategy of OS
         call anomalous_paging()

         open(unit=stat_unit,
     &        file=path//'statistics.'//pid)
         r=fseek(stat_unit,0,EOF)

         write(stat_unit,'(6a)')
     &        ' *** Statistics file for DLA-',version,
     &        ' run ',pid, ' of ',fdate()
         write(stat_unit,*) 'particles access_faults ',
     &        'user sys avg_site varianz n_map'

         open(unit=cluster_unit,
     &        file=path//'cluster.'//pid,
     &        form='unformatted',
     &        access='direct',
     &        recl=4*d*page_size)

         open(unit=site_unit,
     &        form='unformatted',
     &        access='direct',
     &        recl=4*page_size,
     &        status='scratch')
```

```
      open(unit=next_unit,
     &       file=path//'next.'//pid,
     &       form='unformatted',
     &       access='direct',
     &       recl=4*page_size)
      endif

c     terminate simulation
      if (signal.eq.end_of_simulation.or.signal.eq.interrupt) then
c        first set again normal paging strategy
         call normal_paging()

c        write the information file
         open(13,file=path//'readme.'//pid)
         write(13,*)
         write(13,*) '*** Description file for DLA-',version,' of ',fdate()
         write(13,*)
         write(13,1000) 'Dimension',d
         write(13,1000) 'RNG seed',seed
         write(13,*)     ' Generator is '//RNG_name
         write(13,1000) 'Maximal clustersize',m_cluster
         write(13,1000) 'Actual clustersize',n_cluster-1
         write(13,1000) 'Record length',page_size
         write(13,1000) 'Maximal mapsize',m_map
         write(13,1000) 'Actual mapsize',n_map-1
         write(13,1001) 'Total time',etime(etarray)
         write(13,1001) 'User time',etarray(1)
         write(13,1001) 'System time',etarray(2)
         write(13,1000) 'Number of maplevels',maplevels
         write(13,1000) 'Number of sites in the lowest level',dim_0
         write(13,1000) 'Number of access faults',access_faults
         write(13,1001) 'Edge length in the lowest level is',length_0
         write(13,1001) 'The particle diameter is',dparticle
         write(13,1001) 'Radius of the cluster is',rcluster
         write(13,1001) 'Smallest xy coordinate is',xymin
         write(13,1001) 'Largest xy coordinate is',xymax
         write(13,*)
         write(13,*) '    The coordinates are written to cluster.'//pid
         write(13,*) '    To read them please use'
         write(13,*) '          do page=1,',(n_cluster-1-1)/page_size+1
         write(13,*) '             read(unit,rec=page) (c_x(i),c_y(i)',
     &         'i=0,',page_size-1,')'
         write(13,*) '          enddo'
         write(13,*)
         close(13)

c        close all output files
         close(site_unit)
         close(stat_unit)
         close(next_unit)

c        write the data in memory
         do page=0,(m_cluster-1)/page_size
            this_page=page_map(page)
            if (this_page.ne.on_disk.and.
     &          this_page.ne.never_used) then
               if (stack_(flag,this_page).ne.unchanged) then
                  r=fseek(cluster_unit,page*(page_size*d*4),BOF)
                  r=wr_(fd_cluster,
     &                 cluster_(1,0,this_page),
     &                 page_size*d*4)
               endif
            endif
         enddo
         close(cluster_unit)

         stop
 1000    format(1x,a,t45,i11)
 1001    format(1x,a,t45,g20.7)

 2000    write(stderr,*) '*** Sh..., there was an error during the final output...'
         write(stderr,*) '*** Aborted...'
      endif

c     write some information stuff about the current simulation
      if (signal.eq.info_request) then
         write(stderr,*)
         write(stderr,*) ' *** Information for DLA-',version,' run ',pid
         write(stderr,*)
         write(stderr,*) 'Hi guy, are you curious about what I''m doing?'
         write(stderr,*) '  The cluster is ',
     &                   n_cluster-1,'(',m_cluster,')  particles big.'
         write(stderr,*) '  The map has ',n_map,'(',m_map,')  entries.'
         write(stderr,*) '  There were ',access_faults,' access faults.'
         write(stderr,*) '  The smallest (largest) xycoordiantes are ', xymin, xymax
         write(stderr,*) '  The radius of the cluster is ', rcluster
         write(stderr,*) '  I''m a workoholic so don''t disturb me!'
         write(stderr,*) 'Bye bye baby...'
         write(stderr,*)
         return
      endif

      return
      end
```

Dies ist `walk.f`. Hier wird ein Random-Walk bis zum Cluster-Rand hin durchgeführt. `r` ist

die endgültige Position des Walkers und `touch_index` der Index des berührten Teilchens.

```
c*********************************************************************
c
c     perform a new Random-Walk
c     return the position r of the particle after having touched
c     a particle of the cluster (touch_index)
c
c*********************************************************************

      subroutine walk(r,touch_index)
      include 'common.f'
      double precision r(d)
      integer touch_index

      double precision length, length0
      double precision cutoff, cutoff2, dist2
      integer i1, i2
      integer level
      integer m_index
      logical touch
      double precision sqr, x
      integer bit, j, k, l, map,
     &        irand, normalize, zufall
c      double precision bias, drand

      sqr(x)=x*x
      bit(j,k)=and(rshift(j,k),1)
      zufall()=rshift(irand(0),31-sc_bits)
      normalize(x)=int(x)
c     to save memory space map_ is of type byte*3
c     combine everything to give a full integer word
      map(j,k,l)=map_(j,k,l)

c     create a new Random-Walker
c     start on a circle around the cluster with a radius that is
c     somewhat larger than the cluster itself
c     the cutoff circle is somewhere far away
      length0 =       rcluster+5.d0*dparticle
      cutoff  = 10.d0*rcluster+10.d0*dparticle
      cutoff2 = sqr(cutoff)

 9999 length=length0
      r(1)=r0(1)
      r(2)=r0(2)

 2    continue
c     chose direction
      phi = zufall()
      r(1) = r(1)+length*cos_(phi)
      r(2) = r(2)+length*sin_(phi)

c*********************************************************************
c     start a new step of the walker
c*********************************************************************
c     search the top level
 100  level=maplevels
      i1 = normalize(r(1))
      i2 = normalize(r(2))

c     did the particle travel too far away
      dist2 = sqr(r(1)-r0(1))+sqr(r(2)-r0(2))
      if (dist2.gt.cutoff2) goto 9999
      if (i1.lt.0.or.i1.ge.dim_0.or.i2.lt.0.or.i2.ge.dim_0) then
c        particle fell off the edges of the mapped area
         length = sqrt(dist2)-length0
         goto 2
      endif

c     get the first pointer. if there is no particle around
c     then perform the step
      m_index = base_(rshift(i1,maplevels+1), rshift(i2,maplevels+1))
      if (m_index.eq.0) then
         length = max(length_(level), sqrt(dist2)-length0)
         goto 2
      endif

 1    continue
c     check all the levels. if there is no particle around
c     then perform the step.
      m_index=map(bit(i1,level),bit(i2,level),m_index)
      if (m_index.eq.0) then
         length=length_(level)
         goto 2
      endif
c     on the lowest level do a fine check
      if (level.eq.0) then
         if (touch(r,touch_index)) then
            return
         else
            goto 100
         endif
      endif
      level=level-1
      goto 1

      end
```

Dies ist `touch.f`. Hier wird exakt getestet, ob ein Random-Walker bereits ein Teilchen des Clusters berührt. Wenn der Walker ein Teilchen berührt, dann wird `touch=.true.` gesetzt und `r` und `touch_index` beinhalten die letzte Position des Walkers bzw. den Index des berührten Teilchens.

```
c*********************************************************************
c
c     check if a particle has touched the cluster
c     calculate the exact contact point of two moving discs
c
c*********************************************************************

      logical function touch(r,touch_index)
      include 'common.f'
      double precision r(d)
      integer touch_index

      double precision dr(d),delta(d)
      double precision t, tmin, length
      double precision dr2, dist2
      logical hit
      integer c_index
      integer i1, i2, imin(d), imax(d)
      integer page, offset

      integer
     &        irand, zufall,
     &        bit, normalize,
     &        j, k, l,
     &        site, map, map_0,
     &        level, killme, kill, getpid
      double precision sqr, x

      sqr(x)=x*x
      zufall()=rshift(irand(0),31-sc_bits)
      bit(j,k)=and(rshift(j,k),1)
      normalize(x)=int(x)
      killme()=kill(getpid(),interrupt)

      map(j,k,l)=map_(j,k,l)

c     this is an unrolled do loop to calculate the index
c     of a particle on the lowest (7th) map level.
      map_0(i1,i2)=map(bit(i1,0),bit(i2,0),
     &                map(bit(i1,1),bit(i2,1),
     &                 map(bit(i1,2),bit(i2,2),
     &                  map(bit(i1,3),bit(i2,3),
     &                   map(bit(i1,4),bit(i2,4),
     &                    map(bit(i1,5),bit(i2,5),
     &                     map(bit(i1,6),bit(i2,6),
     &                      base_(rshift(i1,maplevels+1),
     &                            rshift(i2,maplevels+1)))))))))

      site(j,k)=site_(j,k)

      hit=.false.
      tmin=rparticle
      touch_index=0

c     determine jump direction
      phi=zufall()
      dr(1)=cos_(phi)
      dr(2)=sin_(phi)

c     determine the area to search
      length=tmin+dparticle
      imin(1)=normalize(r(1)-length)
      imax(1)=normalize(r(1)+length)
      imin(2)=normalize(r(2)-length)
      imax(2)=normalize(r(2)+length)
      do 20 i2=imin(2), imax(2)
         do 10 i1=imin(1), imax(1)

c           this is an unrolled do-loop
            c_index=map_0(i1,i2)

c           check every particle that is recorded
            do while (c_index.gt.0.and.c_index.lt.x0fff)
c              find particle data in virtual memory...
               call get_page(c_index,offset,page)
               delta(1)=dble(cluster_(1,offset,page))-r(1)
               delta(2)=dble(cluster_(2,offset,page))-r(2)
               dr2=sqr(delta(1))+sqr(delta(2))
               t=delta(1)*dr(1)+delta(2)*dr(2)
c              security check for particle overlap
               if (dr2.le.sqr(dparticle)) then
                  write(stderr,*) '###error: particle overlap'
                  level=killme()
                  stop
               endif
               if (dr2.le.sqr(abs(tmin)+dparticle).and.t.gt.0.d0) then
                  dist2=dr2-sqr(t)
c                 is this particle close enough
                  if (dist2.le.sqr(dparticle)) then
                     t=sign(abs(t)-sqrt(sqr(dparticle)-dist2),t)
```

```
                    if (abs(t).lt.abs(tmin)) then
                       tmin=t
                       touch_index=c_index
                       hit=.true.
                    endif
                 endif
              endif
              c_index = site(offset,page)
           enddo

 10      continue
 20   continue

c     ok... now we checked all the particles...
c     tmin contains now the step length...
      r(1)=r(1)+tmin*dr(1)
      r(2)=r(2)+tmin*dr(2)
      touch=hit

      return
      end
```

Dies ist `store.f`. Hier wird ein Walker gespeichert, der das Cluster-Teilchen `touch_index` berührt hat und nun am Ort `r` sitzt.

```
c*********************************************************************
c
c     create new entries in the maps
c
c*********************************************************************

      subroutine store(r,touch_index)
      include 'common.f'
      double precision r(d)
      integer touch_index

      double precision sqr, x
      sqr(x)=x*x

c     mark new particle in the map structure
      call mark_maps(r)
c     insert the new particle into the list of particles
      call mark_site(r,touch_index)
c     do some on-line statistcs
      rcluster=max(rcluster,sqrt(sqr(r(1)-r0(1))+sqr(r(2)-r0(2))))
      xymin=min(xymin, r(1), r(2))
      xymax=max(xymax, r(1), r(2))

      return
      end
```

Dies ist `mark_maps.f`. Hier wird ein neues Teilchen am Ort `r` in die Kartenstruktur eingetragen.

```
c*********************************************************************
c
c     this routine marks the position of the new
c     particle in all map levels
c
c*********************************************************************

      subroutine mark_maps(r)
      include 'common.f'
      double precision r(d)

      integer i1, i2, imin(d), imax(d)
      integer step
      integer level, i
      integer m_index, h_index
      double precision length
      double precision l_1, l_2, l_level

      integer j, k, l,
     &        normalize,
     &        bit, byt,
     &        map,
     &        killme, kill, getpid
      double precision x, sqr

      sqr(x)=x*x
      killme()=kill(getpid(),interrupt)
c     get the discretized position of the particle with a
c     precision of k integer bits, where k is the number of
c     map levels
      normalize(x,k)=lshift(rshift(int(x),k),k)

      bit(j,k) = and(rshift(j,k),1)
      byt(j,k) = and(rshift(j,k*8),x0f)
      map(j,k,l)=map_(j,k,l)

c     scan through all the maplevels
      do 100 level=maplevels,0,-1
         length=length_(level)+dparticle
         step=normalize(length_(level),level)
```

```
         imin(1)=normalize(r(1)-length,level)
         imax(1)=normalize(r(1)+length,level)
         imin(2)=normalize(r(2)-length,level)
         imax(2)=normalize(r(2)+length,level)
         l_level=sqr(length+sqrt(0.5d0)*length_(level))

c     mark on each level all the surrounding sites
         do 20 i2=imin(2), imax(2), step
            l_2=sqr(dble(i2)+0.5d0*length_(level)-r(2))
            do 10 i1=imin(1), imax(1), step
               l_1=sqr(dble(i1)+0.5d0*length_(level)-r(1))
               if (l_1+l_2.gt.l_level) goto 10

c     get or set the root of the pointer tree
c     if necessary create a new entry in the maps
               m_index=base_(rshift(i1,maplevels+1), rshift(i2,maplevels+1))
               if (m_index.eq.0) then
                  base_(rshift(i1,maplevels+1), rshift(i2,maplevels+1))=n_map
                  m_index=n_map
                  n_map=n_map+1
               endif

c     mark the position of the particle in all higher map levels
c     if necessary create a new entry in the maps
               do 1 i=maplevels,level,-1
                  h_index=map_(bit(i1,i),bit(i2,i),m_index)
                  if (h_index.eq.0) then
                     if (i.ne.0) then
                        h_index=n_map
                        n_map=n_map+1
                     else
                        h_index=x0fff
                     endif
                     map_(bit(i1,i),bit(i2,i),m_index)=h_index
                  endif
                  m_index=h_index
 1             continue

 10          continue
 20       continue
 100   continue

c     do a security check
      if (n_map.ge.m_map) then
         write(stderr,*) '###error: map overflow'
         i=killme()
         stop
      endif

      return
      end
```

Dies ist `mark_site.f`. Hier wird eine neues Teilchen, das am Ort `r` sitzt und das Cluster-Teilchen `touch_index` berührt in die Liste der bekannten Teilchen eingetragen. Ebenso wird dieses Teilchen in das Feld `cluster_` eingetragen.

```
c*********************************************************************
c     write the position of the new particle to the lowest level map
c*********************************************************************

      subroutine mark_site(r,touch_index)
      include 'common.f'
      double precision r(d)
      integer touch_index

      integer m_index, s_index
      integer i1, i2
      integer level
      integer page, offset

      integer j, k,
     &        bit, byt,
     &        map,
     &        normalize
      integer  sqr, i
      double precision x

      sqr(i)=i*i
c     see mark_maps.f for comments
      normalize(x,k)=lshift(rshift(int(x),k),k)
      bit(j,k)=and(rshift(j,k),1)
      byt(j,k)=and(rshift(j,k*8),x0f)

c     calculate discretized position of the particle
      i1=normalize(r(1),0)
      i2=normalize(r(2),0)

c     get position on the lowest map level
      m_index=base_(rshift(i1,maplevels+1),
     &              rshift(i2,maplevels+1))
      do level=maplevels,1,-1
         m_index=map_(bit(i1,level),bit(i2,level),m_index)
      enddo
      s_index=map_(bit(i1,0),bit(i2,0),m_index)
```

```
c     store the particle position in the arrays
      call get_page(n_cluster,offset,page)
      stack_(flag,page)=changed
      cluster_(1,offset,page)=real(r(1))
      cluster_(2,offset,page)=real(r(2))
      site_(offset,page)=s_index
      next_(offset)=touch_index

      map_(bit(i1,0),bit(i2,0),m_index)=n_cluster

c     do some statistics
      if (s_index.ne.x0fff) then
         avg_site=avg_site+dble(touch_index)
         avg_site_2=avg_site_2+dble(touch_index)**2
         n_avg=n_avg+1.d0
      endif

c     the clustern has grown...
      n_cluster=n_cluster+1

      return
      end
```

Dies ist `paging.f`. Hier wird die gesamte Verwaltung der im Speicher gehaltenen Koordinaten durchgeführt. Die Routine `get_page` benutzt den `index` des benötigten Teilchens als Input, liest evtl. Daten von Platte. Schließlich wird die Speicherseite `page`, in der sich das Teilchen `index` befindet, und der `offset`, an welcher Stelle sich das Teilchen in der betreffenden Seite befindet, zurückgeliefert. Teilchen können NIEMALS nur über deren Index adressiert werden, sondern nur über diese virtuelle Speicherverwaltung.

```
c*********************************************************************
c     these routines do a virtual memory management
c     the data is organized in different pages. these routines
c     handle a stack of a certain amount of pages. if one page
c     is accessed the stack is consulted if this page is in memory.
c     if it is not, the page at the bottom of the stack is written
c     to disk and the new page is read from disk.
c     in any case the requested page becomes the new top of the
c     stack.
c*********************************************************************

      subroutine write_page(this_page)
      integer this_page
      include 'common.f'

      integer page
      integer BOF
      parameter (BOF=0)
      integer fseek, wr_, r
      external write

      page=stack_(number,this_page)
      if (stack_(flag,this_page).ne.unchanged) then
c         write(stdout,*) n_cluster,' write: ',page
         r=fseek(cluster_unit,page*(page_size*2*4),BOF)
         r=wr_(fd_cluster,cluster_(1,0,this_page),page_size*4*d)
         r=fseek(site_unit,page*(page_size*4*1),BOF)
         r=wr_(fd_site,site_(0,this_page),page_size*4*1)
      endif
      if (page.ne.never_used.and.page.ne.on_disk) then
         page_map(page)=on_disk
      endif
      return
      end

c*********************************************************************

      subroutine read_page(this_page,page)
      integer this_page, page
      include 'common.f'

      integer BOF
      parameter (BOF=0)
      integer fseek, re_, r
      external read

      access_faults=access_faults+1
c      write(stdout,*) n_cluster,' read: ',page
      stack_(flag,this_page)=unchanged
      stack_(number,this_page)=page
      page_map(page)=this_page
      r=fseek(cluster_unit,page*(page_size*2*4),BOF)
      r=re_(fd_cluster,cluster_(1,0,this_page),page_size*2*4)
      r=fseek(site_unit,page*(page_size*4*1),BOF)
      r=re_(fd_site,site_(0,this_page),page_size*4*1)
      return
      end

c*********************************************************************

      subroutine get_page(index,offset,base)
      integer index
```

```
      integer base,offset
      include 'common.f'

      integer next_page, this_page, previous_page
      integer div, j, k
      div(j,k)=j/k

      base     = div(index-1,page_size)
      offset   = mod(index-1,page_size)
      this_page = page_map(base)
      if (this_page.eq.on_disk) then
         this_page = stack_(previous,tail)
         call write_page(this_page)
         call read_page (this_page,base)
         page_map(base)=this_page
      elseif (this_page.eq.never_used) then
         this_page = stack_(previous,tail)
         call write_page(this_page)
         stack_(number,this_page)=base
         page_map(base)=this_page
      endif
      next_page=stack_(next,this_page)
      previous_page=stack_(previous,this_page)
      if (previous_page.ne.root) then
         stack_(next,previous_page)=next_page
         stack_(previous,next_page)=previous_page
         stack_(next,this_page)=stack_(next,root)
         stack_(previous,this_page)=root
         stack_(next,stack_(previous,this_page))=this_page
         stack_(previous,stack_(next,this_page))=this_page
      endif
      base=this_page
      return
      end
```

Dies ist `io.c`. Der gesamte Platten-input/ouput wird nicht über FORTRAN `read` und `write` Operationen durchgeführt, sondern die etwas schnelleren UNIX System-IO Routinen werden aufgerufen. Diese kleinen Routinen stellen ein Interface zu diesen Routinen her.

```
/* this is a small c routine for direct calling the UNIX
read and write functions. It is slightly faster than the
built in fortran read write functions */

int re__(d,buffer,n_bytes)
int *d, *n_bytes;
char *buffer;
{
  read(*d,buffer,*n_bytes);
}

int wr__(d,buffer,n_bytes)
int *d, *n_bytes;
char *buffer;
{
  write(*d,buffer,*n_bytes);
}

#include <sys/vadvise.h>
/* this is a c interface to the vadvise function
   use anomalous paging strategy... */
void anomalous_paging_()
{vadvise(VA_FLUSH);vadvise(VA_NORM);return;}

void normal_paging_()
{vadvise(VA_NORM);return;}
```

Schließlich und endlich kommt das benutzte `makefile`, mit dem alle Teile zusammengebaut werden können. Das gesamte Paket kann mit `make -k` übersetzt und installiert werden.

```
# The size of the cluster will be M_CLUSTER*1024
# Thus, for a one million particle DLA cluster set M_CLUSTER = 1000
M_CLUSTER = 10

# BUFFERED_PAGES defines how many particles of the cluster will actually be held in
# memory. For small clusters of a few thousand sites (say M_CLUSTER = 100) you can keep
# all particles in memory (BUFFERED_PAGES = 100).
# For clusters larger than that you can reduce this number. But BUFFERED_PAGES should
# always be at least half of M_CLUSTER... (Although smaller numbers also work, you only
# have to wait longer...)
BUFFERED_PAGES = 100

# M_MAP defines the memory used to discretize the cluster. The simulation will stop
# when you run out of memory. You can check, whether you define enough memory
# by inspecting the file readme.????? Look at the lines 'Actual mapsize ...'
# and 'Maximal mapsize ...'. Only when the actual mapsize is close to the maximal
# mapsize you might run into trouble.
M_MAP = 200

# D_PARTICLE defines the diameter of the particles. Large particle diameters (like
# 1.d0/1.d0*length_0) give a fast simulation which uses a lot of memory. Small particle
# diameters (1.d0/5.d0*length_0) give relatively slow simulation which require very
# few memory...
# ATTENTION: make sure that you use the correct syntax here, i.e. something like
```

```
#            a.d0/b.d0*length_0, where a and b are some numbers!!!!
D_PARTICLE = 1.d0/1.d0*length_0

# Here are some typical settings...
#
#   M_CLUSTER   BUFFERED_PAGES   M_MAP   D_PARTICLE
#      10            10           200     1.d0/1.d0*length_0
#      20            20           200     1.d0/1.d0*length_0
#      ..            ..           ...     ...
#     100           100           200     1.d0/1.d0*length_0
#     500           500           200     1.d0/5.d0*length_0
#    1000           500           200     1.d0/5.d0*length_0
#
#

# A last remark: Memory and run-time consumption of this program scale almost linearly
#                with the cluster mass M_CLUSTER. Thus you can easily check, whether your
#                M_MAP is set properly
# To run the program you generate a directory 'src', in which you keep all the sources
# and this makefile and a directory 'data', in which all simulation data will be written.
# Whenever you make any changes to the previous parameters you have to build a new program
# by typing 'make' in the directory 'src'. This will generate a new program...

OUTFILE = ../dla.grow.$(M_CLUSTER)
COMPILE = f77 -u -e -O4 -dalign -libmil -Bstatic -cg89
LOCAL_DEFINES = -DM_CLUSTER=$(M_CLUSTER) -DM_MAP=$(M_MAP) -DBUFFERED_PAGES=$(BUFFERED_PAGES) -DD_PARTICLE=$(D_PARTICLE)

dla: common.f dla.o init.o output.o walk.o touch.o paging.o\
        store.o mark_site.o mark_maps.o application.o io.o makefile
 $(COMPILE) dla.o init.o output.o walk.o touch.o paging.o\
        store.o mark_site.o mark_maps.o application.o io.o -o $(OUTFILE)
 strip $(OUTFILE)
common.f: common.dflt makefile
 cpp -P $(LOCAL_DEFINES) common.dflt > common.f
dla.o: dla.f common.f makefile
 $(COMPILE) -c dla.f
init.o: init.f common.f makefile
 $(COMPILE) -c init.f
output.o: output.f common.f makefile
 $(COMPILE) -c output.f
walk.o: walk.f common.f makefile
 $(COMPILE) -c walk.f
touch.o: touch.f common.f makefile
 $(COMPILE) -c touch.f
paging.o: paging.f common.f makefile
 $(COMPILE) -c paging.f
store.o: store.f common.f makefile
 $(COMPILE) -c store.f
mark_site.o: mark_site.f common.f makefile
 $(COMPILE) -c mark_site.f
mark_maps.o: mark_maps.f common.f makefile
 $(COMPILE) -c mark_maps.f
application.o: application.f common.f makefile
 $(COMPILE) -c application.f
io.o: io.c common.f makefile
 cc -O -c io.c


clean:
 rm *.o common.f *~
```

Literaturverzeichnis

[1] A. A. Griffith, Phil. Trans. Roy. Soc. London **A221**, 163 (1921).

[2] H. J. Herrmann and J. Kertész, Physica A **178**, 227 (1991).

[3] M. Barber, J. Donley, and J. S. Langer, Phys. Rev. A **40**, 366 (1989).

[4] R. Ball and R. Blumenfeld, Phys. Rev. Lett. **65**, 1764 (1990).

[5] C. Moukarzel and H. J. Herrmann, J. Stat. Phys., to appear, (1992).

[6] C. Moukarzel, *Laplacian growth on a random lattice*, Physica A, to appear, (1992).

[7] E. Schlangen and J. G. M. Van Mier, in Proceedings of MRS spring meeting, MRS, 1992.

[8] A. H. J. M. Vervuurt, E. Schlangen, and J. G. M. Van Mier, *Pull-out of steel snchors embedded in concrete*, HLRZ-preprint **60**, (1992).

[9] T. A. Witten and L. M. Sander, Phys. Rev. Lett. **47**, 1400 (1981).

[10] S. Roux, J. Stat. Phys **48**, 201 (1987).

[11] S. Chandrasekhar, Rev. Mod. Phys. **15**, 1 (1943).

[12] L. Paterson, Phys. Rev. Lett. **52**, 1621 (1984).

[13] G. Daccord, J. Nittmann, and H. E. Stanley, Phys. Rev. Lett. **56**, 336 (1986).

[14] L. Niemeyer, L. Pietronero, and H. J. Wiesmann, Phys. Rev. Lett. **52**, 1033 (1984).

[15] M. Matsushita and H. Fujikawa, Physica A **168**, 498 (1990).

[16] R. M. Brady and R. C. Ball, Nature **309**, 225 (1984).

[17] T. Vicsek and J. Kertész, Europhys. News **19**, 24 (1988).

[18] E. Ben-Jacob, G. Deutscher, P. Garik, N. D. Goldenfeld, and Y. Lareah, Phys. Rev. Lett. **57**, 1903 (1986).

[19] G. Radnozcy, T. Vicsek, L. M. Sander, and D. Grier, Phys. Rev. A **35**, 4012 (1987).

[20] W. A. Bentley and W. J. Humpreys, *Snow Crystals*, Dover Publications, 1962.

[21] A. Buka, J. Kertész, and T. Vicsek, Nature **323**, 424 (1986).

[22] G. Daccord, J. Nittmann, and H. E. Stanley, Phys. Rev. Lett. **56**, 336 (1986).

[23] E. Ben-Jacob, Y. Godbey, N. Goldenfeld, J. Koplik, H. Levine, T. Müller, and L. M. Sander, Phys. Rev. Lett. **55**, 1315 (1985).

[24] M. Matsushita, M. Sano, Y. Hayakawa, H. Honjo, and Y. Sawada, Phys. Rev. Lett. **53**, 286 (1984).

[25] Y. Sawada, A. Dougherty, and J. P. Gollub, Phys. Rev. Lett. **56**, 1260 (1986).

[26] P. Meakin, in Phase Transition and Critical Phenomena, C. Domb and J. L. Lebowitz, eds., vol. 12, Academic Press Inc., London, 1987.

[27] T. Vicsek, *Fractal growth phenomena*, World Scientific Publishing Co., Singapore, 1989.

[28] H. E. Stanley and N. Ostrowsky, eds., *Random Fluctuations and Pattern Growth — Proc. NATO ASI*, Kluwer Academic Publishers, Dordrecht, 1988.

[29] L. Pietronero and E. Tosatti, eds., *Fractals in Physics*, North Holland, Amsterdam, 1985.

[30] H. E. Stanley, *private mitteilung*.

[31] J. H. S. Hele-Shaw, Nature **58**, 34 (1889).

[32] P. Meakin, R. C. Ball, P. Ramanlal, and L. M. Sander, Phys. Rev. A **35**, 5233 (1987).

[33] P. Meakin, J. Feder, and T. Jøssang, *Radially biased dla*, preprint, (1991).

[34] R. C. Ball and R. M. Brady, J. Phys. A **18**, L809 (1985).

[35] J. Nittmann and H. E. Stanley, Nature **321**, 663 (1986).

[36] C. Tang, Phys. Rev. A **31**, 1977 (1985).

[37] J. Kertész and T. Vicsek, J. Phys. A **19**, L257 (1986).

[38] P. Meakin, Phys. Rev. A **36**, 332 (1987).

[39] J.-P. Eckmann, P. Meakin, I. Procaccia, and R. Zeitak, Phys. Rev. A **39**, 3185 (1989).

[40] P. Meakin, Phys. Rev. A **27**, 605 (1983).

[41] ——, Phys. Rev. A **27**, 1495 (1983).

[42] S. Tolman and P. Meakin, Phys. Rev. A **40**, 428 (1989).

[43] P. Ossadnik, Physica A **176**, 454 (1991).

[44] D. A. Kessler, J. Koplik, and H. Levine, Adv. Phys. **37**, 255 (1988).

[45] P. G. Saffmann and G. I. Taylor, Proc. R. Soc. Lond. A **245**, 312 (1958).

[46] G. I. Taylor and P. G. Saffmann, Quart. J. Mech. Appl. Math. **12**, 265 (1959).

[47] M. Muthukumar, Phys. Rev. Lett. **50**, 839 (1983).

[48] M. Tokayama and K. Kawasaki, Phys. Rev. Lett. **100A**, 337 (1984).

[49] H. G. E. Hentschel, Phys. Rev. Lett. **52**, 212 (1984).

[50] H. Gould, F. Family, and H. E. Stanley, Phys. Rev. Lett. **50**, 686 (1983).

[51] T. C. Halsey and M. Leibig, Phys. Rev. A, accepted for publication, (1992).

[52] P. Meakin, H. E. Stanley, A. Coniglio, and T. A. Witten, Phys. Rev. A **32**, 2364 (1985).

[53] S. Schwarzer, J. Lee, S. Havlin, H. E. Stanley, and P. Meakin, Phys. Rev. A **43**, 1134 (1991).

[54] S. Schwarzer, J. Lee, A. Bunde, S. Havlin, H. E. Roman, and H. E. Stanley, Phys. Rev. Lett. **65**, 603 (1990).

[55] B. B. Mandelbrot and C. Evertsz, Physica A **177**, 386 (1991).

[56] J. Lee and H. E. Stanley, Phys. Rev. Lett. **61**, 2945 (1988).

[57] R. Blumenfeld and A. Aharony, Phys. Rev. Lett. **62**, 2977 (1989).

[58] B. B. Mandelbrot and T. Vicsek, J. Phys. A **22**, L377 (1989).

[59] J. Lee, S. Havlin, and H. E. Stanley, Phys. Rev. A **45**, 1035 (1992).

[60] S. Schwarzer, J. Lee, and H. E. Stanley, *Growth probabilies in 3-d dla*, Phys. Rev. A, submitted.

[61] R. W. Hertzberg, ed., *Deformation and Fracture Mechanics of Engineering Materials*, John Wiley, New York, 1976.

[62] H. Liebowitz, ed., *Fracture*, vol. I-VII, Academic Press, New York, 1986.

[63] H. J. Herrmann and S. Roux, eds., *Statistical Models for the Fracture of Disordered Media*, North Holland, Amsterdam, 1990.

[64] J. C. Charmet, E. Guyon, and S. Roux, eds., *Disorder and Fracture — Proc. NATO ASI*, Plenum Press, New York, 1990.

[65] H. J. Herrmann, in Fractals and Disordered Systems, A. Bunde and S. Havlin, eds., Springer, Berlin, 1990.

[66] H. V. Damme and E. Lemaire, in Disorder and Fracture — Proc. NATO ASI, J. C. Charmet, E. Guyon, and S. Roux, eds., Plenum Press, New York, 1990.

[67] H. V. Damme, E. Alsac, and C. Laroche, C. R. Acad. Sci. Ser. II **309**, 11 (1988).

[68] E. Lemaire and H. V. Damme, C. R. Acad. Sci. Ser. II **39**, 859 (1989).

[69] H. V. Damme, in The Fractal Approach To Heterogeneos Chemistry, D. Avnir, ed., Wiley, 1989.

[70] H. V. Damme, C. Laroche, L. Gatineau, and P. Levitz, J. Physique **48**, 1121 (1987).

[71] H. Zhao and J. V. Maher, *Associating-polymer effects in a hele-shaw experiment*, Phys. Rev. A, submitted, (1992).

[72] L. D. . Landau and E. M. Lifshitz, *Theoretische Physik, Band VII: Elastizitätstheorie*, Akademie Verlag.

[73] W. Nowacki, *Theory of Micorpolar Elasticity*, Springer-Verlag, 1972.

[74] G. Sines, *Elasticity and Strength*, Allyn and Bacon, Inc., Boston, 1969.

[75] J. G. M. Van Mier, E. Schlangen, and M. B. Nooru-Mohamed, in FRAMCOS I, Z. P. Bažant, ed., London, 1992, Elsevier Publishers.

[76] S. Feng and P. N. Sen, Phys. Rev. Lett. **52**, 216 (1984).

[77] Y. Kantor and I. Webman, Phys. Rev. Lett. **52**, 1891 (1984).

[78] S. Roux and E. Guyon, J. Physique Lett. **46**, L999 (1985).

[79] H. J. Herrmann, J. Kertész, and L. de Arcangelis, Europhys. Lett. **10**, 147 (1989).

[80] G. Peng and D. Tian, preprint.

[81] E. Louis and F. Guinea, Europhys. Lett. **3**, 871 (1987).

[82] P. Meakin, G. Li, L. M. Sander, E. Louis, and F. Guinea, J. Phys. A **22**, 1392 (1989).

[83] E. L. Hinrichsen, A. Hansen, and S. Roux, Europhys. Lett. **8**, 1 (1989).

[84] K. J. Måløy, A. Hansen, E. L. Hinrichsen, and S. Roux, Phys. Rev. Lett. **68**, 213 (1992).

[85] P. Ossadnik, *Multiscaling analysis and width of the active zone of large off-lattice dla*, Physica A, (1993).

[86] A. Coniglio and M. Zannetti, Physica A **163**, 325 (1990).

[87] P. Meakin, J. Phys. A **18**, L661 (1985).

[88] M. Plischke and Z. Racz, Phys. Rev. Lett. **53**, 415 (1984).

[89] G. H. Ristow, private Mitteilung.

[90] T. Vicsek, F. Family, and P. Meakin, Europhys. Lett. **12**, 217 (1990).

[91] C. Amitrano, A. Coniglio, P. Meakin, and M. Zannetti, Phys. Rev. B **44**, 4974 (1991).

[92] P. Ossadnik and J. Lee, *Power law tail in the radial growth probability distribution in DLA*, J. Phys. A, submitted, (1992).

[93] T. C. Halsey, P. Meakin, and I. Procaccia, Phys. Rev. Lett. **56**, 854 (1986).

[94] C. Amitrano, A. Coniglio, and F. di Liberto, Phys. Rev. Lett. **57**, 1016 (1986).

[95] P. Meakin and L. M. Sander, Phys. Rev. Lett. **54**, 2053 (1985).

[96] P. Meakin, A. Coniglio, H. E. Stanley, and T. A. Witten, Phys. Rev. A **34**, 3325 (1986).

[97] P. Ossadnik, Phys. Rev. A **45**, 1058 (1992).

[98] J. Vannimenus and X. G. Viennot, J. Stat. Phys **54**, 1529 (1989).

[99] E. L. Hinrichsen, K. J. Måløy, J. Feder, and T. Jøssang, J. Phys. A **22**, L271 (1989).

[100] B. Derrida, V. Hakim, and J. Vannimenus, Phys. Rev. A, to be published (1991).

[101] H. J. Herrmann, D. C. Hong, and H. E. Stanley, J. Phys. A **17**, L261 (1984).

[102] P. Meakin, J. Kertész, and T. Vicsek, J. Phys. A **21**, 1271 (1988).

[103] C. Evertsz, Phys. Rev. A **41**, 1830 (1990).

[104] P. Meakin, I. Majid, S. Havlin, and H. E. Stanley, J. Phys. A **17**, L975 (1984).

[105] T. Vicsek, private Mitteilung.

[106] B. Derrida and V. Hakim, *Needle models of laplacian growth*, submitted to Phys. Rev. A, (1992).

[107] A. Arneodo, F. Argoul, J. F. Muzy, and M. Tabard, Physica A **188**, 217 (1992).

[108] J. M. Combes, A. Grossmann, and P. Tchamitchian, eds., *Wavelets*, Springer, 1988.

[109] A. Arneodo, F. Argoul, J. F. Muzy, and M. Tabard, *Uncovering fibonacci sequences in the fratal morphology of DLA*, submitted to Phys. Lett. A.

[110] F. Argoul, A. Arneodo, G. Grasseau, and H. L. Swinney, Phys. Rev. Lett. **61**, 2558 (1988).

[111] G. Li, L. M. Sander, and P. Meakin, Phys. Rev. Lett. **63**, 1322 (1989).

[112] P. Ossadnik, *Cellular automaton for the fracture of disordered media*, Int. J. Mod. Phys. C, to appear, (1993).

[113] E. Schlangen and J. G. M. van Mier, in Proc. Int. EPRI. Conf. on Dam Fracture, V. E. S. et. al., ed., 1991.

[114] E. H. Yoffe, Phil.Mag. **42**, 739 (1951).

[115] L. Rayleigh, Proc. Lond. Math. Soc. **17**, 4 (1885).

[116] B. Chopard, J. Phys. A **23**, 1671 (1990).

[117] N. H. Margolus, PhD thesis, MIT, 1987.

[118] R. W. Gerling and D. Stauffer, Int. J. Mod. Phys. C **2**, 799 (1991).

[119] S. C. Glotzer, D. Stauffer, and S. Sastry, Physica A **164**, 1 (1990).

[120] in Large Scale Computations in Quantum Chemistry and Physics — Proc. Namur SCF Conf., special issue of Int. J. Quant. Chem., J. M. A. et. al., ed., 1992.

[121] A. Mel'čuk, R. Giles, and H. Gould, Computers in Physics **May/June**, 311 (1991).

[122] P. Ossadnik, *New model for the crack growth using random walkers, in preparation*, (1992).

[123] B. B. Mandelbrot, in Fractals and Disordered Systems, special issue of Physica A, A. Bunde, ed., 1992.

[124] P. Ossadnik, unpublished.

[125] H. J. Herrmann, Physica A **163**, 359 (1990).

[126] J. Kertész, in Disorder and Fracture — Proc. NATO ASI, J. C. Charmet, E. Guyon, and S. Roux, eds., Plenum Press, New York, 1990.

[127] S. Roux, in Statistical Models for the Fracture of Disordered Media, H. J. Herrmann and S. Roux, eds., North Holland, Amsterdam, 1990.

[128] P. Meakin, G. Li, L. M. Sander, H. Yan, F. Guinea, O. Pla, and E. Louis, in Disorder and Fracture — Proc. NATO ASI, J. C. Charmet, E. Guyon, and S. Roux, eds., Plenum Press, New York, 1990.

[129] E. Louis, F. Guinea, and F. Flores, in Fractals in Physics, L. Pietronero and E. Tosatti, eds., North Holland, Amsterdam, 1985.

[130] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series and Products*, Academic Press, San Diego, 1979.

[131] Y. C. Fung, *Foundations of Solid Mechanics*, Prentice Hall, Englewood cliffs, 1965.

[132] N. I. Mushkelishvili, *Einige Grundaufgaben zur mathematischen Elastizitätstheorie*, Hanser, München, 1971.

[133] L. Fernandez, F. Guinea, and E. Louis, J. Phys. A **21**, L301.

[134] L. P. Kadanoff, J. Stat. Phys **39**, 267 (1985).

[135] T. Vicsek, Phys. Rev. Lett. **53**, 2281 (1984).

[136] P. Meakin, F. Family, and T. Vicsek, J. Colloid Interface Sci. **117**, 394 (1987).

[137] T. C. Halsey and M. Leibig, J. Chem Physics **92**, 3756 (1990).

[138] J. G. M. Van Mier, M. B. Nooru-Mohamed, and G. Timmers, *An experimental study of shear fracture and aggregate interlock in cementbased composites*, HERON **36**, 1 (1991).

[139] R. C. Ball, R. M. Brady, G. Rossi, and B. R. Thompson, Phys. Rev. Lett. **55**, 1406 (1985).

[140] E. Bouchaud, G. Lapasset, and J. Planés, Europhys. Lett. **13**, 73 (1990).

[141] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, Dover Publications, Inc., 1972.

An der Fertigstellung einer Arbeit, wie der vorliegenden, wirken immer eine Menge Leute mit, denen an dieser Stelle gedankt werden soll.

Besonderer Dank gilt meinem Doktorvater Prof. Dietrich Stauffer und meinem Betreuer Dr. Hans Herrmann, ohnen deren Rat und Hilfe diese Arbeit nicht hätte angefertigt werden können. Besonders ohne die stete Diskussionsbereitschaft, Hilfe und Ermunterung von Dr. Herrmann wäre diese Arbeit nie zu bis diesem Endpunkt gelangt. Ihm möchte ich meinen speziellen Dank aussprechen.

Ein weiterer Dank gebührt meiner Frau Martina, die mich mit viel Verständnis für manchen Verzicht durch die Arbeit begleitet hat. Auch ohne sie hätte ich nicht fertig werden können. Daneben sei auch meinen Eltern und meiner Schwester gedankt, die mich während meiner ganzen Ausbildung gestützt haben.

Einen großen Einfluß auf das Entstehen der Arbeit hatte das offene und anregende Klima im HLRZ. Hier seien meine Freunde und Mitstreiter aus der Vielteilchengruppe in den Dank eingeschlossen. Besonders hervorheben möchte ich außerdem Dr. Stéphane Roux, der zwar nur kurz im HLRZ war, von dem ich aber durch einige Diskussionen viel gelernt habe.

FORSCHUNGSZENTRUM JÜLICH GmbH

KFA

Jül-2745
März 1993
ISSN 0366-0885