

Florian Himmelbauer, BSc

**Analyse und Implementierung einer
adaptiven schnellen Multipolmethode
für die Potentialrechnung**

MASTERARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

Masterstudium Technomathematik

eingereicht an der

Technischen Universität Graz

Betreuer

Assoc. Prof. Dr. G. Of

Institut für Numerische Mathematik

Graz, September 2017

Vorwort

Diese Arbeit entstand im Rahmen meines Masterstudiums Technomathematik an der TU Graz.

Mein Dank gilt Herrn Dr. Günther Of für das Bereitstellen dieses interessanten Themas und die hervorragende Betreuung und Unterstützung bei der Entstehung dieser Arbeit.

Mein besonderer Dank gilt meiner Familie, insbesondere meinen Eltern, die mir mein Studium ermöglicht und mich in all meinen Entscheidungen unterstützt haben.

Inhaltsverzeichnis

Einleitung	7
1 Kernentwicklung und Multipoloperatoren	13
1.1 Entwicklung des Kerns	13
1.2 Multipolentwicklung und lokale Entwicklung	17
1.2.1 Multipolentwicklung	17
1.2.2 Lokale Entwicklung	19
1.3 Verschiebung und Umrechnung der Entwicklungen	21
2 Der Multipolalgorithmus	27
2.1 Der Clusterbaum	28
2.2 Zulässigkeitskriterien	31
2.3 Aufstellen der Operationslisten	37
2.4 Darstellung der Operationen als Matrix	41
2.5 Der Multipolalgorithmus	43
2.5.1 Der aufsteigende Pfad	43
2.5.2 Der absteigende Pfad	44
3 Numerische Beispiele	47
3.1 Testrechnungen für zufällig erzeugte Punktmengen	47
3.2 Testrechnungen für die Integrationspunkte adaptiver BEM-Netze	52
3.3 Testrechnung für eine gleichmäßige Ladungsverteilung auf einer Würfel- oberfläche	55
3.4 Beschreibung der Implementierung	58
4 Zusammenfassung und Ausblick	61
Literaturverzeichnis	63

Einleitung

Für die Lösung von Randwertproblemen existieren verschiedenste numerische Verfahren. Zwei der gebräuchlichsten Methoden sind die Finite Elemente Methode (FEM) und die Randelementmethode (BEM). Für eine ausführliche Beschreibung dieser Methoden siehe z.B. [22, 23, 24].

Während bei der FEM das gesamte Gebiet diskretisiert wird, erfolgt bei der BEM nur eine Diskretisierung des Randes. Da in der BEM nur der Rand diskretisiert wird, ist die Anzahl der Freiheitsgrade geringer als in der FEM, in welcher eine Diskretisierung des gesamten Gebietes erfolgt, und somit folgt auch eine geringere Dimension der Matrix des resultierenden Gleichungssystems.

Ein Nachteil der Standard BEM ist allerdings, dass die Matrix dieses Gleichungssystems vollbesetzt ist. Aufgrund dessen folgt für das Erstellen und Speichern der Matrix sowie für die Matrixvektormultiplikation ein quadratischer Aufwand. Der quadratische Aufwand ist eine Einschränkung der Standard BEM an die Anzahl der Freiheitsgrade. Mit sogenannten schnellen Methoden lässt sich der Aufwand der BEM auf nahezu lineare Komplexität reduzieren.

In dieser Arbeit wird die schnelle Multipolmethode [8, 9] untersucht. Weitere Methoden sind z.B. Panel Clustering [13], Adaptive Cross Approximation (ACA) [1], Wavelets [2, 5, 6], hierarchische Matrizen [10, 12] und \mathcal{H}^2 -Matrizen [11].

Mit der schnellen Multipolmethode lässt sich insbesondere das Potential innerhalb einer Ladungsverteilung effizient berechnen. Sie findet z.B. auch Anwendung bei der Partikelsimulation physikalischer Systeme [9].

Dabei wird in dieser Arbeit speziell eine adaptive Version der schnellen Multipolmethode [3, 4, 16] behandelt. Diese berücksichtigt, durch das Einführen von zwei zusätzlichen Operationen im Multipolalgorithmus, die Adaptivität der Verteilungen. Solche adaptiven Zerlegungen entstehen z.B. bei Rechnungen aus der adaptiven BEM, siehe z.B. [7, 15]. Wie die numerischen Experimente aus Kapitel 3 zeigen, folgt durch die zusätzlichen Operationen eine weitere Ersparnis an Rechenzeit und Speicheraufwand, insbesondere bei adaptiven Verteilungen, im Vergleich zur Standardversion der schnellen Multipolmethode.

In der schnellen Multipolmethode für die BEM wird ausgenutzt, dass bei der Lösung des Gleichungssystems mit numerischen Methoden, wie dem Konjugierten Gradientenverfahren [25], die Matrix nie aufgestellt werden muss, sondern nur die Realisierung einer Matrixvektormultiplikation benötigt wird. Wie dies auf die BEM Anwendung findet, wird im Folgenden anhand der Lösung eines Randwertproblems mit Hilfe des Einfachschichtpotentialansatz skizziert. Als Testbeispiel wurde die Laplace-Gleichung

im \mathbb{R}^3 gewählt. Für eine detaillierte Darstellung sowie eine Ausführung der anderen Randintegraloperatoren siehe [17, 18].

Der Einfachschichtpotentialansatz

$$(Vt)(x) = \int_{\Gamma} U^*(x, y)t(y)ds_y \quad \text{für } x \in \Gamma$$

führt mit dem Galerkin-Verfahren und einer passenden Diskretisierung auf das Lösen eines linearen Gleichungssystems

$$V_h \underline{t} = \underline{w}.$$

Dabei hat die Matrix V_h mit der Fundamentallösung

$$U^*(x, y) = \frac{1}{4\pi|x - y|}$$

des Laplace-Operators folgende Gestalt:

$$V_h[\ell, k] = \frac{1}{4\pi} \int_{\tau_\ell} \int_{\tau_k} \frac{1}{|x - y|} ds_y ds_x, \quad k, \ell = 1, \dots, N.$$

Diese Matrix ist vollbesetzt. Der Speicheraufwand und der Aufwand der Matrixvektormultiplikation der Matrix V_h beträgt N^2 . Dieser Aufwand wächst somit quadratisch mit der Anzahl der Freiheitsgrade. Für die zeilenweise Matrixvektormultiplikation von $\underline{w} = V_h \underline{t}$ folgt

$$w_\ell = \sum_{k=1}^N V_h[\ell, k]t_k = \sum_{k=1}^N \frac{t_k}{4\pi} \int_{\tau_\ell} \int_{\tau_k} \frac{1}{|x - y|} ds_y ds_x \quad \ell = 1, \dots, N.$$

Wie im weiteren Verlauf der Arbeit gezeigt wird, erfolgt in der Multipolmethode eine Unterscheidung in Nahfeld (NF(ℓ)) und Fernfeld (FF(ℓ)). Die Einträge im Nahfeld müssen direkt berechnet werden, die Beiträge des Fernfelds werden mittels des Multipolalgorithmus berechnet. Grundlage dafür ist eine Reihendarstellung des Kerns

$$U^*(x, y) = \sum_{n=0}^{\infty} \sum_{m=-n}^n \overline{S_n^m}(x) R_n^m(y)$$

mit getrennten Variablen. Die Herleitung dieser Darstellung sowie eine Fehlerabschätzung für die abgeschnittene Entwicklung ist in Kapitel 1 ausgeführt. Anhand der Fehlerabschätzung wird auch die Notwendigkeit der Aufspaltung in Nah- und Fernfeld gezeigt. Mit einer numerischen Quadraturformel

$$\int_{\tau_k} f(x) \approx \sum_{i=1}^G \Delta_k \omega_{k,i} f(x_{k,i})$$

in den Elementen τ_k , mit Integrationsgewichten $\omega_{k,i}$ in den Auswertungspunkten $x_{k,i}$ und der Fläche Δ_k des Elements, folgt mit der Aufspaltung in Nah- und Fernfeld für die zeilenweise Matrixvektormultiplikation

$$\begin{aligned}
w_\ell &= \sum_{k=1}^N V_h[\ell, k] t_k \\
&= \sum_{k \in NF(\ell)} V_h[\ell, k] t_k + \sum_{k \in FF(\ell)} V_h[\ell, k] t_k \\
&= \sum_{k \in NF(\ell)} V_h[\ell, k] t_k + \sum_{k \in FF(\ell)} \frac{t_k}{4\pi} \int_{\tau_\ell} \int_{\tau_k} \frac{1}{|x-y|} ds_y ds_x \\
&\approx \sum_{k \in NF(\ell)} V_h[\ell, k] t_k + \sum_{k \in FF(\ell)} \frac{t_k}{4\pi} \sum_{j=1}^G \Delta_\ell \omega_{\ell,j} \int_{\tau_k} \frac{1}{|x_{\ell,j} - y|} ds_y \\
&\approx \sum_{k \in NF(\ell)} V_h[\ell, k] t_k + \frac{1}{4\pi} \sum_{j=1}^G \Delta_\ell \omega_{\ell,j} \sum_{k \in FF(\ell)} \sum_{i=1}^G t_k \Delta_k \omega_{k,i} \frac{1}{|x_{\ell,j} - x_{k,i}|}.
\end{aligned}$$

Mit der Reihendarstellung des Kerns $U^*(x, y)$ und einer Umsortierung der Summen folgt für die zeilenweise Matrixvektormultiplikation weiter:

$$\begin{aligned}
&\sum_{k \in NF(\ell)} V_h[\ell, k] t_k + \frac{1}{4\pi} \sum_{j=1}^G \Delta_\ell \omega_{\ell,j} \sum_{k \in FF(\ell)} \sum_{i=1}^G t_k \Delta_k \omega_{k,i} \frac{1}{|x_{\ell,j} - x_{k,i}|} \\
&\approx \sum_{k \in NF(\ell)} V_h[\ell, k] t_k + \frac{1}{4\pi} \sum_{j=1}^G \Delta_\ell \omega_{\ell,j} \sum_{k \in FF(\ell)} \sum_{i=1}^G t_k \Delta_k \omega_{k,i} \sum_{n=0}^p \sum_{m=-n}^n \overline{S_n^m}(x_{k,i}) R_n^m(x_{\ell,j}) \\
&= \sum_{k \in NF(\ell)} V_h[\ell, k] t_k + \frac{1}{4\pi} \sum_{j=1}^G \Delta_\ell \omega_{\ell,j} \sum_{n=0}^p \sum_{m=-n}^n R_n^m(x_{\ell,j}) \sum_{k \in FF(\ell)} \sum_{i=1}^G t_k \Delta_k \omega_{k,i} \overline{S_n^m}(x_{k,i}) \\
&= \sum_{k \in NF(\ell)} V_h[\ell, k] t_k + \frac{1}{4\pi} \sum_{j=1}^G \Delta_\ell \omega_{\ell,j} \sum_{n=0}^p \sum_{m=-n}^n R_n^m(x_{\ell,j}) \tilde{L}_n^m(\ell).
\end{aligned}$$

Dabei sind die Koeffizienten des Fernfeldes

$$\tilde{L}_n^m(\ell) = \sum_{k \in FF(\ell)} L_n^m(k) \quad \text{und} \quad L_n^m(k) = \sum_{i=1}^G t_k \Delta_k \omega_{k,i} \overline{S_n^m}(x_{k,i}).$$

Falls es möglich ist, die Koeffizienten $\tilde{L}_n^m(\ell)$ oder eine gute Approximation effizient zu berechnen, ist eine schnelle Realisierung der Matrixvektormultiplikation möglich. Um den Multipolalgorithmus auf den Fernfeldanteil in der Matrixvektormultiplikation

anwenden zu können, werden die Integrationspunkte $x_{k,i}$ mit den Positionen der Ladungsteilchen in der Multipolmethode identifiziert. Dabei ist im Punkt $x_{k,i}$ die Ladung $q_{k,i} = t_k \Delta_k \omega_{k,i}$.

Somit genügt es in dieser Arbeit die schnelle Multipolmethode für eine Menge an Ladungsteilchen zu betrachten. Eine weitere Analyse für die BEM ist in [18] gegeben.

Die effiziente Berechnung der Koeffizienten \tilde{L}_n^m wird in der schnellen Multipolmethode durch das Einführen einer hierarchischen Struktur über die Ladungsteilchen realisiert. Diese ermöglichen es, durch Verschiebung der Entwicklungszentren mehrere Koeffizienten L_n^m zusammenzufassen. Diese zusammengefassten Koeffizienten werden dann für möglichst viele \tilde{L}_n^m wieder verwendet. Der Algorithmus dafür wird in Kapitel 2 detailliert beschrieben.

Bei vielen Rechnungen in der BEM oder auch in der Partikelsimulation ist die Verteilung der Teilchen nicht uniform. Diese adaptiven Zerlegungen stellen einen neuen Anspruch an die schnelle Multipolmethode und die Frage nach einer Erweiterung der Methode für adaptive Netze. Dabei haben numerische Tests [14] ergeben, dass es nicht genügt, nur die Unterscheidung in Nah- und Fernfeld zu ändern.

In dieser Arbeit wird die adaptive schnelle Multipolmethode untersucht. Mit dieser Version der schnellen Multipolmethode wird, durch die Erweiterung der Standardmethode um zwei weitere Operationen, die Adaptivität in den Verteilungen der Ladungsteilchen berücksichtigt. Diese Arbeit ist in drei Kapitel unterteilt.

Im ersten Kapitel wird die Reihendarstellung des Potentials eingeführt. Diese orientiert sich an den Arbeiten [8, 9]. Für eine geschicktere Darstellung werden zusätzlich die Koeffizienten aus [28] verwendet. Weiters werden in diesem Kapitel die für den Multipolalgorithmus benötigten Operationen [8, 9, 17, 28] definiert. Für diese Operationen wird auch der durch die abgeschnittenen Entwicklungen entstandene Fehler untersucht.

Im zweiten Kapitel wird die adaptive Version der schnellen Multipolmethode beschrieben. Dazu wird zunächst die benötigte hierarchische Struktur definiert. Dafür werden die Ladungsteilchen in Cluster zusammengefasst und in einer Baumstruktur angeordnet. Weiters werden in diesem Kapitel Kriterien zur Unterscheidung in Nah- und Fernfeld definiert. An dieser Stelle genügt in der Standardversion der schnellen Multipolmethode ein Kriterium. Für die adaptive Version wird in dieser Arbeit ein zweites Kriterium neu definiert. Der Hintergrund dazu, wird in Kapitel 2 ausführlich dargelegt. Weiters wird der Multipolalgorithmus [3, 4, 8, 9, 16, 17] diskutiert, und insbesondere jene Stellen hervorgehoben, in denen sich die adaptive Version von der Standardversion unterscheidet.

Im abschließenden dritten Kapitel werden die Ergebnisse aus den vorherigen Kapiteln mit numerischen Beispielen weiter untersucht. Für diese Tests wurde der Algorithmus aus Kapitel 2 in C und C++ implementiert. Als Testverteilungen der Ladungen wurden verschiedene Beispiele, insbesondere aber Integrationspunkte aus einer adaptiven BEM Rechnung, verwendet. Dabei wurde das Verhalten des Fehlers bei unterschiedlichen Parametern betrachtet.

Weiters wurden die Unterschiede in den Rechenzeiten zwischen der adaptiven schnellen Multipolmethode und der Standardmethode untersucht. Für diese Testserie wurden die Parameter so gewählt, dass bei vergleichbarem Fehler die bestmögliche Rechenzeit erzielt wurde. In diesen Tests wurde auch insbesondere auf die Unterschiede zwischen der adaptiven Version der schnellen Multipolmethode und der Standardversion eingegangen. Dabei lieferten diese Beispiele die erhofften Ergebnisse. Die adaptive Version lieferte eine zusätzliche Einsparung an Rechenzeit und Speicheraufwand, die insbesondere bei adaptiven Verteilungen deutlich wurde. Dazu ist zu erwähnen, dass der Fehler durch den zusätzlichen adaptiven Anteil im Wesentlichen gleich bleibt.

1 Kernentwicklung und Multipoloperatoren

Im ersten Abschnitt dieses Kapitels wird die Reihendarstellung des Potentials $\frac{1}{|x-y|}$ zwischen zwei Punkten $x \in \mathbb{R}^3$ und $y \in \mathbb{R}^3$ untersucht. In den Abschnitten zwei und drei werden die für den Multipolalgorithmus aus Kapitel 2 benötigten Operationen definiert und Fehlerabschätzungen angegeben. Falls nicht explizit anders angegeben, befindet sich das Zentrum der Entwicklungen in diesem Kapitel im Ursprung. Dabei orientiert sich dieses Kapitel an den Arbeiten [8, 9, 28].

1.1 Entwicklung des Kerns

Für einen Punkt $y \in \mathbb{R}^3$ mit Einheitsladung ist das Potential $\Phi(x)$ in einem Punkt $x \in \mathbb{R}^3$ gegeben als $\Phi(x) = \frac{1}{|x-y|}$. Für die Konstruktion einer Reihendarstellung dieses Potentials werden die Legendre-Polynome P_n benötigt.

Lemma 1.1. [17, S. 24] *Die Legendre-Polynome*

$$P_n(u) := \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} \frac{(2n-2k)!}{2^n (n-k)! k! (n-2k)!} (-1)^k u^{n-2k} = \frac{1}{2^n n!} \frac{d^n}{du^n} (u^2 - 1)^n \quad (1.1)$$

sind eine Lösung der Legendre-Differentialgleichung

$$(1-u^2) \frac{d^2}{du^2} v(u) - 2u \frac{d}{du} v(u) + n(n+1)v(u) = 0. \quad (1.2)$$

Mit der Einführung von Kugelkoordinaten für $x = (r, \theta, \phi)$ und $y = (\rho, \alpha, \beta)$ sowie dem Winkel γ zwischen den Vektoren x und y gilt mit dem Kosinussatz für den Abstand $R = |x-y|$:

$$R^2 = r^2 + \rho^2 - 2r\rho \cos \gamma. \quad (1.3)$$

Mit dem Kosinussatz lässt sich das Potential weiter umformen zu

$$\frac{1}{R} = \frac{1}{r \sqrt{1 - 2\frac{\rho}{r} \cos \gamma + \frac{\rho^2}{r^2}}} = \frac{1}{r \sqrt{1 - 2u\mu + \mu^2}}$$

mit der Wahl der Koeffizienten

$$\mu = \frac{\rho}{r} \quad \text{und} \quad u = \cos \gamma.$$

Das folgende Lemma wird für die weiteren Umformungen benötigt. Ein Beweis von Lemma 1.2 ist z.B. in [17, L. 2.2] zu finden.

Lemma 1.2. *Für $|\mu| < 1$ und $u \in [-1, 1]$ gilt die Potenzreihendarstellung:*

$$\frac{1}{\sqrt{1 - 2u\mu + \mu^2}} = \sum_{n=0}^{\infty} P_n(u)\mu^n. \quad (1.4)$$

Mit diesem Lemma folgt für die Darstellung des Potentials weiter

$$\frac{1}{|x - y|} = \frac{1}{R} = \sum_{n=0}^{\infty} \frac{\rho^n}{r^{n+1}} P_n(\cos \gamma) =: \Phi(x). \quad (1.5)$$

Diese Entwicklung wird als Multipolentwicklung bezeichnet und beschreibt das Fernfeld der Ladung im Punkt y für alle x mit $r > \rho$. Bei Vertauschung der Positionen des Ladungspunktes y und des Auswertungspunktes x wird das Potential ebenfalls durch $\frac{1}{R}$ dargestellt. Somit folgt für alle x mit $r < \rho$

$$\frac{1}{R} = \sum_{n=0}^{\infty} \frac{r^n}{\rho^{n+1}} P_n(\cos \gamma) =: \Psi(x). \quad (1.6)$$

Diese Darstellung ist die lokale Entwicklung $\Psi(x)$. Im folgenden Lemma wird eine Fehlerabschätzung für die beiden Entwicklungen angegeben.

Lemma 1.3. [9, L. 3.2.4] *Sei q eine Ladung im Punkt $y = (\rho, \alpha, \beta)$ und $x = (r, \theta, \phi)$ ein Punkt im \mathbb{R}^3 . Weiters sei γ der Winkel zwischen den Vektoren x und y und $R = |x - y|$. Dann gilt für $r > \rho$ für den Fehler der Multipolentwicklung in (1.5):*

$$\left| \frac{q}{R} - \sum_{n=0}^p \frac{q\rho^n}{r^{n+1}} P_n(\cos \gamma) \right| \leq \frac{|q|}{r - \rho} \left(\frac{\rho}{r} \right)^{p+1}. \quad (1.7)$$

Ebenso gilt für den Fehler der lokalen Entwicklung in (1.6) für $r < \rho$

$$\left| \frac{q}{R} - \sum_{n=0}^p \frac{qr^n}{\rho^{n+1}} P_n(\cos \gamma) \right| \leq \frac{|q|}{\rho - r} \left(\frac{r}{\rho} \right)^{p+1}. \quad (1.8)$$

Beweis. Der Beweis wird für die Multipolentwicklung gebracht. Für die lokale Entwicklung verläuft dieser ident. Im Beweis wird verwendet, dass die Legendre-Polynome folgende Eigenschaft [17, L. 2.3] erfüllen:

$$|P_n(u)| \leq 1, \quad \text{für } u \in [-1, 1]. \quad (1.9)$$

Mit dieser Abschätzung für die Legendre-Polynome folgt für die Fehlerabschätzung der Multipolentwicklung

$$\begin{aligned}
\left| \frac{q}{R} - \sum_{n=0}^p \frac{q\rho^n}{r^{n+1}} P_n(\cos \gamma) \right| &= \left| \sum_{n=p+1}^{\infty} \frac{q\rho^n}{r^{n+1}} P_n(\cos \gamma) \right| \\
&\leq \sum_{n=p+1}^{\infty} \frac{|q|\rho^n}{r^{n+1}} |P_n(\cos \gamma)| \\
&\leq \sum_{n=p+1}^{\infty} \frac{|q|\rho^n}{r^{n+1}} = \frac{|q|}{r} \sum_{n=p+1}^{\infty} \left(\frac{\rho}{r}\right)^n \\
&\leq \frac{|q|}{r-\rho} \left(\frac{\rho}{r}\right)^{p+1}.
\end{aligned}$$

Dabei wurde im letzten Schritt die Summenformel für die geometrische Reihe verwendet. \square

Das Potential, das von ℓ Ladungen q_i in Punkten $y_i = (\rho_i, \alpha_i, \beta_i)$ ausgeht, wird für $x = (r, \theta, \phi)$ mit $r > \rho_i, i = 1, \dots, \ell$, aufgrund von Superposition beschrieben durch

$$\Phi(x) = \sum_{i=1}^{\ell} \Phi_i(x) = \sum_{i=1}^{\ell} \sum_{n=0}^{\infty} \frac{q_i \rho_i^n}{r^{n+1}} P_n(\cos \gamma_i). \quad (1.10)$$

Dabei ist γ_i der Winkel zwischen x und y_i . Die Darstellung der beiden Entwicklungen hängt vom Winkel zwischen den Vektoren ab und muss somit für jedes Paar an Punkten explizit berechnet werden. Um eine effiziente Berechnung zu ermöglichen, wird im Folgenden eine Darstellung mit getrennten Variablen untersucht. Dafür wird folgendes Theorem [26] benötigt.

Theorem 1.4 (Additionstheorem für Legendre-Polynome,[26]). *Seien x und y zwei Punkte mit Kugelkoordinaten (r, θ, ϕ) und (ρ, α, β) . Weiters sei γ der Winkel zwischen den beiden Ortsvektoren. Dann gilt:*

$$P_n(\cos(\gamma)) = \sum_{m=-n}^n \frac{(n-m)!}{(n+m)!} P_n^m(\cos(\alpha)) e^{-im\beta} P_n^m(\cos(\theta)) e^{im\phi}.$$

Dabei ist das zugehörige Legendre-Polynom $P_n^m(u) = (-1)^m (1-u^2)^{\frac{m}{2}} \frac{d^m P_n(u)}{du^m}$. Für P_n^m gelten folgende Eigenschaften[28, Appendix A, (A.1)–(A.3)].

Lemma 1.5 (Eigenschaften der zugehörigen Legendre-Polynome).

- I) $P_n^m(\cos \theta) = 0, \quad (n < |m|)$
- II) $P_n^{-m}(\cos \theta) = (-1)^m \frac{(n-m)!}{(n+m)!} P_n^m(\cos \theta)$

$$III) P_n^m(-\cos\theta) = (-1)^{n+m} P_n^m(\cos\theta)$$

Mit Theorem 1.4 folgt für die Multipolentwicklung $\Phi(x)$ in (1.5) weiter

$$\begin{aligned} \sum_{n=0}^{\infty} \frac{\rho^n}{r^{n+1}} P_n(\cos\gamma) &= \sum_{n=0}^{\infty} \frac{\rho^n}{r^{n+1}} \sum_{m=-n}^n \frac{(n-m)!}{(n+m)!} P_n^m(\cos(\alpha)) e^{-im\beta} P_n^m(\cos(\theta)) e^{im\phi} \\ &= \sum_{n=0}^{\infty} \sum_{m=-n}^n \frac{(n-m)!}{(n+m)!} P_n^m(\cos(\alpha)) e^{-im\beta} P_n^m(\cos(\theta)) e^{im\phi} \frac{\rho^n}{r^{n+1}} \\ &= \sum_{n=0}^{\infty} \sum_{m=-n}^n S_n^m(x) \overline{R}_n^m(y) = \sum_{n=0}^{\infty} \sum_{m=-n}^n \overline{S}_n^m(x) R_n^m(y) \end{aligned}$$

mit den modifizierten Kugelfunktionen [19, 27, 28]

$$R_n^m(y) = \frac{1}{(n+m)!} P_n^m(\cos(\alpha)) e^{im\beta} \rho^n, \quad (1.11)$$

$$S_n^m(x) = (n-m)! P_n^m(\cos(\theta)) e^{im\phi} \frac{1}{r^{n+1}}. \quad (1.12)$$

Die modifizierten Kugelfunktionen erfüllen folgende Eigenschaften [28, Appendix A, (A.5)–(A.8)]

Lemma 1.6. *Sei C_M das Zentrum der Entwicklungen. Dann gilt für die modifizierten Kugelfunktionen R_n^m und S_n^m :*

$$I) R_n^{-m}(x) = (-1)^m \overline{R}_n^m(x)$$

$$II) S_n^{-m}(x) = (-1)^m \overline{S}_n^m(x)$$

$$III) R_n^m(\overrightarrow{yC_M}) = (-1)^n R_n^m(\overrightarrow{C_M y})$$

$$IV) S_n^m(\overrightarrow{yC_M}) = (-1)^n S_n^m(\overrightarrow{C_M y})$$

Beweis. Der Beweis dieses Lemmas folgt aus den Eigenschaften der zugehörigen Legendre-Polynome P_n^m aus Lemma 1.5.

$$\begin{aligned} I) R_n^{-m}(x) &= \frac{1}{(n-m)!} P_n^{-m}(\cos(\theta)) e^{-im\phi} r^n \\ &= \frac{1}{(n-m)!} (-1)^m \frac{(n-m)!}{(n+m)!} P_n^m(\cos(\theta)) e^{-im\phi} r^n \\ &= (-1)^m \frac{1}{(n+m)!} P_n^m(\cos(\theta)) e^{-im\phi} r^n = (-1)^m \overline{R}_n^m(x) \end{aligned}$$

$$\begin{aligned} II) S_n^{-m}(x) &= \frac{(n+m)!}{r^{n+1}} P_n^{-m}(\cos(\theta)) e^{-im\phi} \\ &= \frac{(n+m)!}{r^{n+1}} (-1)^m \frac{(n-m)!}{(n+m)!} P_n^m(\cos(\theta)) e^{-im\phi} \\ &= (-1)^m \frac{(n-m)!}{r^{n+1}} P_n^m(\cos(\theta)) e^{-im\phi} = (-1)^m \overline{S}_n^m(x) \end{aligned}$$

$$\begin{aligned}
\text{III) } R_n^m(\overrightarrow{yC_M}) &= \frac{1}{(n+m)!} P_n^m(\cos(\theta + \pi)) e^{im(\phi+\pi)} r^n \\
&= \frac{1}{(n+m)!} P_n^m(-\cos(\theta)) (-1)^m e^{im\phi} r^n \\
&= \frac{1}{(n+m)!} (-1)^{n+m} P_n^m(\cos(\theta)) (-1)^m e^{im\phi} r^n = (-1)^n R_n^m(\overrightarrow{C_M y}) \\
\text{IV) } S_n^m(\overrightarrow{yC_M}) &= \frac{(n-m)!}{r^{n+1}} P_n^m(\cos(\theta + \pi)) e^{im(\phi+\pi)} \\
&= \frac{(n-m)!}{r^{n+1}} P_n^m(-\cos(\theta)) (-1)^m e^{im\phi} \\
&= \frac{(n-m)!}{r^{n+1}} (-1)^{n+m} P_n^m(\cos(\theta)) (-1)^m e^{im\phi} = (-1)^n S_n^m(\overrightarrow{C_M y}).
\end{aligned}$$

□

1.2 Multipolentwicklung und lokale Entwicklung

In diesem Abschnitt werden die im Multipolalgorithmus benötigten Operationen für das Erstellen und Auswerten der Multipolentwicklung und der lokalen Entwicklung definiert. Die Entwicklungen finden sich bereits in [8, 9] wieder. Allerdings wird eine geschicktere Darstellung [28] verwendet.

1.2.1 Multipolentwicklung

Für das Potential mehrerer Ladungen in (1.10) folgt mit der Darstellung durch die modifizierten Kugelfunktionen

$$\begin{aligned}
\Phi(x) &= \sum_{i=1}^{\ell} \Phi_i(x) = \sum_{i=1}^{\ell} \sum_{n=0}^{\infty} \frac{q_i \rho_i^n}{r^{n+1}} P_n(\cos \gamma_i) \\
&= \sum_{i=1}^{\ell} \sum_{n=0}^{\infty} q_i \sum_{m=-n}^n \overline{S_n^m}(x) R_n^m(y_i) \\
&= \sum_{n=0}^{\infty} \sum_{m=-n}^n \overline{S_n^m}(x) \sum_{i=1}^{\ell} q_i R_n^m(y_i) \\
&= \sum_{n=0}^{\infty} \sum_{m=-n}^n \overline{S_n^m}(x) M_n^m
\end{aligned} \tag{1.13}$$

mit den Multipolkoeffizienten

$$M_n^m = \sum_{i=1}^{\ell} q_i R_n^m(y_i). \tag{1.14}$$

Aus dieser Umformung lassen sich zwei Sätzen formulieren. Dabei wird im Folgenden das Erstellen der Multipolentwicklung als S2M (Source to Multipole) und die Auswertung der Multipolentwicklung als M2T (Multipole to Target) bezeichnet.

Satz 1.7 (Berechnung der Multipolentwicklung (S2M)). *Gegeben seien ℓ Teilchen $y_i \in \mathbb{R}^3$, $i = 1 \dots \ell$, mit Ladungen q_i . Dann gilt für die Koeffizienten M_n^m der Multipolentwicklung:*

$$M_n^m = \sum_{i=1}^{\ell} q_i R_n^m(y_i) \quad \text{für } n = 1, \dots, p; m = 0, \dots, n. \quad (1.15)$$

Zusätzlich gilt für die Koeffizienten M_n^m :

$$M_n^{-m} = (-1)^m \overline{M_n^m}. \quad (1.16)$$

Die Berechnung der Funktionen R_n^m erfolgt mit der Rekursion

$$\begin{aligned} R_0^0(y) &= 1 \\ R_n^n(y) &= \frac{y_1 + iy_2}{2n} R_{n-1}^{n-1}(y) && \text{für } n = 1, \dots, p, \\ R_n^{n-1}(y) &= y_3 R_{n-1}^{n-1}(y) && \text{für } n = 1, \dots, p, \\ R_n^m(y) &= \frac{(2n-1)y_3 R_{n-1}^m(y) - r^2 R_{n-1}^m(y)}{(n+m)(n-m)} && \text{für } n = 1, \dots, p; m = 0, \dots, n-2. \end{aligned}$$

Mit der Eigenschaft in (1.16) ist es in der Implementierung nur nötig, die Koeffizienten für m größer gleich Null abzuspeichern.

Beweis. Die Darstellung der Koeffizienten M_n^m wurde bereits in (1.14) definiert. Für die Eigenschaft in (1.16) gilt mit Lemma 1.6:

$$M_n^{-m} = \sum_{i=1}^{\ell} q_i R_n^{-m}(y_i) = \sum_{i=1}^{\ell} (-1)^m q_i \overline{R_n^m}(y_i) = (-1)^m \overline{M_n^m}.$$

Die Rekursion der Koeffizienten R_n^m folgt aus der Rekursionsformel für die zugehörigen Legendre-Polynome P_n^m und ist in [28, Appendix C] detailliert ausgeführt. \square

Der folgende Satz beschreibt die Auswertung der Multipolentwicklung. Dabei wird mit $\Re(\cdot)$ der Realteil bezeichnet.

Satz 1.8 (Auswertung der Multipolentwicklung (M2T)). *Gegeben seien ℓ Teilchen $y_i \in \mathbb{R}^3$, $i = 1 \dots \ell$, mit Ladungen q_i und deren Multipolentwicklung mit Koeffizienten M_n^m . Dann gilt für alle $x \in \mathbb{R}^3$, mit $|x| > |y_i|$, $i = 1, \dots, \ell$, für die Auswertung der Multipolentwicklung mit Entwicklungsgrad p :*

$$M2T(x) = \sum_{n=0}^p \Re(\overline{S_n^0}(x) M_n^0) + 2 \sum_{n=0}^p \sum_{m=1}^n \Re(\overline{S_n^m}(x) M_n^m). \quad (1.17)$$

Zusätzlich gilt für $p \geq 1$ für den Fehler der Entwicklung:

$$|\Phi(x) - M2T(x)| \leq \frac{\sum_{i=1}^{\ell} |q_i|}{|x| - w} \left(\frac{w}{|x|} \right)^{p+1} \quad (1.18)$$

mit $w = \max |y_i|, i = 1, \dots, \ell$.

Beweis. Ausgehend von der Darstellung in (1.13) folgt mit einer Spaltung der Summe und Verwendung von (1.16)

$$\begin{aligned} M2T(x) &= \sum_{n=0}^p \sum_{m=-n}^n \overline{S_n^m}(x) M_n^m \\ &= \sum_{n=0}^p \overline{S_n^0}(x) M_n^0 + \sum_{n=0}^p \sum_{m=1}^n \overline{S_n^m}(x) M_n^m + \sum_{n=0}^p \sum_{m=1}^n \overline{S_n^{-m}}(x) M_n^{-m} \\ &= \sum_{n=0}^p \overline{S_n^0}(x) M_n^0 + \sum_{n=0}^p \sum_{m=1}^n \overline{S_n^m}(x) M_n^m + \sum_{n=0}^p \sum_{m=1}^n (-1)^m \overline{S_n^m}(x) (-1)^m M_n^m \\ &= \sum_{n=0}^p \overline{S_n^0}(x) M_n^0 + \sum_{n=0}^p \sum_{m=1}^n \overline{S_n^m}(x) M_n^m + S_n^m(x) \overline{M_n^m} \\ &= \sum_{n=0}^p \Re(\overline{S_n^0}(x) M_n^0) + 2 \sum_{n=0}^p \sum_{m=1}^n \Re(\overline{S_n^m}(x) M_n^m). \end{aligned}$$

Die Fehlerabschätzung folgt durch Superposition aus dem Fehler der Multipolentwicklung für einen Punkt in (1.7) und ist in [9] ausgearbeitet. \square

1.2.2 Lokale Entwicklung

Ebenso wie bei der Multipolentwicklung lässt sich die lokale Entwicklung $\Psi(x)$ in (1.6) mit Verwendung von Theorem 1.4 als Reihe mit getrennten Variablen darstellen:

$$\Psi(x) = \sum_{n=0}^{\infty} \frac{r^n}{\rho^{n+1}} P_n(\cos \gamma) = \sum_{n=0}^{\infty} \sum_{m=-n}^n \overline{S_n^m}(y) R_n^m(x).$$

Die Auswertung einer lokalen Entwicklung wird im Folgenden mit L2T (Local to Target) bezeichnet. Für das Erstellen einer lokalen Entwicklung aus einer Menge an Quellen wird S2L (Source to Local) geschrieben. Für die Auswertung der lokalen Entwicklung von ℓ Ladungen q_i in Punkten $y_i \in \mathbb{R}^3$ in einem Punkt $x \in \mathbb{R}^3$, mit

$|x| < |y_i|, i = 1, \dots, \ell$, folgt

$$\begin{aligned} L2T(x) &= \sum_{i=1}^{\ell} q_i \sum_{n=0}^p \sum_{m=-n}^n \overline{S}_n^m(y_i) R_n^m(x) \\ &= \sum_{n=0}^p \sum_{m=-n}^n \sum_{i=1}^{\ell} q_i \overline{S}_n^m(y_i) R_n^m(x) \\ &= \sum_{n=0}^p \sum_{m=-n}^n \overline{L}_n^m R_n^m(x) \end{aligned}$$

mit den lokalen Koeffizienten

$$L_n^m = \sum_{i=1}^{\ell} q_i S_n^m(y_i).$$

Diese Ergebnisse werden wieder in zwei Sätzen für die jeweiligen Operationen zusammengefasst.

Satz 1.9 (Berechnung der lokalen Entwicklung (S2L)). *Gegeben seien ℓ Teilchen $y_i \in \mathbb{R}^3$ mit Ladungen q_i . Dann gilt für die Koeffizienten L_n^m der lokalen Entwicklung $\Psi(x)$:*

$$L_n^m = \sum_{i=1}^{\ell} q_i S_n^m(y_i) \quad \text{für } n = 1, \dots, p; m = 0, \dots, n. \quad (1.19)$$

Weiters gilt für die Koeffizienten L_n^m :

$$L_n^{-m} = (-1)^m \overline{L}_n^m. \quad (1.20)$$

Die Berechnung der Funktionen S_n^m erfolgt mit folgender Rekursion

$$\begin{aligned} S_0^0(y) &= \frac{1}{r} \\ S_n^n(y) &= (2n-1) \frac{y_1 + iy_2}{r^2} S_{n-1}^{n-1}(y) \quad \text{für } n = 1, \dots, p, \\ S_n^{n-1}(y) &= \frac{1}{r^2} (2n-1) y_3 S_{n-1}^{n-1}(y) \quad \text{für } n = 1, \dots, p, \\ S_n^m(y) &= \frac{1}{r^2} [(2n-1) y_3 S_{n-1}^m(y) - (n-1+m)(n-1-m) S_{n-2}^m(y)] \\ &\quad \text{für } n = 1, \dots, p; m = 0, \dots, n-2. \end{aligned}$$

Beweis. Der Beweis von Satz 1.9 erfolgt ident zu Satz 1.7. Die Rekursion ist in [28, Appendix C] ausgearbeitet. \square

Wie bei der Multipolentwicklung genügt es, in der Implementierung der L_n^m , aufgrund der Eigenschaft in (1.20), nur die Koeffizienten mit m größer gleich Null abzuspeichern.

Satz 1.10 (Auswertung der lokalen Entwicklung (L2T)). *Gegeben seien ℓ Teilchen $y_i \in \mathbb{R}^3$ mit Ladungen q_i und deren lokale Entwicklung $\Psi(x)$ aus Satz 1.9 mit Koeffizienten L_n^m . Dann gilt für alle $x \in \mathbb{R}^3$ mit $|x| < |y_i|, i = 1, \dots, \ell$, für die Auswertung der lokalen Entwicklung vom Grad p :*

$$L2T(x) = \sum_{n=0}^p \Re(\overline{L_n^0} R_n^0(x)) + 2 \sum_{n=0}^p \sum_{m=1}^n \Re(\overline{L_n^m} R_n^m(x)). \quad (1.21)$$

Weiters gilt mit $p \geq 1$ für den Fehler der Entwicklung:

$$|\Psi(x) - L2T(x)| \leq \frac{\sum_{i=1}^{\ell} |q_i|}{w - |x|} \left(\frac{|x|}{w} \right)^{p+1} \quad (1.22)$$

mit $w = \min |y_i|, i = 1, \dots, \ell$.

Beweis. Die Darstellung der Operation S2T folgt aus den vorher erarbeiteten Ergebnissen und einer kurzen Umrechnung ident zu dem Beweis von Satz 1.8. Ebenso folgt die Fehlerabschätzung direkt aus dem Fehler der lokalen Entwicklung für einen Punkt in (1.8) durch Superposition und ist in [9] ausgeführt. \square

1.3 Verschiebung und Umrechnung der Entwicklungen

Zusätzlich zu den Operationen für das Erstellen und das Auswerten der Multipolentwicklung und lokalen Entwicklung aus Abschnitt 1.2 werden für den Multipolalgorithmus drei weitere Operationen benötigt. Die Operationen M2M (Multipole to Mutipole) und L2L (Local to Local) verschieben das Entwicklungszentrum der Multipol- und lokalen Entwicklung. Die Operation M2L (Multipole to Local) transformiert eine Multipolentwicklung in eine lokale Entwicklung mit neuem Entwicklungszentrum. Der folgende Satz beschreibt die Operation M2M.

Satz 1.11 (Verschiebung der Multipolentwicklung (M2M),[9]). *Seien $M_s^t(C_N)$ die Koeffizienten einer Multipolentwicklung von ℓ Ladungen q_i in Punkten $y_i \in \mathbb{R}^3$ mit Entwicklungszentrum in C_N . Dann gilt für die Berechnung der Koeffizienten $M_n^m(C_M)$ einer Multipolentwicklung mit Zentrum in C_M aus den Koeffizienten $M_s^t(C_N)$:*

$$M_n^m(C_M) = \sum_{s=0}^n \sum_{t=-s}^s R_s^t(\overrightarrow{C_M C_N}) M_{n-s}^{m-t}(C_N). \quad (1.23)$$

Für den Fehler der neuen Entwicklung gilt für $p \geq 1$, $w = \max |\overrightarrow{C_N y_i}|$ und $|\overrightarrow{C_M x}| > w + |\overrightarrow{C_M C_N}|$:

$$\left| \Phi(x) - \left(\sum_{n=0}^p \Re(\overrightarrow{S_n^0}(\overrightarrow{C_M x}) M_n^m(C_M)) + 2 \sum_{n=0}^p \sum_{m=1}^n \Re(\overrightarrow{S_n^m}(\overrightarrow{C_M x}) M_n^m(C_M)) \right) \right| \leq \frac{\sum_{i=1}^{\ell} |q_i|}{|\overrightarrow{C_M x}| - (w + |\overrightarrow{C_M C_N}|)} \left(\frac{w + |\overrightarrow{C_M C_N}|}{|\overrightarrow{C_M x}|} \right)^{p+1}.$$

Beweis. Die Darstellung der Koeffizienten der Multipolentwicklung $M_n^m(C_M)$ für eine Menge an Punkten $y_i, i = 1, \dots, \ell$, mit Ladungen q_i ist nach Satz 1.7 gegeben durch

$$M_n^m(C_M) = \sum_{i=1}^{\ell} q_i R_n^m(\overrightarrow{C_M y_i}).$$

Durch Einsetzen von [28, Appendix B, (B.11)]

$$R_n^m(\overrightarrow{C_M y_i}) = \sum_{s=0}^n \sum_{t=-s}^s R_s^t(\overrightarrow{C_M C_N}) R_{n-s}^{m-t}(\overrightarrow{C_N y_i}) \quad (1.24)$$

folgt

$$\begin{aligned} M_n^m(C_M) &= \sum_{i=1}^{\ell} q_i R_n^m(\overrightarrow{C_M y_i}) \\ &= \sum_{i=1}^{\ell} q_i \sum_{s=0}^n \sum_{t=-s}^s R_s^t(\overrightarrow{C_M C_N}) R_{n-s}^{m-t}(\overrightarrow{C_N y_i}) \\ &= \sum_{s=0}^n \sum_{t=-s}^s R_s^t(\overrightarrow{C_M C_N}) \sum_{i=1}^{\ell} q_i R_{n-s}^{m-t}(\overrightarrow{C_N y_i}) \\ &= \sum_{s=0}^n \sum_{t=-s}^s R_s^t(\overrightarrow{C_M C_N}) M_{n-s}^{m-t}(C_N). \end{aligned}$$

Ein Beweis für die Fehlerabschätzung folgt aus der Fehlerabschätzung in Satz 1.8 und ist in [9] ausformuliert. \square

Die Verschiebung der Multipolentwicklung hat keinen Einfluss auf die Eigenschaft

in (1.16), da mit Lemma 1.6[I] weiter folgt:

$$\begin{aligned}
M_n^{-m}(C_M) &= \sum_{s=0}^n \sum_{t=-s}^s R_s^t(\overrightarrow{C_M C_N}) M_{n-s}^{-m-t}(C_N) \\
&= \sum_{s=0}^n \sum_{t=-s}^s (-1)^t \overrightarrow{R_s^{-t}}(\overrightarrow{C_M C_N}) (-1)^{m+t} \overrightarrow{M_{n-s}^{m+t}}(C_N) \\
&= (-1)^m \sum_{s=0}^n \sum_{r=-s}^s \overrightarrow{R_s^r}(\overrightarrow{C_M C_N}) \overrightarrow{M_{n-s}^{m-r}}(C_N) = (-1)^m \overrightarrow{M_n^m}(C_M).
\end{aligned}$$

Im nächsten Satz wird die Transformation einer Multipolentwicklung in eine lokale Entwicklung beschrieben.

Satz 1.12 (Transformation der Multipolentwicklung in eine lokale Entwicklung (M2L), [9]). *Seien $M_s^t(C_M)$ die Koeffizienten einer Multipolentwicklung von ℓ Ladungen q_i in Punkten $y_i \in \mathbb{R}^3$ mit Entwicklungszentrum in C_M . Für die Berechnung der Koeffizienten $L_n^m(C_L)$ einer lokalen Entwicklung mit Zentrum in C_L aus den Koeffizienten $M_s^t(C_M)$ gilt:*

$$L_n^m(C_L) = (-1)^n \sum_{s=0}^{\infty} \sum_{t=-s}^s S_{n+s}^{m+t}(\overrightarrow{C_M C_L}) \overrightarrow{M_s^t}(C_M). \quad (1.25)$$

Für den Fehler dieser Entwicklung gilt:

$$\left| \Psi(x) - \left(\sum_{n=0}^p \Re(\overrightarrow{L_n^0} R_n^0(x)) + 2 \sum_{n=0}^p \sum_{m=1}^n \Re(\overrightarrow{L_n^m} R_n^m(x)) \right) \right| \leq \frac{\sum_{i=1}^{\ell} |q_i|}{w - |\overrightarrow{C_L x}|} \left(\frac{|\overrightarrow{C_L x}|}{w} \right)^{p+1} \quad (1.26)$$

mit $w = \min |\overrightarrow{C_L y_i}|, i = 1, \dots, \ell$.

Beweis. Mit [28, Appendix B, (B.5)]

$$\overrightarrow{S_n^m}(\overrightarrow{C_M x}) = \sum_{s=0}^{\infty} (-1)^s \sum_{t=-s}^s \overrightarrow{S_{n+s}^{m+t}}(\overrightarrow{C_M C_L}) R_s^t(\overrightarrow{C_L x}) \quad (1.27)$$

eingesetzt in die Multipolentwicklung folgt

$$\begin{aligned}
\sum_{n=0}^{\infty} \sum_{m=-n}^n \overrightarrow{S_n^m}(\overrightarrow{C_M x}) M_n^m &= \sum_{n=0}^{\infty} \sum_{m=-n}^n \sum_{s=0}^{\infty} \sum_{t=-s}^s (-1)^s \overrightarrow{S_{n+s}^{m+t}}(\overrightarrow{C_M C_L}) R_s^t(\overrightarrow{C_L x}) M_n^m \\
&= \sum_{s=0}^{\infty} \sum_{t=-s}^s (-1)^s \sum_{n=0}^{\infty} \sum_{m=-n}^n \overrightarrow{S_{n+s}^{m+t}}(\overrightarrow{C_M C_L}) M_n^m R_s^t(\overrightarrow{C_L x}) \\
&= \sum_{s=0}^{\infty} \sum_{t=-s}^s \overrightarrow{L_s^t} R_s^t(\overrightarrow{C_L x}).
\end{aligned}$$

Ein Beweis für die Fehlerabschätzung ist in [9] angegeben. □

Die Umwandlung der Multipolentwicklung in eine lokale Entwicklung beeinflusst die Eigenschaft in (1.20) nicht. Mit Lemma 1.6 folgt

$$\begin{aligned}\overline{L_n^{-m}} &= (-1)^n \sum_{s=0}^{\infty} \sum_{t=-s}^s \overline{S_{n+s}^{-m+t}(\overrightarrow{C_M C_L})} M_s^t \\ &= (-1)^n \sum_{s=0}^{\infty} \sum_{t=-s}^s (-1)^{-m+t} S_{n+s}^{m-t}(\overrightarrow{C_M C_L}) (-1)^t \overline{M_s^{-t}} \\ &= (-1)^m (-1)^n \sum_{s=0}^{\infty} \sum_{r=-s}^s S_{n+s}^{m+r}(\overrightarrow{C_M C_L}) \overline{M_s^r} = (-1)^m L_n^m.\end{aligned}$$

Für die Umwandlung der Multipolentwicklung in eine lokale Entwicklung ist zu beachten, dass in der Darstellung der lokalen Koeffizienten sowie der Fehlerabschätzung für die Entwicklung mit der exakten Multipolentwicklung gerechnet wird. In der Implementierung ist die exakte Entwicklung nicht vorhanden, sondern nur die Approximation vom Grad p . Die Koeffizienten L_n^m der M2L-Operation haben also folgende Gestalt:

$$L_n^m(C_L) = (-1)^n \sum_{s=0}^p \sum_{t=-s}^s (\overrightarrow{C_M C_L}) \overline{M_s^t}(C_M).$$

Numerische Experimente haben gezeigt, dass bei einem groß genug gewählten Abstand der Entwicklungszentren, sowie einem ähnlich großen Abstand der Ladungs- und Auswertungspunkte von ihrem Entwicklungszentrum die Konvergenz des erhaltenen Fehlers ähnlich dem der Fehlerabschätzung aus Satz 1.12 ist. Eine genauere Untersuchung des Fehlers mit der abgeschnittenen Entwicklung ist in [20] zu finden. Satz 1.13 beschreibt die Verschiebung des Zentrums einer lokalen Entwicklung. Diese Verschiebung ist exakt, und es wird keine neue Fehlerabschätzung benötigt.

Satz 1.13 (Verschiebung der lokalen Entwicklung (L2L),[9]). *Für die Berechnung der Koeffizienten $L_n^m(C_M)$ einer lokalen Entwicklung mit Zentrum in C_M aus den Koeffizienten $L_s^t(C_N)$ einer gegebenen lokalen Entwicklung mit Zentrum in C_N gilt:*

$$L_n^m(C_M) = \sum_{s=n}^p \sum_{t=-s}^s \overline{R_{s-n}^{t-m}(\overrightarrow{C_N C_M})} L_s^t(C_N). \quad (1.28)$$

Beweis. Mit Lemma 1.6 und (1.24) gilt:

$$\begin{aligned}R_n^m(\overrightarrow{C_N x}) &= (-1)^n R_n^m(\overrightarrow{x C_N}) \\ &= (-1)^n \sum_{s=0}^n \sum_{t=-s}^s (-1)^{n-s} R_{n-s}^{m-t}(\overrightarrow{C_M C_N}) (-1)^s R_s^t(\overrightarrow{x C_M}) \\ &= \sum_{s=0}^n \sum_{t=-s}^s R_{n-s}^{m-t}(\overrightarrow{C_N C_M}) R_s^t(\overrightarrow{C_M x}).\end{aligned}$$

Mit dieser Darstellung folgt für die lokale Entwicklung

$$\begin{aligned}
\sum_{n=0}^p \sum_{m=-n}^n \overline{L}_n^m(C_N) R_n^m(\overrightarrow{C_N x}) &= \sum_{n=0}^p \sum_{m=-n}^n \overline{L}_n^m(C_N) \sum_{s=0}^n \sum_{t=-s}^s R_{n-s}^{m-t}(\overrightarrow{C_N C_M}) R_s^t(\overrightarrow{C_M x}) \\
&= \sum_{s=0}^p \sum_{t=-s}^s \sum_{n=s}^p \sum_{m=-n}^n R_{n-s}^{m-t}(\overrightarrow{C_N C_M}) \overline{L}_n^m(C_N) R_s^t(\overrightarrow{C_M x}) \\
&= \sum_{s=0}^p \sum_{t=-s}^s \overline{L}_s^t(C_M) R_s^t(\overrightarrow{C_M x})
\end{aligned}$$

mit

$$\overline{L}_s^t(C_M) = \sum_{n=s}^p \sum_{m=-n}^n R_{n-s}^{m-t}(\overrightarrow{C_N C_M}) \overline{L}_n^m(C_N).$$

□

Ebense wie bei der Verschiebung der Multipolentwicklung bleibt die Eigenschaft in (1.20) erhalten, denn es gilt mit Lemma 1.6

$$\begin{aligned}
L_n^{-m}(C_M) &= \sum_{s=n}^p \sum_{t=-s}^s \overline{R}_{s-n}^{t+m}(\overrightarrow{C_N C_M}) L_s^t(C_N) \\
&= \sum_{s=n}^p \sum_{t=-s}^s (-1)^{t+m} R_{s-n}^{-t-m}(\overrightarrow{C_N C_M}) (-1)^t \overline{L}_s^{-t}(C_N) \\
&= (-1)^m \sum_{s=n}^p \sum_{r=-s}^s R_{s-n}^{r-m}(\overrightarrow{C_N C_M}) \overline{L}_s^r(C_N) = (-1)^m \overline{L}_n^m(C_M).
\end{aligned}$$

2 Der Multipolalgorithmus

In diesem Kapitel erfolgt die Beschreibung des adaptiven Multipolalgorithmus für die Berechnung des Potentials innerhalb einer Ladungsverteilung. Der beschriebene Algorithmus orientiert sich an den Arbeiten [3, 4, 8, 9, 16]. Dabei kann die Menge der Auswertungspunkte $\{x_j\}_{j=1}^N$ und die Menge der Ladungspunkte $\{q_i, y_i\}_{i=1}^M$ verschieden oder ident sein. Das Potential in jedem Punkt x_i , das von den Quellen in y_i mit Ladung q_i ausgeht, wird mit

$$\Phi(x_i) = \sum_{j=1}^M \frac{1}{|x_i - y_j|} \quad i = 1, \dots, N$$

bezeichnet. Bei dieser direkten Berechnung des Potentials entsteht ein Rechenaufwand von $\mathcal{O}(NM)$. Wie in der Einleitung anhand des Einfachschichtpotentials angedeutet, wird das Potential im Folgenden in einen Nahfeldanteil und einen Fernfeldanteil zerlegt:

$$\Phi = \Phi_{Nahfeld} + \Phi_{Fernfeld}. \quad (2.1)$$

Diese Unterscheidung ist notwendig, da der Fehler der Entwicklungen aus den Sätzen 1.8 und 1.10 von den Abständen der Punkte abhängt. Für das Nahfeld können die Entwicklungen nicht verwendet werden, und die Einträge müssen direkt berechnet werden. Somit muss im Algorithmus für eine effiziente Realisierung beachtet werden, dass der Nahfeldanteil nicht zu groß wird.

Für eine einfachere graphische Darstellung wird der Algorithmus im Folgenden in der Ebene betrachtet. Der Algorithmus für den dreidimensionalen Fall ist ident. Zusätzlich wird von einer identen Menge an Auswertungs- und Ladungspunkten ausgegangen. Falls bei einer unterschiedlichen Menge an Auswertungs- und Ladungspunkten ein Unterschied im Algorithmus entsteht, wird dies explizit angesprochen. Ausgangspunkt des Algorithmus ist eine Menge an Punkten in der Ebene. Diese ist beispielhaft in Abbildung 2.1 dargestellt.

In Abschnitt 2.1 wird eine hierarchische Struktur über diese Punkte definiert. In Abschnitt 2.2 werden Kriterien zur Unterscheidung in Nah- und Fernfeld definiert. Dabei wird das zusätzliche Kriterium für den adaptiven Anteil hier erstmals in dieser allgemeinen Form aufgestellt. In Abschnitt 2.3 werden anhand der hierarchischen Struktur und den Kriterien aus den ersten beiden Abschnitten Listen konstruiert, mit denen die Interaktion im Multipolalgorithmus aus Abschnitt 2.5 bestimmt wird. Zusätzlich werden diese Listen und Interaktionen in Abschnitt 2.4 zur Veranschaulichung als Matrix

dargestellt.

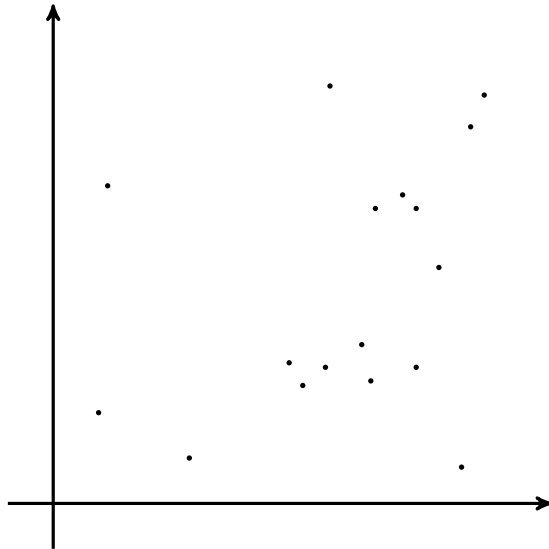


Abbildung 2.1: Punktmenge in der Ebene

2.1 Der Clusterbaum

Bevor der Algorithmus zur Erstellung der hierarchischen Struktur näher diskutiert wird, müssen zuerst einige im Weiteren benötigten Begriffe definiert werden:

1. Ein Cluster ist eine lokale Ansammlung von Punkten.
2. Der Clusterbaum ist eine Menge an Clustern, die in einer hierarchischen Struktur angeordnet sind.
3. Level bezeichnet die Tiefe im Clusterbaum.
4. Ein Cluster Ω_i^λ ist der i -te Cluster im Level λ .

Der Algorithmus startet mit der Punktmenge aus Abbildung 2.1. Zu dieser gegebenen Punktmenge wird ein Clusterbaum erzeugt. Die Ausnützung der hierarchischen Struktur des Clusterbaums und geschicktes Verschieben der Entwicklungen ermöglicht im Weiteren die effiziente Berechnung des Fernfeldanteils. Für eine unterschiedliche Menge an Auswertungs- und Ladungspunkten werden an dieser Stelle zwei Clusterbäume erzeugt. Für jede der beiden Mengen ein eigener. Zu Beginn des Algorithmus werden die maximale Anzahl an Punkten pro Cluster und die maximale Tiefe im Clusterbaum festgelegt. Die Erstellung des Clusterbaums erfolgt rekursiv beginnend mit einem Cluster Ω_1^0 auf Level 0, in welchem alle Punkte enthalten sind. Dieser Cluster wird als die

Wurzel des Clusterbaums bezeichnet. In Abbildung 2.2 ist diese Situation graphisch dargestellt.

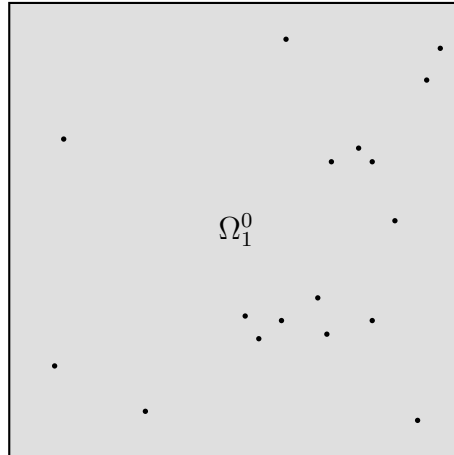


Abbildung 2.2: Clusterung auf Level 0

Falls die Anzahl der Punkte im Cluster die festgelegte maximale erwünschte Anzahl übersteigt und die maximale Tiefe im Clusterbaum noch nicht erreicht wurde, wird der Cluster Ω_1^0 in vier gleich große Cluster $\Omega_1^1, \dots, \Omega_4^1$ zerlegt. Im dreidimensionalen Fall sind die Cluster Würfel anstatt Quadraten. Bei einer Zerlegung des Clusters wird dieser somit in acht neue Cluster zerlegt. Die Cluster $\Omega_1^1, \dots, \Omega_4^1$ werden als die Söhne des Clusters Ω_1^0 bezeichnet. Umgekehrt ist Ω_1^0 der Vater der Cluster $\Omega_1^1, \dots, \Omega_4^1$. Die momentane Zerlegung ist in Abbildung 2.3 zu sehen.

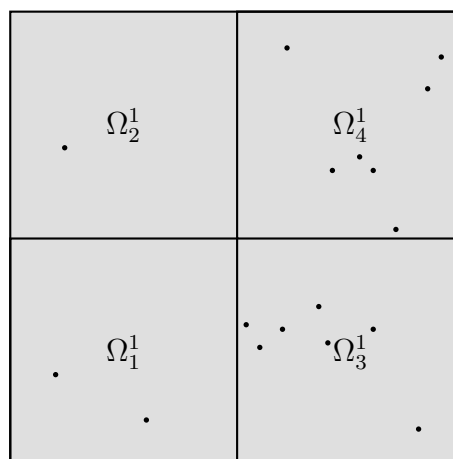


Abbildung 2.3: Clusterung auf Level 1

Im nächsten Schritt wird die Anzahl der Punkte in allen Clustern auf Level 1 betrachtet. Falls diese in einem Cluster die maximale Anzahl übersteigt, wird dieser Cluster wieder verfeinert. Falls durch diese Verfeinerung ein leerer Cluster entsteht, wird dieser verworfen. Abbildung 2.4 zeigt ein Beispiel für eine adaptive Clusterung.

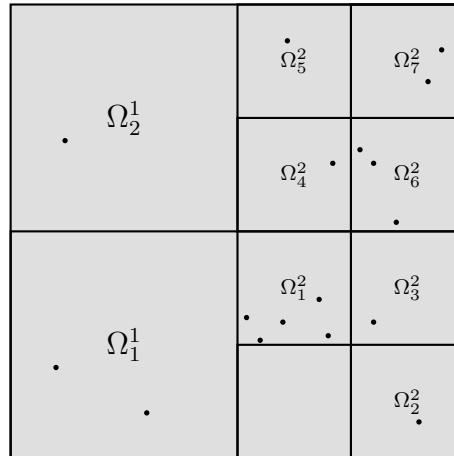


Abbildung 2.4: Clusterung auf Level 2

Die Verfeinerung der Cluster erfolgt solange, bis die Anzahl der Punkte in allen Clustern geringer als die festgelegte maximale Anzahl ist oder die gewählte maximale Tiefe im Clusterbaum erreicht wird. Für das Beispiel in Abbildung 2.4 wird bei einer gewählten maximalen Anzahl von drei Punkten pro Cluster und einer groß genug gewählten maximalen Baumtiefe nochmals verfeinert. Die daraus resultierende Clusterung ist in Abbildung 2.5 dargestellt.

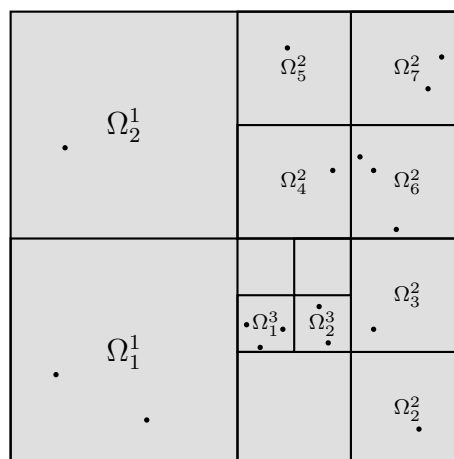


Abbildung 2.5: Clusterung auf Level 3

Die Wahl der Parameter der maximalen Baumtiefe sowie der maximalen Anzahl an Punkten pro Cluster sind bestimmend für die Struktur des Clusterbaums. Bei der Wahl der Parameter ist für eine effiziente Realisierung der Multipolmethode darauf zu achten, dass die Anzahl der Punkte pro Cluster nicht zu gering wird. Die Gestalt des zugehörigen Clusterbaumes ist in Abbildung 2.6 gegeben. Anhand der Darstellung des Clusterbaums und Abbildung 2.5 ist die Adaptivität in der Clusterung gut zu erkennen.

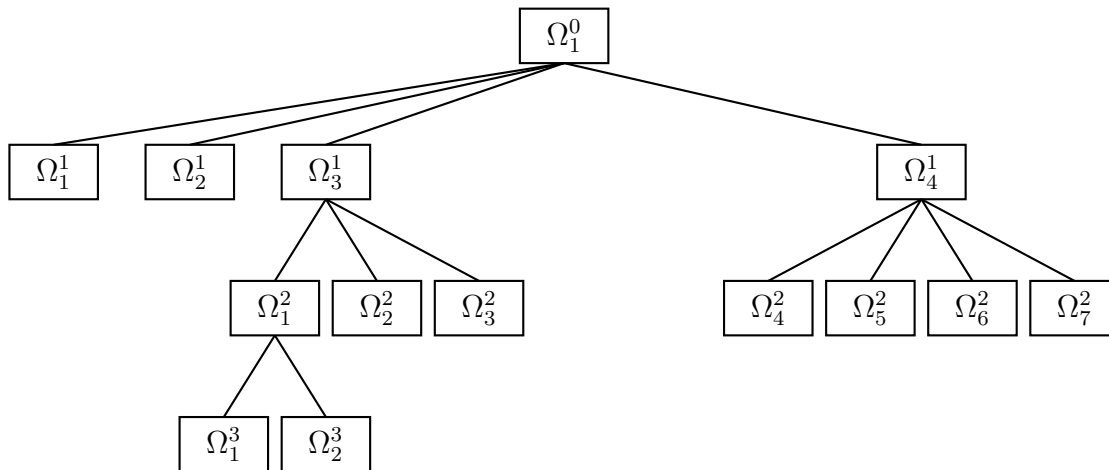


Abbildung 2.6: Clusterbaum

Der Algorithmus zum Erstellen des Clusterbaumes ist für den dreidimensionalen Fall im Wesentlichen ident. Der Unterschied liegt darin, dass die Cluster anstatt Quadraten Würfeln sind und somit bei jeder Verfeinerung ein Cluster in acht Söhne zerlegt wird.

2.2 Zulässigkeitskriterien

Anhand des Clusterbaums wird die Interaktion der Cluster miteinander im Multipolalgorithmus bestimmt. Für jeden Cluster des Clusterbaums wird ein Nahfeld und ein Fernfeld aufgestellt. Im Nahfeld kann die Multipolentwicklung in (1.5) nicht verwendet werden, und das Potential muss direkt berechnet werden.

Daran, wie im Folgenden in Fernfeld und Nahfeld unterschieden wird, unterscheidet sich die adaptive Version der schnellen Multipolmethode von der Standardmethode. In der Standardmethode hat jeder Cluster nur Cluster seines Levels im Fernfeld. Zwischen diesen Clustern wird die M2L Operation aus Satz 1.12 durchgeführt. Bei einer gleichmäßigen Verteilung der Punkte, und einer daraus resultierenden gleichmäßigen Verfeinerung der Cluster, genügt diese Interaktion auf demselben Level für einen geringen Nahfeldanteil.

Bei stark adaptiven Zerlegungen wie z.B. in Abbildung 2.5 genügt dies oft nicht mehr.

Bei adaptiven Zerlegungen können benachbarte Cluster unterschiedlich oft verfeinert werden. Da sich jedoch nur Cluster desselben Levels im Fernfeld befinden, hat dies einen größeren Nahfeldanteil zur Folge. Solche Zerlegungen entstehen z.B. bei adaptiven BEM-Rechnungen [15].

Um den Nahfeldanteil zu verringern, werden in der adaptiven Version der schnellen Multipolmethode die Operationen M2T und S2L aus den Sätzen 1.8 und 1.9 eingeführt. Diese beiden Operationen sind in den Abbildungen 2.19 und 2.18 dargestellt. Mit diesen zusätzlichen Operationen ist es möglich, das Fernfeld auf Cluster unterschiedlichen Levels zu erweitern. Dabei wird die Unterscheidung, ob zwischen Clustern unterschiedlichen Levels im Fernfeld eine S2L oder M2T Operation durchgeführt wird, aufgrund der unterschiedlichen Tiefe der Cluster im Baum getroffen. Vom Cluster, der tiefer im Baum ist, erfolgt eine M2T Operation zum Cluster im höheren Level. Umgekehrt erfolgt vom Cluster im höheren Level eine S2L Operation zum Cluster im tieferen Level. D.h. auch, dass falls von einem Cluster Ω_i^λ eine S2L Operation zu einem Cluster Ω_j^μ durchgeführt wird, von diesem Cluster Ω_j^μ eine M2T Operation zum Cluster Ω_i^λ angewandt wird. Daraus folgt, dass die zusätzlichen Operationen an der Symmetrie in der Approximation der Multipolmethode nichts ändern. Dies wird auch zusätzlich in [14] gezeigt. Mit der Erweiterung des Fernfelds auf Cluster unterschiedlichen Levels, erfolgt somit insbesondere bei adaptiven Ladungsverteilungen ein geringerer Nahfeldanteil.

Die zusätzlichen Operationen werden benötigt, da die Konvergenz des Fehlers bei der M2L Operation für Cluster verschiedenen Levels im Algorithmus nicht garantiert werden kann. Dies wurde in [14] detailliert ausgearbeitet.

Die Unterscheidung in Nah- und Fernfeld wird mit sogenannten Zulässigkeitskriterien getroffen. Im Folgenden ist C_i^λ das Zentrum des Cluster Ω_i^λ . Weiters ist $r_i^\lambda = \max_{x_i \in \Omega_i^\lambda} |x_i - C_i^\lambda|$ der Clusterradius und $c > 1$ ein Parameter zur Regulierung des Nahfeldanteils. Das folgende Kriterium beschreibt die Zulässigkeit zwischen zwei Clustern desselben Levels.

Definition 2.1. *Ein Cluster Ω_i^λ liegt im Fernfeld des Clusters Ω_j^λ desselben Levels, wenn das Zulässigkeitskriterium*

$$\text{dist}(C_i^\lambda, C_j^\lambda) > (c + 1) \max(r_i^\lambda, r_j^\lambda) \quad (2.2)$$

erfüllt ist.

In diesem Fall spricht man, die beiden Cluster sind zulässig. Im Multipolalgorithmus wird zwischen zwei Clustern, die mit diesem Kriterium zulässig sind, eine M2L Operation aus Satz 1.12 durchgeführt. Ein Beispiel für diese Situation, mit einem Auswertungspunkt und einem Ladungspunkt, ist in Abbildung 2.7 dargestellt.

In diesem Beispiel werden zuerst mit der Operation S2M aus Satz 1.7 die Koeffizienten der Multipolentwicklung der Quelle in y_k mit Zentrum in C_j^λ berechnet. Diese Multipolentwicklung wird mit der Operation M2L aus Satz 1.12 in eine lokale Entwicklung

mit Zentrum in C_i^λ transformiert und mit der Operation L2T aus Satz 1.10 in x ausgewertet.

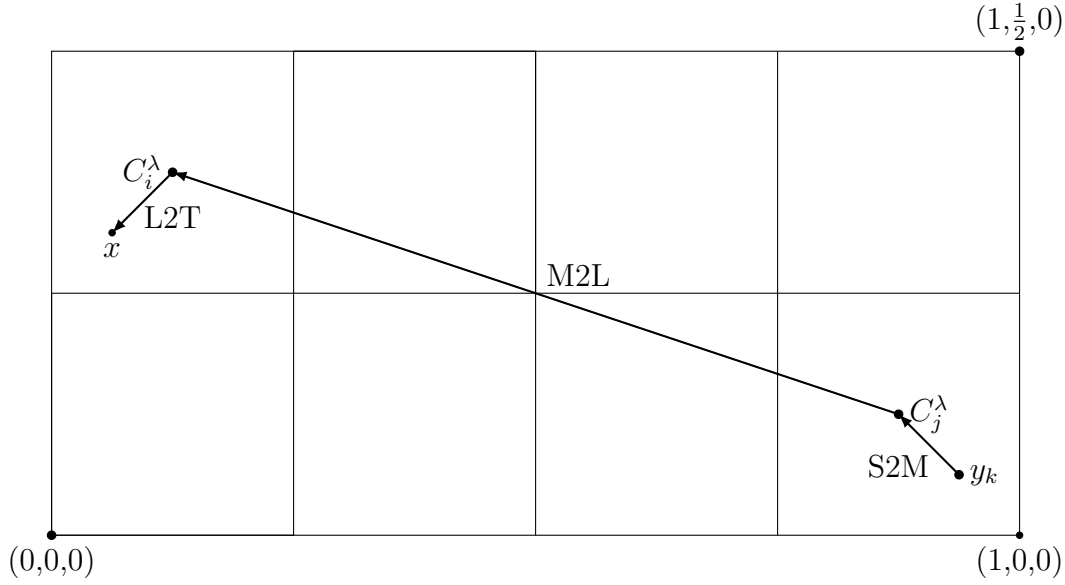


Abbildung 2.7: Multipoloperation für zwei Punkte x und y_k

Die Fehlerabschätzung (1.26) aus Satz 1.12 lässt sich mit dem Zulässigkeitskriterium in (2.2) weiter umformen. Dabei gelten für die Vektoren $\overrightarrow{C_i^\lambda y_k}$ aus Satz 1.12 mit dem Zulässigkeitskriterium in (2.2) folgende Abschätzung:

$$\begin{aligned} \min_k |\overrightarrow{C_i^\lambda y_k}| = w &\geq \text{dist}(C_i^\lambda, C_j^\lambda) - \max(r_i^\lambda, r_j^\lambda) \\ &\geq (c+1) \max(r_i^\lambda, r_j^\lambda) - \max(r_i^\lambda, r_j^\lambda) = cr \end{aligned}$$

mit $r = \max(r_i^\lambda, r_j^\lambda)$. Weiters gilt für alle Auswertungspunkte $|\overrightarrow{C_i^\lambda x}| \leq r$. Mit diesen beiden Abschätzungen folgt für den Fehler der Auswertung der lokalen Entwicklung in (1.26) weiter:

$$\begin{aligned} |\Psi(x) - L2T(x)| &\leq \frac{\sum_{i=1}^{\ell} |q_i|}{w - |\overrightarrow{C_i^\lambda x}|} \left(\frac{|\overrightarrow{C_i^\lambda x}|}{w} \right)^{p+1} \\ &\leq \frac{\sum_{i=1}^{\ell} |q_i|}{cr - r} \left(\frac{r}{cr} \right)^{p+1} \\ &= \frac{\sum_{i=1}^{\ell} |q_i|}{r(c-1)} \left(\frac{1}{c} \right)^{p+1}. \end{aligned} \quad (2.3)$$

An dieser weiteren Abschätzung des Fehlers ist zu erkennen, dass mit den Parametern c und p die Genauigkeit der Multipolmethode gesteuert wird. Höhere Werte für c und p haben aber auch einen erhöhten Rechenaufwand zur Folge.

In Tabelle 2.1 wird die Konvergenz des absoluten Fehlers des Beispiels aus Abbildung 2.7 für steigenden Polynomgrad gezeigt. Dabei hat der Auswertungspunkt x die Koordinaten $(\frac{1}{16}, \frac{5}{16}, 0)$. Die Quelle in y_k mit Einheitsladung hat die Koordinaten $(\frac{15}{16}, \frac{1}{16}, 0)$. Das Clusterzentrum C_i^λ befindet sich in $(\frac{1}{8}, \frac{3}{8}, 0)$ und das Zentrum des Clusters C_j^λ in $(\frac{7}{8}, \frac{1}{8}, 0)$. Zusätzlich ist die Fehlerabschätzung nach (2.3) angegeben. Dabei wurde der Parameter $c = 7.94$ so groß gewählt, dass die beiden Cluster gerade noch nach Kriterium (2.2) zulässig sind. In Tabelle 2.1 ist das Konvergenzverhalten des Fehlers der Entwicklung gut erkennbar. Dabei liegt der Fehler der Auswertung der Entwicklung auf allen Entwicklungsgraden deutlich unter dem theoretischen Fehler nach Abschätzung 2.3.

p	$ \Psi(x) - L2T(x) $	Fehlerabschätzung nach 2.3
1	0.004736	0.025858
2	0.000151	0.003256
3	2.2490e-05	0.000410
4	1.2169e-06	5.1659e-5
5	2.5027e-07	6.5061e-6
6	6.1868e-08	8.1941e-7
7	7.9474e-09	1.0320e-7
8	4.4300e-10	1.2997e-8
9	2.7202e-11	1.6369e-9
10	8.2616e-12	2.0616e-10
11	8.5687e-13	2.5965e-11
12	4.7295e-14	3.2702e-12

Tabelle 2.1: Konvergenz des Fehlers für die lokale Entwicklung aus der M2L Operation

Für Cluster unterschiedlichen Levels kann, aufgrund des großen Unterschiedes in der Größe der Clusterradien, das Zulässigkeitskriterium in (2.2) nicht verwendet werden. Daraus entsteht die Notwendigkeit eines weiteren Zulässigkeitskriterium für den adaptiven Fall. Dabei sollte sich idealerweise der Fehler der Auswertung der Entwicklung in ähnlicher Weise wie jener für die M2L Operation weiter abschätzen lassen. Dafür wird in dieser Arbeit folgendes Kriterium für zwei Cluster unterschiedlichen Levels definiert.

Definition 2.2. *Zwei Cluster Ω_i^λ und Ω_j^μ unterschiedlicher Level λ und μ , mit $\lambda > \mu$ sind zulässig, falls sie*

$$\text{dist}(C_i^\lambda, C_j^\mu) > dr_i^\lambda + r_j^\mu \quad (2.4)$$

erfüllen.

Dabei ist $d > 1$ ein Parameter zur Steuerung des Nahfeldanteils. Die Steuerung dieses Teils des Nahfelds erfolgt über den Radius jenes Clusters, der tiefer im Clusterbaum ist. Mit dem Zulässigkeitskriterium in (2.4) lässt sich die Fehlerabschätzung in (1.22) der Auswertung, der durch die S2L Operation entstandenen lokalen Entwicklung in Satz 1.10, weiter umformen. Dabei ist die Situation für die S2L Operation, für einen Auswertungs- und Ladungspunkt, in Abbildung 2.8 dargestellt.

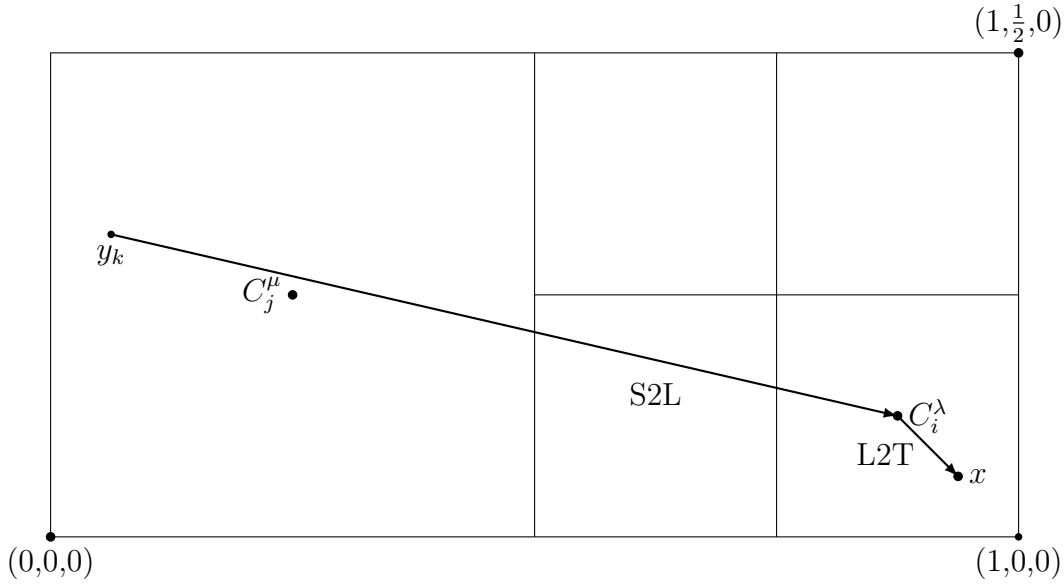


Abbildung 2.8: Darstellung der S2L Operation für zwei Punkte x und y_k

Für die Vektoren $\overrightarrow{C_i^\lambda y_k}$ gilt mit dem Zulässigkeitskriterium in (2.4) weiter:

$$\min_k |\overrightarrow{C_i^\lambda y_k}| = w \geq \text{dist}(C_i^\lambda, C_j^\mu) - r_j^\mu > dr_i^\lambda + r_j^\mu - r_j^\mu = dr_i^\lambda.$$

Ebenso gilt für den Auswertungspunkt $|\overrightarrow{C_i^\lambda x}| \leq r_i^\lambda$. Mit diesen Abschätzungen folgt für den Fehler in (1.22) aus Satz 1.10 weiter:

$$\begin{aligned} |\Psi(x) - L2T(x)| &\leq \frac{\sum_{i=1}^{\ell} |q_i|}{w - |x|} \left(\frac{|x|}{w} \right)^{p+1} \\ &\leq \frac{\sum_{i=1}^{\ell} |q_i|}{dr_i^\lambda - r_i^\lambda} \left(\frac{r_i^\lambda}{dr_i^\lambda} \right)^{p+1} \\ &= \frac{\sum_{i=1}^{\ell} |q_i|}{r_i^\lambda (d-1)} \left(\frac{1}{d} \right)^{p+1}. \end{aligned} \quad (2.5)$$

Die weitere Abschätzung des Fehlers in (1.18) für die Operation M2T, unter Verwendung des Zulässigkeitskriteriums in (2.4), erfolgt ident und liefert genau dieselbe Abschätzung.

Anhand dieser Abschätzung ist zu erkennen, dass mit den Parametern d und p wieder die Genauigkeit in der Multipolmethode gesteuert wird. Auch bei diesen Operationen folgt mit höheren Werten für d und p ein größerer Rechenaufwand. In Tabelle 2.2 ist die Konvergenz des Fehlers für das Beispiel aus Abbildung 2.8 dargestellt. Dabei werden in diesem Beispiel mit der S2L Operation aus Satz 1.9 die Koeffizienten der lokalen Entwicklung der Quelle y_k mit Zentrum in C_i^λ berechnet. Mit der Operation L2T aus Satz 1.10 wird diese Entwicklung in x ausgewertet. Zusätzlich zum absoluten Fehler der Punktauswertung ist wieder die theoretische Fehlerschranke aus (2.5) angegeben. Die Punkte x und y_k haben abermals die Koordinaten $(\frac{15}{16}, \frac{1}{16}, 0)$ und $(\frac{1}{16}, \frac{5}{16}, 0)$. Als Ladung im Punkt y_k wurde wieder die Einheitsladung gewählt. Das Clusterzentrum C_j^λ befindet sich in $(\frac{1}{8}, \frac{7}{8}, 0)$. Der Parameter $d = 6.7$ wurde so groß gewählt, dass die beiden Cluster gerade noch nach Kriterium (2.4) zulässig sind.

In Tabelle 2.2 ist die Konvergenz des Fehlers der Entwicklungen gut zu sehen. Dabei konvergiert auch hier der erhaltene Fehler deutlich schneller als der theoretische Fehler nach Abschätzung (2.5).

p	$ \Psi(x) - L2T(x) $	Fehlerabschätzung nach (2.5)
1	0.00743163	0.044216
2	0.000365548	0.006599
3	4.90674e-06	0.000984
4	4.06964e-06	0.000147
5	6.24444e-07	2.1942e-5
6	6.50642e-08	3.2749e-6
7	4.8304e-09	4.88794832e-7
8	1.70715e-10	7.29544525e-8
9	1.96592e-11	1.08887243e-8
10	5.11569e-12	1.62518272e-9
11	6.7879e-13	2.4256459e-10
12	6.43929e-14	3.620367e-11

Tabelle 2.2: Konvergenz des Fehlers für die lokale Entwicklung aus der S2L Operation

Anhand der beiden Umformungen in (2.3) und (2.5) folgt, dass die beiden Parameter c und d bestimmend für die Konvergenz des Fehlers sind. Weiters folgt anhand der beiden Kriterien, dass sich mittels den Parametern c und d der Nahfeldanteil steuern lässt. Bei der Wahl dieser Parameter ist somit darauf zu achten, dass die Balance zwischen idealem Nahfeldanteil und dem Fehler optimal ist.

Wichtig zu erwähnen ist, dass die Multipoloperationen immer am größten möglichen Level erfolgen. D.h., falls ein Cluster Ω_i^λ bereits im Fernfeld des Vaters eines Clusters

Ω_j^μ liegt, wird zwischen den Clustern Ω_i^λ und Ω_j^μ nicht mehr auf Zulässigkeit getestet. Die Interaktion erfolgt bereits zwischen Ω_i^λ und dem Vater des Clusters Ω_j^μ . Für eine effiziente Realisierung des Multipolalgorithmus werden, unter Verwendung des Clusterbaums und der beiden Zulässigkeitskriterien, für jeden Cluster die Listen L_{NF} , L_{M2L} , L_{S2L} und L_{M2T} erstellt, in denen die Interaktion mit den jeweiligen anderen Clustern gespeichert ist.

2.3 Aufstellen der Operationslisten

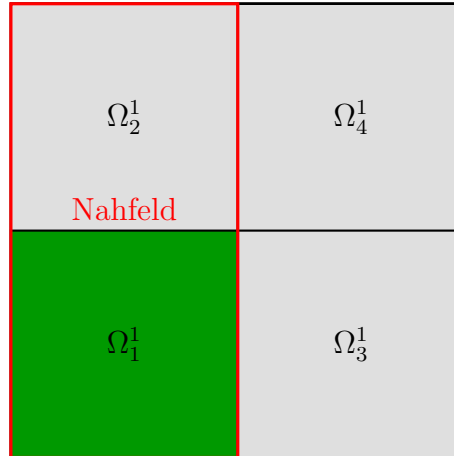
Im Folgenden wird anhand der beispielhaften Verfeinerung der Cluster aus Abschnitt 2.1 der Algorithmus zum Erstellen dieser Listen beschrieben. Dabei wurden die Parameter c und d zur Veranschaulichung so gewählt, dass zwei Cluster zulässig sind, sobald sie sich nicht berühren.

Der Algorithmus startet bei der Wurzel Ω_1^0 des Clusterbaumes auf Level 0. Auf diesem Level existiert nur ein Cluster (Abb. 2.2). Dieser wird mit Kriterium (2.2) mit sich selbst getestet. Er kann nicht zulässig sein und wird somit der Liste $L_{NF}(\Omega_1^0)$ hinzugefügt. Somit sind alle Cluster auf Level 0 abgearbeitet, und es werden im nächsten Schritt die Söhne des Clusters Ω_1^0 betrachtet, siehe dazu Abbildung 2.3. Da an dieser Stelle eine Unterscheidung erfolgt, ob die Cluster Blätter sind, werden hier respektiv der Cluster Ω_1^1 für die Blätter sowie der Cluster Ω_3^1 für die inneren Knoten betrachtet. Für den Cluster Ω_1^1 werden alle Blätter im Nahfeld des Vaters, die am selben oder einem höheren Level wie der Vater sind, auf Zulässigkeit mittels Kriterium (2.4) betrachtet. Bei Zulässigkeit werden sie der Liste $L_{S2L}(\Omega_1^1)$ hinzugefügt, andernfalls der Liste $L_{NF}(\Omega_1^1)$. Im betrachteten Beispiel bleiben die Listen unverändert, da im Nahfeld des Vaters keine Blätter des selben Levels sind.

Als Nächstes werden die Söhne jener Cluster im Nahfeld des Vaters, die am Level des Vaters sind, auf Zulässigkeit mittels (2.2) getestet, d.h., für den Cluster Ω_1^1 die Cluster Ω_1^1 bis Ω_4^1 (Abb. 2.3). Falls sie zulässig sind, werden sie der Liste $L_{M2L}(\Omega_1^1)$ hinzugefügt. Falls sie nicht zulässig und Blätter sind, sind sie in der Liste $L_{NF}(\Omega_1^1)$. Die nicht zulässigen Cluster, die keine Blätter sind, werden weiter betrachtet. In Abbildung 2.9 ist diese Situation dargestellt. Für die momentanen Interaktionslisten des Clusters Ω_1^1 folgt:

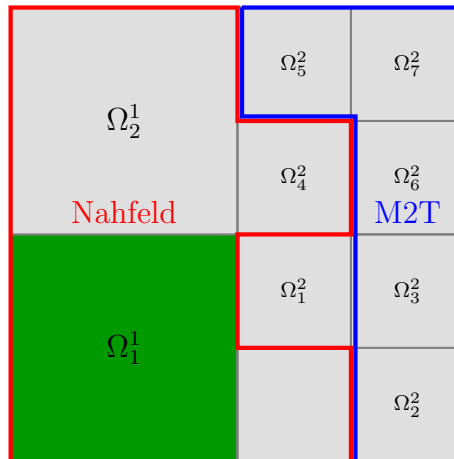
1. $L_{NF}(\Omega_1^1): \{\Omega_1^1, \Omega_2^1\}$
2. $L_{M2L}(\Omega_1^1): \emptyset$
3. $L_{M2T}(\Omega_1^1): \emptyset$
4. $L_{S2L}(\Omega_1^1): \emptyset$

Diese Listen werden im weiteren Verlauf noch ergänzt. Im nächsten Schritt werden die Söhne der Cluster Ω_3^1 und Ω_4^1 mit dem Kriterium in (2.4) auf Zulässigkeit mit Cluster Ω_1^1 getestet. Die Zerlegung der Cluster ist in Abbildung 2.4 dargestellt. Falls

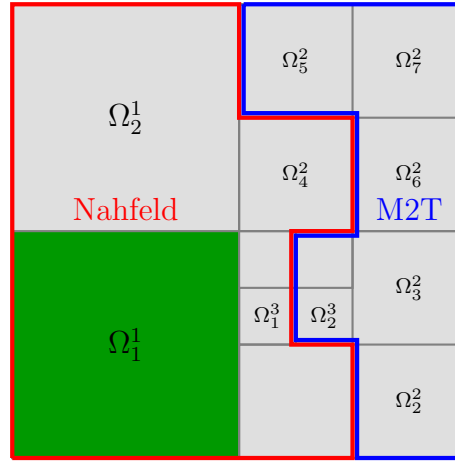
Abbildung 2.9: Konstruktion der Interaktionslisten für Ω_1^1

sie zulässig sind, werden sie der Liste $L_{M2T}(\Omega_1^1)$ hinzugefügt. Falls sie nicht zulässig und Blätter sind, sind sie in $L_{NF}(\Omega_1^1)$. In Abbildung 2.10 ist diese Situation zu sehen. Die Interaktionslisten haben nun zwischenzeitlich folgende Gestalt:

1. $L_{NF}(\Omega_1^1): \{\Omega_1^1, \Omega_2^1, \Omega_4^1\}$
2. $L_{M2L}(\Omega_1^1): \emptyset$
3. $L_{M2T}(\Omega_1^1): \{\Omega_2^2, \Omega_3^2, \Omega_5^2, \Omega_6^2, \Omega_7^2\}$
4. $L_{S2L}(\Omega_1^1): \emptyset$

Abbildung 2.10: Konstruktion der Interaktionslisten für Ω_1^1

Die Cluster, die nicht zulässig und keine Blätter sind, werden abermals weiter am nächsten Level betrachtet. Das endgültigen Bild für den Cluster Ω_1^1 ist in Abbildung

Abbildung 2.11: Konstruktion der Interaktionslisten für Ω_1^1

2.11 dargestellt. Dabei haben die endgültigen Interaktionslisten für den Cluster Ω_1^1 folgende Gestalt:

1. $L_{NF}(\Omega_1^1): \{\Omega_1^1, \Omega_2^1, \Omega_4^2, \Omega_1^3\}$
2. $L_{M2L}(\Omega_1^1): \emptyset$
3. $L_{M2T}(\Omega_1^1): \{\Omega_2^2, \Omega_3^2, \Omega_5^2, \Omega_6^2, \Omega_7^2, \Omega_2^3\}$
4. $L_{S2L}(\Omega_1^1): \emptyset$

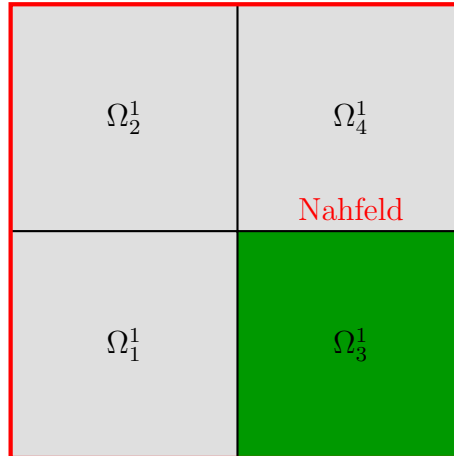
Für den Cluster Ω_3^1 werden ebenfalls alle Blätter im Nahfeld des Vaters, die am selben oder einem höheren Level wie der Vater sind, auf Zulässigkeit mit Kriterium (2.4) getestet und den Listen $L_{NF}(\Omega_3^1)$ sowie $L_{S2L}(\Omega_3^1)$ zugezählt. Ebenso werden im nächsten Schritt die Söhne der Cluster im Nahfeld des Vaters mit (2.2) auf Zulässigkeit getestet und dann den Listen $L_{NF}(\Omega_3^1)$ und $L_{M2L}(\Omega_3^1)$ beigefügt. Anders als beim Blatt Ω_1^1 erfolgt hier kein einseitiger Abstieg bei den Clustern im Nahfeld. Die Interaktion mit diesen wird auf die Söhne von Ω_3^1 verschoben. Die Interaktionslisten für den Cluster Ω_3^1 haben folgende Gestalt:

1. $L_{NF}(\Omega_3^1): \{\Omega_1^1, \Omega_2^1, \Omega_3^1, \Omega_4^1\}$
2. $L_{M2L}(\Omega_3^1): \emptyset$
3. $L_{M2T}(\Omega_3^1): \emptyset$
4. $L_{S2L}(\Omega_3^1): \emptyset$

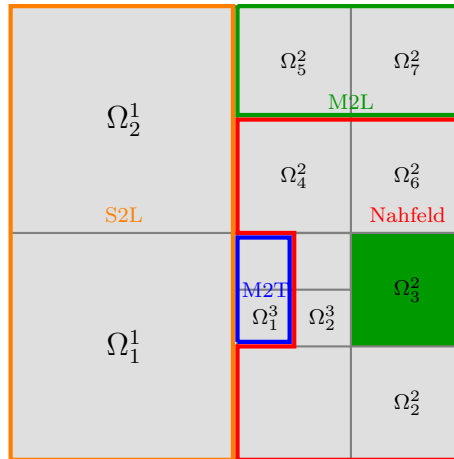
In Abbildung 2.12 ist die Situation graphisch dargestellt.

Sobald für die Cluster auf Level 1 die Interaktionslisten erstellt sind, folgt der Abstieg auf Level 2. Die Erstellung der Interaktionslisten der Cluster auf Level 2 erfolgt ident zu Level 1. Für dieses Level wird beispielhaft der Cluster Ω_3^2 betrachtet.

Abermals werden die Blätter im Nahfeld des Vaters, die am selben oder einem höheren Level wie der Vaters sind, auf Zulässigkeit mit (2.4) getestet. Die Darstellung der

Abbildung 2.12: Konstruktion der Interaktionslisten für Ω_3^1

Cluster ist in Abbildung 2.4 zu sehen. In diesem Fall sind beide Cluster Ω_1^1 und Ω_2^1 zulässig und werden der Liste $L_{S2L}(\Omega_3^2)$ hinzugefügt. Nun wird wieder für die Söhne jener Cluster im Nahfeld des Vaters, die am Level des Vaters sind, die Zulässigkeit mit (2.2) getestet und die Cluster in die jeweiligen Listen geschrieben. Da hier der Cluster Ω_3^2 ein Blatt ist, erfolgt wieder der einseitige Abstieg im Cluster Ω_2^2 . Für den Cluster Ω_3^2 ist die endgültige Darstellung in Abbildung 2.13 zu finden.

Abbildung 2.13: Konstruktion der Interaktionslisten für Ω_3^2

Für die Interaktionslisten des Clusters Ω_3^2 folgt:

1. $L_{NF}(\Omega_4^2): \{\Omega_2^2, \Omega_4^2, \Omega_6^2, \Omega_2^3\}$
2. $L_{M2L}(\Omega_4^2): \{\Omega_5^2, \Omega_7^2\}$
3. $L_{M2T}(\Omega_4^2): \{\Omega_1^3\}$

4. $L_{S2L}(\Omega_4^2): \{\Omega_1^1, \Omega_2^1\}$

Sobald alle Cluster auf Level 2 abgearbeitet sind, folgt der Abstieg auf Level 3. Wiederum erfolgt mit denselben Kriterien die Einteilung in die verschiedenen Listen. Dieser Vorgang wird solange wiederholt, bis der feinste Level erreicht wird und für alle Cluster im Baum die Interaktionslisten erstellt sind.

Aufgrund der Konstruktion folgt, dass für einen Cluster b in der Liste $L_{S2L}(c)$ eines Cluster c , sich wiederum c in der Liste $L_{M2T}(b)$ befindet. Somit bleibt auch mit dem zusätzlichen einseitigen Abstieg die Symmetrie der Multipolmethode erhalten. Anhand dieses Algorithmus ist ebenfalls zu erkennen, dass die Interaktion der Cluster immer am größten möglichen Level erfolgt. Als Kandidaten für die jeweiligen Listen werden nur die Cluster im Nahfeld des Vaters getestet, die restlichen Cluster, die in den Listen L_{M2L} , L_{S2L} oder L_{M2T} des Vaters sind, werden nicht mehr behandelt, da deren Interaktion bereits am Level des Vaters abläuft. Zusammenfassend haben die vier Listen für jeden Cluster folgende Gestalt:

1. L_{NF} : Alle Cluster desselben Levels, die laut Kriterium (2.2) nicht zulässig sind, und alle Blätter eines höheren Levels, welche laut (2.4) nicht zulässig sind. Für Blätter sind zusätzlich alle Blätter eines feineren Levels, welche mit Kriterium (2.4) nicht zulässig sind, ebenfalls in dieser Liste.
2. L_{M2L} : Alle Cluster desselben Levels, die laut Kriterium (2.2) zulässig sind.
3. L_{M2T} : Alle Cluster eines tieferen Levels, deren Vater nicht in L_{M2L} ist und die Kriterium (2.4) erfüllen. Diese Liste ist leer, falls der Cluster kein Blatt ist.
4. L_{S2L} : Alle Blätter im Nahfeld des Vaters, die am selben oder einem höheren Level wie der Vater sind, die mit Kriterium (2.4) zulässig sind.

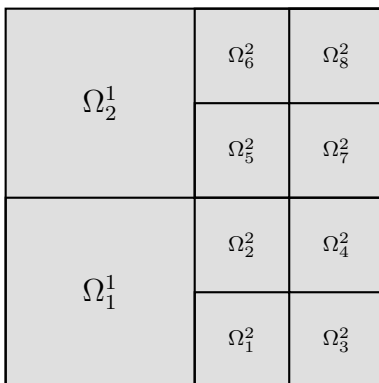
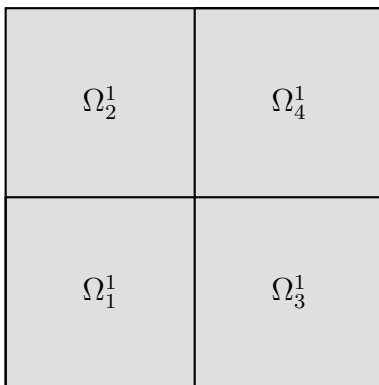
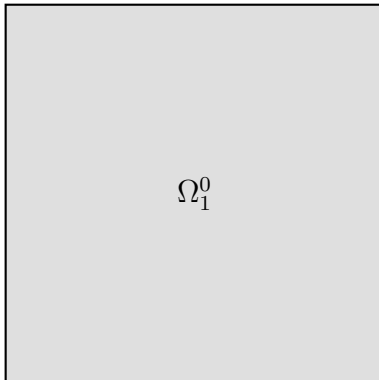
Für den Fall unterschiedlicher Mengen an Auswertungspunkten und Quellpunkten werden zwei Clusterbäume erstellt und deren Cluster gegeneinander auf Zulässigkeit getestet. Der Algorithmus dafür verläuft ident wie mit einem Clusterbaum.

2.4 Darstellung der Operationen als Matrix

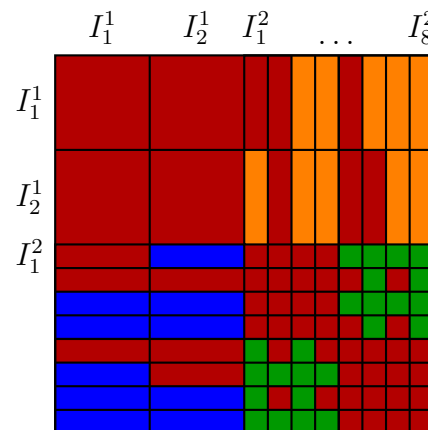
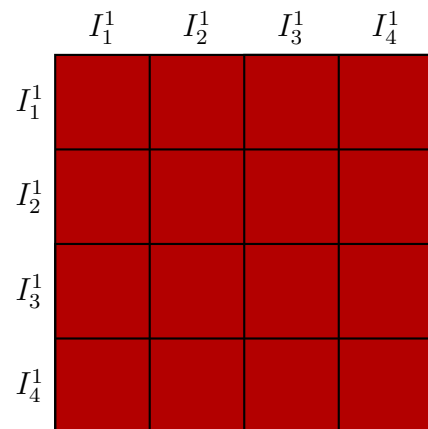
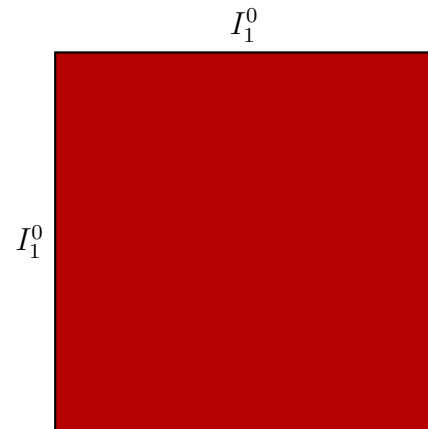
In diesem Abschnitt werden die Interaktionen zwischen den Clustern noch einmal zusätzlich graphisch verdeutlicht. Das linke Bild in den folgenden Abbildungen zeigt jeweils die Clusterung auf verschiedenen Level. Dazu sind jeweils die Operationen der Cluster miteinander in Form einer Matrix dargestellt.

Dabei werden die Parameter c und d aus Veranschaulichung wiederum so gewählt, dass zwei Cluster zulässig sind, solange sie sich nicht berühren. Zusätzlich sind die Auswertungs- und Ladungspunkte ident. Dabei wird mit I_i^λ die Indexmenge der Punkte des Clusters Ω_i^λ bezeichnet. Die Farbe Rot bezeichnet das Nahfeld und Grün eine M2L Operation. Blau und Orange stehen für eine S2L sowie M2T Operation.

geometrische Clusterung



Darstellung der Operationen als Matrix



Anhand dieser Abbildungen ist erkennbar, dass der Nahfeldanteil bei zusätzlicher Verfeinerung sinkt. In den Darstellungen ist ebenfalls die Symmetrie der M2T und S2L Operationen zu sehen. Zusammenfassend hängt die Interaktion der Cluster miteinander von mehreren Faktoren ab. Die ersten beiden sind die maximale Baumtiefe und die maximale Anzahl an Punkten pro Cluster, welche bestimmend für die Struktur des Clusterbaumes sind. Zusätzlich die Parameter c und d aus den Zulässigkeitskriterien,

mit welchen das Nahfeld reguliert wird.

2.5 Der Multipolalgorithmus

Für den Multipolalgorithmus werden die hierarchische Clusterung aus Abschnitt 2.1 und die in Abschnitt 2.3 erzeugten Interaktionslisten benötigt. Diese müssen im Vorhinein erstellt werden. Der im Folgenden definierte Algorithmus orientiert sich an den Arbeiten [3, 4, 9, 16, 17].

Der Multipolalgorithmus besteht aus zwei Teilen. Dem aufsteigenden Pfad, in welchem in allen Clustern eine Multipolentwicklung berechnet wird, sowie dem absteigenden Pfad, in welchem anhand der vorher erstellten Listen die Interaktion der Cluster miteinander durchgeführt wird. Im Folgenden wird der maximale Level im Clusterbaum mit L bezeichnet.

2.5.1 Der aufsteigende Pfad

Schritt 1

Der aufsteigende Pfad startet am feinsten Level L bzw. den Blättern des Baums. Levelweise werden für alle Blätter die Koeffizienten M_n^m der Multipolentwicklung der Punkte im Cluster mit der Operation S2M nach Satz 1.7 berechnet. Diese Operation ist in Abbildung 2.14 dargestellt.

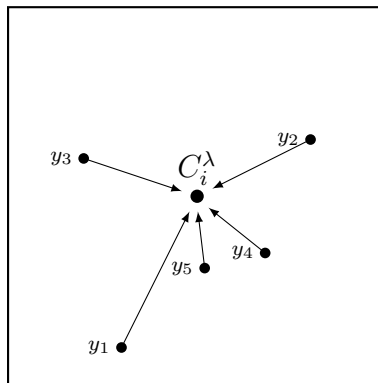


Abbildung 2.14: S2M in den Blättern

Schritt 2

Im nächsten Schritt werden, beginnend bei Level $L-1$ aufsteigend, die Koeffizienten der Multipolentwicklung der Väter als Summe der mit der Operation M2M in Satz 1.11 verschobenen Multipolentwicklungen der Söhne berechnet. Siehe dazu Abbildung 2.15.

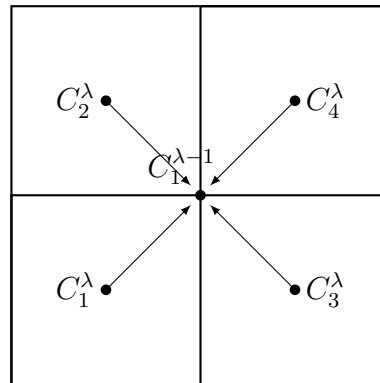


Abbildung 2.15: M2M von den Söhnen auf den Vater

2.5.2 Der absteigende Pfad

Der absteigende Pfad startet am Level 0. Hier werden, levelweise absteigend, für alle Cluster jedes Levels die Interaktionen der Cluster miteinander berechnet. An dieser Stelle werden die im vorhinein erstellten Listen aus Abschnitt 2.3 verwendet. Für jeden Cluster Ω_i^λ werden folgende Schritte durchgeführt:

Schritt 3

Falls der Cluster einen Vater hat, wird der Anteil der lokalen Entwicklung vom Vater auf den Sohn mit der Operation L2L aus Satz 1.13 gemäß Abbildung 2.16 berechnet.

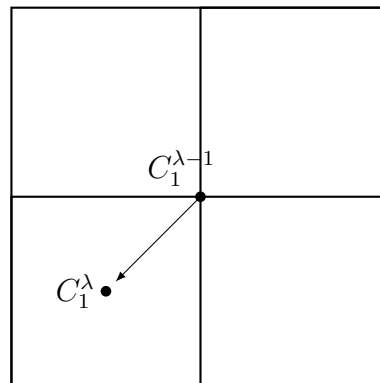


Abbildung 2.16: L2L vom Vater auf den Sohn

Schritt 4

Für alle Cluster Ω_j^λ der Liste $L_{M2L}(\Omega_i^\lambda)$ wird mit der Operation M2L aus Satz 1.12 die Multipolentwicklung des Clusters Ω_j^λ mit Zentrum in C_j^λ in eine lokale Entwicklung des Cluster Ω_i^λ mit Zentrum in C_i^λ transformiert und aufsummiert. Diese Operation ist in Abbildung 2.17 zu finden.

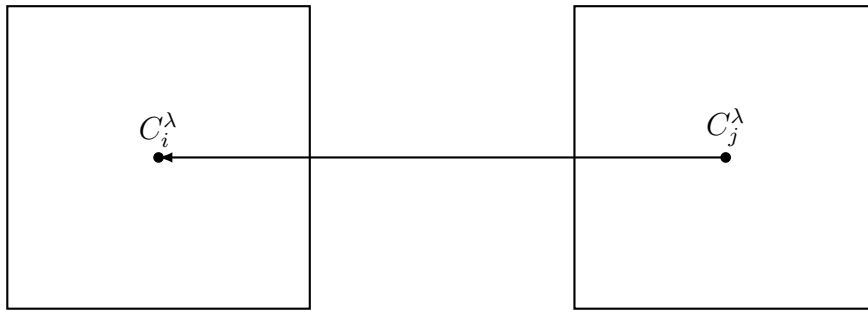


Abbildung 2.17: M2L

Schritt 5

Das Potential der Punkte in den Clustern Ω_j^μ in $L_{S2L}(\Omega_i^\lambda)$ wird mit der Operation S2L aus Satz 1.9 der lokalen Entwicklung mit Zentrum in C_i^λ nach Abbildung 2.18 aufaddiert.

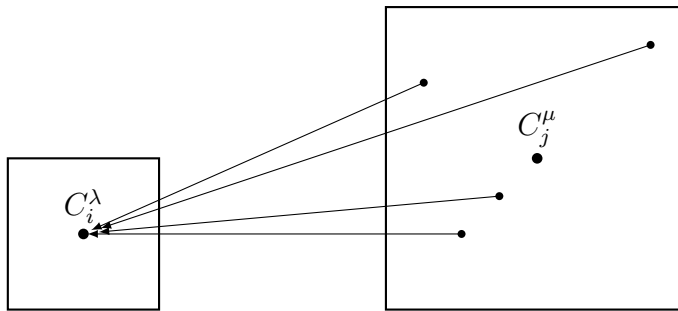


Abbildung 2.18: S2L

Schritt 6

Für alle Cluster Ω_j^μ in $L_{M2T}(\Omega_i^\lambda)$ wird die Multipolentwicklung des Clusters Ω_j^μ mit der Operation M2T aus Satz 1.8 direkt in den Punkten x_i von Ω_i^λ ausgewertet und dem Potential $\Phi_{Fernfeld}(x_i)$ hinzugefügt. Diese Operation ist in Abbildung 2.19 dargestellt.

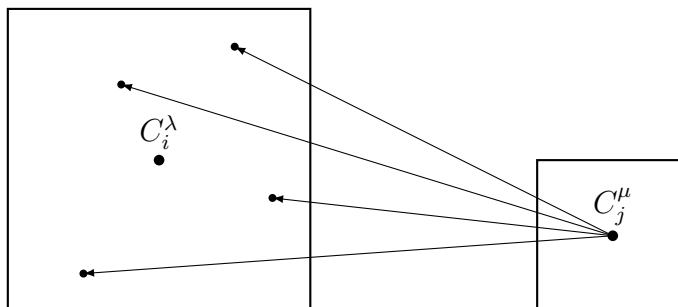


Abbildung 2.19: M2T

Schritt 7

Für alle Blätter wird die lokale Entwicklung in den Clusterpunkten x_i mit der Operation L2T aus Satz 1.10 ausgewertet und auf das Potential $\Phi_{\text{Fernfeld}}(x_i)$ aufaddiert.

Schritt 8

Für alle Blätter Ω_i^λ und für alle Cluster in der jeweiligen Liste $L_{NF}(\Omega_i^\lambda)$ wird das Potential zwischen den Clusterpunkten direkt berechnet und in $\Phi_{\text{Nahfeld}}(x_i)$ der Auswertungspunkte x_i aufsummiert. Das Potential in den Punkten x_i setzt sich dann, wie in (2.1) eingeführt, aus dem Nah- und Fernfeldanteil zusammen.

Der Aufwand für die direkte Berechnung des Potentials zwischen n Ladungspunkten und n Auswertungspunkten ist n^2 . Die Schwierigkeit in der Aufwandsabschätzung der adaptiven schnellen Multipolmethode, insbesondere bei stark adaptiven Verteilungen, liegt in der Bestimmung der Anzahl der S2L und M2T Operationen und der Abschätzung des Nahfeldanteils für die allgemeinen Kriterien in (2.2) und (2.4). Diese Anzahl hängt von der Clusterung und damit auch von der Verteilung der Punktladungen ab. Zusätzlich wird diese Anzahl von den Parametern c und d aus den Zulässigkeitskriterien in (2.2) und (2.4) beeinflusst. Es ist somit nicht ohne weiteres möglich, eine allgemeine Abschätzung der Anzahl dieser Operationen, die für beliebige Punktverteilungen gilt, herzuleiten. Für das einfache Nahfeld bestehend aus angrenzenden Nachbarn und unter gewissen Einschränkungen an die Verteilung der Punktladungen zeigt die adaptive schnelle Multipolmethode lineare Konvergenz [4, 16].

3 Numerische Beispiele

In diesem Kapitel werden die Ergebnisse aus den ersten beiden Kapiteln durch numerische Tests weiter untersucht. Dafür wurden die Konstruktion des Clusterbaums, die Erstellung der Listen und der Multipolalgorithmus in C/C++ implementiert. Eine kurze Beschreibung der Implementierung erfolgt in Abschnitt 3.4.

In den numerischen Tests wird die direkte Berechnung des Potentials einer Ladungsverteilung mit der Berechnung durch die Multipolmethode verglichen. In diesen Beispielen wird insbesondere auch der Unterschied zwischen der adaptiven schnellen Multipolmethode und der Standardversion der schnellen Multipolmethode betrachtet. Dafür wird vor allem das Verhalten des Fehlers, bei steigendem Polynomgrad der Entwicklungen, sowie die Differenz in den Rechenzeiten untersucht. Als Testverteilungen werden in Abschnitt 3.1 zufällig erzeugte Punktemengen verwendet. Zusätzlich werden in Abschnitt 3.2 Verteilungen untersucht, die aus adaptiven Diskretisierungen in der BEM [15] entstehen. Als abschließendes Beispiel werden in Abschnitt 3.3 die Unterschiede anhand einer Gleichverteilung untersucht. Für diese Tests werden zwei verschiedene Fehler untersucht. Der relative maximale Fehler

$$E_{max} = \max_i \left| \frac{\Phi(x_i) - \tilde{\Phi}(x_i)}{\Phi(x_i)} \right| \quad (3.1)$$

und die ℓ_2 -Norm des Fehlervektors geteilt durch die ℓ_2 -Norm der exakten Auswertung

$$E = \left(\frac{\sum_{i=1}^N |\Phi(x_i) - \tilde{\Phi}(x_i)|^2}{\sum_{i=1}^N |\Phi(x_i)|^2} \right)^{\frac{1}{2}}. \quad (3.2)$$

3.1 Testrechnungen für zufällig erzeugte Punktmengen

Für das erste Beispiel wird eine zufällige Verteilung von Ladungen erstellt. Dafür wird, beginnend bei einem Cluster mit Kantenlänge eins, für jeden Clustur anhand eines Zufallsparameters entschieden, ob der Cluster verfeinert oder mit einer Menge an Punkten gefüllt wird. Dieses zufällige Verfeinern ermöglicht uns die Konstruktion einer Zerlegung, in welcher der Anteil der adaptiven Operationen M2T und S2L nicht

zu gering ausfällt. Die auf diese Art für diesen Test konstruierte Verteilung besteht aus 45780 Punkten.

Vor dem Test des Multipolalgorithmus muss aus dieser Punktmenge ein Clusterbaum erzeugt werden. Für die Konstruktion des Clusterbaums wurde die maximale Anzahl an Punkten pro Cluster mit 20 festgelegt. Die maximale Baumtiefe wurde so groß gewählt, dass die Struktur des Clusterbaumes nur durch die maximale gewünschte Anzahl an Punkten pro Cluster bestimmt wird.

Mit dieser Verteilung wird die Konvergenz des Fehlers bei steigendem Polynomgrad der Entwicklungen untersucht. Die Abhängigkeit des Fehlers von den Parametern c und d aus den Zulässigkeitsbedingungen wird in den Fehlerabschätzungen (2.3) und (2.5) theoretisch gezeigt. Zusätzlich wird diese Abhängigkeit mit unterschiedlicher Wahl von c und d in den Experimenten untersucht. Dafür werden zwei Serien an Tests betrachtet. In diesen wird jeweils einer der Parameter c und d so groß gewählt, dass im Algorithmus entweder nur die M2L Operationen oder nur die S2L bzw. M2T Operationen ausgeführt werden. Somit lässt sich vergleichen, wie die Fehler aus der M2L sowie den S2L und M2T Operationen zum Gesamtfehler beitragen. Dies ermöglicht eine spätere Abstimmung der Parameter aufeinander. In Tabelle 3.1 ist der relative maximale Fehler E_{max} für die verschiedenen Entwicklungsgrade p abgebildet. Dabei wird der Parameter d auf 500 gesetzt, die M2T und S2L Operationen werden im Algorithmus daher nicht verwendet.

p	c=2	c=3	c =5
1	0.015123	0.008535	0.004411
2	0.002816	0.001098	0.000342
3	0.000739	0.000183	2.8032e-5
4	0.000168	2.6257e-5	3.1368e-6
5	3.2481e-5	7.1949e-6	2.7224e-7
6	7.8269e-6	2.7363e-6	4.7970e-8
7	3.0868e-6	3.3969e-7	7.3661e-9
8	1.0271e-6	8.8498e-8	7.4652e-10
9	4.0170e-7	1.2495e-8	1.6202e-10
10	1.4046e-7	3.0681e-9	2.6134e-11
11	3.3701e-8	7.6621e-10	3.1094e-12
12	1.5817e-8	1.7405e-10	5.4010e-13

Tabelle 3.1: Konvergenz des Fehlers E_{max} für den M2L Anteil

In Tabelle 3.1 ist klar zu erkennen, dass bei größerem c der Fehler, wie aus Abschätzung (2.3) erwartet, schneller konvergiert. Die Konvergenz des Fehlers ist zusätzlich in Abbildung 3.1 dargestellt. In dieser Abbildung ist besonders gut zu erkennen, wie sich die Fehler für unterschiedliche c entwickelt. Das Konvergenzverhalten $\frac{1}{c}$ aus der Fehlerabschätzung in (2.3) wird immer erreicht, meistens ist es sogar deutlich besser.

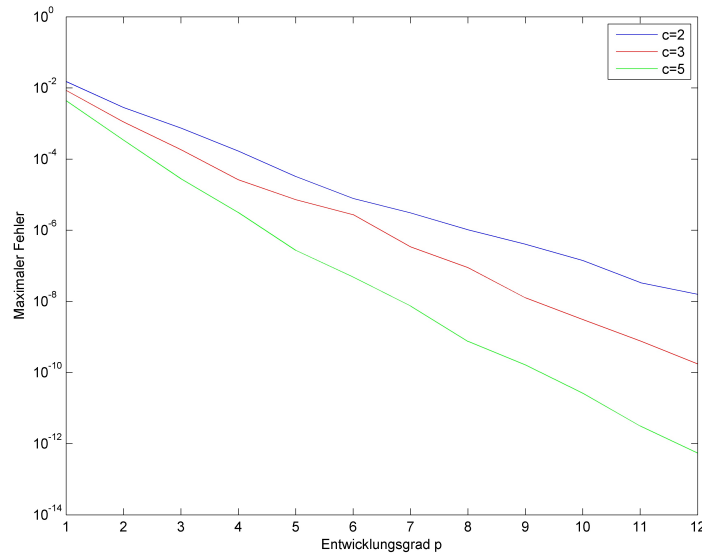


Abbildung 3.1: M2L Anteil des Fehlers E_{max} bei unterschiedlichem Parameter c

Ebenso wird in Tabelle 3.2 der maximale Fehler E_{max} bei unterschiedlicher Wahl von d abgebildet. Für diese Testreihe wird c auf 500 gesetzt. Somit werden im Multipolalgorithmus nur die M2T und S2L Operationen durchgeführt.

p	d=2	d=3	d=5
1	0.000973	0.000757	0.000607
2	0.000137	7.8720e-5	4.4808e-5
3	5.0396e-5	2.3106e-5	5.2591e-6
4	9.9738e-6	6.0764e-6	4.5531e-7
5	4.4063e-6	1.7249e-6	5.2934e-8
6	1.1112e-6	5.4595e-7	1.3058e-8
7	4.3145e-7	1.2516e-7	2.3789e-9
8	1.5946e-7	4.5478e-8	3.3531e-10
9	5.7702e-8	1.1748e-8	5.3424e-11
10	2.0191e-8	3.7964e-9	1.0838e-11
11	8.6559e-9	1.1379e-9	1.82727e-12
12	2.2962e-9	3.30138e-10	3.9058e-13

Tabelle 3.2: Konvergenz des Fehlers E_{max} für den Anteil von M2T und S2L

Anhand dieses Tests ist zu sehen, dass auch für den Parameter d die Konvergenz des Fehlers mit der Fehlerabschätzung in (2.5) übereinstimmt. Auch hier wird wieder zusätzlich die Konvergenz in Abbildung 3.2 dargestellt.

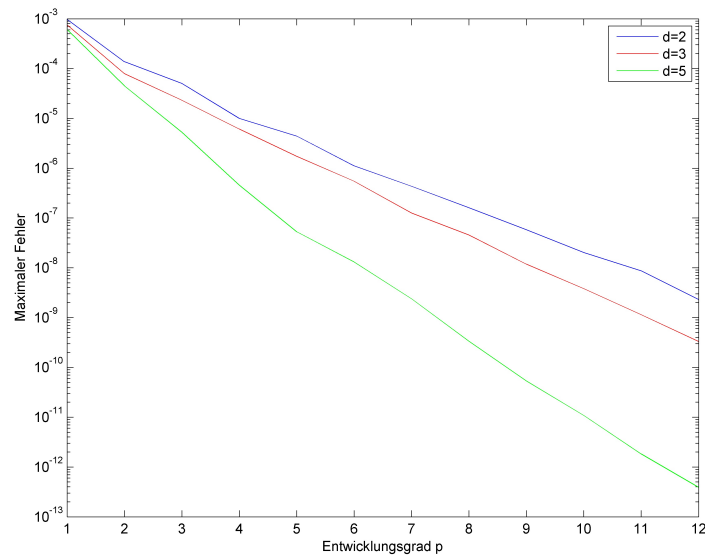


Abbildung 3.2: Anteil der M2T und S2L Operation am Fehler E_{max} für unterschiedliche Parameter d

Beim Vergleich der Fehler, bei identen Parametern c und d , ist zu erkennen, dass der Fehler, der durch die Operationen M2T und S2L entsteht, bei gleichem Polynomgrad meist geringer ist. Dies ist in den Abbildungen 3.3 und 3.4 gut zu erkennen.

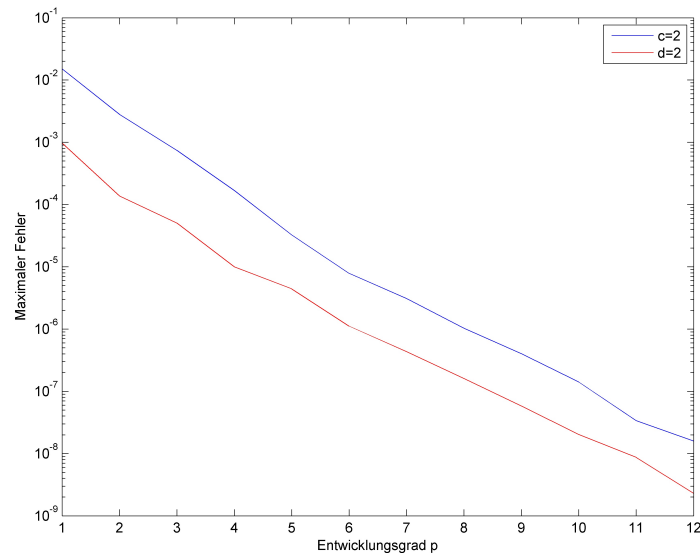


Abbildung 3.3: Vergleich des Fehleranteils von E_{max} bei identen Parametern c und d

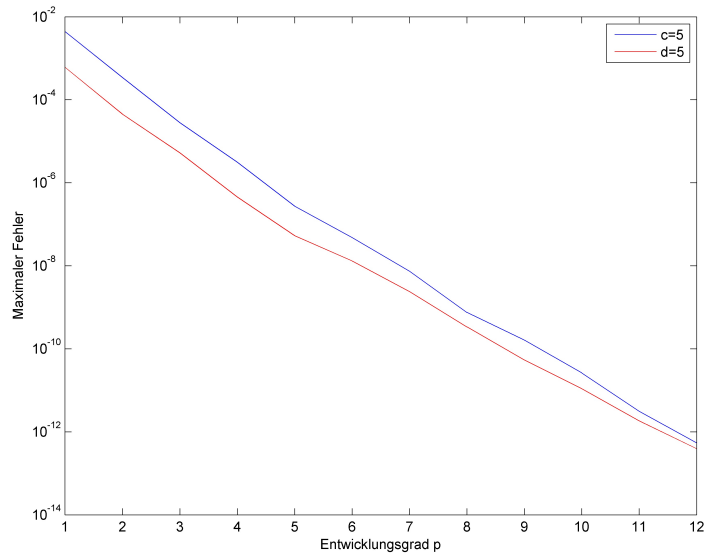


Abbildung 3.4: Vergleich des Fehleranteils von E_{max} bei identen Parametern c und d

Nur bei der Wahl von c und d als drei wird der Fehler durch die adaptive Methode ab dem Polynomgrad zehn größer. Dieser Fall ist zusätzlich in Abbildung 3.5 dargestellt.

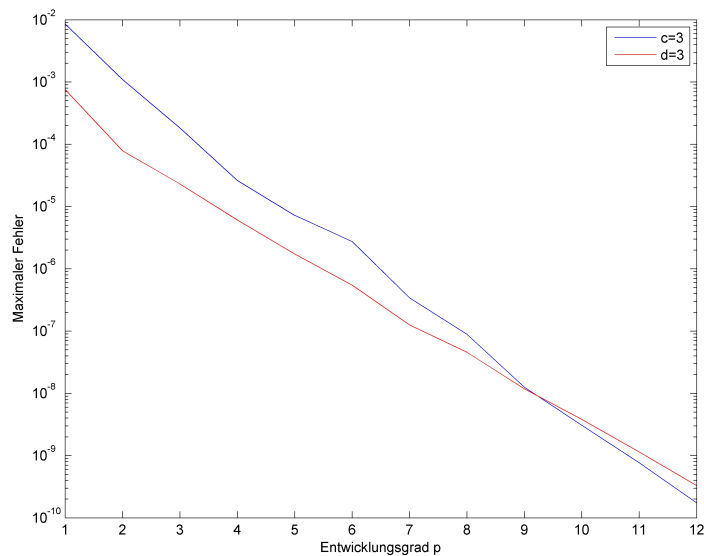


Abbildung 3.5: Vergleich der Konvergenzen bei identen Parametern c und d

Daraus lässt sich schließen, dass d meistens geringer als c gewählt werden kann,

ohne den Fehler negativ zu beeinflussen. Diese Beobachtung hat sich in weiteren Tests bestätigt.

3.2 Testrechnungen für die Integrationspunkte adaptiver BEM-Netze

Als zweites Beispiel wird eine Serie von Punktmengen untersucht, die aus einer adaptiven BEM Rechnung entstehen. Das Testgebiet ist ein stufenförmiges Gebiet. In Abbildung 3.6 ist dieses mit einer adaptiven BEM-Zerlegung dargestellt. In jedem dieser Dreiecke werden 7 Integrationspunkte gemäß [21] verwendet.

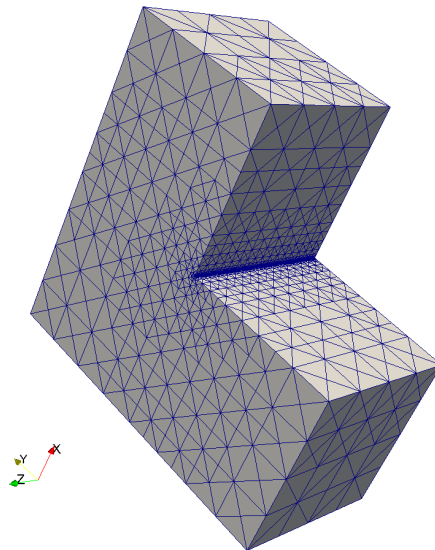


Abbildung 3.6: Stufenförmiges Gebiet mit adaptivem Gitter

Hier wird untersucht, wie sich die Rechenzeit und der Nahfeldanteil in der Multipolmethode bei steigender Anzahl an Punkten verhält. Dafür wurden die Parameter in der Multipolmethode so gewählt, dass bei vergleichbarem Fehler die Rechenzeit optimal ist. Weiters wurde darauf geachtet, dass sich der Nahfeldanteil bei einer Erhöhung der Punkte verringert.

Die Parameter für die adaptive Variante der schnellen Multipolmethode sind in Tabelle 3.5 dargestellt. Die Parameter für die Standardmethode sind bis auf den Parameter d ident. Dieser wurde, wie in Abschnitt 3.1, wieder auf 500 gesetzt, um die Adaptivität im Algorithmus zu verhindern. Dabei ist L_{max} die maximale Tiefe im Clusterbaum und ppc die maximale Anzahl an Punkten pro Cluster.

N	L_{max}	ppc	c	d	p
5992	20	80	2.4	1.5	5
23884	20	80	2.5	1.3	5
92232	20	80	2.7	1.3	5
346080	20	80	2.7	1.3	5

Tabelle 3.3: Multipolparameter für Zerlegungen aus einer adaptiven BEM-Rechnung

Insbesondere wird in dieser Testreihe der Unterschied zwischen der adaptiven Version der schnellen Multipolmethode und der Standardversion untersucht. Dabei sind auch in diesem Vergleich vor allem die Unterschiede in Rechenzeit und Nahfeldanteil interessant.

In Tabelle 3.4 ist diese Testreihe zu sehen. Dabei steht N für die Anzahl der Punkte. Bei jeder Verfeinerung wird die Anzahl der Punkte vervierfacht. D.h. für einen linearen Anstieg in der Rechenzeit würde sich diese ebenfalls vervierfachen. Der Nahfeldanteil müsste sich bei einer linearen Komplexität vierteln.

T_{exakt} ist jene Zeit in Sekunden, die für die direkte Berechnung des Potentials benötigt wird. T ist die Zeit in Sekunden, die der Multipolalgorithmus für die Berechnung des Potentials benötigt. NF ist der Nahfeldanteil in Prozent. E und E_{max} sind die Fehler aus (3.2) und (3.1). N_{M2L} und N_{M2T} stehen für die Anzahl der M2L und M2T Operationen in Tausend. Dabei reicht hier die Anzahl der M2T Operationen anzugeben. Aufgrund der Symmetrie ist die Anzahl der S2L Operationen ident. In der ersten Zeile jeder Verfeinerungsstufe sind die Ergebnisse der adaptiven Variante der schnellen Multipolmethode. In der zweiten Zeile die Ergebnisse der Standardversion der schnellen Multipolmethode.

N	T_{exakt}	T	E_{max}	E	NF	N_{M2L}	N_{M2T}
5992	0.63	0.52	6.8940e-5	7.7632e-6	9.93	5	3
		0.56	7.1742e-5	5.2390e-6	49.49	5	0
23884	9.95	2.3	6.9547e-5	7.4290e-6	2.86	38	10
		3	6.9815e-5	4.8451e-6	15.05	38	0
92232	143	9.7	6.9308e-5	5.0632e-6	0.93	173	32
		13.6	2.7659e-5	2.9400e-6	4.93	173	0
346080	2045	35	8.7396e-5	4.4900e-6	0.23	664	108
		49	8.7398e-5	3.0730e-6	1.17	664	0

Tabelle 3.4: Fehler und Rechenzeiten für die Multipolmethode für mehrere adaptive BEM-Netze

Für die direkte Berechnung des Potentials ist die quadratische Abhängigkeit von der Anzahl der Punkte zu erkennen. Bei einer Vervielfachung der Punkte versechzehnfacht

sich die Rechenzeit. Zusätzlich sind die Ergebnisse für die Rechenzeiten und den Nahfeldanteil in den Abbildungen 3.7 und 3.8 dargestellt. In diesen beiden Abbildungen ist auch jeweils eine Referenzgerade für den linearen Anstieg angegeben. In der Standardversion der schnellen Mutlipolmethode wird der lineare Anstieg in der Rechenzeit um einen geringen Faktor nicht erreicht. Im Vergleich dazu wird in der adaptiven Variante der schnellen Multipolmethode, für dieses Beispiel, ein linearer Anstieg in der Rechenzeit erreicht. Diese beiden Beobachtungen sind besonders gut in Abbildung 3.7 zu erkennen.

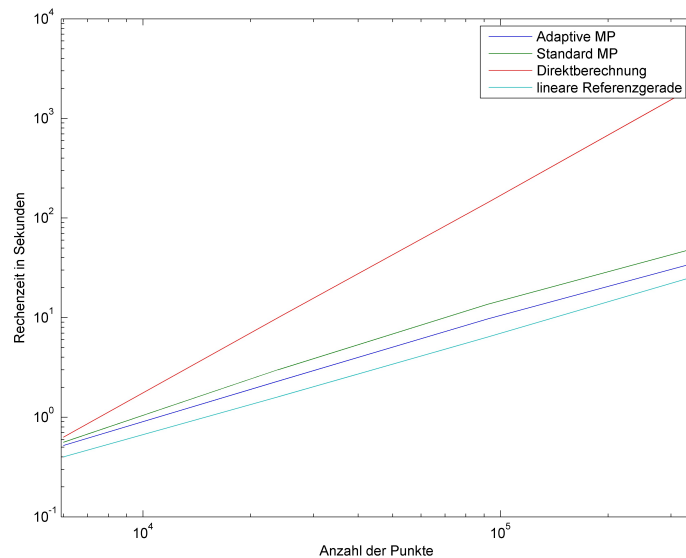


Abbildung 3.7: Entwicklung der Rechenzeit bei wachsender Anzahl an Punkten aus der adaptiven BEM

Dabei ist im weiteren Vergleich der beiden Methoden, die Rechenzeit in der adaptiven Version der schnellen Multipolmethode auf allen Verfeinerungsstufen deutlich geringer als in der Standardversion.

Für den Nahfeldanteil wird für beide Varianten ein nahezu linearer Anstieg erzielt. Dies ist wieder zusätzlich in Abbildung 3.8 dargestellt. Dabei ist zu erwähnen, dass der Nahfeldanteil für die adaptive Version der schnellen Multipolmethode auf allen Verfeinerungsstufen geringer als in der Standardversion ist. Diese Differenz um den Faktor 5 beginnt beim ersten Netz und tritt, infolge des identen linearen Anstiegs, bei allen Verfeinerungsstufen auf.

Auf der feinsten Stufe ist die Standardversion der schnellen Multipolmethode 41 mal schneller als die direkte Berechnung und der Nahfeldanteil beträgt nur 1.17%. Die adaptive schnelle Multipolmethode ist am feinsten Level 58 mal schneller als die direkte Berechnung. Zusätzlich liegt der Nahfeldanteil bei nur 0.23%. Auf dieser Verfeinerungs-

stufe ist die deutliche Einsparung an Rechenzeit und Speicheraufwand zu erkennen. Für weitere Verfeinerungen erhöht sich die Differenz in den Rechenzeiten und dem Speicheraufwand noch mehr.

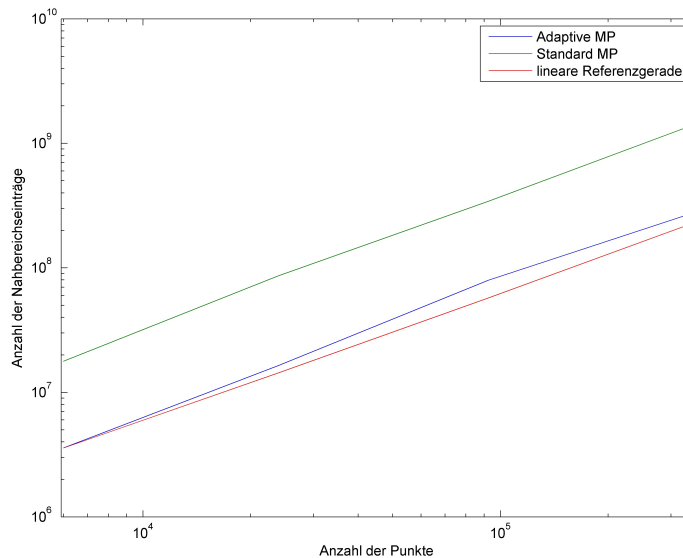


Abbildung 3.8: Anzahl der Nahfeldbeiträge bei wachsender Anzahl an Punkten aus der adaptiven BEM

Zusammenfassen wird durch die adaptive Erweiterung der schnellen Multipolmethode die Rechenzeit, insbesondere bei adaptiven Zerlegungen, zusätzlich stark verringert. Weiters hat sich gezeigt, dass der Nahfeldanteil auf allen Verfeinerungsstufen stark verringert wird. Dabei ist es möglich, den Parameter d so zu wählen, dass bei deutlicher Verringerung der Rechenzeit der Fehler im Wesentlichen gleich bleibt.

3.3 Testrechnung für eine gleichmäßige Ladungsverteilung auf einer Würfeloberfläche

Für dieses Testbeispiel wurde eine gleichmäßige Punktverteilung konstruiert. Dazu wurden auf der Oberfläche eines Würfels, mit Kantenlänge eins, Ladungen gleichmäßig verteilt. In diesem Test wird untersucht, wie sich die adaptive schnelle Multipolmethode und die Standardmethode bei einer Gleichverteilung unterscheiden. Dafür werden die Unterschiede in der Rechenzeit und dem Nahfeldanteil untersucht. Die Parameter im Algorithmus wurden wieder so gewählt, dass bei vergleichbaren Fehlern die bestmögliche Rechenzeit erzielt wurde. Zusätzlich wurde wieder darauf geachtet, dass der Nahfeldanteil bei steigender Punktzahl sinkt.

Für dieses Beispiel war es schwieriger, die optimale Wahl der Parameter zu finden. Dies liegt daran, dass durch die Gleichverteilung die Anzahl der Punkte in fast allen Clustern ident ist. Somit führt eine geringe Änderung in der maximalen erwünschten Anzahl an Punkten pro Cluster dazu, dass alle Cluster entweder verfeinert werden oder nicht. Dies führt weiters zu einer sprunghaften Änderung des Nahfeldanteils und der Anzahl der M2L Operationen.

Die gewählten Parameter sind in Tabelle 3.5 angegeben. Dabei sind in der ersten Zeile jeder Verfeinerungsstufe jeweils die Parameter für die adaptive schnelle Multipolmethode, in der zweiten Zeile die Parameter der Standardmethode. Für die erste Verfeinerungsstufe sind nur die Parameter der Standardmethode angegeben, da in dieser Zerlegung der adaptive Fall nie auftritt.

N	L_{max}	ppc	c	d	p
3174	3	100	1.9	1	4
12696	4	200	2	1.3	4
	4	200	1.8	500	4
50784	5	144	2	1.3	5
	5	140	2	500	5
203136	7	135	2	1.3	5
	7	125	2	500	5

Tabelle 3.5: Multipolparameter für eine Serie an Gleichverteilungen

In Tabelle 3.6 sind die Ergebnisse dieser Tests angeführt. Dabei ist N wieder die Anzahl der Ladungspunkte. E und E_{max} sind die Fehler aus (3.2) und (3.1). T_{exact} ist abermals die Zeit in Sekunden, die für eine direkte Berechnung benötigt wird. T ist die Zeit, die der Multipolalgorithmus benötigt, und NF der Nahfeldanteil. In N_{M2L} und N_{M2T} ist die Anzahl der Operationen M2L und M2T in Tausend. In der ersten Zeile jeder Verfeinerungsstufe sind die Ergebnisse der adaptiven schnellen Multipolmethode, in der zweiten Zeile die Ergebnisse der Standardmethode. Dabei ist in der ersten Verfeinerungsstufe nur eine Rechnung. Bei dieser Zerlegung kam, aufgrund der zu geringen Anzahl an Clustern und der gleichmäßigen Verteilung, der adaptive Fall nie vor.

Wie Tabelle 3.6 zeigt, sind die Unterschiede in der Rechenzeit und dem Nahfeldanteil für die untersuchte Serie an Gleichverteilungen minimal. Für 12696 und 50784 Punkte ist die Rechenzeit in der Standardversion ein wenig schneller als bei der adaptiven Version der schnellen Multipolmethode. Dafür ist bei diesen beiden Rechnungen der Nahfeldanteil in der Standardmethode größer.

Auf der letzten Verfeinerungsstufe wurde, mit Hilfe der Parameter in der Multipolmethode, der Nahfeldanteil in der Standardmethode auf einen ähnlichen Wert wie in der adaptiven Version gebracht. Dies führte aber dafür zu einem deutlichen Anstieg in der Anzahl der M2L Operationen und infolgedessen auch zu einem Anstieg in der

3.3 Testrechnung für eine gleichmäßige Ladungsverteilung auf einer Würfeloberfläche 17

N	T_{exakt}	T	E_{max}	E	NF	N_{M2L}	N_{M2T}
3174	0.19	0.18	2.5466e-5	6.2005e-6	36.71	1.7	0
12696	2.64	0.96 0.91	2.7075e-5 5.1618e-5	9.5191e-6 1.0492e-5	12.40 22.66	9.3 10.1	1.8 0
50784	42	5.1 5.08	1.0162e-5 1.0624e-5	1.9672e-6 1.7401e-6	3.97 5.36	42 68	6.2 0
203136	682	20.1 21	1.3854e-5 1.2723e-5	1.8539e-6 1.8508e-6	0.74 0.69	232.9 456.9	21.9 0

Tabelle 3.6: Fehler und Rechenzeiten für die Multipolmethode für mehrere gleichmäßige Zerlegungen

Rechenzeit. Die Fehler sind wie schon in der Testreihe mit den adaptiven BEM-Netzen auf allen Verfeinerungsstufen nahezu ident.

Diese Ergebnisse sind zusätzlich in den Abbildungen 3.9 und 3.10 dargestellt. Dabei werden in Abbildung 3.9 die unterschiedlichen Rechenzeiten untersucht. Zusätzlich ist auch eine lineare Referenzgerade abgebildet. Aus dieser Abbildung ist zu sehen, dass der lineare Anstieg in der Rechenzeit beinahe erreicht wird.

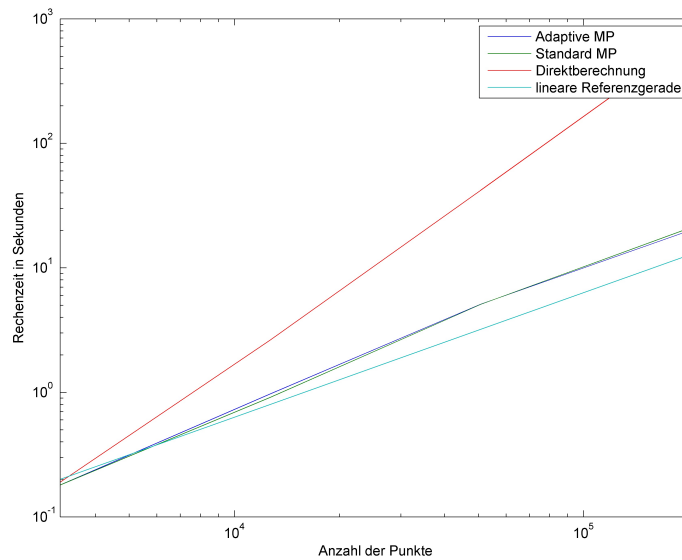


Abbildung 3.9: Entwicklung der Rechenzeit bei wachsender Anzahl an Punkten aus einer gleichmäßigen Ladungsverteilung

In Abbildung 3.10 ist die Anzahl der Nahfeldeinträge abgebildet. Auch hier wird

wieder eine lineare Referenzgerade angegeben. Für die Anzahl der Nahfeldeinträge wird ebenfalls ein annähernd linearer Anstieg erreicht.

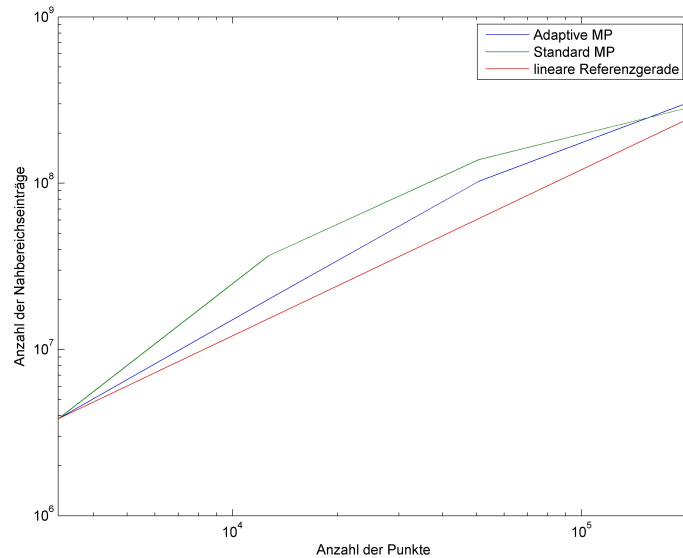


Abbildung 3.10: Anzahl der Nahfeldeinträge bei wachsender Anzahl an Punkten aus einer gleichmäßigen Ladungsverteilung

Aus dieser Testreihe folgt, dass die adaptive Version der schnellen Multipolmethode ohne weiteres auch für gleichmäßige Netze verwendbar ist. Eine Ersparnis in der Rechenzeit ist in dieser Testserie nicht zu erkennen, sehr wohl aber ein geringerer Nahfeldanteil. Auch für diese Testreihe wurde der Fehler durch die zusätzlichen Operationen für die adaptive Variante im Wesentlichen nicht verschlechtert.

3.4 Beschreibung der Implementierung

In diesem Abschnitt erfolgt eine kurze Beschreibung der Implementierung des Multipolalgorithmus. Die in der Implementierung verwendeten Programmiersprachen sind, wie in der Einleitung dieses Kapitels angedeutet, C und C++.

Die Zielsetzung für die Implementierung war es, durch die Aufteilung in Klassen das Programm sauber getrennt zu formulieren. Diese Trennung ermöglicht es auch, nachträglich einfacher Änderungen durchzuführen oder zusätzliche Programmteile hinzuzufügen.

Eine wichtiger Punkt in der Implementierung war es, dass der Clusterbaum im Algorithmus nicht traversiert werden soll. Dies ist bei mehrfachem Aufruf des Multipolalgorithmus für die gleiche Konfiguration von Vorteil, was der Situation in der

schnellen BEM entspricht. Dabei wurde es so umgesetzt, dass die im Multipolalgorithmus benötigten Information in Listen gespeichert werden. Diese Listen sind ein oder mehrdimensionale Arrays. Der Zugriff auf die Informationen in diesen erfolgt direkt durch den Clusterindex. Somit muss der Clusterbaum nicht für die Informationen in den Clustern traversiert werden.

Im Folgenden werden die implementierten Klassen und deren wichtigste Funktionen erklärt. Dabei wurden die folgenden Klassen implementiert:

1. `cluster`: In der Klasse `cluster` sind alle lokalen Daten des Clusters gespeichert. Dazu zählen z.B. der Clusterradius, seine Söhne oder die Anzahl der Punkte im Cluster.
Der Konstruktor dieser Klasse erstellt rekursiv die Baumstruktur für die gewählten Parameter maximale Anzahl an Punkten pro Cluster und maximale Baumtiefe. Dabei ist der Konstruktor aber so implementiert, dass auch einzelne Cluster erzeugt werden können.
2. `cluster_tree`: In der Klasse `cluster_tree` sind die Information der Cluster in Arrays abgespeichert. Dies geschieht folgendermaßen. Der Konstruktor dieser Klasse wird mit der Ladungsmenge, den Daten des Gebiets und den Parametern für die Erstellung des Clusterbaums aufgerufen. Mit diesen Parametern wird der Konstruktor der Klasse `cluster` aufgerufen und der Clusterbaum erstellt.
Als nächstes wird dieser Clusterbaum traversiert, die Information werden aus den Clustern ausgelesen und in die Listen gespeichert. Der Zugriff auf die Information in diesen Listen erfolgt dann direkt über den Clusterindex. Dabei entspricht der Clusterindex der Speicherstelle im Array. Im weiteren Verlauf der Implementierung wird nach der Erstellung dieser Listen nicht mehr direkt auf die Cluster zugegriffen, die Information werden aus den Listen abgefragt.
3. `expansion_set`: In der Klasse `expansion_set` sind die Koeffizienten der Multipol- und lokalen Entwicklung abgespeichert. Auch für diese Entwicklungen erfolgt der Zugriff über den Clusterindex.
4. `cluster_pair`: In `cluster_pair` werden die Interaktionslisten aus Abschnitt 2.3 erzeugt und abgespeichert. Die Erstellung der Listen erfolgt im Konstruktor von `cluster_pair`. Dieser wird mit dem Clusterbaum aus der Klasse `cluster_pair`, den beiden Nahfeldparametern c und d und dem Grad der Multipolentwicklung aufgerufen.
Der Aufruf der Multipolmethode aus Abschnitt 2.5 erfolgt mit der Funktion `go_multipole(double* charges)`. Dabei sind in dem Array `charges` die Ladungen der Teilchen abgespeichert.

Zusätzlich wurden in `fmm_routines` die Operationen zur Berechnung, Verschiebung und Auswertung der Multipol- und lokalen Entwicklung aus Kapitel 1 implementiert.

In der Verwendung der Implementierung ist es wichtig nicht zu vergessen, dass die Ladungsteilchen bei der Erstellung des Clusterbaums umsortiert werden. Für eine weitere Verwendung des Ergebnisvektors aus der Funktion `go_multipole` muss dieser wieder rücksortiert werden oder die Umsortierung in den weiteren Rechnungen berücksichtigt werden.

4 Zusammenfassung und Ausblick

In dieser Arbeit wurde die schnelle Multipolmethode, insbesondere die adaptive Variante, für die effiziente Berechnung des Potentials in einer Ladungsverteilung untersucht. Dabei wurden die theoretischen Grundlagen für die Methode betrachtet, der Algorithmus implementiert und die theoretischen Ergebnisse weiter mit numerischen Beispielen untersucht.

Dafür wurden in Kapitel 1 die Reihenentwicklung des Potentials $\frac{1}{|x-y|}$ eingeführt und die für den Multipolalgorithmus benötigten Operationen definiert. Diese Operationen beinhalten einerseits die Erstellung und Auswertung der Multipolentwicklung und lokalen Entwicklung, andererseits auch die Verschiebungs- und Transformationsoperationen. Für diese Operationen wurden auch stets Fehlerabschätzungen angegeben. Aus diesen Fehlerabschätzungen folgte die Notwendigkeit einer Unterscheidung in Nah- und Fernfeld.

Für diese Unterscheidung wurden in Kapitel 2 Zulässigkeitskriterien definiert. Dabei wurde in dieser Arbeit ein neues Zulässigkeitskriterium für den adaptiven Fall definiert. Aus der weiteren Umformung der Fehlerabschätzungen mit den Zulässigkeitskriterien folgte, dass der Fehler der Entwicklungen von den Nahfeldparametern c und d sowie dem Entwicklungsgrad p abhängt. Daraus folgt für die Implementierung, dass mit diesen Parametern der Fehler gesteuert werden kann. Dabei muss aber berücksichtigt werden, dass eine Erhöhung des Polynomgrads p oder der Nahfeldparameter c und d eine erhöhte Rechenzeit zur Folge hat.

Im Weiteren wurde der Algorithmus zur Erstellung der für die adaptive schnelle Multipolmethode notwendigen Interaktionslisten beschrieben. Dabei musste beachtet werden, dass durch die adaptiven Operation zwei zusätzliche Interaktionslisten benötigt werden. Zum Abschluss des Kapitels wurde der adaptive Multipolalgorithmus im Detail diskutiert.

Im abschließenden dritten Kapitel wurden die Ergebnisse der ersten beiden Kapitel durch numerische Tests weiter untersucht. Dafür wurde in Abschnitt 3.1 anhand einer Zufallsverteilung untersucht, wie sich die Fehleranteile der adaptiven Operationen M2T und S2L im Vergleich mit der Operation M2L verhalten. Aus diesen Tests folgte, dass bei gleichem Fehler der Parameter d aus dem neuen adaptiven Zulässigkeitskriterium meist deutlich geringer als der Parameter c des herkömmlichen Zulässigkeitskriteriums gewählt werden kann.

In den weiteren Tests wurde untersucht, wie sich die Rechenzeit und der Nahfeldanteil für die adaptive schnelle Multipolmethode und die Standardversion unterscheiden. Dabei wurden zwei Testserien an Zerlegungen betrachtet. Die erste Serie entstand aus

einer adaptiven BEM-Rechnung, die zweite aus einer Gleichverteilung auf einer Würfeloberfläche.

Für die Gleichverteilung konnte in der Rechenzeit zwischen den beiden Methoden nur ein geringer Unterschied festgestellt werden. Der Nahfeldanteil war bei der adaptiven Variante minimal geringer. Dies hängt damit zusammen, dass bei einer Gleichverteilung der Anteil der adaptiven Operationen sehr gering ist.

Für die adaptive Serie an Zerlegungen konnte eine deutliche Einsparung in der Rechenzeit und dem Nahfeldanteil in der adaptiven schnellen Multipolmethode im Vergleich zur Standardversion festgestellt werden. Im Vergleich zur direkten Berechnung lieferten die adaptive schnelle Multipolmethode und die Standardmethode die erhofften Ergebnisse. Bei der gleichmäßigen und der adaptiven Zerlegung wurde für beide Methoden ein annähernd linearer Anstieg in der Rechenzeit und im Speicheraufwand erreicht.

Diese beiden Testserien bestätigten auch die Beobachtung für die Parameter c und d aus dem ersten Test. Es war möglich, den Parameter d stets deutlich geringer als den Parameter c zu wählen und dabei den Fehler nur minimal zu verschlechtern. Dies hat eine weitere Verringerung des Nahfeldanteils zur Folge.

Zusammenfassend folgt, dass die Adaptivität im Multipolalgorithmus eine gute Erweiterung für die schnelle Multipolmethode ist, deren Auswirkungen vor allem bei adaptiven Zerlegungen zu bemerken sind. Wie die Tests gezeigt haben, kann die Erweiterung aber auch ohne weiteres für gleichmäßige Verteilungen verwendet werden. Dabei wurde vor allem der Fehler durch die zusätzliche Adaptivität in keinem der Tests wesentlich schlechter.

Für die BEM ist durch die Reduzierung des Speicheraufwandes eine Lösung von Problemen größerer Dimension möglich. Zusätzlich ist durch den Multipolalgorithmus eine effiziente Realisierung der Matrixvektormultiplikation möglich. Dies verringert die Rechenzeit für die Lösung des in der BEM entstandenen Gleichungssystems mit iterativen Verfahren deutlich. Die zusätzlich Verbesserung durch die adaptiven Operationen in der schnellen Multipolmethode ist insbesondere für adaptiven BEM-Rechnungen von Bedeutung.

Literaturverzeichnis

- [1] Mario Bebendorf and S. Rjasanow. Adaptive low-rank approximation of collocation matrices. *Computing*, 70(1):1–24, 2003.
- [2] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms I. *Comm. Pure Appl. Math.*, 44(2):141–183, 1991.
- [3] J. Carrier, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm for particle simulations. *SIAM J. Sci. Statist. Comput.*, 9(4):669–686, 1988.
- [4] H. Cheng, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm in three dimensions. *J. Comput. Phys.*, 155(2):468–498, 1999.
- [5] W. Dahmen, S. Prössdorf, and R. Schneider. Wavelet approximation methods for pseudodifferential equations. II. Matrix compression and fast solution. *Adv. Comput. Math.*, 1(3-4):259–335, 1993.
- [6] W. Dahmen, S. Prössdorf, and R. Schneider. Wavelet approximation methods for pseudodifferential equations. I. Stability and convergence. *Math. Z.*, 215(4):583–620, 1994.
- [7] Michael Feischl, Thomas Führer, Norbert Heuer, Michael Karkulik, and Dirk Praetorius. Adaptive boundary element methods: A posteriori error estimators, adaptivity, convergence, and implementation. *Arch. Comput. Methods Eng.*, 22(3):309–389, 2015.
- [8] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73(2):325–348, 1987.
- [9] Leslie Greengard. *The rapid evaluation of potential fields in particle systems*. ACM Distinguished Dissertations. MIT Press, Cambridge, MA, 1988.
- [10] W. Hackbusch. A sparse matrix arithmetic based on \mathcal{H} -matrices. I: Introduction to \mathcal{H} -matrices. *Computing*, 62(2):89–108, 1999.
- [11] W. Hackbusch, B. Khoromskij, and S. A. Sauter. On \mathcal{H}^2 -matrices. In *Lectures on applied mathematics (Munich, 1999)*, pages 9–29. Springer, Berlin, 2000.
- [12] W. Hackbusch and B.N. Khoromskij. A sparse \mathcal{H} -matrix arithmetic. II: Application to multi-dimensional problems. *Computing*, 64(1):21–47, 2000.
- [13] W. Hackbusch and Z.P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numer. Math.*, 54(4):463–491, 1989.

-
- [14] F. Himmelbauer. *Untersuchung von Modifikationen der schnellen Multipolmethode für adaptive Netze*. Masterprojekt TM, Institut für Numerische Mathematik, Technische Universität Graz, 2016.
- [15] Michael Karkulik, Günther Of, and Dirk Praetorius. Convergence of adaptive 3D BEM for weakly singular integral equations based on isotropic mesh-refinement. *Numer. Methods Partial Differ. Equ.*, 29(6):2081–2106, 2013.
- [16] K. Nabors, F. T. Korsmeyer, F. T. Leighton, and J. White. Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory. *SIAM J. Sci. Comput.*, 15(3):713–735, 1994. Iterative methods in numerical linear algebra (Copper Mountain Resort, CO, 1992).
- [17] G. Of. *Die Multipolmethode für Randintegralgleichungen*. Diplomarbeit, Universität Stuttgart, 2001.
- [18] G. Of, O. Steinbach, and W. L. Wendland. The fast multipole method for the symmetric boundary integral formulation. *IMA J. Numer. Anal.*, 26(2):272–296, 2006.
- [19] J. M. Perez-Jorda and W. Yang. A concise redefinition of the solid spherical harmonics and its use in the fast multipole methods. *J. Chem. Phys.*, 104(20):8003–8006, 1996.
- [20] H. G. Petersen, E. R. Smith, J. W. Perram, and D. Soelvason. Error estimates for the fast multipole method.II The three-dimensional case. *Mathematical and Physical Sciences*, 448:401–418, 1995.
- [21] Johann Radon. Zur mechanischen Kubatur. *Monatsh. Math.*, 52:286–300, 1948.
- [22] Sergej Rjasanow and Olaf Steinbach. *The Fast Solution of Boundary Integral Equations*. Mathematical and Analytical Techniques with Applications to Engineering. Springer, New York, 2007.
- [23] Stefan A. Sauter and Christoph Schwab. *Boundary element methods*, volume 39 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2011. Translated and expanded from the 2004 German original.
- [24] O. Steinbach. *Numerische Näherungsverfahren für elliptische Randwertprobleme. Finite Elemente und Randelemente*. Teubner, Stuttgart-Leipzig-Wiesbaden, 2003.
- [25] O. Steinbach. *Lösungsverfahren für lineare Gleichungssysteme*. Teubner, Stuttgart-Leipzig-Wiesbaden, 2005.
- [26] Philip R. Wallace. *Mathematical analysis of physical problems*. Dover Publications, Inc., New York, 1984. Corrected reprint of the 1972 original.
- [27] Ch. A. White and M. Head-Gordon. Derivation and efficient implementation of the fast multipole method. *J. Chem. Phys.*, 101(8):6593–6605, 1994.

- [28] K. Yoshida. *Applications of Fast Multipole Method to Boundary Integral Equation Method*. Dissertation, Kyoto University, 2001.

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Datum

Unterschrift