

Effiziente numerische Simulation von Teilchensystemen

Bachelor-Arbeit
im 1-Fach Bachelorstudiengang Mathematik
der Mathematisch-Naturwissenschaftlichen Fakultät
der Christian-Albrechts-Universität zu Kiel

vorgelegt von
Margaretha Franck

Erstgutachter: Prof. Dr. Steffen Börm
Zweitgutachter: Prof. Dr. Thomas Slawig

Kiel im August 2010

Inhaltsverzeichnis

1	Einleitung	1
2	Physikalische Grundlagen	3
2.1	Grundlegende Annahmen und Notationen	3
2.2	Newtonsches Gravitationsgesetz	3
2.3	Bewegungsgleichung	4
3	Behandlung der Differentialgleichung	7
4	Effiziente Auswertung der Kraftfunktion	11
4.1	Idee	11
4.2	Interpolation im \mathbb{R}^d	12
4.3	Anwendung der Tensorinterpolation auf das Problem	16
4.4	Gebietszerteilung	20
4.5	Gesamtaufwand und Gesamtfehler	28
5	Implementierung	33
5.1	Programmaufbau	33
5.2	Testergebnisse	37
6	Fazit	43

Abbildungsverzeichnis

2.1	Einzel- und Gesamtgravitationskraft (schwarz bzw. rot) auf ein Teilchen ausgeübt von mehreren anderen Teilchen	4
4.1	Zwischen „weit“ entfernten Teilchen ist eine gute Approximation der Gravitationskraft möglich.	12
4.2	Beispiel eines Baumes. Grüne Knoten entsprechen Blättern.	21
4.3	Beispiel eines Clusterbaumes für die Indexmenge $\{1, 2, \dots, 6\}$. Die Bewertungen sind neben den zugehörigen Knoten abgebildet.	22
4.4	Durch Anwendung des Algorithmus' entstandene Quader und der zugehörige Clusterbaum im zweidimensionalen Fall für elf Teilchen und maximale Blattgröße eins.	25
4.5	Durchführung des Algorithmus' im zweidimensionalen Fall. Die Intervall links und unten entsprechen dem Quader auf den ersten fünf Stufen des Clusterbaums. Die Blöcke entsprechen den Knoten des Blockclusterbaums. Rote Blöcke sind unzulässig, weil die zugehörigen Intervalle identisch sind oder aneinandergrenzen. Grüne Blöcke sind zulässig.	27
4.6	16 gleichmäßig verteilte Teilchen. Quader zu den ersten drei Stufen des Clusterbaums im zweidimensionalen Fall. $h = \frac{H}{\sqrt{16}}$	30
4.7	Die rotgepunkteten 3^2 Quader bilden im zweidimensionalen Fall mit dem mittleren Quader Q_{t^*} unzulässige Knoten des Blockclusterbaumes.	31
5.1	Erstellung eines Clusterbaums	34
5.2	Erstellung eines Blockclusterbaums	35
5.3	Berechnung der Gravitationskraft für ein unzulässiges Blatt	35
5.4	Berechnung der Gravitationskraft für ein zulässiges Blatt	36
5.5	Berechnung der Gravitationskraft	37
5.6	Relativer Fehler für wachsende Stützstellenzahl bei $n = 50.000$, $r = 250$, $\eta = 1$	38
5.7	Rechenaufwand für verschiedene Teilchenzahlen bei $\eta = 1$	40
5.8	Rechenaufwand für verschiedene Anzahl von Stützstellen	41

Tabellenverzeichnis

5.1	Relativer Fehler für wachsende Stützstellenzahl bei $r = 250, \eta = 1$. . .	38
5.2	Relativer Fehler für wachsende Teilchenzahl bei $\eta = 1$	39
5.3	Rechenaufwand für verschiedene Teilchenzahlen bei $\eta = 1$	39
5.4	Rechenaufwand für wachsende Stützstellenzahl bei $r = 250, \eta = 1$. .	41

1 Einleitung

In der Natur treten häufig Wechselwirkungen zwischen mehreren Objekten auf, die durch physikalische Gesetze beschrieben werden können. Beispiele sind Ionen in Plasmen, Rauchpartikel in Qualm oder auch Sonnen im Weltraum. Diese Kräfte zwischen den Objekte führen zu Bewegungen, welche genauer analysiert und vorhergesagt werden sollen.

In der vorliegenden Bachelorarbeit werden Systeme untersucht, die nur durch die Gravitationskraft miteinander wechselwirken, also Systeme von Himmelskörpern. Das wohl bekannteste Beispiel ist unser Sonnensystem, bestehend aus acht Planeten und der Sonne. Doch die Zahl der beteiligten Körper lässt sich stark erhöhen, wenn man beispielsweise die Sterne in einem Sternhaufen — etwa 1.000 bis 1.000.000 — oder in einer Galaxie betrachtet, dort sind etwa 100 Millionen Sterne beteiligt.

Die Bewegung dieser Himmelskörper wird durch ein Differentialgleichungssystem beschrieben, das sich aus den zugrunde liegenden physikalischen Gesetzmäßigkeiten ergibt. Dies wird im zweiten Kapitel näher erläutert. Zur Vorhersage der Bewegung muss dieses Differentialgleichungssystem gelöst werden, womit sich das dritte Kapitel befasst. Als problematisch erweist sich hierbei der hohe Rechenaufwand. Im vierten Kapitel wird eine Methode vorgestellt, um diesen zu reduzieren. Die Implementierung dieser Methode wird im fünften Kapitel behandelt.

2 Physikalische Grundlagen

In diesem Kapitel werden grundlegende Annahmen und Notationen zur Problembeschreibung eingeführt und die physikalischen Zusammenhänge kurz erläutert.

2.1 Grundlegende Annahmen und Notationen

Es werden Systeme von Himmelskörpern, beispielsweise Planeten, Sonnen oder Galaxien, betrachtet, die durch die Gravitationskraft miteinander wechselwirken.

Hierbei sollen nur Systeme behandelt werden, in denen die Körper nicht kollidieren. Weiter wird angenommen, dass man die betrachteten Körper als Massepunkte auffassen kann, weswegen im Folgenden auch nicht mehr von Körpern, sondern von Teilchen gesprochen wird. Dies ist sinnvoll, da die Ausdehnung der Körper sehr viel kleiner ist als die Abstände untereinander. An der Gravitationskraft verändert sich dadurch nichts, weil sie nicht von der Ausdehnung der Körper abhängig ist. Dieses Vorgehen vereinfacht die Modellbildung, denn es wird eine Unterscheidung zwischen Schwerpunkts- und Relativbewegung vermieden.

Außerdem sei angenommen, dass relativistische Effekte vernachlässigbar sind. Dies ist ohne große Genauigkeitsverluste möglich, weil die Geschwindigkeiten der Teilchen weit unterhalb der Lichtgeschwindigkeit liegen.

Es seien $n \in \mathbb{N}$ Teilchen im \mathbb{R}^3 gegeben. Ihre Ortsvektoren zum Zeitpunkt t werden mit $x_i(t) = (x_{i,1}(t), x_{i,2}(t), x_{i,3}(t))^T$ und ihre Massen mit m_i für alle $i \in \{1, \dots, n\}$ bezeichnet. Zur Vereinfachung der Notation wird oft x_i anstelle von $x_i(t)$ geschrieben. Die Ableitung des Ortes nach der Zeit wird als Geschwindigkeit v und die Ableitung der Geschwindigkeit nach der Zeit als Beschleunigung a des Teilchens zum Zeitpunkt t bezeichnet, d.h. $v(t) = \dot{x}(t)$ und $a(t) = \dot{v}(t) = \ddot{x}(t)$. Mit F_i bezeichnet man die Gesamtkraft die auf das i -te Teilchen wirkt.

2.2 Newtonsches Gravitationsgesetz

Nach dem Newtonschen Gravitationsgesetz übt ein Teilchen j auf ein Teilchen i eine Kraft gemäß der Formel

$$F_i^j = \gamma m_i m_j \frac{x_j - x_i}{\|x_j - x_i\|_2^3}$$

aus, wobei γ die Gravitationskonstante ist. Die Kraft wirkt also proportional zum Abstandsquadrat in Richtung des Verbindungsvektors zwischen x_i und x_j .

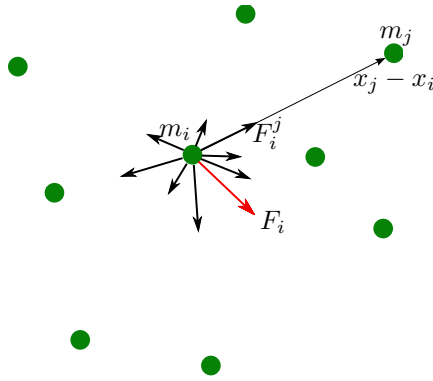


Abbildung 2.1: Einzel- und Gesamtgravitationskraft (schwarz bzw. rot) auf ein Teilchen ausgeübt von mehreren anderen Teilchen

Auf Teilchen i wirken alle übrigen $n - 2$ Teilchen nach demgleichen Gesetz. Mit Hilfe des Superpositionsprinzips, das besagt, dass sich die Einzelkräfte addieren, erhält man die Gesamtkraft auf das Teilchen i

$$F_i = \sum_{\substack{j=1 \\ i \neq j}}^n \gamma m_i m_j \frac{x_j - x_i}{\|x_j - x_i\|_2^3}. \quad (2.1)$$

2.3 Bewegungsgleichung

Das zweite Newtonsche Axiom besagt, dass die Kraft, die auf ein Teilchen wirkt, proportional zum Produkt aus Masse und Beschleunigung des Teilchens ist. Daraus ergibt sich die Bewegungsgleichung

$$F_i = m_i a_i = m_i \ddot{x}_i.$$

Durch Einsetzen der Gesamtkraft aus Gleichung (2.1) erhält man ein Differentialgleichungssystem 2. Ordnung mit $3n$ Gleichungen beschrieben durch:

$$\ddot{x}_i = \sum_{\substack{j=1 \\ i \neq j}}^n \gamma m_j \frac{x_j - x_i}{\|x_j - x_i\|_2^3} \text{ für } i \in \{1, \dots, n\}.$$

Dieses System kann in ein System 1. Ordnung mit $6n$ Gleichungen überführt werden. Hierzu definiert man für alle $i \in \{1, \dots, n\}$:

$$\hat{s}_i := \begin{pmatrix} x_i \\ \dot{x}_i \end{pmatrix}, \text{ dann ist } \dot{\hat{s}}_i = \begin{pmatrix} \dot{x}_i \\ \ddot{x}_i \end{pmatrix} = \begin{pmatrix} \dot{x}_i \\ \sum_{\substack{j=1 \\ i \neq j}}^n \gamma m_j \frac{x_j - x_i}{\|x_j - x_i\|_2^3} \end{pmatrix} = \begin{pmatrix} \hat{s}_{i,2} \\ \sum_{\substack{j=1 \\ i \neq j}}^n \gamma m_j \frac{\hat{s}_{j,1} - \hat{s}_{i,1}}{\|\hat{s}_{j,1} - \hat{s}_{i,1}\|_2^3} \end{pmatrix}.$$

Dieses System von Differentialgleichungen lässt sich wie folgt in einer Differentialgleichung zusammenfassen. Es werden

$$s(t) := \begin{pmatrix} x_1(t) \\ \vdots \\ x_n(t) \\ v_1(t) \\ \vdots \\ v_n(t) \end{pmatrix} \in \mathbb{R}^{6n} \text{ und } f : \mathbb{R}^{6n} \rightarrow \mathbb{R}^{6n}, f(s(t)) = \begin{pmatrix} (s_i(t))_{i=n+1, \dots, 2n} \\ \left(\sum_{\substack{j=1 \\ i \neq j}}^n \gamma m_j \frac{s_j(t) - s_i(t)}{\|s_j(t) - s_i(t)\|^3} \right)_{i=1, \dots, n} \end{pmatrix},$$

definiert, dann ist $\dot{s}(t) = f(s(t))$. (2.2)

3 Behandlung der Differentialgleichung

Für die Differentialgleichung (2.2) existiert bei gegebenen Anfangswerten für die Orte und Geschwindigkeiten nach dem Satz von Picard-Lindelöf auf geeigneten Teilbereichen des \mathbb{R}^{6n} eine Lösung, da die Funktion f lokal Lipschitz-stetig ist.

Lemma 1 (Lipschitz-Stetigkeit von f)

Seien $U_i, i = 1, \dots, n$, Teilmengen des \mathbb{R}^3 , sodass ein $d > 0$ existiert, mit der Eigenschaft, dass gilt:

$$\|x - y\|_2 > d \text{ für alle } x \in U_i, y \in U_j, i \neq j.$$

Dann gibt es ein $L > 0$, sodass für alle $s, \tilde{s} \in U_1 \times U_2 \times \dots \times U_n \times \mathbb{R}^{3n}$ gilt:

$$\|f(s) - f(\tilde{s})\|_\infty \leq L \|s - \tilde{s}\|_\infty,$$

mit f wie in (2.2) definiert.

Beweis

Seien s, \tilde{s} wie vorausgesetzt.

$$\text{Es ist } \|f(s) - f(\tilde{s})\|_\infty = \left\| \left(\begin{array}{c} (s_i - \tilde{s}_i)_{i=n+1, \dots, 2n} \\ \left(\sum_{\substack{j=1 \\ i \neq j}}^n \gamma \left(m_j \frac{s_j - s_i}{\|s_j - s_i\|_2^3} - \tilde{m}_j \frac{\tilde{s}_j - \tilde{s}_i}{\|\tilde{s}_j - \tilde{s}_i\|_2^3} \right) \right)_{i=1, \dots, n} \end{array} \right) \right\|_\infty.$$

Weil $\|f(s) - f(\tilde{s})\|_\infty = \max_{1 \leq i \leq 2n} \max_{1 \leq k \leq 3} \{|f(s)_{i,k} - f(\tilde{s})_{i,k}|\}$ ist, können zwei Fälle unterschieden werden.

1. Fall: Das Maximum wird in $(i, k) \in \{1, \dots, n\} \times \{1, 2, 3\}$ angenommen, dann ist

$$\|f(s) - f(\tilde{s})\|_\infty = |s_{i+n,k} - \tilde{s}_{i+n,k}| \leq \|s - \tilde{s}\|_\infty.$$

In diesem Fall ist $L = 1$.

2. Fall: Das Maximum wird in $(i, k) \in \{n+1, \dots, 2n\} \times \{1, 2, 3\}$ angenommen. Wähle

$M > 0$ so, dass für die Massen $m_i, i \in \{1, \dots, n\}$, $M \geq m$ ist, dann gilt

$$\begin{aligned} \|f(s) - f(\tilde{s})\|_\infty &= \left| \sum_{\substack{j=1 \\ i \neq j}}^n \gamma \left(m_j \frac{s_{j,k} - s_{i,k}}{\|s_j - s_i\|_2^3} - \tilde{m}_j \frac{\tilde{s}_{j,k} - \tilde{s}_{i,k}}{\|\tilde{s}_j - \tilde{s}_i\|_2^3} \right) \right| \\ &\leq M \frac{1}{d^3} \gamma \left| \sum_{\substack{j=1 \\ i \neq j}}^n (s_{j,k} - \tilde{s}_{j,k}) - (s_{i,k} - \tilde{s}_{i,k}) \right| \\ &\leq M \frac{1}{d^3} \gamma \sum_{\substack{j=1 \\ i \neq j}}^n |(s_{j,k} - \tilde{s}_{j,k})| + |(s_{i,k} - \tilde{s}_{i,k})| \leq 2M \frac{1}{d^3} \gamma (n-1) \|s - \tilde{s}\|_\infty. \end{aligned}$$

Hier ist $L = 2M \frac{1}{d^3} \gamma (n-1)$.

□

Der Satz von Picard-Lindelöf liefert die Existenz und Eindeutigkeit der Lösung, gibt aber keine explizite Darstellung an. Dies reicht für die Anwendung nicht aus, da hier eine Darstellung der Lösung benötigt wird. Man stellt fest, dass für Systeme, die aus mehr als zwei Teilchen bestehen, keine exakte Lösung der Differentialgleichung bekannt ist. Deswegen greift man auf numerische Verfahren zur Lösung von Differentialgleichungen zurück, um die Lösung zu approximieren.

In der folgenden Betrachtung soll dieses das explizite Eulerverfahren sein, da es sich um ein sehr einfach zu beschreibendes Verfahren handelt, bei dem aber die allgemein auftretende Schwierigkeit deutlich zu erkennen ist. Für eine Implementierung bevorzugt man andere Verfahren, wie etwa Runge-Kutta-Verfahren, die ein besseres Konvergenz- und Konsistenzverhalten bieten.

Explizites Eulerverfahren

Zunächst wird das vorliegende kontinuierliche Problem, $\dot{s}(t) = f(s(t)), t \in [a, b]$ mit Anfangswert $s(a) = s_0$, diskretisiert. Dazu zerteilt man das betrachtete Zeitintervall in k Teilintervalle, in diesem Fall äquidistant mit Unterteilungspunkten $t_i = a + hi$, $h = \frac{1}{k}$, $i \in \{0, \dots, n\}$. Nun definiert man rekursiv eine Näherungslösung $\tilde{s}(t)$ für $t \in \{t_0, \dots, t_k\}$

$$\begin{aligned} \tilde{s}(t_0) &:= s(t_0) = s_0 \\ \tilde{s}(t_{i+1}) &:= \tilde{s}(t_i) + hf(\tilde{s}(t_i)) \text{ für } i \in \{0, \dots, k-1\}. \end{aligned}$$

Der Rechenaufwand für dieses Verfahren hängt vom Aufwand für eine Auswertung von f ab, da in jedem Schritt des Verfahrens f einmal ausgewertet, sowie eine Skalarmultiplikation und eine Vektoraddition durchgeführt wird. Um f in den ersten $3n$ Komponenten auszuwerten, muss nichts berechnet werden, weil es sich um die bereits bekannten Geschwindigkeiten handelt. Für die letzten $3n$ Komponenten muss jedoch für jede Komponente (i, d) $\sum_{\substack{j=1 \\ i \neq j}}^n \gamma m_j \frac{s_{j,d} - s_{i,d}}{\|s_j - s_i\|_2^3}$ berechnet werden, was jeweils

einen größeren Aufwand als $n - 1$ verursacht. Da $F_i^j = -F_j^i$ ist, ergibt sich für einen Schritt des Eulerverfahrens ein Rechenaufwand, der höher ist als $\frac{n(n-1)}{2}$.

Alle üblichen numerischen Verfahren zur Lösung von Differentialgleichungen basieren auf einer Zeitdiskretisierung und der Auswertung der Funktion f zu den so erhaltenen Zeitpunkten. Deshalb kann der potenziell hohe Rechenaufwand in jedem Fall zu Effizienzproblemen führen.

4 Effiziente Auswertung der Kraftfunktion

Im vorherigen Kapitel wurde aufgezeigt, dass zum Lösen der Bewegungsgleichung in jedem Zeitdiskretisierungsschritt die Kraft jedes Teilchens auf jedes andere Teilchen berechnet werden muss. Diese Auswertung der Kraftfunktion führt zu einem Rechenaufwand von $\mathcal{O}(n^2)$ in jedem Lösungsschritt. Für typische Probleme mit großen Partikelanzahlen ist dies sehr hoch. Deshalb soll in diesem Kapitel analysiert werden, wie man die Kraftfunktion geeignet approximieren kann, um den Rechenaufwand zu verringern.

4.1 Idee

Die Kraftfunktion

$$\frac{F_i}{m_i} = \gamma \sum_{\substack{j=0 \\ i \neq j}}^n m_j \frac{x_j - x_i}{\|x_j - x_i\|_2^3}$$

wird im Wesentlichen von der Funktion

$$g : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^3, (x, y) \mapsto \frac{y - x}{\|y - x\|_2^3} \quad (4.1)$$

bestimmt, weshalb sich die Approximationsanalyse auf diese Funktion beschränken wird.

Falls x und y hinreichend weit voneinander entfernt sind, ist g auf einer Umgebung von (x, y) eine glatte Funktion, was auf eine gute Approximierbarkeit hindeutet.

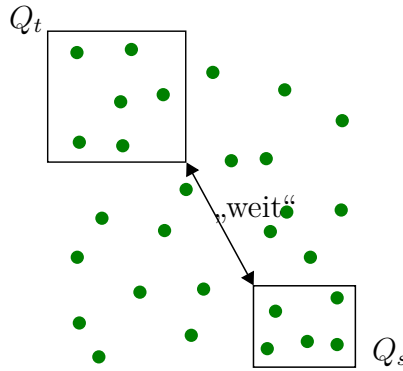


Abbildung 4.1: Zwischen „weit“ entfernten Teilchen ist eine gute Approximation der Gravitationskraft möglich.

Für zwei Quader $Q_t, Q_s \subseteq \mathbb{R}^3$, die „weit“ voneinander entfernt sind (Näheres dazu im Abschnitt 4.4), definiert man die Indexmengen $\hat{t} := \{i \in \{1, \dots, n\} : x_i \in Q_t\}$ und $\hat{s} := \{i \in \{1, \dots, n\} : x_i \in Q_s\}$. Diese enthalten die Indizes der Teilchen, die in den zugehörigen Quadern liegen. Betrachtet man den Abstand der Teilchen im Quader Q_t zu denen im Quader Q_s , so stellt man fest, dass sie alle „weit“ voneinander entfernt sind. Für ein Teilchen $x_i \in Q_t$ kann der Anteil der Gesamtkraft, der von den Teilchen aus Q_s ausgeübt wird, also gut approximiert werden. Wenn man annimmt, dass die Funktion g durch Interpolation mit Stützstellen $\xi_\nu, \nu \in M$, und Lagrange-Polynomen \mathcal{L}_ν approximiert werden kann (hierzu siehe Abschnitt 4.2), erhält man als Teilkraft

$$\frac{F_i^{\hat{s}}}{m_i} = \gamma \sum_{j \in \hat{s}} m_j g(x_i, x_j) \approx \gamma \sum_{j \in \hat{s}} m_j \sum_{\nu \in M} g(\xi_\nu, x_j) \mathcal{L}_\nu(x_i) = \gamma \sum_{\nu \in M} \mathcal{L}_\nu(x_i) \underbrace{\sum_{j \in \hat{s}} m_j g(\xi_\nu, x_j)}_{=: h_\nu}.$$

Hierbei sei M eine geeignete Indexmenge mit $\#M = k$. Der Term h_ν ist von i unabhängig und kann damit zur Berechnung der Teilkraft für jedes $i \in \hat{t}$ genutzt werden. Der Aufwand zur Bestimmung der h_ν ist $\mathcal{O}(k \cdot \#\hat{s})$. Um für jedes $i \in \hat{t}$ die Teilkraft zu berechnen, werden dazu noch $\mathcal{O}(k \cdot \#\hat{t})$ Rechenoperationen benötigt. Dies führt insgesamt zu einem Rechenaufwand von $\mathcal{O}(k(\#\hat{t} + \#\hat{s}))$. Dies ist für kleine k und große $\#\hat{t}, \#\hat{s}$ wesentlich günstiger als die direkte Berechnung in Form der Summe über die Einzelkräfte, die einen Aufwand von $\mathcal{O}(\#\hat{t} \cdot \#\hat{s})$ hat.

4.2 Interpolation im \mathbb{R}^d

Es folgt zunächst ein allgemein gehaltener Abschnitt über Interpolation im \mathbb{R}^d . Diese kann auf die Komponentenfunktionen von g angewendet werden.

Gegeben sei eine stetige Funktion $f : Q \rightarrow \mathbb{R}$, wobei Q ein achsenparalleler Quader im \mathbb{R}^d , $d \in \mathbb{N}$, sei, also von der Form $Q = [a_1, b_1] \times \dots \times [a_d, b_d]$ mit $a_i, b_i \in \mathbb{R}, a_i \leq b_i$.

Die Vorgehensweise ist nun, auf jedem eindimensionalen Intervall zu interpolieren und die dabei genutzten eindimensionalen Interpolationsoperatoren zu verketteten.

Es sei an die Interpolation im eindimensionalen Fall erinnert: Für eine stetige Funktion $\check{f} : [a, b] \rightarrow \mathbb{R}$ und Stützstellen $\xi_{[a,b],0}, \dots, \xi_{[a,b],m} \in [a, b]$ wird der Lagrange-Interpolationsoperator definiert durch

$$I_{[a,b]} : \mathcal{C} \rightarrow \Pi_m, \check{f} \mapsto \sum_{\nu=0}^m \check{f}(\xi_{[a,b],\nu}) \mathcal{L}_{[a,b],\nu}$$

mit den zu den Stützstellen gehörigen Lagrangepolynomen $\mathcal{L}_{[a,b],\nu}(x) = \prod_{\substack{\mu=0 \\ \mu \neq \nu}}^m \frac{x - \xi_{[a,b],\mu}}{\xi_{[a,b],\nu} - \xi_{[a,b],\mu}}$.

Partielle Interpolation

Um dies auf f zu übertragen, wird für alle $\iota \in \{1, \dots, d\}$ und $x = (x_1, \dots, x_d) \in \mathbb{R}^d$ die Funktion

$$f_\iota : [a_\iota, b_\iota] \rightarrow \mathbb{R}, y \mapsto f(x_1, \dots, x_{\iota-1}, y, x_{\iota+1}, \dots, x_d) \quad (4.2)$$

definiert. Hierauf kann $I_{[a_\iota, b_\iota]}$ angewandt werden. Der partielle Interpolationsoperator wird definiert durch:

$$I_{Q,\iota} : \mathcal{C}(Q) \rightarrow \mathcal{C}(Q),$$

$$I_{Q,\iota}[f](x) = I_{[a_\iota, b_\iota]}[f_\iota](x_\iota) = \sum_{\nu=0}^m f(x_1, \dots, x_{\iota-1}, \xi_{[a_\iota, b_\iota],\nu}, x_{\iota+1}, \dots, x_d) \mathcal{L}_{[a_\iota, b_\iota],\nu}(x_\iota).$$

Damit ist $I_{Q,\iota}[f]$ eine Funktion, die sich in der ι -ten Komponente wie ein Polynom verhält.

Tensorinterpolation

Durch Hintereinanderausführung der partiellen Interpolationsoperatoren angewandt auf f erhält man dann eine Funktion $I_Q[f]$, die sich in allen Komponenten wie ein Polynom verhält. Daher definiert man den Tensorinterpolationsoperator durch

$$I_Q = I_{Q,1} \circ \dots \circ I_{Q,d},$$

$$I_Q[f](x) = \sum_{\nu_1=0}^m \dots \sum_{\nu_d=0}^m f(\xi_{[a_1, b_1],\nu_1}, \dots, \xi_{[a_d, b_d],\nu_d}) \mathcal{L}_{[a_1, b_1],\nu_1}(x_1) \cdot \dots \cdot \mathcal{L}_{[a_d, b_d],\nu_d}(x_d).$$

Zur Verkürzung der Notation werden für alle $\nu \in M := \{0, \dots, m\}^d$

$$\xi_{Q,\nu} := (\xi_{[a_1, b_1],\nu_1}, \dots, \xi_{[a_d, b_d],\nu_d}) \text{ und } \mathcal{L}_{Q,\nu}(x) = \mathcal{L}_{[a_1, b_1],\nu_1}(x_1) \cdot \dots \cdot \mathcal{L}_{[a_d, b_d],\nu_d}(x_d)$$

definiert. Damit lässt sich der Interpolationsoperator kürzer schreiben als

$$I_Q[f] = \sum_{\nu \in M} f(\xi_\nu) \mathcal{L}_{Q,\nu}.$$

Fehlerabschätzung

Für den eindimensionalen Fall ist bekannt, dass die Wahl der Stützstellen als Tschebyscheff-Punkte zu einem optimalen Ergebnis führt. Der Konvergenzbeweis für die Tensorinterpolation baut auf die Konvergenz der partiellen Interpolation, die eindimensionalen Interpolationsoperatoren sind, auf. Daher ist es sinnvoll, als Stützstellen die Tschebyscheff-Punkte zu wählen, denn mit diesen gilt im eindimensionalen Fall für $\check{f} \in \mathcal{C}^{m+1}([a, b])$ folgende Fehlerabschätzung

$$\|\check{f} - I_{[a,b]}[\check{f}]\|_{\infty,[a,b]} \leq 2 \left(\frac{b-a}{4} \right)^{m+1} \frac{\|\check{f}^{(m+1)}\|_{\infty,[a,b]}}{(m+1)!}$$

und für $\check{f} \in \mathcal{C}([a, b])$ die Stabilitätsaussage $\|I_{[a,b]}[\check{f}]\|_{\infty,[a,b]} \leq \Lambda \|\check{f}\|_{\infty,[a,b]}$, wobei $\Lambda := \max\{\sum_{\nu=0}^m |\mathcal{L}_\nu(x)| : x \in [a, b]\}$ die Lebesgue-Zahl ist.

Daraus folgen im Wesentlichen die folgenden Abschätzungen:

Lemma 2 (Konvergenz und Stabilität der partiellen Interpolation)

Sei $\iota \in \{1, \dots, d\}$, dann gilt

$$\|f - I_{Q,\iota}[f]\|_{\infty,Q} \leq 2 \left(\frac{b_\iota - a_\iota}{4} \right)^{m+1} \frac{\|\frac{\partial^{m+1}}{\partial x_\iota^{m+1}} f\|_{\infty,Q}}{(m+1)!} \text{ für alle } f \in \mathcal{C}^{m+1}(Q)$$

sowie

$$\|I_{Q,\iota}[f]\|_{\infty,Q} \leq \Lambda \|f\|_{\infty,Q} \text{ für alle } f \in \mathcal{C}(Q).$$

Beweis

Zuerst zur Stabilitätsaussage:

Sei $f \in \mathcal{C}(Q)$, $\iota \in \{1, \dots, d\}$ und $x \in \mathbb{R}^d$. Mit $f_\iota : [a_\iota, b_\iota] \rightarrow \mathbb{R}$ wie in (4.2) definiert, ist $f(x) = f_\iota(x_\iota)$ und $I_{Q,\iota}[f](x) = I_{[a_\iota, b_\iota]}[f_\iota](x_\iota)$. Dann ist mit der Stabilitätsabschätzung für Interpolation im eindimensionalen Fall

$$|I_{Q,\iota}[f](x)| = |I_{[a_\iota, b_\iota]}[f_\iota](x_\iota)| \leq \|I_{[a_\iota, b_\iota]}[f_\iota]\|_{\infty,[a_\iota, b_\iota]} \leq \Lambda \|f_\iota\|_{\infty,[a_\iota, b_\iota]}.$$

Außerdem ist

$$\begin{aligned} \|f_\iota\|_{\infty,[a_\iota, b_\iota]} &= \max\{|f_\iota(y)| : y \in [a_\iota, b_\iota]\} \\ &= \max\{|f(x_1, \dots, x_{\iota-1}, y, x_{\iota+1}, \dots, x_d)| : y \in [a_\iota, b_\iota]\} \\ &\leq \max\{|f(y_1, \dots, y_d)| : y_l \in [a_l, b_l], 1 \leq l \leq d\} = \|f\|_{\infty,Q}. \end{aligned}$$

Durch Bildung des Maximums über alle $x \in Q$ erhält man die gewünschte Aussage.

Nun zur Konvergenzaussage:

Für $f \in \mathcal{C}^{m+1}$ ist $f_\iota^{(m+1)}(y) = \frac{\partial^{m+1}}{\partial x_\iota^{m+1}} f(x_1, \dots, x_{\iota-1}, y, x_{\iota+1}, \dots, x_d)$ für alle $y \in [a_\iota, b_\iota]$. Mit der Konvergenzabschätzung aus dem eindimensionalen Fall erhält man nun für

jedes $x \in Q$

$$|f(x) - I_{Q,\iota}[f](x)| = |f_\iota(x_\iota) - I_{[a_\iota, b_\iota]}[f_\iota](x_\iota)| \leq 2 \left(\frac{b_\iota - a_\iota}{4} \right)^{m+1} \frac{\|f_\iota^{(m+1)}\|_{\infty, Q}}{(m+1)!}.$$

Analog zur obigen Abschätzung der Supremumsnorm erhält man

$$\|f_\iota^{(m+1)}\|_{\infty, [a_\iota, b_\iota]} \leq \left\| \frac{\partial^{m+1}}{\partial x_\iota^{m+1}} f \right\|_{\infty, Q},$$

es folgt die Behauptung. □

Lemma 3 (Konvergenz und Stabilität der Tensorinterpolation)

Es gilt

$$\|f - I_Q[f]\|_{\infty, Q} \leq 2 \sum_{\iota=1}^d \Lambda^{\iota-1} \left(\frac{b_\iota - a_\iota}{4} \right)^{m+1} \frac{\left\| \frac{\partial^{m+1}}{\partial x_\iota^{m+1}} f \right\|_{\infty, Q}}{(m+1)!} \text{ für alle } f \in \mathcal{C}^{m+1}(Q),$$

sowie

$$\|I_Q[f]\|_{\infty, Q} \leq \Lambda^d \|f\|_{\infty, Q} \text{ für alle } f \in \mathcal{C}(Q).$$

Beweis

Die Aussage des Lemmas wird durch endliche Induktion bewiesen. Hierzu wird für alle $\iota \in \{0, \dots, d\}$ die Hilfsfunktion

$$P_\iota : \mathcal{C}(Q) \rightarrow \mathcal{C}(Q), P_\iota[f] := \begin{cases} f & , \text{ falls } \iota = 0 \\ I_{Q_\iota} P_{\iota-1}[f] & , \text{ sonst} \end{cases}$$

eingeführt. Es ist $P_d[f] = I_{Q_d} \circ P_{d-1}[f] = I_{Q_d} \circ I_{Q_{d-1}} \circ P_{d-2}[f] = \dots = I_Q[f]$.
Zunächst zum Beweis der Stabilitätsaussage, hierzu wird die Hilfsbehauptung

$$\|P_\iota[f]\|_{\infty, Q} \leq \Lambda^\iota \|f\|_{\infty, Q}$$

bewiesen, woraus nach obiger Überlegung sofort die Stabilitätsaussage folgt.

Sei $f \in \mathcal{C}(Q)$.

I.A.: Sei $\iota = 0$. Dann gilt $\|P_0[f]\|_{\infty, Q} = \|f\|_{\infty, Q} = \Lambda^0 \|f\|_{\infty, Q}$.

I.S.: Die Behauptung gelte für ein $\iota \in \{0, \dots, d-1\}$. Dann gilt für $\iota + 1$

$$\begin{aligned} \|P_{\iota+1}[f]\|_{\infty, Q} &= \|I_{Q_{\iota+1}} P_\iota[f]\|_{\infty, Q} \stackrel{\text{Lemma 2(2)}}{\leq} \Lambda \|P_\iota[f]\|_{\infty, Q} \\ &\stackrel{\text{I. Ann.}}{\leq} \Lambda \Lambda^\iota \|f\|_{\infty, Q} = \Lambda^{\iota+1} \|f\|_{\infty, Q}. \end{aligned}$$

Also ist die Zwischenbehauptung bewiesen, mit $\iota = d$ folgt die Stabilitätsaussage.

Nun zum Beweis der Konvergenzaussage. Man beachte, dass aufgrund der Kommutativität der partiellen Interpolationsoperatoren für alle $\iota \in \{1, \dots, d\}$ $P_\iota[f] = I_{Q_\iota} P_{\iota-1} = P_{\iota-1} I_{Q_\iota}$ gilt. Für $f \in \mathcal{C}^{m+1}$ ist:

$$\begin{aligned}
\|f - I_Q[f]\|_{\infty, Q} &= \|P_0[f] - P_d[f]\|_{\infty, Q} \\
&= \|(P_0[f] - P_1[f]) + (P_1[f] - P_2[f]) + \dots + (P_{d-1}[f] - P_d[f])\|_{\infty} \\
&= \left\| \sum_{\iota=1}^d P_{\iota-1}[f] - P_\iota[f] \right\|_{\infty, Q} \leq \sum_{\iota=1}^d \|P_{\iota-1}[f] - P_\iota[f]\|_{\infty, Q} \\
&= \sum_{\iota=1}^d \|P_{\iota-1}[f - I_{Q_\iota}[f]]\|_{\infty, Q} \stackrel{s.o.}{\leq} \sum_{\iota=1}^d \Lambda^{\iota-1} \|f - I_{Q_\iota}\|_{\infty, Q} \\
&\stackrel{\text{Lemma } 2(1)}{\leq} \sum_{\iota=1}^d \Lambda^{\iota-1} 2 \left(\frac{b_\iota - a_\iota}{4} \right)^{m+1} \frac{\|\frac{\partial^{m+1}}{\partial x_\iota^{m+1}} f\|_{\infty, Q}}{(m+1)!}.
\end{aligned}$$

□

Man sieht, dass sich die Fehler, die bei der Interpolation in einer Komponente auftreten, summieren und, da sie sich bei jeder weiteren partiellen Interpolation auf die Genauigkeit auswirken, verstärken.

4.3 Anwendung der Tensorinterpolation auf das Problem

Es kann also jede Komponentenfunktion von g durch Tensorinterpolation approximiert werden. Die Konvergenz der Tensorinterpolation ist abhängig vom Verhalten der $(m+1)$ -ten partiellen Ableitung in jede Koordinatenrichtung. Es ist daher sinnvoll das Verhalten der Ableitungen genauer zu betrachten.

Asymptotisch glatt

Definition 1 (asymptotisch glatt)

Eine Funktion $f : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ heißt (C_{as}, c_0, σ) -asymptotisch glatt, falls es Konstanten $C_{as} \in \mathbb{R}_{>0}, c_0 \in \mathbb{R}_{\geq 1}, \sigma \in \mathbb{N}_0$ gibt, sodass f auf der Menge $\{(x, y) : x, y \in \mathbb{R}^d, x \neq y\}$ \mathcal{C}^∞ -Funktion ist und sich die Ableitungen für alle $\nu \in \mathbb{N}, \iota \in \{1, \dots, 2d\}, x, y \in \mathbb{R}^d, x \neq y$, durch

$$\left| \frac{\partial^\nu}{\partial x_\iota^\nu} f(x, y) \right| \leq C_{as} \frac{(\nu + \sigma - 1)! c_0^\nu}{\|x - y\|_2^{\nu + \sigma}}$$

abschätzen lassen.

Hierbei ist zu beachten, dass für $\iota > d$ die partiellen Ableitungen von f nach $y_{\iota-d}$ gemeint sind.

Lemma 4

Jede Komponentenfunktion $g_i, i \in \{1, 2, 3\}$, von

$$g : \{(x, y) \in \mathbb{R}^3 \times \mathbb{R}^3 : x \neq y\} \rightarrow \mathbb{R}^3, g(x, y) = \frac{y - x}{\|y - x\|_2^3}$$

ist $(1, 3, 2)$ -asymptotisch glatt.

Beweis

Es werden zunächst die partiellen Ableitungen von

$$\tilde{g} : \{(x, y) \in \mathbb{R}^3 \times \mathbb{R}^3 : x \neq y\} \rightarrow \mathbb{R}, \tilde{g}(x, y) = \frac{1}{\|x - y\|_2^3}$$

betrachtet.

Es ist für $\iota \in \{1, 2, 3\}$

$$\begin{aligned} \frac{\partial^\nu}{\partial x_\iota^\nu} \|x - y\|_2^{-3} &= \frac{d^\nu}{ds^\nu} \Big|_{s=0} \|x + se_\iota - y\|_2^{-3} \stackrel{\text{Subst. } w := \frac{x-y}{\|x-y\|_2}}{=} \frac{d^\nu}{ds^\nu} \Big|_{s=0} \left\| \|x - y\|_2 w + se_\iota \right\|_2^{-3} \\ &= \frac{1}{\|x - y\|_2^3} \frac{d^\nu}{ds^\nu} \Big|_{s=0} \left\| w + \frac{s}{\|x - y\|_2} e_\iota \right\|_2^{-3} \stackrel{\text{Subst. } t = \frac{s}{\|x-y\|_2}}{\text{Kettenregel}} = \frac{1}{\|x - y\|_2^{\nu+3}} \frac{d^\nu}{dt^\nu} \Big|_{t=0} \|w + te_\iota\|_2^{-3}. \end{aligned}$$

Für die Ableitungen nach den y -Koordinaten kommt man zum gleichen Ergebnis.

Die weiteren Betrachtungen gehen von $w \in \mathbb{R}^3$ mit $\|w\|_2 = 1$ aus. Da die euklidische Norm invariant unter orthogonalen Transformationen ist, gilt für eine orthogonale Transformation Q mit $Qw = e_1$

$$\frac{d^\nu}{dt^\nu} \|w + te_\iota\|_2^{-3} = \frac{d^\nu}{dt^\nu} \|Qw + t \underbrace{Qe_\iota}_{=:h}\|_2^{-3} = \frac{d^\nu}{dt^\nu} \left((1 + th_1)^2 + t^2(h_2^2 + h_3^2) \right)^{-3/2}.$$

Es folgen einige Vorüberlegungen zur Berechnung dieser Ableitungen.

Für $z \in \mathbb{C}$ mit $|z| < 1$ ist nach Taylorentwicklung:

$$\begin{aligned} (1 - z)^{-3/2} &= \sum_{n=0}^{\infty} A_n z^n \text{ mit } A_n = \frac{\prod_{j=0}^{n-1} \frac{3}{2} + j}{n!} \text{ und} \\ (1 - z)^{-3} &= \sum_{n=0}^{\infty} \hat{A}_n z^n \text{ mit } \hat{A}_n = \frac{\prod_{j=0}^{n-1} 3 + j}{n!} = \frac{(n+1)(n+2)}{2}. \end{aligned}$$

Durch Quadrieren erhält man mit dem Reihenprodukt nach Cauchy

$$\begin{aligned}\sum_{n=0}^{\infty} \hat{A}_n z^n &= (1-z)^{-3} = \sum_{n=0}^{\infty} A_n z^n \sum_{n=0}^{\infty} A_n z^n \\ &= \sum_{n=0}^{\infty} \sum_{k=0}^n A_k z^k A_{n-k} z^{n-k} = \sum_{n=0}^{\infty} z^n \sum_{k=0}^n A_k A_{n-k}.\end{aligned}$$

Daraus folgt mit Koeffizientenvergleich für alle n : $\hat{A}_n = \sum_{k=0}^n A_k A_{n-k}$.

$$\begin{aligned}\text{Es ist } (1+th_1)^2 + t^2(h_2^2 + h_3^2) &= \left(1 - t \underbrace{(-h_1 + i\sqrt{h_2^2 + h_3^2})}_{:=\zeta}\right) \left(1 - t(-h_1 - i\sqrt{h_2^2 + h_3^2})\right).\end{aligned}$$

Für t mit $|t| < 1$ ist auch $|t\zeta| < 1$ und damit:

$$\begin{aligned}\left((1+th_1)^2 + t^2(h_2^2 + h_3^2)\right)^{-3/2} &= \sum_{n=0}^{\infty} A_n t^n \zeta^n \sum_{n=0}^{\infty} A_n t^n \bar{\zeta}^n \\ &= \sum_{n=0}^{\infty} t^n \underbrace{\sum_{k=0}^n A_k A_{n-k} \zeta^k \bar{\zeta}^{n-k}}_{=:B_n}.\end{aligned}$$

$$\begin{aligned}\text{Also ist } \frac{d^\nu}{dt^\nu} \Big|_{t=0} \left((1+th_1)^2 + t^2(h_2^2 + h_3^2)\right)^{-3/2} &= \frac{d^\nu}{dt^\nu} \Big|_{t=0} \sum_{n=0}^{\infty} t^n B_n = \sum_{n=\nu}^{\infty} t^{n-\nu} \frac{n!}{(n-\nu)!} B_n \Big|_{t=0} = \nu! B_\nu.\end{aligned}$$

Für B_ν gilt:

$$\begin{aligned}|B_\nu| &= \left| \sum_{k=0}^{\nu} A_k A_{\nu-k} \zeta^k \bar{\zeta}^{\nu-k} \right| \leq \sum_{k=0}^{\nu} A_k A_{\nu-k} |\zeta^k| |\bar{\zeta}^{\nu-k}| \\ &= \sum_{k=0}^{\nu} A_k A_{\nu-k} \|h\|_2^2 = \sum_{k=0}^{\nu} A_k A_{\nu-k} = \hat{A}_\nu = \frac{(\nu+1)(\nu+2)}{2}.\end{aligned}$$

Insgesamt erhält man also

$$\frac{\partial^\nu}{\partial x_i^\nu} \|x-y\|_2^{-3} = \frac{1}{\|x-y\|_2^{\nu+3}} \frac{d^\nu}{dt^\nu} \Big|_{t=0} \left((1+th_1)^2 + t^2(h_2^2 + h_3^2)\right)^{-3/2} = \frac{1}{\|x-y\|_2^{\nu+3}} \nu! B_\nu.$$

Nun werden die partiellen Ableitungen der Komponentenfunktionen von g , $g_i(x, y) = \frac{y_i - x_i}{\|y - x\|_2^3}$, betrachtet. Sei $\iota \in \{1, \dots, 6\}$ und $i \in \{1, 2, 3\}$.

1. Fall: $\iota \neq i$ und $\iota \neq i + 3$, dann ist

$$\begin{aligned} \left| \frac{\partial^\nu}{\partial x_\iota^\nu} \frac{y_i - x_i}{\|y - x\|_2^3} \right| &= \left| (y_i - x_i) \frac{\partial^\nu}{\partial x_\iota^\nu} \frac{1}{\|y - x\|_2^3} \right| = \left| \frac{y_i - x_i}{\|y - x\|_2^{\nu+3}} \nu! B_\nu \right| \\ &\leq \frac{(\nu + 1)!}{2\|y - x\|_2^{\nu+2}} (\nu + 2) \leq 3^\nu \frac{(\nu + 1)!}{\|y - x\|_2^{\nu+2}}. \end{aligned}$$

2. Fall: $\iota = i$, dann ist

$$\begin{aligned} \left| \frac{\partial^\nu}{\partial x_\iota^\nu} \frac{y_\iota - x_\iota}{\|y - x\|_2^3} \right| &= \left| -\nu \frac{\partial^{\nu-1}}{\partial x_\iota^{\nu-1}} \frac{1}{\|y - x\|_2^3} + (y_\iota - x_\iota) \frac{\partial^\nu}{\partial x_\iota} \frac{1}{\|y - x\|_2^3} \right| \\ &\leq \left| \nu \frac{\partial^{\nu-1}}{\partial x_\iota^{\nu-1}} \frac{1}{\|y - x\|_2^3} \right| + \left| (y_\iota - x_\iota) \frac{\partial^\nu}{\partial x_\iota} \frac{1}{\|y - x\|_2^3} \right| \\ &\leq \frac{(\nu + 1)! \nu}{2\|y - x\|_2^{\nu+2}} + \frac{(\nu + 1)! (\nu + 2)}{2\|y - x\|_2^{\nu+2}} \\ &\leq \frac{(\nu + 1)! (\nu + 2)}{\|y - x\|_2^{\nu+2}} \leq 3^\nu \frac{(\nu + 1)!}{\|y - x\|_2^{\nu+2}}. \end{aligned}$$

3. Fall: $\iota = i + 3$ analog zum 2. Fall.

Also sind die Komponentenfunktionen (1, 3, 2)-asymptotisch glatt. □

Um die Konvergenzaussage zu konkretisieren, werden zusätzlich folgende Beschreibungen der Quader benötigt:

Definition 2 (Durchmesser und Abstand)

Der Durchmesser eines Quaders Q_t wird durch

$$\text{diam}(Q_t) := \max\{\|x - \tilde{x}\| : x, \tilde{x} \in Q_t\}$$

und der Abstand zwischen zwei Quadern Q_t und Q_s durch

$$\text{dist}(Q_t, Q_s) := \min\{\|x - y\| : x \in Q_t, y \in Q_s\}$$

definiert.

Es gilt für alle $x \in Q_t, y \in Q_s$: $\|x - y\| \geq \text{dist}(Q_t, Q_s)$.

Mit diesen Größen und Lemma 4 lässt sich die Konvergenzaussage der Tensorinterpolation, angewandt auf eine Komponentenfunktion g_i , konkretisieren.

Für die partiellen Ableitungen gilt folgende Abschätzung

$$\left| \frac{\partial^\nu}{\partial x_\iota} g_i(x, y) \right| \leq \frac{3^\nu (\nu + 1)!}{\text{dist}(Q_t, Q_s)^{\nu+2}}.$$

Damit ist auch

$$\left\| \frac{\partial^\nu}{\partial x_\iota} g_i \right\|_{\infty, Q_t \times Q_s} \leq \frac{3^\nu (\nu + 1)!}{\text{dist}(Q_t, Q_s)^{\nu+2}}. \quad (4.3)$$

Weiter ist mit $Q_t = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$ $b_\iota - a_\iota \leq \text{diam}(Q_t)$ für alle $\iota \in \{1, 2, 3\}$. Außerdem kann die Lebesguezahl (vgl.[1]Bsp. 6.19) durch

$$\Lambda \leq \frac{2}{\pi} \log(m + 1) + 1 \leq m + 1 \text{ abgeschätzt werden.}$$

Für die Konvergenzaussage mit Interpolation in der ersten Komponente gilt also:

$$\begin{aligned} \|g_\iota - I_{Q_t}[g_i]\|_{\infty, Q_t \times Q_s} &\leq 2 \sum_{\iota=1}^3 \Lambda^{\iota-1} \left(\frac{b_\iota - a_\iota}{4} \right)^{m+1} \frac{\| \frac{\partial^{m+1}}{\partial x_\iota^{m+1}} g_i \|_{\infty, Q_t \times Q_s}}{(m+1)!} \\ &\stackrel{(4.3)}{\leq} 2 \sum_{\iota=1}^3 \Lambda^{\iota-1} \left(\frac{b_\iota - a_\iota}{4} \right)^{m+1} \frac{3^{m+1} (m+2)!}{\text{dist}(Q_t, Q_s)^{m+3} (m+1)!} \\ &\leq 2 \sum_{\iota=1}^3 \Lambda^{\iota-1} \left(\frac{3 \text{diam}(Q_t)}{4 \text{dist}(Q_t, Q_s)} \right)^{m+1} \frac{m+2}{\text{dist}(Q_t, Q_s)^2} \\ &\leq 2 \sum_{\iota=1}^3 (m+1)^{\iota-1} \left(\frac{3 \text{diam}(Q_t)}{4 \text{dist}(Q_t, Q_s)} \right)^{m+1} \frac{m+2}{\text{dist}(Q_t, Q_s)^2} \\ &\leq 2 \cdot 3 \cdot (m+1)^2 \frac{m+2}{\text{dist}(Q_t, Q_s)^2} \left(\frac{3 \text{diam}(Q_t)}{4 \text{dist}(Q_t, Q_s)} \right)^{m+1} \\ &= C(m) \frac{1}{\text{dist}(Q_t, Q_s)^2} \left(\frac{3 \text{diam}(Q_t)}{4 \text{dist}(Q_t, Q_s)} \right)^{m+1}, \end{aligned} \quad (4.4)$$

wobei $C(m) := 2 \cdot 3(m+1)^2(m+2)$ ist und damit nur polynomiell von m abhängt. Die Konvergenz wird durch den Term $\left(\frac{3 \text{diam}(Q_t)}{4 \text{dist}(Q_t, Q_s)} \right)^{m+1}$ bestimmt. Daher liegt exponentielle Konvergenz vor, falls $\text{diam}(Q_t) < \frac{4}{3} \cdot \text{dist}(Q_t, Q_s)$ gilt. Diese Ungleichung bietet sich als Kriterium für „weit“ voneinander entfernt an.

4.4 Gebietszerteilung

Es wurde gezeigt, dass für Gruppen von Teilchen, die „weit“ voneinander entfernt liegen, die Gravitationskraftsfunktion gut interpoliert werden kann. In diesem Abschnitt wird das Vorgehen systematisiert, um alle wirkenden Kräfte zu summieren. Dies geschieht dadurch, dass der Bereich rekursiv in Quader unterteilt und diese

miteinander in Beziehung gesetzt werden. Außerdem wird der Begriff „weit“ voneinander entfernt präzisiert.

Clusterbaum

Um alle wirkenden Kräfte effizient zu erfassen und gegebenenfalls vorher zu approximieren, ist es sinnvoll, die Indexmenge und damit auch das Gebiet zu partitionieren. Das dazu gewählte Verfahren kann gut mit Hilfe von Bäumen beschrieben werden.

Definition 3 (Baum)

Sei V eine nichtleere, endliche Menge — die Knotenmenge — und $S : V \rightarrow \text{Pot}(V)$ eine Abbildung. Falls $t \in V$ und $t^* \in S(t)$ liegt, heißt t Vater von t^* und t^* Sohn von t . Die Menge der Söhne von t bezeichnet man mit $\text{sons}(t)$.

Eine Folge $(t_0, t_1, \dots, t_k) \in V^{k+1}$ heißt Pfad, falls t_{i+1} Sohn von t_i für alle $0 \leq i < k$ ist.

$T = (S, V)$ heißt Baum, falls es genau ein Element $r \in V$ gibt, das keinen Vater hat, und für jedes $t \in V$ genau ein Pfad von r nach t existiert. Den Knoten r nennt man Wurzel — $\text{root}(T)$ — von T .

Falls für jedes $t \in V$ eine Teilmenge \hat{t} aus einer geeigneten Indexmenge fixiert ist, heißt T bewerteter Baum.

Im Folgenden sei $t \in T$ äquivalent zu $t \in V$.

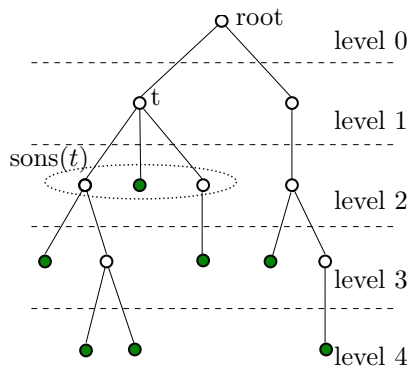


Abbildung 4.2: Beispiel eines Baumes. Grüne Knoten entsprechen Blättern.

Definition 4 (Blatt)

Ein Element $t \in T$ eines Baumes T heißt Blatt, falls es keine Söhne besitzt. Die Menge der Blätter von T wird mit \mathcal{L}_T bezeichnet.

Definition 5 (Nachfahren)

Es bezeichnet

$$\text{sons}^*(t) := \begin{cases} \{t\} & \text{falls } t \in \mathcal{L}_T \\ \{t\} \cup \left(\bigcup_{t^* \in \text{sons}(t)} \text{sons}^*(t^*) \right) & \text{sonst} \end{cases}$$

die Menge der Nachfahren eines Knotens $t \in T$.

Definition 6 (Stufe)

Die Abbildung

$$\text{level} : T \rightarrow \mathbb{N}_0, t \mapsto \begin{cases} \text{level}(t^*) + 1 & \text{falls } t \in \text{sons}(t^*) \\ 0 & \text{falls } t = \text{root}(T) \end{cases}$$

ordnet jedem Element des Baums eine Stufe (ein level) zu.

Definition 7 (Clusterbaum)

Sei I eine Indexmenge und T_I ein endlicher, bewerteter Baum. Zu jedem Knoten $t \in T_I$ sei $\hat{t} \subseteq I$ fixiert. Falls gilt:

- $\widehat{\text{root}}(T_I) = I$
- $t_1, t_2 \in \text{sons}(t)$ mit $t_1 \neq t_2$ impliziert $\hat{t}_1 \cap \hat{t}_2 = \emptyset$
- $\hat{t} = \bigcup_{t' \in \text{sons}(t)} \hat{t}'$,

so heißt T_I Clusterbaum zu I .

Die zu den Knoten gehörigen Indexmengen werden auch als Cluster bezeichnet.

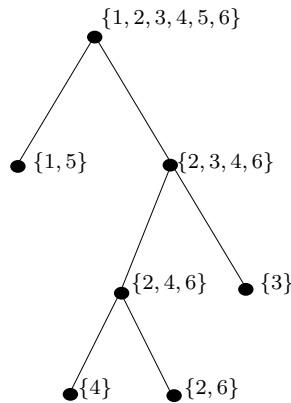


Abbildung 4.3: Beispiel eines Clusterbaumes für die Indexmenge $\{1, 2, \dots, 6\}$. Die Bewertungen sind neben den zugehörigen Knoten abgebildet.

Eine besondere Eigenschaft von Clusterbäumen, die für die Zerlegung der Gesamtkraft in Teilkkräfte nützlich ist, ist:

Lemma 5

Die zu den Blätter eines Clusterbaums T_I gehörigen Indexmengen bilden eine Partition der Ausgangsindexmenge, also $I = \dot{\bigcup}_{t \in \mathcal{L}_{T_I}} \hat{t}$.

Um dieses zu beweisen, müssen noch einige Struktureigenschaften gezeigt werden. Dazu sei im Folgenden I eine Indexmenge und T_I ein dazugehöriger Clusterbaum.

Lemma 6

Die Vereinigung der zu den Blätter eines Clusterbaums gehörenden Indexmengen ist die gesamte Indexmenge, d.h. $I = \bigcup_{t \in \mathcal{L}_{T_I}} \hat{t}$.

Beweis

Sei $i \in I$. Es wird die Menge D aller Knoten in deren Bewertung i auftritt $D := \{t \in T_I : i \in \hat{t}\}$ und $m := \max\{\text{level}(t) : t \in D\}$ definiert. Sei $t \in D$ mit $\text{sons}(t) \neq \emptyset$, dann existiert nach Definition ein $t^* \in \text{sons}(t)$ mit $i \in \hat{t}^*$. Es gilt $\text{level}(t) = \text{level}(t^*) + 1 \leq m - 1 < m$. Durch Kontraposition folgt also aus $\text{level}(t) = m$ $\text{sons}(t) = \emptyset$, also ist t ein Blatt.

Da nach Definition $\bigcup_{t \in \mathcal{L}_{T_I}} \hat{t} \subseteq I$ gilt, ist die Behauptung bewiesen. □

Lemma 7

Cluster auf derselben Stufe sind disjunkt, d.h. für $t, t' \in T_I$ mit $t \neq t'$ und $\text{level}(t) = \text{level}(t')$ gilt $\hat{t} \cap \hat{t}' = \emptyset$.

Beweis

Seien $t, t' \in T_I$ mit $t \neq t'$ und $\text{level}(t) = \text{level}(t') =: \ell$. Der Beweis wird über die Stufenzahl geführt.

I.A.: $\ell = 0$, dann ist $t = \text{root}(T_I) = t'$ und damit nichts zu zeigen.

I.S.: Die Behauptung gelte für $\ell = k$ für ein $k \in \mathbb{N}_0$. Für $\ell = k + 1$ existieren dann t^*, t'^* mit $t \in \text{sons}(t^*)$, $t' \in \text{sons}(t'^*)$. Es müssen zwei Fälle unterschieden werden:

1. Fall: $t^* = t'^*$, dann gilt nach Definition $\hat{t} \cap \hat{t}' = \emptyset$.

2. Fall: $t^* \neq t'^*$. Da $\text{level}(t^*) = \text{level}(t'^*) = k$ ist, folgt mit der Induktionsvoraussetzung $\hat{t}^* \cap \hat{t}'^* = \emptyset$. Weil $\hat{t} \subseteq \hat{t}^*$ und $\hat{t}' \subseteq \hat{t}'^*$ ist, gilt dann auch $\hat{t} \cap \hat{t}' = \emptyset$. □

Lemma 8

Für alle $t, t^* \in T_I$ folgt aus $\hat{t} \cap \hat{t}^* \neq \emptyset$ und $\text{level}(t) \leq \text{level}(t^*)$ $t^* \in \text{sons}^*(t)$.

Beweis

Die Behauptung wird per Induktion über die Differenz der Stufen bewiesen.

I.A.: $\text{level}(t^*) = \text{level}(t)$: Nach Lemma 7 gilt dann entweder $\hat{t} \cap \hat{t}^* = \emptyset$, dies steht aber im Widerspruch zur Voraussetzung, oder $t = t^*$ und damit $t^* \in \text{sons}^*(t)$.

I.S.: Die Behauptung gelte für $t, t^* \in T_I$ mit $\text{level}(t^*) - \text{level}(t) = l$.

Für $t, t^* \in T_I$ mit $\text{level}(t^*) - \text{level}(t) = l + 1$ gibt es ein $t' \in T_I$ mit $t^* \in \text{sons}(t')$, denn, da $\text{level}(t^*) > 0$ ist, ist $t^* \neq \text{root}(T_I)$. Weiter ist $\text{level}(t') - \text{level}(t) = \text{level}(t^*) - 1 - \text{level}(t) = l$ und $\emptyset \neq \hat{t} \cap \hat{t}^* \subseteq \hat{t} \cap \hat{t}'$. Mit der Induktionsvoraussetzung folgt also $t' \in \text{sons}^*(t)$ und damit ist auch $t^* \in \text{sons}^*(t)$. \square

Mit diesen Erkenntnissen kann nun Lemma 5 gezeigt werden.

Beweis von Lemma 5

Nach Lemma 6 ist $I = \bigcup_{t \in \mathcal{L}_{T_I}} \hat{t}$. Seien $t, t' \in \mathcal{L}_{T_I}$, es gelte o.B.d.A. $\text{level}(t) \leq \text{level}(t')$.

Angenommen $t \cap t' \neq \emptyset$, dann gilt nach Lemma 8 $t' \in \text{sons}^*(t) = \{t\}$, also $t = t'$. Damit ist entweder $t = t'$ oder $t \cap t' = \emptyset$ für alle $t, t' \in \mathcal{L}_{T_I}$. \square

Um das Gebiet in Quader zu zerlegen wird es, ausgehend von der Indexmenge $I = \{1, \dots, n\}$, so rekursiv unterteilt, dass ein zu I gehöriger Clusterbaum entsteht.

Ausgangspunkt ist ein Quader $Q_t = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$, der alle Teilchen umfasst. Im ersten Schritt wird dieser entlang der ersten Koordinatenrichtung in der Mitte geteilt, wozu der Mittelpunkt des ersten Intervalls $m_1 = \frac{a_1 + b_1}{2}$ berechnet und die Teilquader $Q_{t_1} := [a_1, m_1] \times [a_2, b_2] \times [a_3, b_3]$ sowie $Q_{t_2} := [m_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$ definiert werden. Die dazugehörigen Indexmengen sind dann $\hat{t}_1 := \{i \in I : x_i \in Q_{t_1}\}$ und $\hat{t}_2 := I \setminus \hat{t}_1$. Durch diese Definition wird vermieden, dass Partikeln, die in der Teilungsebene liegen, doppelt gezählt werden. Nach dem gleichen Prinzip werden Q_{t_1} und Q_{t_2} entlang der zweiten Koordinatenrichtung geteilt, die dabei entstehenden Quader werden dann nochmals entlang der dritten Koordinatenrichtung geteilt. Man erhält also $2^3 = 8$ Teilquader.

Nach diesem Schema wird für jeden der entstehenden acht Teilquader solange vorgegangen, bis sich weniger als eine bestimmte Anzahl von Teilchen — die maximale Blattgröße des Clusterbaums — in ihm befinden. Dann wird dieser Quader nicht weiter unterteilt.

Durch diesen Algorithmus erhält man einen Clusterbaum T_I zu I , dessen Knoten Mengen von Teilchenindizes und passende Quader zugeordnet sind.

Wie oben gezeigt, ist die Konvergenzgeschwindigkeit der Tensorinterpolation potentiell höher je geringer der Durchmesser des betrachteten Quaders ist. Um die Durchmesser der Quader schneller zu verkleinern, gibt es zum einen die Möglichkeit, nicht der Reihe nach entlang jeder Koordinatenrichtung zu unterteilen, sondern immer entlang derjenigen Richtung, für die die entstehenden zwei Teilquader minimalen Durchmesser haben würden. Zum anderen können die Quadergrößen nach der Teilung so verkleinert werden, dass sie genau die in ihnen liegenden Teilchen umfassen.

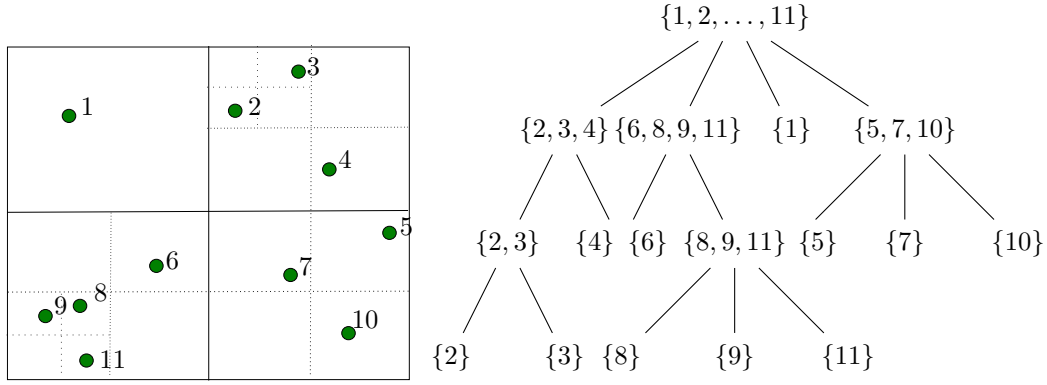


Abbildung 4.4: Durch Anwendung des Algorithmus' entstandene Quader und der zugehörige Clusterbaum im zweidimensionalen Fall für elf Teilchen und maximale Blattgröße eins.

Blockclusterbaum

Um die einzelnen Teilquader miteinander in Beziehung zu setzen definiert man:

Definition 8 (Blockclusterbaum)

Seien I, J endliche Indermengen und T_I, T_J zugehörige Clusterbäume. Sei $T_{I \times J}$ ein bewerteter Baum. Falls gilt

- für jedes $b \in T_{I \times J}$ existieren $t \in T_I, s \in T_J$ mit $b = (t, s)$
- $root(T_{I \times J}) = (root(T_I), root(T_J))$
- $b = (t, s)$ impliziert $\hat{b} = \hat{t} \times \hat{s}$
- für $sons(b) \neq \emptyset, b = (t, s)$, ist $sons(t) \neq \emptyset, sons(s) \neq \emptyset$ und $sons(b) = \{(t', s') : t' \in sons(t), s' \in sons(s)\}$,

dann heißt $T_{I \times J}$ Blockclusterbaum zu T_I und T_J .

Der Blockclusterbaum entsteht aus zwei Clusterbäumen und erhält dadurch ebenfalls die Struktur eines Clusterbaums, denn

- $\widehat{root}(T_{I \times J}) = \widehat{root}(T_I) \times \widehat{root}(T_J) = I \times J$.
- Für $b \in T_{I \times J}, b_1 = (t_1, s_1), b_2 = (t_2, s_2) \in sons(b)$ mit $b_1 \neq b_2$, ist $t_1, t_2 \in sons(t), s_1, s_2 \in sons(s)$ und o.B.d.A. $t_1 \neq t_2$. Weil T_I ein Clusterbaum ist, gilt dann $\hat{t}_1 \cap \hat{t}_2 = \emptyset$ und damit $\hat{b}_1 = \hat{t}_1 \times \hat{s}_1 \neq \hat{t}_2 \times \hat{s}_2 = \hat{b}_2$.
- Sei $b = (t, s) \in T_{I \times J}$ mit $sons(b) \neq \emptyset$. Dann ist $sons(t) \neq \emptyset$ und $sons(s) \neq \emptyset$, damit ist $\hat{b} = \hat{t} \times \hat{s} = \bigcup_{t^* \in sons(t)} \hat{t}^* \times \bigcup_{s^* \in sons(s)} \hat{s}^* = \bigcup_{b^* \in sons(b)} \hat{b}^*$.

Aus dieser Eigenschaft folgt wegen Lemma 5:

Lemma 9

Es ist $I \times J = \dot{\bigcup}_{b \in \mathcal{L}_{T_I \times J}} \hat{b}$.

In einem Blockclusterbaum $T_{I \times I}$, der zu einem zum vorliegenden Problem passenden Clusterbaum T_I gehört, tritt also jede mögliche Kombination von Indizes aus I in genau einem Blatt auf. Daher kann man die Kraft auf ein Teilchen i anstatt als

$$\frac{F_i}{m_i} = \gamma \sum_{\substack{j=1 \\ i \neq j}}^n m_j g(x_i, x_j) \text{ auch schreiben als } \frac{F_i}{m_i} = \gamma \sum_{\substack{(t,s) \in \mathcal{L}_{T_I \times I} \\ i \in \hat{t}}} \sum_{\substack{j \in \hat{s} \\ i \neq j}} m_j g(x_i, x_j). \quad (4.5)$$

Falls die zu den Indexmengen \hat{t}, \hat{s} gehörigen Quader Q_t, Q_s die Bedingung $\text{diam}(Q_t)/\text{dist}(Q_t, Q_s) \leq \eta$ für ein bestimmtes $\eta \in (0, 4/3)$ erfüllen, liegt bei der Interpolation von g exponentielle Konvergenz vor und der Rechenaufwand zur Bestimmung der Teilkraft kann, wie in Abschnitt 4.1 beschrieben, reduziert werden. Diese Bedingung bietet sich als Kriterium für „weit“ genug voneinander entfernt an und wird auch Zulässigkeitsbedingung im Sinn der folgenden Definition genannt.

Definition 9 (Zulässigkeitsbedingung) Sei $z : T_I \times T_J \rightarrow \{\text{wahr}, \text{falsch}\}$, dann heißt z Zulässigkeitsbedingung für T_I, T_J .

Ziel ist es, einen Blockclusterbaum zu konstruieren, der Blätter hat, die entweder aus möglichst großen Clustern bestehen, die die obige Bedingung erfüllen, oder, falls dies nicht möglich ist, aus kleinen Clustern. Im ersten Fall kann die Gravitationskraftfunktion zwischen ihnen interpoliert und somit Rechenaufwand gespart werden. Im zweiten Fall muss die Kraft direkt berechnet werden, was bei wenigen Teilchen mit vertretbarem Rechenaufwand möglich ist.

Bei der Konstruktion des Blockclusterbaumes geht man wieder rekursiv vor, beginnend mit $t = s = \text{root}(T_I)$. Falls t und s die obige Zulässigkeitsbedingung erfüllen, erhält man ein Blatt des Blockclusterbaums.

Sonst sind die Söhne von (t, s) alle Kombinationen von Söhnen von t und s . Falls t oder s bereits ein Blatt des Clusterbaums ist, wird (t, s) zu einem Blatt des Blockclusterbaums.

Für die entstandenen Söhne wird der Vorgang wiederholt.

Blätter, die dadurch entstehen, dass die Zulässigkeitsbedingung erfüllt wird, nennt man zulässig. Die übrigen, sowie die Knoten, die keine Blätter sind, heißen unzulässig.

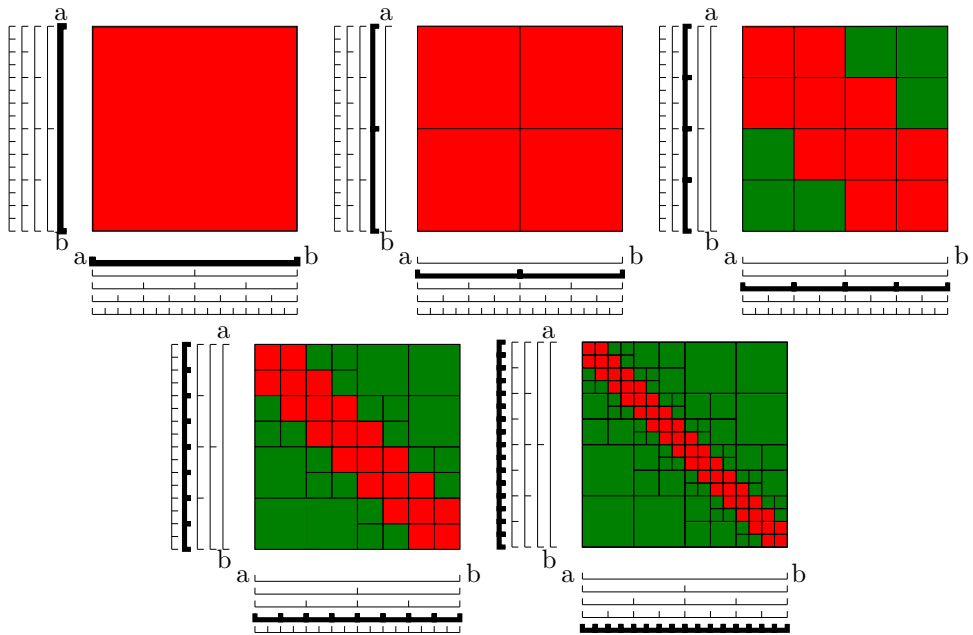


Abbildung 4.5: Durchführung des Algorithmus' im zweidimensionalen Fall. Die Intervall links und unten entsprechen dem Quader auf den ersten fünf Stufen des Clusterbaums. Die Blöcke entsprechen den Knoten des Blockclusterbaums. Rote Blöcke sind unzulässig, weil die zugehörigen Intervalle identisch sind oder aneinandergrenzen. Grüne Blöcke sind zulässig.

4.5 Gesamtaufwand und Gesamtfehler

In Abschnitt 4.1 bzw. Abschätzung (4.4) wurde gezeigt, dass sich auf den zulässigen Blöcken der Rechenaufwand bei einem kontrollierbaren Fehler jeweils verringert. Überträgt sich dies auch auf die komplizierter aussehende Darstellung des Problems mit Hilfe von Bäumen?

Gesamtaufwand

Zunächst zum Rechenaufwand. Dazu einige Vorüberlegungen:

Definition 10 (schwachbesetzter Blockbaum)

Falls es eine Konstante C_{sp} gibt, sodass

$$\#\{s \in T_I : (t^*, s) \in T_{I \times I}\} \leq C_{sp} \text{ und } \#\{t \in T_I : (t, s^*) \in T_{I \times I}\} \leq C_{sp}$$

für alle $t^*, s^* \in T_I$ gilt, heißt $T_{I \times I}$ C_{sp} -schwachbesetzt.

C_{sp} beschreibt also, mit wie vielen anderen Einträgen ein Eintrag aus dem Clusterbaum einen Knoten des Blockclusterbaums bilden kann.

Definition 11 (Baumtiefe, maximale Blattgröße)

Als Baumtiefe wird die Anzahl der Stufen eines Baumes bezeichnet.

Die maximale Blattgröße r wird definiert durch $r := \max\{\#\hat{t} : t \in \mathcal{L}_T\}$, wobei T ein bewerteter Baum sei.

Im Folgenden sei P_I die Baumtiefe eines Clusterbaums T_I , r bezeichne dessen maximale Blattgröße.

Lemma 10

Für einen C_{sp} -schwachbesetzten Blockclusterbaum $T_{I \times I}$ gilt:

$$\sum_{b \in \mathcal{L}_{T_{I \times I}}} \#\hat{t} \leq C_{sp}(P_I + 1)\#I.$$

Beweis

Sei $T_{I \times I}$ ein C_{sp} -schwachbesetzter Blockclusterbaum. Da jedes Indexpaar (t, s) in genau einem Blatt des Blockclusterbaums und jeder Index höchstens einmal auf jeder Stufe des zugehörigen Clusterbaums auftritt, gilt

$$\sum_{b \in \mathcal{L}_{T_{I \times I}}} \#\hat{t} = \sum_{t \in T_I} \sum_{\substack{s \in T_I \\ (t,s) \in \mathcal{L}_{T_{I \times I}}} \#\hat{t} \leq C_{sp} \sum_{t \in T_I} \#\hat{t} = C_{sp} \sum_{l=0}^{P_I} \sum_{\substack{t \in T_I \\ \text{level}(t)=l}} \#\hat{t} \leq C_{sp}(P_I + 1)\#I.$$

□

Unter der Annahme, dass die vorgestellten Algorithmen Clusterbäume T_I der Tiefe P_I sowie Blockclusterbäume, die C_{sp} -schwachbesetzt sind, liefern, lässt sich folgendes Lemma zeigen:

Lemma 11

Es sei $m + 1$ die Anzahl der Interpolationspunkte in eine Koordinatenrichtung und $k := (m + 1)^3$. Der Gesamtrechenaufwand zur approximativen Berechnung der Gravitationskraft lässt sich durch $2 \max\{r, k\} C_{sp}(P_I + 1)n$ beschränken.

Beweis

Nach Gleichung (4.5) lässt sich die Gravitationskraft auf ein Teilchen i darstellen durch $\frac{F_i}{m_i} = \gamma \sum_{\substack{(t,s) \in \mathcal{L}_{T_I \times I} \\ i \in \hat{t}}} \sum_{\substack{j \in \hat{s} \\ i \neq j}} m_j g(x_i, x_j)$. Um alle Kräfte zu berechnen, genügt es daher,

die Blätter $b = (t, s)$ des Blockclusterbaumes zu durchlaufen und für die Teilchen in \hat{t} die von den Teilchen in \hat{s} ausgeübte Kraft zu bestimmen. Die Summe aus diesen Teilkräften ergibt die Gesamtkraft.

In einem zulässigen Blatt ist der Rechenaufwand zur approximativen Berechnung, wie in Abschnitt (4.1) gezeigt, in der Größenordnung von $\mathcal{O}(k(\#\hat{t} + \#\hat{s}))$.

Für ein unzulässiges Blatt lässt sich der Rechenaufwand wie folgt beschränken $\#\hat{t} \cdot \#\hat{s} \leq r(\#\hat{t} + \#\hat{s})$, weil entweder t oder s schon ein Blatt von T_I ist.

Damit ist der Rechenaufwand für ein Blatt kleiner als $\max\{r, k\}(\#\hat{t} + \#\hat{s})$.

Der Gesamtaufwand für die approximative Berechnung der Kraft auf alle Teilchen lässt sich als wie folgt beschränken:

$$\begin{aligned} \sum_{b=(t,s) \in \mathcal{L}_{T_I \times I}} \max\{r, k\}(\#\hat{t} + \#\hat{s}) &= \max\{r, k\} \left(\sum_{b=(t,s) \in \mathcal{L}_{T_I \times I}} \#\hat{t} + \sum_{b=(t,s) \in \mathcal{L}_{T_I \times I}} \#\hat{s} \right) \\ &\leq 2 \max\{r, k\} C_{sp}(P_I + 1) \#I \\ &= 2 \max\{r, k\} C_{sp}(P_I + 1)n. \end{aligned}$$

□

Nun muss noch gezeigt werden, dass der Blockclusterbaum C_{sp} -schwachbesetzt ist. Außerdem wird eine Größenordnung für die Baumtiefe der Clusterbäume benötigt.

In der Supremumsnorm gilt für Quader $Q_t = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$, $Q_s = [c_1, d_1] \times [c_2, d_2] \times [c_3, d_3]$:

$$\begin{aligned} \text{diam}(Q_t) &= \max\{\|x - \tilde{x}\|_\infty : x, \tilde{x} \in Q_t\} = \max\{b_j - a_j : 1 \leq j \leq 3\} \text{ und} \\ \text{dist}(Q_t, Q_s) &= \min\{\|x - y\|_\infty : x \in Q_t, y \in Q_s\} \\ &= \max\{0, c_j - b_j, a_j - d_j : 1 \leq j \leq 3\}. \end{aligned}$$

Zur Vereinfachung der folgenden Betrachtungen wird ein Würfel mit der Kantenlänge H als Ausgangsquader gewählt. Dann sind alle Teilquader wieder Würfel. Da sich die Kantenlänge in jeder Richtung halbiert, ist für $t \in T$ und $t^* \in \text{sons}(t)$

$$\text{diam}(Q_{t^*}) = \frac{1}{2} \text{diam}(Q_t). \text{ Also ist } \text{diam}(Q_t) = \left(\frac{1}{2}\right)^{\text{level}(t)} H.$$

Es bezeichne h den minimalen Abstand zwischen den Teilchen, also $h := \min\{\|x_i - x_j\|_\infty : i, j \in \hat{t}\}$.

Lemma 12

Sei T_I ein zum Problem passender Clusterbaum. Seine Baumtiefe lässt sich durch $\lceil \log_2(\frac{H}{h}) \rceil$ beschränken.

Beweis

Für einen Quader Q_t auf Stufe $p := \lceil \log_2(\frac{H}{h}) \rceil$ gilt $\text{diam}(Q_t) = (\frac{1}{2})^p < h$. Damit kann Q_t höchstens ein Teilchen enthalten. Der Clusterbaum kann also auf Stufe p nur noch Blätter haben, also ist $p = \lceil \log_2(\frac{H}{h}) \rceil$ eine obere Schranke für die Baumtiefe. \square

Für Teilchen, die etwa gleichmäßig im Raum verteilt sind, ist $h \approx \frac{H}{\sqrt[3]{n}}$, weil alle drei Raumrichtungen gleichmäßig ausgenutzt werden. Damit ist $P_I \leq p \approx \frac{1}{3} \log_2 n$.

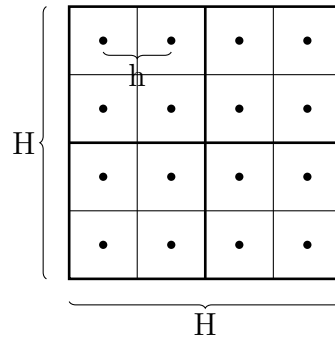


Abbildung 4.6: 16 gleichmäßig verteilte Teilchen. Quader zu den ersten drei Stufen des Clusterbaums im zweidimensionalen Fall. $h = \frac{H}{\sqrt{16}}$

Für die Schwachbesetztheit wird $\text{diam}(Q_t) \leq \text{dist}(Q_t, Q_s)$ für alle $(t, s) \in T_{I \times I}$ als Zulässigkeitsbedingung angenommen. Damit ist als Parameter nur $\eta \in (0, 1]$ anstatt $\eta \in (0, 4/3)$ erlaubt. Dann gilt:

Lemma 13

Ein so entstandener Blockclusterbaum $T_{I \times I}$ ist C_{sp} -schwachbesetzt. Die Schwachbesetztheitskonstante C_{sp} lässt sich durch $6^3 = 216$ beschränken.

Beweis

Sei $t \in T_I$, gesucht wird eine obere Schranke für die Anzahl der $s \in T_I$ mit $(t, s) \in T_{I \times I}$.

1. Fall: $t = \text{root}(T)$, damit gibt es nur ein $s \in T_I$ mit $(t, s) \in T_{I \times I}$, nämlich $s = \text{root}(T_I)$.

2. Fall: Es gibt t^* mit $t \in \text{sons}(t^*)$. In diesem Fall existiert für $(t, s) \in T_{I \times I}$ ein $s^* \in T_I$ mit $s \in \text{sons}(s^*)$.

Wenn die Anzahl der in Frage kommenden s^* abgeschätzt werden kann, kann auch die Anzahl der möglichen s abgeschätzt werden, weil jedes s^* höchstens acht Söhne hat. Der Vorteil ist, dass t^* und s^* die Zulässigkeitsbedingung nicht erfüllen, weil $(t, s) \in \text{sons}((t^*, s^*))$ liegt.

Für die zugehörigen Quader gilt also $\left(\frac{1}{2}\right)^{\text{level}(Q_{t^*})} = \text{diam}(Q_{t^*}) > \text{dist}(Q_{t^*}, Q_{s^*})$. Aufgrund der Konstruktion ist $\text{level}(t^*) = \text{level}(s^*) =: l$ und die Würfel sind von der Form

$$\prod_{j=1}^3 [a_j + i_j \left(\frac{1}{2}\right)^l H, a_j + (i_j + 1) \left(\frac{1}{2}\right)^l H] \text{ mit } i_1, i_2, i_3 \in \{0, \dots, 2^l - 1\}.$$

Also ist

$$\begin{aligned} \text{dist}(Q_{t^*}, Q_{s^*}) &= \max_{1 \leq k \leq 3} \left\{ \left| a_k + (i_k + 1) \left(\frac{1}{2}\right)^l H - \left(a_k + j_k \left(\frac{1}{2}\right)^l H \right) \right| : i_k, j_k \in \{0, \dots, 2^l - 1\} \right\} \\ &= \max\{0, |i + 1 - j| \left(\frac{1}{2}\right)^l H : i, j \in \{0, \dots, 2^l - 1\}\}. \end{aligned}$$

Damit sind nur Würfel mit $\text{dist}(Q_{t^*}, Q_{s^*}) = 0$, also direkt benachbarte oder gleiche, unzulässig. Für Q_{t^*} gibt es $3^3 = 27$ solcher Würfel, also gilt

$$\#\{s : (t, s) \in T_{I \times I}\} \leq 2^3 \cdot 3^3 = 6^3.$$

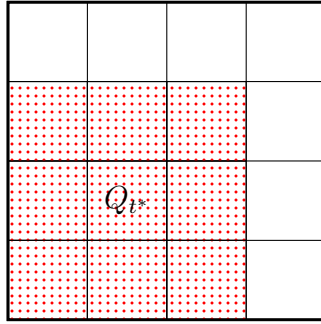


Abbildung 4.7: Die rotgepunkteten 3^2 Quader bilden im zweidimensionalen Fall mit dem mittleren Quader Q_{t^*} unzulässige Knoten des Blockclusterbaumes.

Aufgrund der Symmetrie können dieselben Überlegungen für feste s vorgenommen werden, damit lässt sich C_{sp} durch 6^3 beschränken. □

Beim Aufstellen der Bäume fallen Gleitkommaoperationen nur bei der Berechnung der Intervallmittelpunkte an, sonst handelt es sich um Umsortierungen, Abfragen und rekursive Aufrufe. Intervallmittelpunkte müssen für jeden Knoten des Baumes mit Ausnahme der Blätter bestimmt werden. Die Anzahl der Knoten des Baumes ist beschränkt durch $\#T_I \leq P_I \#\mathcal{L}_{\mathcal{T}_I}$. Da es höchstens n Blätter gibt, werden weniger als $\mathcal{O}(n \log_2 n)$ Gleitkommaoperationen für die Erstellung der Bäume benötigt.

Also ist der Gesamtaufwand für die Auswertung der Kraftfunktion durch $6^3 \max\{r, k\}(\log_2 n + 1)n$ beschränkt. Er liegt damit in der Größenordnung von $\mathcal{O}(n \log_2 n)$. Dies ist für große Teilchenzahlen deutlich günstiger, als die direkte Berechnung mit einem Aufwand von $\mathcal{O}(n^2)$.

Gesamtfehler

Die Gravitationskraft für das Teilchensystem wird stückweise über die Blätter des Blockclusterbaums berechnet. In den Blättern wird die Berechnung der Teilkraft entweder exakt berechnet oder durch Interpolation mit einem exponentiell zum Wachstum der verwendeten Stützstellen sinkenden Fehler approximiert.

Seien $F, \tilde{F} \in \mathbb{R}^{3 \times n}$ die exakte bzw. die durch Verwendung dieses Verfahrens erhaltene Gravitationskraft. Der (d, i) -ten Eintrag der Matrix entspricht der Kraft auf ein Teilchen i in einer Dimension d . Für den Summanden von $F_{i,d}$, der durch die Wechselwirkung mit einem Teilchen j entsteht, lässt sich daher ein $\varepsilon(m)$ finden, für das gilt $|F_{d,i}^j - \tilde{F}_{d,i}^j| \leq \varepsilon(m)$. Dieses $\varepsilon(m)$ hängt von der Stützstellenzahl der Interpolation $m + 1$ ab und konvergiert für $m \rightarrow \infty$ mit exponentieller Geschwindigkeit gegen Null.

Der Fehler zwischen exakter und durch Approximation erhaltener Gravitationskraft in einer Dimension lässt sich dann in der euklidischen Norm wie folgt abschätzen:

$$\begin{aligned} \|F_d - \tilde{F}_d\|_2^2 &= \left| \sum_{i=1}^n \sum_{\substack{j=1 \\ i \neq j}}^n (F_{d,i}^j - \tilde{F}_{d,i}^j)^2 \right| = \left| \sum_{\substack{b \in \mathcal{L}_{I \times I} \\ b=(t,s)}} \sum_{i \in \hat{t}} \sum_{j \in \hat{s}} (F_{d,i}^j - \tilde{F}_{d,i}^j)^2 \right| \\ &\leq \sum_{\substack{b \in \mathcal{L}_{I \times I} \\ b=(t,s)}} \sum_{i \in \hat{t}} \sum_{j \in \hat{s}} |F_{d,i}^j - \tilde{F}_{d,i}^j|^2 \leq \sum_{\substack{b \in \mathcal{L}_{I \times I} \\ b=(t,s)}} \sum_{i \in \hat{t}} \sum_{j \in \hat{s}} \varepsilon(m)^2 = \varepsilon(m)^2 \sum_{\substack{b \in \mathcal{L}_{I \times I} \\ b=(t,s)}} \#\hat{t} \#\hat{s} \\ &= \varepsilon(m)^2 \# \left(\bigcup_{\substack{b \in \mathcal{L}_{I \times I} \\ b=(t,s)}} \hat{t} \times \hat{s} \right) = \varepsilon(m)^2 \# \bigcup_{b \in \mathcal{L}_{I \times I}} \hat{b} = \varepsilon(m)^2 \#(I \times I) = \varepsilon(m)^2 n^2. \end{aligned}$$

Damit folgt für den Gesamtfehler gemessen in der Frobeniusnorm:

$$\|F - \tilde{F}\|_F^2 = \|F_1 - \tilde{F}_1\|_2^2 + \|F_2 - \tilde{F}_2\|_2^2 + \|F_3 - \tilde{F}_3\|_2^2 \leq 3\varepsilon(m)^2 n^2. \quad (4.6)$$

Also überträgt sich die exponentielle Konvergenz in m auf das gewählte Verfahren. Außerdem sinkt die Genauigkeit mit zunehmender Teilchenzahl nur linear. Dadurch kann für größere Problemdimensionen in etwa die gleiche Genauigkeit erzielt werden wie für kleinere, ohne die Anzahl der Stützstellen erhöhen zu müssen.

5 Implementierung

Im Rahmen dieser Bachelorarbeit wird eine Funktion zur Berechnung der Gravitationskraft nach dem in Kapitel 4 vorgestellten Verfahren implementiert und anschließend auf Rechenaufwand und Konvergenz getestet.

5.1 Programmaufbau

Das Programm besteht aus zwei Hauptteilen; dem Erstellen von Cluster- und Blockclusterbaum auf der einen, sowie der approximativen und direkten Berechnung der Gravitationskraft auf der anderen Seite.

Zunächst zu den verwendeten Bezeichnungen: Die Datenstruktur *par* enthält Informationen über die Teilchen, also deren Massen und Ortsvektoren, und die Variable *F* die Werte der Gravitationskraft. *t*, *s* bezeichnen Clusterbäume, \hat{t} , \hat{s} die dazugehörigen Indexmengen. Die Grenzen der Quader werden durch *a* und *c* beschrieben, *r* ist die maximale Blattgröße, *b* bezeichnet einen Blockclusterbaum und η ist die Variable der Zulässigkeitsbedingung. $(m + 1)^3$ ist die Anzahl der Stützstellen der Tensorinterpolation und *d* steht für eine Raumrichtung.

Zum Aufstellen der Baumstrukturen werden Routinen aus der H2Lib¹ verwendet.

Zur Erstellung des Clusterbaums wird der auf Seite 24 erläuterte Algorithmus angewandt mit dem Unterschied, dass die entstehenden Teilquader in jedem Schritt minimiert werden. Außerdem geht man bereits jeweils nach der Teilung in eine Koordinatenrichtung, anstatt nach der Teilung in alle Richtungen, in die nächste Stufe des Clusterbaums über.

¹Programmbibliothek für die Programmierung mit \mathcal{H}^2 -Matrizen. Die vorgestellte Verfahrensweise bietet einen Einstieg in das thematische Umfeld dieser Klasse von Matrizen.

```

procedure build_cluster(par, r, d, t, a, c, var t1, t2)
for i ∈ {1, 2, 3}
    ai ← min{xi,d : i ∈  $\hat{t}$ }
    ci ← max{xi,d : i ∈  $\hat{t}$ }
end for
if # $\hat{t}$  ≤ r
    sons(t) ← ∅
else
    m ←  $\frac{a_d+c_d}{2}$ 
    for i ∈  $\hat{t}$ 
        if xi,d ≤ m
             $\hat{t}_1$  ←  $\hat{t}_1 \cup \{i\}$ 
        else
             $\hat{t}_2$  ←  $\hat{t}_2 \cup \{i\}$ 
        end if
    end for
    if  $\hat{t}_1 \neq \emptyset$  und  $\hat{t}_2 \neq \emptyset$ 
        sons(t) ← {t1, t2}
        build_cluster(par, r, (d mod 3) + 1, t1, a, c)
        build_cluster(par, r, (d mod 3) + 1, t2, a, c)
    else if  $\hat{t}_1 = \emptyset$ 
        sons(t) ← {t2}
        build_cluster(par, (d mod 3) + 1, t2, a, c)
    else if  $\hat{t}_2 = \emptyset$ 
        sons(t) ← {t1}
        build_cluster(par, (d mod 3) + 1, t1, a, c)
    end if
end if

```

Abbildung 5.1: Erstellung eines Clusterbaums

Die Funktion zur Erstellung des Blockclusterbaums arbeitet nach dem auf Seite 26 vorgestellten Schema, wobei hier für die Zulässigkeitsbedingung der euklidische Abstand benutzt wurde.

```

procedure build_block( $t, s, \eta, b$ )
 $b \leftarrow (t, s)$ 
if  $\text{diam}(Q_t) \leq \eta \text{ dist}(Q_t, Q_s)$ 
   $\text{sons}(b) \leftarrow \emptyset$ 
else
  if  $\text{sons}(t) = \emptyset$  oder  $\text{sons}(s) = \emptyset$ 
     $\text{sons}(b) \leftarrow \emptyset$ 
  else
    for  $t^* \in \text{sons}(t)$ 
      for  $s^* \in \text{sons}(s)$ 
         $\text{sons}(b) \leftarrow \text{sons}(b) \cup b^* = (t^*, s^*)$ 
        build_block( $t^*, s^*, \eta, b^*$ )
      end for
    end for
  end if
end if

```

Abbildung 5.2: Erstellung eines Blockclusterbaums

Nun zur Ermittlung der Gravitationskraft. Für alle Blätter werden die zugehörigen Teilkräfte, deren Summe die Gesamtkraft ergibt, berechnet.

Bei unzulässigen Blättern $b = (t, s)$ — das bedeutet, dass die zugehörigen Cluster gleich sind oder dicht beieinander liegen — muss für jedes Teilchen aus Q_t die Kraft, die jedes Teilchen aus Q_s auf dieses auswirkt, berechnet und zur Gesamtkraft addiert werden. Daraus entsteht folgender Algorithmus:

```

procedure Kraft_direkt( $b = (t, s), F, par$ )
for  $i \in \hat{t}$ 
  for  $j \in \hat{s}$ 
    if  $i \neq j$ 
       $F_i \leftarrow F_i + m_j \gamma \frac{x_j - x_i}{\|x_j - x_i\|_2^3}$ 
    end if
  end for
end for

```

Abbildung 5.3: Berechnung der Gravitationskraft für ein unzulässiges Blatt

Für zulässige Blätter $b = (t, s)$ wird der in Abschnitt 4.3 beschriebene Interpolationsansatz angewendet, weil Q_t und Q_s so weit auseinander liegen, dass die Gravitationskraft durch dieses Vorgehen gut approximiert werden kann.

Es werden dazu zunächst in Vorbereitungsschritten die $m+1$ Tschebyscheffpunkte für die jeweiligen Koordinatenrichtungen und für alle $\nu \in M$ die Hilfsgrößen h_ν berechnet. Dann werden die Teilkräfte für die Partikel aus \hat{t} approximiert. Hierzu

werden die Produkte aus dem in x_i ausgewerteten Lagrangepolynom $\mathcal{L}_\nu(x_i)$ und h_ν summiert.

```

procedure Kraft_approx( $b = (t, s), m, par, F, \mathbf{var} h, \xi, \mathcal{L}$ )
for  $\nu \in M$ 
  for  $i \in \{1, 2, 3\}$ 
     $\xi_{\nu_i} \leftarrow \frac{c_i + a_i}{2} + \frac{c_i - a_i}{2} \cos\left(\pi \frac{2i+1}{2+2m}\right)$ 
  end for
end for
for  $i \in \hat{s}$ 
  for  $\nu \in M$ 
     $h_\nu \leftarrow h_\nu + m_i \frac{x_j - \xi_\nu}{\|x_i - \xi_\nu\|_2^3}$ 
  end for
end for
for  $i \in \hat{t}$ 
  for  $\nu \in M$ 
     $\mathcal{L} \leftarrow 1$ 
    for  $\mu \in M$ 
      if  $\mu \neq \nu$ 
         $\mathcal{L} \leftarrow \mathcal{L} \frac{x_{i,1} - \xi_{\mu_1}}{\xi_{\nu_1} - \xi_{\mu_1}} \frac{x_{i,2} - \xi_{\mu_2}}{\xi_{\nu_2} - \xi_{\mu_2}} \frac{x_{i,3} - \xi_{\mu_3}}{\xi_{\nu_3} - \xi_{\mu_3}}$ 
      end if
    end for
     $F_i \leftarrow F_i + \mathcal{L} h_\nu$ 
  end for
   $F_i \leftarrow \gamma F_i$ 
end for

```

Abbildung 5.4: Berechnung der Gravitationskraft für ein zulässiges Blatt

Man erhält damit folgenden Algorithmus für die effiziente Berechnung der Gravitationskraft eines Teilchensystems:

```

procedure Kraftauswertung( $m, \eta, par, F, r, a, c, \mathbf{var} t, b$ )
 $F \leftarrow 0$ 
 $t \leftarrow \text{build\_cluster}(par, r, 1, t, a, c)$ 
 $b \leftarrow \text{build\_block}(t, t, \eta, b)$ 
for( $b \in \mathcal{L}_{I \times I}$ )
    if  $b$  zulässig
        Kraft_approx( $b, m, par, F$ )
    else
        Kraft_direkt( $b, F, par$ )
    end if
end for

```

Abbildung 5.5: Berechnung der Gravitationskraft

5.2 Testergebnisse

Um die Effizienz des Algorithmus mit Hilfe der theoretischen Resultate zu überprüfen, werden numerische Experimente durchgeführt. Zum einen muss die Genauigkeit des Programms getestet werden. Für wachsende Stützstellenzahl wird aufgrund von Abschätzung (4.6) exponentielle Konvergenz des Gesamtfehlers gegen 0 erwartet. Weiterhin sollte, um gleichbleibende Genauigkeit bei einer größer werdenden Anzahl von Teilchen zu erhalten, die Anzahl der Stützstellen nicht steigen müssen.

Zum anderen wird der Rechenaufwand beschränkt durch $6^3 \max\{r, k\}n \log(n)$. Deshalb wird für die Anzahl der Teilchen ein Aufwand von $\mathcal{O}(n \log(n))$ anstelle von $\mathcal{O}(n^2)$ erwartet. Außerdem sollte polynomieller Aufwand in Abhängigkeit von der Stützstellenzahl zu beobachten sein.

Genauigkeit

Um die exponentielle Konvergenz zu testen, wird für eine feste Anzahl von Teilchen n und variable Stützstellenanzahl $(m + 1)^3$ sowohl eine exakte als auch eine approximative Berechnung durchgeführt. Der entstehende Fehler wird in der Frobeniusnorm gemessen.

Dabei wurden folgende Ergebnisse erzielt:

Stützstellenzahl	relativer Fehler	
	Teilchenzahl	
	50.000	70.000
4^3	$2,634291 \cdot 10^{-5}$	$2,670027 \cdot 10^{-5}$
5^3	$2,826359 \cdot 10^{-6}$	$2,836424 \cdot 10^{-6}$
6^3	$2,318399 \cdot 10^{-7}$	$2,319399 \cdot 10^{-7}$
7^3	$2,860541 \cdot 10^{-8}$	$2,845077 \cdot 10^{-8}$
8^3	$2,691689 \cdot 10^{-9}$	$2,692706 \cdot 10^{-9}$

Tabelle 5.1: Relativer Fehler für wachsende Stützstellenzahl bei $r = 250$, $\eta = 1$

Sowohl bei 50.000 als auch bei 70.000 Teilchen verringert sich der relative Fehler um eine Zehnerpotenz. Dies entspricht der erwarteten exponentiellen Konvergenz.

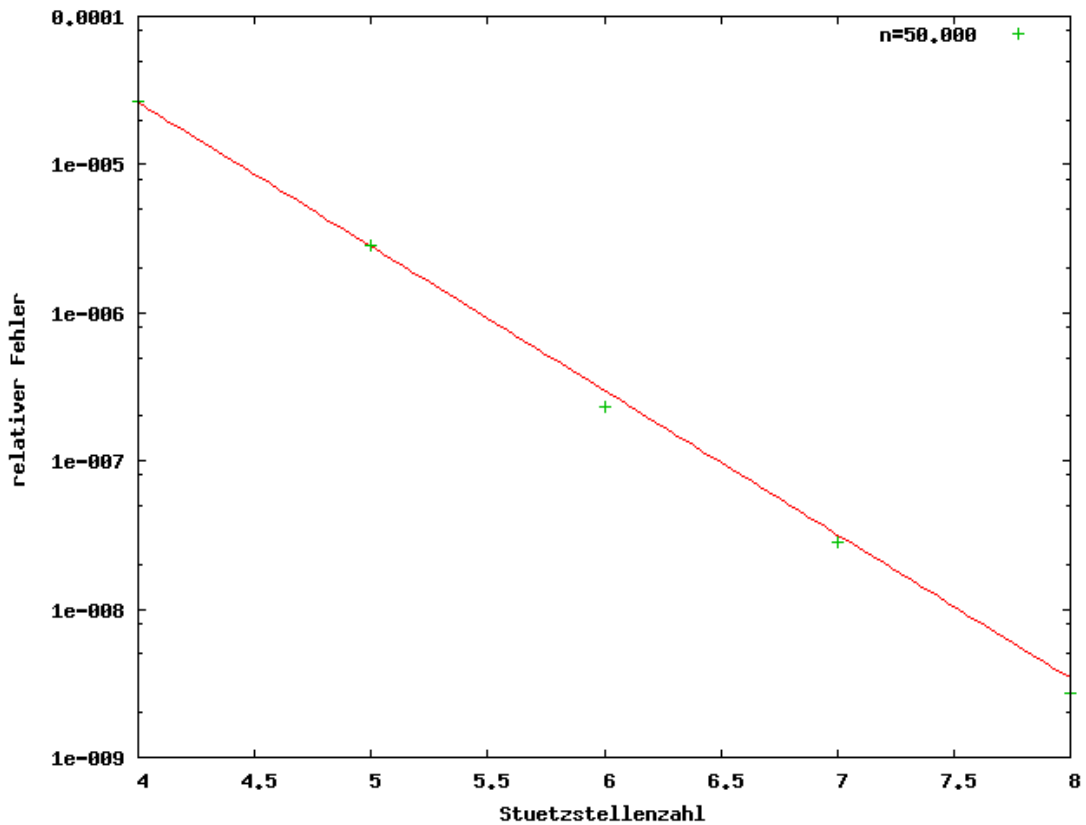


Abbildung 5.6: Relativer Fehler für wachsende Stützstellenzahl bei $n = 50.000$, $r = 250$, $\eta = 1$

Auch in Abbildung 5.6 kann man die exponentielle Konvergenz gut erkennen. Bei logarithmischer Einteilung der Ordinate liegen die Werte für $n = 50.000$ annähernd

auf einer Geraden mit negativer Steigung. Die Genauigkeit steigt also exponentiell zur Stützstellenzahl.

Desweiteren fällt auf, dass bei derselben Stützstellenzahl die Fehler bei 50.000 und 70.000 Teilchen annähernd gleich sind. Dieses Verhalten wurde weiter untersucht, es ergaben sich die folgenden Werte:

Teilchenzahl	relativer Fehler	
	Stützstellenzahl (max. Blattgröße)	
	$4^2(128)$	$5^2(250)$
10.000	$1.700421 \cdot 10^{-5}$	$5.162795 \cdot 10^{-7}$
20.000	$2.376422 \cdot 10^{-5}$	$1.043420 \cdot 10^{-6}$
40.000	$2.680013 \cdot 10^{-5}$	$2.774428 \cdot 10^{-6}$
80.000	$2.815869 \cdot 10^{-5}$	$2.613455 \cdot 10^{-6}$
100.000	$2.680312 \cdot 10^{-5}$	$2.918913 \cdot 10^{-6}$

Tabelle 5.2: Relativer Fehler für wachsende Teilchenzahl bei $\eta = 1$

Die Werte bleiben alle in etwa derselben Größenordnung. Von jeweils einer Ausnahme abgesehen, ist eine leichte Vergrößerung des Fehlers bei Erhöhung der Teilchenzahl festzustellen. Diese liegt deutlich unterhalb eines linearen Wachstums in n , was nach der Abschätzung in (4.6) eine obere Schranke für das Wachstum des Fehlers ist.

Geschwindigkeit

Um die Aufwandsreduzierung nachzuweisen, wird die Zeit zur Auswertung der Kraftfunktion sowohl direkt als auch approximativ für verschiedene Teilchenzahlen gemessen.

Teilchenzahl	Zeit in s			
	direkt	approximiert ($m(r)$)		
		3 (128)	4 (250)	6 (432)
10.000	26,85	21,19	25,49	27,03
20.000	105,04	67,93	83,66	112,69
40.000	420,78	201,32	267,21	378,28
80.000	1.685,91	521,29	782,47	1.246,80
100.000	2.633,01	706,30	1.059,46	1.762,17

Tabelle 5.3: Rechenaufwand für verschiedene Teilchenzahlen bei $\eta = 1$

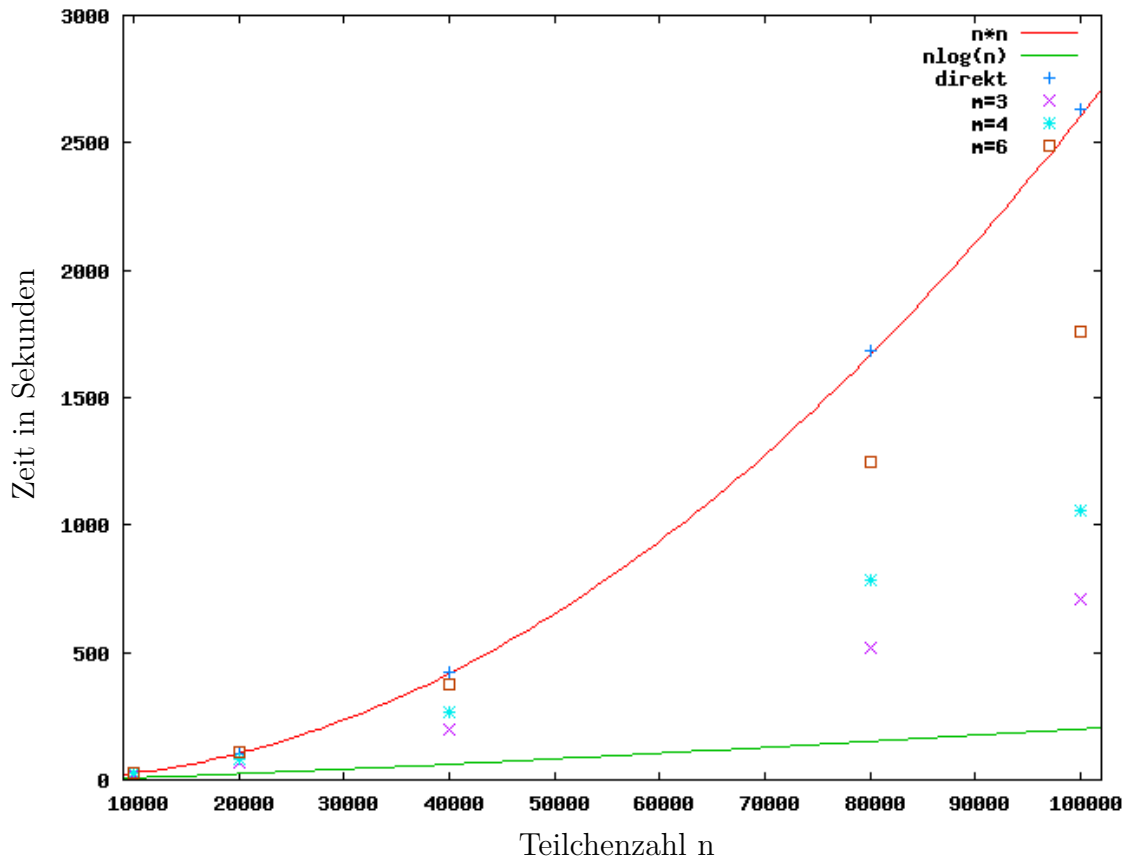


Abbildung 5.7: Rechenaufwand für verschiedene Teilchenzahlen bei $\eta = 1$

Die Messwerte für die direkte Berechnung liegen alle annähernd auf der n^2 -Kurve. Daran erkennt man gut den Rechenaufwand von $\mathcal{O}(n^2)$ für die direkte Berechnung. Die Daten für die approximative Berechnung der Gravitationskraft liegen alle unterhalb dieser Kurve. Hieraus wird deutlich, dass bei dieser Methode mit wachsender Teilchenzahl der Rechenaufwand geringer ist als beim direkten Vorgehen. Bei $m = 3, 4$ ist zu erkennen, dass das Wachstumsverhalten des Rechenaufwands dem von $n \log(n)$ ähnlicher ist als dem von n^2 . Auch bei $m = 6$ deutet sich dieses Verhalten bei 100.000 Teilchen an. Dies zeigt, dass der vorgestellte Ansatz, um eine gewisse Genauigkeit sicherzustellen, erst bei einer großen Teilchenzahl sinnvoll einzusetzen ist. Der Grund hierfür ist, dass das Aufstellen und Auswerten der Baumstrukturen mit Verwaltungsaufwand verbunden ist. Dieser wird erst ab einer gewissen Problemdimension durch die Rechenersparnis aufgewogen.

Desweiteren wird das Verhalten des Rechenaufwands für wachsende Stützstellenanzahl, also für größer werdende Genauigkeit, betrachtet:

Stützstellenzahl	Zeit in Sekunden	
	Teilchenzahl	
	50.000	70.000
4^3	308,29	465,61
5^3	381,42	656,91
6^3	487,70	942,33
7^3	635,60	1.337,59
8^3	831,91	1.862,26

Tabelle 5.4: Rechenaufwand für wachsende Stützstellenzahl bei $r = 250$, $\eta = 1$

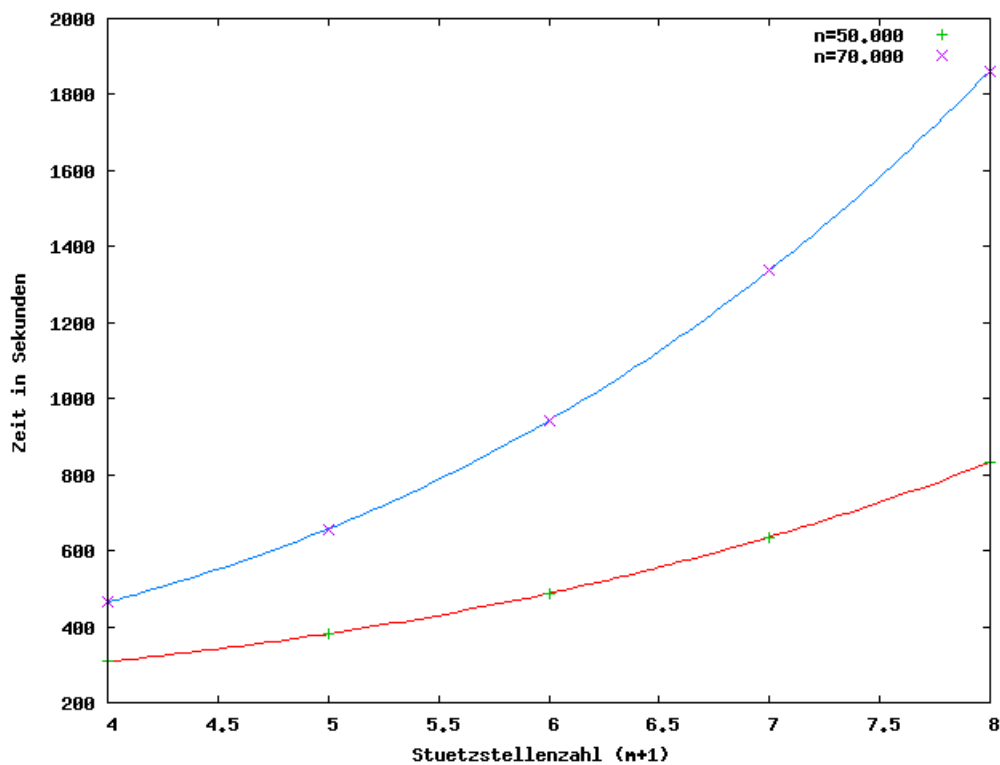


Abbildung 5.8: Rechenaufwand für verschiedene Anzahl von Stützstellen

Der Verlauf der Messwerte lässt sich in beiden betrachteten Fällen gut durch eine Polynomfunktion dritten Grades annähern. Damit erhöht sich der Rechenaufwand nur polynomiell mit der Stützstellenzahl.

Eine hohe Genauigkeit ist bei vertretbare Erhöhung des Rechenaufwandes sicher-

gestellt, da schon bei verglichen mit der Teilchenzahl niedriger Stützstellenzahl ein Fehler im Bereich der Maschinengenauigkeit erwartet werden kann.

6 Fazit

In der vorliegenden Bachelorarbeit wird gezeigt, dass bei der numerischen Simulation von Partikelsystemen der effizienten Auswertung der Gravitationskraft im System eine entscheidende Bedeutung zukommt.

Es wird eine Methode zur approximativen Bestimmung dieser Kraft vorgestellt, die darauf beruht die betrachtete Indexmenge zu partitionieren. Dadurch wird auch das betrachtete Gebiet unterteilt und die Gravitationskraftsfunktion kann zwischen geeigneten Bereichen interpoliert werden. Für große Teilchenzahlen kann dadurch der Rechenaufwand von $\mathcal{O}(n^2)$ auf $\mathcal{O}(n \log(n))$ gesenkt werden, wobei trotzdem eine gute Genauigkeit gewährleistet bleibt.

Das vorgestellte Verfahren wurde implementiert, die theoretischen Resultate können auch in Experimenten nachgewiesen werden.

Literaturverzeichnis

- [1] Börm, Steffen: *Einführung in die Numerik*. (Vorlesungsskript), 2009.
- [2] Börm, Steffen: *Numerik gewöhnlicher Differentialgleichungen*. (Vorlesungsskript), 2010.
- [3] Börm, Steffen: *Numerik nicht-lokaler Operatoren*. (Vorlesungsskript), 2010.
- [4] Bronstein, I.N., Semendjajew, K.A. *et al.*: *Taschenbuch der Mathematik*. Harri Deutsch, 2008.
- [5] Grehn, Joachim und Krause, Joachim: *Metzler Physik*. Schroedel, 2006.
- [6] Hackbusch, Wolfgang: *Hierarchische Matrizen. Algorithmen und Analysis*. Springer, 2009.
- [7] Schütt, Carsten: *Gewöhnliche Differentialgleichungen*. (Vorlesungsskript), 2009.
- [8] Tipler, Paul A. und Mosca, Gene: *Physik für Wissenschaftler und Ingenieure*. Spektrum Akademischer Verlag, 2009.

Danksagung

An dieser Stelle möchte ich mich bei mehreren Personen bedanken, die mich bei der Erstellung der vorliegenden Bachelorarbeit unterstützt haben. Mein Dank gilt:

- Prof. Dr. Steffen Börm für die Vergabe des Themas und dafür, dass er jederzeit bereit war, Fragen zu beantworten,
- Dipl.-Math. Knut Reimer für die Erklärung der H2Lib-Routinen, seine Unterstützung bei der Implementierung und die intensive Betreuung,
- den übrigen Mitgliedern der Arbeitsgruppe Scientific Computing für ihre Unterstützung,
- Arne Johanson für viele Diskussionen, Denkanstöße und mehrfaches Korrekturlesen,
- Katja Hauschildt und Hans-Wilhelm Franck für mehrfaches Korrekturlesen,
- Markus Pfeil für \LaTeX - und C-Programmiertipps.

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Weiterhin versichere ich, dass diese Arbeit noch nicht als Abschlussarbeit an anderer Stelle vorgelegen hat.

18. August 2010

Margaretha Franck