

Effiziente Approximation von Integraloperatoren mit Hilfe der Green'schen Darstellungsformel

Dissertation

von

Jessica Gördes

vorgelegt an der Technischen Fakultät
der Christian-Albrechts-Universität zu Kiel

Kiel, den 31.1.2014

Für Sonja

Referent: Prof. Dr. Steffen Börm
Korreferent: Prof. Dr. Malte Braack

Datum der mündlichen Prüfung: 13.12.2013

Zum Druck genehmigt: 13.12.2013

Zusammenfassung

In den Naturwissenschaften müssen immer wieder große vollbesetzte Matrizen behandelt werden. Die Approximation von Integraloperatoren mit einer Standard-Galerkin-Diskretisierung führt typischerweise genau auf solche Matrizen. Um die quadratische Komplexität zu vermeiden, die es braucht um eine vollbesetzte Matrix zu berechnen und zu speichern, sind bereits einige Ansätze eingeführt worden, inklusive hierarchischer Matrizen. Dieses kann erreicht werden, indem die Matrix partitioniert und in einer hierarchischen Repräsentation gespeichert wird, in der gewisse Matrixblöcke durch Niedrigrangmatrizen approximiert werden.

Das analytische Herangehen besteht dabei in einer Approximation der Kernfunktion durch eine separierte Funktion, die dann auf eine Niedrigrangmatrix führt. Es gibt auch einen algebraischen Ansatz, der sich bestimmter Matrixeinträge bedient, um die Niedrigrangapproximation zu berechnen.

In dieser Arbeit stelle ich eine neue Methode vor, um Niedrigrangmatrizen für hierarchischer Matrizen zu konstruieren. Ich bediene mich dabei des Beispiels von Integraloperatoren, um unsere Algorithmen einzuführen.

Diese neue Methode baut eine entartete Kernfunktion auf, indem sie Quadratur auf die durch die Green'sche Formel dargestellte Kernfunktion anwendet. Dieses führt zu einem Algorithmus, für den exponentielle Konvergenz bewiesen wird.

Obwohl die theoretischen Betrachtungen vielversprechend aussehen, kann der Algorithmus sich im dreidimensionalen Fall nicht gegenüber anderen gängigen Verfahren behaupten aufgrund der hohen lokalen Ränge, die dabei produziert werden.

Das Verfahren kann aber noch weiter verbessert werden, indem diese analytische kernbasierte Methode mit der Kreuzapproximation kombiniert wird, um die hohen lokalen Ränge zu reduzieren. Mit einem zusätzlichen Interpolationsansatz ergibt sich ein sehr effizienter und zu gängigen Verfahren konkurrenzfähiger Algorithmus.

Abstract

In the natural sciences there are often large densely populated matrices to be handled. Approximating integral operators by a standard Galerkin discretisation typically leads to this kind of matrices. To avoid the quadratic complexity it takes to compute and store a dense matrix, several approaches have been introduced including hierarchical matrices. This is achieved by partitioning the matrix in a hierarchical representation, where certain matrix blocks are approximated by low rank matrices.

The analytic approach is to approximate the kernel function by a separable function which leads to a low rank matrix. There is also the algebraic approach that uses certain entries of the matrix itself to compute a low rank approximation.

In this thesis I present a new method of low rank approximation to construct hierarchical matrices. I use the example of integral operators to introduce our algorithms.

This new method builds a degenerated kernel function by using quadrature on the kernel function represented with Green's formula. It leads to an algorithm for which rigorous proofs yield exponential convergence.

Although the theoretical results are very promising, the algorithm cannot compete with other currently used methods in the three-dimensional case due to high local ranks.

The method can be further enhanced by combining this analytic kernel-based algorithm with cross approximation to reduce the local ranks. With an additional interpolation ansatz the resulting hybrid algorithm is very efficient and on a competitive basis with currently used methods.

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Einleitung | 1 |
| 1.1 | Handhabung großer vollbesetzter Matrizen | 1 |
| 1.2 | Aufbau dieser Arbeit | 2 |
| 2 | Problemstellung | 5 |
| 2.1 | Modellproblem: Laplace-Gleichung | 5 |
| 2.2 | Weitere Anwendungsbeispiele | 6 |
| 2.2.1 | Einfachschicht- und Doppelschichtpotential | 6 |
| 2.2.2 | Lamé Gleichung | 8 |
| 3 | Hierarchische Matrizen | 9 |
| 3.1 | Matrixpartition | 9 |
| 3.1.1 | Clusterbäume | 9 |
| 3.1.2 | Blockclusterbäume | 11 |
| 3.2 | \mathcal{H} -Matrizen | 13 |
| 3.3 | Anwendung auf Integraloperatoren | 14 |
| 3.3.1 | Clusterung und Zulässigkeit | 15 |
| 3.3.2 | Entartete Kernfunktion | 17 |
| 3.4 | \mathcal{H}^2 -Matrizen | 18 |
| 4 | Derzeit verwendete Verfahren | 21 |
| 4.1 | Interpolation | 21 |
| 4.1.1 | Tensor-Interpolationsoperatoren | 21 |
| 4.1.2 | Kernapproximation | 23 |
| 4.1.3 | Interpolationsfehler | 24 |
| 4.2 | Multipol-Methode | 27 |
| 4.2.1 | Multipol-Entwicklung | 28 |
| 4.2.2 | Clusterung | 29 |
| 4.2.3 | Fehlerbetrachtung | 30 |
| 4.3 | Kreuzapproximation | 30 |
| 4.3.1 | Matrixstruktur | 31 |
| 4.3.2 | Adaptive Kreuzapproximation | 32 |
| 4.3.3 | Fehlerabschätzung | 33 |
| 5 | Die Green-Quadratur-Methode | 37 |
| 5.1 | Anwendung der Green'schen Formel auf die Kernfunktion | 37 |
| 5.2 | Approximation durch Quadratur | 39 |

| | | |
|-----------|--|------------|
| 5.3 | Verallgemeinerungen | 43 |
| 5.3.1 | 3D-Fall | 43 |
| 5.3.2 | Kreis-Parametrisierung | 47 |
| 5.3.3 | Anwendung auf SLP und DLP | 48 |
| 5.3.4 | Ansatz für uniforme \mathcal{H} -Matrizen | 50 |
| 6 | Fehleranalyse der Green-Quadratur-Methode | 53 |
| 6.1 | Fehlerabschätzung im zweidimensionalen Fall | 53 |
| 6.1.1 | Eigenschaften der Parametrisierung | 53 |
| 6.1.2 | Quadraturfehler-Abschätzung | 57 |
| 6.2 | Alternative Fehlerabschätzung | 63 |
| 6.2.1 | Eigenschaften der Parametrisierung im dreidimensionalen Fall | 64 |
| 6.2.2 | Allgemeinere Quadraturfehler-Abschätzung | 66 |
| 7 | Numerische Experimente für das Green-Quadratur-Verfahren | 73 |
| 7.1 | 2D-Fall | 73 |
| 7.1.1 | Verfahrensanalysen | 73 |
| 7.1.2 | Vergleiche mit anderen analytischen Verfahren | 78 |
| 7.2 | 3D-Fall | 82 |
| 8 | Das effiziente Green-Hybrid-Verfahren | 85 |
| 8.1 | Kombination der Green-Quadratur mit ACA und Interpolation | 85 |
| 8.1.1 | Rangreduzierung im Green-Quadratur-Verfahren mit ACA | 86 |
| 8.1.2 | Effizienz durch eine interpolierende Matrix | 87 |
| 8.2 | Anwendung auf \mathcal{H}^2 -Matrizen | 89 |
| 8.2.1 | Idee anhand uniformer \mathcal{H} -Matrizen | 89 |
| 8.2.2 | Geschachtelte Clusterbasen | 90 |
| 8.2.3 | \mathcal{H}^2 -Matrix-Approximation | 96 |
| 9 | Fehleranalyse des Green-Hybrid-Verfahrens | 99 |
| 9.1 | Fehlerabschätzung für das Hybrid-Verfahren | 99 |
| 9.2 | Fehlerabschätzung im \mathcal{H}^2 -Fall | 102 |
| 10 | Numerische Experimente für das Hybrid-Verfahren | 111 |
| 10.1 | Hybrid-Verfahren für \mathcal{H} -Matrizen | 111 |
| 10.1.1 | 2D-Fall Analyse | 111 |
| 10.1.2 | 2D-Fall Vergleiche | 116 |
| 10.1.3 | 3D-Fall Analyse | 118 |
| 10.1.4 | 3D-Fall Vergleiche | 122 |
| 10.2 | Hybrid-Verfahren für \mathcal{H}^2 -Matrizen | 125 |
| 10.2.1 | 2D-Fall | 125 |
| 10.2.2 | 3D-Fall | 128 |
| 11 | Zusammenfassung und Ausblick | 133 |

1 Einleitung

In den Naturwissenschaften treten immer wieder Probleme auf, die das Lösen sehr großer Gleichungssysteme erfordern, also das Speichern und Rechnen mit sehr großen Matrizen. Wenn diese in herkömmlicher Art und Weise als zweidimensionales Array gespeichert werden, wächst die Größe der Matrizen quadratisch mit der Problemgröße, so dass es schnell zu Speicherproblemen und zu enorm langen Berechnungszeiten kommen kann. Daher ist es wichtig nach Möglichkeiten zu suchen, diese Speicherprobleme und die langen Berechnungszeiten zu vermeiden.

Das Ziel dieser Arbeit ist, große vollbesetzte Matrizen, wie sie zum Beispiel bei der Diskretisierung von Randintegraloperatoren entstehen, so zu approximieren, dass sie hinreichend genau sind, man sie günstig speichern und effizient mit ihnen rechnen kann.

Mit Hilfe von \mathcal{H} -Matrizen wird dieses erreicht, indem die Matrix partitioniert und in hierarchischer Repräsentation abgespeichert wird, in der bestimmte Teilblöcke durch Niedrigrangmatrizen approximiert werden.

Es sind bisher diverse Möglichkeiten bekannt, diese Niedrigrangapproximationen zu berechnen (siehe Abschnitt 1.1). Einige davon nutzen aus, dass der Matrix eine Kernfunktion zugrunde liegt und führen die eigentliche Approximation auf dem Kern aus. Andere wiederum benutzen direkt die Matrixeinträge der Matrix ohne Kenntnis darüber, woraus diese aufgebaut wird.

Im Rahmen dieser Arbeit wird eine weitere Methode zur Niedrigrangapproximation über die Kernfunktion vorgestellt, die auf der Green'schen Formel und Quadratur beruht.

1.1 Handhabung großer vollbesetzter Matrizen

Sehr große vollbesetzte Matrizen, die bei der Simulation von Vielkörpersystemen oder der Diskretisierung von Differential- und Integraloperatoren auftauchen, sind numerisch schwer handhabbar. Daher wurden bereits etliche Ansätze entworfen, um die quadratische Komplexität zu vermeiden, die das Abspeichern und die Berechnung einer in üblicher Art und Weise als zweidimensionales Array mit sich bringen würde.

Ein Ansatz besteht darin, eine Diskretisierung mit Wavelets durchzuführen, so dass die resultierende Matrix als schwach besetzte Matrix abgespeichert werden kann [32, 6, 13, 12]. Allerdings ist es keine einfache Aufgabe, auf komplexeren Gebieten Wavelet-Basen zu konstruieren, was die Nützlichkeit dieser Methode einschränkt.

Die Approximationsalgorithmen, die den effizienten Umgang mit großen vollbesetzten Matrizen erlauben, kann man in zwei Kategorien und Mischformen daraus, die sogenannten hybriden Verfahren, aufteilen. In der ersten Kategorie befinden sich die analytischen Methoden, die auf der Approximation der Kernfunktion basieren.

1 Einleitung

Dazu gehören das Panel-Clustering-Verfahren, das eine Taylor-Entwicklung zur Approximation benutzt [23, 24] und die weit verbreitete Multipol-Methode [30, 20], die auf einer speziellen optimierten Multipol-Entwicklung beruht und im Laufe dieser Arbeit noch etwas genauer betrachtet wird.

Ein Spezialfall davon ist die *Multipol-Methode ohne Multipole* [1, 25, 35], bei der Punkte entlang von Hilfskurven um das Approximationsgebiet herum benutzt werden, um die Matrixapproximation zu konstruieren.

Ein weiteres Verfahren in dieser Kategorie der kernbasierten Verfahren ist die Interpolation, mit der Niedrigrangapproximationen zum Aufbau einer hierarchischen Matrix berechnet werden [9]. Dieses Verfahren wird noch eine größere Rolle im Laufe dieser Arbeit spielen und später genau erklärt werden.

Neuerdings gibt es neben der Kategorie mit den analytischen Verfahren auch eine zweite Kategorie mit algebraischen Verfahren. Die Algorithmen in dieser zweiten Kategorie bedienen sich direkt der Matrixeinträge für die Approximation anstatt über die Kernfunktion zu gehen, wie die Algorithmen der ersten Kategorie.

So werden bei den Mosaic-Skeleton-Methoden und den Kreuzapproximations-Verfahren eine kleine Anzahl an Matrixeinträgen nach heuristischen Methoden aus- gesucht und damit dann Niedrigrangapproximationen von bestimmten Matrixblöcken aufgebaut [16, 17, 34, 2, 4].

Es gibt auch Verfahren, die versuchen die Vorteile beider Kategorien zu kombinieren. Dabei entstehen Mischformen, die als hybride Verfahren bezeichnet werden, so wie etwa die hybride Kreuzapproximation [10].

In der vorliegenden Arbeit wird eine neue Methode zur Konstruktion von Niedrigrangapproximationen für hierarchische Matrizen vorgestellt. Diese basiert auf der Green'schen Formel und Quadratur und führt zu einem flexiblen und eleganten Algorithmus, für den eine exponentielle Konvergenz bewiesen wird. Der Algorithmus gehört dabei in die erste Kategorie der kernapproximierenden Verfahren.

Eine Weiterentwicklung des Green-Quadratur-Verfahrens stellt dann das Green-Hybrid-Verfahren dar. Es bedient sich an Methoden aus beiden Kategorien und ist im Grunde eine geschickte Verknüpfung der Green-Quadratur-Methode und der Kreuzapproximation, die aber auch noch einen Interpolations-Ansatz enthält.

Für dieses Hybrid-Verfahren existieren ebenso beweisbare Konvergenzaussagen und es ist absolut konkurrenzfähig gegenüber den altbewährten Verfahren, wie anhand von numerischen Experimenten deutlich gemacht wird.

1.2 Aufbau dieser Arbeit

Die Idee der Green-Quadratur und die beiden daraus entstandenen Algorithmen werden anhand der Anwendung auf Integraloperatoren vorgestellt. Dazu wird in Kapitel 2 das Modellproblem formuliert, auf das in der gesamten Arbeit Bezug genommen wird. Ansonsten bietet Kapitel 2 noch eine knappe Übersicht über weitere Anwendungsbeispiele.

Da die Konstruktionen sowie die Beweise zu den neuen Algorithmen eine eingehende Kenntnis der Theorie der hierarchischen Matrizen erfordern, werden \mathcal{H} -Matrizen und \mathcal{H}^2 -Matrizen und ihr Umfeld in Kapitel 3 definiert und ihre Anwendung auf Integraloperatoren erklärt.

Um die neue Methode gut einordnen zu können und die Verfahren, mit denen sie verglichen wird, zu verstehen, bietet Kapitel 4 eine kleine Einführung in die Interpolation, die Multipol-Methode und die Kreuzapproximation als drei der aktuell wichtigsten Verfahren.

In Kapitel 5 wird dann die Idee der neuen Green-Quadratur-Methode vorgestellt und beschrieben, wie daraus ein anwendbarer Algorithmus entsteht.

Die Fehlerabschätzungen dazu werden in Kapitel 6 bewiesen und eine praxisnahe Untersuchung des Algorithmus anhand numerischer Experimente ist in Kapitel 7 zu finden.

Die Weiterentwicklung der Green-Quadratur-Idee zu dem effizienten Green-Hybrid-Verfahren für \mathcal{H} -Matrizen und auch die Anwendung für \mathcal{H}^2 -Matrizen ist in Kapitel 8 erklärt.

Die zugehörige Fehleranalyse für beide Fälle folgt dann in Kapitel 9.

Numerische Experimente mit Vergleichen zu anderen Verfahren zur Belegung der Konkurrenzfähigkeit des Green-Hybrid-Verfahrens sind in Kapitel 10 beschrieben.

Den Abschluss dieser Arbeit bildet Kapitel 11 mit einer Zusammenfassung und einem Ausblick auf mögliche Ansätze zur weiteren Erforschung dieses Themas.

Danksagung

Zu großem Dank verpflichtet bin ich Prof. Dr. Steffen Börm, der für alle fachlichen wie nichtfachlichen Probleme stets ein offenes Ohr hatte und mich bei der Entstehung dieser Arbeit mit viel Geduld gut betreut hat.

Weiter möchte ich Julia Falke für die aufschlussreichen Gespräche über das Green-Quadratur-Verfahren danken, die wir im Rahmen meiner Mitbetreuung ihrer Diplomarbeit führten und die mir selbst noch neue Erkenntnisse für diese Arbeit lieferten.

Mein ganz besonderer Dank gilt meinem tapferen Mitstreiter und Wunder-Programmierer Sven Christophersen, der ausgehend von meiner ersten groben Green-Quadratur-Implementierung die Programme zu unseren Algorithmen erweiterte, optimierte und jede neue fixe Idee prompt umzusetzen wusste.

Großer Dank geht auch an Dr. Verena Heidrich-Meisner für das Korrekturlesen dieser Arbeit.

Ferner möchte ich meinem Vater Bernd Gördes danken, ohne dessen Unterstützung ich nie angefangen hätte, zu studieren und Jens Burmeister, ohne den mein Studium nie zu einem Diplom geführt hätte.

Für den nötigen seelischen Beistand in meiner Promotionszeit sorgten meine Freundin Sandra Lemmerz und mein Lebensgefährte Erik Steffen: Ohne euch wäre das hier nicht möglich gewesen! Ich danke euch!

2 Problemstellung

In diesem Kapitel wird zunächst die zugrundeliegende Problemstellung in Form eines Modellproblems erläutert und danach noch kurz ein weiteres Anwendungsbeispiel vorgestellt.

2.1 Modellproblem: Laplace-Gleichung

Die Methode, die im Rahmen dieser Arbeit entwickelt und untersucht wurde, wird anhand des folgenden einfachen Modellproblems, der *Laplace-Gleichung*, vorgestellt. Dazu wird ein Gebiet $\Omega \subset \mathbb{R}^d$ mit $d \in \{2, 3\}$ und das wie folgt lautende Dirichlet-Problem

$$\begin{aligned} \Delta u(x) &= 0 && \text{für alle } x \in \Omega, \\ u(x) &= f(x) && \text{für alle } x \in \Gamma := \partial\Omega \end{aligned} \quad (2.1)$$

mit gegebenen Dirichlet-Werten in Form der stetigen Funktion $f : \Gamma \rightarrow \mathbb{R}$ auf dem Rand von Ω betrachtet. Gesucht ist dabei die Funktion $u : \Omega \rightarrow \mathbb{R}$.

Mit Anwendung der Randelementmethode erhält man einen Fredholm Integraloperator der Form

$$\mathcal{G}[u](x) = \int_{\Gamma} g(x, y) u(y) \, d\Gamma(y) \quad (2.2)$$

über dem Rand Γ von Ω . Dabei ist $g : \Omega \times \Omega \rightarrow \mathbb{R}$ die *Fundamentallösung des Laplace-Operators* und gegeben durch

$$g(x, y) = \begin{cases} -\frac{1}{2\pi} \log \|x - y\| & \text{für } d = 2 \\ \frac{1}{4\pi} \|x - y\|^{-1} & \text{für } d = 3 \end{cases} \quad (2.3)$$

mit $x, y \in \Omega$ und $x \neq y$. Eine Standard-Galerkin-Diskretisierung von \mathcal{G} für eine Basis $(\varphi_i)_{i \in \mathcal{I}}$ mit einer Indexmenge \mathcal{I} führt zu einer Matrix $G \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}}$ mit den Einträgen

$$G_{i,j} := \int_{\Gamma} \int_{\Gamma} \varphi_i(x) g(x, y) \varphi_j(y) \, d\Gamma(x) \, d\Gamma(y). \quad (2.4)$$

Im Standardfall ergibt sich für G eine vollbesetzte Matrix.

Um nun die quadratische Komplexität zu vermeiden, die das Berechnen und Speichern dieser vollbesetzten Matrix erfordern würde, soll eine geeignete speichergünstigere Approximation gefunden werden. Diese wird im Rahmen der vorliegenden Arbeit in Form einer hierarchischen Matrix erarbeitet.

Durch den engen Zusammenhang zwischen der Matrix G und der Kernfunktion g ist es sinnvoll, dazu den analytischen Weg über eine Kernapproximation zu gehen. Dieses

2 Problemstellung

Vorgehen wird in Abschnitt 3.3.2 genau beschrieben, nachdem hierarchische Matrizen im Allgemeinen im vorangehenden Teil von Kapitel 3 eingeführt werden.

Die spezielle, in dieser Arbeit vorgestellte, Kernapproximation ist gestützt auf die zweite Green'sche Formel [18, 2. art 3.]

$$u(x) = \int_{\Gamma} g(x, z) \frac{\partial u}{\partial n_z}(z) \, d\Gamma(z) - \int_{\Gamma} \frac{\partial g}{\partial n_z}(x, z) u(z) \, d\Gamma(z), \quad (2.5)$$

und leitet daraus eine sogenannte *entartete Kernfunktion*, wie in Abschnitt 3.3.2 beschrieben, ab. Die genaue Beschreibung des resultierenden Algorithmus findet sich in Kapitel 5.

2.2 Weitere Anwendungsbeispiele

Auch wenn die hier vorgestellten Algorithmen alle nur für das Problem (2.1) erklärt und untersucht werden, so ist dies durchaus nicht die einzige Anwendungsmöglichkeit. Sobald es eine Darstellungsformel in der Art von (2.5) gibt, kann diese für die Kernapproximation der hier gezeigten Verfahren genutzt werden.

In Abschnitt 2.2.1 werden zwei hier relevante Lösungen des Modellproblems (2.1) kurz beschrieben und in Abschnitt 2.2.2 ein anderes Problem als weitere Anwendungsmöglichkeit skizziert, das jedoch im Verlauf dieser Arbeit nicht weiter betrachtet wird.

2.2.1 Einfachschicht- und Doppelschichtpotential

Im Rahmen der numerischen Experimente in Kapitel 7 werden sowohl Resultate für das Einfachschichtpotential als auch für das Doppelschichtpotential vorgeführt. Daher sollen diese Begriffe an dieser Stelle formal definiert werden und in den Kontext des Modellproblems gebacht werden. So bekommt der zuvor erwähnte Integraloperator (2.2) noch einen weiteren Namen:

Definition 2.1 (Einfachschichtpotential) *Der Integraloperator $\mathcal{V} : H^{-\frac{1}{2}}(\Gamma) \rightarrow H^{\frac{1}{2}}(\Gamma)$ mit*

$$\mathcal{V}[u](x) := \int_{\Gamma} g(x, y) u(y) \, d\Gamma(y) \quad \text{für alle } u \in H^{-\frac{1}{2}}(\Gamma)$$

wird Einfachschichtpotential (single layer potential) genannt oder abgekürzt durch SLP entsprechend der englischen Bezeichnung.

Für die Diskretisierung mit dem Galerkin-Verfahren wird hier immer davon ausgegangen, dass sich $\Gamma = \partial\Omega$ durch einen geeigneten Polygonzug im zweidimensionalen Fall oder eine geeignete Triangulierung im dreidimensionalen Fall approximieren lässt, so dass darüber dann stückweise konstante Basisfunktionen $(\varphi_i)_{i \in \mathcal{I}}$ über jedem der i Elemente der Zerlegung gewählt werden können.

Die Algorithmen in dieser Arbeit sind genauso auch für die Wahl von stückweise linearen Basisfunktionen ohne größere Probleme durchführbar, es wird jedoch der Einfachheit halber die Theorie anhand der konstanten Basisfunktionen aufgespannt.

Die Galerkin-Diskretisierung liefert schließlich die Matrix mit den Einträgen, die bereits in (2.4) zu sehen waren:

Definition 2.2 (Matrix des Einfachschichtpotentials) Die Galerkin-Diskretisierung von \mathcal{V} für Basisfunktionen $(\varphi_i)_{i \in \mathcal{I}}$ liefert die Matrix des Einfachschichtpotentials

$$V \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}} \quad \text{mit} \quad V_{i,j} := \int_{\Gamma} \varphi_i(x) \int_{\Gamma} g(x,y) \varphi_j(y) \, d\Gamma(y) \, d\Gamma(x), \quad (2.6)$$

die auch kurz SLP-Matrix genannt wird.

Um die Kernapproximation zu konstruieren, kann man jetzt die Green'sche Formel (2.5) auf $x \mapsto g(x,y)$ anwenden und bekommt die folgende Gleichung:

$$g(x,y) = \int_{\Gamma} g(x,z) \frac{\partial g}{\partial n_z}(z,y) \, d\Gamma(z) - \int_{\Gamma} \frac{\partial g}{\partial n_z}(x,z) g(z,y) \, d\Gamma(z). \quad (2.7)$$

Das genaue Vorgehen dazu wird ausführlich in Kapitel 5 und den folgenden Kapiteln behandelt.

Es soll noch ein weiterer Integraloperator zur Laplace-Gleichung definiert werden:

Definition 2.3 (Doppelschichtpotential) Der Integraloperator $\mathcal{K} : H^{\frac{1}{2}}(\Gamma) \rightarrow H^{\frac{1}{2}}(\Gamma)$ mit

$$\mathcal{K}[u](x) := \int_{\Gamma} \frac{\partial g}{\partial n_y}(x,y) u(y) \, d\Gamma(y)$$

wird Doppelschichtpotential (double layer potential) genannt oder abgekürzt durch DLP entsprechend der englischen Bezeichnung.

Nach analogem Vorgehen zur Diskretisierung des Einfachschichtpotentials erhält man auch für das Doppelschichtpotential eine Matrix:

Definition 2.4 (Matrix des Doppelschichtpotentials) Die Galerkin-Diskretisierung von \mathcal{K} für Basisfunktionen $(\varphi_i)_{i \in \mathcal{I}}$ liefert die Matrix des Doppelschichtpotentials

$$K \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}} \quad \text{mit} \quad K_{i,j} := \int_{\Gamma} \varphi_i(x) \int_{\Gamma} \frac{\partial g}{\partial n_y}(x,y) \varphi_j(y) \, d\Gamma(y) \, d\Gamma(x), \quad (2.8)$$

die auch kurz DLP-Matrix genannt wird.

Im Falle des Doppelschichtpotentials kann man die Green'sche Formel (2.5) anstatt auf $x \mapsto g(x,y)$, wie in (2.7), diesmal auf $x \mapsto \frac{\partial g}{\partial n_y}(x,y)$ anwenden und die Gleichung

$$\frac{\partial g}{\partial n_y}(x,y) = \int_{\Gamma} g(x,z) \frac{\partial^2 g}{\partial n_z \partial n_y}(z,y) \, d\Gamma(z) - \int_{\Gamma} \frac{\partial g}{\partial n_z}(x,z) \frac{\partial g}{\partial n_y}(z,y) \, d\Gamma(z) \quad (2.9)$$

erhalten, mit der man dann ebenso weiter verfährt, wie mit Gleichung (2.7).

2 Problemstellung

2.2.2 Lamé Gleichung

Nur um aufzuzeigen, dass die hier vorgestellten Algorithmen nicht auf die Problemstellung der Laplace-Gleichung (2.1) beschränkt sind, sondern dass es durchaus mehr Anwendungsmöglichkeiten gibt, wird hier als Beispiel kurz die Problemstellung der linearen Elastostatik herangezogen. Um das zugehörige System partieller Differentialgleichungen dazu zu bekommen, muss in (2.1) die Laplace-Gleichung $\Delta u(x) = 0$ durch die *Lamé-Gleichung*

$$-\mu \Delta u(x) - (\lambda + \mu) \operatorname{grad} \operatorname{div} u(x) = 0 \quad \text{für alle } x \in \Omega$$

ersetzt werden, wobei es sich bei λ und μ um die sogenannten *Laméschen Elastizitätskonstanten* handelt. Diese sind definiert durch

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad \text{und} \quad \mu = \frac{E}{2(1+\nu)}$$

mit dem *Elastizitätsmodul* $E > 0$ und der *Poissonzahl* $\nu \in (0, \frac{1}{2})$.

Gesucht ist dabei die vektorwertige Funktion $u : \Omega \rightarrow \mathbb{R}^d$, die auch *Verschiebungsfeld* genannt wird. Die matrixwertige Fundamentallösung $H : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^{d \times d}$ dazu ist gegeben durch

$$H_{k,l}(x,y) = \frac{1}{2E} \frac{1+\nu}{1-\nu} \left[(3-4\nu) g(x,y) \delta_{kl} + \frac{(x_k - y_k)(x_l - y_l)}{\|x - y\|^d} \right]$$

für $k, l \in \{1, \dots, d\}$, wobei δ_{kl} das Kronecker-Delta-Symbol ist und $g(x, y)$ wie in (2.3) definiert ist.

Auch für dieses Problem lässt sich eine Darstellungsformel finden:

$$u(x) = \int_{\Gamma} H(x,y) \gamma_1^{int} u(y) \, d\Gamma(y) - \int_{\Gamma} \gamma_{1,y}^{int} H(x,y) u(y) \, d\Gamma(y) \quad (2.10)$$

wobei der *Randspannungsoperator* γ_1^{int} die Rolle der Normalenabbildung in (2.7) einnimmt.

Eine ausführliche Beschreibung des Randwertproblems zur linearen Elastostatik ist zum Beispiel nachzulesen in [33].

Mit der Darstellungsformel (2.10) wäre also das geeignete Gegenstück zur Green'schen Formel (2.5) gefunden und damit der Ansatzpunkt für eine Kernapproximation mit Hilfe der hier vorgestellten Algorithmen geschaffen.

3 Hierarchische Matrizen

Die grundlegende Idee hinter dem Format der hierarchischen Matrizen ist, sehr große vollbesetzte Matrizen speichergünstig aber möglichst gut zu approximieren und damit Rechenoperationen wie die Matrix-Vektor-Multiplikation, Matrix-Addition und -Multiplikation oder auch die Inversion der Matrix mit geringem Rechenaufwand durchführen zu können.

Um dieses zu erreichen, werden geeignete Untermatrizen durch Niedrigrangmatrizen approximiert, die sehr speichergünstig abgelegt werden können. Die Unterteilung der Matrix in disjunkte Untermatrizen, also die Bereitstellung einer Blockpartition der Matrix, wird über eine Partition ihrer Indexmengen erreicht, die hierarchisch in Baumstrukturen organisiert sind. So werden die Zeilenindexmenge und die Spaltenindexmenge jeweils in sogenannten Clusterbäumen angeordnet, die dann zu einem Blockclusterbaum zusammengefügt werden.

Mit Hilfe einer Zulässigkeitsbedingung wird dann entschieden, welche Blöcke der Matrix durch Niedrigrangmatrizen approximiert werden können.

Diese Matrixpartition wird in Abschnitt 3.1 genauer behandelt, um damit in Abschnitt 3.2 und 3.4 hierarchische Matrizen definieren zu können.

Eine ausführliche Behandlung der Theorie hierarchischer Matrizen ist zu finden in [22] und [7].

Abschließend wird in Abschnitt 3.3 dieses Kapitels die Anwendung der Technik der hierarchischen Matrizen auf Integraloperatoren aufgezeigt.

3.1 Matrixpartition

Im Folgenden werden die hierarchischen Strukturen der Cluster- und Blockclusterbäume vorgestellt. Sei dazu $M \in \mathbb{R}^{\mathcal{I} \times \mathcal{J}}$ eine Matrix mit endlichen Indexmengen \mathcal{I} und \mathcal{J} .

Notationen und Aufbau dieses Abschnitts richten sich größtenteils nach dem Buch von Steffen Börm [7].

3.1.1 Clusterbäume

Um den Begriff des Clusterbaums einführen zu können, werden zunächst grundlegende Notationen zu allgemeinen Bäumen benötigt, die in den folgenden Definitionen gegeben werden.

Definition 3.1 (Baum) *Das Tripel $\mathcal{T} = (V, S, \varrho)$ bestehend aus einer endlichen Menge V , einer Abbildung $S : V \rightarrow \mathcal{P}(V)$ und einem Element $\varrho \in V$ heißt Baum genau dann,*

3 Hierarchische Matrizen

wenn für jedes $t \in V$ genau ein Tupel (t_0, t_1, \dots, t_l) existiert mit $t_i \in S(t_{i-1})$ für alle $i \in \{1, \dots, l\}$ und $t_0 = \varrho$ und $t_l = t$.

Notation 3.2 (zu Bäumen) Sei $\mathcal{T} = (V, S, \varrho)$ ein Baum. Dann wird V als Knotenmenge und jedes $t \in V$ als Knoten bezeichnet. Es wird auch abkürzend $t \in \mathcal{T}$ für einen Knoten t des Baumes \mathcal{T} geschrieben.

S wird auch als Sohnabbildung bezeichnet, $S(t)$ als Menge der Söhne von t und $s \in S(t)$ als Sohn von t . Falls $S(t) = \emptyset$, so wird t Blatt des Baumes genannt. Die Menge aller Blätter von \mathcal{T} wird mit $\mathcal{L}(\mathcal{T})$ bezeichnet.

Das Element $\varrho = \text{root}(\mathcal{T})$ heißt Wurzel des Baumes \mathcal{T} , und das Tupel (t_0, t_1, \dots, t_l) mit $t_i \in S(t_{i-1})$ für alle $i \in \{1, \dots, l\}$ ist der Pfad von t_0 nach t_l .

Definition 3.3 (Nachfahren eines Knotens) Sei $\mathcal{T} = (V, S, \varrho)$ ein Baum und $t \in V$ ein Knoten. Dann ist durch

$$\text{sons}(t) := S(t)$$

die Menge der Söhne von t bezeichnet. Die Menge der Nachfahren $\text{sons}^*(t)$ von t wird induktiv über die Söhne von t definiert:

$$\text{sons}^*(t) := \begin{cases} \{t\} \cup \bigcup_{t' \in \text{sons}(t)} \text{sons}^*(t') & \text{falls } \text{sons}(t) \neq \emptyset \\ \{t\} & \text{sonst} \end{cases} .$$

Von besonderem Interesse sind hier bewertete Bäume, bei denen jedem Knoten noch zusätzliche Informationen als Bewertung zugeordnet werden. Sei B eine Menge solcher Bewertungen.

Definition 3.4 (Bewerteter Baum) Sei $\mathcal{T} = (V, S, \varrho)$ ein Baum und $\lambda : V \rightarrow B$ eine Abbildung, die jedem $t \in V$ eine Bewertung $\hat{t} := \lambda(t) \in B$ zuordnet. Dann ist $\mathcal{T} = (V, S, \varrho, \lambda, B)$ ein bewerteter Baum.

Bemerkung 3.5 (Bewertungen) Oft wird auf die Unterscheidung zwischen einem Knoten t und seiner Bewertung \hat{t} verzichtet und beide werden synonym verwendet.

Mit Hilfe dieser Notationen kann nun die Definition eines Clusterbaums formuliert werden.

Definition 3.6 (Clusterbaum) Sei \mathcal{T} ein bewerteter Baum. \mathcal{T} ist genau dann ein Clusterbaum für die Indexmenge \mathcal{I} , wenn folgendes erfüllt ist:

- Für die Wurzel $\varrho \in \mathcal{T}$ gilt $\hat{\varrho} = \mathcal{I}$.
- Für $t \in \mathcal{T}$ mit $t \notin \mathcal{L}(\mathcal{T})$ gilt:

$$\bigcup_{t' \in \text{sons}(t)} \hat{t}' = \hat{t} .$$

- Für Söhne $t', s' \in \text{sons}(t)$ von t mit $t' \neq s'$ gilt

$$\hat{t}' \cap \hat{s}' = \emptyset.$$

Notation 3.7 (zu Clusterbäumen) Ein Clusterbaum \mathcal{T} zur Indexmenge \mathcal{I} wird mit $\mathcal{T}_{\mathcal{I}}$ bezeichnet. Ein Knoten $t \in \mathcal{T}_{\mathcal{I}}$ heißt Cluster und die Menge aller Blätter eines Clusterbaums $\mathcal{T}_{\mathcal{I}}$ wird als $\mathcal{L}_{\mathcal{I}}$ geschrieben.

Mit Hilfe von Clusterbäumen lassen sich die Zeilenindexmenge \mathcal{I} und die Spaltenindexmenge \mathcal{J} einer Matrix $M \in \mathbb{R}^{\mathcal{I} \times \mathcal{J}}$ hierarchisch gliedern, so dass immer die Bewertungen der Söhne innerhalb der Bäume disjunkte Partitionen der Bewertungen der Väter bilden.

3.1.2 Blockclusterbäume

Mit den im letzten Abschnitt 3.1.1 definierten Clusterbäumen können die Zeilen- und Spaltenindizes einer Matrix $M \in \mathbb{R}^{\mathcal{I} \times \mathcal{J}}$ hierarchisch in $\mathcal{T}_{\mathcal{I}}$ und $\mathcal{T}_{\mathcal{J}}$ organisiert werden. Um die Teilmatrizen $M|_{\hat{t} \times \hat{s}}$ mit $t \in \mathcal{T}_{\mathcal{I}}$ und $s \in \mathcal{T}_{\mathcal{J}}$ mit einer einzigen Struktur hierarchisch ordnen zu können, wird die Definition der Blockclusterbäume herangezogen.

Definition 3.8 (Blockclusterbaum) Sei \mathcal{T} ein bewerteter Baum und seien $\mathcal{T}_{\mathcal{I}}$ und $\mathcal{T}_{\mathcal{J}}$ Clusterbäume. Dann ist \mathcal{T} genau dann ein Blockclusterbaum für $\mathcal{T}_{\mathcal{I}}$ und $\mathcal{T}_{\mathcal{J}}$, wenn folgendes erfüllt ist:

- Für die Wurzel $\varrho \in \mathcal{T}$ gilt $\varrho = (\text{root}(\mathcal{T}_{\mathcal{I}}), \text{root}(\mathcal{T}_{\mathcal{J}}))$.
- Für jeden Knoten $b \in \mathcal{T}$ existieren Cluster $t \in \mathcal{T}_{\mathcal{I}}$ und $s \in \mathcal{T}_{\mathcal{J}}$ mit $b = (t, s)$ und für deren Bewertungen gilt $\hat{b} = \hat{t} \times \hat{s}$.
- Für $b = (t, s) \in \mathcal{T}$ mit $\text{sons}(b) \neq \emptyset$ gilt

$$\text{sons}(b) = \begin{cases} \{t\} \times \text{sons}(s) & \text{falls } \text{sons}(t) = \emptyset, \text{sons}(s) \neq \emptyset \\ \text{sons}(t) \times \{s\} & \text{falls } \text{sons}(t) \neq \emptyset, \text{sons}(s) = \emptyset \\ \text{sons}(t) \times \text{sons}(s) & \text{sonst} \end{cases}.$$

Notation 3.9 (zu Blockclusterbäumen) Ein Blockclusterbaum \mathcal{T} für die Clusterbäume $\mathcal{T}_{\mathcal{I}}$ und $\mathcal{T}_{\mathcal{J}}$ wird mit $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$ bezeichnet. Ein Knoten $b = (t, s) \in \mathcal{T}_{\mathcal{I} \times \mathcal{J}}$ heißt Blockcluster und $t \in \mathcal{T}_{\mathcal{I}}$ Zeilencluster und $s \in \mathcal{T}_{\mathcal{J}}$ Spaltencluster von b . Die Menge aller Blätter eines Blockclusterbaums $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$ wird als $\mathcal{L}_{\mathcal{I} \times \mathcal{J}}$ geschrieben.

Die Rechtfertigung für die Notation $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$ für einen Blockclusterbaum liefert das folgende Lemma.

Lemma 3.10 (Blockclusterbaum als Clusterbaum) Sei $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$ ein Blockclusterbaum für $t \in \mathcal{T}_{\mathcal{I}}$ und $s \in \mathcal{T}_{\mathcal{J}}$. Dann ist $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$ selbst auch wieder ein Clusterbaum für die Indexmenge $\mathcal{I} \times \mathcal{J}$.

3 Hierarchische Matrizen

Beweis. [7, Corollary 3.14] ■

Mit Hilfe des nächsten Lemmas ist das Zwischenziel einer hierarchisch aufgebauten Matrixpartition von $M \in \mathbb{R}^{\mathcal{I} \times \mathcal{J}}$ über eine Partition ihrer Indexmenge $\mathcal{I} \times \mathcal{J}$ erreicht.

Lemma 3.11 (Blattpartition) *Die Bewertungen der Blätter eines Blockclusterbaums $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$ bilden eine disjunkte Zerlegung der Indexmenge $\mathcal{I} \times \mathcal{J}$, das heißt*

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

Beweis. [7, Corollary 3.15] ■

Jetzt wird noch entschieden werden müssen, welche Teilmatrizen, also welche durch den Blockclusterbaum $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$ definierten Matrixblöcke $M|_{\hat{t} \times \hat{s}}$ mit $b = (t, s) \in \mathcal{T}_{\mathcal{I} \times \mathcal{J}}$, als zulässige und welche als unzulässige Blöcke gekennzeichnet werden. Die zulässigen Matrixblöcke sollten sich später durch Niedrigrangmatrizen gut approximieren lassen, während die unzulässigen Blöcke möglichst klein sein sollten, da sie als vollbesetzte Matrizen in herkömmlicher Weise gespeichert werden sollen.

Ein Kriterium, das diese Entscheidung trifft, nennt man Zulässigkeitsbedingung.

Definition 3.12 (Zulässigkeitsbedingung) *Eine Abbildung*

$$\text{Adm} : \mathcal{T}_{\mathcal{I}} \times \mathcal{T}_{\mathcal{J}} \rightarrow \{\text{zulässig, nicht zulässig}\}$$

wird als Zulässigkeitsbedingung bezeichnet.

Definition 3.13 (Zulässiger Blockclusterbaum) *Sei $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$ ein Blockclusterbaum und Adm eine Zulässigkeitsbedingung. Falls für alle $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}$*

$$b \text{ zulässig} \quad \text{oder} \quad \text{sons}(t) = \emptyset \quad \text{oder} \quad \text{sons}(s) = \emptyset$$

gilt, heißt $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$ zulässig bezüglich Adm. Ansonsten heißt $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$ nicht zulässig oder unzulässig bezüglich Adm.

Falls für alle $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}$

$$b \text{ zulässig} \quad \text{oder} \quad \text{sons}(t) = \emptyset = \text{sons}(s)$$

gilt, heißt $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$ streng zulässig bezüglich Adm. Ansonsten heißt $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$ nicht streng zulässig bezüglich Adm.

Für die Mengen der zulässigen und unzulässigen Blätter eines Blockclusterbaums sind folgende Bezeichnungen üblich:

Definition 3.14 (Nahfeld und Fernfeld) *Sei $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$ ein Blockclusterbaum für die Clusterbäume $\mathcal{T}_{\mathcal{I}}$ und $\mathcal{T}_{\mathcal{J}}$. Sei Adm eine Zulässigkeitsbedingung auf $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$. Die Menge aller zulässigen Blätter von $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$*

$$\mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+ := \{b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}} : b \text{ zulässig bzgl. Adm}\}$$

wird auch Fernfeld genannt, während die Menge aller unzulässigen Blätter von $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$

$$\mathcal{L}_{\mathcal{I} \times \mathcal{J}}^- := \{b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}} : b \text{ nicht zulässig bzgl. Adm}\}$$

als Nahfeld bezeichnet wird.

Die Begriffe des Nah- und Fernfeldes sind motiviert durch Begriffe aus der Physik, wo sie im Zusammenhang mit elektrischen und magnetischen Feldern die Auswirkungen in unterschiedlichen Entfernungen von der Quelle beschreiben. Diese Quelle ist im Zusammenhang mit den \mathcal{H} -Matrizen zum Beispiel als die Singularität zu verstehen, in deren Nähe eine Niedrigrangapproximation nicht möglich ist, weiter davon entfernt, also im Fernfeld, jedoch schon.

3.2 \mathcal{H} -Matrizen

Mit Hilfe der im vorherigen Abschnitt 3.1 definierten Matrixpartition mittels eines zulässigen Blockclusterbaums kann nun die Struktur einer hierarchischen Matrix definiert werden. Dabei sollen die Teilmatrizen, die den zulässigen Blättern des Blockclusterbaums entsprechen, durch Niedrigrangmatrizen approximiert und als solche abgespeichert werden, während die Teilmatrizen, die den nicht zulässigen Blättern entsprechen, in herkömmlicher Weise als vollbesetzte Matrizen gespeichert werden.

Niedrigrangmatrizen mit einem Rang von höchstens k werden in der Definition der Rang- k -Matrizen zusammengefasst.

Definition 3.15 (Rang- k -Matrix) Sei $k \in \mathbb{N}_0$. Seien $t \in \mathcal{T}_{\mathcal{I}}$ und $s \in \mathcal{T}_{\mathcal{J}}$. Eine Matrix $M \in \mathbb{R}^{\hat{t} \times \hat{s}}$ heißt Rang- k -Matrix, falls Matrizen

$$A \in \mathbb{R}^{\hat{t} \times k} \quad \text{und} \quad B \in \mathbb{R}^{\hat{s} \times k}$$

existieren, so dass sich M in der faktorisierten Weise

$$M = AB^* \tag{3.1}$$

schreiben lässt. B^* bezeichnet dabei die adjungierte Matrix von B . Die Menge aller Rang- k -Matrizen zu $\hat{t} \times \hat{s}$ wird mit $\mathcal{R}(k, t, s)$ oder kurz mit $\mathcal{R}(k)$ bezeichnet.

Damit hat jede Rang- k -Matrix höchstens den Rang k und jede Matrix mit einem Rang $\leq k$ kann als Rang- k -Matrix geschrieben werden.

Definition 3.16 (Rang- k -Darstellung) Sei $M \in \mathcal{R}(k)$ eine Rang- k -Matrix. Dann wird das Matrixpaar (A, B) aus (3.1) auch faktorisierte Rang- k -Darstellung genannt.

Wenn eine Rang- k -Matrix M durch ihre Rang- k -Darstellung, also in Form der Matrizen A und B , repräsentiert wird, müssen nur $k(\#\hat{t} + \#\hat{s})$ Koeffizienten gespeichert werden gegenüber der üblichen Darstellung als vollbesetzte Matrix mit $(\#\hat{t}) \cdot (\#\hat{s})$ Koeffizienten. Offensichtlich ist die Rang- k -Darstellung wesentlich effizienter, wenn $k \ll \min\{\#\hat{t}, \#\hat{s}\}$ gilt.

Mit Hilfe der Rang- k -Matrizen und einem zulässigen Blockclusterbaum lassen sich nun hierarchische Matrizen formal einführen.

3 Hierarchische Matrizen

Definition 3.17 (\mathcal{H} -Matrix) Sei $k \in \mathbb{N}_0$ und sei $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$ ein zulässiger Blockclusterbaum. Eine Matrix $M \in \mathbb{R}^{\mathcal{I} \times \mathcal{J}}$ heißt \mathcal{H} -Matrix, falls für alle Blockcluster des Fernfeldes $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+$ die entsprechende Teilmatrix $M|_{\hat{t} \times \hat{s}} \in \mathcal{R}(k, t, s)$ eine Rang- k -Matrix ist. Dann wird k als lokaler Rang von M bezeichnet.

Die Menge aller \mathcal{H} -Matrizen mit lokalem Rang k zum Blockclusterbaum $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$ wird als $\mathcal{H}(k, \mathcal{T}_{\mathcal{I} \times \mathcal{J}})$ oder kurz $\mathcal{H}(k)$ geschrieben, wenn der zugrundeliegende Blockclusterbaum eindeutig ist.

Damit lässt sich eine speichergünstige Repräsentation im Rechner für die \mathcal{H} -Matrizen formulieren:

Definition 3.18 (\mathcal{H} -Matrix-Darstellung) Sei $M \in \mathcal{H}(k, \mathcal{T}_{\mathcal{I} \times \mathcal{J}})$ eine hierarchische Matrix mit lokalem Rang $k \in \mathbb{N}_0$. Dann existieren Familien von Matrizen

$$(A_b)_{b \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+} \quad \text{mit } A_b \in \mathbb{R}^{\hat{t} \times k} \quad \text{und} \quad (B_b)_{b \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+} \quad \text{mit } B_b \in \mathbb{R}^{s \times k}$$

so, dass sich die Matrizen des Fernfeldes effizient darstellen lassen durch

$$M|_{\hat{t} \times \hat{s}} = A_b B_b^* \quad \text{für alle } b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+.$$

Die Matrizen des Nahfeldes werden in der Familie $(N_b)_{b \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^-}$ von vollbesetzten Matrizen zusammengefasst:

$$N_b := M|_{\hat{t} \times \hat{s}} \quad \text{für alle } b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^-.$$

Damit ist die \mathcal{H} -Matrix-Darstellung von M gegeben durch das Tripel

$$((A_b)_{b \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+}, (B_b)_{b \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+}, (N_b)_{b \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^-}).$$

Die genaue Untersuchung des Speicherbedarfs einer hierarchischen Matrix soll hier nicht weiter ausgeführt werden, kann jedoch in [7, Abschnitt 3.4] nachgelesen werden. An dieser Stelle sei nur die folgende grobe Aussage über den Speicheraufwand festgehalten:

Bemerkung 3.19 (Speicherbedarf) Unter den üblichen Voraussetzungen liegt der Speicherbedarf für eine hierarchische Matrix $M \in \mathcal{H}(k, \mathcal{T}_{\mathcal{I} \times \mathcal{J}})$ mit lokalem Rang k und $\max\{\#\mathcal{I}, \#\mathcal{J}\} \leq n \in \mathbb{N}$ in der Komplexitätsklasse

$$\mathcal{O}(nk \log n).$$

3.3 Anwendung auf Integraloperatoren

Die Technik der hierarchischen Matrizen soll nun auf das in Kapitel 2.1 vorgestellte Problem der Integraloperatoren angewendet werden. Dazu wird die vollbesetzte Matrix aus (2.4) mit den Koeffizienten

$$G_{i,j} := \int_{\Gamma} \int_{\Gamma} \varphi_i(x) g(x, y) \varphi_j(y) \, d\Gamma(x) \, d\Gamma(y)$$

durch eine hierarchische Matrix approximiert. Um dieses zu erreichen, muss zunächst entschieden werden, welche Teilmatrizen von G sich gut durch Niedrigrangmatrizen approximieren lassen, also wie man in diesem Fall zu einem zulässigen Blockclusterbaum kommt, so wie er in Abschnitt 3.1.2 eingeführt wurde. Der folgende Abschnitt 3.3.1 beschäftigt sich mit dieser Fragestellung.

Die nächste Aufgabe besteht dann darin, die zu den zulässigen Blättern des Blockclusterbaums gehörigen Teilmatrizen durch Rang- k -Matrizen, wie in Definition 3.15 definiert, zu approximieren. Durch die enge Verbindung zwischen der Matrix G und der Kernfunktion g aus (2.3) kann dieses über eine Kernapproximation bewerkstelligt werden.

Um dieses Vorgehen der Kernapproximation möglichst klar zu machen, wird in Abschnitt 3.3.2 vorerst in allgemeiner Form auf die Konstruktion einer sogenannten entarteten Kernfunktion eingegangen, die die Grundlage für die Interpolations-Methode in Abschnitt 4.1 und später auch für unser auf der Green'schen Formel basierendes Verfahren bildet, das in den Kapiteln 5 und 8 ausführlich behandelt wird.

3.3.1 Clustering und Zulässigkeit

Die Basis $(\varphi_i)_{i \in \mathcal{I}}$ für die Galerkin-Diskretisierung aus Kapitel 2 wird in der Praxis meistens so gewählt, dass der Träger $\text{supp}(\varphi_i)$ der Basisfunktionen

$$\text{supp}(\varphi_i) := \overline{\{x \in \Gamma : \varphi_i(x) \neq 0\}} \quad \text{für alle } i \in \mathcal{I}$$

relativ klein ist. Um nun Clusterbäume konstruieren zu können, eignet sich die Wahl von *charakteristischen Punkten* $x_i \in \text{supp}(\varphi_i)$ für alle $i \in \mathcal{I}$ innerhalb der Träger.

Definition 3.20 (Familie charakteristischer Punkte) *Eine Familie $(x_i)_{i \in \mathcal{I}}$ im \mathbb{R}^d mit $x_i \in \text{supp}(\varphi_i)$ heißt Familie charakteristischer Punkte für die Indexmenge \mathcal{I} .*

Mit Hilfe dieser Punkte lässt sich eine Clustering vornehmen: Jedem Cluster $t \in \mathcal{T}_{\mathcal{I}}$ wird dabei eine Menge dieser charakteristischen Punkte $\{x_i : i \in \hat{t}\} \subseteq \mathbb{R}^d$ zugeordnet. Falls diese Menge noch zu groß ist, wird sie disjunkt in Teilmengen zerlegt. Dadurch entstehen dann die Söhne des Clusters. Diese Zerlegung wird rekursiv weiter fortgeführt bis die Cluster eine bestimmte Größe erreicht oder unterschritten haben. Diese Größe sei wie folgt definiert:

Definition 3.21 (Maximale Blattgröße) *Seien $\mathcal{T}_{\mathcal{I}}$ und $\mathcal{T}_{\mathcal{J}}$ Clusterbäume und $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$ der zugehörige Blockclusterbaum. Dann wird die Größe $n_{\mathcal{L}}(\mathcal{T}_{\mathcal{I}}) \in \mathbb{N}$ mit*

$$n_{\mathcal{L}}(\mathcal{T}_{\mathcal{I}}) := \max_{t \in \mathcal{L}_{\mathcal{I}}} \{\#\hat{t}\}$$

als maximale Blattgröße bezeichnet. Auf den Blockclusterbaum $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$ übertragen, ergibt sich dann

$$n_{\mathcal{L}}(\mathcal{T}_{\mathcal{I} \times \mathcal{J}}) := \max\{n_{\mathcal{L}}(\mathcal{T}_{\mathcal{I}}), n_{\mathcal{L}}(\mathcal{T}_{\mathcal{J}})\}$$

als maximale Blattgröße.

3 Hierarchische Matrizen

Um nun eine Zulässigkeitsbedingung auf dem so konstruierten Blockclusterbaum formulieren zu können, werden die folgenden beiden Mengen eingeführt:

$$\tau := \bigcup_{i \in \hat{t}} \text{supp}(\varphi_i) \quad \text{für alle } t \in \mathcal{T}_{\mathcal{I}}$$

und

$$\sigma := \bigcup_{i \in \hat{s}} \text{supp}(\varphi_i) \quad \text{für alle } s \in \mathcal{T}_{\mathcal{J}}.$$

Zulässigkeitsbedingungen werden üblicherweise über den Durchmesser und den Abstand dieser Mengen definiert. Da es aufwändiger Berechnungen bedarf, diese Werte für alle Cluster zueinander zu berechnen, werden diese Mengen in achsenparallele Quader „verpackt“, für die diese Werte, Durchmesser und Abstand, wesentlich einfacher zu berechnen sind. Diese Quader erhalten die englische Bezeichnung der *Bounding Box*.

Definition 3.22 (Bounding Box) Seien $a_i, b_i \in \mathbb{R}$ mit $a_i < b_i$ für alle $i \in \{1, \dots, d\}$. Ein achsenparalleler Quader $\mathcal{B}_\tau \subset \mathbb{R}^d$ mit

$$\mathcal{B}_\tau = [a_1, b_1] \times \dots \times [a_d, b_d]$$

und der überdeckenden Eigenschaft

$$\tau \subseteq \mathcal{B}_\tau$$

wird Bounding Box genannt.

Für diese Bounding Boxen lassen sich verschiedene Abstands- und Durchmesser-Begriffe formulieren. Im weiteren Verlauf dieser Arbeit wird von den zwei folgenden Definitionen dazu Gebrauch gemacht.

Definition 3.23 (Abstand und Durchmesser) Seien $\mathcal{B}_\tau = [a_1, b_1] \times \dots \times [a_d, b_d]$ und $\mathcal{B}_\sigma = [c_1, d_1] \times \dots \times [c_d, d_d]$ Bounding Boxen im \mathbb{R}^d . Dann sind folgende Durchmesser dazu definiert:

$$\begin{aligned} \text{diam}_2(\mathcal{B}_\tau) &:= \sup\{\|x - y\|_2 : x, y \in \mathcal{B}_\tau\} \\ &= \left(\sum_{i=1}^d \text{diam}([a_i, b_i])^2 \right)^{\frac{1}{2}} \end{aligned}$$

und

$$\begin{aligned} \text{diam}_{\max}(\mathcal{B}_\tau) &:= \sup\{\|x - y\|_\infty : x, y \in \mathcal{B}_\tau\} \\ &= \max\{|b_i - a_i| : i \in \{1, \dots, d\}\}. \end{aligned}$$

Die Abstände zwischen Bounding Boxen sind entsprechend definiert durch

$$\begin{aligned} \text{dist}_2(\mathcal{B}_\tau, \mathcal{B}_\sigma) &:= \inf\{\|x - y\|_2 : x \in \mathcal{B}_\tau, y \in \mathcal{B}_\sigma\} \\ &= \left(\sum_{i=1}^d \text{dist}([a_i, b_i], [c_i, d_i])^2 \right)^{\frac{1}{2}} \end{aligned}$$

und

$$\begin{aligned} \text{dist}_{\max}(\mathcal{B}_\tau, \mathcal{B}_\sigma) &:= \inf\{\|x - y\|_\infty : x \in \mathcal{B}_\tau, y \in \mathcal{B}_\sigma\} \\ &= \max\{\text{dist}([a_i, b_i], [c_i, d_i]) : i \in \{1, \dots, d\}\}. \end{aligned}$$

Mit Hilfe dieser Definition sind typische Zulässigkeitsbedingungen wie zum Beispiel

$$\text{diam}(\mathcal{B}_\tau) \leq \text{dist}(\mathcal{B}_\tau, \mathcal{B}_\sigma)$$

sowohl für diam_2 und dist_2 als auch für diam_{\max} und dist_{\max} einfach zu prüfen. Die genaue Formulierung einer Zulässigkeitsbedingung ist dabei allerdings abhängig von der Art des gewählten Approximationsverfahrens, wie im weiteren Verlauf dieser Arbeit ersichtlich wird.

Eine wichtige Eigenschaft der Kernfunktion, die später vor allem in Fehleranalysen eine entscheidende Rolle spielt, soll an dieser Stelle noch genannt werden:

Definition 3.24 (Asymptotische Glattheit) *Eine Kernfunktion $g : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ ist asymptotisch glatt, falls es Konstanten $C_{\text{as}} > 0$ und $c_0 \geq 1$, sowie einen Singularitätsgrad $\sigma \in \mathbb{N}$ so gibt, dass für alle $n \in \mathbb{N}_0$, alle $x, y \in \mathbb{R}^d$ mit $x \neq y$ und alle Richtungen $p \in \mathbb{R}^d \times \mathbb{R}^d$ die Ungleichung*

$$|\partial_p^n g(x, y)| \leq C_{\text{as}} \frac{(\sigma - 1 + n)! c_0^n \|p\|^n}{\|x - y\|^{\sigma+n}} \quad (3.2)$$

erfüllt ist.

Die Funktion heißt ebenso asymptotisch glatt, falls (3.2) für $\sigma = 0$ und alle $n \in \mathbb{N}$, alle $x, y \in \mathbb{R}^d$ mit $x \neq y$ und alle Richtungen $p \in \mathbb{R}^d \times \mathbb{R}^d$ gilt. $\sigma = 0$ entspricht dabei logarithmischen Singularitäten.

Mit dem, wie hier beschrieben, konstruierten zulässigen Blockclusterbaum kann nun die Frage nach der Niedrigrangapproximation über die Kernfunktion für die zulässigen Blockcluster im nächsten Abschnitt 3.3.2 geklärt werden.

3.3.2 Entartete Kernfunktion

Das Ziel der Kernapproximation ist hier die Trennung der Variablen so, dass für die zulässigen Blattcluster $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}$ Rang- k -Darstellungen gefunden werden können, deren Matrizen $A_b \in \mathbb{R}^{\hat{t} \times k}$ und $B_b \in \mathbb{R}^{\hat{s} \times k}$ mit

$$G|_{\hat{t} \times \hat{s}} \approx A_b B_b^*$$

über die approximierte Kernfunktion definiert werden können.

Definition 3.25 (Entartete Kernfunktion) *Es sei eine Kernfunktion $\tilde{g} : \Omega \times \Omega \rightarrow \mathbb{R}$ gegeben. Falls es Funktionen $a_\nu : \tau \rightarrow \mathbb{R}$ und $b_\nu : \sigma \rightarrow \mathbb{R}$ für alle $\nu \in \{1, \dots, k\}$ gibt mit*

$$\tilde{g}(x, y) = \sum_{\nu=1}^k a_\nu(x) b_\nu(y) \quad \text{für alle } x \in \tau, y \in \sigma,$$

so heißt \tilde{g} entartete Kernfunktion.

3 Hierarchische Matrizen

Wenn man die Kernfunktion g aus (2.3) durch eine entartete Kernfunktion \tilde{g} approximieren kann, also

$$g(x, y) \approx \tilde{g}(x, y) \quad \text{für alle } x \in \tau, y \in \sigma, \quad (3.3)$$

so folgt für die Matrix $G|_{\hat{t} \times \hat{s}}$ mit

$$G_{i,j} := \int_{\Gamma} \int_{\Gamma} \varphi_i(x) g(x, y) \varphi_j(y) \, d\Gamma(x) \, d\Gamma(y)$$

dank der entscheidenden Eigenschaft der Variablentrennung

$$\begin{aligned} G_{i,j} &\approx \int_{\Gamma} \int_{\Gamma} \varphi_i(x) \tilde{g}(x, y) \varphi_j(y) \, d\Gamma(x) \, d\Gamma(y) \\ &= \sum_{\nu=1}^k \int_{\Gamma} \varphi_i(x) a_{\nu}(x) \, d\Gamma(x) \int_{\Gamma} \varphi_j(y) b_{\nu}(y) \, d\Gamma(y) \end{aligned}$$

für alle $i \in \hat{t}$ und $j \in \hat{s}$. Mit den durch

$$A_{i,\nu} := \int_{\Gamma} \varphi_i(x) a_{\nu}(x) \, d\Gamma(x) \quad \text{für alle } i \in \hat{t}, \nu \in \{1, \dots, k\}$$

und

$$B_{j,\nu} := \int_{\Gamma} \varphi_j(y) b_{\nu}(y) \, d\Gamma(y) \quad \text{für alle } j \in \hat{s}, \nu \in \{1, \dots, k\}$$

definierten Matrizen $A \in \mathbb{R}^{\hat{t} \times k}$ und $B \in \mathbb{R}^{\hat{s} \times k}$ ist die gewünschte Rang- k -Matrix

$$G|_{\hat{t} \times \hat{s}} \approx AB^*$$

erreicht. Für die eigentliche Kernapproximation (3.3) gibt es verschiedene Verfahren, wie zum Beispiel die Interpolation, die in Abschnitt 4.1 vorgestellt wird oder natürlich der Algorithmus, der im Zentrum dieser Arbeit steht und in den späteren Kapiteln beschrieben und untersucht wird.

3.4 \mathcal{H}^2 -Matrizen

Die Motivation hinter den \mathcal{H}^2 -Matrizen besteht darin, einen noch geringeren Speicherbedarf als für die einfachen \mathcal{H} -Matrizen bei sehr großskaligen Problemen zu erhalten und gleichzeitig damit auch noch effizienter rechnen zu können. Dabei bleiben die Clusterstrategien größtenteils dieselben wie im \mathcal{H} -Matrix-Fall, nur die Zulässigkeit der Blockclusterbäume muss etwas verschärft werden. Ansonsten besteht der Unterschied nur in einer modifizierten Niedrigrang-Darstellung für die Matrizen des Fernfeldes. Um diese Darstellung definieren zu können, muss zunächst der Begriff der Clusterbasis eingeführt werden.

Definition 3.26 (Clusterbasis) Sei $\mathcal{T}_{\mathcal{I}}$ ein Clusterbaum. Eine Familie von Matrizen

$$V = (V_t)_{t \in \mathcal{T}_{\mathcal{I}}} \quad \text{mit } V_t \in \mathbb{R}^{\hat{t} \times k} \text{ f\u00fcr alle } t \in \mathcal{T}_{\mathcal{I}}$$

he\u00dft Clusterbasis vom Rang k f\u00fcr den Clusterbaum $\mathcal{T}_{\mathcal{I}}$. Falls noch eine weitere Familie von Matrizen $(E_t)_{t \in \mathcal{I}}$ existiert mit der Eigenschaft

$$V_t|_{\hat{t} \times k} = V_{t'} E_{t'} \quad \text{f\u00fcr alle } t \in \mathcal{T}_{\mathcal{I}} \text{ und } t' \in \text{sons}(t),$$

so nennt man V eine geschachtelte Clusterbasis und die Matrizen $(E_t)_{t \in \mathcal{T}_{\mathcal{I}}}$ sind die korrespondierenden Transfermatrizen.

Nach dieser Definition lassen sich bei geschachtelten Clusterbasen alle V_t f\u00fcr Nicht-Blattcluster $t \in \mathcal{T}_{\mathcal{I}} \setminus \mathcal{L}_{\mathcal{I}}$ mit Hilfe der Transfermatrizen aus den Nachfahren rekonstruieren. Also m\u00fcssen nur f\u00fcr die Blattcluster $t \in \mathcal{L}_{\mathcal{I}}$ die Matrizen $V_t \in \mathbb{R}^{\hat{t} \times k}$ gespeichert werden und ansonsten nur die kleineren Matrizen $E_t \in \mathbb{R}^{k \times k}$ f\u00fcr $t \in \mathcal{T}_{\mathcal{I}}$. Es ergibt sich damit folgende Darstellung:

Definition 3.27 (Geschachtelte Clusterbasis-Darstellung) Sei $V = (V_t)_{t \in \mathcal{T}_{\mathcal{I}}}$ eine geschachtelte Clusterbasis. Dann ist die geschachtelte Clusterbasis-Darstellung von V gegeben durch das Paar

$$((V_t)_{t \in \mathcal{L}_{\mathcal{I}}}, (E_t)_{t \in \mathcal{T}_{\mathcal{I}}}).$$

Mit dem Begriff der geschachtelten Clusterbasis und mit einem streng zul\u00e4ssigen Blockclusterbaum im Sinne von Definition 3.13 kann jetzt eine \mathcal{H}^2 -Matrix definiert werden.

Definition 3.28 (\mathcal{H}^2 -Matrix) Sei $k \in \mathbb{N}_0$ und sei $\mathcal{T}_{\mathcal{I} \times \mathcal{J}}$ ein streng zul\u00e4ssiger Blockclusterbaum. Seien V und W geschachtelte Clusterbasen vom Rang k f\u00fcr die Clusterb\u00e4ume $\mathcal{T}_{\mathcal{I}}$ und $\mathcal{T}_{\mathcal{J}}$. Eine Matrix $M \in \mathbb{R}^{\mathcal{I} \times \mathcal{J}}$ hei\u00dft \mathcal{H}^2 -Matrix mit Zeilenbasis V und Spaltenbasis W , falls f\u00fcr alle Blockcluster des Fernfeldes $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+$ eine Matrix $S_b \in \mathbb{R}^{k \times k}$ existiert, so dass die entsprechende Teilmatrix $M|_{\hat{t} \times \hat{s}}$ in faktorisierter Weise dargestellt werden kann als

$$M|_{\hat{t} \times \hat{s}} = V_t S_b W_s^*.$$

Dann wird k als lokaler Rang von M bezeichnet und die $(S_b)_{b \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+}$ werden Kopplungsmatrizen genannt.

Die Menge aller \mathcal{H}^2 -Matrizen mit Zeilenbasis V , Spaltenbasis W und lokalem Rang k wird als $\mathcal{H}^2(k, \mathcal{T}_{\mathcal{I} \times \mathcal{J}}, V, W)$ geschrieben.

Auch hier kann man wieder eine speicherg\u00fcnstige Darstellung formulieren:

Definition 3.29 (\mathcal{H}^2 -Matrix-Darstellung) Sei $M \in \mathcal{H}^2(k, \mathcal{T}_{\mathcal{I} \times \mathcal{J}}, V, W)$. Dann existiert eine Familie von Matrizen

$$(S_b)_{b \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+} \quad \text{mit } S_b \in \mathbb{R}^{k \times k}$$

3 Hierarchische Matrizen

und geschachtelte Clusterbasis Darstellungen

$$((V_t)_{t \in \mathcal{L}_{\mathcal{I}}}, (E_t)_{t \in \mathcal{T}_{\mathcal{I}}}) \quad \text{für die Clusterbasis } V$$

und

$$((W_s)_{s \in \mathcal{L}_{\mathcal{J}}}, (F_s)_{s \in \mathcal{T}_{\mathcal{J}}}) \quad \text{für die Clusterbasis } W$$

so, dass sich die Matrizen des Fernfeldes effizient darstellen lassen durch

$$M|_{\hat{t} \times \hat{s}} = V_t S_b W_s^* \quad \text{für alle } b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+.$$

Die Matrizen des Nahfeldes werden in der Familie $(N_b)_{\mathcal{L}_{\mathcal{I} \times \mathcal{J}}^-}$ von vollbesetzten Matrizen zusammengefasst:

$$N_b := M|_{\hat{t} \times \hat{s}} \quad \text{für alle } b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^-.$$

Damit ist die \mathcal{H}^2 -Matrix-Darstellung von M gegeben durch das Tupel

$$((V_t)_{t \in \mathcal{L}_{\mathcal{I}}}, (E_t)_{t \in \mathcal{T}_{\mathcal{I}}}, (W_s)_{s \in \mathcal{L}_{\mathcal{J}}}, (F_s)_{s \in \mathcal{T}_{\mathcal{J}}}, (S_b)_{b \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+}, (N_b)_{\mathcal{L}_{\mathcal{I} \times \mathcal{J}}^-}).$$

Wie bereits bei den \mathcal{H} -Matrizen soll auch hier keine detaillierte Untersuchung des Speicherbedarfs stattfinden, sondern wie zuvor verwiesen werden auf [7, Abschnitte 3.5 und 3.6]. Und wieder soll nur eine grobe Angabe über den Speicheraufwand zitiert werden:

Bemerkung 3.30 (Speicherbedarf) *Der Speicherbedarf für eine \mathcal{H}^2 -Matrix $M \in \mathcal{H}^2(k, \mathcal{T}_{\mathcal{I} \times \mathcal{J}}, V, W)$ mit lokalem Rang k und $\max\{\#\mathcal{I}, \#\mathcal{J}\} \leq n \in \mathbb{N}$ liegt in der Komplexitätsklasse*

$$\mathcal{O}(nk).$$

Der Speicher wächst also nur linear mit der Anzahl der Unbekannten und ist daher von optimaler Ordnung.

Bemerkung 3.31 (uniforme \mathcal{H} -Matrizen) *Es existiert noch eine Art Vorstufe der \mathcal{H}^2 -Matrizen, die sogenannten uniformen \mathcal{H} -Matrizen. Diese Matrizen sind aufgebaut wie \mathcal{H}^2 -Matrizen, jedoch mit dem Unterschied, dass ihre Clusterbasen nicht geschachtelt sind. Dadurch ist der Vorteil des wesentlich geringeren Speicherbedarfs gegenüber \mathcal{H} -Matrizen hinfällig, da für uniforme \mathcal{H} -Matrizen jedes der $(V_t)_{t \in \mathcal{T}_{\mathcal{I}}}$ gespeichert werden muss, weil es in diesem Fall keine Transfermatrizen gibt.*

4 Derzeit verwendete Verfahren

Wie bereits in Kapitel 1 erwähnt, existiert eine Vielzahl an Verfahren, die sich damit beschäftigen, die quadratische Komplexität von vollbesetzten Matrizen zu vermeiden.

In Kapitel 3 wurde nun eine mächtige Struktur dazu vorgestellt, mit der so etwas möglich ist. Aber auch im Kontext von hierarchischen Matrizen gibt es verschiedene Methoden, die Niedrigrangmatrizen für die zulässigen Blockcluster des Fernfeldes zu konstruieren.

In diesem Kapitel werden drei der bedeutendsten dieser Methoden erklärt. Diese Verfahren finden zur Zeit auch in der Praxis Anwendung und bieten sich daher ideal für numerische Vergleichsrechnungen zu dem in Kapitel 5 und 8 neu eingeführten Verfahren an.

Das erste analytische Verfahren, das in Abschnitt 4.1 beschrieben wird, arbeitet wie später auch die Green-Quadratur-Methode in Kapitel 5 mit einer Kernapproximation und zwar basierend auf einer Lagrange-Interpolation.

In Abschnitt 4.2 wird dann die, ebenfalls analytische und altbewährte, Multipol-Methode erklärt.

Neuere algebraische Verfahren wie die in Abschnitt 4.3 vorgestellte Kreuzapproximation nutzen direkt die Matrixeinträge für eine Approximation, anstatt über die Kernfunktion zu gehen.

Die Vorteile von matrixbasierten Verfahren und Kernapproximations-Verfahren vereinigen die sogenannten Hybrid-Verfahren wie die hybride Kreuzapproximation, die hier nicht näher ausgeführt wird, oder auch das in Kapitel 8 vorgestellte Green-Hybrid-Verfahren.

4.1 Interpolation

Ein robustes und oft verwendetes Verfahren zur Konstruktion der Niedrigrangmatrizen für die hierarchischen Matrizen ist die Interpolation. Dabei gelangt man mit Hilfe von Lagrange-Tensor-Interpolationsoperatoren zu den gewünschten, wie in Abschnitt 3.3.2 definierten, entarteten Kernfunktionen. Diese Interpolations-Technik in ihrer Anwendung auf Integraloperatoren wurde vorgestellt in [9].

4.1.1 Tensor-Interpolationsoperatoren

Das Ziel ist es, eine Lagrange-Interpolation auf mehrdimensionalen Gebieten, wie den Bounding Boxen \mathcal{B}_τ aus Definition 3.22, durchführen zu können.

Da $\mathcal{B}_\tau = [a_1, b_1] \times \cdots \times [a_d, b_d]$ mit $a_i < b_i$ für alle $i \in \{1, \dots, d\}$ als Tensorgebiet definiert ist, liegt es nahe, auch auf diese Weise den Interpolationsoperator darauf zu

4 Derzeit verwendete Verfahren

konstruieren. Dazu wird zunächst ein Interpolationsoperator \mathcal{I} auf dem Referenzintervall $[-1, 1]$ definiert. Die Interpolations-Stützstellen seien dabei mit $(\xi_\nu)_{\nu=0}^m$ bezeichnet, so dass die *Lagrange-Polynome* gegeben sind durch

$$\mathcal{L}_\nu(x) := \prod_{\mu=0, \mu \neq \nu}^m \frac{x - \xi_\mu}{\xi_\nu - \xi_\mu} \quad \text{für alle } x \in [-1, 1]. \quad (4.1)$$

Dann ist \mathcal{I} definiert durch

$$\mathcal{I} : C([-1, 1]) \rightarrow \Pi_m, \quad f \mapsto \sum_{\nu=0}^m f(\xi_\nu) \mathcal{L}_\nu, \quad (4.2)$$

wobei Π_m der Raum der Polynome eines Grades von höchstens m ist. Mit der Transformation

$$\Phi_{[a,b]} : [-1, 1] \rightarrow [a, b], \quad x \mapsto \frac{b-a}{2}x + \frac{b+a}{2}$$

kann mit \mathcal{I} ein Operator $\mathcal{I}_{[a,b]}$ auf beliebigen Intervallen $[a, b]$ definiert werden mit

$$\mathcal{I}_{[a,b]} : C([a, b]) \rightarrow \Pi_m, \quad f \mapsto (\mathcal{I}[f \circ \Phi_{[a,b]}]) \circ \Phi_{[a,b]}^{-1}.$$

Mit $\xi_{[a,b],\nu} := \Phi_{[a,b]}(\xi_\nu)$ und

$$\mathcal{L}_{[a,b],\nu}(x) := \prod_{\mu=0, \mu \neq \nu}^m \frac{x - \xi_{[a,b],\mu}}{\xi_{[a,b],\nu} - \xi_{[a,b],\mu}} \quad \text{für alle } x \in [a, b]$$

sowie der Tatsache, dass

$$\mathcal{L}_\nu(\xi_\nu) = \delta_{\nu\mu} = \mathcal{L}_{[a,b],\nu}(\xi_{[a,b],\nu}) \quad \text{für alle } \nu, \mu \in \{0, \dots, m\},$$

kann $\mathcal{I}_{[a,b]}$ wieder in der aus (4.2) vertrauten Form

$$\mathcal{I}_{[a,b]}[f] = \sum_{\nu=0}^m f(\xi_{[a,b],\nu}) \mathcal{L}_{[a,b],\nu} \quad \text{für alle } f \in C[a, b]$$

geschrieben werden.

Als weiterer Zwischenschritt sei der Operator $\mathcal{I}_{\mathcal{B}_\tau, \iota}$ auf der Bounding Box \mathcal{B}_τ definiert durch

$$\mathcal{I}_{\mathcal{B}_\tau, \iota} : C(\mathcal{B}_\tau) \rightarrow C(\mathcal{B}_\tau), \quad f \mapsto \sum_{\nu=0}^m f(x_1, \dots, x_{\iota-1}, \xi_{[a_\iota, b_\iota], \nu}, x_{\iota+1}, \dots, x_d) \mathcal{L}_{[a_\iota, b_\iota], \nu}(x_\iota).$$

Durch Kombination solcher Operatoren erhält man schließlich den gesuchten Tensor-Interpolationsoperator

$$\mathcal{I}_{\mathcal{B}_\tau} := \mathcal{I}_{\mathcal{B}_\tau, 1} \circ \dots \circ \mathcal{I}_{\mathcal{B}_\tau, d}.$$

Mit der Multiindexmenge $M = \{0, \dots, m\}^d$ können die transformierten d -dimensionalen Stützstellen geschrieben werden als

$$\xi_{\tau, \nu} := (\xi_{[a_1, b_1], \nu_1}, \dots, \xi_{[a_d, b_d], \nu_d}) \quad \text{für alle } \nu \in M$$

und die entsprechenden Lagrange-Polynome nehmen die Gestalt

$$\mathcal{L}_{\tau, \nu}(x) := \mathcal{L}_{[a_1, b_1], \nu_1}(x_1) \dots \mathcal{L}_{[a_d, b_d], \nu_d}(x_d) \quad \text{für alle } \nu \in M, x \in \mathcal{B}_\tau$$

an, so dass die vertraute Darstellung auch für $\mathcal{I}_{\mathcal{B}_\tau}$ formuliert werden kann:

$$\mathcal{I}_{\mathcal{B}_\tau}[f] := \sum_{\nu \in M} f(\xi_{\tau, \nu}) \mathcal{L}_{\tau, \nu} \quad \text{für alle } f \in C(\mathcal{B}_\tau).$$

4.1.2 Kernapproximation

Um mit den in Abschnitt 4.1.1 eingeführten Tensor-Interpolationsoperatoren eine entartete Kernfunktion zu erhalten, muss garantiert werden, dass sich die Kernfunktion auf dem entsprechenden Gebiet auch vernünftig approximieren lässt. Wie schon in Abschnitt 3.3 beschrieben wurde, kann man dieses mit Hilfe einer Zulässigkeitsbedingung erreichen, die dafür sorgt, dass die Funktion auf den entsprechenden Teilgebieten hinreichend glatt ist. Im Falle der Interpolation sieht die konkrete Zulässigkeitsbedingung für zwei Bounding Boxen \mathcal{B}_τ und \mathcal{B}_σ , wie in [9] beschrieben, wie folgt aus:

$$\min\{\text{diam}(\mathcal{B}_\tau), \text{diam}(\mathcal{B}_\sigma)\} \leq 2\eta \text{dist}(\mathcal{B}_\tau, \mathcal{B}_\sigma) \quad (4.3)$$

mit einem Parameter $\eta > 0$, der dieser Art von Zulässigkeit auch die Bezeichnung *η -Zulässigkeit* verleiht.

Die Kernapproximation $\tilde{g}(x, y)$ für $x \in \tau$ und $y \in \sigma$ wird nun abhängig von den Durchmessern der Bounding Boxen \mathcal{B}_τ und \mathcal{B}_σ definiert:

$$\tilde{g}(x, y) := \begin{cases} \mathcal{I}_{\mathcal{B}_\tau}[g(\cdot, y)](x) & \text{falls } \text{diam}(\mathcal{B}_\tau) \leq \text{diam}(\mathcal{B}_\sigma) \\ \mathcal{I}_{\mathcal{B}_\sigma}[g(x, \cdot)](y) & \text{sonst.} \end{cases}$$

Das bedeutet im ersten Fall $\text{diam}(\mathcal{B}_\tau) \leq \text{diam}(\mathcal{B}_\sigma)$, dass die Kernapproximation durch $\tilde{g}(x, y) = \mathcal{I}_{\mathcal{B}_\tau}[g(\cdot, y)](x)$ gegeben ist, welche eine entartete Kernfunktion

$$\tilde{g}(x, y) = \sum_{\nu \in M} g(\xi_{\tau, \nu}, y) \mathcal{L}_{\tau, \nu}(x) \quad (4.4)$$

im Sinne von Definition 3.25 ist.

Wenn man nun für η -zulässige Blöcke $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+$ den Matrixblock $G|_{\hat{t} \times \hat{s}}$ mit \tilde{g} anstelle der Kernfunktion aufstellt, erhält man

$$G_{i,j} \approx \tilde{G}_{i,j} := \int_{\Gamma} \int_{\Gamma} \varphi_i(x) \tilde{g}(x, y) \varphi_j(y) \, d\Gamma(x) \, d\Gamma(y)$$

4 Derzeit verwendete Verfahren

und mit

$$A_{i,\nu} := \int_{\Gamma} \varphi_i(x) \mathcal{L}_{\tau,\nu}(x) \, d\Gamma(x) \quad \text{und} \quad B_{j,\nu} := \int_{\Gamma} \varphi_j(y) g(\xi_{\tau,\nu}, y) \, d\Gamma(y)$$

für alle $i \in \hat{t}, j \in \hat{s}$ und $\nu \in M$ die Niedrigrangapproximation

$$G|_{\hat{t} \times \hat{s}} \approx \tilde{G} = AB^* .$$

Damit hat \tilde{G} einen Rang von höchstens der Mächtigkeit der Multiindexmenge M , also

$$\text{rank}(\tilde{G}) \leq \#M = m^d . \quad (4.5)$$

Auf analoge Weise kommt man auch im Falle von $\text{diam}(\mathcal{B}_{\sigma}) < \text{diam}(\mathcal{B}_{\tau})$ über

$$\tilde{g}(x, y) = \sum_{\nu \in M} g(x, \xi_{\sigma,\nu}) \mathcal{L}_{\sigma,\nu}(y)$$

zu einem Rang von höchstens m^d für \tilde{G} .

4.1.3 Interpolationsfehler

An dieser Stelle soll eine kurze Übersicht über die Fehleranalyse für diese Art der Kernapproximation dargelegt werden. Dazu sei zunächst ein wichtiger Begriff eingeführt:

Definition 4.1 (Stabilitätskonstante) Falls für ein $\Lambda \in \mathbb{R}$

$$\|\mathcal{I}[f]\|_{\infty,[-1,1]} \leq \Lambda \|f\|_{\infty,[-1,1]} \quad \text{für alle } f \in C[-1,1]$$

gilt, so wird Λ Stabilitätskonstante des Interpolationsoperators \mathcal{I} genannt.

Die beste Stabilitätskonstante für einen Lagrange-Interpolationsoperator ist durch die Lebesgue-Zahl gegeben.

Lemma 4.2 (Lebesgue-Zahl) Die Lebesgue-Zahl des Interpolationsoperators \mathcal{I} ist definiert durch

$$\Lambda := \max_{x \in [-1,1]} \left\{ \sum_{\nu=0}^m |\mathcal{L}_{\nu}(x)| \right\}$$

und stellt die kleinstmögliche Stabilitätskonstante von \mathcal{I} dar.

Beweis. [8, Lemma 3.6] ■

Für die Tschebyscheff-Interpolationsoperatoren erhält man eine besonders kleine Lebesgue-Zahl. Daher wird der Interpolationsfehler auch für diesen Fall hier beschrieben. Zur Definition der Tschebyscheff-Interpolation werden erst einmal die Tschebyscheff-Polynome benötigt.

Definition 4.3 (Tschebyscheff-Polynome) Sei $n \in \mathbb{N}_0$ und $x \in \mathbb{R}$. Dann ist das n -te Tschebyscheff-Polynom definiert durch

$$T_n(x) := \begin{cases} 1 & \text{falls } n = 0 \\ x & \text{falls } n = 1 \\ 2x T_{n-1}(x) - T_{n-2}(x) & \text{sonst.} \end{cases}$$

Zur Definition der Tschebyscheff-Interpolation sind die folgenden Eigenschaften der Tschebyscheff-Polynome interessant.

Lemma 4.4 (Eigenschaften der Tschebyscheff-Polynome) Sei $n \in \mathbb{N}_0$. Für das n -te Tschebyscheff-Polynom gilt

$$T_n(x) = \cos(n \arccos(x)) \quad \text{für alle } x \in [-1, 1]$$

und damit $\|T_n(x)\|_{\infty, [-1, 1]} = 1$.

T_n besitzt n reelle einfache Nullstellen t_ν , die im Intervall $[-1, 1]$ liegen und von folgender Form sind:

$$t_\nu := \cos\left(\pi \frac{2\nu + 1}{2n}\right) \quad \text{für alle } \nu \in \{0, \dots, n-1\}.$$

Beweis. folgt aus den Additionstheoremen und den Eigenschaften des Cosinus. ■

Wenn man die Nullstellen des $(m+1)$ -ten Tschebyscheff-Polynoms T_{m+1} als Stützstellen $\xi_\nu = \cos(\pi \frac{2\nu+1}{2m})$ mit $\nu \in \{0, \dots, m\}$ für die Interpolation wählt, spricht man von *Tschebyscheff-Interpolationspunkten* und mit so gewählten Stützstellen heißt \mathcal{I} dann auch *Tschebyscheff-Interpolationsoperator*.

Wie bereits erwähnt und in [28] gezeigt, erhält man für diesen Operator eine besonders kleine Lebesgue-Zahl

$$\Lambda := \frac{2}{\pi} \log(m+1) \quad \text{für alle } m \in \mathbb{N},$$

die sich durch

$$\Lambda \leq m+1 \quad \text{für alle } m \in \mathbb{N} \tag{4.6}$$

leicht abschätzen lässt. Andere Interpolationspunkte können nur zu unwesentlich kleineren Stabilitätskonstanten führen, womit die Tschebyscheff-Interpolationspunkte nahe an der Optimalität liegen.

Damit kann nun der Fehler für die Tensor-Tschebyscheff-Interpolation zitiert werden:

Satz 4.5 (Fehler der Tensor-Interpolation) Sei \mathcal{I} der Tschebyscheff-Interpolationsoperator, sei $\mathcal{I}_{[a,b]}$ der zugehörige transformierte Interpolationsoperator auf dem Intervall $[a, b]$ und sei $\mathcal{I}_{\mathcal{B}_\tau}$ der d -dimensionale Interpolationsoperator

$$\mathcal{I}_{\mathcal{B}_\tau} := \mathcal{I}_{\mathcal{B}_{\tau,1}} \circ \dots \circ \mathcal{I}_{\mathcal{B}_{\tau,d}}.$$

4 Derzeit verwendete Verfahren

Dann gilt für den Approximationsfehler

$$\|f - \mathcal{I}_{\mathcal{B}_\tau}[f]\|_{\infty, \mathcal{B}_\tau} \leq 2 \sum_{\iota=1}^d \Lambda^{\iota-1} \left(\frac{b_\iota - a_\iota}{4} \right)^{m+1} \frac{\|\partial_\iota^{m+1} f\|_{\infty, \mathcal{B}_\tau}}{(m+1)!} \quad (4.7)$$

für alle $f \in C^{m+1}(\mathcal{B}_\tau)$ und für die Stabilität

$$\|\mathcal{I}_{\mathcal{B}_\tau}[f]\|_{\infty, \mathcal{B}_\tau} \leq \Lambda^d \|f\|_{\infty, \mathcal{B}_\tau}.$$

Beweis. [8, Lemma 3.9] ■

Wenn die Zulässigkeitsbedingung (4.3) für Bounding Boxen \mathcal{B}_τ und \mathcal{B}_σ greift und $\text{diam}(\mathcal{B}_\tau) \leq \text{diam}(\mathcal{B}_\sigma)$ gilt, kann die Kernfunktion g dort wie in (4.4) approximiert werden. Mit der asymptotischen Glattheit (3.2) von g und der Fehlerabschätzung für die Tensor-Interpolation (4.7) folgt für den Fehler der Kernapproximation

$$\begin{aligned} \|g - \tilde{g}\|_{\infty, \mathcal{B}_\tau \times \mathcal{B}_\sigma} &\leq 2 \sum_{\iota=1}^d \Lambda^{\iota-1} \left(\frac{b_\iota - a_\iota}{4} \right)^{m+1} \frac{1}{(m+1)!} C_{\text{as}} \frac{(\sigma - 1 + m + 1)! c_0^{m+1}}{\|x - y\|^{\sigma+m+1}} \\ &\leq 2C_{\text{as}} \frac{(m + \sigma)!}{(m + 1)!} \sum_{\iota=1}^d \Lambda^{\iota-1} \left(\frac{\text{diam}(\mathcal{B}_\tau)}{4} \right)^{m+1} \frac{c_0^{m+1}}{\text{dist}(\mathcal{B}_\tau, \mathcal{B}_\sigma)^{\sigma+m+1}} \\ &= \frac{2C_{\text{as}}}{\text{dist}(\mathcal{B}_\tau, \mathcal{B}_\sigma)^\sigma} \frac{(m + \sigma)!}{(m + 1)!} \left(\frac{c_0 \text{diam}(\mathcal{B}_\tau)}{4 \text{dist}(\mathcal{B}_\tau, \mathcal{B}_\sigma)} \right)^{m+1} \sum_{\iota=1}^d \Lambda^{\iota-1}. \end{aligned}$$

Mit (4.6) und der Definition

$$P(m) := 2C_{\text{as}} \frac{(m + \sigma)!}{(m + 1)!} d (m + 1)^{d-1}$$

folgt weiter

$$\|g - \tilde{g}\|_{\infty, \mathcal{B}_\tau \times \mathcal{B}_\sigma} \leq \frac{P(m)}{\text{dist}(\mathcal{B}_\tau, \mathcal{B}_\sigma)^\sigma} \left(\frac{c_0 \text{diam}(\mathcal{B}_\tau)}{4 \text{dist}(\mathcal{B}_\tau, \mathcal{B}_\sigma)} \right)^{m+1}$$

und mit der η -Zulässigkeit (4.3) schließlich

$$\begin{aligned} \|g - \tilde{g}\|_{\infty, \mathcal{B}_\tau \times \mathcal{B}_\sigma} &\leq \frac{P(m)}{\text{dist}(\mathcal{B}_\tau, \mathcal{B}_\sigma)^\sigma} \left(\frac{c_0 2 \eta \text{dist}(\mathcal{B}_\tau, \mathcal{B}_\sigma)}{4 \text{dist}(\mathcal{B}_\tau, \mathcal{B}_\sigma)} \right)^{m+1} \\ &= \frac{P(m)}{\text{dist}(\mathcal{B}_\tau, \mathcal{B}_\sigma)^\sigma} \left(\frac{c_0 \eta}{2} \right)^{m+1}. \end{aligned}$$

Mit der Wahl von $\eta < \frac{2}{c_0}$ dominiert damit die exponentielle Konvergenz über das polynomielle Wachstum von $P(m)$.

4.2 Multipol-Methode

Im Gegensatz zur Interpolation aus Abschnitt 4.1 und der Kreuzapproximation aus Abschnitt 4.3 ist die Multipol-Methode kein Bestandteil eines der hier neu vorgestellten Algorithmen, sondern wird lediglich zu numerischen Vergleichen herangezogen. Als eines der bekanntesten Verfahren sollte die Multipol-Methode in einer Übersicht wie in diesem Kapitel nicht fehlen, wird allerdings nur kurz und speziell auf das hier relevante Problem zugeschnitten vorgestellt und für den interessierten Leser wird auf die entsprechende Literatur verwiesen.

Wie die bereits vorgestellte Interpolations-Methode und später die Green-basierten Methoden beruht auch die Multipol-Methode auf der Konstruktion einer entarteten Kernfunktion. Dazu werden Bäume benutzt, die im Prinzip genauso aufgebaut werden, wie die in Abschnitt 3.1.1 kennengelernten Clusterbäume. Wie in der eng verwandten *Panel Clustering Methode* [23, 24, 31, 15] findet das Erstellen dieser Bäume über sogenannte Panels statt, also Geradenstücke im zweidimensionalen und Dreiecke im dreidimensionalen auf dem Rand des betrachteten Gebiets.

Das Ziel der Clusterung ist dasselbe, wie bei den anderen vorgestellten Verfahren: Es soll eine Partitionierung der Indexmenge und eine Aufteilung in Nah- und Fernfeld-Blockcluster erreicht werden. Damit wird garantiert, dass die Kernfunktion auf dem Fernfeld hinreichend glatt ist und damit, im Kontext der Multipol Methode, gut approximierbar durch eine abgebrochene Taylorreihe.

Ursprünglich wurde die Multipol-Methode für die Simulation von Vielteilchen-Systemen vorgestellt [20, 21], woher auch die Beobachtung stammt, dass in einiger Entfernung von der Quelle das Gravitationsfeld glatt ist und sich dadurch gut in einer komprimierteren Form darstellen lassen sollte. Diese Beobachtung führte schließlich zu der Unterteilung in die Begriffe Nah- und Fernfeld, die ursprünglich aus der Elektrodynamik stammen.

Es existiert aber auch eine Vielzahl an Arbeiten, welche die Anwendung der Multipol-Methode auf die hier relevanten Integralgleichungen behandeln, allen voran [30, 26, 19].

Weitere Abarten des Verfahrens, die unter der Bezeichnung *multipole methods without multipoles* zusammengefasst werden können, sind beschrieben in [1, 25, 35, 36] und seien hier nur am Rande erwähnt.

Wie bereits angekündigt, wird das Multipol-Verfahren an dieser Stelle nur anhand des zweidimensionalen Falls des Modellproblems aus Abschnitt 2.1 erklärt. Die zu approximierende Kernfunktion ist also gegeben durch

$$g(x, y) = -\frac{1}{2\pi} \log \|x - y\| \quad \text{für alle } x, y \in \mathbb{R}^2.$$

In Abschnitt 4.2.1 wird die Multipol-Entwicklung dazu vorgestellt, in Abschnitt 4.2.2 die Clusterung der in Kapitel 10 genutzten Vergleichs-Implementierung beschrieben und in Abschnitt 4.2.3 wird kurz die Fehlerabschätzung skizziert.

4.2.1 Multipol-Entwicklung

Die Multipol-Entwicklung ist im Grunde eine Taylor-Entwicklung und der Übergang zur Approximation findet durch ein *Abschneiden* der Taylorreihe an ihrem k -ten Glied statt.

Um zu dieser abgebrochenen Taylorreihe zu gelangen, wird zunächst die Kernfunktion ins Komplexe übertragen. Genauer werden die Punkte x und y aus der reellen Zahlenebene \mathbb{R}^2 als komplexe Zahlen aus \mathbb{C} aufgefasst in folgender Weise:

$$\begin{aligned} x &:= (x_1, x_2)^\top \in \mathbb{R}^2 & \rightsquigarrow & z_x := x_1 + ix_2 \in \mathbb{C} \\ y &:= (y_1, y_2)^\top \in \mathbb{R}^2 & \rightsquigarrow & z_y := y_1 + iy_2 \in \mathbb{C} \end{aligned}$$

so, dass die Kernfunktion g die Form

$$g(x, y) := -\frac{1}{2\pi} \operatorname{Re}(\log(z_x - z_y)) \quad (4.8)$$

mit dem komplexen Logarithmus annimmt. Da g translationsinvariant ist, kann das Problem so verschoben werden, dass das Zentrum des Zeilenclusters im Ursprung liegt, um damit eine Taylor-Entwicklung zu ermöglichen. Dazu werden die Notationen aus Kapitel 3 verwendet: Seien $x \in \tau$ und $y \in \sigma$ und der Mittelpunkt der Bounding Box $\mathcal{B}_\tau = [a_1, b_1] \times \dots \times [a_d, b_d]$ sei definiert durch

$$\operatorname{center}(\mathcal{B}_\tau) := \begin{pmatrix} \frac{b_1+a_1}{2} \\ \vdots \\ \frac{b_d+a_d}{2} \end{pmatrix}. \quad (4.9)$$

Die Verschiebung findet nun um $c_t := \operatorname{center}(\mathcal{B}_\tau)$ statt. Dann kann für $|z_x - c_t| < |z_y - c_t|$ die Taylor-Entwicklung von $z_x \mapsto \log(z_x - z_y)$ im Entwicklungspunkt c_t angegeben werden durch

$$\log(z_x - z_y) = \log(c_t - z_y) - \sum_{\nu=1}^{\infty} \frac{1}{\nu} \left(\frac{z_x - c_t}{z_y - c_t} \right)^\nu. \quad (4.10)$$

Wenn man diese Entwicklung beim k -ten Glied abschneidet und dann in (4.8) einsetzt, erhält man die gewünschte Kernapproximation

$$g(x, y) \approx \tilde{g}(x, y) := \frac{1}{2\pi} \operatorname{Re} \left((-1) \log(c_t - z_y) + \sum_{\nu=1}^{k-1} \frac{(z_x - c_t)^\nu}{\nu} \frac{1}{(z_y - c_t)^\nu} \right),$$

die mit den Definitionen

$$a_\nu(z_x) := \begin{cases} -1 & \text{falls } \nu = 0 \\ \frac{(z_x - c_t)^\nu}{\nu} & \text{sonst} \end{cases} \quad \text{und} \quad b_\nu(z_y) := \begin{cases} \log(c_t - z_y) & \text{falls } \nu = 0 \\ \frac{1}{(z_y - c_t)^\nu} & \text{sonst} \end{cases}$$

die kompaktere Form

$$\tilde{g}(x, y) = \frac{1}{2\pi} \operatorname{Re} \left(a_0(z_x) b_0(z_y) + \sum_{\nu=1}^{k-1} a_\nu(z_x) b_\nu(z_y) \right) = \sum_{\nu=0}^{k-1} \frac{\operatorname{Re}(a_\nu(z_x) b_\nu(z_y))}{2\pi}$$

annimmt. Die Matrixapproximation von $G|_{\hat{t} \times \hat{s}}$ mit den Koeffizienten aus (2.4) ist demnach

$$G_{i,j} \approx \tilde{G}_{i,j} := \sum_{\nu=0}^{k-1} \int_{\Gamma} \int_{\Gamma} \varphi_i(x) \frac{\operatorname{Re}(a_{\nu}(z_x)b_{\nu}(z_y))}{2\pi} \varphi_j(y) \, dx \, dy \quad (4.11)$$

für alle $i, j \in \mathcal{I}$. Da für zwei komplexe Zahlen $u := u_1 + iu_2$ und $v := v_1 + iv_2$

$$\begin{aligned} \operatorname{Re}(u \cdot v) &= \operatorname{Re}((u_1 + iu_2)(v_1 + iv_2)) \\ &= \operatorname{Re}(u_1v_1 + iu_2v_1 + iu_1v_2 - u_2v_2) \\ &= u_1v_1 - u_2v_2 \\ &= \begin{pmatrix} u_1 & u_2 \end{pmatrix} \begin{pmatrix} v_1 \\ -v_2 \end{pmatrix} \end{aligned}$$

gilt, also

$$\operatorname{Re}(a_{\nu}(z_x)b_{\nu}(z_y)) = \begin{pmatrix} \operatorname{Re}(a_{\nu}(z_x)) & \operatorname{Im}(a_{\nu}(z_x)) \end{pmatrix} \begin{pmatrix} \operatorname{Re}(b_{\nu}(z_y)) \\ -\operatorname{Im}(b_{\nu}(z_y)) \end{pmatrix}$$

ist, muss die Niedrigrang-Darstellung für \tilde{G} etwas anders definiert werden als bei der Interpolation in Abschnitt 4.1.2. Mit

$$A_{i,\nu} := \begin{cases} \frac{1}{2\pi} \int_{\Gamma} \varphi_i(x) \operatorname{Re}(a_{\nu}(z_x)) dx & \text{für } \nu \in \{0, \dots, k-1\} \\ \frac{1}{2\pi} \int_{\Gamma} \varphi_i(x) \operatorname{Im}(a_{\nu-k}(z_x)) dx & \text{für } \nu \in \{k, \dots, 2k-1\} \end{cases}$$

und

$$B_{j,\nu} := \begin{cases} \int_{\Gamma} \operatorname{Re}(b_{\nu}(z_y)) \varphi_j(y) dy & \text{für } \nu \in \{0, \dots, k-1\} \\ - \int_{\Gamma} \operatorname{Im}(b_{\nu-k}(z_y)) \varphi_j(y) dy & \text{für } \nu \in \{k, \dots, 2k-1\} \end{cases}$$

folgt dann $\tilde{G} = AB^*$, wobei durch die komplexe Sichtweise eine Rangverdoppelung stattgefunden hat, so dass $A \in \mathbb{R}^{\hat{t} \times 2k}$ und $B \in \mathbb{R}^{\hat{s} \times 2k}$ gelten, also $G|_{\hat{t} \times \hat{s}}$ durch eine Rang- $2k$ -Matrix approximiert wird.

4.2.2 Clusterung

Wie bereits am Anfang dieses Kapitels 4.2 erwähnt wurde, geschieht die Clusterung für die Multipol-Methode im Wesentlichen genauso, wie für die bereits vorgestellten Verfahren für hierarchische Matrizen. Der Unterschied besteht in der benutzten Zulässigkeitsbedingung.

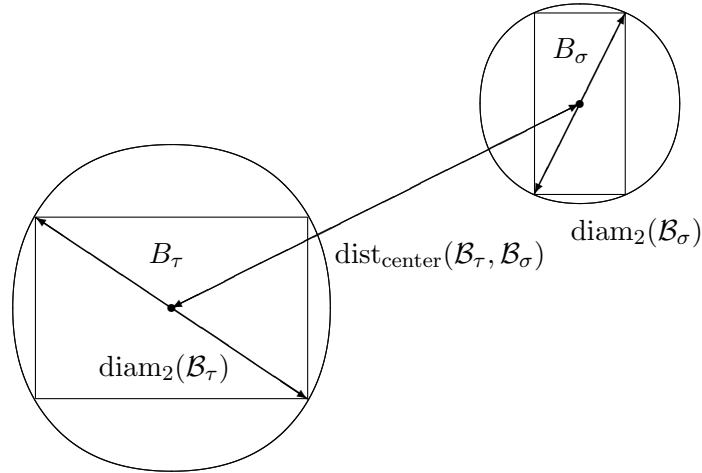
Da das Multipol-Verfahren durch die Taylor-Entwicklung mit kreisförmigen Gebieten arbeitet, ist es naheliegend, die *Kreis-Zulässigkeitsbedingung* zu nutzen, die definiert ist durch

$$\max\{\operatorname{diam}_2(\mathcal{B}_{\tau}), \operatorname{diam}_2(\mathcal{B}_{\sigma})\} \leq \operatorname{dist}_{\text{center}}(\mathcal{B}_{\tau}, \mathcal{B}_{\sigma}) - \frac{1}{2} (\operatorname{diam}_2(\mathcal{B}_{\tau}) + \operatorname{diam}_2(\mathcal{B}_{\sigma})) \quad (4.12)$$

wobei

$$\operatorname{dist}_{\text{center}}(\mathcal{B}_{\tau}, \mathcal{B}_{\sigma}) := \|\operatorname{center}(\mathcal{B}_{\tau}) - \operatorname{center}(\mathcal{B}_{\sigma})\|_2$$

mit $\operatorname{center}(\mathcal{B}_{\tau})$ dem Mittelpunkt der Bounding Box \mathcal{B}_{τ} wie in (4.9) definiert. In der folgenden Abbildung sind nochmal die Größen aus der Zulässigkeitsbedingung verdeutlicht.



4.2.3 Fehlerbetrachtung

Der Fehler des Multipol-Verfahrens leitet sich direkt aus dem Fehler der Taylor-Entwicklung ab. Für die Entwicklung aus (4.10), abgebrochen beim k -ten Glied, ist das Taylor-Restglied $R_{k-1}(z_x)$ gegeben durch

$$R_{k-1}(z_x) = \frac{1}{k} \left(\frac{z_x - c_t}{z_y - \xi} \right)^k$$

für ein $\xi \in \mathbb{C}$ mit $|\xi - c_t| \leq \frac{1}{2} \text{diam}(B_\tau)$. Da $x \in \tau$, gilt auch $|z_x - c_t| \leq \frac{1}{2} \text{diam}(B_\tau)$. Und da $y \in \sigma$, ist $|\xi - z_y| \geq \text{dist}_{\text{center}}(B_\tau, B_\sigma) - \frac{1}{2}(\text{diam}(B_\tau) + \text{diam}(B_\sigma))$. Durch die Zulässigkeitsbedingung (4.12) folgt damit

$$\begin{aligned} |z_x - c_t| &\leq \frac{1}{2} \text{diam}(B_\tau) \\ &\leq \frac{1}{2} \left(\text{dist}_{\text{center}}(B_\tau, B_\sigma) - \frac{1}{2}(\text{diam}(B_\tau) + \text{diam}(B_\sigma)) \right) \\ &\leq \frac{1}{2} |\xi - z_y|, \end{aligned}$$

womit die exponentielle Konvergenz des Restgliedes klar wird.

Für Genaueres zur Fehleranalyse der Multipol-Methode sei verwiesen auf [20].

4.3 Kreuzapproximation

Bisher wurden hier nur analytische Verfahren zur Berechnung von Niedrigrangapproximationen für hierarchische Matrizen vorgestellt. Alle diese Verfahren beruhen auf der Konstruktion einer entarteten Kernfunktion. Es existiert aber auch eine jüngere Klasse an algebraischen Verfahren, die auf der sogenannten Kreuzapproximation aufbauen und

die nur mit den Einträgen der Matrix selbst arbeiten, um daraus eine Niedrigrangapproximation zu berechnen.

Diese Kreuzapproximations-Verfahren können im Prinzip ohne spezielle Kenntnis der Kernfunktion angewendet werden. Dabei wird die Matrix wieder, wie in Abschnitt 3.1, in Blöcke partitioniert und jeder Block unabhängig von den übrigen behandelt.

Aus k Zeilen und k Spalten eines zulässigen Matrixblocks wird dann eine Rang- k -Matrix konstruiert. Entscheidend dabei ist allerdings, wie diese Zeilen und Spalten ausgewählt werden. Für einigermaßen sinnvolle Pivot-Strategien wird die Kenntnis benötigt, dass die Matrix aus einer asymptotisch glatten Kernfunktion entstanden ist.

Es gibt verschiedene Möglichkeiten für die Pivotierung, wobei in den meisten Fällen mit zunehmender Effizienz die Zuverlässigkeit immer weniger garantiert ist.

Diesen Effekt kann man vermeiden, indem man von dem rein algebraischen Verfahren zu einem hybriden Verfahren übergeht, das dann wieder auf die Kernfunktion zurückgreift. Die Rede ist von der hybriden Kreuzapproximation [10], welche aus einer Kombination aus Interpolation und Kreuzapproximation besteht und damit die Zuverlässigkeit der Interpolation mit der Effizienz der Kreuzapproximation verbindet. Diese hybride Version soll nur der Vollständigkeit halber erwähnt werden und wird hier nicht näher betrachtet.

Das Kreuzapproximations-Verfahren, das in dieser Arbeit von Bedeutung ist, ist die *adaptive Kreuzapproximation*, kurz mit ACA bezeichnet (aus der englischen Bezeichnung *adaptive cross approximation*) [2, 4, 3].

Dieses Verfahren ist nicht nur Teil des in Kapitel 8 vorgestellten Green-Hybrid-Verfahrens, sondern dient auch als Vergleichs-Verfahren in den numerischen Experimenten in Kapitel 7 und 10. Im Folgenden wird daher in Abschnitt 4.3.1 eine kurze Bemerkung zur Matrixstruktur gemacht, in Abschnitt 4.3.2 das Verfahren erklärt und in Abschnitt 4.3.3 eine kurze Zusammenfassung zur Fehleranalyse angegeben.

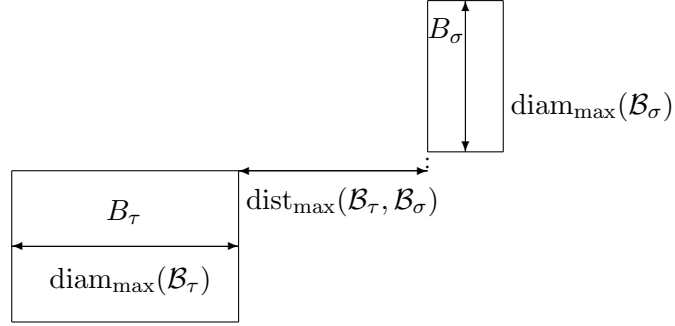
4.3.1 Matrixstruktur

Da die Kreuzapproximation direkt für hierarchische Matrizen gedacht ist, findet die Partitionierung der Matrix mit Hilfe von Cluster- und Blockclusterbäumen über die Indexmengen exakt so statt, wie in Kapitel 3 beschrieben wurde. Im Rahmen dieser Arbeit wird für die adaptive Kreuzapproximation die *max-Zulässigkeitsbedingung* verwendet, die auch für die Green-basierten Verfahren genutzt wird und die definiert ist durch

$$\max\{\text{diam}_{\max}(\mathcal{B}_\tau), \text{diam}_{\max}(\mathcal{B}_\sigma)\} \leq \text{dist}_{\max}(\mathcal{B}_\tau, \mathcal{B}_\sigma) \quad (4.13)$$

mit $\text{diam}_{\max}(\mathcal{B}_\tau)$ und $\text{dist}_{\max}(\mathcal{B}_\tau, \mathcal{B}_\sigma)$ aus Definition 3.23. Diese Zulässigkeitsbedingung ist nicht die optimale Wahl, führt jedoch zu besseren Ergebnissen als die Kreiszulässigkeitsbedingung 4.12, die der Bedingung [29, (3.15)] entspricht. In den numerischen Experimenten in [29] verwenden die Autoren eine Zulässigkeitsbedingung, die der Optimalität näher kommt, die allerdings noch unveröffentlicht ist.

Auch hier sollen nochmal die Größen aus der Zulässigkeitsbedingung in der folgenden Abbildung verdeutlicht werden.



4.3.2 Adaptive Kreuzapproximation

Die Idee bei der adaptiven Kreuzapproximation ist, einen Fehlerwert vorzugeben und dann ausgehend von einem Rang von 1 eine Folge von Approximationen mit zunehmendem Rang zu konstruieren, die dann abgebrochen wird, wenn der verbleibende Fehler dem vorgegebenem Wert entspricht oder ihn unterschreitet.

Sei $G|_{\hat{t} \times \hat{s}}$ mit $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+$ der durch eine Rang- k -Matrix zu approximierende Matrixblock. In jedem Schritt k müssen zunächst Pivot-Indizes $i_k \in \hat{t}$ und $j_k \in \hat{s}$ gewählt werden. Diese können durch das betragsgrößte Element des Matrixblocks

$$|G_{i_k, j_k}| = \max_{i \in \hat{t}, j \in \hat{s}} |G_{i, j}|$$

bestimmt werden. Dazu müsste allerdings der gesamte Block im Speicher vorliegen und durchsucht werden. Diese Art der Pivotsuche nennt man daher *vollständige Pivottierung*.

Da aber für eine Rang- k -Approximation des Matrixblocks eigentlich nur k Zeilen und k Spalten benötigt werden, wird hier die sogenannte *partielle Pivottierung* benutzt. Dazu wird der Pivot-Index $i_k \in \hat{t}$ zum Beispiel als Index der ersten Zeile gewählt, und davon abhängig dann $j_k \in \hat{s}$ durch

$$|G_{i_k, j_k}| = \max_{j \in \hat{s}} |G_{i_k, j}|$$

bestimmt. Für Strategien zur Wahl von i_k sei zum Beispiel auf [3] verwiesen.

Mit den Pivot-Indizes i_k und j_k können nun die Einträge der Matrizen $A \in \mathbb{R}^{\hat{t} \times k}$ und $B \in \mathbb{R}^{\hat{s} \times k}$ für die Rang- k -Darstellung

$$G|_{\hat{t} \times \hat{s}} \approx AB^*$$

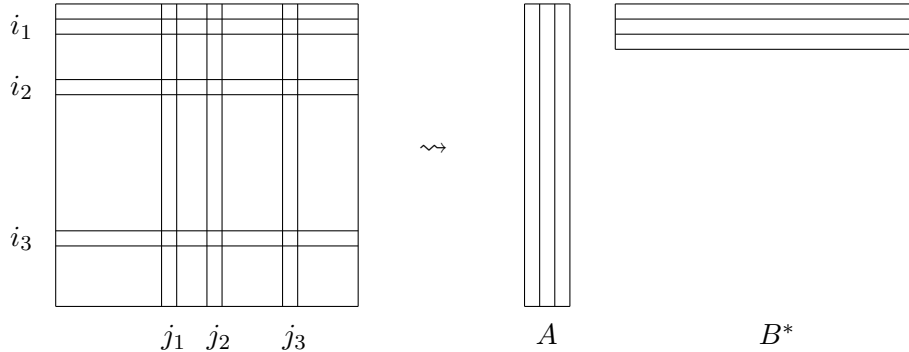
berechnet werden durch

$$A_{i, k} = \frac{\hat{A}_{i, k}}{\hat{A}_{i_k, k}} \quad \text{mit} \quad \hat{A}_{i, k} = G_{i, j_k} - \sum_{l=1}^{k-1} A_{i, l} B_{j_k, l} \quad \text{für alle } i \in \hat{t}$$

und

$$B_{j, k} = G_{i_k, j} - \sum_{l=1}^{k-1} B_{j, l} A_{i_k, l} \quad \text{für alle } j \in \hat{s}.$$

Die folgende Abbildung zeigt ein Schema für die ersten drei Schritte der Berechnung.



Bei vollständiger Pivotierung werden jeweils die Pivot-Zeilen und Spalten zu der Rang- k -Matrix hinzugefügt und der Restfehler berechnet, indem die Rang- k -Matrix von der bisherigen Matrix abgezogen wird. Da dieses so lange wiederholt wird, bis der Restfehler dem vorgegebenem entspricht, ist der letztendliche Rang k nicht von vornherein bekannt. Vorteil an diesem Verfahren ist aber, dass so garantiert ist, dass der Restfehler klein genug ist.

Bei partieller Pivotierung liegt der Fehler nicht mehr explizit vor, da nur die für die Approximation benötigten Zeilen und Spalten berechnet werden, der gesamte originale Matrixblock aber gar nicht vorliegt. In diesem Fall muss der Fehler abgeschätzt werden.

4.3.3 Fehlerabschätzung

Um die Darstellung des Fehlers möglichst einfach zu halten, seien, ohne Beschränkung der Allgemeinheit, die Pivot-Indizes der Reihe nach in den ersten Zeilen beziehungsweise Spalten angesiedelt. Genauer heißt das für die Pivot-Indizes $i_l = j_l = l$ für $l = 1, \dots, k$.

Sei die Matrix $G := G|_{\hat{t} \times \hat{s}}$ in vier Teilmatrizen aufgeteilt

$$G = \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix} \quad \text{mit } G_{11} \in \mathbb{R}^{k \times k},$$

wobei nur G_{11} , G_{12} und G_{21} von der Kreuzapproximation benutzt wurden, die Teilmatrix G_{22} jedoch bisher nicht angefasst wurde. Durch die Wahl der Pivot-Elemente als betragsgrößte Einträge kann davon ausgegangen werden, dass G_{11} invertierbar ist. Also kann folgende Zerlegung gemacht werden

$$G = \begin{pmatrix} I & 0 \\ G_{21}G_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} G_{11} & G_{12} \\ 0 & G_{22} - G_{21}G_{11}^{-1}G_{12} \end{pmatrix}.$$

4 Derzeit verwendete Verfahren

Mit dem Schurkomplement $S := G_{22} - G_{21}G_{11}^{-1}G_{12}$ gilt dann

$$\begin{aligned} G &= \begin{pmatrix} I & 0 \\ G_{21}G_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} G_{11} & G_{12} \\ 0 & S \end{pmatrix} \\ &= \begin{pmatrix} I & \\ G_{21}G_{11}^{-1} & \end{pmatrix} \begin{pmatrix} G_{11} & G_{12} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & S \end{pmatrix}. \end{aligned}$$

Unter den gemachten Annahmen sollte im ersten Summanden das Ergebnis der Kreuzapproximation nach k Schritten stehen und folglich im zweiten Summanden der Restfehler.

Bei der vollständigen Pivotierung ist die gesamte Matrix bekannt und der Fehler kann in Form der Matrix S direkt ausgerechnet werden und mit dem vorgegebenem Fehlerwert ε verglichen werden.

In der Implementierung von ACA mit vollständiger Pivotierung, die Teil des hybriden Algorithmus aus den Kapiteln 8, 9 und 10 ist, wurde das Abbruchkriterium wie folgt realisiert: Sobald das betragsgrößte Element der Restmatrix, also G_{22} , kleiner oder gleich dem betragsgrößten Element der Originalmatrix $G|_{\hat{t} \times \hat{s}}$ skaliert mit der Toleranz ε ist, wird abgebrochen, also wenn

$$\max_{i,j} \{|S_{i,j}|\} \leq \varepsilon \cdot \max_{i \in \hat{t}, j \in \hat{s}} |G_{i,j}|$$

oder mit Hilfe der Maximumsnorm

$$\|A\|_{\max} := \max_{i,j} |A_{i,j}| \quad \text{für } A \in \mathbb{R}^{m \times n}$$

ausgedrückt, wenn

$$\|S\|_{\max} \leq \varepsilon \|G\|_{\max} \tag{4.14}$$

gilt. Als zusätzliche Sicherheitsmaßnahme wird kontrolliert, ob $k < \min\{\#\hat{t}, \#\hat{s}\}$ ist, da nur dann eine Approximation Sinn macht.

Bei der partiellen Pivotierung werden nur die für die Rang- k -Approximation benötigten Zeilen und Spalten der Ausgangsmatrix berechnet, das heißt, dass die Teilmatrix G_{22} nicht nur vom Algorithmus bisher unangetastet ist, sondern auch gar nicht im Speicher vorliegt. Damit ist natürlich auch der Restfehler in $S := G_{22} - G_{21}G_{11}^{-1}G_{12}$ nicht direkt zu berechnen.

Um dennoch eine sinnvolle Abbruchbedingung formulieren zu können, müssen Abschätzungen für die unbekanntenen Komponenten herangezogen werden. Dazu wird folgende Annahme gemacht: Das mit der partiellen Pivotsuche ermittelte Element unterscheidet sich nur in einer Konstante c von dem Pivot-Element, das eine vollständige Pivotierung gefunden hätte, also

$$c |G_{i_k, j_k}| \geq \|S\|_{\max}. \tag{4.15}$$

Damit lässt sich S zum Beispiel in der Frobeniusnorm gut abschätzen durch

$$\begin{aligned} \|S\|_F^2 &= \sum_{i=1}^{\#\hat{t}-k} \sum_{j=1}^{\#\hat{s}-k} |S_{i,j}|^2 \\ &\leq \sum_{i=1}^{\#\hat{t}-k} \sum_{j=1}^{\#\hat{s}-k} c |G_{i_k,j_k}|^2 \\ &= (\#\hat{t} - k)(\#\hat{s} - k) c |G_{i_k,j_k}|^2. \end{aligned}$$

In der Implementierung von ACA mit partieller Pivotierung, die in Kapitel 10 zu Vergleichszwecken herangezogen wird, wurde im Prinzip dasselbe Abbruchkriterium (4.14) wie für die vollständige Pivotsuche genutzt, mit dem Unterschied, dass $\|S\|_{\max}$ und $\|G\|_{\max}$ nur abgeschätzt werden können. Dazu ist wieder eine Annahme wie (4.15) notwendig.

Die Norm der Ausgangsmatrix wird hier durch das Produkt der euklidischen Vektornormen der ersten Spalten der Matrizen A und B der Rang- k -Approximation abgeschätzt:

$$\|G\|_{\max} \approx \left\| \begin{pmatrix} A_{1,1} \\ \vdots \\ A_{m,1} \end{pmatrix} \right\|_2 \left\| \begin{pmatrix} B_{1,1} \\ \vdots \\ B_{n,1} \end{pmatrix} \right\|_2 \quad \text{mit } m := \#\hat{t} \text{ und } n := \#\hat{s}.$$

Da man jedesmal davon ausgeht, *ungefähr* das betragsgrößte Element der Restmatrix zu bestimmen, ist dies durchaus sinnvoll. Auf dieselbe Art kann auch die Norm der Restmatrix in jedem Schritt abgeschätzt werden

$$\|S\|_{\max} \approx \left\| \begin{pmatrix} A_{1,k+1} \\ \vdots \\ A_{m,k+1} \end{pmatrix} \right\|_2 \left\| \begin{pmatrix} B_{1,k+1} \\ \vdots \\ B_{n,k+1} \end{pmatrix} \right\|_2$$

und so wieder auf (4.14) zurückgegriffen werden.

Natürlich funktioniert eine solche Vorgehensweise nicht für jede Matrix gleichermaßen zuverlässig. Es gibt durchaus Matrizen, die mit der Kreuzapproximation und einer partiellen Pivotsuche nicht gut approximiert werden können. Diese Tatsache erklärt sich damit, dass die gesamte Fehlerabschätzung auf einer unbewiesenen Annahme (4.15) beruht.

Die Zuverlässigkeit von ACA ist also nur mit vollständiger Pivotierung zu erreichen, die Effizienz allerdings nur mit partieller Pivotierung, was wiederum bereits am Anfang dieses Abschnitts 4.3 als Balanceakt zwischen Effizienz und Zuverlässigkeit angedeutet wurde.

Detailliertere Fehleranalysen und Ansätze zur Kombination von Effizienz und Zuverlässigkeit für ACA können nachgelesen werden in [3, 2, 4].

5 Die Green-Quadratur-Methode

Die Green-Quadratur-Methode, die in diesem Kapitel anhand des in Kapitel 2.1 beschriebenen Modellproblems ausführlich vorgestellt werden soll, beruht auf der Konstruktion einer entarteten Kernfunktion und ist damit der Klasse der analytischen Verfahren zuzuordnen, wie zuvor die Interpolation aus Abschnitt 4.1 und die Multipol-Methode aus Abschnitt 4.2. Allerdings sollte dieses neue Verfahren eine klare Überlegenheit gegenüber dem Interpolations-Verfahren zeigen: Für ein Problem im d -dimensionalen Raum muss bei der Interpolation in jede Raumdimension mit m Interpolationspunkten interpoliert werden, was zu einer Komplexität in der Größenordnung von m^d führt. Das Green-Quadratur-Verfahren dagegen benutzt Quadratur anstatt Interpolation, angewendet auf die durch die Green'sche Formel dargestellte Kernfunktion und führt zu einer Komplexität in der Größenordnung von m^{d-1} . Die Ersparnis von einem Faktor m ist darauf zurückzuführen, dass durch die Darstellung über die Green'sche Formel nur Approximationen über den Rand anstatt über das gesamte Gebiet wie bei der Interpolation gebraucht werden.

Leider muss diese vermeintlich klare Überlegenheit später noch etwas relativiert werden, wie auch die numerischen Experimente in Kapitel 7 zeigen.

In diesem Kapitel wird die neue Methode der Einfachheit halber zunächst nur für den zweidimensionalen Fall des Modellproblems betrachtet. Wie die Kernfunktion durch die Green'sche Formel repräsentiert werden kann, wird in Abschnitt 5.1 erläutert. In Abschnitt 5.2 wird die Quadratur darauf erklärt und die Gestalt der Niedrigrangmatrizen beschrieben. Abschließend wird in Abschnitt 5.3 unter Anderem auf die Übertragung der Methode auf den dreidimensionalen Fall eingegangen. Die entscheidenden Punkte aus diesem Kapitel 5 und den folgenden Kapiteln 6 und 7 sind zusammengefasst in [11].

5.1 Anwendung der Green'schen Formel auf die Kernfunktion

Die Kernfunktion für das zweidimensionale Modellproblem aus Abschnitt 2.1 ist gegeben durch

$$g(x, y) = -\frac{1}{2\pi} \log \|x - y\|. \quad (5.1)$$

Im weiteren Verlauf dieser Arbeit werden oft die Normalenableitungen der Kernfunktion benötigt. Diese berechnen sich im zweidimensionalen Fall wie folgt:

$$\frac{\partial g}{\partial n_x}(x, y) = \langle n(x), \nabla_x g(x, y) \rangle, \quad (5.2)$$

5 Die Green-Quadratur-Methode

wobei der Gradient für $x \neq y$

$$\begin{aligned}
 \nabla_x g(x, y) &= \left(\frac{\partial g}{\partial x_1}(x, y), \frac{\partial g}{\partial x_2}(x, y) \right)^\top \\
 &= \left(\frac{\partial g}{\partial x_1} \left(-\frac{1}{2\pi} \log \|x - y\| \right), \frac{\partial g}{\partial x_2} \left(-\frac{1}{2\pi} \log \|x - y\| \right) \right)^\top \\
 &= -\frac{1}{2\pi} \left(\frac{\partial g}{\partial x_1} \log \sqrt{\|x - y\|^2}, \frac{\partial g}{\partial x_2} \log \sqrt{\|x - y\|^2} \right)^\top \\
 &= -\frac{1}{2\pi} \left(\frac{1}{2} \frac{\partial g}{\partial x_1} \log \|x - y\|^2, \frac{1}{2} \frac{\partial g}{\partial x_2} \log \|x - y\|^2 \right)^\top \\
 &= -\frac{1}{2\pi} \left(\frac{1}{2} \frac{2(x_1 - y_1)}{\|x - y\|^2}, \frac{1}{2} \frac{2(x_2 - y_2)}{\|x - y\|^2} \right)^\top \\
 &= -\frac{x - y}{2\pi \|x - y\|^2}
 \end{aligned}$$

ist und mit (5.2)

$$\begin{aligned}
 \frac{\partial g}{\partial n_x}(x, y) &= \left\langle n(x), -\frac{x - y}{2\pi \|x - y\|^2} \right\rangle \\
 &= -\frac{\langle n(x), x - y \rangle}{2\pi \|x - y\|^2}
 \end{aligned}$$

folgt. Mit analoger Rechnung kommt man auf

$$\frac{\partial g}{\partial n_y}(x, y) = -\frac{\langle n(y), y - x \rangle}{2\pi \|x - y\|^2}. \quad (5.3)$$

Nun sind alle Größen bekannt, um den folgenden Satz zu formulieren. Er basiert auf der zweiten Green'schen Identität [18, 2. art 3.] und beinhaltet die für die neuen Verfahren so wichtige Formel, die im Folgenden nur noch *Green'sche Formel* genannt wird. Der Satz wird hier nur angegeben, aber nicht bewiesen. Für den Beweis sei auf die Originalarbeit [18] von George Green aus dem Jahre 1828 verwiesen.

Satz 5.1 (Green'sche Formel) *Sei $\Theta \subseteq \mathbb{R}^d$ ein Normalengebiet mit seinem Rand $\Xi = \partial\Theta$ und bezeichne $n : \Xi \rightarrow \mathbb{R}^d$ die normierte äußere Normale. Falls $u \in C^2(\overline{\Theta})$ die Potentialgleichung $\Delta u = 0$ erfüllt, gilt*

$$u(x) = \int_{\Xi} g(x, z) \frac{\partial u}{\partial n_z}(z) d\Xi(z) - \int_{\Xi} \frac{\partial g}{\partial n_z}(x, z) u(z) d\Xi(z) \quad (5.4)$$

für alle $x \in \Theta$.

Beweis. [18, art. 7] ■

Ziel ist es, die Green'sche Formel auf die Kernfunktion selbst anzuwenden. Dazu wird y festgehalten, so dass $g_y(x) := g(x, y)$ eine Funktion in x ist. Dann kann folgende Tatsache festgestellt werden:

Lemma 5.2 (Lösung der Potentialgleichung) Für ein festes $y \in \mathbb{R}^2$ erfüllt die Kernfunktion $g_y(x) := g(x, y)$ die Potentialgleichung $\Delta g_y(x) = 0$ im $\mathbb{R}^2 \setminus \{y\}$.

Beweis. Seien $x, y \in \mathbb{R}^2$ mit $x \neq y$. Dann gilt

$$\begin{aligned}
\Delta_x g(x, y) &= \Delta_x \left(-\frac{1}{2\pi} \log \|x - y\| \right) \\
&= -\frac{1}{2\pi} \Delta_x \log \sqrt{\|x - y\|^2} \\
&= -\frac{1}{4\pi} \left(\frac{\partial^2}{\partial x_1^2} \log \|x - y\|^2 + \frac{\partial^2}{\partial x_2^2} \log \|x - y\|^2 \right) \\
&= -\frac{1}{4\pi} \left(\frac{\partial}{\partial x_1} \frac{2(x_1 - y_1)}{\|x - y\|^2} + \frac{\partial}{\partial x_2} \frac{2(x_2 - y_2)}{\|x - y\|^2} \right) \\
&= -\frac{1}{4\pi} \left(\frac{2}{\|x - y\|^2} - \frac{4(x_1 - y_1)^2}{\|x - y\|^4} + \frac{2}{\|x - y\|^2} - \frac{4(x_2 - y_2)^2}{\|x - y\|^4} \right) \\
&= -\frac{1}{\pi \|x - y\|^2} \left(1 - \frac{(x_1 - y_1)^2 + (x_2 - y_2)^2}{\|x - y\|^2} \right) \\
&= -\frac{1}{\pi \|x - y\|^2} (1 - 1) \\
&= 0.
\end{aligned}$$

■

Damit ist die Voraussetzung gegeben, so dass die Green'sche Formel (5.4) auf die Kernfunktion selbst mittels $u(x) := g(x, y)$ angewendet werden kann. Für $x, y \in \mathbb{R}^2$ ergibt sich dann die Formel

$$g(x, y) = \int_{\Xi} g(x, z) \frac{\partial g}{\partial n_z}(z, y) dz - \int_{\Xi} \frac{\partial g}{\partial n_z}(x, z) g(z, y) dz. \quad (5.5)$$

Bis hierher ist diese Formel eine exakte Darstellung der Kernfunktion. Die Approximation der Kernfunktion findet erst statt, wenn die Wegintegrale über Ξ durch Quadratur berechnet werden. Der folgende Abschnitt befasst sich anhand des zweidimensionalen Problems anschaulich und ausführlich mit diesem Vorgehen.

5.2 Approximation durch Quadratur

Die Tatsache, dass die Kernfunktion selbst die Laplace-Gleichung erfüllt, reicht noch nicht ganz, um die Green'sche Formel hier anwenden zu dürfen: Es fehlt noch das Normalengebiet $\Theta \subset \mathbb{R}^2$ mit seinem Rand $\Xi = \partial\Theta$.

Die naheliegende Wahl dafür wäre ein Kreis oder ein Rechteck um das relevante Gebiet herum. Da die Fehleranalyse in Kapitel 6 im Falle eines Rechtecks gut handhabbar ist, fällt die Wahl an dieser Stelle auf ein Rechteck als Normalengebiet Θ .

Für diesen Fall wird, wie bereits bei der Kreuzapproximation in Abschnitt 4.3, die Max-Zulässigkeitsbedingung (4.13) benutzt. Und wie schon bei der Interpolation in Abschnitt

5 Die Green-Quadratur-Methode

4.1, wird die Approximation abhängig von den Durchmessern der Bounding Boxen \mathcal{B}_τ und \mathcal{B}_σ gewählt.

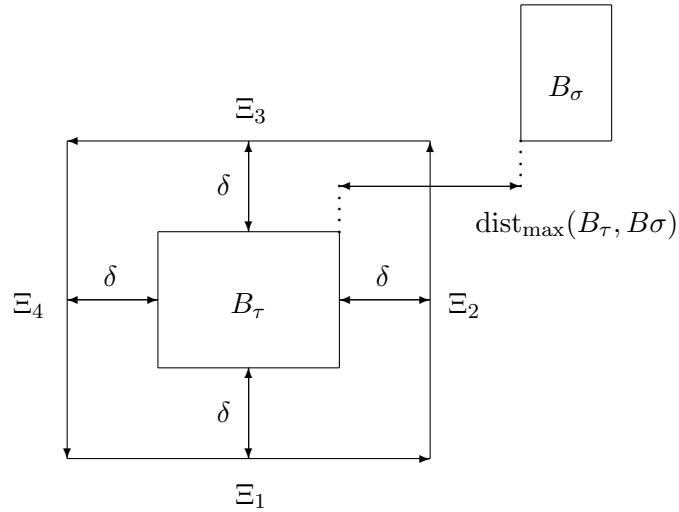
Es wird nachfolgend zunächst das Verfahren für den Fall $\text{diam}_{\max}(\mathcal{B}_\tau) \geq \text{diam}_{\max}(\mathcal{B}_\sigma)$ ausführlich beschrieben für die Bounding Boxen $\mathcal{B}_\tau = [a_1, b_1] \times [a_2, b_2]$ und $\mathcal{B}_\sigma = [c_1, d_1] \times [c_2, d_2]$. Dazu wird ein Normalengebiet Θ als Rechteck um die Bounding Box \mathcal{B}_τ gewählt, also mit $\mathcal{B}_\tau \subset \Theta$ und zwar so, dass der Rand $\Xi = \partial\Theta$ in einem δ -Abstand zu \mathcal{B}_τ liegt. Dabei garantiert die Wahl

$$\delta := \frac{1}{2} \text{diam}_{\max}(\mathcal{B}_\tau), \quad (5.6)$$

dass $\delta \leq \frac{1}{2} \text{dist}_{\max}(\mathcal{B}_\tau, \mathcal{B}_\sigma)$ gilt und damit der Rand Ξ zwischen den Bounding Boxen liegt.

An diesem Punkt sind alle Voraussetzungen gegeben, um die Formel (5.5) anzuwenden. Die praktische Berechnung, also die Approximation der Wegintegrale über Ξ durch Quadratur, erfordert die Konstruktion einer Parametrisierung von Ξ . In diesem zweidimensionalen Fall wird die Parametrisierung noch explizit angegeben.

Sei dazu $\Xi = \bigcup_{i=1}^4 \Xi_i$ das Rechteck mit der folgenden vierteiligen Parametrisierung:



$$\begin{aligned} \xi_1 : [-1, 1] &\rightarrow \Xi_1 \subseteq \mathbb{R}^2, & \xi_1(t) &= \begin{pmatrix} \frac{b_1+a_1}{2} + \frac{b_1-a_1+2\delta}{2} t \\ a_2 - \delta \end{pmatrix}, \\ \xi_2 : [-1, 1] &\rightarrow \Xi_2 \subseteq \mathbb{R}^2, & \xi_2(t) &= \begin{pmatrix} b_1 + \delta \\ \frac{b_2+a_2}{2} + \frac{b_2-a_2+2\delta}{2} t \end{pmatrix}, \\ \xi_3 : [-1, 1] &\rightarrow \Xi_3 \subseteq \mathbb{R}^2, & \xi_3(t) &= \begin{pmatrix} \frac{b_1+a_1}{2} - \frac{b_1-a_1+2\delta}{2} t \\ b_2 + \delta \end{pmatrix}, \\ \xi_4 : [-1, 1] &\rightarrow \Xi_4 \subseteq \mathbb{R}^2, & \xi_4(t) &= \begin{pmatrix} a_1 - \delta \\ \frac{b_2+a_2}{2} - \frac{b_2-a_2+2\delta}{2} t \end{pmatrix}. \end{aligned} \quad (5.7)$$

Es wird nun zusammengesetzte Gauß-Quadratur verwendet: Für ein Integral

$$\mathcal{I} : C([-1, 1]) \rightarrow \mathbb{R}, \quad f \mapsto \int_{-1}^1 f(x) \, dx$$

ist die m -te Quadraturformel \mathcal{Q}_m gegeben durch

$$\mathcal{Q}_m : C([-1, 1]) \rightarrow \mathbb{R}, \quad f \mapsto \sum_{\nu=1}^m w_\nu f(x_\nu), \quad (5.8)$$

wobei $x_1, \dots, x_m \in [-1, 1]$ die Quadraturpunkte und $w_1, \dots, w_m \in \mathbb{R}$ die zugehörigen Quadraturgewichte sind. Die Gewichte werden dabei über die Lagrange-Polynome (4.1) definiert:

$$w_\nu := \int_{-1}^1 \mathcal{L}_\nu(x) \, dx \quad \text{für alle } \nu \in \{1, \dots, m\}.$$

Die Polynome $L_m \in \Pi_m \setminus \{0\}$ mit der Eigenschaft

$$\int_{-1}^1 L_m(x)p(x) \, dx = 0 \quad \text{für alle } p \in \Pi_{m-1}$$

heißen *Legendre-Polynome*. Ein Legendre-Polynom L_m besitzt genau m Nullstellen, die alle in $[-1, 1]$ liegen. Wenn nun die Quadraturpunkte x_1, \dots, x_m gerade als diese Nullstellen gewählt werden, so spricht man von *Gauß-Quadraturpunkten* und (5.8) heißt dann *m -te Gauß-Quadraturformel*.

Für die zusammengesetzte Quadratur sei ℓ die Anzahl der äquidistant verteilten Teilintervalle des Intervalls $[-1, 1]$. Um eine kompakte Schreibweise zu ermöglichen, werden die Quadraturpunkte und Gewichte in ihrer Gesamtheit durchnummeriert: Seien $t_1, \dots, t_{m\ell} \in [-1, 1]$ die Gauß-Quadraturpunkte und $w_1, \dots, w_{m\ell} \in \mathbb{R}$ die zugehörigen Quadraturgewichte.

Diese Quadratur angewendet auf $g(x, y)$ in der Repräsentation durch die Green'sche Formel (5.5) ergibt die Approximation $\tilde{g}(x, y) \approx g(x, y)$ mit

$$\begin{aligned} \tilde{g}(x, y) = & \sum_{i=1}^4 \sum_{\nu=1}^{m\ell} \left(\|\xi'_i(t_\nu)\| w_\nu g(x, \xi_i(t_\nu)) \frac{\partial g}{\partial n_z}(\xi_i(t_\nu), y) \right. \\ & \left. - \|\xi'_i(t_\nu)\| w_\nu \frac{\partial g}{\partial n_z}(x, \xi_i(t_\nu)) g(\xi_i(t_\nu), y) \right). \end{aligned} \quad (5.9)$$

Damit ist eine Trennung der Variablen erreicht und mit \tilde{g} ist die gesuchte entartete Kernfunktion gefunden.

Nun kann die Kernapproximation genutzt werden, um einen zulässigen Matrixblock $G|_{\hat{t} \times \hat{s}}$ mit $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+$ und den Einträgen aus (2.4) durch eine Niedrigrangmatrix $\tilde{G}|_{\hat{t} \times \hat{s}}$ zu approximieren. Dazu sei \tilde{G} definiert durch

$$\tilde{G}_{ij} := \int_{\Gamma} \int_{\Gamma} \varphi_i(x) \tilde{g}(x, y) \varphi_j(y) \, dx \, dy \quad \text{für } i \in \hat{t}, j \in \hat{s}.$$

5 Die Green-Quadratur-Methode

Mit (5.9) folgt

$$\begin{aligned} \tilde{G}_{i,j} = & \sum_{\iota=1}^4 \sum_{\nu=1}^{m\ell} \left(\int_{\Gamma} \|\xi'_{\iota}(t_{\nu})\| w_{\nu} g(x, \xi_{\iota}(t_{\nu})) \varphi_i(x) \, dx \int_{\Gamma} \frac{\partial g}{\partial n_z}(\xi_{\iota}(t_{\nu}), y) \varphi_j(y) \, dy \right. \\ & \left. - \int_{\Gamma} \|\xi'_{\iota}(t_{\nu})\| w_{\nu} \frac{\partial g}{\partial n_z}(x, \xi_{\iota}(t_{\nu})) \varphi_i(x) \, dx \int_{\Gamma} g(\xi_{\iota}(t_{\nu}), y) \varphi_j(y) \, dy \right), \end{aligned}$$

welches sich mit Hilfe der Definitionen der Matrizen $\check{A}, \hat{A} \in \mathbb{R}^{\hat{t} \times m\ell}$ und $\check{B}, \hat{B} \in \mathbb{R}^{\hat{s} \times m\ell}$ durch

$$\begin{aligned} (\check{A}_{\iota})_{i\nu} &= \|\xi'_{\iota}(t_{\nu})\| w_{\nu} \int_{\Gamma} g(x, \xi_{\iota}(t_{\nu})) \varphi_i(x) \, dx \\ (\hat{A}_{\iota})_{i\nu} &= \|\xi'_{\iota}(t_{\nu})\| w_{\nu} \int_{\Gamma} \frac{\partial g}{\partial n_z}(x, \xi_{\iota}(t_{\nu})) \varphi_i(x) \, dx \\ (\check{B}_{\iota})_{j\nu} &= \int_{\Gamma} \frac{\partial g}{\partial n_z}(\xi_{\iota}(t_{\nu}), y) \varphi_j(y) \, dy \\ (\hat{B}_{\iota})_{j\nu} &= \int_{\Gamma} g(\xi_{\iota}(t_{\nu}), y) \varphi_j(y) \, dy \end{aligned} \tag{5.10}$$

für $i \in \hat{t}, j \in \hat{s}$ und $\nu \in \{1, \dots, m\ell\}$, kurz als

$$\tilde{G}|_{\hat{t} \times \hat{s}} = \sum_{\iota=1}^4 \check{A}_{\iota} \check{B}_{\iota}^* - \hat{A}_{\iota} \hat{B}_{\iota}^*$$

schreiben lässt. Wenn man die Matrizen \check{A}_{ι} und \hat{A}_{ι} sowie die Matrizen \check{B}_{ι} und \hat{B}_{ι} für alle $\iota = 1, \dots, 4$ zusammenfasst, ergibt sich mit

$$\begin{aligned} \tilde{G}|_{\hat{t} \times \hat{s}} &= [\check{A}_1 \dots \check{A}_4 \hat{A}_1 \dots \hat{A}_4] [\check{B}_1 \dots \check{B}_4 \quad -\hat{B}_1 \dots \quad -\hat{B}_4]^* \\ &=: AB^*. \end{aligned} \tag{5.11}$$

wieder die gewohnte Rang- k -Darstellung. Allerdings gilt nach Definition der Einzelmatrizen $\check{A}_{\iota}, \hat{A}_{\iota}, \check{B}_{\iota}$ und \hat{B}_{ι} mit ihren Rängen von je $m\ell$, das $A \in \mathbb{R}^{\hat{t} \times k}$ und $B \in \mathbb{R}^{\hat{s} \times k}$ mit $k := 8m\ell$.

Im Fall von $\text{diam}_{\max}(\mathcal{B}_{\tau}) < \text{diam}_{\max}(\mathcal{B}_{\sigma})$ wird das Normalengebiet Θ als Rechteck um \mathcal{B}_{σ} , also $\mathcal{B}_{\sigma} \subset \Theta$ gewählt. Mit analogem Vorgehen erhält man in dem Fall auch eine Rang- k -Darstellung mit $k = 8m\ell$.

Bemerkung 5.3 (Komplexitätsvergleich) Die Approximation eines Matrixblocks $G|_{\hat{t} \times \hat{s}}$ zu einem zulässigen Block $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+$ erfordert mit dem Green-Quadratur-Verfahren $\mathcal{O}((\#\hat{t} + \#\hat{s})8m\ell)$ Operationen. Im Vergleich dazu benötigt die Interpolation hier $\mathcal{O}((\#\hat{t} + \#\hat{s})m^2)$ Operationen.

Also liegt der Aufbau der gesamten \mathcal{H} -Matrix in $\mathcal{O}(nm\ell \log n)$, wenn die Green-Quadratur benutzt wird und in $\mathcal{O}(nm^2 \log n)$, wenn die Interpolation verwendet wird.

Dieser Komplexitätsvergleich legt nahe, dass man für d -dimensionale Probleme eine Komplexität von m^{d-1} für das Green-Quadratur-Verfahren gegenüber m^d für die Interpolation erwarten darf.

Die Begründung dafür liegt darin, dass in der Repräsentation durch die Green'sche Formel nur über den Rand integriert werden muss, anstatt über das gesamte Gebiet, womit sich die Ersparnis von einer Raumdimension, also einem Faktor m erklärt.

5.3 Verallgemeinerungen

Das Green-Quadratur-Verfahren wurde hier für den zweidimensionalen Fall und mit einer Rechteckparametrisierung vorgestellt. Es ist jedoch auch im dreidimensionalen Fall und für andere Normalengebiete Θ mit entsprechenden Parametrisierungen anwendbar, wie die folgenden Abschnitte 5.3.1 und 5.3.2 verdeutlichen.

Dann wurde bisher nur die Matrix des Einfachschichtpotentials (2.6) approximiert, obwohl das Verfahren genauso gut auch zur Berechnung der Matrix des Doppelschichtpotentials (2.8) genutzt werden kann, wie in Abschnitt 5.3.3 dargelegt wird.

Außerdem wurden für dieses Verfahren auch schon Überlegungen angestellt, wie es für \mathcal{H}^2 -Matrizen aussehen würde. Einer dieser ersten Schritte in diese Richtung wird in Abschnitt 5.3.4 skizziert.

5.3.1 3D-Fall

Die Kernfunktion im dreidimensionalen Fall des Modellproblems ist

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

Auch hier kann das folgende Lemma formuliert werden:

Lemma 5.4 (Lösung der Potentialgleichung) Für ein festes $y \in \mathbb{R}^3$ erfüllt die Kernfunktion $g_y(x) := g(x, y)$ die Potentialgleichung $\Delta g_y(x) = 0$ im $\mathbb{R}^3 \setminus \{y\}$.

Beweis. Seien $x, y \in \mathbb{R}^3$ mit $x \neq y$. Dann gilt

$$\begin{aligned} \Delta_x g(x, y) &= \frac{1}{4\pi} \Delta_x \|x - y\|^{-1} \\ &= \frac{1}{4\pi} \left(\frac{\partial^2}{\partial x_1^2} \|x - y\|^{-1} + \frac{\partial^2}{\partial x_2^2} \|x - y\|^{-1} + \frac{\partial^2}{\partial x_3^2} \|x - y\|^{-1} \right) \\ &= \frac{1}{4\pi} \left(\frac{\partial}{\partial x_1} \frac{y_1 - x_1}{\|x - y\|^3} + \frac{\partial}{\partial x_2} \frac{y_2 - x_2}{\|x - y\|^3} + \frac{\partial}{\partial x_3} \frac{y_3 - x_3}{\|x - y\|^3} \right) \end{aligned}$$

5 Die Green-Quadratur-Methode

$$\begin{aligned}
 &= \frac{1}{4\pi} \left(\frac{-1}{\|x-y\|^3} + \frac{3(x_1-y_1)^2}{\|x-y\|^5} + \frac{-1}{\|x-y\|^3} + \frac{3(x_2-y_2)^2}{\|x-y\|^5} \right. \\
 &\quad \left. + \frac{-1}{\|x-y\|^3} + \frac{3(x_3-y_3)^2}{\|x-y\|^5} \right) \\
 &= \frac{1}{4\pi} \left(\frac{-3}{\|x-y\|^3} + \frac{3\|x-y\|^2}{\|x-y\|^5} \right) \\
 &= \frac{-3+3}{4\pi\|x-y\|^3} \\
 &= 0.
 \end{aligned}$$

■

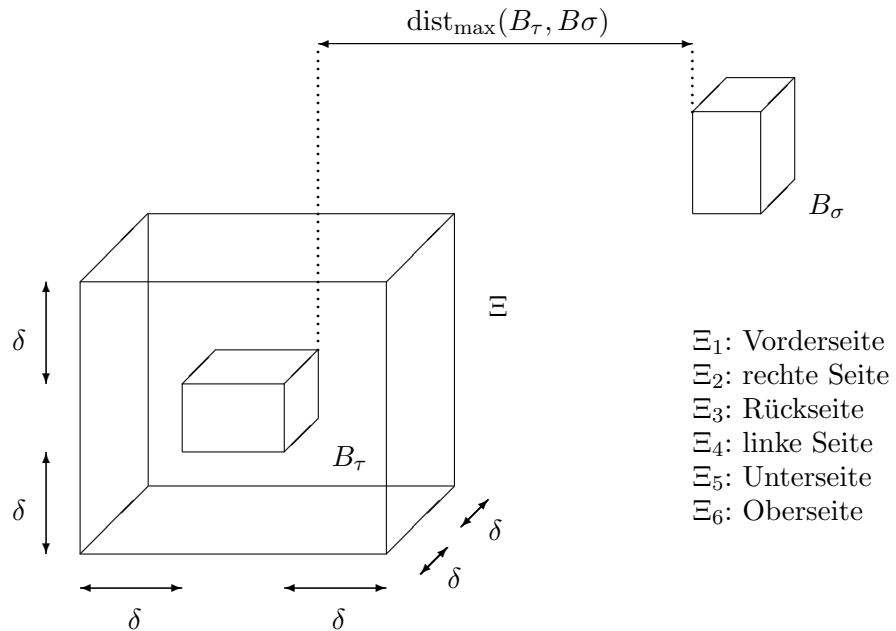
Damit kann die Green'sche Formel (5.4) auf $u(x) := g(x, y)$ angewendet werden. Es ergibt sich für $x, y \in \mathbb{R}^3$

$$g(x, y) = \int_{\Xi} g(x, z) \frac{\partial g}{\partial n_z}(z, y) d\Xi(z) - \int_{\Xi} \frac{\partial g}{\partial n_z}(x, z) g(z, y) d\Xi(z),$$

wobei die Integrale über $\partial\Theta = \Xi$ nun keine Wegintegrale wie im zweidimensionalen Fall, sondern Oberflächenintegrale sind. Das Normalengebiet Θ wird als Quader mit δ -Abstand um die Bounding Box $\mathcal{B}_\tau = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$ gewählt mit

$$\delta := \frac{1}{2} \text{diam}_{\max}(\mathcal{B}_\tau)$$

wie schon im zweidimensionalen Fall. Ebenso wird analog eine Parametrisierung für Ξ aufgestellt. Im dreidimensionalen Fall besteht diese aus sechs Teilen $\Xi := \bigcup_{\ell=1}^6 \Xi_\ell$, wobei jedes Ξ_ℓ eine der Seitenflächen des Quaders beschreibt, wie die Grafik verdeutlicht.



Die sechs Parametrisierungsteile ξ_l sehen dabei wie folgt aus:

$$\begin{aligned}
 \xi_1 : [-1, 1] \times [-1, 1] &\rightarrow \Xi_1 \subseteq \mathbb{R}^3, & \xi_1(t, s) &= \begin{pmatrix} \frac{b_1+a_1}{2} - \frac{b_1-a_1+2\delta}{2} t \\ \frac{b_2+a_2}{2} + \frac{b_2-a_2+2\delta}{2} s \\ a_3 - \delta \end{pmatrix}, \\
 \xi_2 : [-1, 1] \times [-1, 1] &\rightarrow \Xi_2 \subseteq \mathbb{R}^3, & \xi_2(t, s) &= \begin{pmatrix} b_1 + \delta \\ \frac{b_2+a_2}{2} + \frac{b_2-a_2+2\delta}{2} t \\ \frac{b_3+a_3}{2} + \frac{b_3-a_3+2\delta}{2} s \end{pmatrix}, \\
 \xi_3 : [-1, 1] \times [-1, 1] &\rightarrow \Xi_3 \subseteq \mathbb{R}^3, & \xi_3(t, s) &= \begin{pmatrix} \frac{b_1+a_1}{2} + \frac{b_1-a_1+2\delta}{2} t \\ \frac{b_2+a_2}{2} + \frac{b_2-a_2+2\delta}{2} s \\ b_3 + \delta \end{pmatrix}, \\
 \xi_4 : [-1, 1] \times [-1, 1] &\rightarrow \Xi_4 \subseteq \mathbb{R}^3, & \xi_4(t, s) &= \begin{pmatrix} a_1 - \delta \\ \frac{b_2+a_2}{2} - \frac{b_2-a_2+2\delta}{2} t \\ \frac{b_3+a_3}{2} + \frac{b_3-a_3+2\delta}{2} s \end{pmatrix}, \\
 \xi_5 : [-1, 1] \times [-1, 1] &\rightarrow \Xi_5 \subseteq \mathbb{R}^3, & \xi_5(t, s) &= \begin{pmatrix} \frac{b_1+a_1}{2} - \frac{b_1-a_1+2\delta}{2} t \\ a_2 - \delta \\ \frac{b_3+a_3}{2} + \frac{b_3-a_3+2\delta}{2} s \end{pmatrix}, \\
 \xi_6 : [-1, 1] \times [-1, 1] &\rightarrow \Xi_6 \subseteq \mathbb{R}^3, & \xi_6(t, s) &= \begin{pmatrix} \frac{b_1+a_1}{2} + \frac{b_1-a_1+2\delta}{2} t \\ b_2 + \delta \\ \frac{b_3+a_3}{2} + \frac{b_3-a_3+2\delta}{2} s \end{pmatrix}.
 \end{aligned} \tag{5.12}$$

Es soll wieder die zusammengesetzte Gauß-Quadratur der Ordnung m mit ℓ Teilintervallen angewendet werden. Dazu sind für jede der zwei Richtungen $m\ell$ Gauß-Quadraturpunkte $t_1, \dots, t_{m\ell} \in [-1, 1]$ mit zugehörigen Gewichten $w_1, \dots, w_{m\ell} \in \mathbb{R}$ nötig. Die Kernapproximation $\tilde{g}(x, y) \approx g(x, y)$ durch Quadratur sieht damit wie folgt aus:

$$\begin{aligned}
 \tilde{g}(x, y) &= \sum_{\iota=1}^6 \sum_{\nu_1=1}^{m\ell} \sum_{\nu_2=1}^{m\ell} \left(c_\iota w_{\nu_1} w_{\nu_2} g(x, \xi_\iota(t_{\nu_1}, t_{\nu_2})) \frac{\partial g}{\partial n_z}(\xi_\iota(t_{\nu_1}, t_{\nu_2}), y) \right. \\
 &\quad \left. - c_\iota w_{\nu_1} w_{\nu_2} \frac{\partial g}{\partial n_z}(x, \xi_\iota(t_{\nu_1}, t_{\nu_2})) g(\xi_\iota(t_{\nu_1}, t_{\nu_2}), y) \right),
 \end{aligned}$$

wobei $c_\iota := \left\| \frac{\partial \xi}{\partial t}(t, s) \times \frac{\partial \xi}{\partial s}(t, s) \right\|$ wieder die Gramsche Determinante ist.

Natürlich kann dabei auch für die eine Richtung eine andere Quadraturordnung m oder eine andere Anzahl an Teilintervallen ℓ als für die andere Richtung gewählt werden.

Durch Einsetzen in (2.4) erhält man die Approximation $\tilde{G}|_{\hat{t} \times \hat{s}}$ des zulässigen Matrixblocks $G|_{\hat{t} \times \hat{s}}$ mit $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+$ und kann mit dem Zusammenfassen von $\nu_1 = 1, \dots, m\ell$ und $\nu_2 = 1, \dots, m\ell$ zu $\nu = 1, \dots, m^2\ell^2$, also auch $t_\nu := (t_{\nu_1}, t_{\nu_2})$ und

5 Die Green-Quadratur-Methode

$w_\nu := w_{\nu_1} w_{\nu_2}$, die Matrizen $\check{A}, \hat{A} \in \mathbb{R}^{\hat{t} \times m^2 \ell^2}$ und $\check{B}, \hat{B} \in \mathbb{R}^{\hat{s} \times m^2 \ell^2}$ definieren durch

$$\begin{aligned}
 (\check{A}_\iota)_{i\nu} &= c_\iota w_\nu \int_\Gamma g(x, \xi_\iota(t_\nu)) \varphi_i(x) \, d\Gamma(x) \\
 (\hat{A}_\iota)_{i\nu} &= c_\iota w_\nu \int_\Gamma \frac{\partial g}{\partial n_z}(x, \xi_\iota(t_\nu)) \varphi_i(x) \, d\Gamma(x) \\
 (\check{B}_\iota)_{j\nu} &= \int_\Gamma \frac{\partial g}{\partial n_z}(\xi_\iota(t_\nu), y) \varphi_j(y) \, d\Gamma(y) \\
 (\hat{B}_\iota)_{j\nu} &= \int_\Gamma g(\xi_\iota(t_\nu), y) \varphi_j(y) \, d\Gamma(y)
 \end{aligned} \tag{5.13}$$

für $i \in \hat{t}, j \in \hat{s}$ und $\nu \in \{1, \dots, m^2 \ell^2\}$. Damit ist die Matrixapproximation gegeben durch

$$\tilde{G}|_{\hat{t} \times \hat{s}} = \sum_{\iota=1}^6 \check{A}_\iota \check{B}_\iota^* - \hat{A}_\iota \hat{B}_\iota^*$$

und entsprechend ist

$$\begin{aligned}
 \tilde{G}|_{\hat{t} \times \hat{s}} &= [\check{A}_1 \dots \check{A}_6 \hat{A}_1 \dots \hat{A}_6] [\check{B}_1 \dots \check{B}_6 \quad -\hat{B}_1 \dots \quad -\hat{B}_6]^* \\
 &=: AB^*
 \end{aligned} \tag{5.14}$$

die Rang- k -Darstellung mit $k := 12m^2 \ell^2$.

Der Fall mit dem Normalengebiet um die andere Bounding Box, also $\mathcal{B}_\sigma \subset \Theta$, geht wieder analog.

Bemerkung 5.5 (Komplexitätsvergleich) *Die Approximation eines Matrixblocks $G|_{\hat{t} \times \hat{s}}$ zu einem zulässigen Block $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+$ erfordert mit dem Green-Quadratur-Verfahren im dreidimensionalen Fall $\mathcal{O}((\#\hat{t} + \#\hat{s})12m^2 \ell^2)$ Operationen. Im Vergleich dazu benötigt die Interpolation hier $\mathcal{O}((\#\hat{t} + \#\hat{s})m^3)$ Operationen.*

Also liegt der Aufbau der gesamten \mathcal{H} -Matrix in $\mathcal{O}(nm^2 \ell^2 \log n)$, wenn die Green-Quadratur benutzt wird und in $\mathcal{O}(nm^3 \log n)$, wenn die Interpolation verwendet wird.

Auch wenn dieser Komplexitätsvergleich auf den ersten Blick nicht viel schlechter aussieht, als der für den zweidimensionalen Fall in Bemerkung 5.3, macht sich die höhere Konstante von 12 und das quadratische Eingehen der Teilintervalle ℓ in praktischen Anwendungen bemerkbar. So ist das Verfahren im dreidimensionalen Fall auch bei einer Einschränkung auf $\ell = 1$ der Interpolation nicht vorzuziehen, wie die numerischen Experimente in Abschnitt 7.2 belegen.

Aus dieser Erkenntnis ist die Notwendigkeit einer Verbesserung des Green-Quadratur-Ansatzes entstanden, die schließlich zur Entwicklung des Green-Hybrid-Verfahrens geführt hat, das in Kapitel 8 vorgestellt wird.

5.3.2 Kreis-Parametrisierung

Das Gebiet Θ für die Green'sche Formel (5.4) kann theoretisch beliebig gewählt werden, solange es sich um ein Normalengebiet handelt und die ausgewählte Bounding Box umschließt.

In der Wahl von Θ ist man aber durch die numerische Handhabbarkeit der Parametrisierung über den Rand $\Xi := \partial\Theta$ eingeschränkt. Da die einfache und schnelle numerische Berechnung von zentraler Bedeutung ist, bietet sich neben dem in Abschnitt 5.2 behandelten Rechteck noch ein Kreis für Ξ an.

Sei dazu erneut \mathcal{B}_τ die Bounding Box, die von Θ überdeckt werden soll. Mit den zuvor definierten Größen $c_t := \text{center}(\mathcal{B}_\tau)$ aus (4.9) und $\text{diam}_2(\mathcal{B}_\tau)$ aus Definition 3.23 lässt sich eine Kreisparametrisierung ξ von Ξ in δ -Abstand zur Bounding Box \mathcal{B}_τ definieren:

$$\xi : [-1, 1] \rightarrow \Xi \subseteq \mathbb{R}^2, \quad \xi(t) = (r_t + \delta) \begin{pmatrix} \cos(t\pi) \\ \sin(t\pi) \end{pmatrix} + c_t,$$

wobei $r_t := \frac{1}{2} \text{diam}_2(\mathcal{B}_\tau)$ der *Radius* der Bounding Box \mathcal{B}_τ ist. Damit ergibt sich für $\tilde{g}(x, y) \approx g(x, y)$

$$\tilde{g}(x, y) = \sum_{\nu=1}^{m\ell} \left(\|\xi'(t_\nu)\| w_\nu g(x, \xi(t_\nu)) \frac{\partial g}{\partial n_z}(\xi(t_\nu), y) - \|\xi'(t_\nu)\| w_\nu \frac{\partial g}{\partial n_z}(x, \xi(t_\nu)) g(\xi(t_\nu), y) \right),$$

wobei es bei der Kreisparametrisierung sinnvoller ist, die Quadraturpunkte t_ν äquidistant auf dem Kreis Ξ zu verteilen, als wieder auf Gauß-Quadraturpunkte zurückzugreifen.

Ebenso sollte die Wahl von δ an diese Parametrisierung angepasst werden. Numerische Experimente haben gezeigt, dass hier $\delta = 0.4 \cdot \text{diam}_{\max}(\mathcal{B}_\tau)$ eine geeignete Wahl ist.

Mit der Kreisparametrisierung gelangt man für die Matrixapproximation $\tilde{G}|_{\hat{t} \times \hat{s}} \approx G|_{\hat{t} \times \hat{s}}$ auf analoge Weise wie in Abschnitt 5.2 schließlich zu der Rang- k -Darstellung

$$\tilde{G}|_{\hat{t} \times \hat{s}} = \check{A}\check{B}^* - \hat{A}\hat{B}^* = [\check{A}\hat{A}][\check{B}\hat{B}]^* =: AB^*$$

mit $\check{A}, \hat{A} \in \mathbb{R}^{\hat{t} \times m\ell}$ und $\check{B}, \hat{B} \in \mathbb{R}^{\hat{s} \times m\ell}$ wie zuvor bei der Rechteckparametrisierung, aber mit $A \in \mathbb{R}^{\hat{t} \times k}$ und $B \in \mathbb{R}^{\hat{s} \times k}$ mit nur $k := 2m\ell$.

Der offensichtliche Vorteil der Kreisparametrisierung auf einen lokalen Rang von nur $2m\ell$ anstatt $8m\ell$ bei der Rechteckparametrisierung zu führen, macht sich auch in numerischen Experimenten bemerkbar (siehe Abschnitt 7.1). Dennoch soll diese Parametrisierung, aus den folgenden Gründen, hier nur am Rande erwähnt sein: Zum Einen ist damit nur im zweidimensionalen Fall gut zu rechnen, da man im dreidimensionalen Fall, also bei einem Kreis und einer Sphärenparametrisierung, zunächst vor der nicht-trivialen Aufgabe steht, Quadraturpunkte äquidistant auf der Kugeloberfläche zu verteilen. Zum Anderen werden die Fehlerabschätzungen für die Rechtecks- bzw. Quaderparametrisierungen sehr viel handlicher.

5.3.3 Anwendung auf SLP und DLP

Bisher wurde das Green-Quadratur-Verfahren nur auf Matrizen des Einfachschichtpotentials, also SLP-Matrizen wie in Definition 2.2 angewendet. Es geht aber genauso auch für Matrizen des Doppelschichtpotentials, also DLP-Matrizen wie in Definition 2.4.

Wie bereits in (2.9) vorweggenommen, kann auch hier die Green'sche Formel (5.4) angewendet werden, und zwar auf $u(x) := \frac{\partial g}{\partial n_y}(x, y)$. Es ergibt sich

$$\frac{\partial g}{\partial n_y}(x, y) = \int_{\Gamma} g(x, z) \frac{\partial^2 g}{\partial n_z \partial n_y}(z, y) d\Gamma(z) - \int_{\Gamma} \frac{\partial g}{\partial n_z}(x, z) \frac{\partial g}{\partial n_y}(z, y) d\Gamma(z).$$

Die gleiche Parametrisierung wie im SLP-Fall kann genutzt werden und damit die Quadratur wie bisher durchgeführt werden.

Die einzige noch offene Frage ist, ob $\frac{\partial g}{\partial n_y}(x, y)$ im zwei- wie auch im dreidimensionalen Fall ebenso die Potentialgleichung erfüllt, wird mit folgendem Lemma beantwortet.

Lemma 5.6 (Lösung der Potentialgleichung) Sei $d \in \{2, 3\}$. Für ein festes $y \in \mathbb{R}^d$ erfüllt die Kernfunktion $\frac{\partial g_y}{\partial n_y}(x) := \frac{\partial g}{\partial n_y}(x, y)$ die Potentialgleichung $\Delta \frac{\partial g_y}{\partial n_y}(x) = 0$ im $\mathbb{R}^d \setminus \{y\}$.

Beweis. Es bezeichne $n_i(y)$ mit $i = 1, \dots, d$ die i -te Komponente des Normalenvektors. Sei $d = 2$, also seien $x, y \in \mathbb{R}^2$ mit $x \neq y$. Nach (5.3) gilt

$$\frac{\partial g}{\partial n_y}(x, y) = -\frac{\langle n(y), y - x \rangle}{2\pi \|x - y\|^2}.$$

Die partiellen Ableitungen berechnen sich für $i = 1, 2$ durch

$$\begin{aligned} \frac{\partial^2}{\partial^2 x_i} \frac{\partial g}{\partial n_y}(x, y) &= -\frac{1}{2\pi} \frac{\partial^2}{\partial^2 x_i} \left(\frac{\langle n(y), y - x \rangle}{\|x - y\|^2} \right) \\ &= -\frac{1}{2\pi} \frac{\partial}{\partial x_i} \left(\frac{2(y_i - x_i) \langle n(y), y - x \rangle}{\|x - y\|^4} - \frac{n_i(y)}{\|x - y\|^2} \right) \\ &= -\frac{1}{2\pi} \left(\frac{8(y_i - x_i)^2 \langle n(y), y - x \rangle}{\|x - y\|^6} \right. \\ &\quad \left. + \frac{-2n_i(y)(y_i - x_i) - 2 \langle n(y), y - x \rangle}{\|x - y\|^4} \right. \\ &\quad \left. + \frac{-2n_i(y)(y_i - x_i)}{\|x - y\|^4} \right) \\ &= -\frac{1}{\pi \|x - y\|^4} \\ &\quad \left(\frac{4(y_i - x_i)^2 \langle n(y), y - x \rangle}{\|x - y\|^2} - \langle n(y), y - x \rangle - 2n_i(y)(y_i - x_i) \right). \end{aligned}$$

Also folgt

$$\begin{aligned}
\Delta_x \frac{\partial g}{\partial n_y}(x, y) &= \frac{\partial^2}{\partial^2 x_1} \frac{\partial g}{\partial n_y}(x, y) + \frac{\partial^2}{\partial^2 x_2} \frac{\partial g}{\partial n_y}(x, y) \\
&= -\frac{1}{\pi \|x - y\|^4} \left(\frac{4(y_1 - x_1)^2 \langle n(y), y - x \rangle + 4(y_2 - x_2)^2 \langle n(y), y - x \rangle}{\|x - y\|^2} \right. \\
&\quad \left. - 2 \langle n(y), y - x \rangle - 2n_1(y)(y_1 - x_1) - 2n_2(y)(y_2 - x_2) \right) \\
&= -\frac{2}{\pi \|x - y\|^4} \left(\frac{2 \langle n(y), y - x \rangle \|x - y\|^2}{\|x - y\|^2} - 2 \langle n(y), y - x \rangle \right) \\
&= -\frac{2 \langle n(y), y - x \rangle}{\pi \|x - y\|^4} (2 - 2) \\
&= 0.
\end{aligned}$$

Sei nun $d = 3$, also seien $x, y \in \mathbb{R}^3$ mit $x \neq y$. Es gilt

$$\begin{aligned}
\nabla_y g(x, y) &= \left(\frac{\partial g}{\partial x_1}(x, y), \frac{\partial g}{\partial x_2}(x, y), \frac{\partial g}{\partial x_3}(x, y) \right)^\top \\
&= \frac{1}{4\pi} \left(\frac{\partial}{\partial x_1} \|x - y\|^{-1}, \frac{\partial}{\partial x_2} \|x - y\|^{-1}, \frac{\partial}{\partial x_3} \|x - y\|^{-1} \right)^\top \\
&= \frac{1}{4\pi} \left(\frac{y_1 - x_1}{\|x - y\|^3}, \frac{y_2 - x_2}{\|x - y\|^3}, \frac{y_3 - x_3}{\|x - y\|^3} \right)^\top \\
&= \frac{y - x}{4\pi \|x - y\|^3}
\end{aligned}$$

und damit

$$\frac{\partial g}{\partial n_y}(x, y) = \frac{\langle n(y), y - x \rangle}{4\pi \|x - y\|^3}.$$

Die partiellen Ableitungen berechnen sich für $i = 1, 2, 3$ durch

$$\begin{aligned}
\frac{\partial^2}{\partial^2 x_i} \frac{\partial g}{\partial n_y}(x, y) &= \frac{1}{4\pi} \frac{\partial^2}{\partial^2 x_i} \left(\frac{\langle n(y), y - x \rangle}{\|x - y\|^3} \right) \\
&= \frac{1}{4\pi} \frac{\partial}{\partial x_i} \left(\frac{3(y_i - x_i) \langle n(y), y - x \rangle}{\|x - y\|^5} - \frac{n_i(y)}{\|x - y\|^3} \right) \\
&= \frac{1}{4\pi} \left(\frac{15(y_i - x_i)^2 \langle n(y), y - x \rangle}{\|x - y\|^7} \right. \\
&\quad \left. + \frac{-3n_i(y)(y_i - x_i) - 3 \langle n(y), y - x \rangle}{\|x - y\|^5} \right. \\
&\quad \left. + \frac{-3n_i(y)(y_i - x_i)}{\|x - y\|^5} \right) \\
&= \frac{3}{4\pi \|x - y\|^5} \left(\frac{5(y_i - x_i)^2 \langle n(y), y - x \rangle}{\|x - y\|^2} \right. \\
&\quad \left. - \langle n(y), y - x \rangle - 2n_i(y)(y_i - x_i) \right).
\end{aligned}$$

Also folgt

$$\begin{aligned}
 \Delta_x \frac{\partial g}{\partial n_y}(x, y) &= \frac{\partial^2}{\partial^2 x_1} \frac{\partial g}{\partial n_y}(x, y) + \frac{\partial^2}{\partial^2 x_2} \frac{\partial g}{\partial n_y}(x, y) + \frac{\partial^2}{\partial^2 x_3} \frac{\partial g}{\partial n_y}(x, y) \\
 &= \frac{3}{4\pi \|x - y\|^5} \left(\frac{5 \langle n(y), y - x \rangle \|x - y\|^2}{\|x - y\|^2} - 5 \langle n(y), y - x \rangle \right) \\
 &= \frac{3 \langle n(y), y - x \rangle}{4\pi \|x - y\|^5} (5 - 5) \\
 &= 0.
 \end{aligned}$$

■

Damit sind alle Voraussetzungen erfüllt, um das Green'sche Lemma auch im Falle des Doppelschichtpotentials anwenden zu können.

5.3.4 Ansatz für uniforme \mathcal{H} -Matrizen

Ein erster Versuch, das Green-Quadratur-Verfahren auf \mathcal{H}^2 -Matrizen anzuwenden, führte zunächst nur zu einem Algorithmus für uniforme \mathcal{H} -Matrizen wie in Bemerkung 3.31.

Die Idee für diesen Algorithmus wird hier nur kurz angedeutet. Sie besteht darin, die Green'sche Formel zweimal hintereinander, erst in der x -Variablen, dann in der y -Variablen, anzuwenden. Dabei werden Normalengebiete um beide Bounding Boxen gelegt, etwa $\mathcal{B}_\tau \subset \Theta_\tau$ und $\mathcal{B}_\sigma \subset \Theta_\sigma$, und Parametrisierungen für $\partial\Theta_\tau =: \Xi_\tau$ und $\partial\Theta_\sigma =: \Xi_\sigma$ konstruiert.

Zunächst wird die Green'sche Formel auf $u(x) := g(x, y)$ angewendet, um die bereits aus (5.5) bekannte Formel

$$g(x, y) = \int_{\Xi_\tau} g(x, z) \frac{\partial g}{\partial n_z}(z, y) dz - \int_{\Xi_\tau} \frac{\partial g}{\partial n_z}(x, z) g(z, y) dz \quad (5.15)$$

zu erhalten. Danach wird die Green'sche Formel auf die Teilterme von (5.15) angewendet, die noch von y abhängen, also auf $u(y) := \frac{\partial g}{\partial n_z}(z, y)$, welches

$$\frac{\partial g}{\partial n_z}(z, y) = \int_{\Xi_\sigma} g(\omega, y) \frac{\partial^2 g}{\partial n_z \partial n_\omega}(z, \omega) d\omega - \int_{\Xi_\sigma} \frac{\partial g}{\partial n_\omega}(\omega, y) \frac{\partial g}{\partial n_z}(z, \omega) d\omega$$

ergibt, und auf $u(y) := g(z, y)$, welches dann

$$g(z, y) = \int_{\Xi_\sigma} g(\omega, y) \frac{\partial g}{\partial n_\omega}(z, \omega) d\omega - \int_{\Xi_\sigma} \frac{\partial g}{\partial n_\omega}(\omega, y) g(z, \omega) d\omega$$

ergibt. Diese Formeln eingesetzt in (5.15) liefern schließlich die Formel

$$\begin{aligned}
 g(x, y) &= \int_{\Xi_\tau} g(x, z) \int_{\Xi_\sigma} g(\omega, y) \frac{\partial^2 g}{\partial n_z \partial n_\omega}(z, \omega) \, d\omega \, dz \\
 &\quad - \int_{\Xi_\tau} g(x, z) \int_{\Xi_\sigma} \frac{\partial g}{\partial n_\omega}(\omega, y) \frac{\partial g}{\partial n_z}(z, \omega) \, d\omega \, dz \\
 &\quad - \int_{\Xi_\tau} \frac{\partial g}{\partial n_z}(x, z) \int_{\Xi_\sigma} g(\omega, y) \frac{\partial g}{\partial n_\omega}(z, \omega) \, d\omega \, dz \\
 &\quad + \int_{\Xi_\tau} \frac{\partial g}{\partial n_z}(x, z) \int_{\Xi_\sigma} \frac{\partial g}{\partial n_\omega}(\omega, y) g(z, \omega) \, d\omega \, dz .
 \end{aligned}$$

An dieser Stelle kann wie zuvor mit der Gauß-Quadratur approximiert werden, um die Niedrigrangdarstellung zu erhalten. Dabei werden die von x abhängigen Terme in der Clusterbasis V und die von y abhängenden in der Clusterbasis W verarbeitet. Die restlichen Terme werden zu den Kopplungsmatrizen $(S_b)_{b \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+}$.

Eine ausführliche Beschreibung und Fehleranalysen zu diesem Algorithmus sind in der Diplomarbeit von Julia Falke [14] zu finden.

6 Fehleranalyse der Green-Quadratur-Methode

Dieses Kapitel beschäftigt sich mit der Fehleranalyse des Verfahrens aus Kapitel 5. Da die Darstellung (5.5) mit der Green'schen Formel exakt ist und die Approximation erst mit der Quadratur stattfindet, kann hier auch von dem Quadraturfehler gesprochen werden.

Im ersten Abschnitt 6.1 dieses Kapitels wird für den zweidimensionalen Fall ein präziser Beweis durchgeführt, der auf den in Abschnitt 6.1.1 untersuchten Eigenschaften der Parametrisierung beruht. Abschnitt 6.2 bietet einen alternativen Beweis, der auf Eigenschaften der Gauß-Quadratur aufbaut und den zwei- wie auch den dreidimensionalen Fall gleichermaßen abdeckt.

6.1 Fehlerabschätzung im zweidimensionalen Fall

Der zentrale Beweis in diesem Abschnitt nutzt direkt die Eigenschaften der Rechteck-Parametrisierung (5.7). Daher werden zunächst die später benötigten Eigenschaften in Abschnitt 6.1.1 bewiesen.

Auch wenn es hier wie eine Einschränkung auf eine ganz spezielle Parametrisierung aussieht, so funktioniert diese Art der Abschätzung des Quadraturfehlers, sobald Lemma 6.5 erfüllt ist, also sobald eine Parametrisierung genutzt wird, für die die zweite Ableitung verschwindet.

6.1.1 Eigenschaften der Parametrisierung

Im Folgenden seien ξ_1, \dots, ξ_4 die vier Teile der Rechteck-Parametrisierung aus (5.7) mit $\delta := \frac{1}{2} \text{diam}_{\max}(\mathcal{B}_\tau)$ wie in (5.6). Dann gelten die folgenden Aussagen.

Lemma 6.1 (Abschätzung der Ableitungen) Sei $\iota \in \{1, \dots, 4\}$. Dann gilt

$$\|\xi'_\iota(t)\| \leq \text{diam}_{\max}(\mathcal{B}_\tau).$$

Beweis. Sei $\iota = 1$. Dann gilt

$$\xi'_1(t) = \begin{pmatrix} \frac{b_1 - a_1 + 2\delta}{2} \\ 0 \end{pmatrix}$$

und mit (5.6) folgt

$$\begin{aligned} \|\xi'_1(t)\| &= \frac{b_1 - a_1 + 2\delta}{2} = \frac{b_1 - a_1 + \text{diam}_{\max}(\mathcal{B}_\tau)}{2} \\ &\leq \frac{2 \text{diam}_{\max}(\mathcal{B}_\tau)}{2} = \text{diam}_{\max}(\mathcal{B}_\tau). \end{aligned}$$

6 Fehleranalyse der Green-Quadratur-Methode

Für $\iota = 2, 3, 4$ geht der Beweis analog. ■

Lemma 6.2 (Abstandsabschätzung) *Sei die max-Zulässigkeitsbedingung aus (4.13) erfüllt und sei $\Xi = \bigcup_{\iota=1}^4 \Xi_\iota$ das durch ξ_1, \dots, ξ_4 parametrisierte Rechteck. Dann gilt*

$$\delta \leq \min\{\text{dist}(x, \Xi), \text{dist}(\Xi, y) : x \in \mathcal{B}_\tau, y \in \mathcal{B}_\sigma\}.$$

Beweis. Sei $y \in \mathcal{B}_\sigma$ und wähle $z \in \Xi$ so, dass $\|y - z\|_\infty$ minimal ist. Nach Konstruktion der Parametrisierung gilt für alle $x \in \mathcal{B}_\tau$, dass $\text{dist}(x, \Xi) \geq \delta$. Wähle jetzt $x \in \mathcal{B}_\tau$ so, dass $\|x - z\|_\infty$ minimal ist, also $\|x - z\|_\infty = \delta$. Dann gilt

$$\begin{aligned} \|y - z\|_2 &\geq \|y - z\|_\infty \\ &\geq \|y - x\|_\infty - \|x - z\|_\infty \\ &\geq \text{dist}_{\max}(\mathcal{B}_\tau, \mathcal{B}_\sigma) - \delta. \end{aligned}$$

Mit der max-Zulässigkeitsbedingung folgt schließlich

$$\begin{aligned} \|y - z\|_2 &\geq \text{dist}_{\max}(\mathcal{B}_\tau, \mathcal{B}_\sigma) - \delta = \text{dist}_{\max}(\mathcal{B}_\tau, \mathcal{B}_\sigma) - \frac{1}{2} \text{diam}_{\max}(\mathcal{B}_\tau) \\ &\geq \frac{1}{2} \text{diam}_{\max}(\mathcal{B}_\tau) = \delta. \end{aligned}$$

Also gilt auch $\text{dist}(\Xi, y) = \|y - z\|_2 \geq \delta$. ■

Mit der Einschränkung, dass das gesamte Gebiet Ω einen Durchmesser von nicht mehr als 1 hat, also $\text{diam}(\Omega) \leq 1$, kann die folgende Abschätzung für die Hintereinanderausführung von Kernfunktion und Parametrisierung formuliert werden.

Lemma 6.3 (Abschätzung der Kernfunktion) *Sei $\text{diam}(\Omega) \leq 1$. Gelte die max-Zulässigkeitsbedingung für \mathcal{B}_τ und \mathcal{B}_σ . Seien $x \in \mathcal{B}_\tau$, $y \in \mathcal{B}_\sigma$ und ξ_1, \dots, ξ_4 und δ wie zuvor. Dann gilt für alle $\iota \in \{1, \dots, 4\}$*

$$|g(x, \xi_\iota(t))| \leq \frac{1}{2\pi} |\log \delta| \quad \text{und} \quad |g(\xi_\iota(t), y)| \leq \frac{1}{2\pi} |\log \delta|.$$

Beweis. Sei $\iota \in \{1, \dots, 4\}$. Dann gilt

$$\|x - \xi_\iota(t)\|_2 \geq \text{dist}(x, \Xi)$$

und mit Lemma 6.2 folgt daraus

$$\|x - \xi_\iota(t)\|_2 \geq \delta.$$

Mit der Einschränkung $\text{diam}(\Omega) \leq 1$ ist nach der Zulässigkeitsbedingung $\text{diam}_{\max}(\mathcal{B}_\tau) \leq \frac{1}{2}$ und damit $\delta \leq \frac{1}{4}$. Also ist $\|x - \xi_\iota(t)\|_2 \in (0, 1)$. Da die Funktion $|\log \cdot|$ auf dem Intervall $(0, 1)$ monoton fallend ist, folgt

$$\begin{aligned} |g(x, \xi_\iota(t))| &= \left| -\frac{1}{2\pi} \log \|x - \xi_\iota(t)\|_2 \right| = \frac{1}{2\pi} |\log \|x - \xi_\iota(t)\|_2| \\ &\leq \frac{1}{2\pi} |\log \delta|. \end{aligned}$$

6.1 Fehlerabschätzung im zweidimensionalen Fall

Mit analoger Argumentation folgt auch

$$|g(\xi_\iota(t), y)| \leq \frac{1}{2\pi} |\log \delta|.$$

■

Ebenso ist es möglich, eine Abschätzung für die Normalenableitung anzugeben.

Lemma 6.4 (Abschätzung der Normalenableitung) *Gelte die max-Zulässigkeitsbedingung für \mathcal{B}_τ und \mathcal{B}_σ . Sei $y \in \mathcal{B}_\sigma$ und ξ_1, \dots, ξ_4 und δ wie zuvor. Dann gilt für alle $\iota \in \{1, \dots, 4\}$*

$$\left| \frac{\partial g}{\partial n_z}(\xi_\iota(t), y) \right| \leq \frac{1}{2\pi\delta}.$$

Beweis. Sei $\iota \in \{1, \dots, 4\}$. Dann gilt

$$\left| \frac{\partial g}{\partial n_z}(\xi_\iota(t), y) \right| = \left| -\frac{\langle n(\xi_\iota(t)), \xi_\iota(t) - y \rangle}{2\pi \|\xi_\iota(t) - y\|^2} \right| = \frac{|\langle n(\xi_\iota(t)), \xi_\iota(t) - y \rangle|}{2\pi \|\xi_\iota(t) - y\|^2}$$

und mit der Cauchy-Schwarz-Ungleichung folgt dann

$$\left| \frac{\partial g}{\partial n_z}(\xi_\iota(t), y) \right| \leq \frac{\|n(\xi_\iota(t))\| \|\xi_\iota(t) - y\|}{2\pi \|\xi_\iota(t) - y\|^2} = \frac{\|n(\xi_\iota(t))\|}{2\pi \|\xi_\iota(t) - y\|}.$$

Da $\|n(\xi_\iota(t))\| = 1$ und nach Lemma 6.2 $\|\xi_\iota(t) - y\| \geq \text{dist}(\Xi, y) \geq \delta$ gilt, ist schließlich

$$\left| \frac{\partial g}{\partial n_z}(\xi_\iota(t), y) \right| \leq \frac{1}{2\pi\delta}.$$

■

Die für die Quadraturfehler-Abschätzung wichtigste Eigenschaft der Parametrisierung ist das Verschwinden ihrer zweiten Ableitung, damit das folgende Lemma anwendbar ist.

Lemma 6.5 (n -te Ableitungen) *Sei $\Theta \subseteq \mathbb{R}^d$ ein Gebiet mit $d \in \{2, 3\}$ und $\partial\Theta = \Xi$ und sei $U \subseteq \mathbb{R}$. Sei $u \in C^n(\Theta)$ eine Funktion und sei $\xi : U \rightarrow \Xi$ eine Funktion mit $\frac{\partial^2}{\partial x_i \partial x_j} \xi(x) = 0$ für alle $x = (x_1, \dots, x_{d-1})$. Dann gilt*

$$\frac{\partial^n}{\partial x_i^n} u(\xi(x)) = \left(\frac{\partial^n}{\partial x_i^n} u \right) (\xi(x))$$

für alle $i = 1, \dots, d-1$.

Beweis. Da $\frac{\partial^2}{\partial x_i \partial x_j} \xi(x) = 0$ gilt, muss

$$\frac{\partial}{\partial x_i} \xi(x) = \partial_{e_i} \xi(x) = D\xi(x)e_i =: q$$

6 Fehleranalyse der Green-Quadratur-Methode

eine Konstante sein, also unabhängig von x . Zeige zunächst

$$\frac{\partial^n}{\partial t^n}(u(\xi(x + te_i))) = (\partial_q^n u)(\xi(x + te_i)) \quad (6.1)$$

durch Induktion über n . Für $n = 0$ ist die Aussage 6.1 trivial.

Gelte nun die Aussage 6.1 für ein $n \in \mathbb{N}_0$. Für $n + 1$ gilt mit der Induktionsvoraussetzung

$$\begin{aligned} \frac{\partial^{n+1}}{\partial t^{n+1}}u(\xi(x + te_i)) &= \frac{\partial}{\partial t} \frac{\partial^n}{\partial t^n}u(\xi(x + te_i)) \\ &= \frac{\partial}{\partial t} (\partial_q^n u)(\xi(x + te_i)) \\ &= D(\partial_q^n u)(\xi(x + te_i))D\xi(x + te_i)e_i. \end{aligned}$$

Da die erste Ableitung von ξ eine Konstante ist, gilt $D\xi(x + te_i)e_i = D\xi(x)e_i$ und es folgt weiter

$$\begin{aligned} \frac{\partial^{n+1}}{\partial t^{n+1}}u(\xi(x + te_i)) &= D(\partial_q^n u)(\xi(x + te_i))D\xi(x)e_i \\ &= D(\partial_q^n u)(\xi(x + te_i))q \\ &= \partial_q(\partial_q^n u)(\xi(x + te_i)) \\ &= (\partial_q^{n+1}u)(\xi(x + te_i)) \end{aligned}$$

womit die Aussage 6.1 bewiesen ist.

Schließlich gilt für $v := u \circ \xi$

$$\begin{aligned} \frac{\partial^n}{\partial x_i^n}u(\xi(x)) &= \partial_{e_i}^n u(\xi(x)) \\ &= \partial_{e_i}^n v(x) \\ &= \frac{\partial^n}{\partial t^n}v(x + te_i)|_{t=0} \\ &= \frac{\partial^n}{\partial t^n}u(\xi(x + te_i))|_{t=0} \\ &= (\partial_q^n u)(\xi(x + te_i))|_{t=0} \\ &= (\partial_q^n u)(\xi(x)) \\ &= \left(\frac{\partial^n}{\partial x_i} \xi(x) u \right) (\xi(x)). \end{aligned}$$

■

Mit diesem Lemma gilt dann für alle $\iota \in \{1, \dots, 4\}$

$$\frac{d^n}{dt^n}g(x, \xi_\iota(t)) = \partial_{\xi_\iota(t)}^n g(x, \xi_\iota(t)),$$

wodurch es in der Quadraturfehler-Abschätzung möglich wird, die asymptotische Glattheit der Kernfunktion g direkt zu nutzen.

6.1.2 Quadraturfehler-Abschätzung

Bevor die eigentliche Fehlerabschätzung bewiesen werden kann, ist noch eine weitere technische Aussage und die Erinnerung an Eigenschaften der Gauß-Quadratur nötig.

Lemma 6.6 (Polynom) Sei $\sigma \in \mathbb{N}$. Dann ist für alle $x \in \mathbb{N}$

$$\sum_{k=0}^x \frac{(k + \sigma - 1)!(x - k + \sigma)!}{k!(x - k)!} \leq p(x)$$

wobei $p \in \Pi_{2\sigma}$.

Beweis. Es gilt

$$\frac{(k + \sigma - 1)!}{k!} = \prod_{l=k+1}^{k+\sigma-1} l \leq (k + \sigma - 1)^{\sigma-1}$$

und

$$\frac{(x - k + \sigma)!}{(x - k)!} = \prod_{l=x-k+1}^{x-k+\sigma} l \leq (x - k + \sigma)^\sigma.$$

Damit folgt

$$\begin{aligned} \sum_{k=0}^x \frac{(k + \sigma - 1)!(x - k + \sigma)!}{k!(x - k)!} &\leq \sum_{k=0}^x (k + \sigma - 1)^{\sigma-1} (x - k + \sigma)^\sigma \\ &\leq \sum_{k=0}^x (x + \sigma - 1)^{\sigma-1} (x + \sigma)^\sigma \\ &= \underbrace{(x + 1)}_{\in \Pi_1} \underbrace{(x + \sigma - 1)^{\sigma-1}}_{\in \Pi_{\sigma-1}} \underbrace{(x + \sigma)^\sigma}_{\in \Pi_\sigma}. \end{aligned}$$

Also ist $p(x) := (x + 1)(x + \sigma - 1)^{\sigma-1}(x + \sigma)^\sigma$ ein Polynom aus $\Pi_{2\sigma}$. ■

Zur Erinnerung sind wichtige Eigenschaften der Quadratur in den folgenden Aussagen zusammengefasst, die hier jedoch nicht bewiesen werden sollen.

Lemma 6.7 (Exaktheit der Gauß-Quadratur) Seien $n, m \in \mathbb{N}_0$. Eine Quadraturformel \mathcal{Q} heißt exakt für Polynome n -ten Grades oder auch Quadraturformel n -ten Grades, falls sie für Polynome n -ten Grades das exakte Integral berechnet.

Damit ist eine Gauß-Quadraturformel \mathcal{Q}_m der Ordnung m exakt für Polynome des Grades $n = 2m - 1$.

Beweis. [27, Lemma 6.37] ■

Lemma 6.8 (Stabilitätskonstante der Gauß-Quadratur) Sei \mathcal{Q}_m eine Quadraturformel der Ordnung m mit Gewichten w_1, \dots, w_m . Dann heißt

$$C_{\mathcal{Q}_m} := \sum_{\nu=1}^m |w_\nu|$$

Stabilitätskonstante von \mathcal{Q}_m .

Ist \mathcal{Q}_m eine Gauß-Quadraturformel, so gilt $w_\nu > 0$ für alle $\nu \in \{1, \dots, m\}$ und die Stabilitätskonstante ist mit

$$C_{\mathcal{Q}_m} = \sum_{\nu=1}^m w_\nu = 2$$

die kleinstmögliche Stabilitätskonstante für eine Quadraturformel.

Beweis. [27, Bemerkung 6.6] ■

Die wichtigste Aussage ist hier die Fehlerabschätzung für zusammengesetzte Quadraturformeln n -ten Exaktheitsgrades.

Satz 6.9 (Genauigkeit der zusammengesetzten Quadraturformel) Sei \mathcal{Q} eine Quadraturformel n -ten Grades auf dem Intervall $[-1, 1]$ mit ℓ Teilintervallen. Dann gilt

$$|\mathcal{I}(f) - \mathcal{Q}(f)| \leq 2(2 + C_{\mathcal{Q}}) \left(\frac{1}{2\ell}\right)^{n+1} \frac{\|f^{(n+1)}\|_\infty}{(n+1)!}. \quad (6.2)$$

Beweis. folgt für Gauß-Quadratur aus [27, Theorem 6.12] ■

Für die Abschätzung des Quadraturfehlers wird das exakte Integral weiter mit

$$\mathcal{I} : C[-1, 1] \rightarrow \mathbb{R}, \quad f \mapsto \int_{-1}^1 f(x) \, dx$$

bezeichnet, während die m -te Gauß-Quadraturformel \mathcal{Q}_m für ℓ Teilintervalle zusammengefasst als $\mathcal{Q}_{m,\ell}$ geschrieben wird mit

$$\mathcal{Q}_{m,\ell} : C[-1, 1] \rightarrow \mathbb{R}, \quad f \mapsto \sum_{\nu=1}^{m\ell} w_\nu f(x_\nu).$$

Weiterhin wird die asymptotische Glattheit aus Definition 3.24 der Kernfunktion g ausgenutzt. Der folgende Hauptsatz behandelt zunächst den Fall einer Kernfunktion g mit Singularitätsgrad $\sigma \in \mathbb{N}$. Der Fall $\sigma = 0$ wird separat betrachtet.

Satz 6.10 (Quadraturfehler der Green-Quadratur-Methode) Sei $\text{diam}(\Omega) \leq 1$. Sei $g(x, y)$ asymptotisch glatt mit Singularitätsgrad $\sigma \in \mathbb{N}$. Seien damit $\nabla_x g(x, y)$ und $\nabla_y g(x, y)$ asymptotisch glatt in jeder Komponente mit Singularitätsgrad $\sigma + 1$. Der Quadraturfehler für die zusammengesetzte Gauß-Quadratur n -ten Grades mit ℓ Teilintervallen mit je m Quadraturpunkten wie in Formel (5.9) lässt sich abschätzen durch

$$|g(x, y) - \tilde{g}(x, y)| \leq 8 C p(n+1) \left(\frac{c_0}{\ell}\right)^{n+1} \quad (6.3)$$

für alle $x \in \mathcal{B}_\tau$ und $y \in \mathcal{B}_\sigma$ mit $c_0 \geq 1$, $C > 0$ und einem Polynom p .

6.1 Fehlerabschätzung im zweidimensionalen Fall

Beweis. Sei $\iota \in \{1, \dots, 4\}$ fest. Sei der Term unter dem ersten Integral in Formel 5.5 mit f bezeichnet; genauer sei

$$f(t) := g(x, \xi_\iota(t)) \frac{\partial g}{\partial n_z}(\xi_\iota(t), y) \quad (6.4)$$

mit dem exakten Integral

$$\mathcal{I}(f) = \int_{-1}^1 g(x, \xi_\iota(t)) \frac{\partial g}{\partial n_z}(\xi_\iota(t), y) dt.$$

Um den Fehler $|g(x, y) - \tilde{g}(x, y)|$ zu berechnen, wird die Fehlerabschätzung (6.2) auf die Quadraturformel

$$\mathcal{Q}_{m\ell}(f) = \sum_{\nu=1}^{m\ell} w_\nu g(x, \xi_\iota(t_\nu)) \frac{\partial g}{\partial n_z}(\xi_\iota(t_\nu), y)$$

angewendet. Zunächst wird daher $f^{(n+1)}$ berechnet. Dazu sollen die beiden Faktoren von f etwas genauer betrachtet werden:

$g(x, z)$ ist nach Voraussetzung asymptotisch glatt mit Singularitätsgrad σ und Konstanten $C_{as,g}$ und $c_{0,g}$.

Um eine solche Aussage über den zweiten Faktor machen zu können, definiere

$$h(z, y) := \frac{\partial g}{\partial n_z}(z, y).$$

Da $\|n(z)\| = 1$ ist, folgt für eine Richtung p mit der Cauchy-Schwarz-Ungleichung

$$\begin{aligned} |\partial_p^n h(z, y)| &= |\partial_p^n \langle n(z), \nabla_z g(z, y) \rangle| \\ &= |\langle n(z), \partial_p^n \nabla_z g(z, y) \rangle| \\ &\leq \|n(z)\| \|\partial_p^n \nabla_z g(z, y)\| \\ &= \|\partial_p^n \nabla_z g(z, y)\|, \end{aligned}$$

so dass auch hier die Definition 3.24 der asymptotischen Glattheit zutrifft mit Konstanten $C_{as,h}$, $c_{0,h}$ und Singularitätsgrad $\sigma + 1$, da nach Voraussetzung $\nabla_x g(x, y)$ asymptotisch glatt mit $\sigma + 1$ in jeder Komponente ist.

Da $\xi_\iota''(t) \equiv 0$ für alle $\iota \in \{1, \dots, 4\}$ ist, gilt nach Lemma 6.5

$$\frac{d^n}{dt^n} g(x, \xi_\iota(t)) = \left(\partial_{\xi_\iota(t)}^n g \right) (x, \xi_\iota(t)) \quad \text{und} \quad \frac{d^n}{dt^n} h(\xi_\iota(t), y) = \left(\partial_{\xi_\iota(t)}^n h \right) (\xi_\iota(t), y) \quad (6.5)$$

für alle $\iota \in \{1, \dots, 4\}$. Mit dieser Erkenntnis (6.5) und mit Hilfe der Leibniz-Formel kann $|f^{(n+1)}|$ folgendermaßen abgeschätzt werden

$$\begin{aligned} |f^{(n+1)}(t)| &= \left| \frac{d^{n+1}}{dt^{n+1}} \left(g(x, \xi_\iota(t)) \frac{\partial g}{\partial n_z}(\xi_\iota(t), y) \right) \right| \\ &\leq \sum_{k=0}^{n+1} \binom{n+1}{k} \left| \left(\partial_{\xi_\iota(t)}^k g \right) (x, \xi_\iota(t)) \right| \left| \left(\partial_{\xi_\iota(t)}^{n+1-k} h \right) (\xi_\iota(t), y) \right|, \end{aligned}$$

6 Fehleranalyse der Green-Quadratur-Methode

worauf hin dann die asymptotische Glattheit (3.2) der Kernfunktion g sowie der Funktion h ausgenutzt werden kann:

$$|f^{(n+1)}(t)| \leq \sum_{k=0}^{n+1} \binom{n+1}{k} C_{\text{as},g} \frac{(\sigma-1+k)! c_{0,g}^k \|\xi'_l(t)\|^k}{\|x - \xi_l(t)\|^{\sigma+k}} \\ C_{\text{as},h} \frac{(\sigma+1+n-k)! c_{0,h}^{n+1-k} \|\xi'_l(t)\|^{n+1-k}}{\|\xi_l(t) - y\|^{\sigma+n+2-k}}.$$

Definiere $C_{\text{as}} := C_{\text{as},g} C_{\text{as},h}$ und $c_0 := \max\{c_{0,g}, c_{0,h}\}$. Dann gilt weiter

$$|f^{(n+1)}(t)| \leq C_{\text{as}} \sum_{k=0}^{n+1} \binom{n+1}{k} \frac{c_0^{n+1} \|\xi'_l(t)\|^{n+1} (\sigma-1+k)! (\sigma+1+n-k)!}{\|x - \xi_l(t)\|^{\sigma+k} \|\xi_l(t) - y\|^{\sigma+n+2-k}}.$$

Da $\|x - \xi_l(t)\| \geq \text{dist}(x, \Xi)$ und $\|\xi_l(t) - y\| \geq \text{dist}(\Xi, y)$ gilt, und da

$$\delta \leq \min\{\text{dist}(x, \Xi), \text{dist}(\Xi, y)\}$$

nach Lemma 6.2 gilt, folgt

$$|f^{(n+1)}(t)| \leq C_{\text{as}} \sum_{k=0}^{n+1} \binom{n+1}{k} \frac{c_0^{n+1} \|\xi'_l(t)\|^{n+1} (\sigma-1+k)! (\sigma+1+n-k)!}{\delta^{2\sigma+n+2}} \\ = \frac{C_{\text{as}}}{\delta^{2\sigma+1}} \left(\frac{c_0 \|\xi'_l(t)\|}{\delta} \right)^{n+1} \sum_{k=0}^{n+1} \binom{n+1}{k} (\sigma-1+k)! (\sigma+1+n-k)! \\ = \frac{C_{\text{as}}}{\delta^{2\sigma+1}} \left(\frac{c_0 \|\xi'_l(t)\|}{\delta} \right)^{n+1} (n+1)! \sum_{k=0}^{n+1} \frac{(k+\sigma-1)! (n+1-k+\sigma)!}{(n+1-k)! k!} \\ \leq \frac{C_{\text{as}}}{\delta^{2\sigma+1}} \left(\frac{c_0 \|\xi'_l(t)\|}{\delta} \right)^{n+1} (n+1)! p(n+1),$$

wobei $p \in \Pi_{2\sigma}$ nach Lemma 6.6 ist. Es ist $\delta = \frac{1}{2} \text{diam}_{\max}(\mathcal{B}_\tau)$, und nach Lemma 6.1 gilt $\|\xi'_l(t)\| \leq \text{diam}_{\max}(\mathcal{B}_\tau)$, also ergibt sich

$$|f^{(n+1)}(t)| \leq \frac{C_{\text{as}}}{\delta^{2\sigma+1}} \left(\frac{c_0 \text{diam}_{\max}(\mathcal{B}_\tau)}{\frac{1}{2} \text{diam}_{\max}(\mathcal{B}_\tau)} \right)^{n+1} (n+1)! p(n+1) \\ = \frac{C_{\text{as}}}{\delta^{2\sigma+1}} (2c_0)^{n+1} (n+1)! p(n+1).$$

Dieser Ausdruck für $|f^{(n+1)}(t)|$ kann nun in die Fehlerabschätzung (6.2) für die zusammengesetzte Quadratur eingesetzt werden, und dann folgt

$$|\mathcal{I}(f) - \mathcal{Q}(f)| \leq 2(2 + C_Q) \left(\frac{1}{2\ell} \right)^{n+1} \frac{1}{(n+1)!} \frac{C_{\text{as}}}{\delta^{2\sigma+1}} (2c_0)^{n+1} (n+1)! p(n+1) \\ \leq \frac{2(2 + C_Q) C_{\text{as}}}{\delta^{2\sigma+1}} \left(\frac{c_0}{\ell} \right)^{n+1} p(n+1). \quad (6.6)$$

6.1 Fehlerabschätzung im zweidimensionalen Fall

Wenn der Term unter dem zweiten Integral aus (5.5) als f in (6.4) gewählt wird, kommt man zu einem ähnlichen Ergebnis, so dass es zu zweimal dem Fehler (6.6) für jedes der Ξ_1, \dots, Ξ_4 kommt.

Wenn der Fehler für die Kernapproximation insgesamt betrachtet wird, kommt dann noch die Gram'sche Determinante, also in diesem Fall $\|\xi'_\ell(t)\| \leq \text{diam}_{\max}(\mathcal{B}_\tau)$ hinzu, so dass es schließlich auf einen Fehler

$$\begin{aligned} |g(x, y) - \tilde{g}(x, y)| &\leq 4 \cdot 2 \cdot \text{diam}_{\max}(\mathcal{B}_\tau) \cdot \frac{2(2 + C_Q)C_{\text{as}}}{\delta^{2\sigma+1}} \left(\frac{c_0}{\ell}\right)^{n+1} p(n+1) \\ &\leq 8 \cdot 2\delta \cdot \frac{2(2 + C_Q)C_{\text{as}}}{\delta^{2\sigma+1}} \left(\frac{c_0}{\ell}\right)^{n+1} p(n+1) \\ &\leq 8 \frac{4(2 + C_Q)C_{\text{as}}}{\delta^{2\sigma}} \left(\frac{c_0}{\ell}\right)^{n+1} p(n+1) \end{aligned}$$

hinausläuft. Mit $C := \frac{4(2+C_Q)C_{\text{as}}}{\delta^{2\sigma}}$ folgt die Behauptung. ■

Der Satz 6.9 besagt also, dass bei einer Wahl von $\ell > c_0$ die gewünschte exponentielle Konvergenz mit höheren n erreicht wird. Für hinreichend hohe Quadraturordnungen m , also auch höheren $n = 2m - 1$, kann das polynomielle Wachstum des Terms, der im Satz mit p zusammengefasst wird, im Gegensatz zur exponentiellen Konvergenz bei asymptotischen Betrachtungen vernachlässigt werden.

Wie bereits angedeutet, deckt der Satz 6.10 zwar die Fehlerbetrachtung bei Kernfunktionen mit Singularitätsgrad $\sigma > 0$ ab, aber die Kernfunktion (5.1) aus dem zweidimensionalen Modellproblem (2.1) besitzt einen Singularitätsgrad von $\sigma = 0$, was dem logarithmischen Fall entspricht. Daher soll mit dem folgenden Satz nun auch dieser Spezialfall abgehandelt werden.

Satz 6.11 (Quadraturfehler für den Spezialfall $\sigma = 0$) Sei $\text{diam}(\Omega) \leq 1$. Sei $g(x, y)$ asymptotisch glatt mit Singularitätsgrad $\sigma = 0$. Seien damit $\nabla_x g(x, y)$ und $\nabla_y g(x, y)$ asymptotisch glatt in jeder Komponente mit Singularitätsgrad $\sigma = 1$. Der Quadraturfehler für die zusammengesetzte Gauß-Quadratur n -ten Grades mit ℓ Teilintervallen mit je m Quadraturpunkten wie in Formel (5.9) lässt sich abschätzen durch

$$|g(x, y) - \tilde{g}(x, y)| \leq 8 C \left(\frac{|\log \delta|}{2\pi(n+1)} + C_1 + C_2 \log(n+1) \right) \left(\frac{c_0}{\ell}\right)^{n+1} \quad (6.7)$$

für alle $x \in \mathcal{B}_\tau$ und $y \in \mathcal{B}_\sigma$ mit $c_0 \geq 1$, $C > 0$ und $C_1, C_2 > 0$.

Beweis. Der Anfang des Beweises ist genau wie im Beweis des Satzes 6.10 bis zur Anwendung der Leibniz-Formel.

Da in der Definition der asymptotischen Glattheit 3.24 im Fall $\sigma = 0$ ein $n = 0$ ausgeschlossen wird, muss der erste Summand, der durch die Leibniz-Formel entsteht,

6 Fehleranalyse der Green-Quadratur-Methode

separat betrachtet werden. Daher ist erst nach der Umformung

$$\begin{aligned} |f^{(n+1)}(t)| &\leq \sum_{k=0}^{n+1} \binom{n+1}{k} \left| \partial_{\xi'_l(t)}^k g(x, \xi_l(t)) \right| \left| \partial_{\xi'_l(t)}^{n+1-k} h(\xi_l(t), y) \right| \\ &= \binom{n+1}{0} |g(x, \xi_l(t))| \left| \partial_{\xi'_l(t)}^{n+1} h(\xi_l(t), y) \right| \\ &\quad + \sum_{k=1}^{n+1} \binom{n+1}{k} \left| \partial_{\xi'_l(t)}^k g(x, \xi_l(t)) \right| \left| \partial_{\xi'_l(t)}^{n+1-k} h(\xi_l(t), y) \right| \end{aligned}$$

die asymptotische Glattheit anwendbar. Da g asymptotisch glatt mit $\sigma = 0$ und h asymptotisch glatt mit $\sigma = 1$ ist, folgt

$$\begin{aligned} |f^{(n+1)}(t)| &\leq |g(x, \xi_l(t))| C_{as,h} \frac{(1-1+n)! c_{0,h}^{n+1} \|\xi'_l(t)\|^{n+1}}{\|\xi_l(t) - y\|^{1+n+1}} \\ &\quad + \sum_{k=1}^{n+1} \binom{n+1}{k} C_{as,g} \frac{(-1+k)! c_{0,g}^k \|\xi'_l(t)\|^k}{\|x - \xi_l(t)\|^k} \\ &\quad C_{as,h} \frac{(1-1+n+1-k)! c_{0,h}^{n+1-k} \|\xi'_l(t)\|^{n+1-k}}{\|\xi_l(t) - y\|^{1+n+1-k}}. \end{aligned}$$

Mit $c_0 := \max\{c_{0,g}, c_{0,h}\}$ sowie

$$\delta \leq \min\{\|x - \xi_l(t)\|, \|\xi_l(t) - y\|\},$$

wie im Beweis zu Satz 6.10, folgt weiter

$$\begin{aligned} |f^{(n+1)}(t)| &\leq C_{as,h} \left(|g(x, \xi_l(t))| \frac{n! c_0^{n+1} \|\xi'_l(t)\|^{n+1}}{\delta^{n+2}} \right. \\ &\quad \left. + C_{as,g} \sum_{k=1}^{n+1} \binom{n+1}{k} \frac{(k-1)!(n+1-k)! c_0^{n+1} \|\xi'_l(t)\|^{n+1}}{\delta^{n+2}} \right) \\ &= \frac{C_{as,h}}{\delta} \left(\frac{c_0 \|\xi'_l(t)\|}{\delta} \right)^{n+1} \\ &\quad \left(|g(x, \xi_l(t))| n! + C_{as,g} \sum_{k=1}^{n+1} \frac{(n+1)!(k-1)!(n+1-k)!}{(n+1-k)! k!} \right) \\ &= \frac{C_{as,h}}{\delta} \left(\frac{c_0 \|\xi'_l(t)\|}{\delta} \right)^{n+1} (n+1)! \left(\frac{|g(x, \xi_l(t))|}{n+1} + C_{as,g} \sum_{k=1}^{n+1} \frac{1}{k} \right). \end{aligned}$$

Mit Lemma 6.1 und der Definition von δ in (5.6) sowie der Abschätzung aus Lemma 6.3 kann dies weiter abgeschätzt werden durch

$$\begin{aligned} |f^{(n+1)}(t)| &\leq \frac{C_{as,h}}{\delta} (2c_0)^{n+1} (n+1)! \left(\frac{|\log \delta|}{2\pi(n+1)} + C_{as,g} \sum_{k=1}^{n+1} \frac{1}{k} \right) \\ &\leq \frac{C_{as,h}}{\delta} (2c_0)^{n+1} (n+1)! \left(\frac{|\log \delta|}{2\pi(n+1)} + C_{as,g} (\log(n+1) + \tilde{c}) \right) \end{aligned}$$

mit einer Konstanten $\tilde{c} \in \mathcal{O}(\frac{1}{n})$ aus der Abschätzung der Harmonischen Reihe. Einsetzen in die Quadraturfehler-Abschätzung 6.9 liefert schließlich

$$\begin{aligned} |\mathcal{I}(f) - \mathcal{Q}(f)| &\leq 2(2 + C_Q) \left(\frac{1}{2\ell}\right)^{(n+1)} \frac{1}{(n+1)!} \frac{C_{as,h}}{\delta} (2c_0)^{n+1} (n+1)! \\ &\quad \left(\frac{|\log \delta|}{2\pi(n+1)} + C_{as,g}(\log(n+1) + \tilde{c}) \right) \\ &= \frac{2(2 + C_Q)C_{as,h}}{\delta} \left(\frac{c_0}{\ell}\right)^{(n+1)} \left(\frac{|\log \delta|}{2\pi(n+1)} + C_{as,g} \tilde{c} + C_{as,g} \log(n+1) \right). \end{aligned}$$

Auch hier gilt, dass dieser Fehler ebenso bei der Approximation des zweiten Integrals aus (5.5) auftritt und das dann für jedes der vier Teile des Rechtecks Ξ . Für die gesamte Kernapproximation kommt erneut die Gram'sche Determinante $\|\xi'_t(t)\| \leq \text{diam}_{\max}(\mathcal{B}_\tau)$ hinzu, so dass mit den Konstanten $C_1 := C_{as,g} \tilde{c}$ und $C_2 := C_{as,g}$ der Fehler wie folgt aussieht:

$$\begin{aligned} |g(x, y) - \tilde{g}(x, y)| &\leq 4 \cdot 2 \cdot \text{diam}_{\max}(\mathcal{B}_\tau) \cdot \frac{2(2 + C_Q)C_{as,h}}{\delta} \left(\frac{c_0}{\ell}\right)^{(n+1)} \\ &\quad \left(\frac{|\log \delta|}{2\pi(n+1)} + C_1 + C_2 \log(n+1) \right) \\ &= 8 \cdot 4(2 + C_Q)C_{as,h} \left(\frac{c_0}{\ell}\right)^{(n+1)} \left(\frac{|\log \delta|}{2\pi(n+1)} + C_1 + C_2 \log(n+1) \right). \end{aligned}$$

Mit $C := 4(2 + C_Q)C_{as,h}$ folgt die Behauptung. ■

Auch hier erhält man für $\ell > c_0$ die gewünschte exponentielle Konvergenz und die logarithmischen Terme sind dagegen bei asymptotischer Betrachtung vernachlässigbar.

Mit den Sätzen 6.10 und 6.11 ist die Fehleranalyse für speziell den zweidimensionalen Fall mit der in Kapitel 5.2 vorgestellten Parametrisierung ausgeführt.

Im folgenden Abschnitt 6.2 wird eine alternative Fehlerbetrachtung gezeigt, die den dreidimensionalen Fall auch gleich mit einschließt.

6.2 Alternative Fehlerabschätzung

In diesem Abschnitt soll eine alternative Fehlerabschätzung vorgestellt werden. Dabei wird der zwei- und der dreidimensionale Fall gleichzeitig betrachtet, das heißt, im \mathbb{R}^d hat die Rechteck- beziehungsweise Quader-Parametrisierung $2d$ Teile.

Da auch hier wieder die Eigenschaften der Parametrisierung eine Rolle spielen und diese bisher für den dreidimensionalen Fall nicht untersucht wurden, soll dieses im folgenden Abschnitt 6.2.1 nachgeholt werden, so dass die Eigenschaften der zwei- und dreidimensionalen Parametrisierungen in den Beweis zur alternativen Fehlerabschätzung eingehen können, der in Abschnitt 6.2.2 geführt wird.

6.2.1 Eigenschaften der Parametrisierung im dreidimensionalen Fall

Im Folgenden seien ξ_1, \dots, ξ_6 die sechs Teile der Quader-Parametrisierung aus (5.12) mit $\delta := \frac{1}{2} \text{diam}_{\max}(\mathcal{B}_\tau)$ wie in (5.6) zuvor. Dann gelten die folgenden Aussagen.

Lemma 6.12 (Abschätzung der Ableitungen) *Sei $\iota \in \{1, \dots, 6\}$. Dann gilt*

$$\left\| \frac{\partial}{\partial t} \xi_\iota(t, s) \right\| \leq \text{diam}_{\max}(\mathcal{B}_\tau) \quad \text{und} \quad \left\| \frac{\partial}{\partial s} \xi_\iota(t, s) \right\| \leq \text{diam}_{\max}(\mathcal{B}_\tau).$$

Beweis. Sei $\iota = 1$. Dann gilt

$$\frac{\partial}{\partial t} \xi_\iota(t, s) = \begin{pmatrix} -\frac{b_1 - a_1 + 2\delta}{2} \\ 0 \\ 0 \end{pmatrix} \quad \text{und} \quad \frac{\partial}{\partial s} \xi_\iota(t, s) = \begin{pmatrix} 0 \\ \frac{b_2 - a_2 + 2\delta}{2} \\ 0 \end{pmatrix}$$

und mit (5.6) folgt

$$\begin{aligned} \left\| \frac{\partial}{\partial t} \xi_\iota(t, s) \right\| &= \frac{b_1 - a_1 + 2\delta}{2} = \frac{b_1 - a_1 + \text{diam}_{\max}(\mathcal{B}_\tau)}{2} \\ &\leq \frac{2 \text{diam}_{\max}(\mathcal{B}_\tau)}{2} = \text{diam}_{\max}(\mathcal{B}_\tau) \end{aligned}$$

und

$$\begin{aligned} \left\| \frac{\partial}{\partial s} \xi_\iota(t, s) \right\| &= \frac{b_2 - a_2 + 2\delta}{2} = \frac{b_2 - a_2 + \text{diam}_{\max}(\mathcal{B}_\tau)}{2} \\ &\leq \frac{2 \text{diam}_{\max}(\mathcal{B}_\tau)}{2} = \text{diam}_{\max}(\mathcal{B}_\tau). \end{aligned}$$

Für $\iota = 2, 3, 4, 5, 6$ geht der Beweis analog. ■

Die Abstandsabschätzung aus Lemma 6.2 gilt im dreidimensionalen Fall genauso:

Folgerung 6.13 (Abstandsabschätzung) *Sei die max-Zulässigkeitsbedingung aus (4.13) erfüllt und sei $\Xi = \bigcup_{\iota=1}^6 \Xi_\iota$ der durch ξ_1, \dots, ξ_6 parametrisierte Rand des Quaders. Dann gilt*

$$\delta \leq \min\{ \text{dist}(x, \Xi), \text{dist}(\Xi, y) : x \in \mathcal{B}_\tau, y \in \mathcal{B}_\sigma \}.$$

Beweis. Analog zum Beweis von Lemma 6.2. ■

Die Abschätzungen für die Hintereinanderausführung von Kernfunktion und Parametrisierung wie auch für die Normalenableitung davon müssen im dreidimensionalen Fall neu formuliert werden.

Lemma 6.14 (Abschätzung der Kernfunktion) *Gelte die max-Zulässigkeitsbedingung für \mathcal{B}_τ und \mathcal{B}_σ . Seien $x \in \mathcal{B}_\tau$, $y \in \mathcal{B}_\sigma$ und ξ_1, \dots, ξ_6 und δ wie zuvor. Dann gilt*

$$|g(x, \xi_\iota(t, s))| \leq \frac{1}{4\pi\delta} \quad \text{und} \quad |g(\xi_\iota(t, s), y)| \leq \frac{1}{4\pi\delta}.$$

Beweis. Sei $\iota \in \{1, \dots, 6\}$. Es gilt mit Folgerung 6.13

$$\|x - \xi_\iota(t, s)\|_2 \geq \text{dist}(x, \Xi) \geq \delta.$$

Also folgt

$$|g(x, \xi_\iota(t, s))| = \left| \frac{1}{4\pi} \|x - \xi_\iota(t, s)\|^{-1} \right| = \frac{1}{4\pi \|x - \xi_\iota(t, s)\|} \leq \frac{1}{4\pi\delta}.$$

Mit analoger Argumentation folgt auch

$$|g(\xi_\iota(t, s), y)| \leq \frac{1}{4\pi\delta}.$$

■

Lemma 6.15 (Abschätzung der Normalenableitung) *Gelte die max-Zulässigkeitsbedingung für \mathcal{B}_τ und \mathcal{B}_σ . Sei $y \in \mathcal{B}_\sigma$ und ξ_1, \dots, ξ_6 und δ wie zuvor. Dann gilt*

$$\left| \frac{\partial g}{\partial n_z}(\xi_\iota(t, s), y) \right| \leq \frac{1}{4\pi\delta^2}.$$

Beweis. Sei $\iota \in \{1, \dots, 6\}$. Dann gilt für die dreidimensionale Kernfunktion g

$$\frac{\partial g}{\partial z_i}(z, y) = \frac{\partial}{\partial z_i} \left(\frac{1}{4\pi} \|z - y\|^{-1} \right) = \frac{1}{4\pi} \frac{\partial}{\partial z_i} \|z - y\|^{-1} = \frac{y_i - z_i}{4\pi \|z - y\|^3}$$

für alle $i = 1, \dots, 3$. Damit gilt

$$\begin{aligned} \left| \frac{\partial g}{\partial n_z}(\xi_\iota(t, s), y) \right| &= |\langle n(\xi_\iota(t, s)), \nabla_z g(\xi_\iota(t, s), y) \rangle| = \left| \frac{\langle n(\xi_\iota(t, s)), y - \xi_\iota(t, s) \rangle}{4\pi \|\xi_\iota(t, s) - y\|^3} \right| \\ &= \frac{|\langle n(\xi_\iota(t, s)), y - \xi_\iota(t, s) \rangle|}{4\pi \|\xi_\iota(t, s) - y\|^3} \end{aligned}$$

und mit der Cauchy-Schwarz-Ungleichung folgt dann

$$\left| \frac{\partial g}{\partial n_z}(\xi_\iota(t, s), y) \right| \leq \frac{\|n(\xi_\iota(t, s))\| \|\xi_\iota(t, s) - y\|}{4\pi \|\xi_\iota(t, s) - y\|^3} = \frac{\|n(\xi_\iota(t, s))\|}{4\pi \|\xi_\iota(t, s) - y\|^2}.$$

Da $\|n(\xi_\iota(t, s))\| = 1$ und nach Folgerung 6.13 $\|\xi_\iota(t, s) - y\| \geq \text{dist}(\Xi, y) \geq \delta$ gilt, ist schließlich

$$\left| \frac{\partial g}{\partial n_z}(\xi_\iota(t, s), y) \right| \leq \frac{1}{4\pi\delta^2}.$$

■

6.2.2 Allgemeinerer Quadraturfehler-Abschätzung

Vor der alternativen Beweisführung ist noch einmal ein Lemma über eine Polynomabschätzung und zusätzlich ein technisches Lemma zur Abschätzung des Quadraturfehlers in etwas allgemeinerer Form notwendig.

Lemma 6.16 (Polynomabschätzung) Sei $d \in \{2, 3\}$. Seien $\sigma \in \mathbb{N}_0$ und $c_1, c_2 > 0$. Dann existiert für alle $x \in \mathbb{N}$ ein Polynom $p \in \Pi_{\sigma+(d-2)}$ mit

$$c_1 \frac{(\sigma + x + 1)!}{(x + 1)!} (x + 1)^{d-2} + c_2 \frac{(\sigma + x - 1)!}{x!} (x + 1)^{d-1} \leq p(x).$$

Beweis. Durch Induktion über σ . Sei $\sigma = 0$. Da $x \geq 1$ gilt dann für $d = 2$

$$\begin{aligned} c_1 \frac{(x + 1)!}{(x + 1)!} (x - 1)^0 + c_2 \frac{(x - 1)!}{x!} (x + 1)^1 &= c_1 + c_2 \frac{x + 1}{x} \\ &\leq c_1 + c_2 + \frac{c_2}{x} \\ &\leq c_1 + 2c_2 \in \Pi_0 = \Pi_{\sigma+(d-2)} \end{aligned}$$

und für $d = 3$ kann man damit dann folgern, dass

$$\begin{aligned} c_1(x - 1) + c_2 \frac{(x - 1)^2}{x} &= \left(c_1 + c_2 \frac{x - 1}{x} \right) (x - 1) \\ &\leq (c_1 + c_2)(x - 1) \in \Pi_1 = \Pi_{\sigma+(d-2)}. \end{aligned}$$

Sei $\sigma = 1$. Dann gilt für $d = 2$

$$\begin{aligned} c_1 \frac{(x + 2)!}{(x + 1)!} + c_2 \frac{x!}{x!} (x + 1) &= c_1(x + 2) + c_2(x + 1) \\ &= (c_1 + c_2)x + 2c_1 + c_2 \in \Pi_1 = \Pi_{\sigma+(d-2)} \end{aligned}$$

und damit für $d = 3$

$$c_1(x + 2)(x + 1) + c_2(x + 1)^2 = ((c_1 + c_2)x + 2c_1 + c_2)(x + 1) \in \Pi_2 = \Pi_{\sigma+(d-2)}.$$

Gelte die Behauptung für ein $\sigma \in \mathbb{N}_0$. Für $\sigma + 1$ gilt

$$\begin{aligned} &c_1 \frac{(\sigma + x + 2)!}{(x + 1)!} (x + 1)^{d-2} + c_2 \frac{(\sigma + x)!}{x!} (x + 1)^{d-1} \\ &= c_1 \frac{(\sigma + x + 1)!}{(x + 1)!} (\sigma + x + 2)(x + 1)^{d-2} + c_2 \frac{(\sigma + x - 1)!}{x!} (\sigma + x)(x + 1)^{d-1} \\ &\leq \left(c_1 \frac{(\sigma + x + 1)!}{(x + 1)!} (x + 1)^{d-2} + c_2 \frac{(\sigma + x - 1)!}{x!} (x + 1)^{d-1} \right) (\sigma + x + 2). \end{aligned}$$

Da nach Induktionsvoraussetzung ein $p \in \Pi_{\sigma+(d-2)}$ existiert, das größer gleich dem ersten Faktor ist, folgt schließlich

$$c_1 \frac{(\sigma + x + 2)!}{(x + 1)!} (x + 1)^{d-2} + c_2 \frac{(\sigma + x)!}{x!} (x + 1)^{d-1} \leq p(x) \cdot (\sigma + x + 2) \in \Pi_{\sigma+1+(d-2)}.$$

■

Lemma 6.17 (Quadraturfehler für Produkte) Sei $U = [-1, 1]^{d-1}$ und $M := \{1, \dots, \ell\}^{d-1}$. Sei $U := \bigcup_{j \in M} U_j$ eine disjunkte Zerlegung von U mit

$$U_j = \bigotimes_{\iota=1}^{d-1} \left[\frac{2}{\ell}(j_\iota - 1) - 1, \frac{2}{\ell}j_\iota - 1 \right] \quad (6.8)$$

wobei $j_\iota \in [0, 1]$. Seien $f_1 : U \rightarrow \mathbb{R}$ und $f_2 : U \rightarrow \mathbb{R}$ zwei Funktionen aus $C^m(U)$ und $f := f_1 f_2$. Sei mit

$$\mathcal{I}(f) = \int_U f(z) \, dz$$

das exakte Integral von f bezeichnet und sei

$$\mathcal{Q}(f) = \sum_{j \in M} \mathcal{Q}_{U_j}(f) = \sum_{j \in M} \sum_{\nu} w_\nu f(z_\nu)$$

die Tensor-Gauß-Quadraturformel für die Quadraturordnung m mit ℓ Teilintervallen. Sei mit Λ die Stabilitätskonstante der Tschebyscheff-Interpolation bezeichnet. Dann gilt

$$\begin{aligned} |\mathcal{I}(f) - \mathcal{Q}(f)| &\leq \frac{4}{m!} \left(\frac{\text{diam}_{\max} U}{4\ell} \right)^m \\ &\quad \sum_{j \in M} |U_j| \left(\|f_1\|_{\infty, U_j} \frac{\text{diam}_{\max} U}{4\ell(m+1)} \sum_{i=1}^d \Lambda^{i-1} \|\partial_i^{m+1} f_2\|_{\infty, U_j} \right. \\ &\quad \left. + \Lambda \|f_2\|_{\infty, U_j} \sum_{i=1}^d \Lambda^{i-1} \|\partial_i^m f_1\|_{\infty, U_j} \right). \end{aligned}$$

Beweis. Es gilt

$$|\mathcal{I}(f) - \mathcal{Q}(f)| = |\mathcal{I}(f_1 f_2) - \mathcal{Q}(f_1 f_2)| = \left| \sum_{j \in M} \mathcal{I}_{U_j}(f_1 f_2) - \mathcal{Q}_{U_j}(f_1 f_2) \right|.$$

Da \mathcal{Q}_{U_j} exakt für Polynome vom Grad $2m-1$ ist nach Lemma 6.7, gilt $\mathcal{I}_{U_j}(p) = \mathcal{Q}_{U_j}(p)$ für $p \in \Pi_{2m-1}$ und es folgt

$$\begin{aligned} |\mathcal{I}(f) - \mathcal{Q}(f)| &= \left| \sum_{j \in M} \mathcal{I}_{U_j}(f_1 f_2) - \mathcal{I}_{U_j}(p) + \mathcal{Q}_{U_j}(p) - \mathcal{Q}_{U_j}(f_1 f_2) \right| \\ &\leq \sum_{j \in M} |\mathcal{I}_{U_j}(f_1 f_2 - p)| + |\mathcal{Q}_{U_j}(p - f_1 f_2)|. \end{aligned}$$

Wegen

$$\begin{aligned} |\mathcal{I}_{U_j}(f_1 f_2 - p)| &= \left| \int_{U_j} f_1 f_2(z) - p(z) \, dz \right| \leq \int_{U_j} |f_1 f_2(z) - p(z)| \, dz \\ &\leq \int_{U_j} \|f_1 f_2 - p\|_{\infty, U_j} \, dz = |U_j| \|f_1 f_2 - p\|_{\infty, U_j} \end{aligned}$$

6 Fehleranalyse der Green-Quadratur-Methode

und

$$\begin{aligned} |\mathcal{Q}_{U_j}(p - f_1 f_2)| &= \left| \sum_{\nu} w_{\nu} (p(z_{\nu}) - f_1 f_2(z_{\nu})) \right| \leq \sum_{\nu} |w_{\nu}| |p(z_{\nu}) - f_1 f_2(z_{\nu})| \\ &\leq \sum_{\nu} |w_{\nu}| \|p - f_1 f_2\|_{\infty, U_j} = C_{\mathcal{Q}_{U_j}} \|p - f_1 f_2\|_{\infty, U_j} \end{aligned}$$

und da die Gauß-Quadratur genutzt wird mit $C_{\mathcal{Q}_{U_j}} = |U_j|$, folgt weiter

$$|\mathcal{I}(f) - \mathcal{Q}(f)| \leq \sum_{j \in M} 2|U_j| \|f_1 f_2 - p\|_{\infty, U_j}. \quad (6.9)$$

Um eine Abschätzung für $\|f_1 f_2 - p\|_{\infty, U_j}$ mit $p \in \Pi_{2m-1}$ zu erhalten, werden f_1 und f_2 durch Polynome $p_1 \in \Pi_{m-1}$ und $p_2 \in \Pi_m$ mit Hilfe der Tschebyscheff-Interpolation approximiert mit $p := p_1 p_2$.

Mit der Abschätzung des Interpolationsfehlers aus Satz 4.5 und da $\text{diam}_{\max} U_j = \frac{\text{diam}_{\max} U}{\ell}$ ist, gilt

$$\begin{aligned} \|f_1 - p_1\|_{\infty, U_j} &\leq 2 \sum_{i=1}^d \Lambda^{i-1} \left(\frac{\text{diam}_{\max} U_j}{4} \right)^m \frac{\|\partial_i^m f_1\|_{\infty, U_j}}{m!} \\ &\leq \frac{2}{m!} \left(\frac{\text{diam}_{\max} U}{4\ell} \right)^m \sum_{i=1}^d \Lambda^{i-1} \|\partial_i^m f_1\|_{\infty, U_j} \end{aligned} \quad (6.10)$$

und ebenso

$$\|f_2 - p_2\|_{\infty, U_j} \leq \frac{2}{(m+1)!} \left(\frac{\text{diam}_{\max} U}{4\ell} \right)^{m+1} \sum_{i=1}^d \Lambda^{i-1} \|\partial_i^{m+1} f_2\|_{\infty, U_j}. \quad (6.11)$$

Damit folgt nun

$$\begin{aligned} \|f_1 f_2 - p\|_{\infty, U_j} &= \|f_1 f_2 - p_1 p_2\|_{\infty, U_j} \\ &= \|f_1 f_2 - p_2 f_1 + p_2 f_1 - p_1 p_2\|_{\infty, U_j} \\ &\leq \|f_1 f_2 - p_2 f_1\|_{\infty, U_j} + \|p_2 f_1 - p_1 p_2\|_{\infty, U_j} \\ &\leq \|f_1\|_{\infty, U_j} \|f_2 - p_2\|_{\infty, U_j} + \|p_2\|_{\infty, U_j} \|f_1 - p_1\|_{\infty, U_j}. \end{aligned}$$

Mit (6.10) und (6.11) und der Tatsache, dass $\|p_2\| \leq \Lambda \|f_2\|$ ist, kann dies weiter abgeschätzt werden zu

$$\begin{aligned} \|f_1 f_2 - p\|_{\infty, U_j} &\leq \|f_1\|_{\infty, U_j} \left(\frac{2}{(m+1)!} \left(\frac{\text{diam}_{\max} U}{4\ell} \right)^{m+1} \sum_{i=1}^d \Lambda^{i-1} \|\partial_i^{m+1} f_2\|_{\infty, U_j} \right) \\ &\quad + \Lambda \|f_2\|_{\infty, U_j} \left(\frac{2}{m!} \left(\frac{\text{diam}_{\max} U}{4\ell} \right)^m \sum_{i=1}^d \Lambda^{i-1} \|\partial_i^m f_1\|_{\infty, U_j} \right) \\ &= \frac{2}{m!} \left(\frac{\text{diam}_{\max} U}{4\ell} \right)^m \left(\|f_1\|_{\infty, U_j} \frac{\text{diam}_{\max} U}{4\ell(m+1)} \sum_{i=1}^d \Lambda^{i-1} \|\partial_i^{m+1} f_2\|_{\infty, U_j} \right. \\ &\quad \left. + \Lambda \|f_2\|_{\infty, U_j} \sum_{i=1}^d \Lambda^{i-1} \|\partial_i^m f_1\|_{\infty, U_j} \right). \end{aligned}$$

Einsetzen in (6.9) liefert schließlich die Behauptung. \blacksquare

Mit diesen Vorbereitungen kann nun die alternative Fehlerabschätzung formuliert werden.

Satz 6.18 (Alternative Abschätzung des Quadraturfehlers) Sei $\text{diam}(\Omega) \leq 1$ mit der Dimension $d \in \{2, 3\}$. Sei $U = [-1, 1]^{d-1}$ und sei $f := f_1 f_2$ die Funktion mit

$$f_1 : U \rightarrow \mathbb{R}, \quad t \mapsto g(x, \xi_\iota(t)) \quad \text{und} \quad f_2 : U \rightarrow \mathbb{R}, \quad t \mapsto \frac{\partial g}{\partial n_z}(\xi_\iota(t), y),$$

wobei für $d = 3$ und $U \subseteq \mathbb{R}^2$ gilt: $t := (t_1, t_2)^\top$. Seien $\xi_\iota : U \rightarrow \Xi_\iota$ mit $\frac{\partial^2}{\partial t_i^2} \xi \equiv 0$ für $i = \{1, \dots, d-1\}$ die $\iota = 1, \dots, 2d$ Teile der Parametrisierung. Sei $M := \{1, \dots, \ell\}^{d-1}$ und $U := \bigcup_{j \in M} U_j$ die disjunkte Zerlegung von U aus (6.8). Dann gilt für den Quadraturfehler der zusammengesetzten Gauß-Quadratur mit m Quadraturpunkten und ℓ Teilintervallen

$$|g(x, y) - \tilde{g}(x, y)| \leq 4d C p(m) \left(\frac{c_0}{\ell} \right)^m$$

für alle $x \in \mathcal{B}_\tau$ und $y \in \mathcal{B}_\sigma$ mit $c_0 \geq 1$, $C > 0$ und einem Polynom p .

Beweis. Nach Lemma 6.17 gilt

$$\begin{aligned} |\mathcal{I}(f) - \mathcal{Q}(f)| &\leq \frac{4}{m!} \left(\frac{\text{diam}_{\max} U}{4\ell} \right)^m \sum_{j \in M} |U_j| \\ &\quad \left(\|f_1\|_{\infty, U_j} \frac{\text{diam}_{\max} U}{4\ell(m+1)} \sum_{i=1}^{d-1} \Lambda^{i-1} \left\| \frac{\partial^{m+1}}{\partial t_i^{m+1}} f_2 \right\|_{\infty, U_j} + \Lambda \|f_2\|_{\infty, U_j} \sum_{i=1}^{d-1} \Lambda^{i-1} \left\| \frac{\partial^m}{\partial t_i^m} f_1 \right\|_{\infty, U_j} \right). \end{aligned}$$

Sei wieder $h(z, y) := \frac{\partial g}{\partial n_z}(z, y)$ wie im Beweis von 6.10. Dann gilt nach Lemma 6.5 für $i = \{1, \dots, d-1\}$

$$\frac{\partial^m}{\partial t_i^m} g(x, \xi_\iota(t)) = \left(\frac{\partial^m}{\partial t_i^m} g \right) (x, \xi_\iota(t)) \quad \text{und} \quad \frac{\partial^m}{\partial t_i^m} h(\xi_\iota(t), y) = \left(\frac{\partial^m}{\partial t_i^m} h \right) (\xi_\iota(t), y)$$

6 Fehleranalyse der Green-Quadratur-Methode

und es folgt

$$\begin{aligned}
 |\mathcal{I}(f) - \mathcal{Q}(f)| &\leq \frac{4}{m!} \left(\frac{2}{4\ell}\right)^m \sum_{j \in M} |U_j| \\
 &\quad \left(\|g(x, \xi_\iota(t))\|_{\infty, U_j} \frac{2}{4\ell(m+1)} \sum_{i=1}^{d-1} \Lambda^{i-1} \left\| \left(\frac{\partial^{m+1}}{\partial t_i} h \right) (\xi_\iota(t), y) \right\|_{\infty, U_j} \right. \\
 &\quad \left. + \Lambda \|h(\xi_\iota(t), y)\|_{\infty, U_j} \sum_{i=1}^{d-1} \Lambda^{i-1} \left\| \left(\frac{\partial^m}{\partial t_i} g \right) (x, \xi_\iota(t)) \right\|_{\infty, U_j} \right).
 \end{aligned}$$

Nun kann die asymptotische Glattheit von g mit Singularitätsgrad σ und von h mit Singularitätsgrad $\sigma + 1$ angewendet werden:

$$\begin{aligned}
 |\mathcal{I}(f) - \mathcal{Q}(f)| &\leq \frac{4}{m!} \left(\frac{1}{2\ell}\right)^m \sum_{j \in M} |U_j| \\
 &\quad \left(\|g(x, \xi_\iota(t))\|_{\infty, U_j} \frac{1}{2\ell(m+1)} \sum_{i=1}^{d-1} \Lambda^{i-1} C_{\text{as}, h} \frac{(\sigma + m + 1)! c_{0, h}^{m+1} \left\| \frac{\partial}{\partial t_i} \xi_\iota(t) \right\|^{m+1}}{\|\xi_\iota(t) - y\|^{\sigma+m+2}} \right. \\
 &\quad \left. + \Lambda \|h(\xi_\iota(t), y)\|_{\infty, U_j} \sum_{i=1}^{d-1} \Lambda^{i-1} C_{\text{as}, g} \frac{(\sigma - 1 + m)! c_{0, g}^m \left\| \frac{\partial}{\partial t_i} \xi_\iota(t) \right\|^m}{\|x - \xi_\iota(t)\|^{\sigma+m}} \right).
 \end{aligned}$$

Mit $c_0 := \max\{c_{0, g}, c_{0, h}\}$, $C_{\text{as}} := \max\{C_{\text{as}, g}, C_{\text{as}, h}\}$ und wegen $\delta \leq \|x - \xi_\iota(t)\|$, $\|\xi_\iota(t) - y\|$ folgt daraus

$$\begin{aligned}
 |\mathcal{I}(f) - \mathcal{Q}(f)| &\leq \frac{4C_{\text{as}}}{m!} \left(\frac{c_0}{2\delta\ell}\right)^m \sum_{j \in M} |U_j| \\
 &\quad \left(\|g(x, \xi_\iota(t))\|_{\infty, U_j} \frac{c_0}{2\delta^{\sigma+2}\ell(m+1)} \sum_{i=1}^{d-1} \Lambda^{i-1} (\sigma + m + 1)! \left\| \frac{\partial}{\partial t_i} \xi_\iota(t) \right\|^{m+1} \right. \\
 &\quad \left. + \frac{\Lambda}{\delta^\sigma} \|h(\xi_\iota(t), y)\|_{\infty, U_j} \sum_{i=1}^{d-1} \Lambda^{i-1} (\sigma - 1 + m)! \left\| \frac{\partial}{\partial t_i} \xi_\iota(t) \right\|^m \right)
 \end{aligned}$$

und da nach Lemma 6.1 und Lemma 6.12 $\left\| \frac{\partial}{\partial t_i} \xi_\iota(t) \right\| \leq \text{diam}_{\max}(\mathcal{B}_\tau) = 2\delta$ für $i = 1, \dots, d-1$ gilt, ergibt sich

$$\begin{aligned} |\mathcal{I}(f) - \mathcal{Q}(f)| &\leq \frac{4C_{\text{as}}}{\delta^\sigma} \left(\frac{2\delta c_0}{2\delta\ell} \right)^m \sum_{j \in M} |U_j| \left(\|g(x, \xi_\iota(t))\|_{\infty, U_j} \frac{2\delta c_0 (\sigma + m + 1)!}{2\delta^2 \ell (m + 1)!} \sum_{i=1}^{d-1} \Lambda^{i-1} \right. \\ &\quad \left. + \Lambda \|h(\xi_\iota(t), y)\|_{\infty, U_j} \frac{(\sigma - 1 + m)!}{m!} \sum_{i=1}^{d-1} \Lambda^{i-1} \right) \\ &\leq \frac{4C_{\text{as}}}{\delta^\sigma} \left(\frac{c_0}{\ell} \right)^m \sum_{j \in M} |U_j| \left(\|g(x, \xi_\iota(t))\|_{\infty, U_j} \frac{c_0 (\sigma + m + 1)!}{\delta \ell (m + 1)!} (d-1) \Lambda^{d-2} \right. \\ &\quad \left. + \|h(\xi_\iota(t), y)\|_{\infty, U_j} \frac{(\sigma - 1 + m)!}{m!} (d-1) \Lambda^{d-1} \right). \end{aligned}$$

Nach [7, Example 4.12] kann die Stabilitätskonstante der Tschebyscheff-Interpolation einfach abgeschätzt werden durch $\Lambda \leq m + 1$, also folgt

$$\begin{aligned} |\mathcal{I}(f) - \mathcal{Q}(f)| &\leq \frac{4C_{\text{as}}(d-1)}{\delta^\sigma} \left(\frac{c_0}{\ell} \right)^m \sum_{j \in M} |U_j| \left(\|g(x, \xi_\iota(t))\|_{\infty, U_j} \frac{c_0 (\sigma + m + 1)!}{\delta \ell (m + 1)!} (m + 1)^{d-2} \right. \\ &\quad \left. + \|h(\xi_\iota(t), y)\|_{\infty, U_j} \frac{(\sigma - 1 + m)!}{m!} (m + 1)^{d-1} \right). \end{aligned}$$

An dieser Stelle ist eine Fallunterscheidung nach der Dimension d erforderlich. Nach Lemma 6.3 und Lemma 6.14 gilt

$$\|g(x, \xi_\iota(t))\|_{\infty, U_j} \leq \begin{cases} \frac{|\log \delta|}{2\pi} & \text{für } d = 2 \\ \frac{1}{4\pi\delta} & \text{für } d = 3 \end{cases}.$$

Ebenso gilt nach Lemma 6.4 und Lemma 6.15

$$\|h(\xi_\iota(t), y)\|_{\infty, U_j} \leq \begin{cases} \frac{1}{2\pi\delta} & \text{für } d = 2 \\ \frac{1}{4\pi\delta^2} & \text{für } d = 3 \end{cases}.$$

Damit folgt für $d = 3$ und mit der vorausgesetzten Definition der U_j als disjunkte Partition von U , dass

$$\begin{aligned} |\mathcal{I}(f) - \mathcal{Q}(f)| &\leq \frac{4C_{\text{as}}(d-1)}{\delta^\sigma} |U| \left(\frac{c_0}{\ell} \right)^m \left(\frac{c_0 (\sigma + m + 1)!}{4\pi\delta^2 \ell (m + 1)!} (m + 1)^{d-2} \right. \\ &\quad \left. + \frac{1}{4\pi\delta^2} \frac{(\sigma - 1 + m)!}{m!} (m + 1)^{d-1} \right) \\ &= \frac{C_{\text{as}}(d-1)}{\pi\delta^{\sigma+2}} |U| \left(\frac{c_0}{\ell} \right)^m \left(\frac{c_0 (\sigma + m + 1)!}{\ell (m + 1)!} (m + 1)^{d-2} \right. \\ &\quad \left. + \frac{(\sigma - 1 + m)!}{m!} (m + 1)^{d-1} \right) \\ &\leq C_{3d}(\delta) \left(\frac{c_0}{\ell} \right)^m q_1(m) \end{aligned} \tag{6.12}$$

6 Fehleranalyse der Green-Quadratur-Methode

nach Lemma 6.16 mit einem Polynom $q_1 \in \Pi_{\sigma+(d-2)}$ und mit $C_{3d}(\delta) := \frac{C_{\text{as}}(d-1)}{\pi\delta^{\sigma+2}}|U|$.

Für $d = 2$ folgt

$$\begin{aligned} |\mathcal{I}(f) - \mathcal{Q}(f)| &\leq \frac{4C_{\text{as}}(d-1)}{\delta^\sigma} |U| \left(\frac{c_0}{\ell}\right)^m \left(\frac{c_0 |\log \delta|}{2\pi\delta\ell} \frac{(\sigma+m+1)!}{(m+1)!} (m+1)^{d-2} \right. \\ &\quad \left. + \frac{1}{2\pi\delta} \frac{(\sigma-1+m)!}{m!} (m+1)^{d-1} \right) \\ &= \frac{2C_{\text{as}}(d-1)}{\pi\delta^{\sigma+1}} |U| \left(\frac{c_0}{\ell}\right)^m \left(\frac{c_0 |\log \delta|}{\ell} \frac{(\sigma+m+1)!}{(m+1)!} (m+1)^{d-2} \right. \\ &\quad \left. + \frac{(\sigma-1+m)!}{m!} (m+1)^{d-1} \right). \end{aligned}$$

Wegen $\text{diam}(\Omega) \leq 1$ und $\delta = \frac{1}{2} \text{diam}_{\max}(\mathcal{B}_\tau)$ ist $\delta \leq \frac{1}{2}$ und damit gilt

$$1 = \frac{1}{\left|\log \frac{1}{2}\right|} \left| \log \frac{1}{2} \right| \leq \frac{1}{\left|\log \frac{1}{2}\right|} |\log \delta|,$$

so dass sich die Gleichung weiter abschätzen lässt durch

$$\begin{aligned} |\mathcal{I}(f) - \mathcal{Q}(f)| &\leq \frac{2C_{\text{as}}(d-1)|\log \delta|}{\pi\delta^{\sigma+1}} |U| \left(\frac{c_0}{\ell}\right)^m \left(\frac{c_0}{\ell} \frac{(\sigma+m+1)!}{(m+1)!} (m+1)^{d-2} \right. \\ &\quad \left. + \frac{1}{\left|\log \frac{1}{2}\right|} \frac{(\sigma-1+m)!}{m!} (m+1)^{d-1} \right) \tag{6.13} \\ &\leq C_{2d}(\delta) \left(\frac{c_0}{\ell}\right)^m q_2(m) \end{aligned}$$

nach Lemma 6.16 mit einem Polynom $q_2 \in \Pi_{\sigma+(d-2)}$ und mit $C_{2d}(\delta) := 2|\log \delta|\delta C_{3d}(\delta)$. Der Fehler (6.13) beziehungsweise (6.12) tritt ebenso bei der Quadratur für das zweite Integral aus (5.5) auf, und das dann für jeden der $2d$ Teile der Parametrisierung. Für die gesamte Kernapproximation kommt dann noch die Gram'sche Determinante

$$\begin{aligned} G_{\text{det}} &:= \begin{cases} \left\| \frac{\partial}{\partial t} \xi_\iota(t) \right\| \leq \text{diam}_{\max}(\mathcal{B}_\tau) & \text{für } d = 2 \\ \left\| \frac{\partial}{\partial t_1} \xi_\iota(t) \times \frac{\partial}{\partial t_2} \xi_\iota(t) \right\| \leq (\text{diam}_{\max}(\mathcal{B}_\tau))^2 & \text{für } d = 3 \end{cases} \\ &\leq (\text{diam}_{\max}(\mathcal{B}_\tau))^{d-1} \\ &= 2\delta^{d-1} \end{aligned}$$

hinzu, so dass schließlich mit $\tilde{C}(\delta) := \max\{C_{2d}(\delta), C_{3d}(\delta)\}$, $C := 2\delta^{d-1}\tilde{C}(\delta)$ und mit einem Polynom $p \in \Pi_{\sigma+(d-2)}$ mit $p(m) \geq \max\{q_1(m), q_2(m)\}$

$$\begin{aligned} |g(x, y) - \tilde{g}(x, y)| &\leq 2d \cdot 2 \cdot 2\delta^{d-1} \tilde{C}(\delta) \left(\frac{c_0}{\ell}\right)^m p(m) \\ &= 4d C \left(\frac{c_0}{\ell}\right)^m p(m) \end{aligned}$$

folgt, womit die Behauptung bewiesen ist. ■

7 Numerische Experimente für das Green-Quadratur-Verfahren

Nachdem im vorangegangenen Kaptitel 6 das Green-Quadratur-Verfahren von der Theorie her analysiert wurde, soll dieses Kapitel dazu dienen, das Verfahren in der praktischen Anwendung zu untersuchen.

Die Implementierung der Green-Quadratur-Methode ist im Rahmen der Programm-bibliothek *H2Lib* der Arbeitsgruppe von Steffen Börm entstanden. Die *H2Lib* ist in der Sprache C geschrieben und greift auf Funktionen der BLAS- und LAPACK-Softwarebibliotheken zurück.

Als Testsystem für die folgenden Experimente diene eine Maschine mit NUMA-Architektur und AMD Opteron Prozessoren mit 48 CPU-Kernen zu je 2,4 GHz und mit 512 GB Arbeitsspeicher.

Im folgenden Abschnitt 7.1 wird der durchaus vorzeigbare zweidimensionale Fall des Verfahrens in ausführlicher Weise betrachtet und der eher weniger konkurrenzfähige dreidimensionale Fall getrennt davon in Abschnitt 7.2.

7.1 2D-Fall

Um dieses neue Verfahren in der Anwendung etwas kennenzulernen, wird in Abschnitt 7.1.1 zunächst der Einfluss der verschiedenen Parameter auf die Effizienz des Algorithmus untersucht und erst danach in Abschnitt 7.1.2 das Verfahren anderen analytischen Methoden gegenübergestellt.

7.1.1 Verfahrensanalysen

Für die numerischen Experimente in diesem Abschnitt 7.1.1 und auch für die ersten Experimente in Abschnitt 7.1.2 sei ein Randintegral-Operator betrachtet, diskretisiert durch das Galerkin-Verfahren mit stückweise konstanten Basisfunktionen auf einer polygonalen Approximation des Kreises Ω . Die Dimension der resultierenden Matrix ist mit n bezeichnet.

Im ersten Experiment geht es um einen Vergleich der verschiedenen Parametrisierungen. Die Anzahl der Freiheitsgrade ist dabei $n = 65\,536$. Die in Abschnitt 5.2 ausführlich beschriebene Rechteck-Parametrisierung und die Kreis-Parametrisierung aus Abschnitt 5.3.2 werden in Abbildung 7.1 für unterschiedlich viele Teilintervalle ℓ und für verschiedene Quadraturordnungen m in der linken Teilabbildung und für verschiedene Ränge k in der rechten Teilabbildung verglichen.

7 Numerische Experimente für das Green-Quadratur-Verfahren

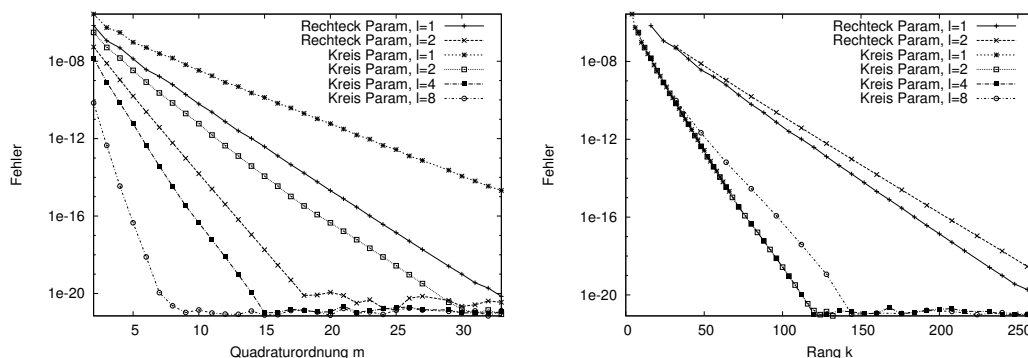


Abbildung 7.1: Kreis-Geometrie, $n = 65\,536$: Green-Quadratur-Methode, $\delta = 0.5 \text{ diam}$ für die Rechteck-Parametrisierung und $\delta = 0.4 \text{ diam}$ für die Kreis-Parametrisierung, maximale Blattgröße $clf = 2k$, Vergleich der Quadraturordnungen m (links) und der entstehenden Ränge k (rechts)

Dass in der rechten Teilabbildung die Kurven für die Kreis-Parametrisierung für verschiedene ℓ alle übereinander verlaufen, liegt an der äquidistanten Verteilung der Quadraturpunkte auf dem Kreis, während auf dem Rechteck Gauß-Quadraturpunkte gewählt werden. Es ist in dem Fall gleich, ob beim Kreis m oder ℓ erhöht wird.

Auch wenn die Kreis-Parametrisierung im zweidimensionalen Fall in der Praxis Vorteile hat, wird im Folgenden nur die Rechteck-Parametrisierung eingehender untersucht. Die Gründe dafür wurden schon am Ende von Abschnitt 5.3.2 erwähnt: Die rechenintensive äquidistante Verteilung der Quadraturpunkte auf der Kugeloberfläche im dreidimensionalen Fall und die Tatsache, dass die Fehlerabschätzungen hier für die Rechteck-Parametrisierung ausgeführt wurden und sie im Falle der Kreis-Parametrisierung unhandlicher werden.

| n | $m = 4$ | $m = 5$ | $m = 6$ | $m = 7$ | $m = 8$ | $m = 9$ | $m = 10$ | Faktor |
|--------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------|
| 1024 | 1.66_{-06} | 4.30_{-07} | 8.21_{-08} | 3.53_{-08} | 1.01_{-08} | 2.68_{-09} | 1.12_{-09} | 3.38 |
| 2048 | 1.11_{-06} | 2.86_{-07} | 7.00_{-08} | 2.93_{-08} | 9.82_{-09} | 2.87_{-09} | 9.82_{-10} | 3.23 |
| 4096 | 7.02_{-07} | 1.87_{-07} | 5.55_{-08} | 2.13_{-08} | 7.52_{-09} | 2.37_{-09} | 7.85_{-10} | 3.10 |
| 8192 | 3.64_{-07} | 9.52_{-08} | 2.82_{-08} | 1.20_{-08} | 4.31_{-09} | 1.31_{-09} | 4.49_{-10} | 3.05 |
| 16384 | 1.89_{-07} | 5.20_{-08} | 1.51_{-08} | 6.33_{-09} | 2.09_{-09} | 7.32_{-10} | 2.53_{-10} | 3.01 |
| 32768 | 9.41_{-08} | 2.58_{-08} | 7.52_{-09} | 3.17_{-09} | 1.17_{-09} | 3.64_{-10} | 1.23_{-10} | 3.02 |
| 65536 | 4.86_{-08} | 1.34_{-08} | 3.66_{-09} | 1.65_{-09} | 6.18_{-10} | 1.90_{-10} | 6.24_{-11} | 3.03 |
| 131072 | 2.86_{-08} | 8.68_{-09} | 2.47_{-09} | 1.00_{-09} | 3.67_{-10} | 1.17_{-10} | 3.95_{-11} | 3.00 |

Tabelle 7.1: Approximationsfehler: $\|G - \tilde{G}\|_2$ für $\ell = 1$ (SLP)

In den folgenden Tabellen sind die Matrix-Approximationsfehler $\|G - \tilde{G}\|_2$ für verschiedene Quadraturordnungen m und verschiedene Freiheitsgrade n dargestellt. Die

mit ‘‘Faktor’’ bezeichnete Spalte in den Tabellen gibt das geometrische Mittel der Fehlerreduktion mit h"oher werdendem m f"ur ein bestimmtes n an.

Tabelle 7.1 zeigt die Fehlerentwicklung f"ur $\ell = 1$. Der Fehler verh"alt sich in etwa wie $5 \cdot 10^{-5} \cdot q^m$ mit $q \approx e^{-1} \approx 0,37$. In Tabelle 7.2 ist der Fehler f"ur $\ell = 2$ zu sehen und der entwickelt sich wie $10^{-5} \cdot q^m$ mit $q \approx e^{-1,8} \approx 0,17$. F"ur $\ell = 3$ sind die Fehler in Tabelle 7.3 aufgef"uhrt mit in etwa $5 \cdot 10^{-6} \cdot q^m$ mit $q \approx e^{-2,5} \approx 0,08$.

| n | $m = 4$ | $m = 5$ | $m = 6$ | $m = 7$ | $m = 8$ | $m = 9$ | $m = 10$ | Faktor |
|--------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|--------|
| 1024 | 2.92 ₋₀₈ | 4.44 ₋₀₉ | 3.30 ₋₁₀ | 5.30 ₋₁₁ | – | – | – | 8.20 |
| 2048 | 1.89 ₋₀₈ | 2.66 ₋₀₉ | 3.41 ₋₁₀ | 5.32 ₋₁₁ | 8.34 ₋₁₂ | 1.32 ₋₁₂ | 2.10 ₋₁₃ | 6.69 |
| 4096 | 1.32 ₋₀₈ | 1.88 ₋₀₉ | 2.31 ₋₁₀ | 3.62 ₋₁₁ | 5.71 ₋₁₂ | 9.11 ₋₁₃ | 1.28 ₋₁₃ | 6.85 |
| 8192 | 7.90 ₋₀₉ | 1.15 ₋₀₉ | 1.60 ₋₁₀ | 2.50 ₋₁₁ | 3.97 ₋₁₂ | 6.14 ₋₁₃ | 9.61 ₋₁₄ | 6.59 |
| 16384 | 4.11 ₋₀₉ | 6.00 ₋₁₀ | 9.06 ₋₁₁ | 1.43 ₋₁₁ | 2.27 ₋₁₂ | 3.58 ₋₁₃ | 5.65 ₋₁₄ | 6.46 |
| 32768 | 2.10 ₋₀₉ | 3.09 ₋₁₀ | 4.74 ₋₁₁ | 7.46 ₋₁₂ | 1.18 ₋₁₂ | 1.88 ₋₁₃ | 3.00 ₋₁₄ | 6.42 |
| 65536 | 1.08 ₋₀₉ | 1.58 ₋₁₀ | 2.43 ₋₁₁ | 3.82 ₋₁₂ | 6.03 ₋₁₃ | 9.63 ₋₁₄ | 1.55 ₋₁₄ | 6.41 |
| 131072 | 5.76 ₋₁₀ | 8.57 ₋₁₁ | 1.31 ₋₁₁ | 2.07 ₋₁₂ | 3.28 ₋₁₃ | 5.26 ₋₁₄ | 8.49 ₋₁₅ | 6.39 |

Tabelle 7.2: Approximationsfehler: $\|G - \tilde{G}\|_2$ f"ur $\ell = 2$ (SLP)

| n | $m = 4$ | $m = 5$ | $m = 6$ | $m = 7$ | $m = 8$ | $m = 9$ | $m = 10$ | Faktor |
|--------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|--------|
| 1024 | 5.55 ₋₁₀ | 4.36 ₋₁₁ | – | – | – | – | – | 12.73 |
| 2048 | 5.42 ₋₁₀ | 4.48 ₋₁₁ | 3.03 ₋₁₂ | 1.28 ₋₁₃ | 8.14 ₋₁₅ | 8.76 ₋₁₆ | 7.64 ₋₁₇ | 13.86 |
| 4096 | 4.54 ₋₁₀ | 3.57 ₋₁₁ | 2.61 ₋₁₂ | 1.04 ₋₁₃ | 5.41 ₋₁₅ | 7.60 ₋₁₆ | 7.18 ₋₁₇ | 13.60 |
| 8192 | 3.06 ₋₁₀ | 2.40 ₋₁₁ | 1.71 ₋₁₂ | 6.36 ₋₁₄ | 5.72 ₋₁₅ | 5.88 ₋₁₆ | 5.01 ₋₁₇ | 13.52 |
| 16384 | 1.81 ₋₁₀ | 1.38 ₋₁₁ | 9.66 ₋₁₃ | 5.13 ₋₁₄ | 4.91 ₋₁₅ | 4.88 ₋₁₆ | 4.12 ₋₁₇ | 12.80 |
| 32768 | 9.66 ₋₁₁ | 7.20 ₋₁₂ | 4.99 ₋₁₃ | 3.24 ₋₁₄ | 3.20 ₋₁₅ | 3.12 ₋₁₆ | 2.61 ₋₁₇ | 12.44 |
| 65536 | 5.10 ₋₁₁ | 3.76 ₋₁₂ | 2.60 ₋₁₃ | 1.66 ₋₁₄ | 1.75 ₋₁₅ | 1.71 ₋₁₆ | 1.42 ₋₁₇ | 12.38 |
| 131072 | 2.90 ₋₁₁ | 2.15 ₋₁₂ | 1.54 ₋₁₃ | 1.04 ₋₁₄ | 1.02 ₋₁₅ | 9.64 ₋₁₇ | 7.86 ₋₁₈ | 12.43 |

Tabelle 7.3: Approximationsfehler: $\|G - \tilde{G}\|_2$ f"ur $\ell = 3$ (SLP)

| n | $m = 4$ | $m = 5$ | $m = 6$ | $m = 7$ | $m = 8$ | $m = 9$ | $m = 10$ |
|--------|---------|---------|---------|---------|---------|---------|----------|
| 1024 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| 2048 | 0.3 | 0.4 | 0.5 | 0.5 | 0.5 | 0.5 | 0.6 |
| 4096 | 0.7 | 0.8 | 1.1 | 1.2 | 1.3 | 1.3 | 1.5 |
| 8192 | 1.6 | 2.0 | 2.4 | 2.6 | 2.8 | 3.1 | 3.3 |
| 16384 | 3.7 | 4.4 | 5.3 | 6.0 | 6.5 | 7.2 | 7.8 |
| 32768 | 8.2 | 9.9 | 11.7 | 13.1 | 14.5 | 16.0 | 17.5 |
| 65536 | 18.2 | 22.0 | 25.9 | 29.2 | 32.5 | 36.2 | 39.2 |
| 131072 | 39.6 | 47.8 | 56.3 | 63.7 | 70.9 | 78.3 | 85.8 |

Tabelle 7.4: Aufbau der \mathcal{H} -Matrix in Sekunden f"ur $\ell = 1$ (SLP)

7 Numerische Experimente für das Green-Quadratur-Verfahren

Die Laufzeiten, die zum Aufbau der \mathcal{H} -Matrix-Struktur in diesen drei Fällen nötig sind, können in Tabelle 7.4 (für $\ell = 1$), Tabelle 7.5 (für $\ell = 2$) und Tabelle 7.6 (für $\ell = 3$) nachgelesen werden. Man kann daran ein Verhalten $\sim \mathcal{O}(nk \log n)$ erkennen, wie die Theorie der \mathcal{H} -Matrizen vorhersagt.

| n | $m = 4$ | $m = 5$ | $m = 6$ | $m = 7$ | $m = 8$ | $m = 9$ | $m = 10$ |
|--------|---------|---------|---------|---------|---------|---------|----------|
| 1024 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 |
| 2048 | 0.5 | 0.6 | 0.7 | 0.7 | 0.8 | 0.8 | 0.8 |
| 4096 | 1.2 | 1.5 | 1.8 | 1.9 | 2.0 | 2.1 | 2.3 |
| 8192 | 2.8 | 3.3 | 4.2 | 4.6 | 4.9 | 5.4 | 5.9 |
| 16384 | 6.5 | 7.7 | 9.4 | 10.4 | 11.3 | 12.4 | 13.6 |
| 32768 | 14.4 | 17.3 | 20.8 | 23.3 | 25.6 | 28.2 | 30.9 |
| 65536 | 32.1 | 38.8 | 46.2 | 52.0 | 57.5 | 63.5 | 69.4 |
| 131072 | 69.8 | 84.9 | 100.7 | 114.0 | 126.7 | 139.9 | 153.6 |

Tabelle 7.5: Aufbau der \mathcal{H} -Matrix in Sekunden für $\ell = 2$ (SLP)

| n | $m = 4$ | $m = 5$ | $m = 6$ | $m = 7$ | $m = 8$ | $m = 9$ | $m = 10$ |
|--------|---------|---------|---------|---------|---------|---------|----------|
| 1024 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 |
| 2048 | 0.7 | 0.7 | 0.8 | 0.8 | 1.0 | 1.0 | 1.1 |
| 4096 | 1.9 | 2.1 | 2.2 | 2.4 | 2.8 | 2.9 | 3.0 |
| 8192 | 4.3 | 4.8 | 5.4 | 6.5 | 7.6 | 7.9 | 8.2 |
| 16384 | 9.5 | 11.0 | 12.5 | 15.2 | 16.9 | 18.0 | 19.0 |
| 32768 | 21.1 | 24.8 | 28.5 | 33.2 | 37.6 | 40.7 | 43.4 |
| 65536 | 47.2 | 55.9 | 64.5 | 74.3 | 84.1 | 91.9 | 99.0 |
| 131072 | 103.2 | 122.9 | 142.5 | 164.0 | 184.4 | 205.4 | 218.4 |

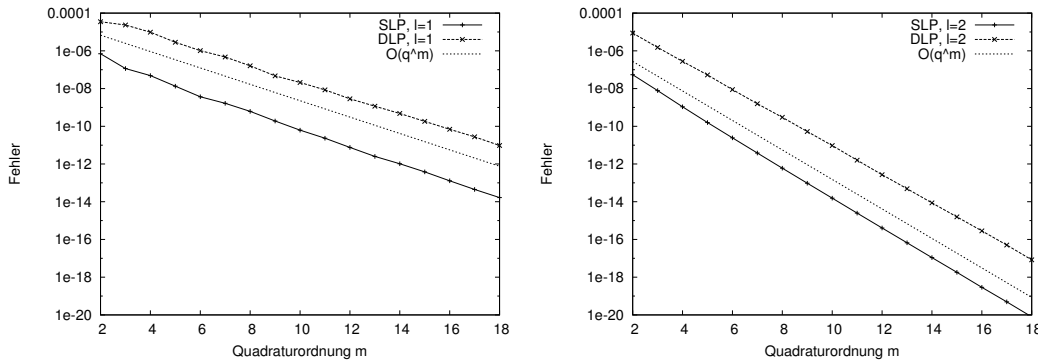
Tabelle 7.6: Aufbau der \mathcal{H} -Matrix in Sekunden für $\ell = 3$ (SLP)

In Abschnitt 5.3.3 wurde argumentiert, dass die Green-Quadratur-Methode nicht nur auf das Einzelschichtpotential (SLP) angewendet werden kann, sondern ebenso für das Doppelschichtpotential (DLP). Dazu kann dieselbe Rechteck- oder Kreis-Parametrisierung verwendet werden, wie zuvor für SLP.

Die Tabelle 7.7 zeigt die Fehlerentwicklung für DLP mit $\ell = 1$. Es ist zu beobachten, dass der Fehler etwas höher ist als im SLP-Fall unter denselben Umständen, sich aber ansonsten sehr ähnlich verhält.

Einen direkten Vergleich zwischen SLP- und DLP-Fehlerwerten für relevante Quadraturordnungen m und erneut für $n = 65\,536$ liefert die Abbildung 7.2. Dabei zeigt die linke Teilabbildung den Vergleich zwischen der Methode für SLP und DLP bei $\ell = 1$ und die rechte Teilabbildung bei $\ell = 2$ Teilintervallen in der Quadratur. Um noch deutlicher zu machen, dass sich in beiden Teilabbildungen der Fehler wie $\mathcal{O}(q^m)$ verhält (wie bereits anhand der Tabellen gesehen), ist zur Orientierung noch eine weitere $\mathcal{O}(q^m)$ -Kurve eingezeichnet. Natürlich ist zusätzlich wieder zu beobachten, dass die Kurven bei $\ell = 2$ deutlich steiler verlaufen als bei $\ell = 1$.

| n | $m = 4$ | $m = 5$ | $m = 6$ | $m = 7$ | $m = 8$ | $m = 9$ | $m = 10$ | Faktor |
|--------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------|
| 1024 | 3.56_{-05} | 1.27_{-05} | 2.27_{-06} | 1.03_{-06} | 4.22_{-07} | 1.53_{-07} | 6.27_{-08} | 2.88 |
| 2048 | 6.68_{-05} | 2.08_{-05} | 2.90_{-06} | 1.33_{-06} | 5.47_{-07} | 2.17_{-07} | 8.40_{-08} | 3.04 |
| 4096 | 5.71_{-05} | 1.73_{-05} | 4.16_{-06} | 2.00_{-06} | 7.33_{-07} | 2.31_{-07} | 9.44_{-08} | 2.91 |
| 8192 | 3.85_{-05} | 1.14_{-05} | 3.85_{-06} | 1.83_{-06} | 6.56_{-07} | 2.09_{-07} | 8.84_{-08} | 2.75 |
| 16384 | 2.52_{-05} | 7.67_{-06} | 2.45_{-06} | 1.15_{-06} | 4.05_{-07} | 1.18_{-07} | 5.12_{-08} | 2.81 |
| 32768 | 1.58_{-05} | 4.64_{-06} | 1.57_{-06} | 7.40_{-07} | 2.53_{-07} | 7.14_{-08} | 3.27_{-08} | 2.80 |
| 65536 | 9.76_{-06} | 2.83_{-06} | 1.03_{-06} | 4.71_{-07} | 1.61_{-07} | 4.69_{-08} | 2.10_{-08} | 2.78 |
| 131072 | 6.04_{-06} | 1.70_{-06} | 6.36_{-07} | 2.90_{-07} | 9.88_{-08} | 2.78_{-08} | 1.24_{-08} | 2.81 |

Tabelle 7.7: Approximationsfehler: $\|G - \tilde{G}\|_2$ für $\ell = 1$ (DLP)Abbildung 7.2: Kreis-Geometrie, $n = 65\,536$: Green-Quadratur-Methode mit Rechteck-Parametrisierung und $\delta = 0.5 \text{ diam}$, maximale Blattgröße $clf = 2k$, $l = 1$ (links) und $l = 2$ (rechts)

Die Zeit, die nötig ist, eine \mathcal{H} -Matrix für DLP mit dem Verfahren aufzubauen für $\ell = 1$, ist in Tabelle 7.8 festgehalten. Die Zeiten sind deutlich besser, weil die aufwändige Berechnung des Logarithmus im Gegensatz zur Berechnung der SLP-Matrixapproximation entfällt.

Als letztes Experiment vor dem Vergleich der Green-Quadratur-Methode mit anderen Verfahren im nächsten Abschnitt 7.1.2 wird noch der zeitliche Aufwand für größer werdende Probleme betrachtet. Dazu sei eine Quadraturordnung m festgehalten. Es wurde hier eine mittlere Ordnung von $m = 7$ gewählt und Freiheitsgrade von $n = 256$ bis hin zu $n = 131\,072$ getestet.

Abbildung 7.3 zeigt die Zeit in Sekunden, die das Verfahren pro individuellem Freiheitsgrad benötigt bei einer wachsenden Problemgröße. Die Teilabbildung links zeigt die Kurven für das Einzelschichtpotential mit $\ell = 1$ und $\ell = 2$ und die rechte Teilabbildung zeigt die Kurven für das Doppelschichtpotential, ebenfalls für $\ell = 1$ und $\ell = 2$. Es ist sofort zu sehen, dass eine höhere Anzahl von Teilintervallen ℓ auch mehr Rechenzeit in Anspruch nimmt und dass die Kurven für SLP stärker steigen, als für DLP. Letzteres Verhalten ist

7 Numerische Experimente für das Green-Quadratur-Verfahren

| n | $m = 4$ | $m = 5$ | $m = 6$ | $m = 7$ | $m = 8$ | $m = 9$ | $m = 10$ |
|--------|---------|---------|---------|---------|---------|---------|----------|
| 1024 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| 2048 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 |
| 4096 | 0.4 | 0.5 | 0.5 | 0.6 | 0.6 | 0.7 | 0.7 |
| 8192 | 1.0 | 1.1 | 1.3 | 1.5 | 1.6 | 1.7 | 1.9 |
| 16384 | 2.3 | 2.7 | 3.0 | 3.4 | 3.8 | 4.2 | 4.5 |
| 32768 | 5.1 | 6.0 | 6.9 | 7.8 | 8.7 | 9.6 | 10.4 |
| 65536 | 11.4 | 13.6 | 15.6 | 17.8 | 19.9 | 21.9 | 23.9 |
| 131072 | 25.3 | 30.2 | 34.7 | 39.5 | 44.2 | 48.7 | 53.0 |

Tabelle 7.8: Aufbau der \mathcal{H} -Matrix in Sekunden für $\ell = 1$ (DLP)

bei der zweidimensionalen Kernfunktion in dem zu berechnenden logarithmischen Term beim Einfachschichtpotential zu erklären.

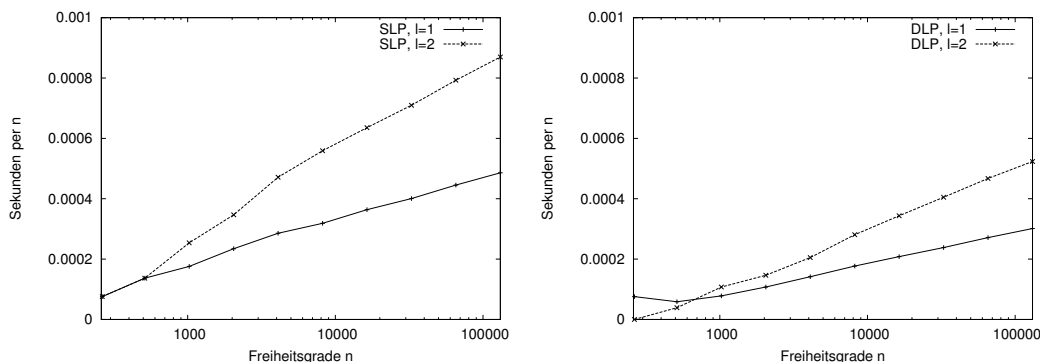


Abbildung 7.3: Kreis-Geometrie: Green-Quadratur-Methode mit Rechteck-Parametrisierung und $\delta = 0.5 \text{ diam}$, maximale Blattgröße $clf = 2k$, Zeit pro Freiheitsgrad für ein festes $m = 7$ für SLP (links) und DLP (rechts)

Viel wichtiger jedoch ist, dass man durch die logarithmische x -Achse sehr deutlich das logarithmische Wachstum der Kurven in beiden Teilabbildungen erkennen kann; in etwa $\sim \log n$, wie die Theorie vorhersagt.

Alle diese Experimente bestätigen die theoretischen Analysen aus Kapitel 6 und zeigen die exponentielle Konvergenz des Verfahrens. Es kann somit jede gewünschte Genauigkeit erreicht werden.

7.1.2 Vergleiche mit anderen analytischen Verfahren

Nachdem nun geklärt ist, dass das Green-Quadratur-Verfahren sich in der Praxis so verhält, wie die Theorie angekündigt hat, bleibt noch der Vergleich mit anderen Methoden. Da es sich bei dem Green-Quadratur-Verfahren um ein analytisches Verfahren

handelt, wird es in diesem Abschnitt auch nur mit anderen analytischen Verfahren verglichen, genauer, mit der in Abschnitt 4.1 vorgestellten Interpolation und dem Multipol-Verfahren aus Abschnitt 4.2.

Um zu sehen, wie sich der zuvor in Bemerkung 5.3 gemachte Komplexitätsvergleich zwischen dem Green-Quadratur-Verfahren und der Interpolation in der Praxis auswirkt, soll sich zunächst auf die Interpolation als Vergleichsverfahren beschränkt werden.

Zur Erinnerung, wie sich die lokalen Ränge berechnen, hier nochmal eine kurze Zusammenfassung: Für den Interpolations-Ansatz, der auf der Tensorprodukt-Interpolation beruht, erhält man bei m Interpolationspunkten einen lokalen Rang von $k = m^2$.

Bei der Green-Quadratur-Methode mit m Quadraturpunkten und ℓ Teilintervallen kommt man mit der vierteiligen Rechteck-Parametrisierung auf einen lokalen Rang von $k = 8m\ell$ und mit der Kreis-Parametrisierung auf $k = 2m\ell$.

Auch für diese ersten Experimente wurde für Ω die Kreis-Geometrie mit $n = 65\,536$ Freiheitsgraden gewählt.

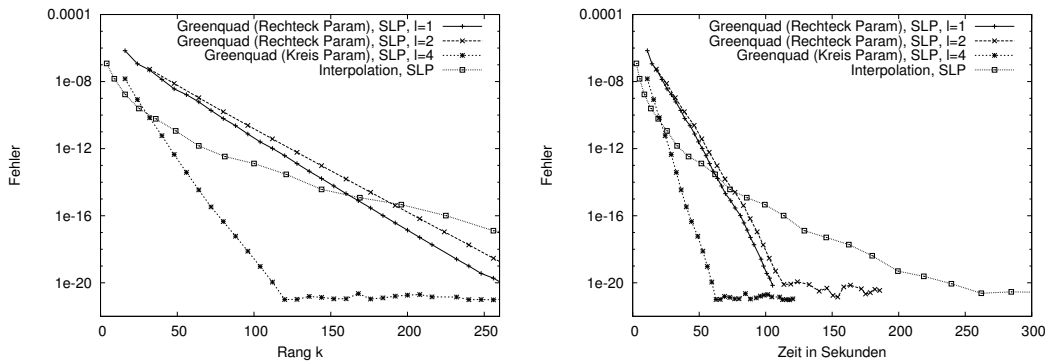


Abbildung 7.4: Kreis-Geometrie, $n = 65\,536$: maximale Blattgröße $clf = 2k$, SLP, Green-Quadratur-Methode mit Rechteck-Parametrisierung und $\delta = 0.5 \text{ diam}$ und Green-Quadratur-Methode mit Kreis-Parametrisierung und $\delta = 0.4 \text{ diam}$ im Vergleich zu Interpolation in Bezug auf die lokalen Ränge

In Abbildung 7.4 wird die Green-Quadratur-Methode im Falle des Einfachschichtpotentials in drei Varianten mit der Interpolation verglichen. Neben der Kurve für die Interpolation sind die Kurven für die Rechteck-Parametrisierung mit $\ell = 1$ und $\ell = 2$ zu sehen, sowie die für die Kreis-Parametrisierung mit $\ell = 4$ (welche dem Vergleichswert gegenüber der Rechteck-Parametrisierung mit $\ell = 1$ entspricht).

Es ist zu erwarten, dass der Fehler für die Green-Quadratur-Methode mit $(\frac{c_0}{\ell})^n = (\frac{c_0}{\ell})^{2m-1}$ konvergiert, während er für die Interpolation mit $(c_0\eta)^{m+1}$ konvergiert nach [9]. Damit spielt $\frac{1}{\ell}$ hier die selbe Rolle, wie der Zulässigkeits-Parameter η in Bezug auf die Konvergenzgeschwindigkeit.

In der linken Teilabbildung von Abbildung 7.4 sind die Ränge zu sehen, die für einen

7 Numerische Experimente für das Green-Quadratur-Verfahren

bestimmten Fehler benötigt werden und in der rechten Teilabbildung entsprechend die Zeit in Sekunden.

Es ist zu erkennen, dass das Green-Quadratur-Verfahren für höhere Genauigkeiten nicht nur kleinere Ränge benötigt, sondern auch schneller als die Interpolation ist. Sehr deutlich ist es in diesem Experiment bei dem Green-Quadratur-Verfahren mit der Kreis-Parametrisierung, die hier besonders gut abschneidet.

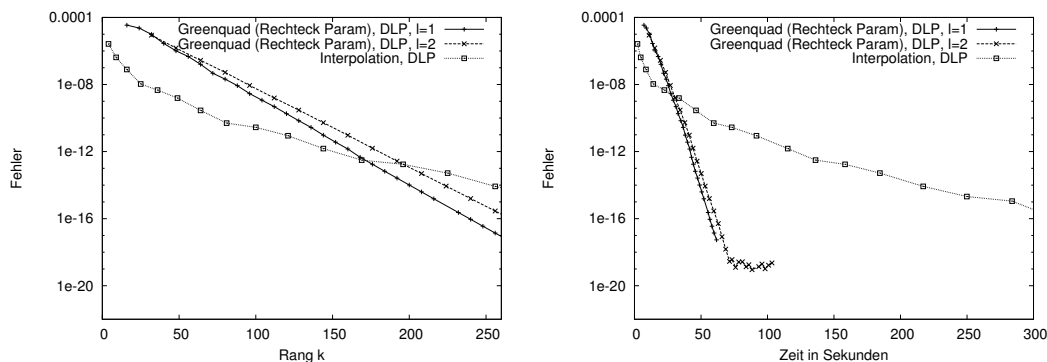


Abbildung 7.5: Kreis-Geometrie, $n = 65\,536$: maximale Blattgröße $clf = 2k$, DLP, Green-Quadratur-Methode mit Rechteck-Parametrisierung und $\delta = 0.5 \text{ diam}$ im Vergleich zu Interpolation in Bezug auf die lokalen Ränge

In Abbildung 7.5 sind die Ergebnisse desselben Experiments für das Doppelschichtpotential dargestellt, nur ohne Kurve für die Kreis-Parametrisierung. Der Vorteil der Green-Quadratur-Methode ist hier auch noch zu erkennen, besonders anhand des viel geringeren zeitlichen Aufwands.

Diese Experimente zeigen, dass bei größeren Rängen und höheren Genauigkeiten, das Green-Quadratur-Verfahren der Interpolation vorzuziehen ist.

In einem letzten Experiment zum zweidimensionalen Fall wird eine etwas kompliziertere Geometrie zu Grunde gelegt, um unrealistische Effekte zu vermeiden, die im Vergleich der Methoden durch die zu simple Kreis-Geometrie unter Umständen auftreten könnten.

Daher ist das Gebiet Ω hier die Hilbert-Kurve der Ordnung 8, also mit $n = 2^{16} - 1 = 65\,535$ Freiheitsgraden. Zur Veranschaulichung sind in Abbildung 7.6 die ersten sechs Ordnungen von Hilbertkurven dargestellt.

Wie zuvor sollen die Green-Quadratur-Methode, einmal mit Rechteck-Parametrisierung und einmal mit Kreis-Parametrisierung, und die Interpolation verglichen werden, aber diesmal zusätzlich mit einem Vergleich zu der anderen in Abschnitt 4.2 vorgestellten analytischen Methode, dem Multipol-Verfahren.

Der Vergleich ist in Abbildung 7.7 dargestellt und geht in diesem Fall über die Größe der entstandenen \mathcal{H} -Matrix im Speicher gegenüber dem relativen Fehler in der ersten

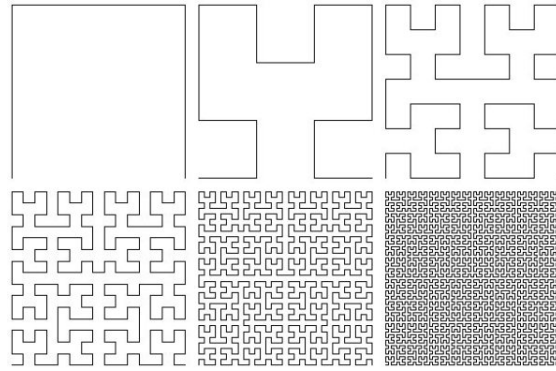


Abbildung 7.6: Hilbertkurven der Ordnungen 1 bis 6

Teilabbildung, da ein Vergleich der lokalen Ränge wie zuvor im Kontext der Multipol-Methode nicht zu sinnvollen Vergleichen führen würde. In der zweiten Teilabbildung ist die benötigte Zeit zum Aufstellen der \mathcal{H} -Matrix in Sekunden für den jeweiligen relativen Fehler zu sehen.

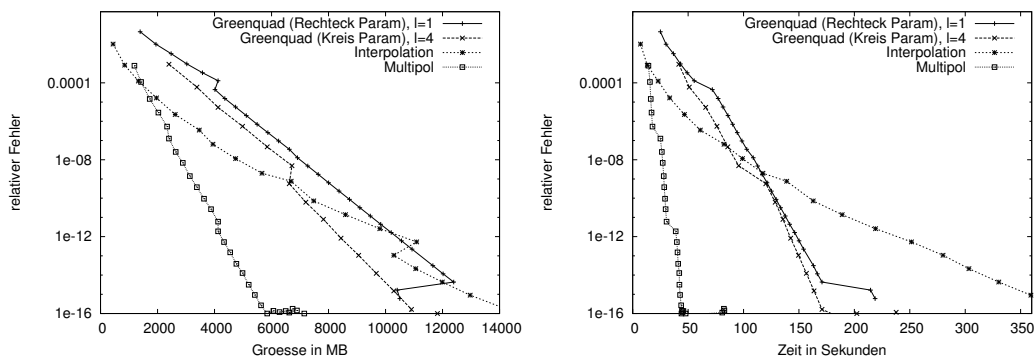


Abbildung 7.7: Hilbertkurven-Geometrie, $n = 65\,535$: maximale Blattgröße $clf = 2k$, SLP, Green-Quadratur-Methode mit Rechteck-Parametrisierung und $\delta = 0.5 \text{ diam}$ und Green-Quadratur-Methode mit Kreis-Parametrisierung und $\delta = 0.4 \text{ diam}$ im Vergleich zu Interpolation und dem Multipol-Verfahren in Bezug auf die Größe der Matrix im Speicher

Die Ergebnisse zeigen auch auf der Hilbertkurve einen leichten Vorteil des Green-Quadratur-Verfahrens bei höheren Genauigkeiten gegenüber der Interpolation, besonders in Bezug auf die Laufzeit. Das Multipol-Verfahren allerdings produziert speichergünstigere \mathcal{H} -Matrizen und dazu noch in sehr viel geringerer Zeit und steht damit für das analytische Green-Quadratur-Verfahren noch außer Konkurrenz.

Diese Tatsache und auch die Ergebnisse, die im folgenden Abschnitt 7.2 für den drei-

dimensionalen Fall vorgestellt werden, haben es zur Notwendigkeit gemacht, das Green-Quadratur-Verfahren zu verbessern und zu ergänzen, um mit dem daraus resultierenden Green-Hybrid-Verfahren ein konkurrenzfähiges, aber noch immer beweisbares Verfahren zu erhalten.

7.2 3D-Fall

Im dreidimensionalen Fall entspricht die Effizienz des Green-Quadratur-Verfahrens nicht den Erwartungen. Wie bereits in Abschnitt 5.3.1 beschrieben, funktioniert das Verfahren ebenso zuverlässig auch für den dreidimensionalen Fall, liefert allerdings bei der \mathcal{H} -Matrixapproximation zu hohe lokale Ränge von $k = 12m^2\ell^2$ bei m Quadraturpunkten und ℓ Teilintervallen mit der Quader-Parametrisierung.

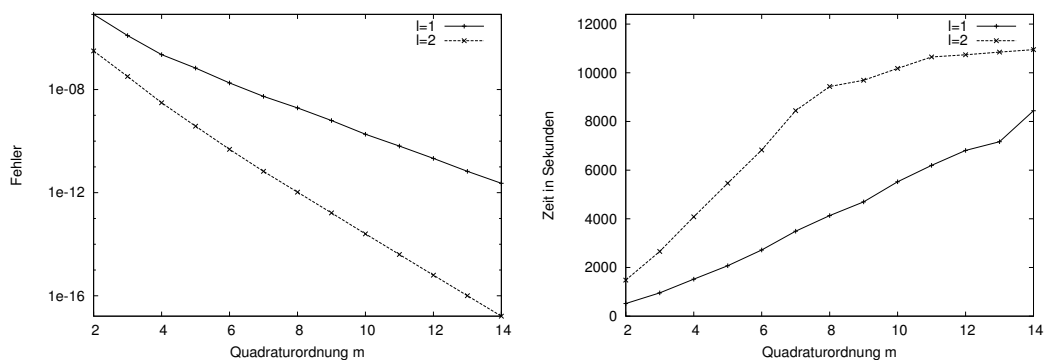


Abbildung 7.8: Kugel-Geometrie, $n = 156\,800$: maximale Blattgröße $clf = 2k$, SLP, Green-Quadratur-Methode mit Quader-Parametrisierung und $\delta = 0.5 \text{ diam}$, Fehler für verschieden Quadraturordnungen m (links) und zugehörige Zeiten (rechts)

Für die numerischen Experimente in diesem Abschnitt ist Ω die Approximation einer Kugeloberfläche mit $n = 156\,800$. Abbildung 7.8 zeigt in der linken Teilabbildung den abnehmenden Fehler für wachsende Quadraturordnungen m bei $\ell = 1$ und $\ell = 2$ und in der rechten Teilabbildung die entsprechenden Zeiten in Sekunden.

Die Kurven zeigen auch hier deutlich die gewünschte Konvergenz, aber obwohl ebenso wie im zweidimensionalen Fall jede Genauigkeit erreicht werden kann, benötigt das Verfahren nicht nur erheblich mehr Speicher, sondern auch wesentlich längere Laufzeiten.

An einem Vergleich mit der Interpolation über die lokalen Ränge und die entsprechenden Zeiten, wie in Abbildung 7.9 zu sehen ist, ist dieses ebenso zu beobachten. Es gilt zwar auch hier, dass das Green-Quadratur-Verfahren für hohe Genauigkeiten mit $k = m^2\ell^2$ gegenüber der Interpolation mit $k = m^3$ geringere lokale Ränge produziert, aber der zeitliche Aufwand dafür liegt deutlich über dem der Interpolation.

Also ist das Green-Quadratur-Verfahren in der bisherigen Form im dreidimensionalen

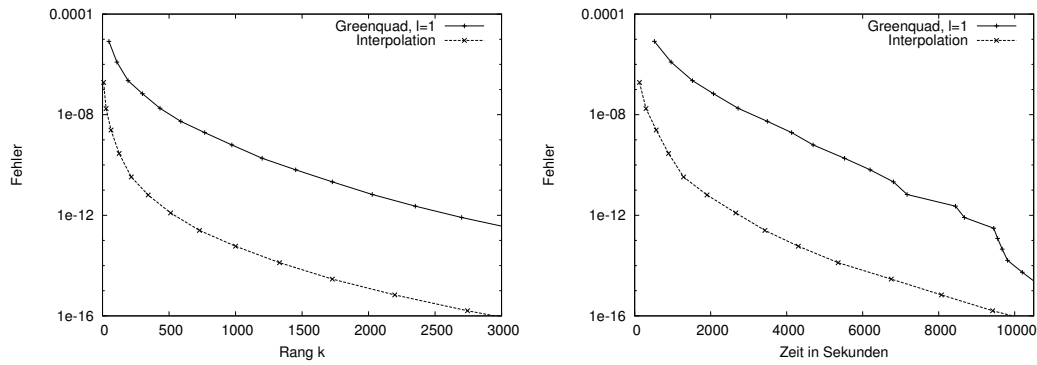


Abbildung 7.9: Kugel-Geometrie, $n = 156\,800$: maximale Blattgröße $clf = 2k$, SLP, Green-Quadratur-Methode mit Quader-Parametrisierung und $\delta = 0.5 \text{ diam}$ im Vergleich mit Interpolation

Fall nicht konkurrenzfähig.

Das in den nächsten Kapiteln behandelte Green-Hybrid-Verfahren allerdings gleicht das Problem mit den hohen Rängen ebenso aus, wie es die benötigte Geschwindigkeit liefert, um nicht nur mit den analytischen Verfahren wie der Interpolation und der Multipol-Methode konkurrieren zu können, sondern auch mit der algebraischen Kreuzapproximation.

8 Das effiziente Green-Hybrid-Verfahren

Die vorangegangenen Kapitel haben gezeigt, dass das Green-Quadratur-Verfahren eine robuste und vor allem beweisbare Methode zur Konstruktion von Niedrigrang-Approximationen ist. Allerdings ist auch klar geworden, dass insbesondere im praxisrelevanten dreidimensionalen Fall bereits etablierte algebraische Verfahren wie die Kreuzapproximation zu besseren numerischen Resultaten führen.

Da sich die Grundidee, mit der Green'schen Formel eine Raumdimension bei der Integration einzusparen, bewährt hat, ist nun die Frage, wie darauf ein effizienter Algorithmus aufgebaut werden kann.

Das größte Problem, das die Green-Quadratur-Methode mitbringt, ist der hohe lokale Rang, der produziert wird. Die Idee ist nun, auf die mit der Green-Quadratur-Methode konstruierten Niedrigrangmatrizen die adaptive Kreuzapproximation anzuwenden, um den Rang noch weiter zu reduzieren.

Mit einer zusätzlichen Modifikation, die auf dem Interpolations-Ansatz beruht, wird der daraus entstandene Algorithmus dann noch effizienter. Die Beschreibung dieser Kombination der Verfahren zu einem schnellen, robusten und dabei weiterhin beweisbaren Algorithmus ist im folgenden Abschnitt 8.1 zu finden.

Dieser effiziente Green-Hybrid-Algorithmus ist zudem noch so konstruiert, dass er sehr gut für \mathcal{H}^2 -Matrizen erweiterbar ist. Damit ist nicht nur eine effiziente Version der Green-Quadratur-Methode entstanden, sondern das eigentliche Ziel, ebenso \mathcal{H}^2 -Matrizen konstruieren zu können, ist mit dem Green-Hybrid-Verfahren erreicht.

Die Anwendung des Algorithmus auf \mathcal{H}^2 -Matrizen wird in Abschnitt 8.2 erklärt.

8.1 Kombination der Green-Quadratur mit ACA und Interpolation

Die Bezeichnung als hybrides Verfahren verdient das Green-Hybrid-Verfahren dadurch, dass analytische Methoden, wie die Green-Quadratur und die Interpolation, mit der algebraischen Kreuzapproximation kombiniert werden.

Wie zuvor bei der reinen Green-Quadratur-Methode, wird bei diesem Verfahren mit Hilfe der Green-Quadratur durch eine Niedrigrangmatrix approximiert. Dann wird auf diese die adaptive Kreuzapproximation (ACA) wie in Abschnitt 4.3 angewendet, damit der Rang noch weiter reduziert werden kann. Um dabei die Beweisbarkeit nicht einzubüßen, muss hier auf ACA mit vollständiger Pivotsuche zurückgegriffen werden. Dieses Vorgehen wird in Abschnitt 8.1.1 beschrieben.

Die Effizienzsteigerung durch einen Interpolations-Ansatz, der gleichzeitig die spätere Erweiterung auf \mathcal{H}^2 -Matrizen ermöglicht, wird dann in Abschnitt 8.1.2 vorgestellt.

8.1.1 Rangreduzierung im Green-Quadratur-Verfahren mit ACA

Es wird erneut das Modellproblem aus Kapitel 2.1 auf $\Omega \subseteq \mathbb{R}^d$ mit $d \in \{2, 3\}$ betrachtet. Sei $G|_{\hat{t} \times \hat{s}}$ ein zulässiger Matrixblock, also mit $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+$, mit den Einträgen aus (2.4), der durch eine Niedrigrangmatrix $\tilde{G}|_{\hat{t} \times \hat{s}}$ approximiert werden soll.

Dazu wird als erstes die Green-Quadratur-Methode wie in Kapitel 5 beschrieben angewendet. Dadurch erhält man wie in den Formeln (5.11) und (5.14) die Rang- q -Matrix $\tilde{G}|_{\hat{t} \times \hat{s}}$ mit

$$G|_{\hat{t} \times \hat{s}} \approx \tilde{G}|_{\hat{t} \times \hat{s}} := A_t B_{t,s}^*, \quad (8.1)$$

wobei

$$q := \begin{cases} 8m\ell & \text{für } d = 2 \\ 12m^2\ell^2 & \text{für } d = 3 \end{cases} \quad (8.2)$$

für den Rang gilt. Dabei ist die Matrix $A_t \in \mathbb{R}^{\hat{t} \times q}$ wie in (5.10) und (5.13) durch

$$A_t := [\check{A}_1 \dots \check{A}_{2d} \hat{A}_1 \dots \hat{A}_{2d}] \quad (8.3)$$

mit

$$\begin{aligned} (\check{A}_\iota)_{i\nu} &= G_{\det}(t) w_\nu \int_{\Gamma} g(x, \xi_\iota(t_\nu)) \varphi_i(x) \, d\Gamma(x) \\ (\hat{A}_\iota)_{i\nu} &= G_{\det}(t) w_\nu \int_{\Gamma} \frac{\partial g}{\partial n_z}(x, \xi_\iota(t_\nu)) \varphi_i(x) \, d\Gamma(x) \end{aligned} \quad (8.4)$$

für $i \in \hat{t}$, $\nu \in \{1, \dots, m^{d-1}\ell^{d-1}\}$ und $\iota = 1, \dots, 2d$ definiert. Dabei ist mit

$$G_{\det}(t) := \begin{cases} \left\| \frac{\partial}{\partial t} \xi_\iota(t) \right\| & \text{für } d = 2 \\ \left\| \frac{\partial}{\partial t_1} \xi_\iota(t) \times \frac{\partial}{\partial t_2} \xi_\iota(t) \right\| & \text{für } d = 3 \end{cases} \quad (8.5)$$

die Gramsche Determinante bezeichnet.

Da die Parametrisierung durch die ξ_ι mit $\iota = 1, \dots, 2d$ nur über die Daten der Bounding Box \mathcal{B}_τ konstruiert wird und $x \in \mathcal{B}_\tau$ ist, hängt damit die Matrix A_t nur von dem Cluster t ab.

Die Matrix $B_{t,s} \in \mathbb{R}^{\hat{s} \times q}$ allerdings hängt von beiden Clustern t und s ab. Sie ist definiert durch

$$B_{t,s} := [\check{B}_1 \dots \check{B}_{2d} -\hat{B}_1 \dots -\hat{B}_{2d}] \quad (8.6)$$

mit

$$\begin{aligned} (\check{B}_\iota)_{j\nu} &= \int_{\Gamma} \frac{\partial g}{\partial n_z}(\xi_\iota(t_\nu), y) \varphi_j(y) \, d\Gamma(y) \\ (\hat{B}_\iota)_{j\nu} &= \int_{\Gamma} g(\xi_\iota(t_\nu), y) \varphi_j(y) \, d\Gamma(y) \end{aligned} \quad (8.7)$$

für $j \in \hat{s}$, $\nu \in \{1, \dots, m^{d-1}\ell^{d-1}\}$ und $\iota = 1, \dots, 2d$. Dabei hängen die ξ_ι von der Bounding Box \mathcal{B}_τ und damit vom Cluster t ab und durch $y \in \mathcal{B}_\sigma$ ist die Abhängigkeit vom Cluster s erklärt.

8.1 Kombination der Green-Quadratur mit ACA und Interpolation

Diese Abhängigkeiten werden später in der Effizienzsteigerung und für die Anwendung auf \mathcal{H}^2 -Matrizen eine wichtige Rolle spielen.

Der nächste Schritt ist nun, auf die Matrix (8.1) die adaptive Kreuzapproximation anzuwenden, um den Rang q weiter zu reduzieren. Dazu reicht es aus, nur auf die Matrix A_t die Kreuzapproximation anzuwenden.

Wie bereits erwähnt, muss hier auf ACA mit vollständiger Pivotsuche (siehe dazu Abschnitt 4.3.2) zurückgegriffen werden, um wieder ein robustes und beweisbares Verfahren zu erhalten. Da die Matrix A_t vollständig im Speicher vorliegt und mit ihrem Rang von q schon relativ klein ist, bedeutet dies allerdings keinen allzu großen Mehraufwand.

Die für ACA festgelegte Fehlerschranke sei hier mit ε_{aca} bezeichnet, so dass das Ergebnis der Approximation durch ACA mit voller Pivotierung gegeben ist durch

$$A_t \approx C_t D_t^* \quad \text{mit} \quad \|A_t - C_t D_t^*\|_2 \leq \varepsilon_{aca} \quad (8.8)$$

mit den Matrizen $C_t \in \mathbb{R}^{\hat{t} \times k}$ und $D_t \in \mathbb{R}^{q \times k}$, die einen kleineren Rang $k \leq q$ haben.

Eingesetzt in (8.1) folgt dann

$$\tilde{G}|_{\hat{t} \times \hat{s}} := A_t B_{t,s}^* \approx C_t D_t^* B_{t,s}^* =: \bar{A}_t \bar{B}_{t,s}^* \quad (8.9)$$

mit $\bar{A}_t := C_t$ und $\bar{B}_{t,s}^* := B_{t,s}^* D_t$. Damit wurde also die vom Green-Quadratur-Verfahren berechnete Rang- q -Matrix $A_t B_{t,s}^*$ weiter komprimiert durch ACA zu einer rangreduzierten Rang- k -Matrix $\bar{A}_t \bar{B}_{t,s}^*$.

Auf diese Weise werden die relativ hohen lokalen Ränge, die das reine Green-Quadratur-Verfahren liefert, vermieden, ohne viel an Genauigkeit einzubüßen und ohne viel zusätzlichen Rechenaufwand zu benötigen, da ACA nur auf kleine Niedrigrangmatrizen A_t angewendet wird.

Damit ist das größte Problem des Green-Quadratur-Verfahrens behoben.

Aber es geht noch besser: Mit der im nächsten Abschnitt 8.1.2 beschriebenen Verbesserung kann nicht nur die Genauigkeit noch erhöht und gleichzeitig Speicher- und Rechenaufwand reduziert werden, sondern auch der Weg bereitet werden, um den Algorithmus auf \mathcal{H}^2 -Matrizen erweitern zu können.

8.1.2 Effizienz durch eine interpolierende Matrix

Zur Konstruktion der Rang- k -Matrix-Approximation aus (8.8) durch ACA werden in jedem der k Schritte Zeilen-Pivotelemente μ_k des betragsgrößten Eintrags ausgewählt. Mit den sich daraus ergebenden Zeilen-Pivot-Indizes $\mu_1, \dots, \mu_k \in \hat{t}$ wird eine lineare Abbildung $R_t \in \mathbb{R}^{k \times \hat{t}}$ definiert durch

$$R_t x = \begin{pmatrix} x_{\mu_1} \\ \vdots \\ x_{\mu_k} \end{pmatrix} \in \mathbb{R}^k$$

8 Das effiziente Green-Hybrid-Verfahren

für alle $x \in \mathbb{R}^{\hat{t}}$. Diese Abbildung angewendet auf die Matrix $C_t \in \mathbb{R}^{\hat{t} \times k}$ aus (8.8) ergibt die Matrix

$$R_t C_t \in \mathbb{R}^{k \times k},$$

die nach Definition von R_t über die Zeilen-Pivot-Indizes von ACA eine untere Dreiecksmatrix ist. (Der erste Zeilen-Pivot-Index μ_1 entspricht dem ersten ACA-Schritt und liefert den einen Eintrag in der ersten Zeile, μ_2 entspricht dem zweiten ACA-Schritt und liefert zwei Einträge in der zweiten Zeile, usw.)

Im Algorithmus geht man nun so vor, dass nach der Berechnung von (8.8) durch ACA die Matrix C_t und die Zeilen-Pivot-Indizes $\mu_1, \dots, \mu_k \in \hat{t}$ gespeichert werden. Anstatt sich ebenso die Matrix $D_t \in \mathbb{R}^{q \times k}$ zu merken, wird aber nur die kleinere Dreiecksmatrix $R_t C_t \in \mathbb{R}^{k \times k}$ abgespeichert und D_t verworfen.

Die Matrix $\bar{B}_{t,s}$ kann so zwar nicht mehr durch D_t und $B_{t,s}$ berechnet werden, aber durch die Dreiecksgestalt von $R_t C_t$ kann $\bar{B}_{t,s}$ durch Vorwärtseinsetzen berechnet werden: Im Grunde wird dabei eine Matrix $\tilde{B}_{t,s}$ gesucht mit

$$G|_{\hat{t} \times \hat{s}} \approx C_t \tilde{B}_{t,s}^* \quad (8.10)$$

wie in (8.9). Wenn dabei nur die Zeilen der Pivot-Indizes $\mu_1, \dots, \mu_k \in \hat{t}$, die R_t auswählt, betrachtet werden, also

$$R_t G|_{\hat{t} \times \hat{s}} \approx R_t C_t \tilde{B}_{t,s}^*, \quad (8.11)$$

ist dies nicht nur ein durch die Dreiecksgestalt von $R_t C_t$ durch Vorwärtseinsetzen leicht zu lösendes Problem, sondern es spart gegenüber (8.10) auch Speicher- und Rechenaufwand. Die Aufwandsersparnis liegt daran, dass k klein genug ist und somit nur wenige Einträge von $G|_{\hat{t} \times \hat{s}}$ direkt berechnet werden müssen. Diese Idee geht nicht zu Lasten der Genauigkeit, da hier die relevanten, von ACA gewählten, Zeilen betrachtet werden.

Formal lässt sich dieser Ansatz über einen Interpolationsoperator \mathcal{I}_t beschreiben, der C_t exakt interpoliert: Mit der Definition

$$\mathcal{I}_t := C_t (R_t C_t)^{-1} R_t \quad (8.12)$$

gilt offensichtlich $\mathcal{I}_t C_t = C_t$. Angewendet auf $G|_{\hat{t} \times \hat{s}}$ ergibt sich

$$G|_{\hat{t} \times \hat{s}} \approx \mathcal{I}_t G|_{\hat{t} \times \hat{s}} = C_t (R_t C_t)^{-1} R_t G|_{\hat{t} \times \hat{s}}.$$

Nach (8.10) wäre damit $\tilde{B}_{t,s}$ gefunden mit

$$\tilde{B}_{t,s}^* := (R_t C_t)^{-1} R_t G|_{\hat{t} \times \hat{s}}, \quad (8.13)$$

welches umgeformt gerade (8.11), also

$$(R_t C_t) \tilde{B}_{t,s}^* \approx R_t G|_{\hat{t} \times \hat{s}},$$

entspricht.

An dieser Stelle ist nun festzustellen, dass mit diesem Interpolations-Ansatz nicht einfach eine weitere Approximation nach der Green-Quadratur und ACA stattfindet, sondern sogar eine Verbesserung der Genauigkeit erreicht wird, indem $\tilde{B}_{t,s}$ durch Vorwärtseinsetzen anstatt durch $B_{t,s}D_t$ berechnet wird. Das liegt allein schon daran, dass $B_{t,s}$ und D_t aus Approximationen stammen ($B_{t,s}$ aus der Green-Quadratur und D_t aus ACA), während beim Vorwärtseinsetzen außer auf C_t nur auf direkt berechnete Einträge von $G|_{\hat{t} \times \hat{s}}$ zurückgegriffen wird.

Im Vergleich zum reinen Green-Quadratur-Verfahren produziert das Green-Hybrid-Verfahren deutlich kleinere lokale Ränge durch ACA und verschafft zusätzliche Genauigkeit durch die Interpolation.

Dadurch, dass A_t nur von dem Cluster t abhängt, müssen zudem A_t , C_t und R_t nur für die Cluster, nicht aber für alle Blöcke $b = (t, s)$ neu berechnet werden. Letzteres bedeutet dabei nicht nur einen geringeren Aufwand, sondern ermöglicht auch die Anwendung auf \mathcal{H}^2 -Matrizen, wie im kommenden Abschnitt 8.2 gezeigt wird.

8.2 Anwendung auf \mathcal{H}^2 -Matrizen

In diesem Abschnitt geht es um das Green-Hybrid-Verfahren für \mathcal{H}^2 -Matrizen. Die grundlegende Idee dabei, die Green'sche Formel für beide zu einem Block gehörenden Bounding Boxen anzuwenden, wird erst einmal am Beispiel von uniformen \mathcal{H} -Matrizen in Abschnitt 8.2.1 skizziert. Wie dann der Übergang zu \mathcal{H}^2 -Matrizen, also zu *geschachtelten* Clusterbasen funktioniert, wird in Abschnitt 8.2.2 entwickelt.

In Abschnitt 8.2.3 werden dann die Vorbereitungen aus den Abschnitten 8.2.1 und 8.2.2 genutzt, um die endgültige \mathcal{H}^2 -Matrix-Approximation zu formulieren.

8.2.1 Idee anhand uniformer \mathcal{H} -Matrizen

Wie bereits am Ende von Abschnitt 8.1.2 angedeutet, lässt die spezielle Struktur der Green-Quadratur und der Aufbau des Green-Hybrid-Verfahrens eine deutliche Effizienzsteigerung zu, indem die Cluster-Abhängigkeiten genauer betrachtet werden.

Die Kreuzapproximation im Green-Hybrid-Verfahren wird nur auf Cluster, nicht aber auf Blockcluster angewendet. Ebenso werden die Interpolationsoperatoren \mathcal{I}_t für jeden individuellen Cluster konstruiert.

Das bedeutet, dass es möglich ist, Rechenzeit einzusparen, indem die aufwändigen Berechnungen der Quadraturpunkte und von ACA mit vollständiger Pivotierung in einer vorbereitenden Phase (Setup-Phase) abgehandelt werden. Jeder Cluster wird in der Setup-Phase vorbehandelt, bevor die Blockcluster betrachtet werden.

In Abschnitt 5.2 wurde schon darauf hingewiesen, dass man ebenso gut das Normalengebiet Θ für die Green'sche Formel und damit die Parametrisierung $\Xi := \partial\Theta$ auch um die Bounding Box \mathcal{B}_σ legen kann, anstatt um \mathcal{B}_τ . Also wird dann für das Green-Hybrid-Verfahren die Approximation über den Cluster s gemacht.

Zumindest im Falle des Einfachschichtpotentials ist die Matrix G symmetrisch, so dass es keinen Grund gibt, warum $\mathcal{I}_t G|_{\hat{t} \times \hat{s}}$ gegenüber $G|_{\hat{t} \times \hat{s}} \mathcal{I}_s^*$ vorgezogen werden sollte.

8 Das effiziente Green-Hybrid-Verfahren

Wenn nun von *beiden* Seiten der Interpolationsoperator angewendet wird, läuft dieses auf die symmetrische Darstellung

$$G|_{\hat{t} \times \hat{s}} \approx \mathcal{I}_t G|_{\hat{t} \times \hat{s}} \mathcal{I}_s^*$$

hinaus. Mit der Definition des Interpolationsoperators aus (8.12) bedeutet das

$$\begin{aligned} G|_{\hat{t} \times \hat{s}} &\approx \underbrace{C_t (R_t C_t)^{-1}}_{=: V_t} \underbrace{R_t G|_{\hat{t} \times \hat{s}} R_s^*}_{=: S_b} \underbrace{((R_s C_s)^{-1})^* C_s^*}_{=: W_s^*} \\ &= V_t S_b W_s^*, \end{aligned} \quad (8.14)$$

welches bereits der Faktorisierung bei der Niedrigrangapproximation für \mathcal{H}^2 -Matrizen entspricht. Die kleine Matrix $S_b := R_t G|_{\hat{t} \times \hat{s}} R_s^* \in \mathbb{R}^{k \times k}$ ist dabei die Kopplungsmatrix wie in Definition 3.28. Mit $V_t := C_t (R_t C_t)^{-1} \mathbb{R}^{\hat{t} \times k}$ und $W_s := C_s (R_s C_s)^{-1} \mathbb{R}^{\hat{s} \times k}$ sind dann die Clusterbasen $(V_t)_{t \in \mathcal{T}_{\mathcal{I}}}$ und $(W_s)_{s \in \mathcal{T}_{\mathcal{J}}}$ beschrieben.

Auch wenn die Speicherung in der Niedrigrangapproximation (8.14) effizient ist, so ist gegenüber der \mathcal{H} -Matrix-Approximation aber noch nicht viel gewonnen. Dadurch, dass bis hierhin noch keine Schachtelung der Clusterbasen vorliegt, ist noch keine echte \mathcal{H}^2 -Matrix-Struktur erreicht, sondern lediglich die einer uniformen \mathcal{H} -Matrix wie in Bemerkung 3.31.

Mit der Konstruktion von geschachtelten Clusterbasen beschäftigt sich der folgende Abschnitt 8.2.2, so dass dieser Ansatz dann doch noch zu \mathcal{H}^2 -Matrizen und damit zu einer erheblichen Effizienzsteigerung im Vergleich zu \mathcal{H} -Matrizen führt.

8.2.2 Geschachtelte Clusterbasen

In Abschnitt 8.2.1 wurde bereits beschrieben, wie Clusterbasen $(V_t)_{t \in \mathcal{T}_{\mathcal{I}}}$ und $(W_s)_{s \in \mathcal{T}_{\mathcal{J}}}$ mit dem Green-Hybrid-Verfahren konstruiert werden können.

Um den Schritt zu \mathcal{H}^2 -Matrizen machen zu können, müssen diese jedoch geschachtelt sein, das heißt nach Definition 3.26, dass Transfermatrizen $(E_t)_{t \in \mathcal{T}_{\mathcal{I}}}$ und $(F_s)_{s \in \mathcal{T}_{\mathcal{J}}}$ existieren mit

$$V_t|_{\hat{t}' \times k_t} = V_{t'} E_{t'} \quad \text{für alle } t \in \mathcal{T}_{\mathcal{I}} \text{ und } t' \in \text{sons}(t)$$

und

$$W_s|_{\hat{s}' \times k_s} = W_{s'} F_{s'} \quad \text{für alle } s \in \mathcal{T}_{\mathcal{J}} \text{ und } s' \in \text{sons}(s).$$

Wenn diese Matrizen existieren, müssen nur noch für die Blätter $t \in \mathcal{L}_{\mathcal{I}}$ und $s \in \mathcal{L}_{\mathcal{J}}$ die V_t beziehungsweise die W_s gespeichert werden, da die restlichen V_t und W_s mit $t \in \mathcal{T}_{\mathcal{I}} \setminus \mathcal{L}_{\mathcal{I}}$ und $s \in \mathcal{T}_{\mathcal{J}} \setminus \mathcal{L}_{\mathcal{J}}$ mit Hilfe der Transfermatrizen berechnet werden können. Damit wäre eine \mathcal{H}^2 -Matrix-Darstellung nach Definition 3.29 ermöglicht.

In diesem Abschnitt soll daher ausführlich auf die Konstruktion der geschachtelten Clusterbasen $((V_t)_{t \in \mathcal{L}_{\mathcal{I}}}, (E_t)_{t \in \mathcal{T}_{\mathcal{I}}})$ und $((W_s)_{s \in \mathcal{L}_{\mathcal{J}}}, (F_s)_{s \in \mathcal{T}_{\mathcal{J}}})$ mit dem Green-Hybrid-Verfahren eingegangen werden.

Zunächst wird im Folgenden die Clusterbasis V_t eingehend betrachtet und dann die analoge Konstruktion der Clusterbasis W_s kurz vorgeführt.

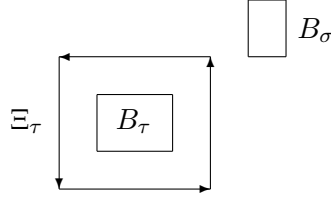
Clusterbasis V

Als erstes werden die V_t auf Blattebene konstruiert. Sei also $t \in \mathcal{L}_{\mathcal{I}}$ ein Blattcluster.

Die Kernapproximation findet wie zuvor statt. Genauer heißt das, dass ein Normalengebiet Θ_τ in Quaderform um die Bounding Box B_τ gelegt wird und die Green'sche Formel auf $u(x) := g(x, y)$ angewendet wird, so dass

$$g(x, y) = \int_{\Xi_\tau} g(x, z) \frac{\partial g}{\partial n_z}(z, y) d\Xi_\tau(z) - \int_{\Xi_\tau} \frac{\partial g}{\partial n_z}(x, z) g(z, y) d\Xi_\tau(z)$$

mit $\Xi_\tau := \partial\Theta_\tau$ heraus kommt.



Dann wird eine Parametrisierung für die $2d$ Teile von $\Xi_\tau := \bigcup_{\iota=1}^{2d} (\Xi_\tau)_\iota$ durch $\xi_\tau : U \rightarrow (\Xi_\tau)_\iota$ mit $U := [-1, 1]^{d-1}$ definiert und dafür die m Gauß-Quadraturpunkte auf ℓ Teilintervallen berechnet.

Die Green-Quadraturordnung q ist dabei wie in (8.2) und das Ergebnis der Quadratur ist gegeben durch

$$G|_{\hat{t} \times \hat{s}} \approx \sum_{\iota=1}^{2d} \check{A}_\iota \check{B}_\iota^* - \hat{A}_\iota \hat{B}_\iota^*$$

mit den $\check{A}_\iota, \hat{A}_\iota$ aus (8.4) und den $\check{B}_\iota, \hat{B}_\iota$ aus (8.7), so dass die Niedrigrangapproximation von $G|_{\hat{t} \times \hat{s}}$ durch die Green-Quadratur

$$G|_{\hat{t} \times \hat{s}} \approx A_t B_{t,s}^*$$

ist, wobei $A_t \in \mathbb{R}^{\hat{t} \times q}$ und $B_{t,s} \in \mathbb{R}^{\hat{s} \times q}$ wie in (8.3) und (8.6).

Wie zuvor bei dem Green-Hybrid-Verfahren für einfache \mathcal{H} -Matrizen, wird ACA mit vollständiger Pivotsuche auf die Matrix A_t wie in (8.8) angewendet mit dem Ergebnis

$$A_t \approx C_t D_t^*$$

mit $C_t \in \mathbb{R}^{\hat{t} \times k_t}$ und $D_t \in \mathbb{R}^{q \times k_t}$, wobei $k_t \leq q$ der Rang ist, den die Kreuzapproximation produziert. Es werden die Pivotindizes ξ_1, \dots, ξ_{k_t} , die Matrix C_t und die Dreiecksmatrix $R_t C_t \in \mathbb{R}^{k_t \times k_t}$ gespeichert, wie bereits im Abschnitt 8.1.2 erklärt wurde. Die Matrix D_t wird verworfen.

Mit dem Interpolationsoperator \mathcal{I}_t aus (8.12) ergibt sich dann die Niedrigrangapproximation

$$G|_{\hat{t} \times \hat{s}} \approx \mathcal{I}_t G|_{\hat{t} \times \hat{s}} = \underbrace{C_t (R_t C_t)^{-1}}_{=: V_t} \underbrace{R_t G|_{\hat{t} \times \hat{s}}}_{=: \tilde{B}_{t,s}^*} = V_t \tilde{B}_{t,s}^*,$$

8 Das effiziente Green-Hybrid-Verfahren

wobei anders, als bei dem Verfahren für \mathcal{H} -Matrizen, hier für die \mathcal{H}^2 -Matrizen

$$V_t := C_t(R_t C_t)^{-1} \in \mathbb{R}^{\hat{t} \times k_t} \quad (8.15)$$

definiert wird, wodurch die Clusterbasis auf Blattebene bestimmt wird. Außerdem lässt sich der Interpolationsoperator nun schreiben als

$$\mathcal{I}_t = V_t R_t. \quad (8.16)$$

Damit ist der erste Teil der geschachtelten Clusterbasis konstruiert und es fehlt noch der zweite Teil mit den Transfermatrizen.

Sei also $t \in \mathcal{T}_{\mathcal{I}} \setminus \mathcal{L}_{\mathcal{I}}$ und $t_1, t_2 \in \text{sons}(t)$.

Genauso wie im Fall $t \in \mathcal{L}_{\mathcal{I}}$ wird auch für $t \notin \mathcal{L}_{\mathcal{I}}$ als erstes die Green-Quadratur durchgeführt, so dass

$$G|_{\hat{t} \times \hat{s}} \approx A_t B_{t,s}^*$$

mit $A_t \in \mathbb{R}^{\hat{t} \times q}$ und $B_{t,s} \in \mathbb{R}^{\hat{s} \times q}$ berechnet wird.

Da die Clusterbasis von der Blattebene ausgehend aufgebaut wird und von dort aus die übrigen Cluster aus ihren Sohnclustern konstruiert werden, kann davon ausgegangen werden, dass für die Söhne $t_1, t_2 \in \text{sons}(t)$ bereits $C_{t_1}, C_{t_2}, R_{t_1}, R_{t_2}, V_{t_1}, V_{t_2}$ sowie die Interpolationsoperatoren \mathcal{I}_{t_1} und \mathcal{I}_{t_2} vorliegen. Für letztere gilt

$$\mathcal{I}_{t_1} = V_{t_1} R_{t_1} \quad \text{und} \quad \mathcal{I}_{t_2} = V_{t_2} R_{t_2}.$$

Da $\hat{t} = \hat{t}_1 \cup \hat{t}_2$ nach Definition 3.6, kann die Matrix A_t geschrieben werden als

$$A_t = \begin{pmatrix} A_t|_{\hat{t}_1 \times q} \\ A_t|_{\hat{t}_2 \times q} \end{pmatrix}. \quad (8.17)$$

Dazu sei eine Einschränkung auf die Pivotzeilen, die ACA im Schritt für die Sohncluster t_1 und t_2 ausgewählt hat, definiert. Die dort produzierten Ränge seien mit k_{t_1} für den Rang, den ACA für den Sohn t_1 geliefert hat und mit k_{t_2} für den Rang, den ACA für den Sohn t_2 geliefert hat, bezeichnet. Die aus der Einschränkung resultierende Matrix sei mit \hat{A}_t benannt, genauer:

$$\hat{A}_t = \begin{pmatrix} R_{t_1} A_t|_{\hat{t}_1 \times q} \\ R_{t_2} A_t|_{\hat{t}_2 \times q} \end{pmatrix} \in \mathbb{R}^{(k_1+k_2) \times q}. \quad (8.18)$$

Anstatt wie auf der Blattebene die Kreuzapproximation auf A_t anzuwenden, wird hier \hat{A}_t mit ACA komprimiert zu

$$\hat{A}_t \approx \hat{C}_t \hat{D}_t^*$$

mit $\hat{C}_t \in \mathbb{R}^{(k_1+k_2) \times q}$ und $\hat{D}_t \in \mathbb{R}^{q \times k_t}$, wobei k_t der von ACA gelieferte Rang ist. Erneut werden davon \hat{C}_t und die Pivotindizes $\hat{\xi}_1, \dots, \hat{\xi}_{k_t}$ sowie die untere Dreiecksmatrix $\hat{R}_t \hat{C}_t \in \mathbb{R}^{k_t \times k_t}$ anstelle von \hat{D}_t gespeichert.

Aus diesen Matrizen kann analog zu (8.12) ein Interpolationsoperator

$$\widehat{\mathcal{I}}_t := \widehat{C}_t(\widehat{R}_t\widehat{C}_t)^{-1}\widehat{R}_t$$

definiert werden.

Nun fehlt noch der Schritt hin zu der eigentlichen Schachtelung, um die Transfermatrizen definieren zu können. Dazu werden die Interpolationsoperatoren \mathcal{I}_{t_1} und \mathcal{I}_{t_2} der Sohncluster auf die Matrix A_t angewendet:

$$\begin{aligned} A_t &= \begin{pmatrix} A_t|_{\hat{t}_1 \times q} \\ A_t|_{\hat{t}_2 \times q} \end{pmatrix} \approx \begin{pmatrix} \mathcal{I}_{t_1} & \\ & \mathcal{I}_{t_2} \end{pmatrix} \begin{pmatrix} A_t|_{\hat{t}_1 \times q} \\ A_t|_{\hat{t}_2 \times q} \end{pmatrix} \\ &= \begin{pmatrix} V_{t_1} R_{t_1} & \\ & V_{t_2} R_{t_2} \end{pmatrix} \begin{pmatrix} A_t|_{\hat{t}_1 \times q} \\ A_t|_{\hat{t}_2 \times q} \end{pmatrix} \\ &= \begin{pmatrix} V_{t_1} & \\ & V_{t_2} \end{pmatrix} \begin{pmatrix} R_{t_1} A_t|_{\hat{t}_1 \times q} \\ R_{t_2} A_t|_{\hat{t}_2 \times q} \end{pmatrix} = \begin{pmatrix} V_{t_1} & \\ & V_{t_2} \end{pmatrix} \widehat{A}_t. \end{aligned}$$

Mit dem Dazwischenschieben des Interpolationsoperators $\widehat{\mathcal{I}}_t$ folgt

$$\begin{aligned} A_t &\approx \begin{pmatrix} V_{t_1} & \\ & V_{t_2} \end{pmatrix} \widehat{A}_t \\ &\approx \begin{pmatrix} V_{t_1} & \\ & V_{t_2} \end{pmatrix} \widehat{\mathcal{I}}_t \begin{pmatrix} R_{t_1} A_t|_{\hat{t}_1 \times q} \\ R_{t_2} A_t|_{\hat{t}_2 \times q} \end{pmatrix} \\ &= \begin{pmatrix} V_{t_1} & \\ & V_{t_2} \end{pmatrix} \widehat{C}_t(\widehat{R}_t\widehat{C}_t)^{-1}\widehat{R}_t \begin{pmatrix} R_{t_1} A_t|_{\hat{t}_1 \times q} \\ R_{t_2} A_t|_{\hat{t}_2 \times q} \end{pmatrix} \\ &= \begin{pmatrix} V_{t_1} & \\ & V_{t_2} \end{pmatrix} \begin{pmatrix} (\widehat{C}_t(\widehat{R}_t\widehat{C}_t)^{-1})|_{k_{t_1} \times k_t} \\ (\widehat{C}_t(\widehat{R}_t\widehat{C}_t)^{-1})|_{k_{t_2} \times k_t} \end{pmatrix} \widehat{R}_t \begin{pmatrix} R_{t_1} & \\ & R_{t_2} \end{pmatrix} \begin{pmatrix} A_t|_{\hat{t}_1 \times q} \\ A_t|_{\hat{t}_2 \times q} \end{pmatrix} \end{aligned}$$

und mit der Definition von R_t für $t \notin \mathcal{L}_{\mathcal{I}}$ mit $t_1, t_2 \in \text{sons}(t)$ durch

$$R_t := \widehat{R}_t \begin{pmatrix} R_{t_1} & \\ & R_{t_2} \end{pmatrix} \in \mathbb{R}^{k_t \times \hat{t}} \quad (8.19)$$

kann dieses geschrieben werden als

$$A_t \approx \begin{pmatrix} V_{t_1} & \\ & V_{t_2} \end{pmatrix} \begin{pmatrix} \widehat{C}_t(\widehat{R}_t\widehat{C}_t)^{-1}|_{k_{t_1} \times k_t} \\ \widehat{C}_t(\widehat{R}_t\widehat{C}_t)^{-1}|_{k_{t_2} \times k_t} \end{pmatrix} R_t A_t. \quad (8.20)$$

An dieser Stelle sind alle Informationen verfügbar, die eine sinnvolle Definition der Transfermatrizen erlauben durch

$$\begin{pmatrix} E_{t_1} \\ E_{t_2} \end{pmatrix} := \begin{pmatrix} \widehat{C}_t(\widehat{R}_t\widehat{C}_t)^{-1}|_{k_{t_1} \times k_t} \\ \widehat{C}_t(\widehat{R}_t\widehat{C}_t)^{-1}|_{k_{t_2} \times k_t} \end{pmatrix} \in \mathbb{R}^{(k_{t_1} + k_{t_2}) \times k_t} \quad (8.21)$$

8 Das effiziente Green-Hybrid-Verfahren

für $t_1, t_2 \in \text{sons}(t)$. Die Berechnung der V_t für $t \in \mathcal{T}_{\mathcal{I}} \setminus \mathcal{L}_{\mathcal{I}}$ sieht damit wie folgt aus:

$$V_t := \begin{pmatrix} V_{t_1} & \\ & V_{t_2} \end{pmatrix} \begin{pmatrix} E_{t_1} \\ E_{t_2} \end{pmatrix} := \begin{pmatrix} V_{t_1} E_{t_1} \\ V_{t_2} E_{t_2} \end{pmatrix} \in \mathbb{R}^{\hat{t} \times k_t}.$$

Nun kann (8.20) in noch kompakterer Form geschrieben werden als

$$A_t \approx V_t R_t A_t = \mathcal{I}_t A_t$$

mit dem Interpolationsoperator \mathcal{I}_t , wie bereits für $t \in \mathcal{L}_{\mathcal{I}}$, auch für $t \in \mathcal{T}_{\mathcal{I}} \setminus \mathcal{L}_{\mathcal{I}}$ definiert durch

$$\mathcal{I}_t := V_t R_t. \quad (8.22)$$

Zusammengefasst ist die geschachtelte Clusterbasis $((V_t)_{t \in \mathcal{L}_{\mathcal{I}}}, (E_t)_{t \in \mathcal{T}_{\mathcal{I}}})$ also gegeben durch

$$\begin{aligned} V_t &= C_t (R_t C_t)^{-1} && \text{für } t \in \mathcal{L}_{\mathcal{I}} \\ V_t &= \begin{pmatrix} V_{t_1} E_{t_1} \\ V_{t_2} E_{t_2} \end{pmatrix} && \text{für } t \notin \mathcal{L}_{\mathcal{I}} \end{aligned} \quad (8.23)$$

mit den in (8.21) definierten E_{t_1} und E_{t_2} .

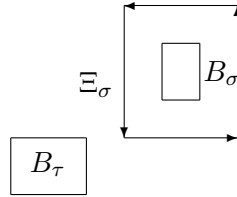
Clusterbasis W

Die Konstruktion der W_s erfolgt analog zu der der V_t , dennoch werden hier die wichtigsten Schritte noch einmal vorgeführt.

Sei $s \in \mathcal{L}_{\mathcal{J}}$ zunächst ein Blattcluster. Es wird ein Normalengebiet Θ_σ in Quaderform um die Bounding Box \mathcal{B}_σ gelegt und die Green'sche Formel auf $u(y) := g(x, y)$ angewendet, so dass

$$g(x, y) = \int_{\Xi_\sigma} g(z, y) \frac{\partial g}{\partial n_z}(x, z) d\Xi_\sigma(z) - \int_{\Xi_\sigma} \frac{\partial g}{\partial n_z}(z, y) g(x, z) d\Xi_\sigma(z)$$

mit $\Xi_\sigma := \partial\Theta_\sigma$ heraus kommt.



Die Parametrisierung $\xi_\sigma : [-1, 1]^{d-1} \rightarrow (\Xi_\sigma)_\iota$ für die $2d$ Teile von $\Xi_\sigma := \bigcup_{\iota=1}^{2d} (\Xi_\sigma)_\iota$ wird definiert und die m Gauß-Quadraturpunkte auf ℓ Teilintervallen berechnet. Die Green-Quadraturordnung q ist erneut wie in (8.2) und das Ergebnis der Quadratur ist

$$G|_{\hat{t} \times \hat{s}} \approx \sum_{\iota=1}^{2d} \check{A}_\iota \check{B}_\iota^* - \hat{A}_\iota \hat{B}_\iota^*$$

mit

$$\begin{aligned} (\check{A}_l)_{i\nu} &= G_{\det}(t) w_\nu \int_\Gamma \frac{\partial g}{\partial n_z}(x, \xi_l(t_\nu)) \varphi_i(x) \, d\Gamma(x) \\ (\hat{A}_l)_{i\nu} &= G_{\det}(t) w_\nu \int_\Gamma g(x, \xi_l(t_\nu)) \varphi_i(x) \, d\Gamma(x) \end{aligned}$$

und

$$\begin{aligned} (\check{B}_l)_{j\nu} &= \int_\Gamma g(\xi_l(t_\nu), y) \varphi_j(y) \, d\Gamma(y) \\ (\hat{B}_l)_{j\nu} &= \int_\Gamma \frac{\partial g}{\partial n_z}(\xi_l(t_\nu), y) \varphi_j(y) \, d\Gamma(y), \end{aligned}$$

also

$$G|_{\hat{t} \times \hat{s}} \approx A_{t,s} B_s^* \quad (8.24)$$

mit $A_{t,s} \in \mathbb{R}^{\hat{t} \times q}$ und $B_s \in \mathbb{R}^{\hat{s} \times q}$. ACA angewendet auf B_s liefert

$$B_s \approx C_s D_s^*$$

mit $C_s \in \mathbb{R}^{\hat{s} \times k_s}$ und $D_s \in \mathbb{R}^{q \times k_s}$ mit dem Rang $k_s \leq q$. Es werden wieder C_s , die Pivotindizes, die damit entstehende Abbildung $R_s \in \mathbb{R}^{k_s \times \hat{s}}$, sowie die untere Dreiecksmatrix $R_s C_s \in \mathbb{R}^{k_s \times k_s}$ gespeichert. Mit dem Interpolationsoperator \mathcal{I}_s , definiert durch

$$\mathcal{I}_s := C_s (R_s C_s)^{-1} R_s,$$

ergibt sich die Niedrigrangapproximation

$$G|_{\hat{t} \times \hat{s}} \approx G|_{\hat{t} \times \hat{s}} \mathcal{I}_s^* = G|_{\hat{t} \times \hat{s}} R_s^* W_s^*$$

mit

$$W_s = C_s (R_s C_s)^{-1} \in \mathbb{R}^{\hat{s} \times k_s}$$

und es ist

$$\mathcal{I}_s = W_s R_s.$$

Damit ist der erste Teil der Clusterbasis $((W_s)_{s \in \mathcal{L}_J}, (F_s)_{s \in \mathcal{T}_J})$ definiert und es fehlen noch die Transfermatrizen.

Seien also $s \in \mathcal{T}_J \setminus \mathcal{L}_J$ und $s_1, s_2 \in \text{sons}(s)$. Die Green-Quadratur liefert wieder

$$G|_{\hat{t} \times \hat{s}} \approx A_{t,s} B_s^*$$

mit $A_{t,s} \in \mathbb{R}^{\hat{t} \times q}$ und $B_s \in \mathbb{R}^{\hat{s} \times q}$. Analog zu (8.18) sei die Einschränkung

$$\hat{B}_s := \begin{pmatrix} R_{s_1} B_s|_{\hat{t}_1 \times q} \\ R_{s_2} B_s|_{\hat{t}_2 \times q} \end{pmatrix} \in \mathbb{R}^{(k_{s_1} + k_{s_2}) \times q}$$

auf B_s definiert. Die Kreuzapproximation liefert

$$\hat{B}_s \approx \hat{C}_s \hat{D}_s^*$$

8 Das effiziente Green-Hybrid-Verfahren

mit $\widehat{C}_s \in \mathbb{R}^{(k_{s_1}+k_{s_2}) \times k_s}$, $\widehat{D}_s \in \mathbb{R}^{q \times k_s}$ und den Rang k_s . Mit Hilfe der durch die Pivotindizes von ACA definierten Matrix $\widehat{R}_s \in \mathbb{R}^{k_s \times (k_{s_1}+k_{s_2})}$ wird analog zu (8.19) die Matrix R_s definiert durch

$$R_s := \widehat{R}_s \begin{pmatrix} R_{s_1} & \\ & R_{s_2} \end{pmatrix} \in \mathbb{R}^{k_s \times \hat{s}}$$

für $s_1, s_2 \in \text{sons}(s)$. Analog zu (8.21) können die Transfermatrizen nun definiert werden als

$$\begin{pmatrix} F_{s_1} \\ F_{s_2} \end{pmatrix} := \begin{pmatrix} \widehat{C}_s(\widehat{R}_s\widehat{C}_s)^{-1}|_{k_{s_1} \times k_s} \\ \widehat{C}_s(\widehat{R}_s\widehat{C}_s)^{-1}|_{k_{s_2} \times k_s} \end{pmatrix} \in \mathbb{R}^{(k_{s_1}+k_{s_2}) \times k_s}, \quad (8.25)$$

womit die Schachtelung der W_s gegeben ist durch

$$W_s := \begin{pmatrix} W_{s_1} & \\ & W_{s_2} \end{pmatrix} \begin{pmatrix} F_{s_1} \\ F_{s_2} \end{pmatrix} := \begin{pmatrix} W_{s_1}F_{s_1} \\ W_{s_2}F_{s_2} \end{pmatrix} \in \mathbb{R}^{\hat{s} \times k_s} \quad (8.26)$$

und der zugehörige Interpolationsoperator ist

$$\mathcal{I}_s := W_s R_s.$$

Zusammengefasst ist die geschachtelte Clusterbasis $((W_s)_{s \in \mathcal{L}_{\mathcal{J}}}, (F_s)_{s \in \mathcal{T}_{\mathcal{J}}})$ also gegeben durch

$$W_s = C_s(R_s C_s)^{-1} \quad \text{für } s \in \mathcal{L}_{\mathcal{J}} \quad (8.27)$$

$$W_s = \begin{pmatrix} W_{s_1}F_{s_1} \\ W_{s_2}F_{s_2} \end{pmatrix} \quad \text{für } s \notin \mathcal{L}_{\mathcal{J}} \quad (8.28)$$

mit den in (8.25) definierten F_{s_1} und F_{s_2} .

8.2.3 \mathcal{H}^2 -Matrix-Approximation

Es bleibt abschließend noch das Aufstellen der Niedrigrangapproximationen zu beschreiben. Sei dazu $G|_{\hat{t} \times \hat{s}}$ mit $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+$ ein zulässiger Matrixblock der Matrix G mit den Einträgen aus (2.4).

Bevor die Blockcluster betrachtet werden, können in einer vorbereitenden Phase, einem Setup, die Clusterbasen $((V_t)_{t \in \mathcal{L}_{\mathcal{I}}}, (E_t)_{t \in \mathcal{T}_{\mathcal{I}}})$ und $((W_s)_{s \in \mathcal{L}_{\mathcal{J}}}, (F_s)_{s \in \mathcal{T}_{\mathcal{J}}})$, wie in Abschnitt 8.2.2 beschrieben wurde, berechnet werden, ebenso wie die Interpolationsoperatoren \mathcal{I}_t und \mathcal{I}_s . Dieses ist möglich, da die Clusterbasen jeweils nur von einem Cluster und seinen Söhnen abhängen und keine Daten der Blockcluster benötigt werden.

Erst in der eigentlichen Berechnung der Niedrigrangapproximation des Matrixblocks $G|_{\hat{t} \times \hat{s}}$ kommen die Blockcluster ins Spiel. Dabei sieht die Approximation ebenso aus, wie bereits im Fall von uniformen \mathcal{H} -Matrizen in Abschnitt 8.2.1 gesehen: Die Interpolationsoperatoren \mathcal{I}_t und \mathcal{I}_s werden von beiden Seiten auf $G|_{\hat{t} \times \hat{s}}$ angewendet mit dem Ergebnis

$$\begin{aligned} G|_{\hat{t} \times \hat{s}} &\approx \mathcal{I}_t G|_{\hat{t} \times \hat{s}} \mathcal{I}_s^* \\ &= V_t R_t G|_{\hat{t} \times \hat{s}} R_s^* W_s^*. \end{aligned}$$

Mit der Definition der Kopplungsmatrix S_b durch

$$S_b = R_t G|_{\hat{t} \times \hat{s}} R_s^* \in \mathbb{R}^{k_t \times k_s}$$

ergibt sich dann die typische \mathcal{H}^2 -Matrix-Niedrigrangapproximation

$$G|_{\hat{t} \times \hat{s}} \approx V_t S_b W_s^* .$$

Damit sind in der Berechnungsphase nur noch Kopplungsmatrizen $(S_b)_{b \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+}$ aufzustellen. Diese bestehen aus k_t Zeilen und k_s Spalten von $G|_{\hat{t} \times \hat{s}}$, die durch die Matrizen R_t und R_s ausgewählt werden. Dabei kommen R_t und k_t aus der ACA-Anwendung zur Konstruktion von E_t , analog R_s und k_s aus der von F_s .

Das Green-Hybrid-Verfahren vermeidet also die relativ hohen lokalen Ränge, die die Green-Quadratur liefert, es kann ebenso gut auch für \mathcal{H}^2 -Matrizen genutzt werden und kann sehr effizient angewendet werden.

Damit bleiben noch zwei offene Fragen:

Die erste Frage ist, ob auch das Green-Hybrid-Verfahren beweisbar ist. Die Antwort ist ja, da hier die vollständige Pivotierung für ACA genutzt wird. Die genaueren Aufwandsabschätzungen befinden sich im folgenden Kapitel 9.

Die zweite Frage ist, wie das Green-Hybrid-Verfahren in numerischen Experimenten abschneidet. Die Ergebnisse zu einigen vergleichenden Experimenten mit dem Green-Quadratur-Verfahren, der Kreuzapproximation, der Interpolation und der Multipol-Methode werden dazu in Kapitel 10 präsentiert.

9 Fehleranalyse des Green-Hybrid-Verfahrens

In diesem Kapitel werden Fehlerabschätzungen für das Green-Hybrid-Verfahren hergeleitet. Im Abschnitt 9.1 wird zunächst der Fehler des Green-Hybrid-Verfahrens für \mathcal{H} -Matrizen untersucht und in Abschnitt 9.2 die Ergebnisse davon für die Abschätzungen für das Verfahren für \mathcal{H}^2 -Matrizen genutzt.

9.1 Fehlerabschätzung für das Hybrid-Verfahren

Im Folgenden wird eine Fehlerschranke für das Green-Hybrid-Verfahren für die Niedrigrangapproximation eines zulässigen Matrixblocks $G|_{\hat{t} \times \hat{s}}$ mit $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+$ für eine \mathcal{H} -Matrix gesucht.

Da das Green-Hybrid-Verfahren in jedem Schritt auf der Green-Quadratur-Methode aufbaut, wird der Fehler dieser Methode in der Fehlerabschätzung des Hybrid-Verfahrens wieder auftauchen. Dazu werden die in Kapitel 6 gefundenen Fehlerschranken für die Green-Quadratur-Methode in der Variablen ε_{green} zusammengefasst, so dass für die Niedrigrangapproximation mit der Green-Quadratur

$$G|_{\hat{t} \times \hat{s}} \approx A_t B_{t,s}^*$$

die Abschätzung

$$\|G|_{\hat{t} \times \hat{s}} - A_t B_{t,s}^*\|_2 \leq \varepsilon_{green} \quad (9.1)$$

gilt. Für die Anwendung von ACA auf A_t wurde bereits in (8.8) die Fehlerschranke mit ε_{aca} bezeichnet, so dass für

$$A_t \approx C_t D_t^*$$

die Abschätzung

$$\|A_t - C_t D_t^*\|_2 \leq \varepsilon_{aca} \quad (9.2)$$

gilt.

Um eine Abschätzung für die Kombination der Green-Quadratur und ACA zu erhalten, wird noch das folgende Lemma benötigt.

Lemma 9.1 (Abschätzung von $B_{t,s}$) Sei $\text{diam}(\Omega) \leq 1$. Die Menge der Gauß-Quadraturpunkte auf den ℓ Teilintervallen sei mit M bezeichnet. Sei $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+$ und sei $B_{t,s}$ die Matrix aus (8.6). Dann gibt es $C > 0$ und $C_\varphi > 0$ unabhängig von t, s und M so, dass

$$\|B_{t,s}\|_2 \leq \frac{C}{\delta^{\sigma+1}} C_\varphi \sqrt{\#\hat{s}} \sqrt{\#M}.$$

9 Fehleranalyse des Green-Hybrid-Verfahrens

Beweis. Da die Kernfunktion g asymptotisch glatt mit Singularitätsgrad σ ist, lässt sich eine Konstante $C_g > 0$ finden, so dass mit Lemma 6.2 und Folgerung 6.13

$$\|g\|_{\infty, \Xi \times \mathcal{B}_\sigma} \leq \frac{C_g}{\|z - y\|^\sigma} \leq \frac{C_g}{\text{dist}(\Xi, y)^\sigma} \leq \frac{C_g}{\delta^\sigma}$$

für $z \in \Xi$ und $y \in \mathcal{B}_\sigma$ gilt. Ebenso kann man $C_h > 0$ für die Normalenableitung von g finden, da diese asymptotisch glatt mit Singularitätsgrad $\sigma + 1$ ist, so dass

$$\left\| \frac{\partial g}{\partial n_z} \right\|_{\infty, \Xi \times \mathcal{B}_\sigma} \leq \frac{C_h}{\|z - y\|^{\sigma+1}} \leq \frac{C_h}{\text{dist}(\Xi, y)^{\sigma+1}} \leq \frac{C_h}{\delta^{\sigma+1}}$$

für $z \in \Xi$ und $y \in \mathcal{B}_\sigma$ gilt. Da $\text{diam}(\Omega) \leq 1$, ist auf jeden Fall auch $\delta \leq 1$ und es gilt mit $C := \max\{C_g, C_h\}$

$$\|g\|_{\infty, \Xi \times \mathcal{B}_\sigma} \leq \frac{C_g}{\delta^\sigma} \leq \frac{C}{\delta^{\sigma+1}} \quad \text{und} \quad \left\| \frac{\partial g}{\partial n_z} \right\|_{\infty, \Xi \times \mathcal{B}_\sigma} \leq \frac{C_h}{\delta^{\sigma+1}} \leq \frac{C}{\delta^{\sigma+1}},$$

so dass eine gemeinsame Abschätzung der Matrixeinträge von $B_{t,s}$ aus (8.6) mit (8.7) möglich ist. Es gilt

$$|(B_{t,s})_{j\nu}| \leq \left| \int_{\Gamma} \frac{C}{\delta^{\sigma+1}} \varphi_j(y) \, d\Gamma(y) \right| \leq \frac{C}{\delta^{\sigma+1}} \int_{\Gamma} |\varphi_j(y)| \, d\Gamma(y) \quad (9.3)$$

für alle $j \in \hat{s}$ und $\nu \in \{1, \dots, m^{d-1} \ell^{d-1}\}$ wobei $d \in \{2, 3\}$. Es kann angenommen werden, dass ein $C_\varphi > 0$ existiert, so dass

$$\int_{\Gamma} |\varphi_j(y)| \, d\Gamma(y) \leq C_\varphi \quad (9.4)$$

für alle $j \in \mathcal{J}$ ist. Mit (9.4) folgt für (9.3) dann

$$|(B_{t,s})_{j\nu}| \leq \frac{C}{\delta^{\sigma+1}} C_\varphi.$$

Damit gilt schließlich

$$\begin{aligned} \|B_{t,s}\|_2 &= \max_{x,y \neq 0} \left\{ \frac{|\langle B_{t,s}x, y \rangle|}{\|x\|_2 \|y\|_2} \right\} \\ &\leq \max_{x,y \neq 0} \left\{ \frac{|\sum_{j \in \hat{s}} \sum_{\nu \in M} (B_{t,s})_{j\nu} x_\nu y_j|}{\|x\|_2 \|y\|_2} \right\} \\ &\leq \max_{x,y \neq 0} \left\{ \frac{C}{\delta^{\sigma+1}} C_\varphi \frac{\sum_{j \in \hat{s}} \sum_{\nu \in M} |x_\nu| |y_j|}{\|x\|_2 \|y\|_2} \right\} \\ &\leq \frac{C}{\delta^{\sigma+1}} C_\varphi \max_{x,y \neq 0} \left\{ \frac{\sqrt{\#\hat{s}} \sqrt{\#M} \|x\|_2 \|y\|_2}{\|x\|_2 \|y\|_2} \right\} \\ &= \frac{C}{\delta^{\sigma+1}} C_\varphi \sqrt{\#\hat{s}} \sqrt{\#M}. \end{aligned}$$

■

9.1 Fehlerabschätzung für das Hybrid-Verfahren

Mit Hilfe dieses Lemmas kann nun die Fehlerabschätzung für die Kombination aus Green-Quadratur und ACA, die im Green-Hybrid-Verfahren genutzt wird, formuliert werden.

Satz 9.2 (Fehler der Green-Quadratur in Kombination mit ACA) Sei $\text{diam}(\Omega) \leq 1$. Sei M die Menge der Quadraturpunkte der Green-Quadratur. Sei $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+$ und \bar{A}_t und $\bar{B}_{t,s}$ wie in (8.9). Dann gilt mit $C, C_\varphi > 0$ aus Lemma 9.1

$$\|G|_{\hat{t} \times \hat{s}} - \bar{A}_t \bar{B}_{t,s}^*\|_2 \leq \varepsilon_{green} + \varepsilon_{aca} \frac{C}{\delta^{\sigma+1}} C_\varphi \sqrt{\#\hat{s}} \sqrt{\#M}.$$

Beweis. Es gilt

$$\begin{aligned} \|G|_{\hat{t} \times \hat{s}} - \bar{A}_t \bar{B}_{t,s}^*\|_2 &= \|G|_{\hat{t} \times \hat{s}} - C_t D_t^* B_{t,s}^*\|_2 \\ &= \|G|_{\hat{t} \times \hat{s}} - A_t B_{t,s}^* + A_t B_{t,s}^* - C_t D_t^* B_{t,s}^*\|_2 \\ &\leq \|G|_{\hat{t} \times \hat{s}} - A_t B_{t,s}^*\|_2 + \|A_t B_{t,s}^* - C_t D_t^* B_{t,s}^*\|_2. \end{aligned}$$

Mit (9.1) lässt sich dieses weiter abschätzen zu

$$\begin{aligned} \|G|_{\hat{t} \times \hat{s}} - \bar{A}_t \bar{B}_{t,s}^*\|_2 &\leq \varepsilon_{green} + \|(A_t - C_t D_t^*) B_{t,s}^*\|_2 \\ &\leq \varepsilon_{green} + \|A_t - C_t D_t^*\|_2 \|B_{t,s}^*\|_2 \end{aligned}$$

und mit (9.2) folgt

$$\|G|_{\hat{t} \times \hat{s}} - \bar{A}_t \bar{B}_{t,s}^*\|_2 \leq \varepsilon_{green} + \varepsilon_{aca} \|B_{t,s}^*\|_2.$$

Mit Lemma 9.1 folgt schließlich die Behauptung

$$\|G|_{\hat{t} \times \hat{s}} - \bar{A}_t \bar{B}_{t,s}^*\|_2 \leq \varepsilon_{green} + \varepsilon_{aca} \frac{C}{\delta^{\sigma+1}} C_\varphi \sqrt{\#\hat{s}} \sqrt{\#M}.$$

■

Damit ist eine Fehlerschranke für die kombinierte Ausführung der Green-Quadratur und ACA beschrieben. Um den Gesamtfehler des Green-Hybrid-Verfahrens fassen zu können, muss noch der Einfluss des Interpolationsoperators \mathcal{I}_t aus (8.12) berücksichtigt werden.

Wie bereits im Falle der Fehleranalyse der Interpolation in Abschnitt 4.1.3 mit der Stabilitätskonstante aus Definition 4.1, kann der Gesamtfehler formuliert werden, indem die Abschätzung aus Satz 9.2 mit einer Stabilitätseigenschaft von \mathcal{I}_t kombiniert wird.

Eine solche Eigenschaft wird aus der Definition von \mathcal{I}_t über das Ergebnis von ACA gefunden. Denn für adaptive Kreuzapproximations-Methoden kann nach [5, Lemma 1] angenommen werden, dass eine Zahl Λ_t existiert, die nur logarithmisch mit ε_{aca} wächst, so dass

$$\|\mathcal{I}_t\|_2 = \|C_t (R_t C_t)^{-1} R_t\|_2 \leq \Lambda_t \quad (9.5)$$

gilt. Mit dieser Stabilitätskonstante kann schließlich der folgende Satz formuliert werden.

Satz 9.3 (Fehlerschranke für das Green-Hybrid-Verfahren) Sei $\text{diam}(\Omega) \leq 1$. Sei M die Menge der Quadraturpunkte der Green-Quadratur. Sei Λ_t wie in (9.5). Sei $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+$ und $\tilde{B}_{t,s}$ wie in (8.13). Dann gilt für die mit dem Green-Hybrid-Verfahren berechnete Niedrigrangapproximation

$$G|_{\hat{t} \times \hat{s}} \approx C_t \tilde{B}_{t,s}^*$$

die Abschätzung

$$\|G|_{\hat{t} \times \hat{s}} - C_t \tilde{B}_{t,s}^*\|_2 \leq (1 + \Lambda_t)(\varepsilon_{\text{green}} + \varepsilon_{\text{aca}} \frac{C}{\delta^{\sigma+1}} C_\varphi \sqrt{\#\hat{s}} \sqrt{\#M})$$

mit $C, C_\varphi > 0$ aus Lemma 9.1.

Beweis. Es gilt mit (8.13)

$$\begin{aligned} \|G|_{\hat{t} \times \hat{s}} - C_t \tilde{B}_{t,s}^*\|_2 &= \|G|_{\hat{t} \times \hat{s}} - C_t (R_t C_t)^{-1} R_t G|_{\hat{t} \times \hat{s}}\|_2 \\ &= \|G|_{\hat{t} \times \hat{s}} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}}\|_2. \end{aligned}$$

Da $\mathcal{I}_t C_t = C_t$ und $\bar{A}_t = C_t$ gilt, folgt weiter

$$\begin{aligned} \|G|_{\hat{t} \times \hat{s}} - C_t \tilde{B}_{t,s}^*\|_2 &= \|G|_{\hat{t} \times \hat{s}} - C_t \bar{B}_{t,s}^* + \mathcal{I}_t C_t \bar{B}_{t,s}^* - \mathcal{I}_t G|_{\hat{t} \times \hat{s}}\|_2 \\ &\leq \|G|_{\hat{t} \times \hat{s}} - \bar{A}_t \bar{B}_{t,s}^*\|_2 + \|\mathcal{I}_t (\bar{A}_t \bar{B}_{t,s}^* - G|_{\hat{t} \times \hat{s}})\|_2. \end{aligned}$$

Mit (9.5) und Satz 9.2 folgt schließlich

$$\begin{aligned} \|G|_{\hat{t} \times \hat{s}} - C_t \tilde{B}_{t,s}^*\|_2 &\leq \|G|_{\hat{t} \times \hat{s}} - \bar{A}_t \bar{B}_{t,s}^*\|_2 + \Lambda_t \|\bar{A}_t \bar{B}_{t,s}^* - G|_{\hat{t} \times \hat{s}}\|_2 \\ &= (1 + \Lambda_t) \|G|_{\hat{t} \times \hat{s}} - \bar{A}_t \bar{B}_{t,s}^*\|_2 \\ &\leq (1 + \Lambda_t)(\varepsilon_{\text{green}} + \varepsilon_{\text{aca}} \frac{C}{\delta^{\sigma+1}} C_\varphi \sqrt{\#\hat{s}} \sqrt{\#M}). \end{aligned}$$

■

Der Satz 9.3 lässt den Schluss zu, dass jede gewünschte Genauigkeit erreicht werden kann, wenn die Green-Quadraturordnung aus m und ℓ sowie die Fehlertoleranz ε_{aca} entsprechend gewählt werden.

9.2 Fehlerabschätzung im \mathcal{H}^2 -Fall

Für die Fehlerabschätzung des Green-Hybrid-Verfahrens im \mathcal{H}^2 -Matrizen-Fall müssen zunächst einige Eigenschaften der Konstruktion aus Abschnitt 8.2.2 untersucht werden. Als erstes wird die Abschätzung, die in Lemma 9.1 für $B_{t,s}$ aus (8.6) gemacht wurde, auch für $A_{t,s}$ aus (8.24) benötigt.

Lemma 9.4 (Abschätzung von $A_{t,s}$) Sei $\text{diam}(\Omega) \leq 1$. Die Menge der Quadraturpunkte der Green-Quadratur sei mit M bezeichnet. Sei $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+$ und sei $A_{t,s}$ die Matrix aus (8.24). Dann gibt es $\tilde{C} > 0$ und $C_\varphi > 0$ unabhängig von t, s und M so, dass

$$\|A_{t,s}\|_2 \leq \frac{\tilde{C}}{\delta^{\sigma+1}} C_\varphi \sqrt{\#\hat{t}} \sqrt{\#M}.$$

Beweis. Wie im Beweis von Lemma 9.1 gilt mit der asymptotischen Glattheit

$$\|g\|_{\infty, \mathcal{B}_\tau \times \Xi} \leq \frac{C_g}{\|x-z\|^\sigma} \leq \frac{C_g}{\delta^\sigma} \leq \frac{C}{\delta^{\sigma+1}}$$

und

$$\left\| \frac{\partial g}{\partial n_z} \right\|_{\infty, \mathcal{B}_\tau \times \Xi} \leq \frac{C_h}{\|x-z\|^{\sigma+1}} \leq \frac{C_h}{\delta^{\sigma+1}} \leq \frac{C}{\delta^{\sigma+1}}$$

für $z \in \Xi$, $x \in \mathcal{B}_\tau$ mit $C_g, C_h > 0$ und $C := \max\{C_g, C_h\}$, so dass die Matrixeinträge von $A_{t,s}$ abzuschätzen sind durch

$$\begin{aligned} |(A_{t,s})_{i\nu}| &\leq \left| G_{\det}(t) w_\nu \int_{\Gamma} \frac{C}{\delta^{\sigma+1}} \varphi_i(x) \, d\Gamma(x) \right| \\ &\leq G_{\det}(t) w_\nu \frac{C}{\delta^{\sigma+1}} \int_{\Gamma} |\varphi_i(x)| \, d\Gamma(x) \\ &= \frac{\tilde{C}}{\delta^{\sigma+1}} \int_{\Gamma} |\varphi_i(x)| \, d\Gamma(x) \end{aligned}$$

mit $\tilde{C} := G_{\det}(t) w_\nu C$. Mit demselben Argument wie in (9.4) existiert $C_\varphi > 0$, so dass

$$|(A_{t,s})_{i\nu}| \leq \frac{\tilde{C}}{\delta^{\sigma+1}} C_\varphi$$

gilt. Damit folgt dann analog zum Beweis von Lemma 9.1

$$\|A_{t,s}\|_2 \leq \frac{\tilde{C}}{\delta^{\sigma+1}} C_\varphi \sqrt{\#\hat{t}} \sqrt{\#M}.$$

■

Eine weitere Eigenschaft der Clusterbasen-Konstruktion ist von Interesse, die Gegenstand des folgenden Lemmas ist.

Lemma 9.5 (Identität) *Seien V und W die durch das Green-Hybrid-Verfahren berechneten Clusterbasen. Dann gilt*

$$\begin{aligned} R_t V_t &= I && \text{für alle } t \in \mathcal{T}_{\mathcal{I}} \text{ und} \\ R_s W_s &= I && \text{für alle } s \in \mathcal{T}_{\mathcal{J}}, \end{aligned}$$

wobei mit I die Einheitsmatrix bezeichnet sei.

Beweis. Zunächst wird die Behauptung für V_t durch Induktion über die Anzahl aller Söhne inklusive des Clusters selbst, also über $\#\text{sons}^*(t)$ bewiesen.

Sei $\#\text{sons}^*(t) = 1$, also $\text{sons}^*(t) = \{t\}$ und damit $t \in \mathcal{L}_{\mathcal{I}}$. Dann gilt für V_t mit $t \in \mathcal{L}_{\mathcal{I}}$ aus (8.15)

$$R_t V_t = R_t C_t (R_t C_t)^{-1} = I.$$

Gelte die Behauptung für $\#\text{sons}^*(t) \leq n$.

9 Fehleranalyse des Green-Hybrid-Verfahrens

Sei $\#\text{sons}^*(t) = n+1$, also $t \notin \mathcal{L}_{\mathcal{I}}$. Für $t_1, t_2 \in \text{sons}(t)$ gilt $\#\text{sons}^*(t_1), \#\text{sons}^*(t_2) \leq n$. Es gilt nach (8.23) und mit (8.21)

$$V_t := \begin{pmatrix} V_{t_1} & \\ & V_{t_2} \end{pmatrix} \widehat{C}_t(\widehat{R}_t \widehat{C}_t)^{-1}$$

und mit (8.19)

$$R_t := \widehat{R}_t \begin{pmatrix} R_{t_1} & \\ & R_{t_2} \end{pmatrix},$$

so dass

$$\begin{aligned} R_t V_t &= \widehat{R}_t \begin{pmatrix} R_{t_1} & \\ & R_{t_2} \end{pmatrix} \begin{pmatrix} V_{t_1} & \\ & V_{t_2} \end{pmatrix} \widehat{C}_t(\widehat{R}_t \widehat{C}_t)^{-1} \\ &= \widehat{R}_t \begin{pmatrix} R_{t_1} V_{t_1} & \\ & R_{t_2} V_{t_2} \end{pmatrix} \widehat{C}_t(\widehat{R}_t \widehat{C}_t)^{-1} \end{aligned}$$

folgt. Nach Induktionsvoraussetzung gilt $R_{t_1} V_{t_1} = I$ und $R_{t_2} V_{t_2} = I$ und es folgt schließlich

$$R_t V_t = \widehat{R}_t \widehat{C}_t(\widehat{R}_t \widehat{C}_t)^{-1} = I.$$

Nach analoger Argumentation erhält man ebenso

$$R_s W_s = I.$$

■

Um weitere Abschätzungen zuzulassen, muss ähnlich wie in (9.5) die Annahme gemacht werden, dass Stabilitätskonstanten Λ_t und Λ_s für die Interpolationsoperatoren \mathcal{I}_t und \mathcal{I}_s existieren, so dass

$$\begin{aligned} \|\mathcal{I}_t\|_2 &\leq \Lambda_t && \text{für alle } t \in \mathcal{T}_{\mathcal{I}} \text{ und} \\ \|\mathcal{I}_s\|_2 &\leq \Lambda_s && \text{für alle } s \in \mathcal{T}_{\mathcal{J}} \end{aligned} \tag{9.6}$$

gilt.

Es lässt sich folgende Aussage über die Clusterbasen beweisen, die diese damit auch gleich in Verbindung mit den Stabilitätskonstanten bringt.

Lemma 9.6 (Gleichheit unter der Norm) *Seien V und W die durch das Green-Hybrid-Verfahren berechneten Clusterbasen und \mathcal{I}_t und \mathcal{I}_s die entsprechenden Interpolationsoperatoren. Dann gilt*

$$\begin{aligned} \|V_t\|_2 &= \|\mathcal{I}_t\|_2 && \text{für alle } t \in \mathcal{T}_{\mathcal{I}} \text{ und} \\ \|W_s\|_2 &= \|\mathcal{I}_s\|_2 && \text{für alle } s \in \mathcal{T}_{\mathcal{J}}. \end{aligned}$$

Beweis. Es gilt nach (8.16) und (8.22)

$$\|\mathcal{I}_t\|_2 = \|V_t R_t\|_2 \leq \|V_t\|_2 \|R_t\|_2$$

und da $\|R_t\|_2 \leq 1$, folgt

$$\|\mathcal{I}_t\|_2 \leq \|V_t\|_2. \quad (9.7)$$

Definiere den Vektor $\hat{x} \in \mathbb{R}^{\hat{t}}$ zu $x \in \mathbb{R}^k$ über die Zeilen-Pivot-Indizes $\mu_1, \dots, \mu_k \in \hat{t}$ von ACA, über die R_t definiert ist. Die Komponenten von \hat{x} seien

$$\hat{x}_i := \begin{cases} x_j & \text{falls } i = \mu_j \\ 0 & \text{sonst} \end{cases}$$

für alle $i \in \hat{t}$. Damit gilt $\|\hat{x}\|_2 = \|x\|_2$ und $R_t \hat{x} = x$ und es folgt

$$\begin{aligned} \|V_t\|_2 &= \sup_x \left\{ \frac{\|V_t x\|_2}{\|x\|_2} \right\} = \sup_x \left\{ \frac{\|V_t R_t \hat{x}\|_2}{\|x\|_2} \right\} = \sup_x \left\{ \frac{\|\mathcal{I}_t \hat{x}\|_2}{\|x\|_2} \right\} \\ &\leq \sup_x \left\{ \frac{\|\mathcal{I}_t\|_2 \|\hat{x}\|_2}{\|x\|_2} \right\} = \sup_x \left\{ \frac{\|\mathcal{I}_t\|_2 \|x\|_2}{\|x\|_2} \right\} \\ &= \|\mathcal{I}_t\|_2. \end{aligned}$$

Zusammen mit (9.7) folgt also

$$\|V_t\|_2 = \|\mathcal{I}_t\|_2.$$

Auf analoge Weise kann man auch

$$\|W_s\|_2 = \|\mathcal{I}_s\|_2$$

zeigen. ■

Aus diesem Lemma folgt sofort, dass sich V_t und W_s auch durch die Stabilitätskonstanten abschätzen lassen, denn es gilt

$$\begin{aligned} \|V_t\|_2 &= \|\mathcal{I}_t\|_2 \leq \Lambda_t && \text{für alle } t \in \mathcal{T}_{\mathcal{I}} \text{ und} \\ \|W_s\|_2 &= \|\mathcal{I}_s\|_2 \leq \Lambda_s && \text{für alle } s \in \mathcal{T}_{\mathcal{J}}. \end{aligned} \quad (9.8)$$

Mit dem nächsten Lemma wird gezeigt, dass ebenso die Transfermatrizen auf diese Weise behandelt werden können.

Lemma 9.7 (Abschätzung der Transfermatrizen) *Seien V und W die durch das Green-Hybrid-Verfahren berechneten Clusterbasen mit den durch (8.21) und (8.25) definierten Transfermatrizen. Dann gilt*

$$\begin{aligned} \left\| \begin{pmatrix} E_{t_1} \\ E_{t_2} \end{pmatrix} \right\|_2 &\leq \Lambda_t && \text{für alle } t \in \mathcal{T}_{\mathcal{I}} \setminus \mathcal{L}_{\mathcal{I}} \text{ und} \\ \left\| \begin{pmatrix} F_{s_1} \\ F_{s_2} \end{pmatrix} \right\|_2 &\leq \Lambda_s && \text{für alle } s \in \mathcal{T}_{\mathcal{J}} \setminus \mathcal{L}_{\mathcal{J}}. \end{aligned}$$

Beweis. Mit Lemma 9.5 gilt

$$\left\| \begin{pmatrix} E_{t_1} \\ E_{t_2} \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} I & E_{t_1} \\ I & E_{t_2} \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} R_{t_1} V_{t_1} E_{t_1} \\ R_{t_2} V_{t_2} E_{t_2} \end{pmatrix} \right\|_2.$$

9 Fehleranalyse des Green-Hybrid-Verfahrens

Da

$$\left\| \begin{pmatrix} R_{t_1} & \\ & R_{t_2} \end{pmatrix} \right\|_2 \leq 1$$

folgt weiter

$$\left\| \begin{pmatrix} R_{t_1} & \\ & R_{t_2} \end{pmatrix} \begin{pmatrix} V_{t_1} E_{t_1} \\ V_{t_2} E_{t_2} \end{pmatrix} \right\|_2 \leq \left\| \begin{pmatrix} R_{t_1} & \\ & R_{t_2} \end{pmatrix} \right\|_2 \|V_t\|_2 \leq \|V_t\|_2$$

und mit (9.6) gilt dann die Behauptung

$$\left\| \begin{pmatrix} E_{t_1} \\ E_{t_2} \end{pmatrix} \right\|_2 \leq \|V_t\|_2 \leq \Lambda_t.$$

Analog folgt

$$\left\| \begin{pmatrix} F_{s_1} \\ F_{s_2} \end{pmatrix} \right\|_2 \leq \|W_s\|_2 \leq \Lambda_s. \quad \blacksquare$$

Mit diesen vorbereitenden Lemmas können die entscheidenden Aussagen zur Fehleranalyse des Green-Hybrid-Verfahrens bewiesen werden. Wieder wird nur der Fehler betrachtet, der auf jeweils einem Matrixblock durch die Niedrigrangapproximation mit dem Verfahren entsteht.

Da hier die Schachtelungseigenschaft der Clusterbasen und damit auch die hierarchische Struktur der Clusterbäume eine große Rolle spielen, wird der besseren Lesbarkeit halber für die Beweise eine weitere Bezeichnung eingeführt: Definiere den Teilbaum \mathcal{T}_t des Clusterbaums $\mathcal{T}_{\mathcal{I}}$ mit Wurzel $t \in \mathcal{T}_{\mathcal{I}}$, so dass durch $\text{sons}^*(t)$ genau alle Cluster aus \mathcal{T}_t beschrieben sind, also $\#\mathcal{T}_t = \#\text{sons}^*(t)$ gilt. Mit \mathcal{L}_t seien die Blätter von \mathcal{T}_t bezeichnet.

Satz 9.8 (Fehlerabschätzung für \mathcal{I}_t) *Seien V und W die durch das Green-Hybrid-Verfahren berechneten Clusterbasen und \mathcal{I}_t und \mathcal{I}_s die entsprechenden Interpolationsoperatoren mit den Stabilitätskonstanten Λ_t und Λ_s . Dann gilt für $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+$*

$$\begin{aligned} \|G|_{\hat{t} \times \hat{s}} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}}\|_2 &\leq \sum_{t \in \mathcal{L}_t} (1 + \Lambda_t) (\varepsilon_{green} + \varepsilon_{aca} \|B_{t,s}\|_2) \\ &\quad + \sum_{t \in \mathcal{T}_{\hat{t}} \setminus \mathcal{L}_t} (1 + \Lambda_t) \left(\sum_{t' \in \text{sons}(t)} \Lambda_{t'} \right) (\varepsilon_{green} + \varepsilon_{aca} \|B_{t,s}\|_2). \end{aligned}$$

Beweis. Durch Induktion über $\#\text{sons}^*(t)$, also $\#\mathcal{T}_t$.

Sei $\#\text{sons}^*(t) = 1$, also $\text{sons}^*(t) = \{t\}$ und somit $t \in \mathcal{L}_{\mathcal{I}}$. Da $\mathcal{I}_t C_t = C_t$ gilt

$$\begin{aligned} \|G|_{\hat{t} \times \hat{s}} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}}\|_2 &= \|G|_{\hat{t} \times \hat{s}} - C_t D_t^* B_{t,s}^* + \mathcal{I}_t C_t D_t^* B_{t,s}^* - \mathcal{I}_t G|_{\hat{t} \times \hat{s}}\|_2 \\ &\leq \|G|_{\hat{t} \times \hat{s}} - C_t D_t^* B_{t,s}^*\|_2 + \|\mathcal{I}_t(C_t D_t^* B_{t,s}^* - G|_{\hat{t} \times \hat{s}})\|_2. \end{aligned}$$

Mit (9.6) folgt

$$\begin{aligned}
 \|G|_{\hat{t} \times \hat{s}} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}}\|_2 &\leq \|G|_{\hat{t} \times \hat{s}} - C_t D_t^* B_{t,s}^*\|_2 + \Lambda_t \|C_t D_t^* B_{t,s}^* - G|_{\hat{t} \times \hat{s}}\|_2 \\
 &= (1 + \Lambda_t) \|G|_{\hat{t} \times \hat{s}} - C_t D_t^* B_{t,s}^*\|_2 \\
 &= (1 + \Lambda_t) \|G|_{\hat{t} \times \hat{s}} - A_t B_{t,s}^* + A_t B_{t,s}^* - C_t D_t^* B_{t,s}^*\|_2 \\
 &\leq (1 + \Lambda_t) (\|G|_{\hat{t} \times \hat{s}} - A_t B_{t,s}^*\|_2 + \|(A_t - C_t D_t^*) B_{t,s}^*\|_2).
 \end{aligned}$$

Nach (9.1) und (9.2) gilt dann

$$\|G|_{\hat{t} \times \hat{s}} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}}\|_2 \leq (1 + \Lambda_t) (\varepsilon_{green} + \varepsilon_{aca} \|B_{t,s}\|_2).$$

Da für $\text{sons}^*(t) = \{t\}$ hier $\mathcal{T}_t \setminus \mathcal{L}_t = \emptyset$ und $\mathcal{L}_t = \{t\}$ gilt, ist damit der Induktionsanfang bewiesen.

Gelte die Behauptung für $\#\text{sons}^*(t) \leq n$.

Sei $\#\text{sons}^*(t) = n+1$, also $t \notin \mathcal{L}_{\mathcal{T}}$. Für $t_1, t_2 \in \text{sons}(t)$ gilt $\#\text{sons}^*(t_1), \#\text{sons}^*(t_2) \leq n$, sowie $\mathcal{T}_t = \mathcal{T}_{t_1} \cup \mathcal{T}_{t_2} \cup \{t\}$ und $\mathcal{L}_t = \mathcal{L}_{t_1} \cup \mathcal{L}_{t_2}$. Es gilt:

$$\begin{aligned}
 \|G|_{\hat{t} \times \hat{s}} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}}\|_2 &= \left\| \begin{pmatrix} G|_{\hat{t}_1 \times \hat{s}} \\ G|_{\hat{t}_2 \times \hat{s}} \end{pmatrix} - \begin{pmatrix} \mathcal{I}_{t_1} G|_{\hat{t}_1 \times \hat{s}} \\ \mathcal{I}_{t_2} G|_{\hat{t}_2 \times \hat{s}} \end{pmatrix} + \begin{pmatrix} \mathcal{I}_{t_1} G|_{\hat{t}_1 \times \hat{s}} \\ \mathcal{I}_{t_2} G|_{\hat{t}_2 \times \hat{s}} \end{pmatrix} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}} \right\|_2 \\
 &\leq \|G|_{\hat{t}_1 \times \hat{s}} - \mathcal{I}_{t_1} G|_{\hat{t}_1 \times \hat{s}}\|_2 + \|G|_{\hat{t}_2 \times \hat{s}} - \mathcal{I}_{t_2} G|_{\hat{t}_2 \times \hat{s}}\|_2 \\
 &\quad + \left\| \begin{pmatrix} \mathcal{I}_{t_1} G|_{\hat{t}_1 \times \hat{s}} \\ \mathcal{I}_{t_2} G|_{\hat{t}_2 \times \hat{s}} \end{pmatrix} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}} \right\|_2.
 \end{aligned} \tag{9.9}$$

Für den dritten Summanden davon gilt weiter

$$\begin{aligned}
 \left\| \begin{pmatrix} \mathcal{I}_{t_1} G|_{\hat{t}_1 \times \hat{s}} \\ \mathcal{I}_{t_2} G|_{\hat{t}_2 \times \hat{s}} \end{pmatrix} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}} \right\|_2 &= \left\| \begin{pmatrix} V_{t_1} R_{t_1} G|_{\hat{t}_1 \times \hat{s}} \\ V_{t_2} R_{t_2} G|_{\hat{t}_2 \times \hat{s}} \end{pmatrix} - V_t R_t G|_{\hat{t} \times \hat{s}} \right\|_2 \\
 &= \left\| \begin{pmatrix} V_{t_1} & \\ & V_{t_2} \end{pmatrix} \begin{pmatrix} R_{t_1} G|_{\hat{t}_1 \times \hat{s}} \\ R_{t_2} G|_{\hat{t}_2 \times \hat{s}} \end{pmatrix} - \begin{pmatrix} V_{t_1} & \\ & V_{t_2} \end{pmatrix} \begin{pmatrix} E_{t_1} \\ E_{t_2} \end{pmatrix} R_t G|_{\hat{t} \times \hat{s}} \right\|_2 \\
 &\leq \left\| \begin{pmatrix} V_{t_1} & \\ & V_{t_2} \end{pmatrix} \right\|_2 \left\| \begin{pmatrix} R_{t_1} G|_{\hat{t}_1 \times \hat{s}} \\ R_{t_2} G|_{\hat{t}_2 \times \hat{s}} \end{pmatrix} - \begin{pmatrix} E_{t_1} \\ E_{t_2} \end{pmatrix} R_t G|_{\hat{t} \times \hat{s}} \right\|_2.
 \end{aligned}$$

Mit (9.8) folgt

$$\begin{aligned}
 &\left\| \begin{pmatrix} \mathcal{I}_{t_1} G|_{\hat{t}_1 \times \hat{s}} \\ \mathcal{I}_{t_2} G|_{\hat{t}_2 \times \hat{s}} \end{pmatrix} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}} \right\|_2 \\
 &\leq (\Lambda_{t_1} + \Lambda_{t_2}) \left\| \begin{pmatrix} R_{t_1} G|_{\hat{t}_1 \times \hat{s}} \\ R_{t_2} G|_{\hat{t}_2 \times \hat{s}} \end{pmatrix} - \widehat{C}_t (\widehat{R}_t \widehat{C}_t)^{-1} R_t G|_{\hat{t} \times \hat{s}} \right\|_2 \\
 &\leq (\Lambda_{t_1} + \Lambda_{t_2}) \left[\left\| \begin{pmatrix} R_{t_1} G|_{\hat{t}_1 \times \hat{s}} \\ R_{t_2} G|_{\hat{t}_2 \times \hat{s}} \end{pmatrix} - \widehat{A}_t B_{t,s}^* \right\|_2 + \left\| \widehat{A}_t B_{t,s}^* - \widehat{C}_t (\widehat{R}_t \widehat{C}_t)^{-1} R_t G|_{\hat{t} \times \hat{s}} \right\|_2 \right] \\
 &= (\Lambda_{t_1} + \Lambda_{t_2}) \left[\left\| \begin{pmatrix} R_{t_1} & \\ & R_{t_2} \end{pmatrix} (G|_{\hat{t} \times \hat{s}} - A_t B_{t,s}^*) \right\|_2 + \left\| \widehat{A}_t B_{t,s}^* - \widehat{C}_t (\widehat{R}_t \widehat{C}_t)^{-1} R_t G|_{\hat{t} \times \hat{s}} \right\|_2 \right]
 \end{aligned}$$

9 Fehleranalyse des Green-Hybrid-Verfahrens

und da

$$\left\| \begin{pmatrix} R_{t_1} & \\ & R_{t_2} \end{pmatrix} \right\|_2 \leq 1$$

und mit (9.1) gilt

$$\begin{aligned} & \left\| \begin{pmatrix} \mathcal{I}_{t_1} G|_{\hat{t}_1 \times \hat{s}} \\ \mathcal{I}_{t_2} G|_{\hat{t}_2 \times \hat{s}} \end{pmatrix} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}} \right\|_2 \\ & \leq (\Lambda_{t_1} + \Lambda_{t_2}) \left[\|G|_{\hat{t} \times \hat{s}} - A_t B_{t,s}^*\|_2 + \left\| \hat{A}_t B_{t,s}^* - \hat{C}_t (\hat{R}_t \hat{C}_t)^{-1} R_t G|_{\hat{t} \times \hat{s}} \right\|_2 \right] \\ & \leq (\Lambda_{t_1} + \Lambda_{t_2}) \left[\varepsilon_{green} + \left\| \hat{A}_t B_{t,s}^* - \hat{C}_t \hat{D}_t^* B_{t,s}^* \right\|_2 + \left\| \hat{C}_t \hat{D}_t^* B_{t,s}^* - \hat{C}_t (\hat{R}_t \hat{C}_t)^{-1} R_t G|_{\hat{t} \times \hat{s}} \right\|_2 \right] \\ & \leq (\Lambda_{t_1} + \Lambda_{t_2}) \left[\varepsilon_{green} + \left\| \hat{A}_t - \hat{C}_t \hat{D}_t^* \right\|_2 \|B_{t,s}\|_2 \right. \\ & \quad \left. + \left\| \hat{C}_t (\hat{R}_t \hat{C}_t)^{-1} \hat{R}_t \hat{C}_t \hat{D}_t^* B_{t,s}^* - \hat{C}_t (\hat{R}_t \hat{C}_t)^{-1} R_t G|_{\hat{t} \times \hat{s}} \right\|_2 \right]. \end{aligned}$$

Darauf kann dann (9.2) angewendet werden und es folgt

$$\begin{aligned} & \left\| \begin{pmatrix} \mathcal{I}_{t_1} G|_{\hat{t}_1 \times \hat{s}} \\ \mathcal{I}_{t_2} G|_{\hat{t}_2 \times \hat{s}} \end{pmatrix} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}} \right\|_2 \\ & \leq (\Lambda_{t_1} + \Lambda_{t_2}) \left[\varepsilon_{green} + \varepsilon_{aca} \|B_{t,s}\|_2 \right. \\ & \quad \left. + \left\| \hat{C}_t (\hat{R}_t \hat{C}_t)^{-1} \hat{R}_t \right\|_2 \left\| \hat{C}_t \hat{D}_t^* B_{t,s}^* - \begin{pmatrix} R_{t_1} G|_{\hat{t}_1 \times \hat{s}} \\ R_{t_2} G|_{\hat{t}_2 \times \hat{s}} \end{pmatrix} \right\|_2 \right] \\ & \leq (\Lambda_{t_1} + \Lambda_{t_2}) \left[\varepsilon_{green} + \varepsilon_{aca} \|B_{t,s}\|_2 \right. \\ & \quad \left. + \left\| \begin{pmatrix} E_{t_1} \\ E_{t_2} \end{pmatrix} \right\|_2 \left\| \hat{R}_t \right\|_2 \left\| \hat{C}_t \hat{D}_t^* B_{t,s}^* - \hat{A} B_{t,s}^* + \hat{A} B_{t,s}^* - \begin{pmatrix} R_{t_1} G|_{\hat{t}_1 \times \hat{s}} \\ R_{t_2} G|_{\hat{t}_2 \times \hat{s}} \end{pmatrix} \right\|_2 \right], \end{aligned}$$

welches mit Lemma 9.7 und wegen $\|\hat{R}_t\|_2 \leq 1$ weiter abzuschätzen ist durch

$$\begin{aligned} & \left\| \begin{pmatrix} \mathcal{I}_{t_1} G|_{\hat{t}_1 \times \hat{s}} \\ \mathcal{I}_{t_2} G|_{\hat{t}_2 \times \hat{s}} \end{pmatrix} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}} \right\|_2 \\ & \leq (\Lambda_{t_1} + \Lambda_{t_2}) \left[\varepsilon_{green} + \varepsilon_{aca} \|B_{t,s}\|_2 \right. \\ & \quad \left. + \Lambda_t \left(\left\| \hat{C}_t \hat{D}_t^* - \hat{A} \right\|_2 \|B_{t,s}\|_2 + \left\| \begin{pmatrix} R_{t_1} & \\ & R_{t_2} \end{pmatrix} \right\|_2 \|A_t B_{t,s}^* - G|_{\hat{t} \times \hat{s}}\|_2 \right) \right] \end{aligned}$$

und mit (9.1) und (9.2) folgt schließlich

$$\begin{aligned} & \left\| \begin{pmatrix} \mathcal{I}_{t_1} G|_{\hat{t}_1 \times \hat{s}} \\ \mathcal{I}_{t_2} G|_{\hat{t}_2 \times \hat{s}} \end{pmatrix} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}} \right\|_2 \leq (\Lambda_{t_1} + \Lambda_{t_2}) \left[\varepsilon_{green} + \varepsilon_{aca} \|B_{t,s}\|_2 \right. \\ & \quad \left. + \Lambda_t (\varepsilon_{green} + \varepsilon_{aca} \|B_{t,s}\|_2) \right] \\ & = (\Lambda_{t_1} + \Lambda_{t_2}) (1 + \Lambda_t) (\varepsilon_{green} + \varepsilon_{aca} \|B_{t,s}\|_2). \end{aligned}$$

Dieses Ergebnis eingesetzt in (9.9) führt zu

$$\begin{aligned} \|G|_{\hat{t} \times \hat{s}} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}}\|_2 & = \|G|_{\hat{t}_1 \times \hat{s}} - \mathcal{I}_{t_1} G|_{\hat{t}_1 \times \hat{s}}\|_2 + \|G|_{\hat{t}_2 \times \hat{s}} - \mathcal{I}_{t_2} G|_{\hat{t}_2 \times \hat{s}}\|_2 \\ & \quad + (\Lambda_{t_1} + \Lambda_{t_2}) (1 + \Lambda_t) (\varepsilon_{green} + \varepsilon_{aca} \|B_{t,s}\|_2) \end{aligned}$$

und mit Anwendung der Induktionsvoraussetzung auf die ersten beiden Summanden folgt

$$\begin{aligned}
 \|G|_{\hat{t} \times \hat{s}} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}}\|_2 &\leq \sum_{r \in \mathcal{L}_{t_1}} (1 + \Lambda_r) (\varepsilon_{green} + \varepsilon_{aca} \|B_{r,s}\|_2) \\
 &\quad + \sum_{r \in \mathcal{T}_{t_1} \setminus \mathcal{L}_{t_1}} (1 + \Lambda_r) \left(\sum_{r' \in \text{sons}(r)} \Lambda_{r'} \right) (\varepsilon_{green} + \varepsilon_{aca} \|B_{r,s}\|_2) \\
 &\quad + \sum_{r \in \mathcal{L}_{t_2}} (1 + \Lambda_r) (\varepsilon_{green} + \varepsilon_{aca} \|B_{r,s}\|_2) \\
 &\quad + \sum_{r \in \mathcal{T}_{t_2} \setminus \mathcal{L}_{t_2}} (1 + \Lambda_r) \left(\sum_{r' \in \text{sons}(r)} \Lambda_{r'} \right) (\varepsilon_{green} + \varepsilon_{aca} \|B_{r,s}\|_2) \\
 &\quad + (1 + \Lambda_t) (\Lambda_{t_1} + \Lambda_{t_2}) (\varepsilon_{green} + \varepsilon_{aca} \|B_{t,s}\|_2).
 \end{aligned}$$

Da $\mathcal{L}_{t_1} \cup \mathcal{L}_{t_2} = \mathcal{L}_t$ und $\mathcal{T}_{t_1} \setminus \mathcal{L}_{t_1} \cup \mathcal{T}_{t_2} \setminus \mathcal{L}_{t_2} \cup \{t\} = \mathcal{T}_t \setminus \mathcal{L}_t$ gilt, lässt sich dies zusammenfassen zu

$$\begin{aligned}
 \|G|_{\hat{t} \times \hat{s}} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}}\|_2 &\leq \sum_{t \in \mathcal{L}_t} (1 + \Lambda_t) (\varepsilon_{green} + \varepsilon_{aca} \|B_{t,s}\|_2) \\
 &\quad + \sum_{t \in \mathcal{T}_t \setminus \mathcal{L}_t} (1 + \Lambda_t) \left(\sum_{t' \in \text{sons}(t)} \Lambda_{t'} \right) (\varepsilon_{green} + \varepsilon_{aca} \|B_{t,s}\|_2).
 \end{aligned}$$

■

Wie zu erwarten war, beinhaltet diese Fehlerschranke die Fehler der Green-Quadratur ε_{green} , die Fehlertoleranz von ACA ε_{aca} und die Stabilitätskonstanten der Interpolationsoperatoren \mathcal{I}_t . Die Norm von $B_{t,s}$ kann dabei durch Konstanten abgeschätzt werden, wie in Lemma 9.1 gezeigt wurde, so dass die Steuerung der Konvergenz an ε_{green} und ε_{aca} festgemacht werden kann.

Analog zu Satz 9.8 kann auch der entsprechende Satz für \mathcal{I}_s formuliert werden.

Satz 9.9 (Fehlerabschätzung für \mathcal{I}_s) *Seien V und W die durch das Green-Hybrid-Verfahren berechneten Clusterbasen und \mathcal{I}_t und \mathcal{I}_s die entsprechenden Interpolationsoperatoren mit den Stabilitätskonstanten Λ_t und Λ_s . Dann gilt für $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+$*

$$\begin{aligned}
 \|G|_{\hat{t} \times \hat{s}} - G|_{\hat{t} \times \hat{s}} \mathcal{I}_s^*\|_2 &\leq \sum_{s \in \mathcal{L}_s} (1 + \Lambda_s) (\varepsilon_{green} + \varepsilon_{aca} \|A_{t,s}\|_2) \\
 &\quad + \sum_{s \in \mathcal{T}_s \setminus \mathcal{L}_s} (1 + \Lambda_s) \left(\sum_{s' \in \text{sons}(s)} \Lambda_{s'} \right) (\varepsilon_{green} + \varepsilon_{aca} \|A_{t,s}\|_2).
 \end{aligned}$$

Beweis. Analog zum Beweis von Satz 9.8. ■

Der Gesamtfehler setzt sich einfach aus den Fehlerschranken aus den Sätzen 9.8 und 9.9 zusammen.

Satz 9.10 (Gesamtfehler des Green-Hybrid-Verfahrens) Seien V und W die durch das Green-Hybrid-Verfahren berechneten Clusterbasen und \mathcal{I}_t und \mathcal{I}_s die entsprechenden Interpolationsoperatoren mit den Stabilitätskonstanten Λ_t und Λ_s . Dann gilt für alle $b = (t, s) \in \mathcal{L}_{\mathcal{I} \times \mathcal{J}}^+$

$$\|G|_{\hat{t} \times \hat{s}} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}} \mathcal{I}_s^*\|_2 \leq \|G|_{\hat{t} \times \hat{s}} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}}\|_2 + \Lambda_t \|G|_{\hat{t} \times \hat{s}} - G|_{\hat{t} \times \hat{s}} \mathcal{I}_s^*\|_2.$$

Beweis. Es gilt mit (9.6)

$$\begin{aligned} \|G|_{\hat{t} \times \hat{s}} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}} \mathcal{I}_s^*\|_2 &= \|G|_{\hat{t} \times \hat{s}} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}} + \mathcal{I}_t G|_{\hat{t} \times \hat{s}} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}} \mathcal{I}_s^*\|_2 \\ &\leq \|G|_{\hat{t} \times \hat{s}} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}}\|_2 + \|\mathcal{I}_t\|_2 \|G|_{\hat{t} \times \hat{s}} - G|_{\hat{t} \times \hat{s}} \mathcal{I}_s^*\|_2 \\ &\leq \|G|_{\hat{t} \times \hat{s}} - \mathcal{I}_t G|_{\hat{t} \times \hat{s}}\|_2 + \Lambda_t \|G|_{\hat{t} \times \hat{s}} - G|_{\hat{t} \times \hat{s}} \mathcal{I}_s^*\|_2. \end{aligned}$$

■

Wie bereits erwähnt, hängt die Konvergenz im wesentlichen an ε_{green} und ε_{aca} und damit daran, wie geschickt die einzelnen Eingabeparameter für den Algorithmus gesetzt werden. Wie genau sich die Wahl der verschiedenen Parameter m , ℓ , δ der Green-Quadratur und die Wahl der Fehlertoleranz ε_{aca} in der praktischen Anwendung auf die Konvergenz auswirkt, wird in Kapitel 10 untersucht.

10 Numerische Experimente für das Hybrid-Verfahren

In diesem Kapitel wird die theoretische Analyse des Green-Hybrid-Verfahrens aus dem vorangegangenen Kapitel 9 um Analysen der praktischen Anwendung ergänzt.

Wie zuvor die Green-Quadratur-Methode ist auch das Green-Hybrid-Verfahren im Rahmen der Programm-Bibliothek *H2Lib* implementiert worden.

Zur Anwendung für \mathcal{H} -Matrizen gibt es in Abschnitt 10.1 eine Analyse im zweidimensionalen sowie im dreidimensionalen Fall und ein paar relevante Vergleiche mit anderen Verfahren. Dasselbe behandelt Abschnitt 10.2 im Falle der Anwendung auf \mathcal{H}^2 -Matrizen.

Alle Experimente zum Hybrid-Verfahren wurden unter denselben technischen Rahmenbedingungen wie die Experimente zum Green-Quadratur-Verfahren durchgeführt.

10.1 Hybrid-Verfahren für \mathcal{H} -Matrizen

In Abschnitt 10.1.1 wird der Einfluss der verschiedenen Parameter des Hybrid-Verfahrens auf seine Effizienz im zweidimensionalen Fall untersucht, um mit einer geeigneten Wahl der Parameter Vergleiche zu anderen Verfahren anzustellen in Abschnitt 10.1.2.

Ein ähnliches Vorgehen für den dreidimensionalen Fall findet sich in den Abschnitten 10.1.3 mit der Effizienz-Analyse und 10.1.4 mit dem Vergleich zu anderen Methoden.

10.1.1 2D-Fall Analyse

In den folgenden Experimenten zum zweidimensionalen Fall wird zunächst wie bereits in Abschnitt 7.1.1, die Galerkin-Approximation der Kreis-Geometrie Ω behandelt.

Im ersten grundlegenden Experiment wird der Einfluss der wichtigsten Parameter der beiden Einzel-Verfahren betrachtet: die Green-Quadraturordnung m und die Fehlertoleranz ε_{aca} der adaptiven Kreuzapproximation.

Abbildung 10.1 zeigt dazu die Ergebnisse auf dem Kreis mit $n = 65\,536$ Freiheitsgraden. In der linken Teilabbildung werden für verschiedene ε_{aca} die Quadraturordnungen m erhöht, während anders herum in der rechten Teilabbildung für verschiedene m die Toleranz ε_{aca} erhöht wird. Dabei wird deutlich, dass das Variieren der Fehlertoleranz ε_{aca} einen wesentlich stärkeren Einfluss auf den vom Green-Hybrid-Verfahren produzierten Fehler hat, als die Veränderung der Green-Quadraturordnung m .

Außerdem kann man sehen, dass eine Quadraturordnung von $m = 4$ ausreicht, um gute Genauigkeiten zu erreichen, und dass mit einer weiteren Erhöhung von m nur unwesentlich mehr erreicht werden kann.

10 Numerische Experimente für das Hybrid-Verfahren

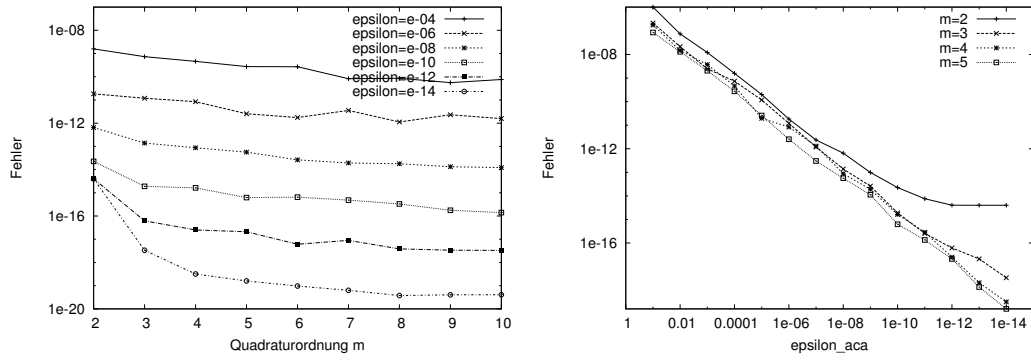


Abbildung 10.1: Kreis-Geometrie, $n = 65\,536$: Green-Hybrid-Verfahren, $\delta = 0.5 \text{ diam}$, $\ell = 1$, Rechteck-Parametrisierung, maximale Blattgröße $clf = 64$, Vergleich der Quadraturordnungen m (links) und der Fehlertoleranzen für ACA ε_{aca} (rechts)

Aus diesem Grund ist für die folgenden Tabellen der Parameter $m = 4$ fest gewählt. Die Tabellen zeigen die Entwicklung des Matrixapproximationsfehlers $\|G - \tilde{G}\|_2$ für verschiedene ACA-Fehlertoleranzen ε_{aca} oder die für den Aufbau der \mathcal{H} -Matrix benötigte Zeit bei verschiedenen Freiheitsgraden n .

| n | ε_{aca} | ε_{aca} | ε_{aca} | ε_{aca} | ε_{aca} | ε_{aca} | ε_{aca} |
|--------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| | $= 10^{-2}$ | $= 10^{-4}$ | $= 10^{-6}$ | $= 10^{-8}$ | $= 10^{-10}$ | $= 10^{-12}$ | $= 10^{-14}$ |
| 1024 | 4.68 ₋₀₇ | 1.33 ₋₀₈ | 1.41 ₋₁₀ | 4.93 ₋₁₃ | 8.72 ₋₁₅ | 2.57 ₋₁₆ | 2.64 ₋₁₈ |
| 2048 | 3.07 ₋₀₇ | 7.18 ₋₀₉ | 1.01 ₋₁₀ | 1.72 ₋₁₂ | 1.29 ₋₁₄ | 2.02 ₋₁₆ | 3.51 ₋₁₈ |
| 4096 | 2.06 ₋₀₇ | 5.71 ₋₀₉ | 8.91 ₋₁₁ | 9.93 ₋₁₃ | 9.15 ₋₁₅ | 1.32 ₋₁₆ | 3.03 ₋₁₈ |
| 8192 | 1.10 ₋₀₇ | 3.24 ₋₀₉ | 4.61 ₋₁₁ | 5.53 ₋₁₃ | 5.95 ₋₁₅ | 8.62 ₋₁₇ | 2.37 ₋₁₈ |
| 16384 | 6.16 ₋₀₈ | 1.62 ₋₀₉ | 3.42 ₋₁₁ | 4.60 ₋₁₃ | 3.62 ₋₁₅ | 5.31 ₋₁₇ | 1.18 ₋₁₈ |
| 32768 | 2.86 ₋₀₈ | 8.18 ₋₁₀ | 1.35 ₋₁₁ | 1.42 ₋₁₃ | 3.65 ₋₁₅ | 4.13 ₋₁₇ | 5.82 ₋₁₉ |
| 65536 | 1.49 ₋₀₈ | 4.60 ₋₁₀ | 8.38 ₋₁₂ | 8.71 ₋₁₄ | 1.63 ₋₁₅ | 2.52 ₋₁₇ | 3.18 ₋₁₉ |
| 131072 | 7.20 ₋₀₉ | 2.27 ₋₁₀ | 4.37 ₋₁₂ | 4.50 ₋₁₄ | 1.44 ₋₁₅ | 1.83 ₋₁₇ | 1.67 ₋₁₉ |

Tabelle 10.1: Approximationsfehler: $\|G - \tilde{G}\|_2$ für $m = 4$ und $\ell = 1$ (SLP)

Tabelle 10.1 zeigt dabei die Fehlerentwicklung für das Einfachschichtpotential für $\ell = 1$ und Tabelle 10.2 für $\ell = 2$. Die jeweils zugehörigen Zeiten sind in den Tabellen 10.3 und 10.4 zu finden.

Anhand dieser Tabellen kann bereits im zweidimensionalen Fall ein Vorteil gegenüber dem einfachen Green-Quadratur-Verfahren erkannt werden: Um eine bestimmte Genauigkeit zu erreichen, ist eine deutlich kürzere Berechnungszeit nötig. Dieser Vorteil sowie der geringere Speicherbedarf im Vergleich zur Green-Quadratur wird in Abschnitt 10.1.2 wieder aufgegriffen.

10.1 Hybrid-Verfahren für \mathcal{H} -Matrizen

| n | ε_{aca} = 10^{-2} | ε_{aca} = 10^{-4} | ε_{aca} = 10^{-6} | ε_{aca} = 10^{-8} | ε_{aca} = 10^{-10} | ε_{aca} = 10^{-12} | ε_{aca} = 10^{-14} |
|--------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 1024 | 1.91 ₋₀₇ | 2.55 ₋₀₉ | 1.08 ₋₁₁ | 1.16 ₋₁₃ | 6.92 ₋₁₅ | 8.28 ₋₁₈ | 4.25 ₋₁₉ |
| 2048 | 1.83 ₋₀₇ | 1.50 ₋₀₉ | 1.20 ₋₁₁ | 4.21 ₋₁₃ | 3.97 ₋₁₅ | 3.18 ₋₁₇ | 3.33 ₋₁₉ |
| 4096 | 1.06 ₋₀₇ | 8.66 ₋₁₀ | 8.63 ₋₁₂ | 1.57 ₋₁₃ | 1.96 ₋₁₅ | 2.31 ₋₁₇ | 1.69 ₋₁₉ |
| 8192 | 5.99 ₋₀₈ | 4.29 ₋₁₀ | 3.90 ₋₁₂ | 1.61 ₋₁₃ | 1.10 ₋₁₅ | 1.41 ₋₁₇ | 1.68 ₋₁₉ |
| 16384 | 3.14 ₋₀₈ | 2.29 ₋₁₀ | 2.10 ₋₁₂ | 7.62 ₋₁₄ | 7.09 ₋₁₆ | 1.31 ₋₁₇ | 1.03 ₋₁₉ |
| 32768 | 1.56 ₋₀₈ | 1.16 ₋₁₀ | 1.00 ₋₁₂ | 2.37 ₋₁₄ | 3.01 ₋₁₆ | 5.71 ₋₁₈ | 8.63 ₋₂₀ |
| 65536 | 8.00 ₋₀₉ | 6.26 ₋₁₁ | 4.79 ₋₁₃ | 9.69 ₋₁₅ | 1.56 ₋₁₆ | 2.12 ₋₁₈ | 5.38 ₋₂₀ |
| 131072 | 4.43 ₋₀₉ | 3.21 ₋₁₁ | 2.54 ₋₁₃ | 5.38 ₋₁₅ | 9.99 ₋₁₇ | 2.42 ₋₁₈ | 2.64 ₋₂₀ |

Tabelle 10.2: Approximationsfehler: $\|G - \tilde{G}\|_2$ für $m = 4$ und $\ell = 2$ (SLP)

| n | ε_{aca} = 10^{-2} | ε_{aca} = 10^{-4} | ε_{aca} = 10^{-6} | ε_{aca} = 10^{-8} | ε_{aca} = 10^{-10} | ε_{aca} = 10^{-12} | ε_{aca} = 10^{-14} |
|--------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 1024 | 0.1 | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 |
| 2048 | 0.3 | 0.3 | 0.3 | 0.4 | 0.4 | 0.4 | 0.5 |
| 4096 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 | 1.1 |
| 8192 | 1.2 | 1.5 | 1.7 | 1.9 | 2.2 | 2.4 | 2.6 |
| 16384 | 2.5 | 3.2 | 3.7 | 4.3 | 4.8 | 5.4 | 5.8 |
| 32768 | 5.6 | 7.2 | 8.3 | 9.6 | 10.9 | 12.2 | 13.0 |
| 65536 | 12.0 | 15.3 | 18.0 | 21.2 | 23.7 | 26.5 | 28.8 |
| 131072 | 26.2 | 33.7 | 39.1 | 46.0 | 50.8 | 57.1 | 62.3 |

Tabelle 10.3: Aufbau der \mathcal{H} -Matrix in Sekunden für $m = 4$ und $\ell = 1$ (SLP)

| n | ε_{aca} = 10^{-2} | ε_{aca} = 10^{-4} | ε_{aca} = 10^{-6} | ε_{aca} = 10^{-8} | ε_{aca} = 10^{-10} | ε_{aca} = 10^{-12} | ε_{aca} = 10^{-14} |
|--------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 1024 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| 2048 | 0.3 | 0.4 | 0.4 | 0.5 | 0.5 | 0.6 | 0.6 |
| 4096 | 0.7 | 0.8 | 1.0 | 1.1 | 1.2 | 1.3 | 1.5 |
| 8192 | 1.5 | 1.9 | 2.2 | 2.5 | 2.8 | 3.1 | 3.4 |
| 16384 | 3.4 | 4.2 | 5.1 | 5.7 | 6.4 | 7.1 | 7.8 |
| 32768 | 7.5 | 9.7 | 11.5 | 12.8 | 14.3 | 16.0 | 17.4 |
| 65536 | 25.4 | 21.2 | 25.7 | 28.5 | 32.1 | 35.6 | 38.6 |
| 131072 | 36.2 | 46.2 | 56.9 | 63.5 | 71.7 | 79.9 | 87.1 |

Tabelle 10.4: Aufbau der \mathcal{H} -Matrix in Sekunden für $m = 4$ und $\ell = 2$ (SLP)

Das Hybrid-Verfahren ist natürlich ebenso vorteilhaft auch auf das Doppelschichtpotential anwendbar. Die entsprechende Fehlerentwicklung für $\ell = 1$ ist in Tabelle 10.5 dargestellt und die zugehörigen Zeiten in Tabelle 10.6.

Den direkten Vergleich zwischen SLP und DLP über die Green-Quadraturordnungen

10 Numerische Experimente für das Hybrid-Verfahren

| n | ε_{aca} $= 10^{-2}$ | ε_{aca} $= 10^{-4}$ | ε_{aca} $= 10^{-6}$ | ε_{aca} $= 10^{-8}$ | ε_{aca} $= 10^{-10}$ | ε_{aca} $= 10^{-12}$ | ε_{aca} $= 10^{-14}$ |
|--------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 1024 | 1.83 ₋₀₆ | 3.73 ₋₀₈ | 3.53 ₋₁₀ | 1.37 ₋₁₂ | 3.19 ₋₁₄ | 6.07 ₋₁₆ | 3.96 ₋₁₇ |
| 2048 | 9.24 ₋₀₇ | 1.88 ₋₀₈ | 1.82 ₋₁₀ | 1.71 ₋₁₂ | 4.27 ₋₁₄ | 1.11 ₋₁₅ | 5.08 ₋₁₇ |
| 4096 | 4.67 ₋₀₇ | 1.00 ₋₀₈ | 1.00 ₋₁₀ | 7.76 ₋₁₃ | 2.29 ₋₁₄ | 2.26 ₋₁₅ | 2.74 ₋₁₇ |
| 8192 | 2.28 ₋₀₇ | 4.98 ₋₀₉ | 5.14 ₋₁₁ | 4.67 ₋₁₃ | 1.34 ₋₁₄ | 5.12 ₋₁₆ | 2.59 ₋₁₇ |
| 16384 | 1.29 ₋₀₇ | 2.56 ₋₀₉ | 2.76 ₋₁₁ | 2.47 ₋₁₃ | 6.85 ₋₁₅ | 4.67 ₋₁₆ | 1.61 ₋₁₇ |
| 32768 | 5.69 ₋₀₈ | 1.28 ₋₀₉ | 1.31 ₋₁₁ | 1.69 ₋₁₃ | 3.28 ₋₁₅ | 2.26 ₋₁₆ | 1.03 ₋₁₇ |
| 65536 | 2.84 ₋₀₈ | 6.37 ₋₁₀ | 6.61 ₋₁₂ | 5.40 ₋₁₄ | 2.24 ₋₁₅ | 1.83 ₋₁₆ | 7.30 ₋₁₈ |
| 131072 | 1.46 ₋₀₈ | 3.38 ₋₁₀ | 3.38 ₋₁₂ | 4.11 ₋₁₄ | 8.78 ₋₁₆ | 6.72 ₋₁₇ | 9.07 ₋₁₈ |

Tabelle 10.5: Approximationsfehler: $\|G - \tilde{G}\|_2$ für $m = 4$ und $\ell = 1$ (DLP)

| n | ε_{aca} $= 10^{-2}$ | ε_{aca} $= 10^{-4}$ | ε_{aca} $= 10^{-6}$ | ε_{aca} $= 10^{-8}$ | ε_{aca} $= 10^{-10}$ | ε_{aca} $= 10^{-12}$ | ε_{aca} $= 10^{-14}$ |
|--------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 1024 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| 2048 | 0.1 | 0.1 | 0.1 | 0.2 | 0.2 | 0.2 | 0.2 |
| 4096 | 0.2 | 0.3 | 0.3 | 0.3 | 0.4 | 0.4 | 0.4 |
| 8192 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 0.9 | 1.0 |
| 16384 | 1.3 | 1.4 | 1.6 | 1.7 | 2.1 | 2.3 | 2.3 |
| 32768 | 2.5 | 3.0 | 3.5 | 3.9 | 4.4 | 4.8 | 5.0 |
| 65536 | 5.8 | 6.7 | 7.9 | 8.7 | 9.7 | 10.7 | 11.1 |
| 131072 | 12.4 | 14.9 | 17.1 | 19.1 | 21.3 | 23.7 | 24.6 |

Tabelle 10.6: Aufbau der \mathcal{H} -Matrix in Sekunden für $m = 4$ und $\ell = 1$ (DLP)

m für $n = 65\,536$ Freiheitsgrade zeigt Abbildung 10.2. Wie bei dem ähnlich aufgebauten Vergleich, der für das einfache Green-Quadratur-Verfahren in Kapitel 7 angestellt wurde und in Abbildung 7.2 dargestellt ist, zeigt die linke Teilabbildung der Abbildung 10.2 die Fehlerkurven für $\ell = 1$ über verschiedene Quadraturordnungen m und die rechte Teilabbildung entsprechend für $\ell = 2$.

Anders als bei der Green-Quadratur ist hier zu beobachten, dass die Kurven nur anfangs bei sehr kleinen Quadraturordnungen stark abfallen, später jedoch eine höhere Quadraturordnung kaum noch zu einer Verbesserung des Fehlers beiträgt. Wie bereits festgestellt, reicht auf der Kreis-Geometrie eine Quadraturordnung von $m = 4$ bis $m = 5$ aus.

Abbildung 10.3 zeigt die Zeit in Sekunden, die das Green-Hybrid-Verfahren bei wachsender Problemgröße auf der Kreis-Geometrie pro einzeltem Freiheitsgrad benötigt. Dabei ist in der linken Teilabbildung die Kurve zu SLP und im rechten die zu DLP, jeweils für $\ell = 1$, dargestellt.

Wie schon bei dem Green-Quadratur-Verfahren ist auch hier bei dem Hybrid-Verfahren zu sehen, dass die Kurve zum Einfeldschichtpotential stärker ansteigt, als die zum Doppelschichtpotential. Das liegt daran, dass bei SLP sich der zeitaufwändig zu

10.1 Hybrid-Verfahren für \mathcal{H} -Matrizen

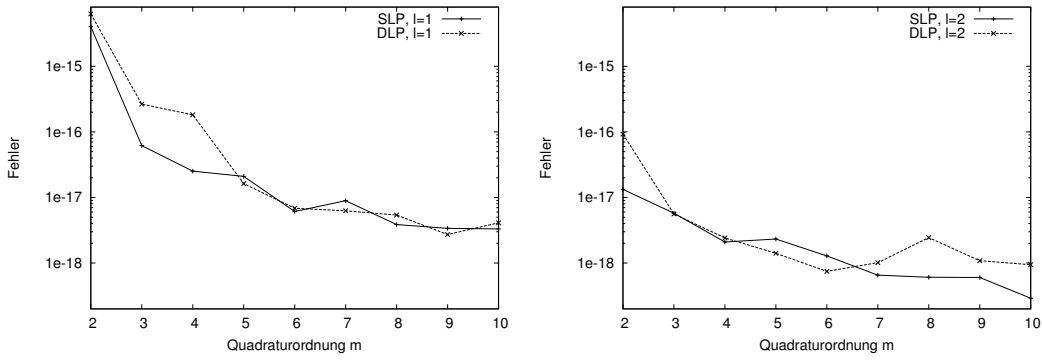


Abbildung 10.2: Kreis-Geometrie, $n = 65\,536$: Green-Hybrid-Verfahren mit Rechteck-Parametrisierung und $\delta = 0.5 \text{ diam}$, maximale Blattgröße $clf = 64$, $l = 1$ (links) und $l = 2$ (rechts) und der Fehlertoleranz $\varepsilon_{aca} = 10^{-12}$

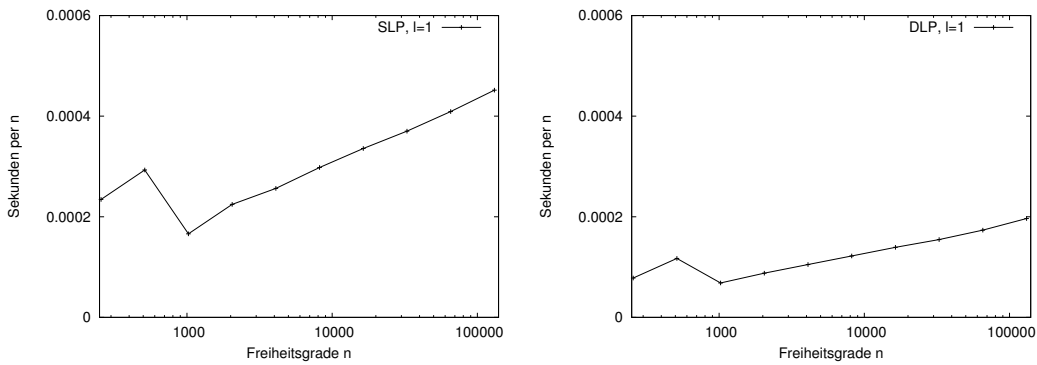


Abbildung 10.3: Kreis-Geometrie: Green-Hybrid-Verfahren mit Rechteck-Parametrisierung und $\delta = 0.5 \text{ diam}$, maximale Blattgröße $clf = 64$, Zeit pro Freiheitsgrad für ein festes $m = 4$ und ein festes $\varepsilon_{aca} = 10^{-12}$ für SLP (links) und DLP (rechts)

berechnende logarithmische Term in der zweidimensionalen Kernfunktion auch in dem hybriden Verfahren bemerkbar macht.

Darüber hinaus ist in diesen Teilabbildungen durch die logarithmisch dargestellten x -Achsen auch das logarithmische Wachstum der Kurven deutlich zu erkennen. Es verhält sich in etwa $\sim \log n$ und bestätigt damit noch einmal die theoretischen Vorhersagen.

Ein Parameter des Hybrid-Verfahrens verdient noch genauere Untersuchung, und zwar δ . Der Abstand δ zwischen der jeweiligen Bounding Box \mathcal{B}_τ und dem Rand Ξ des darum gelegten Normalengebiets Θ wurde für das Green-Quadratur-Verfahren am besten auf $\delta = \frac{1}{2} \text{diam}_{\max}(\mathcal{B}_\tau)$ festgelegt.

Im Fall des hybriden Verfahrens jedoch kann man beobachten, dass mit einem größeren

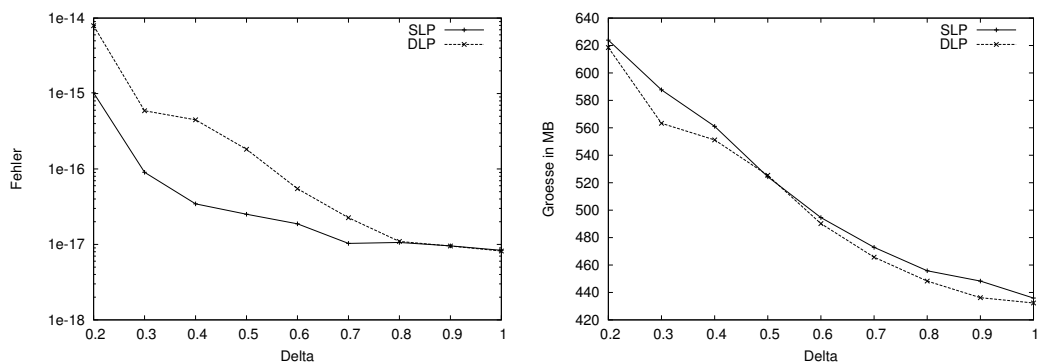


Abbildung 10.4: Kreis-Geometrie, $n = 65\,536$: Green-Hybrid-Verfahren mit Rechteck-Parametrisierung, maximaler Blattgröße $cl_f = 64$ für ein festes $m = 4$, $\ell = 1$ sowie ein festes $\varepsilon_{aca} = 10^{-12}$ und verschiedene Delta δ : Genauigkeit (links) und Matrixgröße (rechts)

δ bessere Ergebnisse erzielt werden. In Abbildung 10.4 sind die Fehlerverläufe und der Speicherbedarf für SLP und DLP auf der Kreis-Geometrie mit $n = 65\,536$ für variierende δ zu sehen. Dabei fällt auf, dass Fehler und Speicherbedarf bei $\delta = \text{diam}_{\max}(\mathcal{B}_\tau)$ für das Einfach- wie auch das Doppelschichtpotential minimal wird.

Anschaulich würde das bedeuten, dass der Rand des Normalengebiets, über den integriert wird, die jeweils andere Bounding Box \mathcal{B}_σ berühren dürfte. Das kommt daher, dass die Zulässigkeitsbedingung (4.13) genutzt wird und damit $\delta = \text{diam}_{\max}(\mathcal{B}_\tau) \leq \text{dist}_{\max}(\mathcal{B}_\tau, \mathcal{B}_\sigma)$ gilt.

Erstaunlicherweise kann man in einzelnen Fällen sogar mit einem noch größeren δ gute Resultate erzielen. Wie diese Tatsache jedoch in die in dieser Arbeit vorgestellte Theorie passt, ist bisher nicht ganz klar und damit Gegenstand weiterer Forschung auf diesem Gebiet.

Nach dem letzten numerischen Experiment in diesem Abschnitt ist also $\delta = \text{diam}_{\max}(\mathcal{B}_\tau)$ für das Hybrid-Verfahren die optimale Wahl. Daher wird δ für die restlichen Experimente im zweidimensionalen Fall mit diesem Wert getestet, so dass die Vergleiche im Abschnitt 10.1.2 mit den jeweils bestmöglichen Ergebnissen der Verfahren durchgeführt werden.

10.1.2 2D-Fall Vergleiche

Die Auswirkungen der einzelnen Parameter des Green-Hybrid-Verfahrens auf dessen Konvergenzverhalten im zweidimensionalen Fall sind ausreichend untersucht worden. In diesem Abschnitt geht es nun um die Vergleichbarkeit mit anderen Verfahren. Nahelegend ist dabei, zunächst mit dem einfachen Green-Quadratur-Verfahren zu vergleichen.

Abbildung 10.5 zeigt das Resultat eines solchen Vergleichs auf der Kreis-Geometrie

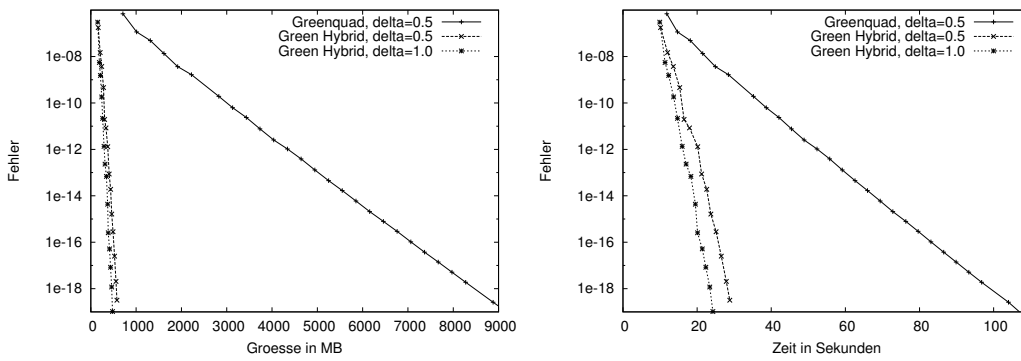


Abbildung 10.5: Kreis-Geometrie, $n = 65\,536$: Green-Quadratur-Methode im Vergleich zum Hybrid-Verfahren, beides mit Rechteck-Parametrisierung, maximaler Blattgröße $clf = 64$ für $\ell = 1$, ein festes $m = 4$ für die Hybrid-Methode und verschiedene δ : Speicherbedarf (links) und Zeiten (rechts)

mit $n = 65\,536$ Freiheitsgraden. Getestet wurde das Green-Quadratur-Verfahren mit $\delta = \frac{1}{2} \text{diam}_{\max}(\mathcal{B}_\tau)$ und das Hybrid-Verfahren einmal ebenfalls mit $\delta = \frac{1}{2} \text{diam}_{\max}(\mathcal{B}_\tau)$ und einmal mit $\delta = \text{diam}_{\max}(\mathcal{B}_\tau)$. Die linke Teilabbildung der Abbildung 10.5 zeigt den Speicherbedarf und die rechte Teilabbildung die benötigte Zeit, um eine gewisse Genauigkeit zu erreichen.

Es ist sofort erkennbar, dass das Hybrid-Verfahren dem einfachen Green-Quadratur-Verfahren weit überlegen ist, ganz besonders in Hinblick auf die Größe der entstandenen \mathcal{H} -Matrix im Speicher. Außerdem ist erneut zu sehen, dass die Wahl von $\delta = \text{diam}_{\max}(\mathcal{B}_\tau)$ für das Hybrid-Verfahren zu noch besseren Resultaten führt.

Also bestätigen die praktischen Tests die theoretischen Erwartungen, dass mit dem Green-Hybrid-Verfahren der Ansatz der Green-Quadratur zu einem sehr viel effizienteren Verfahren gemacht wird.

Wie gut dieses effizientere Hybrid-Verfahren gegen die gängigen Verfahren wie Interpolation, Kreuzapproximation oder die Multipol-Methode aussieht, ist Gegenstand des nächsten Experiments.

Wie bereits in Abschnitt 7.1.2 wird dafür die kompliziertere zweidimensionale Geometrie der Hilbertkurve der Ordnung 8 genutzt, das heißt mit $n = 2^{16} - 1 = 65\,535$ Freiheitsgraden.

Abbildung 10.6 zeigt die Größe der aufgebauten \mathcal{H} -Matrix gegenüber dem relativen Fehler in der linken Teilabbildung und die dazu benötigte Zeit in der rechten Teilabbildung.

Es sind die Kurven für das Hybrid-Verfahren mit einmal der Quadraturordnung $m = 4$ und einmal mit $m = 8$ zu sehen, so wie die Kurven für die Interpolation, die adaptive Kreuzapproximation mit partieller Pivotierung und der Multipol-Methode.

Einerseits ist dabei zu beobachten, dass, während $m = 4$ auf der Kreis-Geometrie ausreichte, auf der komplizierteren Geometrie $m = 8$ nötig ist, damit das Green-Hybrid-

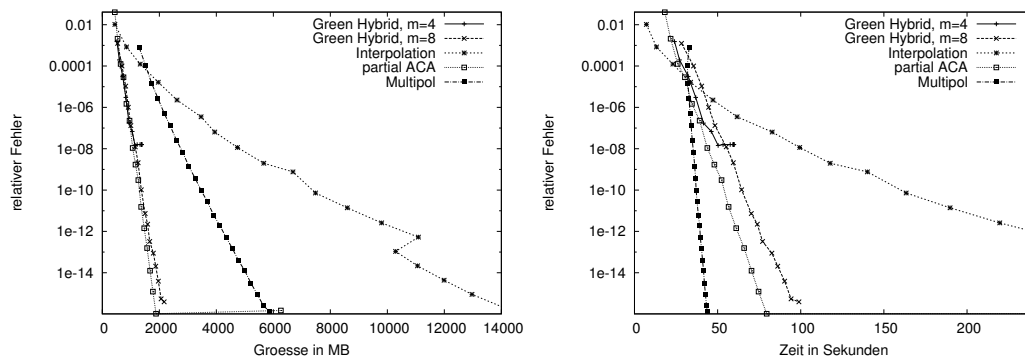


Abbildung 10.6: Hilbertkurven-Geometrie, $n = 65\,535$: Green-Hybrid-Verfahren mit Rechteck-Parametrisierung, maximaler Blattgröße $clf = 64$ für einmal ein festes $m = 4$ und einmal ein festes $m = 8$, $\ell = 1$ und $\delta = \text{diam}$ im Vergleich zur Interpolation, zu ACA mit partieller Pivotierung und der Multipol-Methode: Speicherbedarf (links) und Zeiten (rechts)

Verfahren die volle Genauigkeit erreichen kann.

Andererseits kann das Hybrid-Verfahren mit $m = 8$ mit dem speichergünstigsten Verfahren ACA mithalten, ist dabei aber ein klein wenig langsamer. Dabei sei wieder betont, dass es sich bei dem Green-Hybrid-Verfahren um ein beweisbares Verfahren handelt, welches bei ACA mit partieller Pivotierung nicht der Fall ist.

Gegenüber der Multipol-Methode hat das Hybrid-Verfahren einen wesentlichen Speichervorteil, ist dafür aber nur in etwa halb so schnell.

Das Hybrid-Verfahren ist in Speicherbedarf sowie Zeit der Interpolation weit überlegen.

Damit wurde das gewünschte Ziel erreicht, mit dem Green-Hybrid-Verfahren ein konkurrenzfähiges, effizientes Verfahren zu konstruieren.

Ob sich diese Eigenschaften auch im dreidimensionalen Fall zeigen, ist Gegenstand der folgenden Abschnitte 10.1.3 und 10.1.4.

10.1.3 3D-Fall Analyse

Die Analyse für das Green-Hybrid-Verfahren im dreidimensionalen Fall wird analog zur Vorgehensweise im zweidimensionalen Fall durchgeführt.

Dazu wird als erstes der Einfluss der Parameter der Green-Quadraturordnung m und der Fehlertoleranz ε_{aca} auf der Kugel-Geometrie Ω mit $n = 131\,072$ Freiheitsgraden untersucht. In der linken Teilabbildung von Abbildung 10.7 ist dargestellt, wie die Ergebniskurven für verschiedene Toleranzen ε_{aca} sich bei Variation der Quadraturordnung m verhalten. In der rechten Teilabbildung wird andersherum für die Kurven zu verschiedenen m das ε_{aca} variiert.

Diese Teilabbildungen bestätigen, dass auch im dreidimensionalen Fall der Einfluss von

10.1 Hybrid-Verfahren für \mathcal{H} -Matrizen

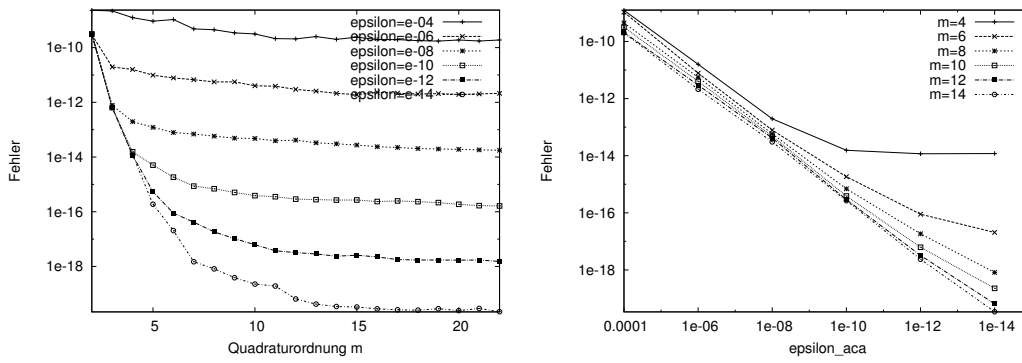


Abbildung 10.7: Kugel-Geometrie, $n = 131\,072$: Green-Hybrid-Verfahren, $\delta = 0.5 \text{ diam}$, $\ell = 1$, Quader-Parametrisierung, maximale Blattgröße $clf = 128$, Vergleich der Quadraturordnungen m (links) und der Fehlertoleranzen für ACA ε_{aca} (rechts)

ε_{aca} auf die Konvergenz des Hybrid-Verfahrens ein größerer ist als der von m . Darüber hinaus ist zu erkennen, dass hier eine Quadraturordnung von $m = 8$ ausreicht, um gute Genauigkeiten zu erhalten.

Also ist $m = 8$ festgesetzt und ε_{aca} variiert in den folgenden Tabellen. Es wird zunächst wieder mit $\delta = \frac{1}{2} \text{diam}_{\max}(\mathcal{B}_\tau)$ getestet, um den theoretischen Überlegungen gerecht zu werden.

| n | $\varepsilon_{aca} = 10^{-2}$ | $\varepsilon_{aca} = 10^{-4}$ | $\varepsilon_{aca} = 10^{-6}$ | $\varepsilon_{aca} = 10^{-8}$ | $\varepsilon_{aca} = 10^{-10}$ | $\varepsilon_{aca} = 10^{-12}$ | $\varepsilon_{aca} = 10^{-14}$ |
|--------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|--------------------------------|--------------------------------|--------------------------------|
| 2048 | 1.07 ₋₀₆ | 1.00 ₋₀₈ | 8.47 ₋₁₁ | — | — | — | — |
| 4608 | 4.98 ₋₀₇ | 5.45 ₋₀₉ | 4.07 ₋₁₁ | 2.50 ₋₁₃ | 4.90 ₋₁₆ | — | — |
| 8192 | 5.87 ₋₀₇ | 4.94 ₋₀₉ | 4.66 ₋₁₁ | 2.57 ₋₁₃ | 2.39 ₋₁₅ | 9.31 ₋₁₇ | 2.24 ₋₁₈ |
| 18432 | 3.09 ₋₀₇ | 2.75 ₋₀₉ | 2.91 ₋₁₁ | 2.29 ₋₁₃ | 2.32 ₋₁₅ | 1.63 ₋₁₆ | 3.90 ₋₁₈ |
| 32768 | 2.04 ₋₀₇ | 1.88 ₋₀₉ | 1.81 ₋₁₁ | 1.54 ₋₁₃ | 2.04 ₋₁₅ | 7.59 ₋₁₇ | 3.21 ₋₁₈ |
| 73728 | 9.48 ₋₀₈ | 8.42 ₋₁₀ | 9.52 ₋₁₂ | 8.81 ₋₁₄ | 1.16 ₋₁₅ | 3.06 ₋₁₇ | 1.29 ₋₁₈ |
| 131072 | 5.98 ₋₀₈ | 4.85 ₋₁₀ | 5.53 ₋₁₂ | 6.06 ₋₁₄ | 6.87 ₋₁₆ | 1.84 ₋₁₇ | 8.16 ₋₁₉ |
| 156800 | 4.77 ₋₀₈ | 4.27 ₋₁₀ | 4.94 ₋₁₂ | 5.32 ₋₁₄ | 5.04 ₋₁₆ | 1.50 ₋₁₇ | 7.40 ₋₁₉ |

Tabelle 10.7: Approximationsfehler: $\|G - \tilde{G}\|_2$ für $m = 8$ und $\ell = 1$ (SLP)

Tabelle 10.7 zeigt die Entwicklung des Matrixapproximationsfehlers $\|G - \tilde{G}\|_2$ für das Einfachschichtpotential und $\ell = 1$ und Tabelle 10.8 liefert die zugehörigen Zeiten.

Es wird an dieser Stelle auf weitere Tests mit $\ell = 2$ oder mehr verzichtet. In vorangegangenen Experimenten hat sich $\ell = 1$ als die beste Wahl herausgestellt, obwohl die Theorie nach Satz 6.10 forderte, dass $\ell > c_0 \geq 1$ für die exponentielle Konvergenz nötig ist und damit zu vermuten war, dass $\ell = 2$ vorteilhafter ausfallen müsste.

10 Numerische Experimente für das Hybrid-Verfahren

| n | ε_{aca} = 10^{-2} | ε_{aca} = 10^{-4} | ε_{aca} = 10^{-6} | ε_{aca} = 10^{-8} | ε_{aca} = 10^{-10} | ε_{aca} = 10^{-12} | ε_{aca} = 10^{-14} |
|--------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 2048 | 2.4 | 2.7 | 2.9 | — | — | — | — |
| 4608 | 10.2 | 12.1 | 14.2 | 15.0 | 15.2 | — | — |
| 8192 | 21.7 | 29.4 | 38.0 | 43.8 | 46.4 | 47.9 | 40.0 |
| 18432 | 67.6 | 95.1 | 132.8 | 166.1 | 191.1 | 209.0 | 219.7 |
| 32768 | 139.7 | 213.4 | 306.5 | 401.8 | 484.1 | 547.4 | 600.9 |
| 73728 | 394.5 | 630.7 | 920.8 | 1257.2 | 1572.7 | 1883.8 | 2187.2 |
| 131072 | 840.6 | 1387.7 | 2038.7 | 2857.2 | 3667.0 | 4404.1 | 5040.8 |
| 156800 | 1047.0 | 1728.9 | 2558.7 | 3609.8 | 4643.9 | 5627.0 | 6587.9 |

Tabelle 10.8: Aufbau der \mathcal{H} -Matrix in Sekunden für $m = 8$ und $\ell = 1$ (SLP)

| n | ε_{aca} = 10^{-2} | ε_{aca} = 10^{-4} | ε_{aca} = 10^{-6} | ε_{aca} = 10^{-8} | ε_{aca} = 10^{-10} | ε_{aca} = 10^{-12} | ε_{aca} = 10^{-14} |
|--------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 2048 | 1.04 ₋₀₆ | 2.24 ₋₀₈ | — | — | — | — | — |
| 4608 | 7.57 ₋₀₇ | 7.86 ₋₀₉ | 3.82 ₋₁₁ | 5.56 ₋₁₃ | — | — | — |
| 8192 | 6.68 ₋₀₇ | 7.85 ₋₀₉ | 6.62 ₋₁₁ | 1.75 ₋₁₁ | 8.53 ₋₁₃ | 2.23 ₋₁₃ | 3.94 ₋₁₅ |
| 18432 | 3.10 ₋₀₇ | 3.92 ₋₀₉ | 3.68 ₋₁₁ | 2.65 ₋₁₂ | 2.66 ₋₁₂ | 2.33 ₋₁₂ | 1.27 ₋₁₂ |
| 32768 | 2.02 ₋₀₇ | 2.60 ₋₀₉ | 2.69 ₋₁₁ | 9.75 ₋₁₃ | 1.00 ₋₁₂ | 1.17 ₋₁₂ | 7.80 ₋₁₃ |
| 73728 | 1.00 ₋₀₇ | 1.08 ₋₀₉ | 2.07 ₋₁₁ | 1.98 ₋₁₃ | 7.90 ₋₁₄ | 7.65 ₋₁₄ | 8.26 ₋₁₄ |
| 131072 | 4.91 ₋₀₈ | 7.83 ₋₁₀ | 1.66 ₋₁₁ | 1.45 ₋₁₃ | 2.36 ₋₁₄ | 2.45 ₋₁₄ | 3.00 ₋₁₄ |
| 156800 | 4.29 ₋₀₈ | 6.13 ₋₁₀ | 1.49 ₋₁₁ | 1.29 ₋₁₃ | 1.42 ₋₁₄ | 2.27 ₋₁₄ | 2.91 ₋₁₄ |

Tabelle 10.9: Approximationsfehler: $\|G - \tilde{G}\|_2$ für $m = 8$ und $\ell = 1$ (DLP)

Die nächsten beiden Tabellen zeigen die Resultate, die das Green-Hybrid-Verfahren im dreidimensionalen Fall für das Doppelschichtpotential liefert. In Tabelle 10.9 ist der Approximationsfehler zu sehen und in Tabelle 10.10 die entsprechenden Zeiten.

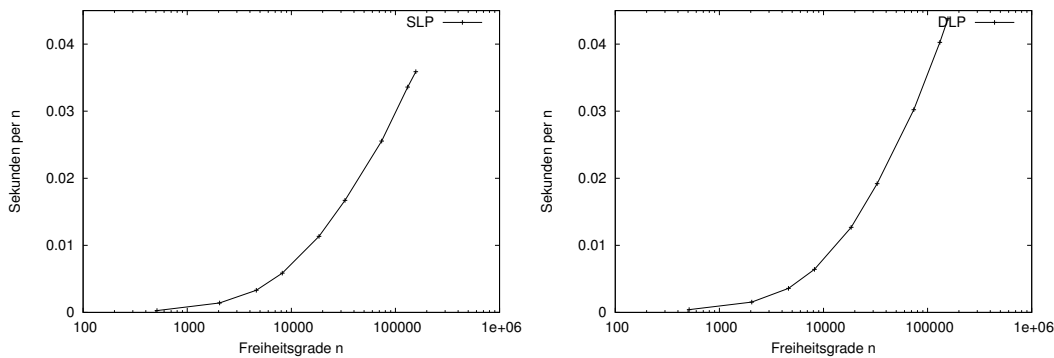
Dabei ist ein weiteres Mal zu beobachten, dass für DLP mit derselben Wahl der Quadraturordnung $m = 8$ nicht die gleichen Genauigkeiten wie für SLP erreicht werden können. Um für DLP auf ähnlich gute Fehlerwerte wie bei SLP in Tabelle 10.7 zu kommen, wäre eine höhere Quadraturordnung von etwa $m = 10$ notwendig. Dadurch würden dann aber auch die Berechnungszeiten steigen.

In der nächsten Abbildung 10.8 ist die bei wachsender Problemgröße ansteigende Berechnungszeit des Hybrid-Verfahrens pro Freiheitsgrad dargestellt, links für das Einfachschichtpotential und rechts für das Doppelschichtpotential.

Im dreidimensionalen Fall steigen die Kurven steiler an als im zweidimensionalen Fall. Während zuvor ein Wachstum von $\sim \log n$ zu beobachten war, verhalten sich hier die Kurven eher $\sim (\log n)^2$, welches aber durch die Konstruktion des Verfahrens zu erwarten war.

Zuletzt in diesem Abschnitt wird der Einfluss des Parameters δ nun auch für den dreidimensionalen Fall untersucht. Auch in diesem Fall wurde δ zuvor für die Theorie auf

| n | $\varepsilon_{aca} = 10^{-2}$ | $\varepsilon_{aca} = 10^{-4}$ | $\varepsilon_{aca} = 10^{-6}$ | $\varepsilon_{aca} = 10^{-8}$ | $\varepsilon_{aca} = 10^{-10}$ | $\varepsilon_{aca} = 10^{-12}$ | $\varepsilon_{aca} = 10^{-14}$ |
|--------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|--------------------------------|--------------------------------|--------------------------------|
| 2048 | 2.7 | 3.1 | — | — | — | — | — |
| 4608 | 11.8 | 14.6 | 16.1 | 16.5 | — | — | — |
| 8192 | 25.4 | 37.4 | 45.3 | 49.3 | 51.2 | 52.5 | 53.2 |
| 18432 | 80.2 | 125.8 | 166.0 | 198.4 | 219.6 | 233.8 | 242.2 |
| 32768 | 168.0 | 285.6 | 397.0 | 494.5 | 571.9 | 629.0 | 681.7 |
| 73728 | 474.8 | 847.1 | 1221.7 | 1592.9 | 1919.4 | 2227.7 | 2480.8 |
| 131072 | 1011.7 | 1866.0 | 2764.4 | 3712.1 | 4529.0 | 5277.6 | 5910.1 |
| 156800 | 1265.0 | 2320.3 | 3475.4 | 4663.5 | 5749.0 | 6865.2 | 7851.6 |

 Tabelle 10.10: Aufbau der \mathcal{H} -Matrix in Sekunden für $m = 8$ und $\ell = 1$ (DLP)

 Abbildung 10.8: Kugel-Geometrie: Green-Hybrid-Verfahren mit Quader-Parametrisierung, $\delta = 0.5 \text{ diam}$, maximale Blattgröße $clf = 128$, Zeit pro Freiheitsgrad für ein festes $m = 8$ und ein festes $\varepsilon_{aca} = 10^{-12}$ für SLP (links) und DLP (rechts)

$\delta = \frac{1}{2} \text{diam}_{\max}(\mathcal{B}_\tau)$ festgelegt. Abbildung 10.9 zeigt jedoch, dass auch hier ein größeres δ zu besseren Resultaten führt.

So ist der Matrixapproximationsfehler für SLP bei $\delta = \frac{9}{10} \text{diam}_{\max}(\mathcal{B}_\tau)$ am kleinsten, während es bei DLP aber keinen großen Unterschied macht, solange δ nicht kleiner als $\frac{1}{10} \text{diam}_{\max}(\mathcal{B}_\tau)$ ist, wie in der linken Teilabbildung der Abbildung 10.9 erkennbar ist.

Beim Speicherbedarf macht die Wahl von δ einen großen Unterschied für SLP und DLP aus. Sehr deutlich zu sehen ist in der rechten Teilabbildung der Abbildung, dass sowohl die Kurve zum Einfach- als auch die zum Doppelschichtpotential bei $\delta = \text{diam}_{\max}(\mathcal{B}_\tau)$ ihr Maximum erreicht.

Damit wird auch im dreidimensionalen Fall für die im nächsten Abschnitt 10.1.4 folgenden vergleichenden Experimente $\delta = \text{diam}_{\max}(\mathcal{B}_\tau)$ festgelegt als optimale von der Theorie unterstützte Wahl des Parameters.

10 Numerische Experimente für das Hybrid-Verfahren

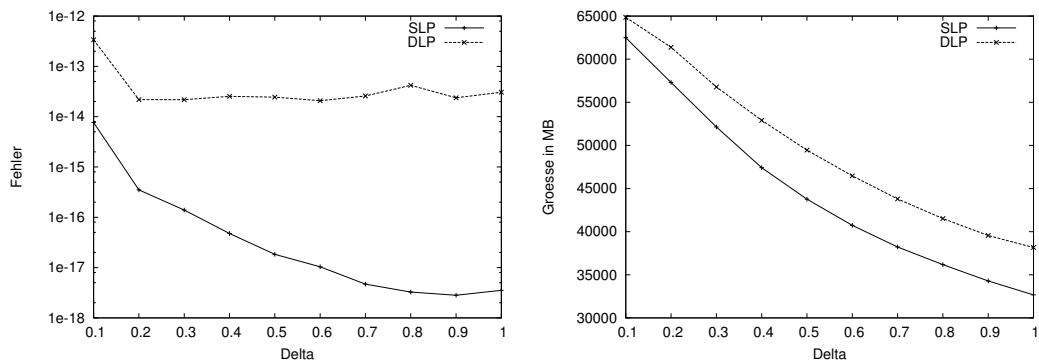


Abbildung 10.9: Kugel-Geometrie, $n = 131072$: Green-Hybrid-Verfahren mit Quader-Parametrisierung, maximaler Blattgröße $clf = 128$ für ein festes $m = 8$, $\ell = 1$ sowie ein festes $\varepsilon_{aca} = 10^{-12}$ und verschiedene δ : Genauigkeit (links) und Matrixgröße (rechts)

10.1.4 3D-Fall Vergleiche

Nach der Analyse des Green-Hybrid-Verfahrens in Bezug auf den Einfluss seiner Parameter auf das Konvergenzverhalten folgt nun auch im dreidimensionalen Fall der Vergleich zu anderen Verfahren.

Ein weiteres Mal wird zuerst mit dem einfachen Green-Quadratur-Verfahren verglichen. Getestet wurden beide Verfahren auf der Kugel-Geometrie als Ω mit $n = 131072$ Freiheitsgraden.

Die Resultate sind in Abbildung 10.10 zu sehen, die Matrixgrößen im Speicher in der linken Teilabbildung und die entsprechenden Zeiten des Matrixaufbaus in der rechten Teilabbildung.

Wie bereits im zweidimensionalen Fall stellt das Hybrid-Verfahren eine erhebliche Verbesserung gegenüber dem Green-Quadratur-Verfahren dar. Da gerade der dreidimensionale Fall das Problem bei dem einfachen Verfahren war, ist dieses Ergebnis recht bemerkenswert.

Ob die Verbesserung jedoch ausreicht, um das Ziel der Konkurrenzfähigkeit gegenüber der Kreuzapproximation und der Interpolation auch im dreidimensionalen Fall zu erreichen, ist Gegenstand des nächsten Experiments.

Dazu wurde eine komplexere Geometrie bemüht, um unrealistische Effekte zu vermeiden, die bei einem Vergleich der Verfahren auf der eventuell zu simplen Kugel-Geometrie auftreten könnten.

Das Gebiet Ω ist hier als die Kurbelwelle gewählt, wie sie in Abbildung 10.11 dargestellt ist. In der Abbildung ist diese allerdings der Anschaulichkeit wegen mit nur 25744 Dreiecken abgebildet, während die Anzahl der Freiheitsgrade im Experiment auf $n = 102976$ verfeinert ist. Das Gitter der Kurbelwelle wurde mit dem Paket *netgen* von Joachim Schöberl erzeugt.

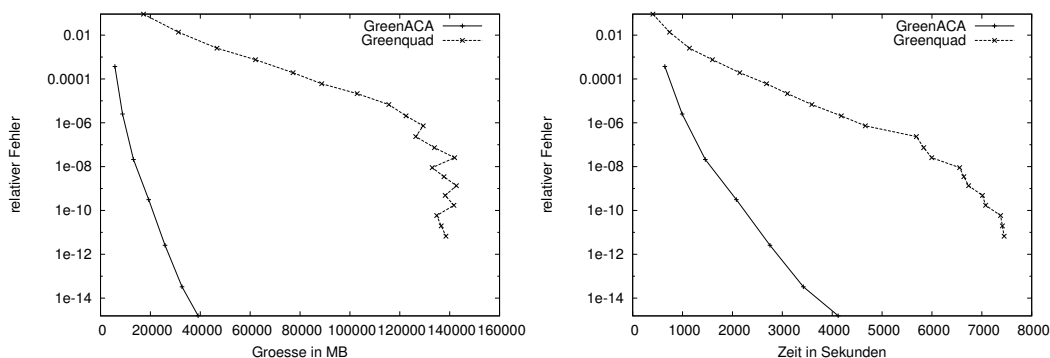


Abbildung 10.10: Kugel-Geometrie, $n = 131\,072$: Green-Hybrid-Verfahren mit Quader-Parametrisierung, maximaler Blattgröße $clf = 128$ für ein festes $m = 8$, $\ell = 1$, $\delta = 1.0$ über verschiedene ε_{aca} gegenüber dem Green-Quadratur-Verfahren mit $clf = 2k$, $\ell = 1$, $\delta = 0.5$ über verschiedene m : Genauigkeit (links) und Zeit (rechts)

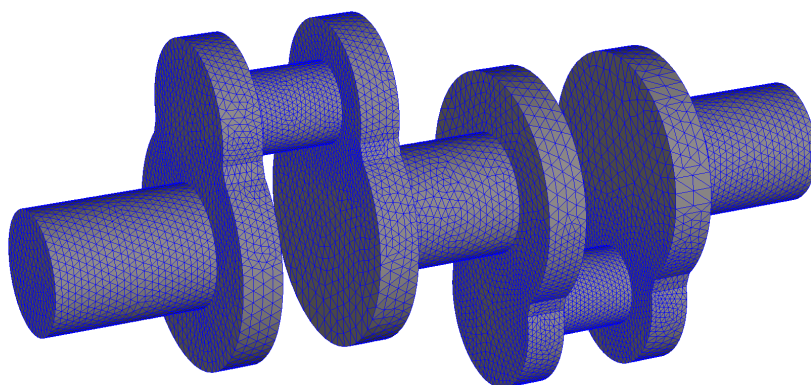


Abbildung 10.11: Kurbelwelle, Darstellung mit 25 744 Dreiecken

Die Resultate für das Green-Hybrid-Verfahren mit den Quadraturordnungen $m = 8$ und $m = 12$, der Interpolation und der adaptiven Kreuzapproximation mit partieller Pivotierung auf dieser Kurbelwellen-Geometrie sind in Abbildung 10.12 zu sehen.

Dabei zeigt sich zunächst, dass hier die Quadraturordnung $m = 8$ für das Hybrid-Verfahren eine gute Wahl ist. Die Vermutung, dass auf einer komplizierteren Geometrie eine höhere Quadraturordnung notwendig wird, als im zweidimensionalen Fall anhand der Hilbertkurven-Geometrie zu beobachten war, trifft hier nicht zu. Eine Erhöhung der Quadraturordnung auf $m = 12$ bringt zwar ein wenig mehr Genauigkeit, jedoch zu dem Preis einer deutlich längeren Berechnungszeit.

10 Numerische Experimente für das Hybrid-Verfahren

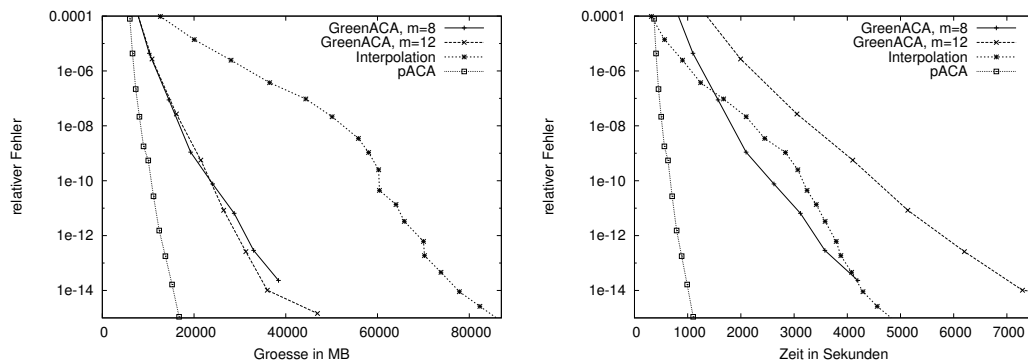


Abbildung 10.12: Kurbelwellen-Geometrie, $n = 102\,986$: Green-Hybrid-Verfahren mit Quader-Parametrisierung, maximaler Blattgröße $clf = 128$ einmal für ein festes $m = 8$ und einmal für ein festes $m = 12$, $\ell = 1$, $\delta = 1.0$ im Vergleich zur Interpolation und ACA mit partieller Pivotierung: Speicherbedarf (links) und Zeit (rechts)

Mit welcher Quadraturordnung auch immer, die Hoffnung an ACA mit partieller Pivotierung heran zu kommen, erfüllt sich bei dem Hybrid-Verfahren für gewöhnliche \mathcal{H} -Matrizen leider nicht. ACA benötigt nur in etwa die Hälfte des Speichers und ist dazu noch wesentlich schneller.

An dieser Stelle kann man dem Green-Hybrid-Verfahren zugute halten, dass es auf recht elegante Weise auf \mathcal{H}^2 -Matrizen zu übertragen ist, wie in Kapitel 8.2 gezeigt wurde. Dieses ist für ACA aber nicht ohne Weiteres möglich.

Gegen die Interpolation hingegen kann sich das Green-Hybrid-Verfahren gut behaupten. Es benötigt weniger als die Hälfte an Speicher und ist dabei (zumindest für $m = 8$) sogar noch ein klein wenig schneller als die Interpolation.

Insgesamt sehen die Ergebnisse für das Green-Hybrid-Verfahren aber auch im dreidimensionalen Fall zufriedenstellend aus.

Es ist eine große Verbesserung gegenüber dem einfachen Green-Hybrid-Verfahren zu erkennen, das ja im dreidimensionalen Fall eher als enttäuschend einzustufen war. Viel wichtiger ist jedoch, dass es in einem vergleichbaren Rahmen mit anderen gängigen Verfahren liegt.

Weitere Verbesserungen des Verfahrens, wie zum Beispiel die Ausnutzung der engen Verbindung zwischen Neumann- und Dirichlet-Werten, könnten dazu beitragen, dass das Verfahren auch mit der Kreuzapproximation gleichziehen oder sogar noch besser werden könnte.

Diese Weiterentwicklungen des Green-Quadratur-Ansatzes sind jedoch nicht mehr Gegenstand dieser Arbeit.

10.2 Hybrid-Verfahren für \mathcal{H}^2 -Matrizen

Dass sich das Green-Hybrid-Verfahren ohne größere Probleme auf \mathcal{H}^2 -Matrizen erweitern lässt, wurde in Kapitel 8.2 vorgeführt. Die Implementierung in der *H2Lib* wurde dementsprechend angepasst, so dass das Hybrid-Verfahren auch für \mathcal{H}^2 -Matrizen getestet werden kann.

Mit den Erkenntnissen aus dem vorangegangenen Abschnitt 10.1 über die Auswirkungen der einzelnen Parameter auf die Konvergenz des Verfahrens, kann für diesen Abschnitt hier sofort $\delta = \text{diam}_{\max}(\mathcal{B}_\tau)$ fest gesetzt werden. Außerdem wird auf zusammengesetzte Quadratur verzichtet und nur noch $\ell = 1$ betrachtet.

Es folgen numerische Experimente zum zweidimensionalen Fall in Abschnitt 10.2.1 und danach zum dreidimensionalen Fall in Abschnitt 10.2.2.

10.2.1 2D-Fall

Die numerischen Tests für das Verfahren für \mathcal{H}^2 -Matrizen im zweidimensionalen Fall werden auf den bereits verwendeten Geometrien für Ω , also den Kreis und die Hilbertkurve, durchgeführt.

Als Erstes ist natürlich interessant, wieviel besser das Green-Hybrid-Verfahren für \mathcal{H}^2 -Matrizen gegenüber dem für \mathcal{H} -Matrizen ist.

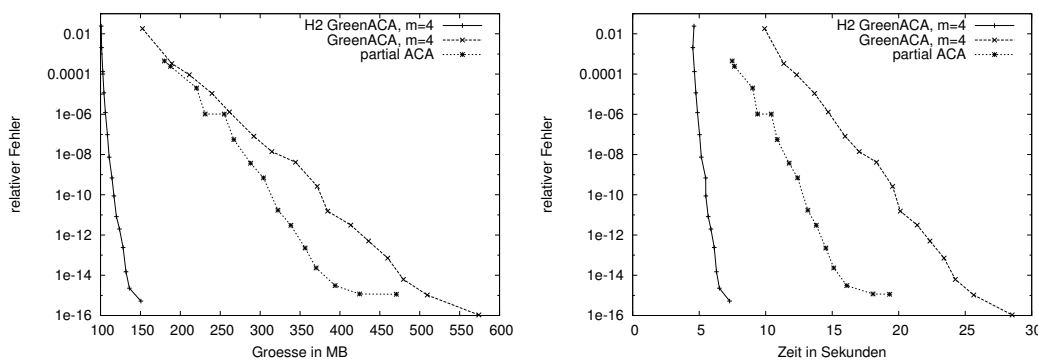


Abbildung 10.13: Kreis-Geometrie, $n = 65\,536$: Green-Hybrid-Verfahren mit Rechtecks-Parametrisierung, maximaler Blattgröße $clf = 64$ für ein festes $m = 4$, $\ell = 1$, $\delta = 1.0$ im Vergleich zu dem Green-Hybrid-Verfahren auf \mathcal{H}^2 -Matrizen mit denselben Parametern und ACA mit partieller Pivotierung: Speicherbedarf (links) und Zeit (rechts)

Dazu wurde das Verfahren auf der Kreis-Geometrie mit $n = 65\,536$ Freiheitsgraden einmal für \mathcal{H}^2 -Matrizen und einmal für \mathcal{H} -Matrizen laufen gelassen. Zum Vergleich wird noch einmal die adaptive Kreuzapproximation mit partieller Pivotsuche dagegen gehalten.

Abbildung 10.13 zeigt die Resultate zu diesem Vergleich. In der linken Teilabbildung

der Abbildung sind die Matrixgrößen im Speicher dargestellt und in der rechten Teilabbildung die zugehörigen Zeiten des Matrixaufbaus.

Es ist zu erkennen, dass das Hybrid-Verfahren für \mathcal{H}^2 -Matrizen nur etwa ein Viertel des Speichers benötigt, den das Verfahren für \mathcal{H} -Matrizen braucht und nur ein Drittel des Speicherbedarfs von ACA mit partieller Pivotierung. Dabei ist es auch noch in etwa doppelt so schnell wie ACA.

Dieses erfreuliche Ergebnis zugunsten des Hybrid-Verfahrens für \mathcal{H}^2 -Matrizen soll auch nochmal in Form von Tabellen festgehalten werden.

So zeigt Tabelle 10.11 den Approximationsfehler $\|G - \tilde{G}\|_2$ für das Einfachschichtpotential für eine feste Quadraturordnung $m = 4$ und Tabelle 10.12 die zugehörigen Berechnungszeiten.

| n | ε_{aca} $= 10^{-2}$ | ε_{aca} $= 10^{-4}$ | ε_{aca} $= 10^{-6}$ | ε_{aca} $= 10^{-8}$ | ε_{aca} $= 10^{-10}$ | ε_{aca} $= 10^{-12}$ | ε_{aca} $= 10^{-14}$ |
|--------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 1024 | 5.59 ₋₀₇ | 2.44 ₋₀₉ | 2.92 ₋₁₁ | 1.91 ₋₁₃ | 2.01 ₋₁₅ | 5.14 ₋₁₇ | 5.69 ₋₁₉ |
| 2048 | 5.70 ₋₀₇ | 1.83 ₋₀₉ | 2.56 ₋₁₁ | 1.20 ₋₁₃ | 2.02 ₋₁₅ | 1.97 ₋₁₇ | 1.69 ₋₁₉ |
| 4096 | 3.64 ₋₀₇ | 1.35 ₋₀₉ | 1.36 ₋₁₁ | 1.01 ₋₁₃ | 1.16 ₋₁₅ | 4.45 ₋₁₇ | 1.75 ₋₁₉ |
| 8192 | 2.23 ₋₀₇ | 9.91 ₋₁₀ | 9.46 ₋₁₂ | 5.03 ₋₁₄ | 8.73 ₋₁₆ | 2.38 ₋₁₇ | 1.77 ₋₁₉ |
| 16384 | 1.29 ₋₀₇ | 6.34 ₋₁₀ | 5.37 ₋₁₂ | 3.42 ₋₁₄ | 4.87 ₋₁₆ | 1.45 ₋₁₇ | 1.15 ₋₁₉ |
| 32768 | 6.72 ₋₀₈ | 3.51 ₋₁₀ | 2.48 ₋₁₂ | 2.04 ₋₁₄ | 2.69 ₋₁₆ | 7.14 ₋₁₈ | 6.75 ₋₂₀ |
| 65536 | 3.43 ₋₀₈ | 1.94 ₋₁₀ | 1.62 ₋₁₂ | 1.15 ₋₁₄ | 1.36 ₋₁₆ | 3.98 ₋₁₈ | 3.77 ₋₂₀ |
| 131072 | 1.77 ₋₀₈ | 9.90 ₋₁₁ | 8.15 ₋₁₃ | 6.93 ₋₁₅ | 8.41 ₋₁₇ | 2.19 ₋₁₈ | 1.98 ₋₂₀ |

Tabelle 10.11: Approximationsfehler der \mathcal{H}^2 -Matrix: $\|G - \tilde{G}\|_2$ für $m = 4$, $\ell = 1$ (SLP)

| n | ε_{aca} $= 10^{-2}$ | ε_{aca} $= 10^{-4}$ | ε_{aca} $= 10^{-6}$ | ε_{aca} $= 10^{-8}$ | ε_{aca} $= 10^{-10}$ | ε_{aca} $= 10^{-12}$ | ε_{aca} $= 10^{-14}$ |
|--------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 1024 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| 2048 | 0.2 | 0.3 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |
| 4096 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.4 | 0.4 |
| 8192 | 0.6 | 0.6 | 0.6 | 0.7 | 0.7 | 0.8 | 0.8 |
| 16384 | 1.1 | 1.2 | 1.2 | 1.3 | 1.4 | 1.5 | 1.6 |
| 32768 | 2.3 | 2.4 | 2.5 | 2.7 | 2.8 | 3.1 | 3.3 |
| 65536 | 4.5 | 4.8 | 5.0 | 5.5 | 5.7 | 6.1 | 6.5 |
| 131072 | 9.6 | 9.7 | 10.2 | 10.9 | 11.5 | 12.7 | 13.2 |

Tabelle 10.12: Aufbau der \mathcal{H} -Matrix in Sekunden für $m = 4$ und $\ell = 1$ (SLP)

Die Ergebnisse für das Doppelschichtpotential sind in den Tabellen 10.13 (Approximationsfehler) und 10.14 (Berechnungszeit) zu finden. Diese sehen ebenso gut aus, wie die zum Einfachschichtpotential.

Es bleibt noch eine Gegenüberstellung mit einem Verfahren, das ebenfalls für \mathcal{H}^2 -Matrizen anwendbar ist, nämlich der Interpolation.

10.2 Hybrid-Verfahren für \mathcal{H}^2 -Matrizen

| n | ε_{aca} = 10^{-2} | ε_{aca} = 10^{-4} | ε_{aca} = 10^{-6} | ε_{aca} = 10^{-8} | ε_{aca} = 10^{-10} | ε_{aca} = 10^{-12} | ε_{aca} = 10^{-14} |
|--------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 1024 | 1.70 ₋₀₆ | 1.48 ₋₀₈ | 1.66 ₋₁₀ | 1.74 ₋₁₂ | 2.45 ₋₁₄ | 2.95 ₋₁₆ | 3.40 ₋₁₈ |
| 2048 | 1.13 ₋₀₆ | 8.38 ₋₀₉ | 9.52 ₋₁₁ | 1.25 ₋₁₂ | 9.75 ₋₁₅ | 1.43 ₋₁₆ | 1.99 ₋₁₈ |
| 4096 | 6.31 ₋₀₇ | 6.72 ₋₀₉ | 6.78 ₋₁₁ | 8.27 ₋₁₃ | 7.82 ₋₁₅ | 1.35 ₋₁₆ | 1.50 ₋₁₈ |
| 8192 | 3.96 ₋₀₇ | 4.39 ₋₀₉ | 3.30 ₋₁₁ | 4.77 ₋₁₃ | 7.51 ₋₁₅ | 8.09 ₋₁₇ | 1.11 ₋₁₈ |
| 16384 | 2.10 ₋₀₇ | 2.70 ₋₀₉ | 2.05 ₋₁₁ | 2.31 ₋₁₃ | 8.07 ₋₁₅ | 7.35 ₋₁₇ | 1.01 ₋₁₈ |
| 32768 | 1.16 ₋₀₇ | 1.50 ₋₀₉ | 1.09 ₋₁₁ | 1.35 ₋₁₃ | 4.25 ₋₁₅ | 4.23 ₋₁₇ | 7.73 ₋₁₉ |
| 65536 | 5.67 ₋₀₈ | 7.51 ₋₁₀ | 5.54 ₋₁₂ | 8.26 ₋₁₄ | 1.69 ₋₁₅ | 2.65 ₋₁₇ | 4.72 ₋₁₉ |
| 131072 | 3.07 ₋₀₈ | 4.16 ₋₁₀ | 3.36 ₋₁₂ | 4.37 ₋₁₄ | 8.82 ₋₁₆ | 1.59 ₋₁₇ | 3.17 ₋₁₉ |

Tabelle 10.13: Approximationsfehler der \mathcal{H}^2 -Matrix: $\|G - \tilde{G}\|_2$ für $m = 4$, $\ell = 1$ (DLP)

| n | ε_{aca} = 10^{-2} | ε_{aca} = 10^{-4} | ε_{aca} = 10^{-6} | ε_{aca} = 10^{-8} | ε_{aca} = 10^{-10} | ε_{aca} = 10^{-12} | ε_{aca} = 10^{-14} |
|--------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 1024 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2048 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| 4096 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.2 |
| 8192 | 0.2 | 0.2 | 0.2 | 0.3 | 0.3 | 0.3 | 0.3 |
| 16384 | 0.4 | 0.5 | 0.5 | 0.5 | 0.6 | 0.6 | 0.7 |
| 32768 | 0.8 | 0.9 | 1.0 | 1.0 | 1.1 | 1.3 | 1.4 |
| 65536 | 1.7 | 1.8 | 2.0 | 2.1 | 2.3 | 2.5 | 2.9 |
| 131072 | 3.8 | 3.7 | 4.0 | 4.4 | 4.7 | 5.3 | 6.0 |

Tabelle 10.14: Aufbau der \mathcal{H} -Matrix in Sekunden für $m = 4$ und $\ell = 1$ (DLP)

Erneut wird für dieses Vergleichsexperiment die Hilbertkurven-Geometrie der Ordnung 8, also mit $n = 2^{16} - 1 = 65\,535$ Freiheitsgraden für Ω genutzt.

Abbildung 10.14 zeigt die Kurven zum Green-Hybrid-Verfahren für \mathcal{H}^2 -Matrizen für die Quadraturordnungen $m = 4$ und $m = 8$ und die der Interpolation für \mathcal{H}^2 -Matrizen. In der linken Teilabbildung sind dabei die Speicheranforderungen und in der rechten Teilabbildung die entsprechenden Berechnungszeiten zu sehen.

Zunächst ist zu erkennen, dass eine Quadraturordnung von $m = 4$ auf der Hilbertkurve nicht ausreicht, wie auch schon frühere Experimente gezeigt haben. Mit der Quadraturordnung $m = 8$ sind jedoch sehr gute Resultate zu erzielen: So gut, dass ab einer Genauigkeit von etwa 10^{-7} das Hybrid-Verfahren um ein Vielfaches (zwei- bis fünffaches) besser als die Interpolation ist. Es benötigt wesentlich weniger Speicher und ist deutlich schneller, als die Interpolation.

Also kann die Anwendung des Green-Hybrid-Verfahrens auf \mathcal{H}^2 -Matrizen zumindest im zweidimensionalen Fall als Erfolg gewertet werden. Im nächsten Abschnitt 10.2.2 zeigt sich dann, ob das auch im praxisrelevanteren dreidimensionalen Fall so ist.

10 Numerische Experimente für das Hybrid-Verfahren

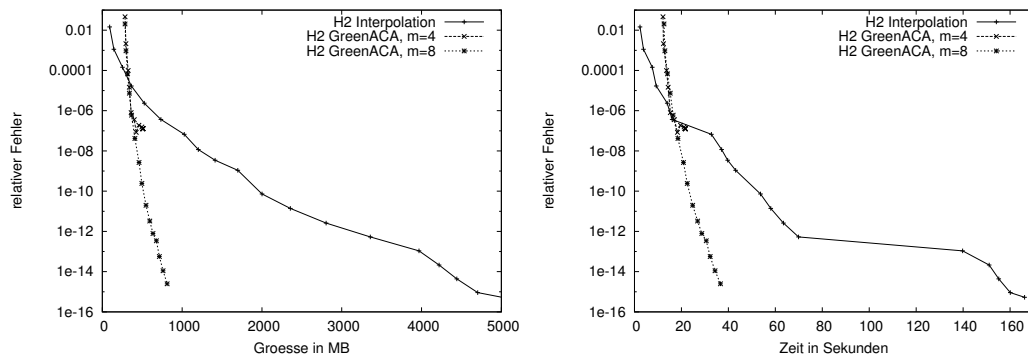


Abbildung 10.14: Hilbertkurven-Geometrie, $n = 65\,535$: Green-Hybrid-Verfahren auf \mathcal{H}^2 -Matrizen mit Rechtecks-Parametrisierung, maximaler Blattgröße $clf = 64$, $\ell = 1$, $\delta = 1.0$, einmal für ein festes $m = 4$ und einmal für ein festes $m = 8$, im Vergleich zur Interpolation auf \mathcal{H}^2 -Matrizen mit $clf = 2m^3$: Speicherbedarf (links) und Zeit (rechts)

10.2.2 3D-Fall

Auch die numerischen Experimente im dreidimensionalen Fall werden auf schon verwendeten Geometrien für Ω durchgeführt: Das sind hier die Kugel-Geometrie und die Kurbelwellen-Geometrie.

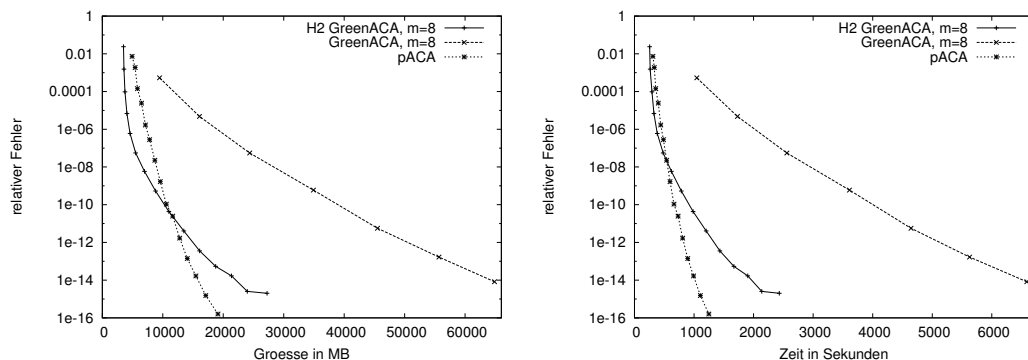


Abbildung 10.15: Kugel-Geometrie, $n = 156\,800$: Green-Hybrid-Verfahren mit Quader-Parametrisierung, maximaler Blattgröße $clf = 128$ für ein festes $m = 8$, $\ell = 1$, $\delta = 1.0$ im Vergleich zu dem Green-Hybrid-Verfahren auf \mathcal{H}^2 -Matrizen mit denselben Parametern und ACA mit partieller Pivotierung: Speicherbedarf (links) und Zeit (rechts)

Im ersten Test in diesem Abschnitt ist Ω die Kugel-Geometrie mit $n = 156\,800$ Freiheitsgraden und es werden darauf das Green-Hybrid-Verfahren für \mathcal{H}^2 -Matrizen und das

10.2 Hybrid-Verfahren für \mathcal{H}^2 -Matrizen

für \mathcal{H} -Matrizen, jeweils mit der Quadraturordnung $m = 8$, und die adaptive Kreuzapproximation mit partieller Pivotierung gegenüber gestellt.

Die Kurven dazu sind in Abbildung 10.15 zu sehen, die jeweiligen Matrixgrößen im Speicher sind dabei in der linken Teilabbildung zu finden und die zugehörige Zeit, die zum Aufstellen der Matrizen nötig war, in der rechten Teilabbildung.

Auch anhand dieser Teilabbildungen sieht man sofort, dass das Hybrid-Verfahren für \mathcal{H}^2 -Matrizen gegenüber dem für \mathcal{H} -Matrizen enorme Speicher- und Zeiteinsparungen zu verbuchen hat. Leider reichen diese Einsparungen nicht aus, um für jede beliebige Genauigkeit auch einen Vorteil gegenüber ACA mit partieller Pivotsuche zu erreichen. So ist das Hybrid-Verfahren für \mathcal{H}^2 -Matrizen der Kreuzapproximation im Speicherbedarf bis zu einer Genauigkeit von etwa 10^{-10} überlegen und zeitlich nur bis etwa 10^{-8} . Bei höheren Genauigkeiten ist ACA sowohl speichergünstiger als auch schneller.

Auch wenn der Vergleich zur Kreuzapproximation im dreidimensionalen Fall nur einen Teilerfolg darstellt, ist der Gewinn des Hybrid-Verfahrens für \mathcal{H}^2 -Matrizen gegenüber dem für \mathcal{H} -Matrizen sehr groß.

| n | ε_{aca} = 10^{-2} | ε_{aca} = 10^{-4} | ε_{aca} = 10^{-6} | ε_{aca} = 10^{-8} | ε_{aca} = 10^{-10} | ε_{aca} = 10^{-12} | ε_{aca} = 10^{-14} |
|--------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 2048 | 1.50 ₋₀₆ | 4.92 ₋₀₉ | 4.77 ₋₁₁ | 1.00 ₋₁₃ | — | — | — |
| 4608 | 6.23 ₋₀₇ | 3.51 ₋₀₉ | 6.49 ₋₁₁ | 3.10 ₋₁₃ | 1.10 ₋₁₅ | 8.71 ₋₁₈ | — |
| 8192 | 8.31 ₋₀₇ | 3.88 ₋₀₉ | 6.65 ₋₁₁ | 2.72 ₋₁₃ | 2.15 ₋₁₅ | 3.25 ₋₁₇ | 3.19 ₋₁₈ |
| 18432 | 4.27 ₋₀₇ | 2.82 ₋₀₉ | 4.05 ₋₁₁ | 2.14 ₋₁₃ | 1.25 ₋₁₅ | 3.14 ₋₁₇ | 3.06 ₋₁₈ |
| 32768 | 4.05 ₋₀₇ | 1.92 ₋₀₉ | 2.20 ₋₁₁ | 1.44 ₋₁₃ | 9.25 ₋₁₆ | 1.88 ₋₁₇ | 1.51 ₋₁₈ |
| 73728 | 2.22 ₋₀₇ | 1.09 ₋₀₉ | 9.80 ₋₁₂ | 8.31 ₋₁₄ | 6.09 ₋₁₆ | 9.00 ₋₁₈ | 7.60 ₋₁₉ |
| 131072 | 1.40 ₋₀₇ | 7.21 ₋₁₀ | 5.76 ₋₁₂ | 5.52 ₋₁₄ | 4.00 ₋₁₆ | 4.95 ₋₁₈ | 3.33 ₋₁₉ |
| 156800 | 1.38 ₋₀₇ | 6.08 ₋₁₀ | 4.97 ₋₁₂ | 4.73 ₋₁₄ | 3.62 ₋₁₆ | 4.81 ₋₁₈ | 2.27 ₋₁₉ |

Tabelle 10.15: Approximationsfehler der \mathcal{H}^2 -Matrix: $\|G - \tilde{G}\|_2$ für $m = 8$ und $\ell = 1$ (SLP)

| n | ε_{aca} = 10^{-2} | ε_{aca} = 10^{-4} | ε_{aca} = 10^{-6} | ε_{aca} = 10^{-8} | ε_{aca} = 10^{-10} | ε_{aca} = 10^{-12} | ε_{aca} = 10^{-14} |
|--------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 2048 | 2.3 | 3.0 | 4.6 | 6.9 | — | — | — |
| 4608 | 6.2 | 7.9 | 11.8 | 18.1 | 24.3 | 30.2 | — |
| 8192 | 12.0 | 15.0 | 22.7 | 35.6 | 49.4 | 60.8 | 71.0 |
| 18432 | 27.4 | 34.6 | 53.2 | 87.9 | 128.0 | 165.8 | 201.4 |
| 32768 | 52.9 | 65.6 | 98.2 | 162.0 | 240.9 | 322.5 | 399.0 |
| 73728 | 114.7 | 144.7 | 222.8 | 373.4 | 556.3 | 763.2 | 971.0 |
| 131072 | 221.5 | 273.3 | 398.6 | 663.1 | 1010.1 | 1390.2 | 1775.2 |
| 156800 | 262.9 | 327.1 | 480.8 | 787.5 | 1203.8 | 1668.9 | 2136.9 |

Tabelle 10.16: Aufbau der \mathcal{H}^2 -Matrix in Sekunden für $m = 8$ und $\ell = 1$ (SLP)

10 Numerische Experimente für das Hybrid-Verfahren

Konkrete Zahlen, die dieses bestätigen, sind in den Tabellen gegeben. Tabelle 10.15 zeigt den Matrixapproximationsfehler $\|G - \tilde{G}\|_2$ der \mathcal{H}^2 -Matrix für SLP und Tabelle 10.16 zeigt die zugehörige Berechnungszeit, die nötig ist, um die Matrix aufzustellen.

Für DLP sind die entsprechenden Daten in den Tabellen 10.17 und 10.18 zu finden.

| n | ε_{aca} $= 10^{-2}$ | ε_{aca} $= 10^{-4}$ | ε_{aca} $= 10^{-6}$ | ε_{aca} $= 10^{-8}$ | ε_{aca} $= 10^{-10}$ | ε_{aca} $= 10^{-12}$ | ε_{aca} $= 10^{-14}$ |
|--------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 2048 | 9.46 ₋₀₇ | 6.29 ₋₀₉ | 8.37 ₋₁₁ | 3.12 ₋₁₃ | — | — | — |
| 4608 | 1.00 ₋₀₆ | 5.61 ₋₀₉ | 9.09 ₋₁₁ | 4.33 ₋₁₃ | 5.89 ₋₁₅ | 5.93 ₋₁₇ | — |
| 8192 | 8.49 ₋₀₇ | 5.49 ₋₀₉ | 8.39 ₋₁₁ | 5.64 ₋₁₃ | 9.77 ₋₁₅ | 2.15 ₋₁₆ | 1.76 ₋₁₇ |
| 18432 | 5.73 ₋₀₇ | 3.36 ₋₀₉ | 4.72 ₋₁₁ | 3.00 ₋₁₃ | 4.05 ₋₁₅ | 2.64 ₋₁₆ | 1.98 ₋₁₇ |
| 32768 | 3.41 ₋₀₇ | 2.60 ₋₀₉ | 2.20 ₋₁₁ | 1.73 ₋₁₃ | 2.56 ₋₁₅ | 1.48 ₋₁₆ | 1.51 ₋₁₇ |
| 73728 | 1.86 ₋₀₇ | 1.40 ₋₀₉ | 1.01 ₋₁₁ | 8.32 ₋₁₄ | 1.06 ₋₁₅ | 6.77 ₋₁₇ | 5.85 ₋₁₈ |
| 131072 | 1.21 ₋₀₇ | 9.20 ₋₁₀ | 6.57 ₋₁₂ | 5.95 ₋₁₄ | 6.07 ₋₁₆ | 3.85 ₋₁₇ | 3.73 ₋₁₈ |
| 156800 | 1.10 ₋₀₇ | 8.13 ₋₁₀ | 5.84 ₋₁₂ | 5.34 ₋₁₄ | 7.03 ₋₁₆ | 3.24 ₋₁₇ | 2.98 ₋₁₈ |

Tabelle 10.17: Approximationsfehler der \mathcal{H}^2 -Matrix: $\|G - \tilde{G}\|_2$ für $m = 8$ und $\ell = 1$ (DLP)

| n | ε_{aca} $= 10^{-2}$ | ε_{aca} $= 10^{-4}$ | ε_{aca} $= 10^{-6}$ | ε_{aca} $= 10^{-8}$ | ε_{aca} $= 10^{-10}$ | ε_{aca} $= 10^{-12}$ | ε_{aca} $= 10^{-14}$ |
|--------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 2048 | 2.7 | 3.8 | 5.8 | 8.2 | — | — | — |
| 4608 | 6.8 | 9.7 | 15.2 | 22.0 | 28.0 | 33.6 | — |
| 8192 | 13.6 | 18.8 | 29.6 | 44.2 | 58.2 | 69.6 | 80.4 |
| 18432 | 30.1 | 42.4 | 70.2 | 111.7 | 155.2 | 195.1 | 231.2 |
| 32768 | 59.6 | 80.65 | 129.7 | 207.6 | 297.4 | 384.3 | 465.8 |
| 73728 | 129.5 | 178.2 | 294.8 | 477.6 | 692.3 | 927.1 | 1150.6 |
| 131072 | 248.3 | 327.6 | 523.3 | 862.7 | 1260.8 | 1686.9 | 2117.8 |
| 156800 | 302.5 | 397.6 | 628.9 | 1024.6 | 1517.0 | 2040.4 | 2563.7 |

Tabelle 10.18: Aufbau der \mathcal{H}^2 -Matrix in Sekunden für $m = 8$ und $\ell = 1$ (DLP)

In einem allerletzten Experiment wird auch im dreidimensionalen Fall das Green-Hybrid-Verfahren für \mathcal{H}^2 -Matrizen mit der Interpolation für \mathcal{H}^2 -Matrizen verglichen.

Dazu sei Ω ein weiteres Mal die Kurbelwellen-Geometrie mit $n = 102\,976$ Freiheitsgraden. Das Hybrid-Verfahren wird wieder für zwei verschiedene Quadraturordnungen $m = 8$ und $m = 12$ durchgeführt und der Interpolation gegenüber gestellt. Die entsprechenden Kurven sind in Abbildung 10.16 dargestellt, erneut der Speicherbedarf in der linken Teilabbildung und die Berechnungszeit in der rechten Teilabbildung.

Die höhere Quadraturordnung $m = 12$ bei dem Hybrid-Verfahren liefert etwas mehr Genauigkeit, lässt das Verfahren aber deutlich länger laufen, als bei $m = 8$, so dass man entscheiden muss, ob einem die relative Genauigkeit von knapp 10^{-13} ausreicht, oder ob man bereit ist, die längere Berechnungszeit für eine relative Genauigkeit von 10^{-14} in Kauf zu nehmen, die fast auf Maschinengenauigkeit exakt ist.

10.2 Hybrid-Verfahren für \mathcal{H}^2 -Matrizen

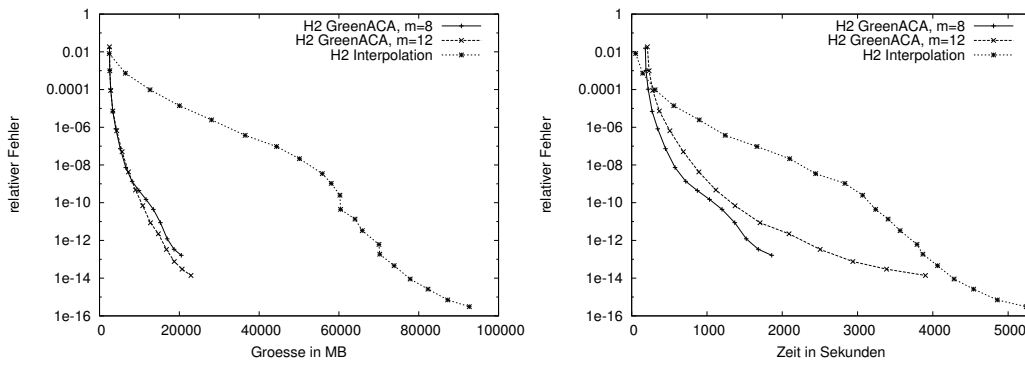


Abbildung 10.16: Kurbelwellen-Geometrie, $n = 102\,976$: Green-Hybrid-Verfahren auf \mathcal{H}^2 -Matrizen mit Quader-Parametrisierung, maximaler Blattgröße $clf = 128$ für ein festes $m = 8$, $\ell = 1$, $\delta = 1.0$ im Vergleich zur Interpolation auf \mathcal{H}^2 -Matrizen mit mit $clf = 2k$: Speicherbedarf (links) und Zeit (rechts)

Für welche Quadraturordnung man sich auch entscheiden mag, in jedem Fall ist das Green-Hybrid-Verfahren der Interpolation weit überlegen was den Speicherbedarf angeht. So braucht die Interpolation vier bis fünf mal soviel Speicher wie das Hybrid-Verfahren.

Auch in der benötigten Zeit für den Aufbau der jeweiligen \mathcal{H}^2 -Matrix schneidet das Hybrid-Verfahren deutlich besser ab als die Interpolation. Nur wenn hohe Quadraturordnungen gewählt werden, um an die Maschinengenauigkeit zu stoßen, ist das Hybrid-Verfahren nur noch wenig schneller als die Interpolation.

Diese erfreulichen Daten zum Green-Hybrid-Verfahren für \mathcal{H}^2 -Matrizen sind ein guter Abschluss dieses Kapitels.

Anhand der hier gezeigten numerischen Experimente kann man den Fortschritt von der theoretischen Idee der Green-Quadratur, die in dem einfachen Green-Quadratur-Verfahren ihre erste Anwendung fand, über die Weiterentwicklung zum Green-Hybrid-Verfahren bis hin zum Green-Hybrid-Verfahren auf \mathcal{H}^2 -Matrizen gut verfolgen.

An weiteren Verbesserungen des Verfahrens wurde bereits gearbeitet, welche aber nicht mehr Teil dieser Arbeit sind.

11 Zusammenfassung und Ausblick

Dieses Abschlusskapitel fasst noch einmal die wichtigsten Erkenntnisse und Ergebnisse zusammen und gibt eine kurze Übersicht über noch offene Fragen und Ansätze zur weiteren Erforschung und Verbesserung der Green-Quadratur-Idee.

Im Rahmen dieser Arbeit wurde eine neue Methode zur Niedrigrangapproximation für hierarchische Matrizen am Beispiel von Randintegralgleichungen vorgestellt.

Die Niedrigrangapproximation wird dabei durch die Konstruktion einer entarteten Kernfunktion erreicht, indem Gauß-Quadratur auf die durch die Green'sche Formel dargestellte Kernfunktion angewendet wird. Das danach benannte Green-Quadratur-Verfahren hat für m Quadraturpunkte im d -dimensionalen Raum eine Komplexität in der Größenordnung von m^{d-1} , da durch die Darstellung in der Green'schen Formel nur über den Rand integriert werden muss anstatt über das gesamte Gebiet und damit ein Faktor m für eine eingesparte Raumdimension entfällt. Damit hat das Verfahren auf jeden Fall einen theoretischen Vorteil gegenüber der Interpolation, der sich praktisch jedoch erst bei sehr großen Problemen und hohen Genauigkeiten zeigt.

Das Verfahren konvergiert exponentiell, wie durch Beweise belegt wurde.

Was jedoch in der Theorie so gut ausgesehen hatte, konnte die Erwartungen in praxisnahen Anwendungen nicht erfüllen. Während es im zweidimensionalen Fall noch ganz gute Ergebnisse lieferte, entpuppte es sich in der Anwendung im dreidimensionalen Fall als nicht konkurrenzfähig zu anderen gängigen Verfahren, welches auch durch numerische Experimente belegt wurde. Das Problem bei dem Green-Quadratur-Verfahren sind die hohen lokalen Ränge, die der Algorithmus im praxisrelevanten dreidimensionalen Fall mit sich bringt.

Aus der Notwendigkeit heraus, den Green-Quadratur-Ansatz zu verbessern, ist das Green-Hybrid-Verfahren entstanden.

Um die hohen lokalen Ränge zu reduzieren, wird die adaptive Kreuzapproximation auf die vom Green-Quadratur-Verfahren konstruierten Niedrigrangmatrizen angewendet, so dass ein hybrides Verfahren daraus entsteht. Eine zusätzliche Modifikation mit einem Interpolations-Ansatz ergibt dann den effizienten Green-Hybrid-Algorithmus.

Ein weiterer bedeutender Vorteil des Green-Hybrid-Verfahrens ist seine Anwendbarkeit auf \mathcal{H}^2 -Matrizen, welche mit dem ursprünglichen Verfahren so nicht möglich ist.

Wenn innerhalb des Verfahrens die adaptive Kreuzapproximation mit vollständiger Pivotierung gewählt wird, können auch hier gute Fehlerabschätzungen zum \mathcal{H} -Matrizen-Fall und zum \mathcal{H}^2 -Matrizen-Fall bewiesen werden.

Die numerischen Experimente zum Green-Hybrid-Verfahren haben vor allem für die \mathcal{H}^2 -Matrizen gute und konkurrenzfähige Ergebnisse geliefert, so dass das Ziel, einen

Algorithmus zu entwerfen, der mit den geläufigen Verfahren mithalten und sie sogar übertreffen kann, damit vollständig erreicht wurde.

Allerdings wurden mit der Einführung der Green-Quadratur-Idee sehr viel mehr Fragen aufgeworfen als in dieser Arbeit beantwortet werden konnten. Auch wenn hier am Ende der Ausführungen ein effizienter, zuverlässiger und konkurrenzfähiger Algorithmus steht, muss dies noch lange nicht das endgültige Ergebnis sein.

So besteht zum Beispiel noch die Möglichkeit, das Green-Hybrid-Verfahren für \mathcal{H}^2 -Matrizen zu einem Green-Hybrid-Kern-Verfahren zu modifizieren, das mit zusätzlicher Interpolation dafür sorgt, dass die Kopplungsmatrizen der \mathcal{H}^2 -Matrizen statt Nahfeldeinträge nur noch Punktauswertungen der Kernfunktion sind. Dadurch kann nochmal Zeit eingespart werden.

Außerdem wurden die Green-basierten Verfahren in dieser Arbeit ausschließlich für das Modellproblem der Laplace-Gleichung betrachtet und nur die Möglichkeit aufgezeigt, dass sie auch in anderen Fällen anwendbar sind. Damit wäre die Anwendbarkeit auf andere Randelement-Probleme wie zum Beispiel die Helmholtz-Differentialgleichung, Stokes, die Maxwell-Gleichungen oder auch, wie in Abschnitt 2.2.2 schon angedeutet, die Lamé-Gleichung, in Zukunft noch weiter zu erforschen.

Jedoch sollten die ersten Bemühungen über diese Arbeit hinaus in die Entwicklung eines effizienten Parameter-Schätzers investiert werden.

Um die Nützlichkeit eines solchen Schätzers zu motivieren, soll zunächst an die numerischen Experimente aus Kapitel 7 und besonders an die aus Kapitel 10 erinnert werden.

Einige Parameter wurden dabei von Anfang an festgehalten, wie zum Beispiel die Anzahl der Quadraturpunkte q der inneren Quadratur, die dazu dient, die Integrale aus (5.10) beziehungsweise aus (5.13) zu lösen. Da es in den Experimenten nur um die Approximationsgüte der Matrizen durch den jeweils betrachteten Algorithmus geht, war $q = 2$ festgesetzt.

Andere Parameter wurden nach den Ergebnissen der Tests festgelegt: Es stellt sich zum Beispiel heraus, dass im Zwei- wie auch im Dreidimensionalen die Anzahl der Teilintervalle bei $\ell = 1$ und die Wahl des Abstands-Parameters $\delta = \text{diam}_{\max}(\mathcal{B}_\tau)$ immer zu den besten Resultaten führen.

Aber es gibt auch Parameter, für die es keine allgemeingültige *beste Wahl* gibt. So genügt dem Green-Hybrid-Verfahren auf der Kreis-Geometrie eine Quadraturordnung von $m = 4$, um gute Approximationen zu erreichen. Sobald die zugrunde liegende Geometrie komplizierter ist, also zum Beispiel die Hilbertkurven-Geometrie, muss eine höhere Quadraturordnung von $m = 8$ gewählt werden, um zu ähnlich guten Ergebnissen zu gelangen. Im Dreidimensionalen sieht es auch so aus: Auf der Kugel-Geometrie ist $m = 8$ ausreichend, auf der komplizierteren Kurbelwellen-Geometrie braucht es aber eine Quadraturordnung von $m = 12$ für ebenso gute Resultate.

Diese Beispiele sind bisher alle nur durch das Aufstellen der \mathcal{H} -Matrix beziehungsweise der \mathcal{H}^2 -Matrix gegeben. Wenn man nun zum Lösen des Gleichungssystems übergeht, um aus gegebenen Dirichlet-Werten die Neumann-Werte zu berechnen, muss die Wahl der Parameter erneut überdacht werden.

Gemeint ist damit das Gleichungssystem

$$Vx = \left(K + \frac{1}{2}M \right) y$$

mit der Matrix $V \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}}$ des Einfachschichtpotentials, der Matrix $K \in \mathbb{R}^{\mathcal{I} \times \mathcal{J}}$ des Doppelschichtpotentials, der Massematrix $M \in \mathbb{R}^{\mathcal{I} \times \mathcal{J}}$ und den Koeffizientenvektoren $x \in \mathbb{R}^{\mathcal{I}}$ und $y \in \mathbb{R}^{\mathcal{J}}$.

Ohne näher auf dieses Problem einzugehen, sollen hier nur die Parameter-Werte angegeben werden, die nötig sind, um brauchbare Resultate zu erzielen. Es bleibt bei $\ell = 1$ und $\delta = \text{diam}_{\max}(\mathcal{B}_\tau)$. Diese beiden Werte sind auch hier die optimale Wahl. Die anderen empirisch ermittelten Werte sind in der folgenden Tabelle 11.1 zusammengefasst.

| Kugel-Geometrie | Kurbelwellen-Geometrie |
|--|---|
| $q = 3$ ($q = 5$ für singuläre Fälle) | $q = 6$ ($q = 8$ für singuläre Fälle) |
| $m = 2$ bis $n = 2^{19} = 524\,288$ $m = 3$ bis $n = 2^{21} = 2\,097\,152$ | $m = 3$ |
| $\varepsilon_{aca} = 5 \cdot 10^{-4}$ für $n = 2^{11} = 2\,048$ $\varepsilon_{aca} = 1 \cdot 10^{-4}$ für $n = 2^{13} = 8\,192$ $\varepsilon_{aca} = 1 \cdot 10^{-5}$ für $n = 2^{15} = 32\,768$ $\varepsilon_{aca} = 5 \cdot 10^{-6}$ für $n = 2^{17} = 131\,072$ $\varepsilon_{aca} = 1 \cdot 10^{-6}$ für $n = 2^{19} = 524\,288$ $\varepsilon_{aca} = 5 \cdot 10^{-7}$ für $n = 2^{21} = 2\,097\,152$ | $\varepsilon_{aca} = 1 \cdot 10^{-5}$ für $n = 25\,744$ $\varepsilon_{aca} = 1 \cdot 10^{-6}$ für $n = 102\,976$ $\varepsilon_{aca} = 1 \cdot 10^{-7}$ für $n = 411\,904$ |

Tabelle 11.1: Empirisch ermittelte Parameter-Werte

Man ist also auf empirische Daten zur Parameterwahl für das Green-Hybrid-Verfahren angewiesen, wenn man das volle Potential des Verfahrens nutzen will.

Daher ist die Entwicklung eines guten Parameter-Schätzers ein sinnvoller nächster Schritt im Ausbau den Green-Hybrid-Verfahrens. Mit einem solchen wäre die optimale Leistung des Verfahrens auch für noch unbekannte Geometrien und Problemstellungen garantiert.

Diese Ansätze zur weiteren Erforschung des Green-Hybrid-Verfahrens haben das Potential das bereits konkurrenzfähige Verfahren noch weiter zu verbessern, ohne die Beweisbarkeit in Frage zu stellen.

Index

- asymptotisch glatt, 17
- Blockclusterbaum, 11
- Bounding Box, 16
- Clusterbasis, 19
- Clusterbaum, 10
- Doppelschichtpotential (DLP), 7
- Einfachschichtpotential (SLP), 6
- Entartete Kernfunktion, 17
- Fernfeld, 12
- Green'sche Formel, 38
- \mathcal{H} -Matrix, 14
- \mathcal{H}^2 -Matrix, 19
- Interpolation, 21
- Kopplungsmatrix, 19
- Kreuzapproximation, 30
 - adaptive Kreuzapproximation, 32
- Multipol-Methode, 27
- Nahfeld, 12
- Quadraturformel, 41
 - Gauß-Quadraturformel, 41, 57
- Rang- k -Matrix, 13
- Transfermatrix, 19
- uniforme \mathcal{H} -Matrix, 20
- Zulässigkeitsbedingung
 - η -Zulässigkeit, 23
 - Definition, 12
 - Kreis-Zulässigkeitsbedingung, 29
 - max-Zulässigkeitsbedingung, 31, 39

Literaturverzeichnis

- [1] C. R. Anderson. An implementation of the fast multipole method without multipoles. *SIAM J. Sci. Stat. Comp.*, 13:923–947, 1992.
- [2] M. Bebendorf. Approximation of boundary element matrices. *Numer. Math.*, 86(4):565–589, 2000.
- [3] M. Bebendorf and R. Grzhibovskis. Accelerating Galerkin BEM for linear elasticity using adaptive cross approximation. *Math. Meth. Appl. Sci.*, 29:1721–1747, 2006.
- [4] M. Bebendorf and S. Rjasanow. Adaptive Low-Rank Approximation of Collocation Matrices. *Computing*, 70(1):1–24, 2003.
- [5] M. Bebendorf and R. Venn. Constructing nested bases approximations from the entries of non-local operators. Technical Report 491, Bonn University, SFB 611, 2012. to appear in *Numer. Math.*
- [6] G. Beylkin, R. Coifman, and V. Rokhlin. The fast wavelet transform and numerical algorithms. *Comm. Pure and Appl. Math.*, 44:141–183, 1991.
- [7] S. Börm. *Efficient Numerical Methods for Non-local Operators: \mathcal{H}^2 -Matrix Compression, Algorithms and Analysis*, volume 14 of *EMS Tracts in Mathematics*. EMS, 2010.
- [8] S. Börm. Numerik nicht-lokaler Operatoren. Vorlesungsskript, Stand: SS 2010.
- [9] S. Börm and L. Grasedyck. Low-rank approximation of integral operators by interpolation. *Computing*, 72:325–332, 2004.
- [10] S. Börm and L. Grasedyck. Hybrid cross approximation of integral operators. *Numer. Math.*, 101:221–249, 2005.
- [11] Steffen Börm and Jessica Gördes. Low-rank approximation of integral operators by using the green formula and quadrature. *Numerical Algorithms*, pages 1–26, 2013.
- [12] W. Dahmen, H. Harbrecht, and R. Schneider. Compression techniques for boundary integral equations — Asymptotically optimal complexity estimates. *SIAM J. Numer. Anal.*, 43(6):2251–2271, 2006.
- [13] W. Dahmen and R. Schneider. Wavelets on manifolds I: Construction and domain decomposition. *SIAM J. Math. Anal.*, 31:184–230, 1999.

- [14] Julia Falke. Niedrig-Rang-Approximation von hierarchischen Matrizen durch Anwendung der Greenschen Formel und der Quadratur. Master's thesis, Christian-Albrechts-Universität zu Kiel, 2013. in preparation.
- [15] K. Giebermann. Multilevel approximation of boundary integral operators. *Computing*, 67:183–207, 2001.
- [16] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. A theory of pseudoskeleton approximations. *Lin. Alg. Appl.*, 261:1–22, 1997.
- [17] S. A. Goreinov, N. L. Zamarashkin, and E. E. Tyrtyshnikov. Pseudo-skeleton approximations by matrices of maximal volume. *Mathematical Notes*, 62:515–519, 1997.
- [18] G. Green. An essay on the application of mathematical analysis to the theories of electricity and magnetism. Nottingham, 1828.
- [19] L. Greengard, D. Gueyffier, P.-G. Martinsson, and V. Rokhlin. Fast direct solvers for integral equations in complex three-dimensional domains. *Acta Numerica*, 18:243–275, 2009.
- [20] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comp. Phys.*, 73:325–348, 1987.
- [21] L. Greengard and V. Rokhlin. A new version of the fast multipole method for the Laplace in three dimensions. In *Acta Numerica 1997*, pages 229–269. Cambridge University Press, 1997.
- [22] W. Hackbusch. *Hierarchische Matrizen — Algorithmen und Analysis*. Springer, 2009.
- [23] W. Hackbusch and Z. P. Nowak. O cloznosti metoda panelej. In G. I. Marchuk, editor, *Vycislitel'nye prozessy i sistemy*, pages 233–244. Nauka, Moskau, 1988.
- [24] W. Hackbusch and Z. P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numer. Math.*, 54:463–491, 1989.
- [25] J. Makino. Yet another fast multipole method without multipoles — pseudoparticle multipole method. *J. Comp. Phys.*, 151:910–920, 1999.
- [26] G. Of, O. Steinbach, and W. L. Wendland. The fast multipole method for the symmetric boundary integral formulation. *IMA J. Numer. Anal.*, 26:272–296, 2006.
- [27] R. Plato. *Numerische Mathematik kompakt: Grundlagenwissen für Studium und Praxis*. Numerische Mathematik. Vieweg, 2000.
- [28] T. J. Rivlin. *The Chebyshev Polynomials*. Wiley-Interscience, New York, 1984.
- [29] S. Rjasanow and O. Steinbach. *The Fast Solution of Boundary Integral Equations*. Mathematical and Analytical Techniques with Applications to Engineering. Springer, 2007.

- [30] V. Rokhlin. Rapid solution of integral equations of classical potential theory. *J. Comp. Phys.*, 60:187–207, 1985.
- [31] S. A. Sauter. Variable order panel clustering. *Computing*, 64:223–261, 2000.
- [32] R. Schneider. *Multiskalen- und Wavelet-Matrixkompression: Analysisbasierte Methoden zur Lösung großer vollbesetzter Gleichungssysteme*. B.G. Teubner, Stuttgart, 1998.
- [33] O. Steinbach. *Numerische Näherungsverfahren für elliptische Randwertprobleme*. Teubner, 2003.
- [34] E. E. Tyrtyshnikov. Incomplete cross approximation in the mosaic-skeleton method. *Computing*, 64:367–380, 2000.
- [35] L. Ying, G. Biros, and D. Zorin. A kernel-independent adaptive fastmultipole algorithm in two and three dimensions. *J. Comp. Phys.*, 196(2):591–626, 2004.
- [36] Lexing Ying, George Biros, Denis Zorin, and Harper Langston. A new parallel kernel-independent fast multipole method. In *in SC2003*, page 14. IEEE Computer Society, 2003.

Erklärung

Hiermit versichere ich,

- a) dass diese Abhandlung, abgesehen von der Beratung durch den Betreuer Steffen Börm, nach Inhalt und Form meine eigene ist,
- b) dass diese Arbeit zum Teil an einer anderen Stelle bereits veröffentlicht bzw. zur Veröffentlichung eingereicht wurde. Im Detail:
 - (i) *Low-rank approximation of integral operators by using the Green formula and quadrature* zusammen mit Steffen Börm, Numerical Algorithms, Springer US, January 2013, 1–26, veröffentlicht
 - (ii) *Efficient low-rank approximation of integral operators with Green's formula and cross approximation* zusammen mit Steffen Börm und Sven Christophersen, 2012, eingereicht, aber nicht veröffentlicht

Diese Artikel weisen einen Eigenanteil gemäß des Schreibens meines Betreuers vor.

- c) und dass diese Arbeit unter Einhaltung der Regeln guter wissenschaftlicher Praxis der Deutschen Forschungsgemeinschaft entstanden ist.

Kiel, den 3. Februar 2014