Universität Leipzig
Fakultät für Mathematik und Informatik
Mathematisches Institut

Steigerung der Effizienz Hierarchischer Matrizen durch Verwendung gemeinsamer Basen

Diplomarbeit

Leipzig, 29. April 2010

vorgelegt von
Roxana Bujack
Diplomstudiengang Mathematik

Betreuender Hochschullehrer: Prof. Dr. Mario Bebendorf (Universität Bonn, Institut für Numerische Simulation)

Motivation und Zielstellung

Viele physikalische Probleme führen zu Randwertproblemen. Dabei gilt es die Lösung einer Differentialgleichung zu finden, so dass auf dem Rand vorgegebene Funktionswerte, die so genannten Randbedingungen, angenommen werden. Differentialgleichungen können nur in wenigen Spezialfällen analytisch gelöst werden. Man muss also auf numerische Verfahren zurückgreifen.

Ein Problem aus der Praxis ist in der Regel von zu hoher Komplexität. Wir können daher nicht davon ausgehen ein Black-Box-Verfahren zu finden, welches jede Differentialgleichung innerhalb akzeptabler Zeit löst. Deshalb brauchen wir auf die Problemklassen zugeschnittene Verfahren, welche ihre speziellen Eigenschaften ausnutzen.

Wir beschränken uns hier auf elliptische Randwertprobleme. Sie werden zu Integralgleichungen umformuliert, mittels Randelementmethode diskretisiert und damit in ein lineares Gleichungssystem überführt. Zur Behandlung des Gleichungssystems bedienen wir uns Hierarchischer Matrizen.

Obwohl diese bereits effektive Hilfsmittel darstellen, wollen wir versuchen ihre Effizienz durch Verwendung gemeinsamer Basen weiter zu steigern.

Danksagung

Ich bedanke mich ganz herzlich bei meinem Betreuer, Prof. Dr. Mario Bebendorf, der mir nicht nur die Möglichkeit gab, diese Arbeit zu schreiben, sondern von dem ich auch fast alles gelernt habe, was dafür nötig war.

Vielen Dank auch an alle Mitarbeiter am Institut, insbesondere Michael Bratsch, die mir geholfen haben, wenn ich Fragen oder Probleme hatte.

Ganz besonders danke ich meiner Familie und meinen Freunden, die mich immer unterstützt haben.

Danke!

Inhaltsverzeichnis

Kapitel 1

Grundlagen

1.1 Einführung

Wir werden in dieser Arbeit sehen, wie Randwertprobleme in Integralgleichungen umgeformt werden, die mit Hilfe der Randelementmethode diskretisiert und somit approximativ durch lineare Gleichungssysteme gelöst werden können.
Da die Zahl der Freiheitsgrade im Allgemeinen zu groß ist, als dass ihre herkömmliche Lösung in annehmbarer Zeit zu erwarten ist, benötigen wir angepasste Algorithmen und Datenstrukturen.

Hierarchische Matrizen erlauben das Speichern einer Vielzahl von Problemen, die normalerweise quadratische Komplexität hätten, mit fast linearem Aufwand. Darüber hinaus ermöglichen sie oft die Durchführung algebraischer Operationen auf den Approximanten in einem Bruchteil der ursprünglich nötigen Zeit. Ihre Erzeugung beinhaltet die Zerlegung des diskretisierten Berechnungsgebietes in approximierbare Bereiche und die anschließende Berechnung der zu speichernden Einträge des Approximanten. Wir werden verschiedene Varianten für beides vorstellen.

Besonderes Augenmerk wird auf dem speziell an das ACA-Verfahren angepasste Rekompressionsverfahren RACA liegen, welches mit Hilfe gemeinsamer Basen den schon sehr guten Speicheraufwand weiter verringert. Wir werden theoretisch und experimentell seine Effizienz auf der Steifigkeitsmatrix des Doppelschichtpotentials des Laplaceoperators untersuchen. Die Schwierigkeit im Vergleich zum Einfachschichtpotential wird in der Abhängigkeit der Glattheit des Operators von der Beschaffenheit der Geometrie liegen.

Eine weitere Herausforderung, der wir uns in dieser Arbeit stellen werden, ist die

Kompression von Steifigkeitsmatrizen, die vom Helmholtzoperator stammen. Dieser Differentialoperator beschreibt Schwingungen und ist in seiner Behandlung sehr aufwendig, da seine Singularitätenfunktion stark oszilliert. Der Grad der Oszillation steigt mit der Frequenz. Wir untersuchen eine Möglichkeit der Zerlegung der Geometrie, die Konvergenz der Approximanten unabhängig von der Wellenzahl garantiert, ihre Komplexität und ihre Kompatibilität mit Verfahren zur Berechnung der Einträge.

1.2 Hilfsmittel

1.2.1 Multiindexschreibweise

Der Übersichtlichkeit halber werden wir häufig mehrere Indizes $j_1, j_2, ..., j_d \in \mathbb{N}$ zu einem Multiindex $\boldsymbol{j}$ zusammenfassen. Er ist ein $d-$Tupel natürlicher Zahlen, $\boldsymbol{j} = (j_1, j_2, ..., j_d) \in \mathbb{N}^d$, wobei die folgenden Notationen gelten:

$$\begin{aligned}
\boldsymbol{j} &= n \Leftrightarrow j_1 = n, j_2 = n, ..., j_d = n, \\
\boldsymbol{j} &< n \Leftrightarrow j_1 < n, j_2 < n, ..., j_d < n, \\
|\boldsymbol{j}| &= j_1 + j_2 + ... + j_d, \\
\boldsymbol{j} - e_i &= (j_1, ..., j_{i-1}, j_i - 1, j_{i+1}, ..., j_d), \\
\partial^{\boldsymbol{j}} f &= \frac{\partial^{|\boldsymbol{j}|} f}{(\partial x)^{\boldsymbol{j}}} = (\frac{\partial}{\partial x_1})^{j_1} ... (\frac{\partial}{\partial x_d})^{j_d} f.
\end{aligned}$$

1.2.2 Tschebyscheffpolynome

Aufgrund ihrer guten numerischen Eigenschaften werden wir Tschebyscheffpolynome als Basis bei Interpolation und Matrixapproximation verwenden. Das Tschebyscheffpolynom vom Grad $p \in \mathbb{N}$ in einer eindimensionalen Variablen $\tau \in [-1, 1]$ hat die Form

$$T_p(\tau) = \cos\big(p \arccos(\tau)\big), \tag{1.1}$$

und die p Nullstellen

$$\mathring{\tau}_i = \cos\left(\frac{2i+1}{2p}\pi\right),\ i = 0, ..., p-1,$$

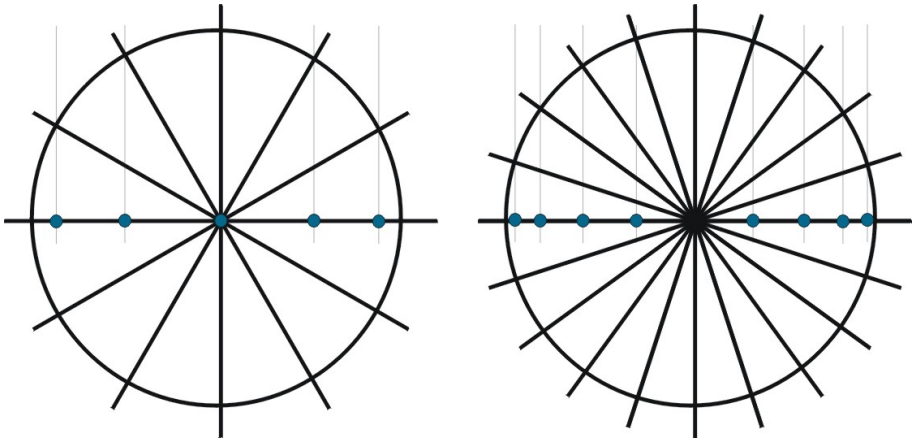

Abbildung 1.1: Lage der Tschebyscheffknoten für $p = 5$ und $p = 9$

welche Tschebyscheffknoten genannt werden. Sie liegen am Rand des Intervalls dichter als in seiner Mitte, siehe Abbildung 1.1.

$T_p(\tau)$ kann auch rekursiv durch die folgende Iterationsvorschrift bestimmt werden:

$$\begin{aligned} T_0(\tau) &= 1, \\ T_1(\tau) &= \tau, \\ T_{i+1}(\tau) &= 2\tau T_i(t) - T_{i-1}(\tau) \text{ für } i = 1, 2, \dots . \end{aligned} \tag{1.2}$$

Man kann diese Tschebyscheffpolynome (1.1) bei der Interpolation einer Funktion $f(t) \in C[a, b]$ als Basis wählen, wenn man $f(t)$ durch die Variablentransformation $\tau = 2\frac{t-a}{b-a} - 1 \, , \forall t \in \mathbb{R}$ in eine Funktion in $C[-1, 1]$ umwandelt. Es ist jedoch handlicher diese Transformation in die Formel für die Polynominterpolation zu integrieren. Die Tschebyscheffknoten vom Grad $p \in \mathbb{N}$ auf einem Intervall $[a, b] \subset \mathbb{R}$ liegen somit für $i = 0, ..., p - 1$ bei

$$\begin{aligned} \mathring{t}_i &= \frac{a+b}{2} + \frac{b-a}{2} \cos\left(\frac{2i+1}{2p}\pi\right), \\ &= \frac{a+b}{2} + \frac{b-a}{2} \mathring{\tau}_i. \end{aligned} \tag{1.3}$$

Wir betrachten die Tschebyscheffpolynominterpolation als Operator $\mathcal{I}_p : C[a, b] \to \Pi_{p-1}$, $\mathcal{I}_p(f(t)) = q(t)$, mit einem Polynom $q(t) \in \Pi_{p-1}$ vom Grad $p - 1$, das in den Tschebyscheffknoten (1.3) mit den Funktionswerten von $f(t)$ übereinstimmt, das heißt $f(\mathring{t}_j) = q(\mathring{t}_j)$, für $j = 0, ..., p - 1$. Der Operator wirkt

folgendermaßen auf $f(t)$:

$$\mathcal{I}_p(f(t)) = \sum_{i=0}^{p-1} c_i T_i(2\frac{t-a}{b-a} - 1), \tag{1.4}$$

mit eindeutig bestimmten Koeffizienten

$$c_i = \frac{2-\delta(i)}{p} \sum_{i=0}^{p-1} f(\mathring{t}_j) \cos\left(\frac{2j+1}{2p} i\pi\right).$$

Funktionen in höherer Dimension $f(t) \in C(D), t = \begin{pmatrix} t_1 \\ \vdots \\ t_d \end{pmatrix}$, wobei das d-dimensionale Gebiet $D = \bigotimes_{\nu=1}^{d} [a_\nu, b_\nu]$ Tensorprodukt von eindimensionalen Intervallen sein soll, können auch mit Hilfe der Tschebyscheffpolynome interpoliert werden. Dafür benutzen wir sogenannte Tensorprodukttschebyscheffknoten

$$\mathring{t}_{\boldsymbol{j}} = \bigotimes_{\nu=1}^{d} \mathring{t}_{j_\nu},$$

wobei $\boldsymbol{j} \in \mathbb{N}^d$ ein Multiindex mit $0 \leq j_\nu < p$, für $\nu = 1, ..., d$, ist. Sie entsprechen in jeder Dimension dem j_ν-ten, eindimensionalen Tschebyscheffknoten (1.3).

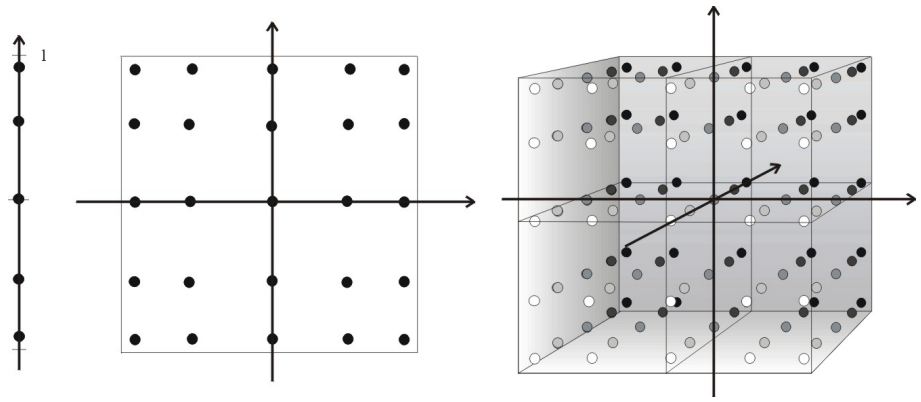

Abbildung 1.2: Lage der Tensortschebyscheffknoten für $p = 5$ in einer, zwei und drei Dimensionen

Den zugehörigen Interpolationsoperator erhält man durch Nacheinanderanwenden der oben definierten 1D-Operatoren (1.4) auf jede der d Koordinaten, also

$$\boldsymbol{\mathcal{I}}_p : [a,b]^d \to \Pi_{p-1}^d,\ \boldsymbol{\mathcal{I}}_p f(t) = \mathcal{I}_p^{(1)} ... \mathcal{I}_p^{(d)} f(t).$$

Dabei ist unter $\mathcal{I}_p^{(\nu)}, \nu = 1, ..., d$, die Interpolation der ν-ten Koordinate zu verstehen,

$$\mathcal{I}_p^{(\nu)} f(\begin{pmatrix} t_1 \\ \vdots \\ t_\nu \\ \vdots \\ t_d \end{pmatrix}) = \sum_{i=0}^{p-1} c_i^{(\nu)} T_i(2\frac{t_\nu - a_\nu}{b_\nu - a_\nu} - 1),$$

mit

$$c_i^{(\nu)} = \frac{2 - \delta(i)}{p} \sum_{j=0}^{p-1} f(\begin{pmatrix} t_1 \\ \vdots \\ \mathring{t}_\nu \\ \vdots \\ t_d \end{pmatrix}) \cos\left(\frac{2j+1}{2p} i\pi\right).$$

Das d-dimensionale Interpolationspolynom $\boldsymbol{\mathcal{I}}_p f(t)$ hat dann folgendes Aussehen:

$$\boldsymbol{\mathcal{I}}_p f(t) = \sum_{\boldsymbol{i} \in \mathbb{N}^d, \boldsymbol{i} < p} c_{\boldsymbol{i}} \prod_{\nu=1}^{d} T_{i_\nu}(2\frac{t_\nu - a_\nu}{b_\nu - a_\nu} - 1), \tag{1.5}$$

mit

$$c_{\boldsymbol{i}}^{(\nu)} = \prod_{\nu=1}^{d} \frac{2 - \delta(i_\nu)}{p} \cdot \sum_{\boldsymbol{j} \in \mathbb{N}^d, \boldsymbol{j} < p} f(\mathring{t}_{\boldsymbol{j}}) \prod_{\nu=1}^{d} \cos\left(\frac{2j_\nu + 1}{2p} i_\nu \pi\right).$$

Obwohl es der $2d$-dimensionale Fall in einer Variable schon impliziert, wollen wir aus Bezeichnungsgründen ganz kurz die d-dimensionale Tschebyscheffpolynominterpolation in zwei Variablen $x \in D_x = \bigotimes_{\nu=1}^{d} [a_\nu^x, b_\nu^x]$ und $y \in D_y = \bigotimes_{\nu=1}^{d} [a_\nu^y, b_\nu^y]$ angeben. Seien $\xi, \epsilon \in \mathbb{R}^d$,

$$\begin{aligned} \xi_\nu &= 2\frac{x_\nu - a_\nu^x}{b_\nu^x - a_\nu^x} - 1, \\ \epsilon_\nu &= 2\frac{y_\nu - a_\nu^y}{b_\nu^y - a_\nu^y} - 1, \end{aligned}$$

die auf den Einheitswürfel transformierten Vektoren und $\mathring{x}_{\boldsymbol{i}}, \mathring{y}_{\boldsymbol{i}}$ die Ergebnisse der inversen Transformationen

$$\begin{aligned} \mathring{x}_{i_\nu} &= \frac{a_\nu^x + b_\nu^x}{2} + \frac{b_\nu^x - a_\nu^x}{2} \mathring{\tau}_{i_\nu}, \\ \mathring{y}_{i_\nu} &= \frac{a_\nu^y + b_\nu^y}{2} + \frac{b_\nu^y - a_\nu^y}{2} \mathring{\tau}_{i_\nu}, \end{aligned}$$

$\nu = 1, ..., d$, der Tschebyscheffknoten auf die Bereiche D_x, bzw. D_y, dann ist

$$\boldsymbol{\mathcal{I}}_p f(x,y) = \boldsymbol{\mathcal{I}}_p^x \boldsymbol{\mathcal{I}}_p^y f(x,y) = \sum_{\boldsymbol{i},\boldsymbol{j} \in \mathbb{N}^d, i_\nu, j_\nu < p} c_{\boldsymbol{ij}} \prod_{\nu=1}^{d} T_{i_\nu}(\xi_\nu) \cdot T_{j_\nu}(\epsilon_\nu),$$

wobei

$$c_{\boldsymbol{ij}} = \prod_{\nu=1}^{d} \frac{(2-\delta(i_\nu))(2-\delta(j_\nu))}{p^2} \sum_{\boldsymbol{k},\boldsymbol{l} \in \mathbb{N}^d, k_\nu, l_\nu < p} f(\mathring{x}_{\boldsymbol{k}}, \mathring{y}_{\boldsymbol{l}}) \prod_{\nu=1}^{d} \cos\left(\frac{2k_\nu + 1}{2p} i_\nu \pi\right) \cos\left(\frac{2l_\nu + 1}{2p} j_\nu \pi\right)$$

als Einträge einer Koeffizientenmatrix $C \in \mathbb{C}^{p^d \times p^d}$ interpretiert werden sollen.

Die bereits erwähnten guten Eigenschaften dieser Polynome sind einerseits die Möglichkeit sie mit Hilfe der Rekursionsformel (1.2) sehr schnell aufzustellen, andererseits numerische Stabilität.

Satz 1.2.1. *Sei $f \in C^p[a,b]$, so ist die Polynominterpolation $\mathcal{I}_p f$ auf den Tschebyscheffknoten eindeutig durch (1.4) gelöst und der Fehler durch*

$$\|f - \mathcal{I}_p f\|_{C[a,b]} \leq 2 \frac{(b-a)^p}{4^p p!} \|f^{(p)}\|_{C[a,b]}$$

beschränkt. Außerdem hängt die Operatornorm $\|\mathcal{I}_p\| = \max_{\|f\|_{C[a,b]}=1} \|\mathcal{I}_p f\|_{C[a,b]}$ mit

$$\|\mathcal{I}_p\| \leq 1 + \frac{2}{\pi} \log p$$

nur logarithmisch von p ab.

Satz 1.2.2. *Der Interpolationsfehler bei der mehrdimensionalen Tschebyscheff-interpolation $\boldsymbol{\mathcal{I}}_p : C(\bigotimes_{\nu=1}^{d} [a_\nu, b_\nu]) \to \Pi_{p-1}^d$ ist durch*

$$\|f - \boldsymbol{\mathcal{I}}_p f\|_{C(D)} \leq \left(1 + \frac{2}{\pi} \log p\right)^{d-1} \sum_{\nu=1}^{d} \|f - \mathcal{I}_p^{(\nu)} f\|_{C(D)}$$

nach oben begrenzt.

Für Beweise und mehr Informationen sei auf [1] und [2] verwiesen.

1.2.3 Gaußkubatur auf Dreiecken

Zur numerischen Berechnung von Integralen werden wir, aufgrund ihrer Effektivität, die Gaußquadratur benutzen. Sie bietet in Abhängigkeit der Freiheitsgrade g eine optimale Approximation. Das eindimensionale Integral wird dabei durch die Summe über g, mit w_i gewichtete Funktionswerte $f_i = f(x_i),\ i = 1, ..., g$, an den Stützstellen x_i annähernd bestimmt.

$$\int_a^b f(x)dx \approx \sum_{i=1}^{g} f_i w_i$$

Die Stützstellen x_i sind die Nullstellen des Legendre-Polynoms vom Grad g und die Gewichte ergeben sich durch Integration der Lagrangebasis,

$$w_i = \int_a^b \prod_{j=1, j\neq i}^{g} \frac{x - x_j}{x_i - x_j} dx, i = 1, ..., g.$$

Bei der Approximation mehrdimensionaler Integrale kann man auch auf Tensorproduktknoten zurückgreifen. Wir müssen aber Integrale über dreieckigen Grundflächen berechnen, weshalb diese Variante nicht von großem Nutzen ist.
Für uns sind speziell auf Dreiecke zugeschnittene Stützstellen sinnvoll und wir benutzen hier $g = 7$ symmetrisch angeordnete Knoten.

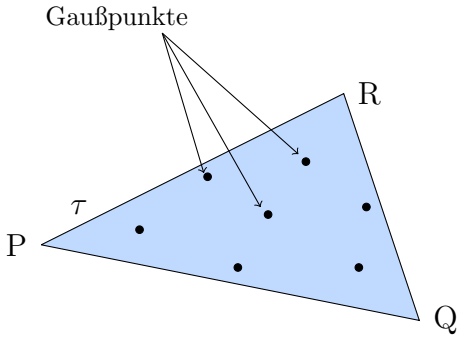

Abbildung 1.3: Verteilung von $g = 7$ Gaußknoten in einem Dreieck

Damit folgt für ein Dreieck τ mit den Eckpunkten P, Q, R, den Bezeichnungen $a = Q - P$ und $b = R - P$ für die beiden von P ausgehenden Kanten und dem

Flächeninhalt $A_{PQR} = \frac{1}{2}\sqrt{|a| + |b| - \langle a, b\rangle}$:

$$
\begin{aligned}
\int_{\tau} f(x)dx &\approx \sum_{i=1}^{7} w_i f(x_i) \\
&= \frac{A_{PQR}}{1200}\Bigg(270 f(P + \frac{1}{3}a + \frac{1}{3}b) \\
&+ (155 + \sqrt{15})\Big(f(P + \frac{6+\sqrt{15}}{21}a + \frac{6+\sqrt{15}}{21}b) + f(P + \frac{9+\sqrt{15}}{21}a + \frac{6+\sqrt{15}}{21}b) \\
&+ f(P + \frac{6+\sqrt{15}}{21}a + \frac{9+\sqrt{15}}{21}b)\Big) \\
&+ (155 - \sqrt{15})\Big(f(P + \frac{6-\sqrt{15}}{21}a + \frac{6-\sqrt{15}}{21}b) + f(P + \frac{9+2\sqrt{15}}{21}a + \frac{6-\sqrt{15}}{21}b) \\
&+ f(P + \frac{6-\sqrt{15}}{21}a + \frac{9+2\sqrt{15}}{21}b)\Big)\Bigg).
\end{aligned}
$$

Für nähere Informationen siehe [3].

Kapitel 2

Elliptische Randwertprobleme

Die meisten physikalischen Phänomene werden durch Differentialgleichungen beschrieben. Ihre Behandlung ist für die Modellierung der Welt unumgänglich. Eine wichtige Klasse sind die elliptischen Differentialgleichungen, die zum Beispiel die Ausbreitung akustischer Signale, Wirbelströme oder Wärmeleitung beschreiben können.

Sei $\Omega \subset \mathbb{R}^d$ ein d-dimensionales Lipschitz-Gebiet oder dessen Komplement, $f \in \mathcal{L}^2(\Omega)$ eine Funktion und g_D, sowie g_N die Beschreibungen der Dirichlet- bzw. der Neumann-Randbedingungen Γ_D, Γ_N auf dem Rand $\partial\Omega = \Gamma = \Gamma_D \cup \Gamma_N$, dann suchen wir Lösungen $u : \Omega \to \mathbb{R}^n$ der partiellen Differentialgleichung

$$\mathscr{L}u = f \text{ in } \Omega,$$

so dass gilt:

$$\begin{aligned} \gamma_0 u &= g_D \text{ auf } \Gamma_D, \\ \gamma_1 u &= g_N \text{ auf } \Gamma_N. \end{aligned}$$

Dabei ist $\mathscr{L}$ ein partieller Differentialoperator zweiter Ordnung

$$\mathscr{L}u = -\sum_{i,j=1}^{d} \partial_i(a_{ij}\partial_j)u + \sum_{i=1}^{d} a_i\partial_i u + au$$

mit reellen Koeffizientenfunktionen a_{ij}, a_i, a, so dass alle Eigenwerte der $d \times d$ Koeffizientenmatrix $A = (a_{ij})$ das gleiche Vorzeichen besitzen und von Null verschieden sind. Weiterhin bezeichnet $\gamma_0 : H^1(\Omega) \to H^{1/2}(\Gamma)$,

$$\gamma_0 u(x) = \lim_{\tilde{x}\in\Omega\to x\in\Gamma} u(\tilde{x}) \text{ für } x \in \Gamma,$$

den Spuroperator, und $\gamma_1 : H^1(\Omega) \to H^{-1/2}(\Gamma)$,

$$\gamma_1 u(x) = \lim_{\tilde{x}\in\Omega\to x\in\Gamma} \sum_{i,j=1}^{d} n_i(x) a_{ij} \partial_j u(\tilde{x}) = \langle A\mathbf{n}, \gamma_0 \nabla u \rangle \text{ für } x \in \Gamma,$$

die Konormalenableitung.
Das eben beschriebene Problem heißt gemischtes elliptisches Randwertproblem.

Beispiel 2.0.3. Der bekannteste elliptische Differentialoperator ist der Laplaceoperator

$$\Delta = \nabla^2 = \sum_{i=1}^{d} \partial_i^2. \tag{2.1}$$

Mit Hilfe der Laplacegleichung

$$\Delta u = 0$$

können viele wichtige physikalische Phänomene, wie zum Beispiel elektrische und gravitative Felder, beschrieben werden.

Beispiel 2.0.4. Der Helmholtzoperator

$$-\Delta - \kappa^2 \tag{2.2}$$

mit der Wellenzahl κ dient der Beschreibung von Wellen. Die Helmholtz-Differentialgleichung

$$\Delta u + \kappa^2 u = 0$$

ist ein Spezialfall der allgemeinen Wellengleichung und kann beispielsweise die Schwingungen auf einer Trommel beschreiben. Dabei entspricht eine reine Dirichletbedingung

$$u = 0 \text{ auf } \Gamma$$

der Tatsache, dass die Membran am Rand eingespannt ist.

2.1 Randintegralgleichungen

Wir betrachten in dieser Arbeit homogene Randwertprobleme der Form

$$\begin{aligned} &\mathscr{L}u = 0 \text{ in } \Omega, \\ &\gamma_0 u = g_D \text{ auf } \Gamma_D, \\ &\gamma_1 u = g_N \text{ auf } \Gamma_N \end{aligned} \tag{2.3}$$

mit einem Differentialopeartor

$$\mathscr{L}u = -\sum_{i,j=1}^{d} \partial_i(a_{ij}\partial_j)u + au \tag{2.4}$$

mit symmetrischer, positiv definiter $d \times d$ Matrix $A = (a_{ij})$. Für viele Anwendungsbereiche ist es nützlich (2.3) in eine Randintegralgleichung umzuformulieren. Dazu multiplizieren wir den Operator (2.4) mit einer hinreichend glatten Testfunktion v und integrieren anschließend über Ω:

$$\begin{aligned}\int_\Omega \mathscr{L}u(x)v(x)\,\mathrm{d}x &= \int_\Omega \Big(-\sum_{i,j=1}^{d} \partial_i(a_{ij}\partial_j)u(x) + au(x)\Big)v(x)\,\mathrm{d}x \\ &= -\int_\Omega \nabla(A\nabla u(x))v(x)\,\mathrm{d}x + \int_\Omega au(x)v(x)\,\mathrm{d}x.\end{aligned} \tag{2.5}$$

Satz 2.1.1 (Erste Greensche Formel)**.** *Seien $u \in H^2(\Omega)$, $v \in H^1(\Omega)$ und $A \in \mathbb{R}^{d\times d}$ positiv definit, dann gilt:*

$$\int_\Omega \nabla(A\nabla u(x))v(x)\,\mathrm{d}x = -\int_\Omega (A\nabla u(x))\nabla v(x)\,\mathrm{d}x + \int_\Gamma \langle A\mathbf{n}, \nabla u(x)\rangle v(x)\,\mathrm{d}s_x.$$

Wir wenden die Greensche Formel aus Satz 2.1.1 auf (2.5) an.

$$\begin{aligned}\int_\Omega \mathscr{L}u(x)v(x)\,\mathrm{d}x = &\int_\Omega (A\nabla u(x))\nabla v(x)\,\mathrm{d}x - \int_\Gamma \langle A\mathbf{n}, \nabla u(x)\rangle v(x)\,\mathrm{d}s_x \\ &+ \int_\Omega au(x)v(x)\,\mathrm{d}x\end{aligned} \tag{2.6}$$

Wenn wir zusätzlich $u \in H^2(\Omega)$ fordern, können wir die Rollen von u und v vertauschen

und (2.6) vom Ergebnis abziehen. Das ergibt die zweite Greensche Formel.

$$\begin{aligned}\int_\Omega \mathscr{L}v(x)u(x)\,\mathrm{d}x &= \int_\Omega \mathscr{L}u(x)v(x)\,\mathrm{d}x + \int_\Omega (A\nabla v(x))\nabla u(x)\,\mathrm{d}x - \int_\Gamma \langle A\mathbf{n}, \nabla v(x)\rangle u(x)\,\mathrm{d}s_x \\ &\quad + \int_\Omega av(x)u(x)\,\mathrm{d}x - \Big(\int_\Omega (A\nabla u(x))\nabla v(x)\,\mathrm{d}x \\ &\quad - \int_\Gamma \langle A\mathbf{n}, \nabla u(x)\rangle v(x)\,\mathrm{d}s_x + \int_\Omega au(x)v(x)\,\mathrm{d}x \Big) \\ &= \int_\Omega \mathscr{L}u(x)v(x)\,\mathrm{d}x - \int_\Gamma \langle A\mathbf{n}, \nabla v(x)\rangle u(x)\,\mathrm{d}s_x \\ &\quad + \int_\Gamma \langle A\mathbf{n}, \nabla u(x)\rangle v(x)\,\mathrm{d}s_x\end{aligned} \tag{2.7}$$

Dabei wurde die Symmetrie der Matrix A benutzt.

Definition 2.1.2. Die Fundamentallösung oder Singularitätenfunktion S eines partiellen Differentialoperators ist die Funktion, die folgende Bedingung erfüllt:

$$\int_\Omega (\mathscr{L}S)(x-y)u(y)\,\mathrm{d}y = u(x),\ \forall x \in \Omega.$$

Das entspricht der distributionellen Lösung der Gleichung $(\mathscr{L}S)(x) = \delta(x)$ mit der Delta-Distribution $\delta(x)$.

Beispiel 2.1.3. Die Singularitätenfunktion des Laplaceoperators (2.1) im Dreidimensionalen hat die Form:

$$S(x) = \frac{1}{4\pi\|x\|}. \tag{2.8}$$

Beispiel 2.1.4. Der Helmholtzoperator (2.2) hat im $\mathbb{R}^3$ die Singularitätenfunktion

$$S(x) = \frac{e^{i\kappa\|x\|}}{\|x\|}.$$

Ist die Fundamentallösung bekannt, erhält man die Darstellungsformel des

Differentialoperators, indem man $S(x-y)$ für $v(x)$ in (2.7) einsetzt:

$$\begin{aligned} u(x) &= \int_\Omega \mathscr{L}u(y)S(x-y)\,\mathrm{d}y + \int_\Gamma \langle A\mathbf{n}, \nabla u(y)\rangle S(x-y)\,\mathrm{d}y \\ &\quad - \int_\Gamma \langle A\mathbf{n}, \nabla_y S(x-y)\rangle u(y)\,\mathrm{d}y. \end{aligned}$$

Definition 2.1.5. Die linearen Integraloperatoren, das Newtonpotential $\mathscr{N}$, das Einfachschichtpotential $\mathscr{V}$ und das Doppelschichtpotential $\mathscr{K}$ sind wie folgt definiert:

$$\begin{aligned} (\mathscr{N}f)(x) &:= \int_\Omega S(x-y)f(y)dy, \\ (\mathscr{V}f)(x) &:= \int_\Gamma S(x-y)f(y)dy \text{ und} \\ (\mathscr{K}f)(x) &:= \int_\Gamma \langle A\mathbf{n}, \nabla_y S(x-y)\rangle f(y)dy. \end{aligned}$$

Diese Definitionen erlauben uns die Darstellungsformel kompakt als

$$u(x) = \mathscr{N}(\mathscr{L}u) + \mathscr{V}(\gamma_1 u) - \mathscr{K}(\gamma_0 u)$$

zu schreiben.
Aufgrund der speziellen Form der von uns betrachteten Randwertprobleme entfällt das Newtonpotential.
Es wird deutlich, dass u nun allein durch die Dirichlet- und die Neumannbedingungen, jeweils auf dem gesamten Rand $\partial\Omega$, eindeutig bestimmt ist.

$$u = \mathscr{V}(\gamma_1 u) - \mathscr{K}(\gamma_0 u) \tag{2.9}$$

Leider kennen wir beim gemischten Randwertproblem beide nur auf einem Teil des Randes. Wir wenden die Spuroperatoren auf (2.9) an

$$\begin{aligned} \gamma_0 u &= \gamma_0\mathscr{V}(\gamma_1 u) - \gamma_0\mathscr{K}(\gamma_0 u), \\ \gamma_1 u &= \gamma_1\mathscr{V}(\gamma_1 u) - \gamma_1\mathscr{K}(\gamma_0 u) \end{aligned}$$

und führen den adjungierten Doppelschichtpotentialoperator $\mathscr{K}^\star$, sowie den hypersingulären Integraloperator $\mathscr{D}$ ein:

$$\begin{aligned} (\mathscr{K}^\star f)(x) &:= \gamma_1(\mathscr{V}f)(x) = \int_\Gamma \langle A\mathbf{n}, \nabla_x S(x-y)\rangle f(y)dy, \\ (\mathscr{D}f)(x) &:= \gamma_1(\mathscr{K}f)(x) = \gamma_1 \int_\Gamma \langle A\mathbf{n}, \nabla_y S(x-y)\rangle f(y)dy. \end{aligned} \tag{2.10}$$

Der Einfachschichtpotential- und der hypersinguläre Operator lassen sich stetig von Ω und $\mathbb{R}^3 \setminus \bar{\Omega}$ auf den Rand Γ fortsetzen. Für $\mathscr{K}$ und $\mathscr{K}^\star$ treten für jeden Punkt $x_0 \in \Gamma$, in dem f stetig ist, die folgenden Sprünge auf:

$$\begin{aligned}
\lim_{x\to x_0} (\mathscr{K} f)(x) &= (\mathscr{K} f)(x) + \frac{1}{2} f(x_0), \text{ für } x \in \Omega, \\
\lim_{x\to x_0} (\mathscr{K} f)(x) &= (\mathscr{K} f)(x) - \frac{1}{2} f(x_0), \text{ für } x \in \mathbb{R}^3 \setminus \bar{\Omega}, \\
\lim_{x\to x_0} (\mathscr{K}^\star f)(x) &= (\mathscr{K}^\star f)(x) + \frac{1}{2} f(x_0), \text{ für } x \in \Omega, \\
\lim_{x\to x_0} (\mathscr{K}^\star f)(x) &= (\mathscr{K}^\star f)(x) - \frac{1}{2} f(x_0), \text{ für } x \in \mathbb{R}^3 \setminus \bar{\Omega}.
\end{aligned} \tag{2.11}$$

Mit den Bezeichnungen aus (2.10) und Definition 2.1.5 erhält man unter Benutzung der Sprungrelationen (2.11) das so genannte Calderon-System:

$$\begin{pmatrix} \gamma_0 u \\ \gamma_1 u \end{pmatrix} = \begin{pmatrix} \pm\frac{1}{2}\,\mathrm{Id} - \mathscr{K} & \mathscr{V} \\ \mathscr{D} & \pm\frac{1}{2}\,\mathrm{Id} + \mathscr{K}' \end{pmatrix} \begin{pmatrix} \gamma_0 u \\ \gamma_1 u \end{pmatrix}. \tag{2.12}$$

Siehe dazu [4], [5] sowie [6].

Es sei noch eine wichtige Eigenschaft aller Fundamentallösungen $S(x - y)$ elliptischer Differentialoperatoren genannt, nämlich asymptotische Glattheit. Nähere Informationen und der Beweis sind in [7] zu finden.

Definition 2.1.6. Eine Funktion $\kappa(x, y) : \mathbb{R}^d \times \Omega \to \mathbb{C}, \kappa(\cdot, y) \in C^\infty(\mathbb{R}^d \setminus \{y\}, \mathbb{C})$, heißt asymptotisch glatt bezüglich x, wenn Konstanten c und γ existieren, so dass für alle Multiindizes α und alle $y \in \mathbb{R}^n$, alle $x \in \mathbb{R}^n \setminus \{y\}$,

$$|\partial_x^\alpha \kappa(x, y)| \leq c|\alpha|!\gamma^{|\alpha|} \frac{|\kappa(x, y)|}{\|x - y\|^{|\alpha|}}$$

gilt.

Dadurch ist die Kernfunktion $\kappa(x, y) = S(x - y)$ des Einfachschicht- und auch die des Doppelschichtpotentials $\kappa(x, y) = \langle A\mathbf{n}, \nabla_y S(x-y)\rangle$, welche nur die Normalenableitung von y enthält, asymptotisch glatt bezüglich x, was uns ihre Approximation durch Hierarchische Matrizen ermöglicht.

Die Unterblöcke der Form $\lambda\,\mathrm{Id} + \mathscr{A}$ von (2.12) werden wir im Folgenden getrennt voneinander behandeln. Durch Diskretisierung werden wir Matrizen $\lambda M + A$ aus den Operatoren erhalten, wobei die so genannte Massematrix M eine

Diagonalmatrix und somit unproblematisch ist. Unser Hauptaugenmerk liegt daher auf den Steifigkeitsmatrizen A, insbesondere für den Fall des Einfach- und des Doppelschichtpotentials $\mathscr{V}, \mathscr{K}$, da der adjungierte Doppelschichtpotentialoperator $\mathscr{K}^\star$ und der hypersinguläre Operator $\mathscr{D}$ auf diese zurückgeführt werden können.

Die betrachteten Integralgleichungen lassen sich in den seltensten Fällen analytisch lösen, weshalb wir die Randelementmethode benutzen. Ihre Grundidee ist es, den unendlichdimensionalen Funktionenraum der Test- und Ansatzfunktionen durch einen geeigneten, endlichdimensionalen Raum und die Integralgleichung durch ein lineares Gleichungssystem zu ersetzen.

2.2 Randelementmethode

Da für Randwertprobleme im Allgemeinen keine analytischen Lösungen existieren, benötigen wir ein Diskretisierungsverfahren zur numerischen Lösung. Beispielhaft stehen verschiedene zur Auswahl: die Finite Differenzenmethode, die Finite-Elemente-Methode oder die Randelementmethode, wobei unser Augenmerk auf der letzteren liegt.
Sie unterscheidet sich gravierend von den anderen, weil nicht das gesamte Gebiet Ω sondern ausschließlich dessen Rand $\partial\Omega$ diskretisiert werden muss. Dies bringt den Vorteil, dass zur Lösung des $\mathbb{R}^d$-Randwertproblems nur die Triangulierung einer $(d-1)$-dimensionalen Mannigfaltigkeit nötig ist. Das verringert den Aufwand der Netzverwaltung und die Anzahl der Freiheitsgrade. Es hat aber den Nachteil, dass entstehende Matrizen, im Gegensatz zu den anderen Verfahren, vollbesetzt und damit komplizierter zu bearbeiten sind. Außerdem muss für die Umformulierung die Fundamentallösung des Differentialoperators bekannt sein. Weitere Vorteile liegen in der Möglichkeit partielle Differentialgleichungen auf unbeschränkten Gebieten zu behandeln, was bei Außenraumproblemen nötig ist, sowie in der generisch höheren Konvergenzordnung.
Wie im vorigen Abschnitt gezeigt, wird das Randwertproblem mit Hilfe der zweiten Greenschen Formel in ein System von Integralgleichungen umgeformt. Durch Diskretisierung erhalten wir dann ein lineares Gleichungssystem, das gelöst werden muss.

Beim Galerkin-Verfahren multiplizieren wir die Integralgleichung zum Operator $\mathscr{A} : V \to W$,

$$(\mathscr{A}u)(x) = f(x) \tag{2.13}$$

mit einer Testfunktion v, integrieren über den Rand

$$\int_\Gamma (\mathscr{A}u)(x)v(x)\,\mathrm{d}s_x = \int_\Gamma f(x)v(x)\,\mathrm{d}s_x \tag{2.14}$$

und suchen eine schwache Lösung $u \in V$, so dass (2.14) $\forall v \in W$ erfüllt ist.

Satz 2.2.1. *Besitzt (2.13) eine Lösung, so stimmt diese mit der Lösung von (2.14) überein.*

Zuerst benötigen wir eine Partition $\mathcal{T} = \{\tau_1, ..., \tau_n\}$ des Randes $\Gamma = \bigcup_{i=1}^{n} \tau_i$ von Ω. Wir gehen von einer zulässigen Zerlegung in Dreiecke τ_i aus. Deren Erzeugung soll jedoch nicht Teil dieser Arbeit sein. Anschließend betrachten wir Test- und Ansatzfunktionen auf den Randelementen τ_i, zum Beispiel stückweise polynomielle oder konstante Funktionen, die die endlichdimensionalen Unterräume, den Ansatzraum $V_h \subset V$ und den Testraum $W_h \subset W$ bilden. Jetzt suchen wir eine Lösung $u_h \in V_h$ von

$$\int_\Gamma (\mathscr{A}u_h)(x)v_h(x)\,\mathrm{d}s_x = \int_\Gamma f(x)v_h(x)\,\mathrm{d}s_x,\ \forall v_h \in W_h. \tag{2.15}$$

Da $u_h \in V_h$ in einem Unterraum liegt, stimmt es im Allgemeinen nicht mit der Lösung $u \in V$ überein. Dass u_h allerdings eine quasi optimale Näherung auf dem Raum V_h darstellt, garantiert das Céa-Lemma.

Satz 2.2.2. *Sind $u \in V$ Lösung von (2.14) und $u_h \in V_h \subset V$ Lösung von (2.15), dann gilt:*

$$\|u - u_h\| \leq \frac{C}{\alpha} \inf_{v_h \in V_h} \|u - v_h\|.$$

Für die Beweise der Sätze 2.2.1 und 2.2.1 sei auf [8] oder [4] verwiesen.

Sei $B_V = \{\varphi_1, ..., \varphi_n\}$ eine Basis des endlichdimensionalen Raums V_h, so hat u_h eine Darstellung bezüglich B_V, $u_h(x) = \sum_{i=1}^{n} \alpha_i \varphi_i$. Das heißt, die gesuchte Lösung u_h ist eindeutig durch ihre Koeffizienten α_i bestimmt. Analog hat jedes v_h eine Darstellung zu einer Basis $B_W = \{\psi_1, ..., \psi_m\}$ von W_h. Damit reicht es für jeden Basisvektor $\psi_j \in B_W$ eine Gleichung aufzustellen.

$$\int_\Gamma (\mathscr{A} \sum_{i=1}^{n} \alpha_i \varphi_i)(x)\psi_j(x)\,\mathrm{d}s_x = \int_\Gamma f(x)\psi_j(x)\,\mathrm{d}s_x,\ j = 1, ..., m$$

Aufgrund der Linearität der Operatoren $\mathscr{A}$ erhalten wir ein lineares $(m \times n)$-Gleichungssystem

$$A\alpha = b, \tag{2.16}$$

mit

$$\begin{aligned} a_{ij} &= \int_\Gamma (\mathscr{A}\varphi_i)(x)\psi_j(x)\,\mathrm{d}s_x, \\ b_j &= \int_\Gamma f(x)\psi_j(x)\,\mathrm{d}s_x \end{aligned}$$

dessen Lösung uns die gesuchten Koeffizienten für die Lösung u_h von (2.15) liefert.

Sei $\mathscr{A}$ der Einfach- oder der Doppelschichtpotentialoperator wie in Definition 2.1.5, so können wir (2.13) als

$$(\mathscr{A}u)(x) = \int_\Gamma \kappa(x,y)u(y)\,\mathrm{d}s_y = f(x) \tag{2.17}$$

schreiben. Dabei ist $\kappa(x,y)$ eine Kernfunktion, die nur aus der Fundamentallösung, beziehungsweise im letzteren Fall aus deren Normalenableitung besteht.
Also wird das lineare Gleichungssystem (2.16) folgende Form annehmen:

$$\begin{aligned} &A\alpha = b, \\ &a_{ij} = \int_\Gamma \int_\Gamma \kappa(x,y)\varphi_i(y)\psi_j(x)\,\mathrm{d}s_y\,\mathrm{d}s_x, \\ &b_j = \int_\Gamma f(x)\psi_j(x)\,\mathrm{d}s_x. \end{aligned} \tag{2.18}$$

Eine weitere Möglichkeit der Diskretisierung ist das Kollokationsverfahren. Dabei gehen wir ähnlich vor wie beim Galerkinverfahren und wählen eine Basis im Ansatzraum V_h und eine Darstellung der Näherungslösung $u_h(x) = \sum_{i=1}^{n} \alpha_i\varphi_i$. Allerdings fordern wir, dass sie an ausgewählten Kollokationspunkten $x_j, j = 1,...,n$, zum Beispiel an den Mittelpunkten der Dreiecke, mit der Lösung des Ausgangsproblems (2.13) übereinstimmt, also dass für $j = 1,...,n$ gilt:

$$\begin{aligned} (\mathscr{A}u_h)(x_j) &= f(x_j), \\ (\mathscr{A}\sum_{i=1}^{n} \alpha_i\varphi_i)(x_j) &= f(x_j). \end{aligned} \tag{2.19}$$

Wir erhalten dadurch wieder ein lineares Gleichungssystem $A\alpha = b$, mit

$$\begin{aligned} a_{ij} &= (\mathscr{A}\varphi_i)(x_j), \\ b_j &= f(x_j). \end{aligned}$$

Mit der Darstellung (2.17) des Integraloperators ergibt sich beim Kollokationsverfahren:

$$\begin{aligned} A\alpha &= b, \\ a_{ij} &= \int_\Gamma \kappa(x_j, y)\varphi_i(y)\,\mathrm{d}s_y, \\ b_j &= f(x_j). \end{aligned} \tag{2.20}$$

Eine dritte, erwähnenswerte Methode zur Diskretisierung ist das Nyströmverfahren. Hierbei fordern wir nicht nur die Gleichheit an den Kollokationspunkten $x_j, j = 1, ..., n$ wie bei (2.19), sondern ersetzen auch das Integral aus dem Operator $\mathscr{A}$ durch eine Quadraturformel. Also führen wir Stützstellen $y_i, i = 1, ..., n$ und Gewichte w_i ein und erhalten schließlich das folgende lineare Gleichungssystem für unsere Operatoren der Form (2.17):

$$\begin{aligned} A\alpha &= b, \\ a_{ij} &= w_i\kappa(x_j, y_i), \\ b_j &= f(x_j). \end{aligned} \tag{2.21}$$

Die Gleichungssysteme (2.20) und (2.21), die als Ergebnis des Kollokations-, beziehungsweise des Nyströmverfahrens auftreten, können als Spezialfälle von (2.18), dem Galerkinverfahren, interpretiert werden. Man betrachtet dabei die Delta-Distribution als Basisfunktionen $\varphi_j(x) = \delta_{x_j}$ und im Nyströmfall zusätzlich $\psi_i(y) = w_i\delta_{y_i}$.

Mit den entstandenen Systemen werden wir uns in dieser Arbeit beschäftigen. Genauer gesagt mit der Frage, wie wir die Steifigkeitsmatrizen A platzsparend speichern können, so dass mit geringen Kosten auf ihnen gearbeitet werden kann, um das Gleichungssystem (2.18) zu lösen. Dafür gibt es verschiedene Ansätze, zum Beispiel Waveletverfahren, bei denen Wavelets als Basisfunktionen der endlichdimensionalen Funktionenräume gewählt werden. Wir wenden uns einem anderen Ansatz zu, den Hierarchischen Matrizen.

Kapitel 3

Hierarchische Matrizen

Wie wir gesehen haben, ergeben durch elliptische Differentialgleichungen beschriebene physikalische Phänomene nach ihrer Diskretisierung lineare Gleichungssysteme. Da in tatsächlichen Anwendungen die Zahl der Freiheitsgrade, also die Größen m, n einer Steifigkeitsmatrix $A \in \mathbb{C}^{m\times n}$, mehrere Millionen beträgt, ist es notwendig eine Lösungsmöglichkeit zu finden, die fast lineare Komplexität hat.
Eine Variante für logarithmisch lineare Komplexität liefert die Annäherung durch Hierarchische Matrizen, siehe [9] und [10]. Dabei wird die Ausgangsmatrix in geeignete, niedrigrangapproximierbare Unterblöcke zerlegt. Ihr großer Vorteil gegenüber anderen Methoden ist, dass auf dem Ergebnis der Kompression immernoch algebraische Operationen wie Addition und Multiplikation durchgeführt werden können, teilweise sogar schneller als auf der Originalmatrix.

3.1 Niedrigrangmatrizen

Definition 3.1.1. Die Menge der komplexen $(m \times n)$-Matrizen mit einem Rang von maximal k bezeichnen wir mit $\mathbb{C}_k^{m\times n}$, das heißt $\mathbb{C}_k^{m\times n} = \{A \in \mathbb{C}^{m\times n}, \operatorname{rang}(A) \leq k\}$.

Wenn eine Matrix keinen vollen Rang k hat, so können ihre Zeilen, beziehungsweise Spalten durch Linearkombinationen von je k erzeugt werden. Das ist Hintergrund des folgenden Theorems.

Satz 3.1.2. *Eine Matrix $A \in \mathbb{C}^{m\times n}$ gehört zur Menge $\mathbb{C}_k^{m\times n}$ genau dann, wenn zwei Matrizen $U \in \mathbb{C}^{m\times k}$ und $V \in \mathbb{C}^{n\times k}$ existieren, so dass*

$$A = UV^H. \tag{3.1}$$

Dabei ist V^H die hermitesche Transposition von V.

Falls k im Vergleich zu m und n klein ist, lohnt es sich die Matrix nicht einträgeweise, sondern als Produkt zu speichern.

Definition 3.1.3. Eine Niedrigrangmatrix ist eine Matrix $A \in \mathbb{C}_k^{m\times n}$ mit

$$k(m+n) < mn.$$

Die Darstellung einer Niedrigrangmatrix in äußerer Produktform ist nicht nur aus Speichergründen sinnvoll. Es lassen sich außerdem übliche Matrixoperationen direkt auf diese Form anwenden.

Beispiel 3.1.4. Bei der Matrix-Vektor-Multiplikation $Av = w, v \in \mathbb{C}^n, w \in \mathbb{C}^m$, benötigen wir $\mathcal{O}(mn)$ Operationen. Liegt $A = UV^H$ in äußerer Produktform vor, berechnen wir erst $V^H v$ und multiplizieren das Ergebnis anschließend mit U, was uns zusammen $\mathcal{O}(k(m+n))$ Operationen kostet und damit bei Niedrigrangmatrizen schneller ist.

Diese guten Eigenschaften gelten zwar vorerst nur für Niedrigrangmatrizen, allerdings lassen sich einige Matrizen mit vollem Rang durch Niedrigrangmatrizen approximieren. Damit meinen wir, dass der nötige Rang k nur logarithmisch von einer vorgegebenen Genauigkeit ε abhängt. Diese Bedingung erfüllen geeignete Unterblöcke der Steifigkeitsmatrizen elliptischer Differentialoperatoren, wie in (2.5).
Die beste k-Rang-Näherung erhält man aus der Singulärwertzerlegung einer Matrix.

Satz 3.1.5. *Sei $A = U\Sigma V^H$ die Singulärwertzerlegung einer Matrix $A \in \mathbb{C}^{m\times n}, m \geq n$, dann gilt für alle $k \geq n$:*

$$\min_{M\in\mathbb{C}_k^{m\times n}} \|A - M\| = \|A - A_k\| = \|A - \Sigma_k\|,$$

wobei $A_k := U\Sigma_k V^H \in \mathbb{C}^{m\times n}$ und $\Sigma_k := \operatorname{diag}(\sigma_1, ..., \sigma_k, 0, ..., 0) \in \mathbb{R}^{k\times k}$ sei.

Um die gewöhnliche äußere Produktform zu erhalten, muss Σ_k noch auf eine der beiden anderen Matrizen multipliziert werden. Da die Berechnung der Singulärwertzerlegung jedoch zu teuer ist, werden wir uns anderer Techniken bedienen.

3.2 Partitionierung

Für gewöhnlich lassen sich Matrizen nicht gänzlich niedrigrangapproximieren, oft treten jedoch Matrizen mit approximierbaren Unterblöcken auf. Wir benötigen also Hilfsmittel zur Beschreibung einer geeigneten Zerlegung.

Definition 3.2.1. Seien $I, J \subset \mathbb{N}$. Eine Teilmenge $P \in \mathcal{P}(I \times J)$ des kartesischen Produktes von I und J heißt Partition, wenn

$$I \times J = \bigcup_{b \in P} b$$

und wenn aus $b_1 \cap b_2 \neq \emptyset$ stets $b_1 = b_2$ für alle $b_1, b_2 \in P$ folgt.

Die Einschränkung der Matrix A auf eine dieser Indexteilmengen $b = t \times s, t \subset I, s \subset J$ bezeichnen wir mit A_{ts} oder A_b und nennen sie Unterblock oder nur Block von A.

Unter all den potenziellen Zerlegungen suchen wir nun eine, die möglichst große Blöcke mit möglichst kleinem Rang erzeugt, denn das würde für uns geringen Speicher- sowie Berechnungsaufwand in der äußeren Produktform bedeuten. Leider ist die Suche nach der optimalen Partition in Anbetracht der einander diametral gegenüberliegenden Kriterien sehr komplex. Wir geben uns daher mit einer Partition zufrieden, die uns Approximationen logarithmisch-linearer Komplexität ermöglicht und in logarithmisch-linearer Zeit erzeugbar ist. Eine solche erhalten wir durch das Einhalten so genannter Zulässigkeitsbedingungen. Je nach Problemstellung gibt es verschiedene. Sie haben jedoch alle, um eine ausreichend gute Partitionierung zu erzeugen, Folgendes gemeinsam:

(i) ist b zulässig, so fallen die Singulärwerte von A_b exponentiell,

(ii) die Zulässigkeitsbedingung kann auf jedem Block $t \times s \in \mathcal{P}(I \times J)$ mit $\mathcal{O}(|t| + |s|)$ Operationen geprüft werden,

(iii) ist b zulässig, so auch jede seiner Teilmengen $b' \subset b$.

Wir werden uns in dieser Arbeit besonders mit Zulässigkeitsbedingungen beschäftigen, die auf die Struktur von Steifigkeitsmatrizen elliptischer Differentialoperatoren angepasst sind, zum Beispiel mit der Fernfeldbedingung. Wie in Kapitel 2 beschrieben, enthält jeder Eintrag a_{ij} einer solchen Matrix $A \in \mathbb{C}^{m \times n}$ die Auswertung einer Kernfunktion $\kappa(x_i, y_j)$ an den beiden Stellen $x_i, y_j \in \Omega \subset \mathbb{R}^d$ im Raum. Eine Partition wie in Definition 3.2.1 von Indizes entspricht hier also ebenso einer Zerlegung des Raumes $\mathbb{R}^d$.

Definition 3.2.2. Seien $X_i = \text{supp}(\varphi_i), Y_j = \text{supp}(\psi_j), i = 1, ..., n, j = 1, ..., m$, die Träger der Test- und Ansatzfunktionen bei der Randelementmethode, so nennen wir die zu jedem Block $b = t \times s$ gehörigen Teile des Raumes

$$\begin{aligned} X_t &= \bigcup_{i \in t} X_i \subset \mathbb{R}^d, \\ Y_s &= \bigcup_{j \in s} Y_j \subset \mathbb{R}^d \end{aligned}$$

Cluster.

Beispiel 3.2.3. Ein Block $b = t \times s$ der Steifigkeitsmatrix eines elliptischen Differentialoperators, wie zum Beispiel des Laplaceoperators, kann als zulässig gelten, wenn das Cluster Y_s zum Cluster X_t die Fernfeldbedingung

$$\text{diam}(X_t) \leq \eta \, \text{dist}(X_t, Y_s) \tag{3.2}$$

erfüllt. Dabei ist $\text{diam}(X_t) = \max_{x_1, x_2 \in X_t} \|x_1 - x_2\|$ der Durchmesser des Clusters X_t, $\text{dist}(X_t, Y_s) = \min_{x \in X_t, y \in Y_s} \|x - y\|$ der Abstand der Cluster und $\eta > 0$ ein Parameter. Wir sagen dann Y_s liegt im Fernfeld von X_t. Es gibt auch symmetrische Varianten dazu, bei denen gefordert wird, dass sich wenigstens eines der beiden oder aber beide Cluster im Fernfeld des jeweils anderen befinden.
Das bedeutet, dass die beiden Cluster X_t und Y_s im Vergleich zu ihrer Größe weit voneinander entfernt liegen sollen, siehe Abbildung 3.1.

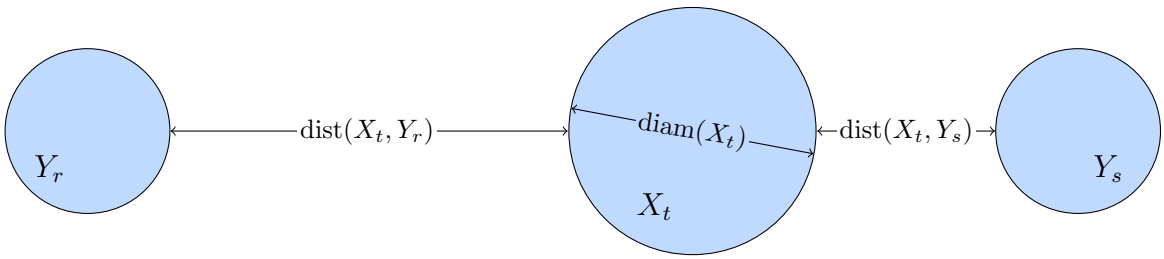

Abbildung 3.1: Das Cluster Y_r liegt im Fernfeld von X_t mit $\eta = 1$, Y_s dagegen nicht

Diese Bedingung ist speziell auf die von uns betrachteten Kernfunktionen $\kappa(x, y)$, die eine Singularität ausschließlich bei $x = y$ aufweisen, angepasst. Die Einträge der Unterblöcke $t \times s$ mit großem Abstand zur Hauptdiagonalen unterscheiden sich nur geringfügig. Man kann zeigen, dass die Eigenwerte dieser Matrixblöcke exponentiell fallen. Also erfüllt (3.2) die erste Eigenschaft einer Zulässigkeitsbedingung, jedoch nicht die zweite, da die Berechnung von $\text{dist}(X_t, Y_s)$ $\mathcal{O}(|t||s|)$ Operationen benötigt. Deshalb

benutzen wir statt (3.2) die etwas strengere, aber in $\mathcal{O}(|t| + |s|)$ Schritten prüfbare Zulässigkeitsbedingung:

$$\frac{1+\eta}{\eta} r_t \leq \operatorname{dist}(m_t, Y_s). \tag{3.3}$$

Hier sind r_t der Radius und m_t der Mittelpunkt des Clusters X_t.

Um Cluster mit möglichst kleinem Durchmesser zu erzeugen und damit die Zulässigkeitsbedingung (3.3) zu erfüllen, teilen wir den Raum Ω sukzessive, zum Beispiel mittels Hauptkomponentenanalyse, siehe [11], bis die Einzelteile eine gegebene Größe n_{min} unterschreiten und speichern die Partition in einem so genannten Clusterbaum, vergleiche Definition 3.2.4 sowie Abbildung 3.2.

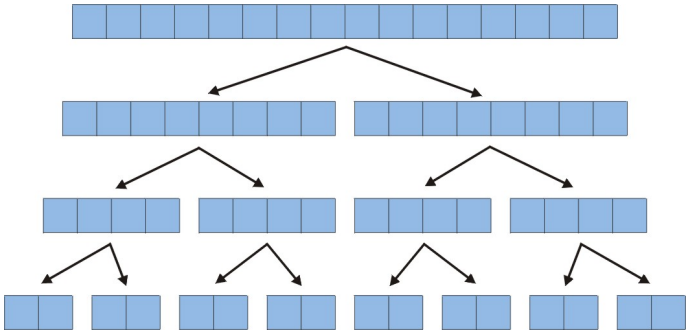

Abbildung 3.2: Beispiel für einen Clusterbaum

In einem Baum $B = (V, E)$ mit Knoten V und Kanten E bezeichnen wir den graphentheoretischen Abstand eines Knotens $t \in V$ zur Wurzel als Tiefe von t und die Menge $B^{(l)} = \{t \in V, t \text{ hat die Tiefe } l\}$ als l-te Ebene. Weiter seien $S(t) := \{t' \in B^{(l+1)} : (t, t') \in E\}$ die Menge aller Söhne eines Knotens $t \in V$, $\deg(t) = |S(t)|$ sein Grad und $\mathcal{L}(B) := \{t \in V : \deg(t) = 0\}$ die Menge der Blätter.

Definition 3.2.4. Ein Baum $T_I = (V, E)$ heißt Clusterbaum zur Indexmenge $I \subset \mathbb{N}$, wenn er folgende Bedingungen erfüllt:

(i) I ist die Wurzel von T_I,

(ii) $\forall t \in V \setminus \mathcal{L}(T_I) : t \neq \emptyset,\ t = \bigcup_{t' \in S(t)} t',\ \forall t_1, t_2 \in S(t) : t_1 \cap t_2 = \emptyset,$

(iii) $\forall t \in V \setminus \mathcal{L}(T_I) : \deg(t) \geq 2.$

Ein Clusterbaum T_I kann die Zerlegung des Raumes, also der zugrunde liegenden Geometrie, speichern. Um aber die Zerlegung der Matrix in ihre Unterblöcke zu

verwalten, benötigen wir eine Struktur auf $I \times J$. Das kartesische Produkt von zwei Clusterbäumen eignet sich dafür leider nicht, weil es zu quadratischem Speicheraufwand führen würde und damit unseren Forderungen nicht nach käme. Wir benutzen daher hierarchische Strukturen, die mit logarithmisch-linearem Speicherbedarf auskommen, nämlich Blockclusterbäume. Streng genommen ist ein Blockclusterbaum die Erweiterung eines Clusterbaums auf eine Indexmenge $I \times J \subset \mathbb{N} \times \mathbb{N}$, wir beschränken uns aber auf zwei spezielle Arten, die jeweils durch die ihnen zugrunde liegenden Zerlegungsalgorithmen bestimmt sind.

So kann ein Blockclusterbaum $T_{I\times J} = (V, E)$ aus zwei Clusterbäumen T_I, T_J aufgebaut werden, indem man fordert, dass $I \times J$ seine Wurzel ist und dass die Söhne $S(t \times s)$ jedes Knotens $t \times s \in V$ eindeutig durch die Vorschrift

$$S_{I\times J}(t \times s) = \begin{cases} \emptyset, & \text{falls } t \times s \text{ zulässig} \vee S(t) = \emptyset \vee S(s) = \emptyset, \\ S(t) \times S(s), & \text{sonst,} \end{cases} \tag{3.4}$$

festgelegt sind.
Wir zerlegen die Matrix also rekursiv und testen für jeden Unterblock, ob er die Zulässigkeitsbedingung erfüllt. Ist dies der Fall, speichern wir ihn approximiert in äußerer Produktform, anderenfalls teilen wir ihn erneut und untersuchen seine Söhne, bis die Größen der Indexmengen n_{min} unterschreiten.
Die Partition P besteht dann aus den Blättern des Blockclusterbaums.

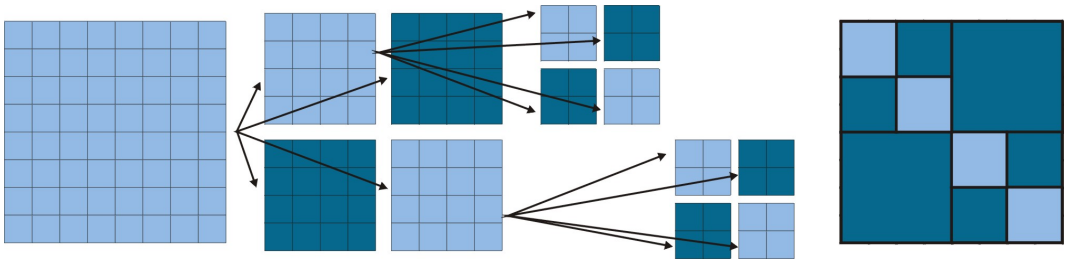

Abbildung 3.3: Beispiel für einen Blockclusterbaum aus zwei Teilclusterbäumen wie in Abb. 3.2 mit resultierender Matrixpartitionierung rechts. Die dunkelblauen Blöcke sollen dabei zulässig sein, die hellblauen dagegen nicht.

Eine andere Möglichkeit einen Blockclusterbaum $T_{I\times J} = (V, E)$ zu erzeugen besteht darin, von nur einem zugrunde liegenden Clusterbaum T_I auszugehen, die Wurzel $I \times J$ zu setzen und seiner Hierarchie folgend, zu jedem Knoten $t \in T_I$ die maximale, zulässige

Indexmenge zu wählen. Der nicht zulässige Rest wird mit den Söhnen $S(t)$ zu neuen Blättern des Blockclusterbaums zusammengefügt, bis die Unterblöcke zu klein werden. Seien also $z(t) \subset J$,

$$z(t) = \{j \in J : t \times s \text{ zulässig nach Fernfeldbedingung (3.3)}\}$$

die Menge aller Indizes, die die jeweilige Zulässigkeitsbedingung zusammen mit t erfüllen und $t_1, ..., t_n \in S(t)$ die Kindknoten von t im Clusterbaum T_I, dann bestimmt die folgende Vorschrift die Söhne zu einem Knoten $t \times s$:

$$S_{I\times J}(t \times s) = \begin{cases} \emptyset, & \text{falls } t \times s \text{ zulässig} \vee |s| < n_{min}, \\ \{t \times (z(t) \cap s)), t \times (s \setminus z(t))\}, & \text{falls } S(t) = \emptyset \vee |s \setminus z(t)| < n_{min}, \\ \{t \times (z(t) \cap s), t_1 \times (s \setminus z(t)), ..., t_n \times (s \setminus z(t))\}, & \text{sonst.} \end{cases} \quad (3.5)$$

Gegenüber der ersten Variante haben wir den Vorteil, eine Zerlegung mit minimaler Anzahl an Blöcken und maximal ausgenutzter Zulässigkeit zu erhalten, vergleiche Abbildungen 3.4 und 3.5, die mit gleicher Zulässigkeitsbedingung aber verschiedener Konstruktionsvariante erzeugt wurden. Da zulässige Blöcke in der äußeren Produktform platzsparender gespeichert werden können, benötigt die Partitionierung in Abbildung 3.5 weniger Speicher als die in Abbildung 3.4. Nachteilig ist allerdings, dass diese im Allgemeinen nicht mehr zusammenhängend und in ihrer Behandlung daher komplexer sind. Außerdem können zwar Matrix-Vektor-Multiplikation und Addition mit Matrizen noch durchgeführt werden, aber die zweite Konstruktionsvariante (3.5) erlaubt die Anwendung einiger höherer Matrixoperationen wie der Matrizenmultiplikation nicht mehr.
Es ein noch angemerkt, dass der Blockclusterbaum in Abbildung 3.3 in dem Sinn optimal ist, als dass die Anwendung der Partitionierung (3.5) mit gleichem η zur exakt selben Zerlegung geführt hätte.

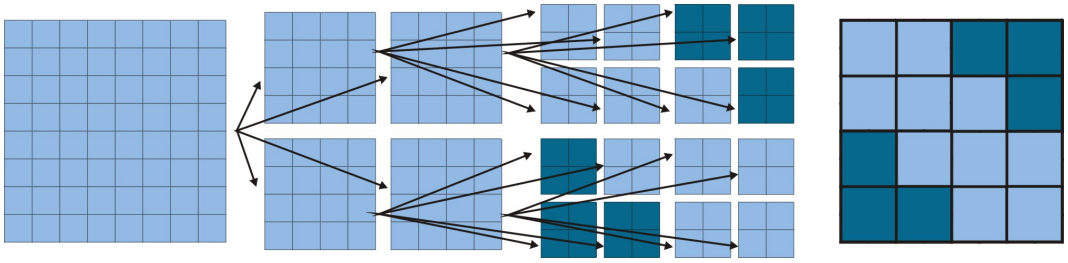

Abbildung 3.4: Beispiel für einen Blockclusterbaum aus zwei Clusterbäumen mit strengerer Zulässigkeitsbedingung durch kleineres η

Es ist in Abbildung 3.5 besonders zu beachten, dass die dunkelblauen, zulässigen Teile der Matrixpartitionierung, die nur zwei Einträge beinhalten, keine eigenständigen

Blöcke sind. Sie gehören zu dem Teil mit vier Einträgen in der gleichen Zeile und bilden damit ein anschauliches Beispiel für nicht zusammenhängende Blöcke. Hier müssen alle Indizes gespeichert werden, bei zusammenhängenden nur Anfang und Ende. Deshalb werden wir, wann immer es die Komplexität zulässt, letztere Variante benutzen. Das wird bei der Fernfeldbedingung der Fall sein.

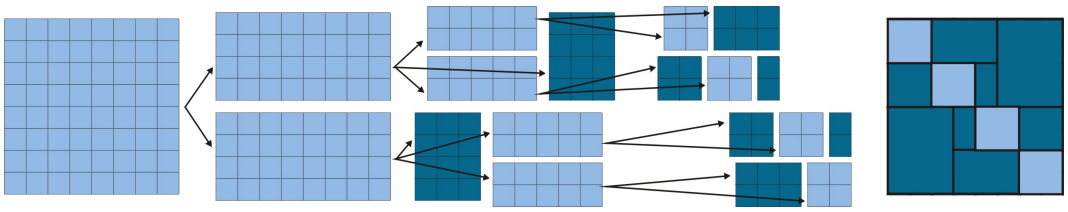

Abbildung 3.5: Beispiel für einen Blockclusterbaum, aus einem Clusterbaum konstruiert

In [7] sind Abschätzungen des Speicher- und Berechnungsaufwands für die Varianten (3.4) und (3.5) zum Aufstellen eines Blockclusterbaums bewiesen. Die Ergebnisse werden in Tabelle 3.1 kurz zusammengefasst. Dabei sind $\mathcal{L}(T_I)$ wie oben die Blätter eines Clusterbaums, $|\mathcal{L}(T_I)| \sim |I|$ deren Anzahl und c_{sp} die so genannte Sparsity-Konstante. Sie beschreibt, wie viele Knoten es zu einer gegebenen Indexmenge maximal gibt. Ihre Definition unterscheidet sich je nach Konstruktionsvariante.

Definition 3.2.5. Für die Partitionierung (3.5) aus einem Clusterbaum T_I ist die Sparsitiy-Konstante $c_{sp}(T_I)$ definiert durch:

$$c_{sp}(T_I) = \max_{t \in T_I} |\{s \in J : t \times s \in T_{I \times J}\}|.$$

Für den Blockclusterbaum $T_{I \times J}$ aus zwei Clusterbäumen (3.4) ist die Sparsitiy-Konstante $c_{sp}(T_{I \times J})$ als Maximum der zeilenweisen

$$c^r_{sp}(T_{I \times J}) = \max_{t \in T_I} |\{s \in J : t \times s \in T_{I \times J}\}|$$

und der spaltenweisen Sparsity-Konstante

$$c^c_{sp}(T_{I \times J}) = \max_{s \in T_J} |\{t \in I : t \times s \in T_{I \times J}\}|$$

definiert, also

$$c_{sp}(T_{I \times J}) = \max\{c^r_{sp}(T_{I \times J}), c^c_{sp}(T_{I \times J})\}.$$

	zwei Clusterbäume	ein Clusterbaum
Operationen zum Aufstellen	$c_{sp}(T_{I\times J})(\lvert I\rvert \log \lvert I\rvert + \lvert J\rvert \log \lvert J\rvert)$	$\eta^{-d-1}\lvert I\rvert \log \lvert I\rvert$
Anzahl der Blöcke	$2c_{sp}(T_{I\times J}) \min\{\lvert\mathscr{L}(T_I)\rvert, \lvert\mathscr{L}(T_J)\rvert\}$	$c_{sp}(T_I)\lvert\mathscr{L}(T_I)\rvert$

Tabelle 3.1: Komplexitätsklassen der verschiedenen Konstruktionsvarianten von Blockclusterbäumen im Vergleich

Bei Konstruktionsvariante (3.4) ist $c_{sp}(T_{I\times J})$ eine von $|I|$ und $|J|$ unabhängige und von η abhängige Konstante, siehe [7]. Für die Konstruktion aus nur einem Clusterbaum lässt sich sogar sofort ablesen, dass $c_{sp}(T_I) = 2$ gilt, da zu jedem $t \in I$ maximal der größtmögliche zulässige Teil und der kleine nicht zulässige Rest als potentielle Kandidaten zur Blockbildung in Frage kommen.

Definition 3.2.6. Eine Partition P heißt zulässig, wenn jeder Block $t \times s \in P$ zulässig oder klein, das heißt $|t| < n_{min} \vee |s| < n_{min}$, ist.

Definition 3.2.7. Die Menge der Hierarchischen Matrizen auf einem Blockclusterbaum $T_{I\times J}$ mit zulässiger Partition P und blockweisem Rang k wird definiert als

$$\mathscr{H}(T_{I\times J}) = \{A \in \mathbb{C}^{|I|\times|J|}, \forall b \in P : \operatorname{rang}(A_b) \leq k\}.$$

Ein Element dieser Menge nennen wir kurz $\mathscr{H}$-Matrix. Die zulässigen Blöcke speichern wir in äußerer Produktform, die kleinen, unzulässigen einträgeweise.

3.3 Multiplikation mit Vektoren

Die Matrix-Vektor-Multiplikation spielt für uns eine besonders wichtige Rolle. Die linearen Gleichungssysteme, die durch die Diskretisierung aus den Differentialoperatoren entstehen, werden selten direkt gelöst. Direkte Verfahren sind in Anbetracht der großen Zahl von Freiheitsgraden im Allgemeinen zu langsam. Viele iterative Methoden können lineare Gleichungssysteme durch wiederholte Matrix-Vektor-Multiplikation lösen. Es ist also wichtig, diese algebraische Operation zur Verfügung zu stellen und effizient zu gestalten.

Die Multiplikation einer Hierarchischen Matrix $A \in \mathscr{H}(T_{I\times J})$ mit einem Vektor $x \in \mathbb{C}^{|J|}$ erfolgt durch Multiplikation der einzelnen Blöcke mit den jeweiligen Teilen des

Vektors und anschließender Summierung über die Ergebnisse. Symbolisch kann man

$$Ax = \sum_{t \times s \in P} A_{ts} x_s$$

schreiben und meint

$$Ax = \sum_{t \times s \in P} y_t$$

mit $x_s \in \mathbb{C}^{|J|}$,

$$(x_s)_j := \begin{cases} x_j & \text{für } j \in s, \\ 0 & \text{sonst,} \end{cases}$$

und $y_t \in \mathbb{C}^{|I|}$,

$$(y_t)_i := \begin{cases} (A_{ts} x_s)_i & \text{für } i \in t, \\ 0 & \text{sonst.} \end{cases}$$

Wie beschrieben können wir von einer Menge zulässiger oder kleiner Blöcke A_{ts} ausgehen. Für die zulässigen sind die Zahl der zu speichernden Einträge und die der Operationen bei der Matrix-Vektor-Multiplikation $A_{ts} x_s = U_{ts} V_{ts}^H x_s$ in äußerer Produktform durch

$$\mathcal{O}(k(|t| + |s|))$$

beschränkt, siehe Beispiel 3.1.4 sowie [7]. Für die kleinen benötigt man, da sie einträgeweise gespeichert werden,

$$\mathcal{O}(|t||s|)$$

Einträge sowie Operationen. Einen Block hatten wir klein genannt, wenn wenigstens einer der Indexmengen eine Konstante n_{min} unterschreitet, also folgt

$$|t||s| \leq \max\{|t|, |s|\} \cdot n_{min} \leq n_{min}(|t| + |s|).$$

Sind k und n_{min} gegebene Konstanten und $c = \max\{k, n_{min}\}$, können die Kosten von blockweise anwendbaren Operationen auf hierarchischen Matrizen mit $c(|t| + |s|)$ nötigen Berechnungsschritten pro Block abgeschätzt werden. In [7] findet man für die Blockclusterbäume aus zwei zu Grunde liegenden, balancierten Clusterbäumen T_I, T_J, vergleiche Definition 3.3.1,

$$\sum_{t \times s \in T_{I \times J}} c(|t| + |s|) \sim |I| \log |I| + |J| \log |J|.$$

Wir gehen davon aus, dass I und J von der gleichen Größenordnung sind, also $|J| = O(|I|)$, woraus

$$\sum_{t\times s\in T_{I\times J}} c(|t| + |s|) \sim |I| \log |I|$$

folgt. Der Aufwand für die Multiplikation einer Hierarchischen Matrix aus zwei Clusterbäumen mit einem Vektor beträgt also $\mathcal{O}(n \log n)$. Die Matrix-Vektor-Multiplikation auf einer unkomprimierten Matrix liegt in der Komplexitätsklasse $\mathcal{O}(n^2)$. Das heißt diese Operation ist auf eben betrachteten $\mathscr{H}$-Matrizen nicht nur durchführbar sondern auch noch schneller als auf gewöhnlichen. Wir werden zeigen, dass diese hervorragende Eigenschaft nicht von der Art der Konstruktion abhängt und auch die Partitionierungsvariante (3.5) aus einem Clusterbaum eine Matrix-Vektor-Multiplikation in $\mathcal{O}(n \log n)$ Schritten erlaubt.

Definition 3.3.1. Ein Baum T_I heißt balanciert, wenn für alle Knoten $t \in T_I$ eine von $|I|$ unabhängige Konstante $R > 0$ existiert, so dass

$$R \leq \frac{|t_1|}{|t_2|}$$

für alle Söhne $t_1, t_2 \in S(t)$ von t gilt.

Satz 3.3.2. *Sei $A \in \mathscr{H}(T_{I\times J})$ eine hierarchische Matrix auf einem Blockclusterbaum $T_{I\times J}$, der aus einem balancierten Clusterbaum T_I mit konstanter, von $|I|$ unabhängiger, maximaler Anzahl $|S| = \max_{t\in T_I} |S(t)|$ an Söhnen pro Knoten durch (3.5) mit der Fernfeldbedingung (3.3) konstruiert wurde. Weiter gelte für alle Cluster $X_i, i \in I$:*

$$\max_{i\in I} \operatorname{diam}(X_i) \leq c_s \min_{i\in I} \operatorname{diam}(X_i).$$

Dann ist die Anzahl der nötigen Operationen bei der Matrix-Vektor-Multiplikation mit einem Vektor $x \in \mathbb{C}^{|J|}$ von der Ordnung

$$\eta^{1-d}|I| \log |I|.$$

Beweis: Wir betrachten einen Knoten $t \in T_I$ mit Söhnen $t_1, ..., t_n \in S(t)$. Sei $|t_{max}| = \max_{t_i\in S(t)} |t_i|$, die maximale Anzahl an Indizes in einem der Kindknoten, dann gilt

$$|t| = |t_1| + ... + |t_n| \leq |S(t)||t_{max}|.$$

Da der Clusterbaum balanciert ist, gilt zusätzlich $|t_i| > R|t_{max}|\ \forall t_i \in S(t)$ und damit folgt $\forall t_i \in S(t)$:

$$|t| \leq \frac{|S(t)|}{R}|t_i| \leq \frac{|S|}{R}|t_i|. \tag{3.6}$$

Wenn man einen Knoten $t \times s$ aus dem Blockclusterbaum und dessen aus (3.5) konstruierten Söhne betrachtet, ist offensichtlich, dass für den zulässigen Teil die Hierarchie sofort endet. Der Vater eines jeden, von der Wurzel verschiedenen, Knotens ist also nicht zulässig. Die Indizes $s \setminus z(t)$, die mit den Söhnen $s(t)$ weiter gepaart werden, sind nur die, die nicht im Fernfeld von t liegen. Für einen nicht vollständig zulässigen Knoten $t \times s \neq I \times J$, der einen Vater $\hat{t} \times \hat{s}$ hat, gilt also

$$|s| = |\hat{s} \setminus z(\hat{t})| \leq |J \setminus z(\hat{t})|.$$

Die Menge $J \setminus z(\hat{t})$ nennt man das Nahfeld und aus [7] Lemma 1.33 ist bekannt, dass seine Größe durch

$$|J \setminus z(\hat{t})| \leq c_1(1 + \frac{1}{\eta})^{d-1}|\hat{t}|$$

beschränkt ist.
Damit folgt für alle nicht zulässigen Knoten $t \times s$.

$$|s| \leq |J \setminus z(\hat{t})| \leq c_1(1 + \frac{1}{\eta})^{d-1}|\hat{t}| \leq c_1(1 + \frac{1}{\eta})^{d-1}\frac{|S|}{R}|t|.$$

Schließlich ergibt sich daraus mit (3.6) für alle zulässigen Knoten $t \times s \in B^{(l)}, l \geq 2$ mit einer Tiefe von wenigstens 2 als Kinder nicht zulässiger Knoten $\hat{t} \times \hat{s}$

$$|s| \leq |\hat{s}| \leq c_1(1 + \frac{1}{\eta})^{d-1}R|S||\hat{t}| \leq c_1(1 + \frac{1}{\eta})^{d-1}\frac{|S|^2}{R^2}|t|.$$

Da R, S, c Konstanten sind, folgt mit $c_2 = c\frac{|S|^2}{R^2}$ für aller Knoten $t \times s \in B^{(l)}, l \geq 2$

$$|s| \leq c_2(1 + \frac{1}{\eta})^{d-1}|t|.$$

Für die Wurzel und ihre direkten Kinder benutzen wir $|s| \leq |J|$ und erhalten maximal $c(|I| + |J|) + c(|I| + |J|)|S|$, also $O(I)$ Operationen. Zusammen mit Lemma 1.21 aus [7], welches besagt

$$\sum_{t \in T_I} |t| \leq L(T_I)|I|,$$

folgt für alle übrigen Knoten bei der Matrix-Vektor-Multiplikation, die maximal $c = \min\{k, n_{min}\}$ Operationen pro Knoten des Blockclusterbaums benötigt:

$$\begin{aligned}
\sum_{t\times s\in T_{I\times J}\setminus(I\times J)\setminus S(I\times J)} c(|t|+|s|) &\leq \sum_{t\times s\in T_{I\times J}\setminus(I\times J)\setminus S(I\times J)} c(|t|+c_2(1+\frac{1}{\eta})^{d-1}|t|) \\
&= \sum_{t\in T_I\setminus(I)\setminus S(I)} \sum_{s\subset J, t\times s\in T_{I\times J}} c(|t|+c_2(1+\frac{1}{\eta})^{d-1}|t|) \\
&\leq \sum_{t\in T_I\setminus(I)\setminus S(I)} c_{sp}c(|t|+c_2(1+\frac{1}{\eta})^{d-1}|t|) \\
&\leq \sum_{t\in T_I\setminus(I)\setminus S(I)} c_{sp}cc_2(1+(1+\frac{1}{\eta})^{d-1})|t| \\
&\leq c_{sp}cc_2(1+(1+\frac{1}{\eta})^{d-1})L(T_I)|I|.
\end{aligned}$$

Mit $0 < \eta < 1$ und dem binomischen Satz folgt

$$\begin{aligned}
(1+\frac{1}{\eta})^{d-1} &= \sum_{k=0}^{d-1} \binom{d-1}{k} 1^k (\frac{1}{\eta})^{d-1-k} \\
&= O(\eta^{1-d})
\end{aligned}$$

und weil T_I balanciert ist, gilt $L(T_I) = O(\log|I|)$.
Die verschiedenen Teile zusammen ergeben:

$$\begin{aligned}
\sum_{t\times s\in T_{I\times J}} c(|t|+|s|) &\leq c_{sp}cc_2(1+(1+\frac{1}{\eta})^{d-1})L(T_I)|I| + c(|I|+|J|) + c(|I|+|J|)|S| \\
&= O(\eta^{1-d}|I|\log|I|).
\end{aligned}$$

□

Bemerkung 3.3.3. Wir haben im Beweis nicht benutzt, dass die Matrix-Vektor-Multiplikation nur auf zulässigen Blöcken, also auf den Blättern des Blockclusterbaums, arbeitet. Die Abschätzung gilt also auch für beliebig andere Algorithmen, die nur $c(|t|+|s|)$ Operationen auf jedem Knoten $t \times s \in T_{I\times J}$ benötigen.

3.4 Berechnung der Einträge

Im letzten Abschnitt haben wir die Matrix in Blöcke zerlegt, die niedrigrangapproximierbar oder klein sind. Wir wollen Blöcke $A_{ts} \in \mathbb{C}^{m\times n}$ der

Steifigkeitsmatrix eines elliptischen Differentialoperators approximieren und erinnern uns, dass diese nach der Randelementmethode folgende Form haben:

$$a_{ij} = \int_\Gamma \int_\Gamma \kappa(x,y)\varphi_i(x)\psi_j(y)\,\mathrm{d}s_x\,\mathrm{d}s_y,\ i \in I, j \in J.$$

Die kleinen Blöcke haben auch genau diese Einträge. Jetzt fehlen uns für die anderen noch die jeweiligen Approximanten, also die tatsächlichen Matrixeinträge von U und V. Die optimale Näherung erhält man aus der Singulärwertzerlegung, die leider viel zu aufwendig ist. Für eine schnelle Berechnung gibt es zwei Möglichkeiten. Zum einen wäre hier die Annäherung der Kernfunktion des Integraloperators mittels einer degenerierten Kernfunktion, zum Beispiel durch Interpolation und zum anderen die Berechnung aus den originalen Matrixeinträgen mit Hilfe des ACA-Verfahrens zu nennen.

3.4.1 Interpolation

Definition 3.4.1. Seien $D_1, D_2 \subset \mathbb{R}^d$ zwei Gebiete. Eine Kernfunktion $\kappa : D_1 \times D_2 \to \mathbb{C}$ heißt degeneriert, wenn ein $k \in \mathbb{N}$ und Funktionen $u_l : D_1 \to \mathbb{C}$ und $v_l : D_2 \to \mathbb{C}$, $l = 1, ..., k$ existieren, so dass

$$\kappa(x,y) = \sum_{l=1}^{k} u_l(x)v_l(y).$$

In [7] wird gezeigt, dass man durch Anwenden desselben Diskretisierungsverfahrens bei der Randelementmethode auf u_l und v_l statt auf κ eine Niedrigrangmatrix UV^H vom Rang k erhält. Also reicht es nach einem degenerierten Approximanten für die Kernfunktion zu suchen.
Eine Möglichkeit dafür ist Interpolation nach einer Variablen. Wir werden die Tschebyscheffpolynominterpolation, siehe (1.4), auf y betrachten. Die Degeneriertheit ist direkt aus

$$\mathcal{I}_p^y \kappa(x,y) = \sum_{\boldsymbol{j} \in \mathbb{N}^d, j_\nu < p} c_{\boldsymbol{j}}(x) \prod_{\nu=1}^{d} T_{j_\nu}(\epsilon_\nu)$$

ablesbar, da $c_{\boldsymbol{j}}$ nur von den Auswertungen der Funktion an der Stelle x und den Tensortschebyscheffknoten, also nicht von y abhängt. Außerdem ist $\epsilon_\nu = 2\frac{y_\nu - a_\nu^y}{b_\nu^y - a_\nu^y} - 1$ von x unabhängig und man kann $k = p^d$ ablesen.
Bei der Interpolation mit Tschebyscheffknoten gilt folgende Fehlerabschätzung.

Satz 3.4.2. *Seien* $D_1 \subset \mathbb{R}^d, D_2 = \prod_{\nu=1}^{d} [a_\nu, b_\nu] \subset \mathbb{R}^d$ *Gebiete, die die Fernfeldbedingung* $\eta \operatorname{dist}(D_1, D_2) \geq \max_{\nu=1,\ldots,d} b_\nu - a_\nu$ *für ein* $\eta > 0$ *erfüllen und* $\kappa : D_1 \times D_2 \to \mathbb{C}$ *eine bezüglich* y *asymptotisch glatte Funktion, so dass für die Konstanten* $c\gamma\eta < 1$ *gilt, dann folgt für alle* $x \in D_1, y \in D_2$

$$|\kappa(x,y) - \mathcal{I}_p^y \kappa(x,y)| \leq \tilde{c}\left(1 + \frac{2}{\pi}\log p\right)^d \left(\frac{\gamma\eta}{4}\right)^p |\kappa(x,y)|.$$

Für den Beweis und detailliertere Informationen, siehe [7], [1].

Um mit Interpolation arbeiten zu können, muss man allerdings die Kernfunktion kennen und auch oft auswerten. Außerdem kann man ein Gebiet nicht immer als Tensorprodukt von Intervallen schreiben. In diesem Fall leidet die Qualität der Approximation gravierend.

3.4.2 Das ACA-Verfahren

Besonders wenn Matrizen schon vorliegen oder eine Kernfunktion nicht bekannt ist, bietet sich das Verfahren der Adaptive Cross Approximation (ACA) an, bei dem nur die originalen Matrixeinträge benötigt werden.
Hier werden so lange geeignete Zeilen $i \in \{1, ..., m\}$ und Spalten $j \in \{1, ..., n\}$ aus der Matrix ausgewählt und deren gewichtetes äußeres Produkt von ihr abgezogen, bis die Norm der verbleibenden Matrix R_k klein genug ist.

$$\begin{aligned} R_0 &:= A, \\ R_{k+1} &:= R_k - \frac{1}{(R_k)_{i_k j_k}} (R_k)_{1:m,j_k} (R_k)_{i_k,1:n}. \end{aligned}$$

Die Wahl der Pivotzeilen wird in [7] beschrieben. Innerhalb einer Zeile bestimmt das größte Element den zugehörigen Spaltenindex.
Diese Beschreibung des Algorithmus soll nur die grundlegende Wirkungsweise verdeutlichen. Die Berechnung der Matrizen A oder R_k ist nicht wünschenswert, da sie zu quadratischer Komplexität führen würde, und auch nicht notwendig. Der Algorithmus benutzt tatsächlich nur wenige der originalen Einträge, um die Niedrigrangapproximation, also $U \in \mathbb{R}^{m \times k}$ und $V \in \mathbb{R}^{n \times k}$ mit $A \approx UV^H$ zu berechnen. Im k-ten Schritt wird nur die j_k-te Spalte und die i_k-te Zeile der Matrix R_{k+1} benutzt

um R_k daraus aufzubauen, deshalb müssen wir auch nur diese berechnen. Durch

$$\begin{aligned} \tilde{v}_k &:= A^H_{i_k,1:n} - \sum_{l=1}^{k-1} (u_l)_{i_k} v_l, \\ v_k &:= (\tilde{v}_k)^{-1}_{j_k} \tilde{v}_k, \\ u_k &:= A_{1:m,j_k} - \sum_{l=1}^{k-1} (v_l)_{j_k} u_l \end{aligned}$$

für $k = 1, 2, ...$ können U und V spaltenweise aus den Vektoren $u_k \in \mathbb{C}^m$ und $v_k \in \mathbb{C}^n$ mit einer Zeitkomplexität von $\mathcal{O}(k^2(m+n))$ bestimmt werden. Dabei entspricht u_k der Spalte $(R_{k-1})_{1:m,j_k}$ und $\tilde{v}_k$ der transponierten Zeile $(R_{k-1})_{i_k,1:n}$ aus den oben beschriebenen Matrizen R_k.

Zusammenfassend ergibt sich der folgende Algorithmus.

Vorgabe: $k = 1$ und $Z = \emptyset$
repeat
  wähle i_k wie in [7] beschrieben
  $\tilde{v}_k := A_{i_k,1:n}$
  for $l = 1, ..., k-1$ **do** $\tilde{v}_k := \tilde{v}_k - (u_l)_{i_k} v_l$
  $Z := Z \cup \{i_k\}$
  if $\tilde{v}_k$ nicht verschwindet **then**
    $j_k := \max_{j=1,...,n} |(\tilde{v}_k)_j|$
    $v_k := (\tilde{v}_k)^{-1}_{j_k} \tilde{v}_k$
    $u_k := A_{1:m,j_k}$
    for $l = 1, ..., k-1$ **do** $u_k := u_k - (v_l)_{j_k} u_l$
    k:=k+1
  endif
until Abbruchbedingung erfüllt oder $Z = \{1, ..., m\}$

Algorithmus 3.1: ACA-Verfahren.

Ein mögliches Abbruchkriterium ist

$$\|u_{k+1}\|_2 \|v_{k+1}\|_2 \leq \frac{\varepsilon(1-\eta)}{1+\varepsilon} \|\sum_{l=1}^{k} u_l v_l^H\|_F, \tag{3.7}$$

wobei η die Konstante aus der Fernfeldbedingung ist. Seine Überprüfung geht sehr schnell, weil nur die zuletzt hinzugefügte Zeile und Spalte betrachtet werden müssen.

Trotzdem garantiert es die Einhaltung einer vorgeschriebenen Genauigkeit für die ganze Matrix. Die Anzahl k der Zeilen und Spalten, die benötigt werden um die Abbruchbedingun zu erfüllen, hängt nur logarithmisch von der geforderten Genauigkeit ε ab.

Der Fehler bei der Approximation durch ACA ist für einen fernfeldzulässigen Block und eine asymptotisch glatte Kernfunktion mit dem bei der Interpolation auf einem beliebigen Funktionensystem vergleichbar, insbesondere mit dem bei der Polynominterpolation, vergleiche Satz 3.4.2. Für Details und Beweise sei auf [7] verwiesen.

Es wird von Nutzen sein eine alternative Formulierung das ACA-Algorithmus zu betrachten. Dabei sollen statt U und V die Matrizen $A^U \in \mathbb{R}^{m\times k}, A_k^{-1} \in \mathbb{R}^{k\times k}$ und $A^V \in \mathbb{R}^{k\times n}$ gespeichert werden, so dass

$$A \approx UV^H = A^U A_k^{-1} A^V \tag{3.8}$$

gilt. Dabei besteht $A^U = A_{1:m,j_{1:k}}$ aus den selben Spalten der Originalmatrix A, die in der ursprünglichen Version von ACA für die Konstruktion von U gewählt wurden, $A^V = A_{i_{1:k},1:n}$ aus den selben Zeilen wie für V und $A_k = A_{i_{1:k},j_{1:k}}$ aus den Pivotelementen. Die Zerlegung $A \approx A^U A_k^{-1} A^V$ in (3.8) wird auch als Pseudoskelett bezeichnet, vergleiche [12].

Zur effizienten Behandlung der Matrix A_k verwenden wir ihre LU-Zerlegung, die im Verlauf des Algorithmus einfach schrittweise mit aufgebaut werden kann, für Details siehe [7] und [1].

Die alternative Form von ACA kann ebenfalls in $\mathcal{O}(k^2(m+n))$ Schritten berechnet werden. Die Pseudoskelttdarstellung benötigt $\mathcal{O}(k(m+n+k))$ Speichereinheiten. Das sind k^2 mehr als bei der äußeren Produktform (3.1). Da k klein ist, stellt dies aber keinen großen Nachteil dar. Der haupsächliche Vorteil der alternativen Darstellung ist, dass die Matrizen A^U, A_k und A^V ausschließlich aus originalen Matrixeinträgen bestehen und somit wichtige Eigenschaften von A erben, insbesondere die asymptotische Glattheit der Einträge.

Kapitel 4

Rekompressionsverfahren zu ACA

Obwohl das ACA-Verfahren, siehe Algorithmus 3.1, sehr gute Niedrigrangapproximationen für die Unterblöcke $A_{ts} \in \mathbb{C}^{m\times n}$ von Steifigkeitsmatrizen elliptischer Differentialoperatoren liefert, kann der Approximant immernoch Redundanzen enthalten. Dies kann in der speziellen Form der zu Grunde liegenden Geometrie oder des Operators begründet sein. In [1] wurde daher ein effizientes Verfahren zur Rekompression der Niedrigrangmatrix $A_{ts} \approx A^U A_k^{-1} A^V$ in der Pseudoskelettdarstellung (3.8) mit der kurzen Bezeichnung RACA für Recompression for Adaptive Cross Approximation vorgestellt.
Wir betrachten anfangs nur die Matrix $A^U = A_{1:m,j_{1:k}} \in \mathbb{C}^{m\times k}, m := |t|$, da $A_k^{-1} \in \mathbb{C}^{k\times k}$ mit $A_k = A_{i_{1:k},j_{1:k}}$ klein und $A^V = A_{i_{1:k},1:n} \in \mathbb{C}^{k\times n}, n := |s|$, ähnlich wie A^U behandelbar ist. A^U enthält nach dem alternativen ACA-Verfahren ausschließlich ausgewählte, originale Matrixeinträge von A_{ts} und ist damit ebenfalls asymptotisch glatt (2.1.6), wenn A_{ts} es ist. Diese Eigenschaft hat die Steifigkeitsmatrix des Einfachschichtpotentialoperators und damit ihre Unterblöcke, da ihre Einträge nur Auswertungen der Singularitätenfunktion enthält, welche nach Kapitel 2 asymptotisch glatt ist.

4.1 RACA beim Einfachschichtpotential

Sei $S(x, y)$ die Singularitätenfunktion eines elliptischen Operators, dann haben die Einträge von $A^U \in \mathbb{C}^{m\times k}$ nach der Randelementmethode die Form:

$$a_{ij}^U = \int_\Gamma \int_\Gamma S(x, y)\varphi_i(x)\psi_j(y)\,\mathrm{d}s_x\,\mathrm{d}s_y,$$

wobei $i = 1, ..., m, j = 1, ..., k$ die Indexmengen des Unterblocks beschreiben sollen, auch falls diese nicht zusammenhängend sind oder nicht mit der ursprünglichen Nummerierung in der gesamten Matrix übereinstimmen. φ_i, ψ_j sind die Test- und Ansatzbasisfunktionen der Randelementmethode, mit $\operatorname{supp}\varphi_i \subset D_1, \operatorname{supp}\psi_j \subset D_2$. Wir erinnern uns, dass D_2 im Fernfeld von D_1 liegt, da A_{ts} sonst nicht zulässig wäre und demnach nicht hätte approximiert werden können.
Die Einträge von A^U können bezüglich x oder y durch Tschebyscheffpolynome interpoliert werden, wenn die Funktionswerte von S an den Tschebyscheffknoten bekannt sind.

Satz 4.1.1. *Seien D_1 konvex, S eine asymptotisch glatte Funktion, die $c\gamma\eta \leq 1$ mit den Bezeichnungen aus Definition 2.1.6 erfüllt, $\tilde{A}^U$ die Interpolationsmatrix zu A^U nach x,*

$$\tilde{a}^U_{ij} = \int_\Gamma \int_\Gamma \boldsymbol{\mathcal{I}}^x_p S(x,y)\varphi_i(x)\psi_j(y)\,\mathrm{d}s_x\,\mathrm{d}s_y,$$

dann ist der Approximationsfehler bezüglich der Frobeniusnorm durch

$$\|A^U - \tilde{A}^U\|_F \leq \bar{c}\left(1 + \frac{2}{\pi}\log p\right)^{d-1}\left(\frac{\gamma\eta}{4}\right)^p \|A^U\|_F$$

beschränkt.

Für den Beweis des Satzes sei auf [1] verwiesen.

Setzen wir (1.5) für $\boldsymbol{\mathcal{I}}^x_p$ ein, erhalten wir mit $\xi_\nu = 2\frac{x_\nu - a_\nu}{b_\nu - a_\nu} - 1, \nu = 1, ..., d$,

$$\tilde{a}^U_{ij} = \int_\Gamma \int_\Gamma \sum_{\boldsymbol{\alpha}\in\mathbb{N}^d, \boldsymbol{\alpha}<p} c_{\boldsymbol{\alpha}} \prod_{\nu=1}^d T_{\alpha_\nu}(\xi_\nu)\varphi_i(x)\psi_j(y)\,\mathrm{d}s_x\,\mathrm{d}s_y.$$

Da ξ_ν von y und $c_{\boldsymbol{\alpha}}$ von x unabhängig sind, können wir durch

$$\tilde{a}^U_{ij} = \sum_{\boldsymbol{\alpha}\in\mathbb{N}^d, \boldsymbol{\alpha}<p} \int_\Gamma \prod_{\nu=1}^d T_{\alpha_\nu}(\xi_\nu)\varphi_i(x)\,\mathrm{d}s_x \int_\Gamma c_{\boldsymbol{\alpha}}\psi_j(y)\,\mathrm{d}s_y \tag{4.1}$$

die Matrix $\tilde{A}^U = B^U X^{ch}$ faktorisieren. Dabei gilt $k' = p^d, B^U \in \mathbb{C}^{m\times k'}, X^{ch} \in \mathbb{C}^{k'\times k}$,

$$\begin{aligned} b^U_{i\boldsymbol{\alpha}} &= \int_\Gamma \prod_{\nu=1}^d T_{\alpha_\nu}(\xi_\nu)\varphi_i(x)\,\mathrm{d}s_x, \\ x^{ch}_{\boldsymbol{\alpha}j} &= \int_\Gamma c_{\boldsymbol{\alpha}}\psi_j(y)\,\mathrm{d}s_y. \end{aligned}$$

Die Matrix B^U lässt sich aufgrund der Rekursionsformel (1.2) mit nur $\mathcal{O}(mk')$ Operationen sehr schnell aufstellen, wann immer man sie benötigt und muss daher nicht gespeichert werden. Das Integral tritt nur bei der Diskretisierung mit dem Galerkinverfahren auf. Wie in Kapitel 1 beschrieben, wird es während der Implementation, durch eine Summe über Gaußknoten ersetzt.
Die Verwaltung von X^{ch} ist ein größeres Problem. Hier benötigt man die Auswertungen der Singularitätenfunktion an den Tschebyscheffknoten, die nur mit großem Aufwand oder möglicherweise gar nicht zu berechnen sind. Dabei ist noch nicht einmal garantiert, dass sie die geeignetste Matrix ist, also ob sie folgendes Problem löst:

$$\min_{X \in \mathbb{C}^{k' \times k}} \|A^U - B^U X\|_F.$$

Eine Rang-k-Bestapproximation liefert die Methode der kleinsten Quadrate mit $X^{ls} := V_B \Sigma^+ U_B^H A^U$, dabei entstehen $B^U = U_B \Sigma V_B^H$ aus der Singulärwertzerlegung von B^U und

$$(\Sigma^+)_{ij} = \begin{cases} \sigma_i^{-1}, & \text{falls } i = j \wedge \sigma_i \neq 0, \\ 0, & \text{sonst.} \end{cases}$$

Da X^{ls} zusammen mit B^U die Matrix A^U am besten annähert, demnach besser als X^{ch}, gilt mit steigendem Polynomgrad p wegen Satz 4.1.1:

$$\|A^U - B^U X^{ls}\|_F \leq \varepsilon \|A^U\|_F. \tag{4.2}$$

Die Spalten von B^U sind je nach Geometrie möglicherweise fast linear abhängig. Mit einer QR-Zerlegung $B^U \Pi^U = QR$ ermitteln wir die Anzahl r_B der Spalten, die nötig sind um weiterhin eine Fehlerabschätzung der Größenordnung wie in (4.2) garantieren zu können. Hier sind $\Pi^U \in \mathbb{C}^{k' \times k'}$ eine Permutationsmatrix, $Q \in \mathbb{C}^{m \times m}$ unitär und $R \in \mathbb{C}^{m \times k'}$ eine obere Dreiecksmatrix. Wir suchen nun nach einer Zerlegung von R,

$$R = \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix},$$

mit möglichst großem R_{22}, das immernoch

$$\| \begin{bmatrix} 0 & R_{22} \end{bmatrix} X^{ls} \|_F \leq \varepsilon \|A^U\|_F$$

erfüllt. Sei r_B die Größe der verbleibenden, quadratischen Untermatrix R_{11} und $B^U \Pi^U_{r_B}$ die Reduzierung von B^U auf die zu R_{11} gehörigen Spalten, dann gilt:

$$\min_{X \in \mathbb{C}^{r_B \times k}} \|A^U - B^U \Pi^U_{r_B} X^{ls}\|_F \leq 2\varepsilon \|A^U\|_F.$$

Abhängig von der Form der Matrix A^U können mehr Einsparungen gemacht werden. Zu diesem Zweck zerlegen wir $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}, Q_1 \in \mathbb{C}^{m \times r}$ mit einem möglichst kleinen r, dass gerade noch

$$\|Q_2^H A^U\|_F \leq 2\varepsilon \|A^U\|_F$$

erfüllt. In [1] wird gezeigt, dass ein solches r die Existenz einer Matrix $X^U \in \mathbb{C}^{r \times k}$ garantiert, für welche

$$\|A^U - B^U \Pi_r^U X^U\|_F \leq 2\varepsilon \|A^U\|_F$$

gilt. $B^U \Pi_r^U$ ist die Reduktion von B^U auf die r zu Q_1 gehörenden Spalten. Man erhält besagtes X^U durch Rückwärtssubstitution aus $\hat{R}X = Q_1^H A^U$, wobei $\hat{R}$ die linke, obere $r \times r$ Untermatrix von R_{11} ist.

Da wir nur die Matrix X^U und die Permutation Π_r^U speichern müssen, reduziert sich der Speicherbedarf von $\mathcal{O}(mk)$ auf $\mathcal{O}(r(k+1))$, wobei $r \leq r_B \leq k' = p^d < m$ gilt.

Bemerkung 4.1.2. Die Interpolation bezüglich y hätte nicht das gewünschte Ergebnis zur Folge gehabt, da die Faktorisierung zu $\tilde{A}^U = X^{ch} B^U$ geführt hätte. Dabei wäre $k' = p^d, X^{ch} \in \mathbb{C}^{m \times k'}, B^U \in \mathbb{C}^{k' \times k}$,

$$x_{i\boldsymbol{\alpha}}^{ch} = \int_\Gamma c_{\boldsymbol{\alpha}} \varphi_i(x) \, \mathrm{d}s_x,$$

$$b_{\boldsymbol{\alpha}j}^U = \int_\Gamma \prod_{\nu=1}^{d} T_{\alpha_\nu}(\epsilon_\nu) \psi_j(y) \, \mathrm{d}s_y,$$

mit $\epsilon_\nu = 2\frac{y_\nu - a_\nu^y}{b_\nu^y - a_\nu^y} - 1, \nu = 1, ..., d$, und die Matrix X^U, die letztendlich gespeichert werden muss, wäre um ein Vielfaches größer.
Ebenso würde die Interpolation bezüglich x bei $A^V \in \mathbb{C}^{k \times n}$ von Nachteil sein, dafür erhalten wir hier durch Tschebyscheffpolynominterpolation bezüglich y die gewünschte Faktorisierung $\tilde{A}^V = X^{ch} B^V$. Hier erhalten wir mit $k' = p^d, X^{ch} \in \mathbb{C}^{k \times k'}, B^V \in \mathbb{C}^{k' \times n}$ und den Einträgen

$$x_{i\boldsymbol{\alpha}}^{ch} = \int_\Gamma c_{\boldsymbol{\alpha}} \varphi_i(x) \, \mathrm{d}s_x,$$

$$b_{\boldsymbol{\alpha}j}^V = \int_\Gamma \prod_{\nu=1}^{d} T_{\alpha_\nu}(\epsilon_\nu) \psi_j(y) \, \mathrm{d}s_y,$$

die erstrebenswerten Dimensionen mit dem kleinen, zu speichernden X^V.

Zur Unterscheidung bezeichne r_U das r von eben und r_V das r, das sich bei der Rekompression der Matrix A^V ergibt. Wenn $X^U \in \mathbb{C}^{r_U \times k}, X^V \in \mathbb{C}^{k \times r_V}$ und die Permutationen $\Pi^U_{r_U}, \Pi^V_{r_V}$ wie beschrieben bestimmt worden sind, so dass für gegebene Genauigkeit $\varepsilon > 0$ und die mit $\Pi^U_{r_U}$ und $\Pi^V_{r_V}$ reduzierten Basismatrizen $B^U_r \in \mathbb{C}^{m \times r_U}, B^V_r \in \mathbb{C}^{r_V \times n}$ gilt:

$$\begin{aligned} \|A^U - B^U_r X^U\|_F &\leq \varepsilon \|A^U\|_F, \\ \|A^V - X^V B^V_r\|_F &\leq \varepsilon \|A^V\|_F, \end{aligned}$$

erhalten wir für den ganzen Matrixblock:

$$A_{ts} \approx A^U A_k^{-1} A^V \approx B^U_r X^U A_k^{-1} X^V B^V_r = B^U_r C B^V_r.$$

Da die Basismatrizen B^U, B^V fest sind und schnell erzeugt werden können, müssen nur die Matrix $C = X^U A_k^{-1} X^V \in \mathbb{C}^{r \times r}$ und die Permutationen gespeichert werden um die Approximation an $A_{ts} \in \mathbb{C}^{m \times n}$ zu beschreiben. Für nähere Informationen siehe [1].

4.2 RACA beim Doppelschichtpotential

Der Doppelschichtpotentialoperator

$$(\mathcal{K} u)(x) = \int_\Gamma \langle A n_y, \nabla_y S(x-y) \rangle u(y) dy,$$

hier ist n_y der Normalenvektor an der Stelle y, hat nicht ganz so schöne Eigenschaften wie das eben behandelte Einfachschichtpotential.

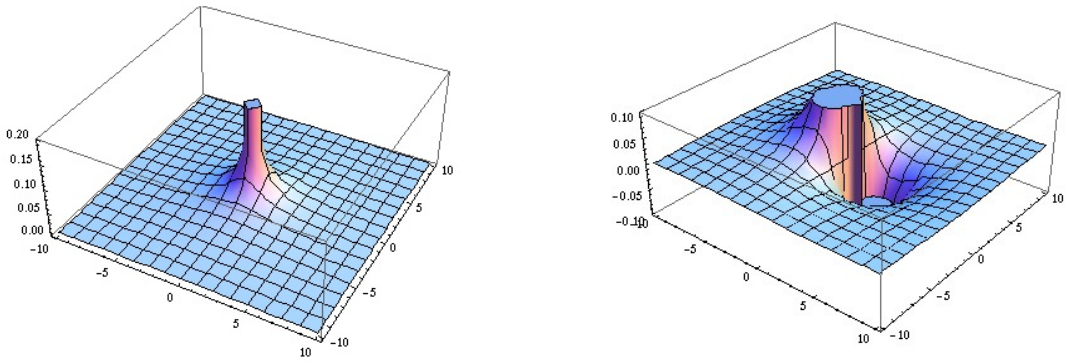

Abbildung 4.1: Kernfunktionen, links des Einfach-, rechts des Doppelschichtpotentials

Zum einen war die diskretisierte Steifigkeitsmatrix symmetrisch und zum anderen asymptotisch glatt bezüglich beider Variablen. Wir verlieren die Symmetrie und die asymptotische Glattheit bezüglich y durch den Normalenvektor n_y, da wir keine weiteren Forderungen an die Oberflächenbeschaffenheit der zu Grunde liegenden Geometrie stellen wollen. Schon eine Kante zerstört jede Glattheit, weil der Normalenvektor dort springt.

Zur Verdeutlichung betrachten wir in Abbildung 4.1 die Kernfunktionen des Einfach- (2.8) und des Doppelschichtpotentials (4.3) vom Laplaceoperator im Bereich $x, y \in \mathbb{R}^3, x = 0, y_1, y_2 \in [-10, 10], y_3 = 0$ überall mit dem gleichen Normalenvektor $(1, 0, 0)^T$.
Nun stellen wir uns als zu Grunde liegende Geometrie eine kleine Treppe vor. Für alle $y_1 < 2$ habe sie den Normalenvektor $(1, 0, 0)^T$ und für die übrigen $(0, 1, 0)^T$. Das Einfachschichtpotentiel behält seine glatte Form unabhängig von der Treppe. Beim Doppelschichtpotential führt es jedoch zu dem in Abbildung 4.2 dargestellten Funktionsverlauf.

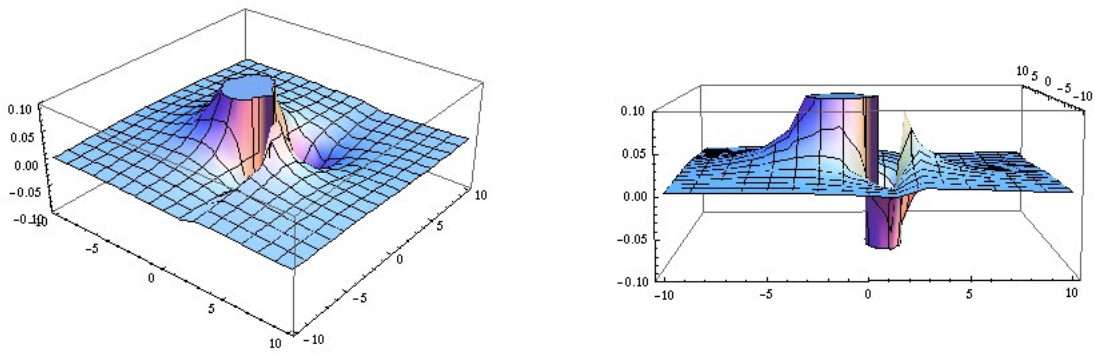

Abbildung 4.2: Kernfunktion des Laplace-Doppelschichtpotentials an einer Stufe aus verschiedenen Perspektiven

Wir betrachten wieder einen zulässigen Block $A_{ts} \approx A^U A_k^{-1} A^V$. Für das Galerkin-, und damit implizit auch für das Nyström- und das Kollokationsverfahren, erhält man bei der Randelementmethode zum Doppelschichtpotential folgende Steifigkeitsmatrix:

$$a_{ij} = \int_\Gamma \int_\Gamma \langle An_y, \nabla_y S(x-y)\rangle \varphi_i(x)\psi_j(y)\,\mathrm{d}s_x\,\mathrm{d}s_y.$$

Nach dem alternativen ACA-Verfahren bestehen $A^U = A_{1:m,j_{1:k}} \in \mathbb{C}^{m\times k}, m := |t|$, $A_k = A_{i_{1:k},j_{1:k}}$ und $A^V = A_{i_{1:k},1:n} \in \mathbb{C}^{k\times n}, n := |s|$, aus originalen Matrixeinträgen, also

gilt auch

$$a_{ij}^U = \int_\Gamma \int_\Gamma \langle An_y, \nabla_y S(x-y)\rangle \varphi_i(x)\psi_j(y)\,\mathrm{d}s_x\,\mathrm{d}s_y.$$

Wie beim Einfachschichtpotential wollen wir bezüglich x eine Tschebyscheffpolynominterpolation durchführen. Da die Kernfunktion $\kappa(x,y) = \langle An_y, \nabla_y S(x-y)\rangle$ in dieser Variable asymptotisch glatt ist, erlaubt uns Satz 4.1.1 A^U analog zum Einfachschichtpotential zu behandendeln.

4.2.1 Die Basis

Um einen Approximanten für A^V zu finden, müssen wir bezüglich y interpolieren, vergleiche Bemerkung 4.1.2. Da wir dabei wegen des Normalenvektors allerdings keine asymptotische Glattheit haben, können wir keine Fehlerabschätzung wie in Satz 4.1.1 erwarten.
In vielen Fällen lässt sich der Normalenvektor jedoch durch geschickte Zerlegung vom asymptotisch glatten Teil, welcher polynomapproximierbar ist, trennen. Wir betrachten wieder unsere Standardbeispiele.

Beispiel 4.2.1. Das Doppelschichtpotential des Laplaceoperators (2.1) hat die Form

$$(\mathscr{K}u)(x) = \int_\Gamma \frac{\langle x-y, n_y\rangle}{|x-y|^3} u(y)\,\mathrm{d}s_y. \tag{4.3}$$

Beispiel 4.2.2. Beim Helmholtzoperator (2.2) nimmt das Doppelschichtpotential die folgende Form an:

$$(\mathscr{K}u)(x) = \int_\Gamma \frac{\langle x-y, n_y\rangle\, e^{i\kappa|x-y|}(1-i\kappa\,|x-y|)}{|x-y|^3} u(y)\,\mathrm{d}s_y \tag{4.4}$$

Neben (4.3) und (4.4) gibt es weitere elliptische Differentialoperatoren, deren Kernfunktionen sich als Produkt von $\langle x-y, n_y\rangle$ mit einer bezüglich beider Variablen asymptotisch glatten Funktion $g(x,y)$ schreiben lassen. Wir nehmen künftig an, wir haben einen solchen Doppelschichtpotentialoperator $\mathscr{K}$,

$$(\mathscr{K}u)(x) = \int_\Gamma \langle x-y, n_y\rangle\, g(x,y)u(y)\,\mathrm{d}s_y \tag{4.5}$$

mit einer per Galerkin- oder der anderen Verfahren diskretisierten Steifigkeitsmatrix:

$$a_{ij} = \int_\Gamma \int_\Gamma \langle x-y, n_y\rangle \, g(x,y)\varphi_i(x)\psi_j(y)\,\mathrm{d}s_x\,\mathrm{d}s_y.$$

Nach der Zerlegung, interpolieren wir nur den asymptotisch glatten Teil $g(x,y)$ und erhalten die Matrix $\tilde{A}^V \in \mathbb{C}^{m\times k}$,

$$\begin{aligned}\tilde{a}_{ij}^V &= \int_\Gamma \int_\Gamma \langle x-y, n_y\rangle \, \boldsymbol{\mathcal{I}}_p^y g(x,y)\varphi_i(x)\psi_j(y)\,\mathrm{d}s_x\,\mathrm{d}s_y \\ &= \int_\Gamma \int_\Gamma \langle x-y, n_y\rangle \sum_{\boldsymbol{\alpha}\in\mathbb{N}^d,\boldsymbol{\alpha}<p} c_{\boldsymbol{\alpha}} \prod_{\nu=1}^d T_{\alpha_\nu}(\epsilon_\nu)\varphi_i(x)\psi_j(y)\,\mathrm{d}s_x\,\mathrm{d}s_y.\end{aligned}$$

Schreiben wir das Skalarprodukt $\langle x-y, n_y\rangle$ als

$$\begin{aligned}\langle x-y, n_y\rangle &= \langle x, n_y\rangle - \langle y, n_y\rangle \\ &= \sum_{l=1}^d x_l n_{yl} - \langle y, n_y\rangle,\end{aligned}$$

wird ähnlich wie oben eine Faktorisierung der Matrix möglich, da die transformierten Tschebyscheffknoten $\epsilon_\nu = 2\frac{y_\nu - a_\nu}{b_\nu - a_\nu} - 1, \nu = 1,...,d$, von x und $c_{\boldsymbol{\alpha}}$ von y unabhängig sind.

$$\begin{aligned}\tilde{a}_{ij}^V &= \sum_{\boldsymbol{\alpha}\in\mathbb{N}^d,\boldsymbol{\alpha}<p} \int_\Gamma \int_\Gamma \Big(\sum_{l=1}^d x_l n_{yl} - \langle y, n_y\rangle\Big) c_{\boldsymbol{\alpha}} \prod_{\nu=1}^d T_{\alpha_\nu}(\epsilon_\nu)\varphi_i(x)\psi_j(y)\,\mathrm{d}s_x\,\mathrm{d}s_y \\ &= \sum_{\boldsymbol{\alpha}\in\mathbb{N}^d,\boldsymbol{\alpha}<p} \Bigg(\sum_{l=1}^d \int_\Gamma \int_\Gamma x_l n_{yl} c_{\boldsymbol{\alpha}} \prod_{\nu=1}^d T_{\alpha_\nu}(\epsilon_\nu)\varphi_i(x)\psi_j(y)\,\mathrm{d}s_x\,\mathrm{d}s_y \\ &\quad - \int_\Gamma \int_\Gamma \langle y, n_y\rangle \, c_{\boldsymbol{\alpha}} \prod_{\nu=1}^d T_{\alpha_\nu}(\epsilon_\nu)\varphi_i(x)\psi_j(y)\,\mathrm{d}s_x\,\mathrm{d}s_y\Bigg) \\ &= \sum_{\boldsymbol{\alpha}\in\mathbb{N}^d,\boldsymbol{\alpha}<p} \Bigg(\sum_{l=1}^d \int_\Gamma x_l c_{\boldsymbol{\alpha}}\varphi_i(x)\,\mathrm{d}s_x \int_\Gamma n_{yl} \prod_{\nu=1}^d T_{\alpha_\nu}(\epsilon_\nu)\psi_j(y)\,\mathrm{d}s_y \\ &\quad - \int_\Gamma \prod_{\nu=1}^d c_{\boldsymbol{\alpha}}\varphi_i(x)\,\mathrm{d}s_x \int_\Gamma \langle y, n_y\rangle \, T_{\alpha_\nu}(\epsilon_\nu)\psi_j(y)\,\mathrm{d}s_y\Bigg)\end{aligned}$$

Daraus kann man die Faktorisierung $\tilde{A}^V = X^{ch}B^V$ ablesen mit $X^{ch} \in \mathbb{C}^{k\times k'}$,

$$X^{ch}_{i\boldsymbol{\alpha}l} = \begin{cases} \int\limits_{\Gamma} x_l c_{\boldsymbol{\alpha}} \varphi_i(x)\,\mathrm{d}s_x, & \text{falls } l = 1,...,d, \\ \int\limits_{\Gamma} c_{\boldsymbol{\alpha}} \varphi_i(x)\,\mathrm{d}s_x, & \text{falls } l = d+1, \end{cases} \tag{4.6}$$

sowie $B^V \in \mathbb{C}^{k' \times n}$,

$$b^V_{\boldsymbol{\alpha}lj} = \begin{cases} \int\limits_{\Gamma} n_{yl} \prod\limits_{\nu=1}^{d} T_{\alpha_\nu}(\epsilon_\nu)\psi_j(y)\,\mathrm{d}s_y, & \text{falls } l = 1,...,d, \\ -\int\limits_{\Gamma} \langle y, n_y \rangle \prod\limits_{\nu=1}^{d} T_{\alpha_\nu}(\epsilon_\nu)\psi_j(y)\,\mathrm{d}s_y, & \text{falls } l = d+1. \end{cases} \tag{4.7}$$

Es ist wichtig zu beachten, dass $k' = (d+1)p^d$ in diesem Fall um den konstanten Faktor $d+1$ größer ist als in allen bisher betrachteten Fällen. Es wird jedoch im übernächsten Abschnitt gezeigt, dass die große Matrix B^V immernoch sehr schnell mit Hilfe der Rekursionsvorschrift für Tschebyscheffpolynome aufgestellt werden kann und deshalb nicht gespeichert werden muss.

Die Matrizen B^V und X^{ch} werden über den Multiindex $\alpha \in \mathbb{N}^d$ und den zusätzlichen Index $l = 1,...,d+1$ angesprochen. Zur Veranschaulichung soll Abbildung 4.3 dienen. Hier ist die genaue Indizierung von B^V für den Fall $d = 2$ und $p = 2$ dargestellt.

			j	1	2	3	4	5	.	.	.	.	n
l	α_1	α_2											
1	0	0											
1	0	1											
1	1	0											
1	1	1											
2	0	0											
2	0	1											
2	1	0											
2	1	1											
3	0	0											
3	0	1											
3	1	0											
3	1	1											

Abbildung 4.3: Beispiel für die Indizierung der Matrix $B^V \in \mathbb{C}^{2^2 \cdot 3 \times n}$, hier blau dargestellt

4.2.2 Diskretisierung

Je nachdem für welches Diskretisierungsverfahren man sich entscheidet, ändert sich das Aussehen der Matrizen B. B^U ist wie oben die Matrix, die nach der Faktorisierung der Interpolation bezüglich x entstanden ist und der Approximation von A^U dient, B^V die Matrix aus (4.7), die A^V approximieren soll.

Beim Nyströmverfahren gilt der Spezialfall $\varphi_i(x) = \delta(x - x^i)$, $\psi_j(y) = \delta(y - y^j)$, dabei sind die $x^i, y^j \in \mathbb{R}^d$ spezielle Kollokationspunkte, in unserem Fall jeweils die Dreiecksmittelpunkte der Dreiecke τ_i, beziehungsweise τ_j. Damit haben die Matrizen $B^U \in \mathbb{C}^{m \times p^d}, B^V \in \mathbb{C}^{(d+1)p^d \times n}$ die Form:

$$
\begin{aligned}
b^U_{i\boldsymbol{\alpha}} &= \prod_{\nu=1}^{d} T_{\alpha_\nu}(\xi^i_\nu), \\
b^V_{\boldsymbol{\alpha} l j} &= \begin{cases} n^j_{yl} \prod_{\nu=1}^{d} T_{\alpha_\nu}(\epsilon^j_\nu), & \text{falls } l = 1, ..., d, \\ -\left\langle y^j, n^j_y \right\rangle \prod_{\nu=1}^{d} T_{\alpha_\nu}(\epsilon^j_\nu), & \text{falls } l = d+1 \end{cases}
\end{aligned}
\tag{4.8}
$$

Dabei sind $\xi^i_\nu = 2\frac{x^i_\nu - a_\nu}{b_\nu - a_\nu} - 1$ die ν-te Koordinate des auf $[-1, 1]^d$ transformierten Punktes x^i und $\epsilon^j_\nu = 2\frac{y^j_\nu - a_\nu}{b_\nu - a_\nu} - 1, \nu = 1, ..., d$, die Transformation von y^j und n^j_{yl} die l-te Koordinate des Normalenvektors im Punkt y^j.

Beim Galerkinverfahren bleiben die Integrale. Im Programm müssen wir sie deshalb numerisch berechnen. Wir integrieren hier der Einfachheit halber jeweils nur über ein Dreieck τ.
Genauer gesagt, haben die Funktionen $\varphi_i(x)$ und $\psi_j(y)$ bei uns die sehr einfache Gestalt:

$$
\varphi_i(x) = \begin{cases} 1, & \text{wenn } x \in \tau_i \\ 0, & \text{sonst} \end{cases}
$$

$$
\psi_j(y) = \begin{cases} 1, & \text{wenn } y \in \tau_j \\ 0, & \text{sonst} \end{cases}
$$

Das Integral simulieren wir mit Hilfe der Gaußquadratur. Dabei ist $g \in \mathbb{N}$ die Anzahl der $(d-1)$-dimensionalen Gaußpunkte x_q und der Gaußgewichte w_q, $q = 1, ..., g$, im Ansatz

$$
\int_\tau f(x)dx = \sum_{q=1}^{g} w_q f(x_q).
$$

Das bedeutet für unsere Matrizen:

$$
\begin{aligned}
b^U_{i\boldsymbol{\alpha}} &= \sum_{q=1}^{g} w_q \prod_{\nu=1}^{d} T_{\alpha_\nu}(\xi_\nu^{i_q}), \\
b^V_{\boldsymbol{\alpha}lj} &= \begin{cases} \sum\limits_{q=1}^{g} w_q n_{yl}^{j_q} \prod\limits_{\nu=1}^{d} T_{\alpha_\nu}(\epsilon_\nu^{i_q}), & \text{falls } l = 1, ..., d, \\ -\sum\limits_{q=1}^{g} w_q \left\langle y^{j_q}, n_y^{j_q} \right\rangle \prod\limits_{\nu=1}^{d} T_{\alpha_\nu}(\epsilon_\nu^{j_q}), & \text{falls } l = d+1. \end{cases}
\end{aligned} \tag{4.9}
$$

Hier sind x^{i_q} der q-te Gaußknoten im Dreieck τ_i, y^{j_q} der q-te Gaußknoten im Dreieck τ_j, sowie $\xi_\nu^{i_q}$ und $\epsilon_\nu^{i_q}$ deren Transformationen auf das Einheitsintervall in Richtung der Koordinate ν.
Bei $b^V_{\boldsymbol{\alpha}lj}$ können wir $n_y^{j_q}$ durch n_y^j, die Normale am Mittelpunkt des Dreiecks, ersetzen, weil die Normale auf allen Punkten eines Dreiecks gleich ist. Analog vereinfachen wir $\left\langle y^{j_q}, n_y^{j_q} \right\rangle = \left\langle y^j, n_y^j \right\rangle$, weil die Verschiebung vom Dreiecksmittelpunkt zu den Gaußpunkten senkrecht auf der Normalen steht.

Beim Kollokationsverfahren werden die Basisfunktionen des diskreten Raumes unterschiedlich behandelt. Die einen werden wie beim Nyströmverfahren ersetzt und die anderen wie im Galerkinfall. Je nachdem wie man sich entscheidet, haben die Matrizen das Aussehen $B^U \in \mathbb{C}^{m \times p^d}$, $b^U_{i\boldsymbol{\alpha}}$ wie in (4.8), $B^V \in \mathbb{C}^{(d+1)p^d \times n}$, $b^V_{\boldsymbol{\alpha}lj}$ wie in (4.9) oder umgekehrt.

4.2.3 Generierung der Matrizen

Analog zum Einfachschichtpotentialoperator können auch diese Matrizen sehr effektiv aufgestellt werden. Da B^U die gleiche Form hat wie im ersten Abschnitt, kann es analog mit $\mathcal{O}(mp^d)$ Operationen aufgestellt werden, vergleiche [1].

Betrachten wir die Matrixeinträge von $B^V \in \mathbb{C}^{(d+1)p^d \times n}$ zunächst im Nyströmfall (4.8):

$$
b^V_{\boldsymbol{\alpha}lj} \;=\; c_l^j \prod_{\nu=1}^{d} T_{\alpha_\nu}(\epsilon_\nu^j).
$$

Dabei sollen die c_l^j den jeweiligen Faktor in jeder Zeile vor dem Tschebyscheffpolynom

repräsentieren, also

$$c_l^j = \begin{cases} n_{yl}^j, & \text{falls } l = 1, ..., d, \\ -\left\langle y^j, n_y^j \right\rangle, & \text{falls } l = d+1. \end{cases}$$

Satz 4.2.3. *Die Matrix $B^V \in \mathbb{C}^{(d+1)p^d \times n}$ lässt sich im Nyströmfall (4.8) mit $\mathcal{O}((d+1)p^d n)$ Operationen aufstellen.*

Beweis: Aus der Rekursionsformel (1.2) folgt für $0 \leq \alpha_\nu \leq 1 \forall \nu = 1, ..., d$:

$$b_{\boldsymbol{\alpha} lj}^V = c_l^j \prod_{\nu=1}^{d} (\epsilon_\nu^j)^{\alpha_\nu}.$$

Wenn es einen Index $i = 1, ..., d$ gibt, so dass $\alpha_i > 1$ gilt, dann haben die Einträge die Form:

$$b_{\boldsymbol{\alpha} lj}^V = 2\epsilon_i^j b_{(\boldsymbol{\alpha}-e_i)lj}^V - b_{(\boldsymbol{\alpha}-2e_i)lj}^V,$$

denn es gilt

$$\begin{aligned} b_{\alpha lj}^V &= c_l^j \prod_{\nu=1}^{d} T_{\alpha_\nu}(\epsilon_\nu^j) \\ &= c_l^j T_{\alpha_i}(\epsilon_i^j) \prod_{\nu=1, \nu \neq i}^{d} T_{\alpha_\nu}(\epsilon_\nu^j) \\ &= c_l^j (2\epsilon_i^j T_{\alpha_i - 1}(\epsilon_i^j) - T_{\alpha_i - 2}(\epsilon_i^j)) \prod_{\nu=1, \nu \neq j}^{d} T_{\alpha_\nu}(\epsilon_\nu^j) \\ &= 2c_l^j \epsilon_i^j T_{\alpha_i - 1}(\epsilon_i^j) \prod_{\nu=1, \nu \neq i}^{d} T_{\alpha_\nu}(\epsilon_\nu^j) - c_l^j T_{\alpha_i - 2}(\epsilon_i^j) \prod_{\nu=1, \nu \neq i}^{d} T_{\alpha_\nu}(\epsilon_\nu^j) \\ &= 2\epsilon_j^i b_{(\boldsymbol{\alpha}-e_i)lj}^V - b_{(\boldsymbol{\alpha}-2e_i)lj}^V. \end{aligned}$$

Auf diese Weise hat der jeweilige Faktor c_l^j nur an $(d+1)2^d n$ Stellen Einfluss auf die Berechnung der Matrix B^V. Diese sind dunkelblau in Abbildung 4.4 dargestellt. Jeder Eintrag wird entweder aus c_l^j und den transformierten Koordinaten oder aus zwei bereits berechneten Einträgen bestimmt. Die gesamte Matrix kann dadurch spaltenweise von oben nach unten mit einem Aufwand von $\mathcal{O}((d+1)p^d n)$ aufgestellt werden. □

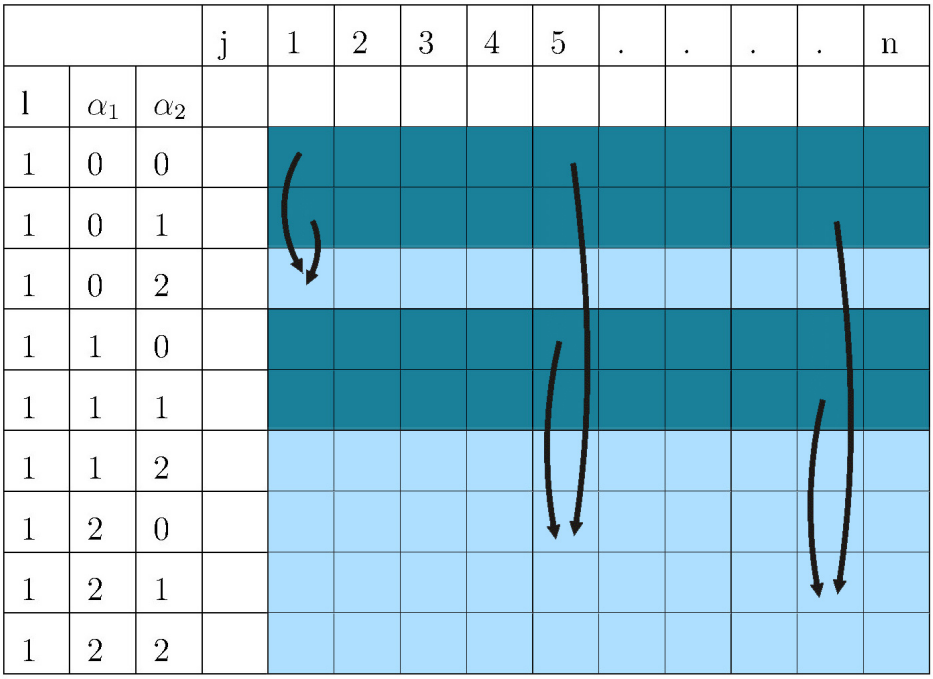

Abbildung 4.4: Beispiel für die Generierung des ersten $p^d \times n$ Teils von $B^V \in \mathbb{C}^{(d+1)p^d \times n}$ durch die Rekursionsformel (1.2) mit $d = 2$ und $p = 3$

Satz 4.2.4. *Die Matrix $B^V \in \mathbb{C}^{(d+1)p^d \times n}$ lässt sich im Galerkinfall (4.8) mit $\mathcal{O}((d+1)p^d gn)$ Operationen aufstellen.*

Beweis: Wir interpretieren die Gaußknoten als eigenständige Punkte und stellen die Matrix $\bar{B}^V \in \mathbb{C}^{(d+1)p^d \times gn}$ wie im Nyströmfall für die gn Punkte $y^{j_q}, q = 1, ..., g$, auf:

$$\bar{b}^V_{\boldsymbol{\alpha} lqj} = \begin{cases} w_q n^j_{yl} \prod_{\nu=1}^{d} T_{\alpha_\nu}(\epsilon^{i_q}_\nu), & \text{falls } l = 1, ..., d, \\ -w_q \left\langle y^j, n^j_y \right\rangle \prod_{\nu=1}^{d} T_{\alpha_\nu}(\epsilon^{j_q}_\nu), & \text{falls } l = d+1. \end{cases} \tag{4.10}$$

Das geschieht nach Satz 4.2.3 mit $\mathcal{O}((d+1)p^d gn)$ Operationen. Noch einmal so viele sind nötig, um $B^V \in \mathbb{C}^{(d+1)p^d \times gn}$ daraus zu berechnen, indem man die jeweils g Zeilen, die zum gleichen Dreieck gehören, aufaddiert. □

Korollar 4.2.5. *Die Matrix $B^V \in \mathbb{C}^{(d+1)p^d \times n}$ lässt sich in jedem der drei betrachteten Diskretisierungsverfahren mit $\mathcal{O}((d+1)p^d n)$ Operationen aufstellen.*

Beweis: Aus Satz 4.2.3 und Satz 4.2.4 folgt die Behauptung auch für das Kollokationsverfahren, da die Matrizen eine der beiden Formen annehmen und die Anzhal g der Gaußknoten konstant ist. □

4.2.4 Bewertung der Approximation

Die Güte der Approximation von A^U und A^V ist von der gleichen Ordnung wie beim Einfachschichtpotential. Für den Beweis im Fall von A^U sei auf [1] verwisen, denn wie gesagt verläuft die Approximation analog zu A^U beim Einfachschichtpotentialoperator.

Wir betrachten also die Matrix $A^V \in \mathbb{C}^{k\times n}$,

$$a_{ij}^V = \int_\Gamma \int_\Gamma \langle x-y, n_y\rangle\, g(x,y)\varphi_i(x)\psi_j(y)\,\mathrm{d}s_x\,\mathrm{d}s_y.$$

mit einer asymptotisch glatten Funktion $g : D_1 \times D_2 \to \mathbb{C}$, siehe Definition 2.1.6, sowie ihren Tschebyscheffapproximanten $A^V \approx \tilde{A}^V \in \mathbb{C}^{k\times n}$,

$$\tilde{a}_{ij}^V \;=\; \int_\Gamma \int_\Gamma \langle x-y, n_y\rangle\, \boldsymbol{\mathcal{I}}_p^y g(x,y)\varphi_i(x)\psi_j(y)\,\mathrm{d}s_x\,\mathrm{d}s_y.$$

Satz 4.2.6. *Seien* $D_1 = \bigcup_{i\in t} X_i \subset \mathbb{R}^d, D_2 = \prod_{\nu=1}^d [a_\nu, b_\nu] = \bigcup_{j\in s} X_j \subset \mathbb{R}^d$ *Gebiete, die die Fernfeldbedingung* $\eta\,\mathrm{dist}(D_1,D_2) \geq \max_{\nu=1,\ldots,d} b_\nu - a_\nu$ *für ein* $\eta > 0, c\gamma\eta < 1$ *erfüllen, dann gilt für die Matrix* A^V *aus* $A_{ts} = A^U A_k^{-1} A^V$

$$\|A^V - \tilde{A}^V\|_F \leq \tilde{c}p\big(1+\frac{2}{\pi}\log p\big)^d\big(\frac{\gamma\eta}{4}\big)^p\|A^V\|_F.$$

Beweis: Wegen Satz 3.4.2 gilt für das asymptotisch glatte $g(x,y)$:

$$|g(x,y) - \boldsymbol{\mathcal{I}}_p^y g(x,y)| \leq \tilde{c}\big(1+\frac{2}{\pi}\log p\big)^d\big(\frac{\gamma\eta}{4}\big)^p|g(x,y)|,$$

damit folgt:

$$
\begin{aligned}
\|A^V - \tilde{A}^V\|_F^2 &= \sum_{i=1}^{m}\sum_{j=1}^{k} \left|a_{ij}^V - \tilde{a}_{ij}^V\right|^2 \\
&= \sum_{i=1}^{m}\sum_{j=1}^{k} \Big| \int_\Gamma \int_\Gamma \langle x-y, n_y\rangle \, g(x,y)\varphi_i(x)\psi_j(y) ds_x ds_y \\
&\quad - \int_\Gamma \int_\Gamma \langle x-y, n_y\rangle \, \boldsymbol{\mathcal{I}}_p^y g(x,y)\varphi_i(x)\psi_j(y) ds_x ds_y \Big|^2 \\
&= \sum_{i=1}^{m}\sum_{j=1}^{k} \Big(\int_\Gamma \int_\Gamma |\, \langle x-y, n_y\rangle \, | |g(x,y) - \boldsymbol{\mathcal{I}}_p^y g(x,y)| \varphi_i(x)\psi_j(y) ds_x ds_y \Big)^2 \\
&\leq \sum_{i=1}^{m}\sum_{j=1}^{k} \Big(\int_\Gamma \int_\Gamma |\, \langle x-y, n_y\rangle \, | \tilde{c}\big(1+\frac{2}{\pi}\log p\big)^d \big(\frac{\gamma\eta}{4}\big)^p |g(x,y)| \varphi_i(x)\psi_j(y) ds_x ds_y \Big)^2 \\
&= \tilde{c}^2\big(1+\frac{2}{\pi}\log p\big)^{2d}\big(\frac{\gamma\eta}{4}\big)^{2p} \sum_{i=1}^{m}\sum_{j=1}^{k} \Big(\int_\Gamma \int_\Gamma |\, \langle x-y, n_y\rangle \, g(x,y)| \varphi_i(x)\psi_j(y) ds_x ds_y \Big)^2 \\
&= \tilde{c}^2\big(1+\frac{2}{\pi}\log p\big)^{2d}\big(\frac{\gamma\eta}{4}\big)^{2p} \sum_{i=1}^{m}\sum_{j=1}^{k} \left|a_{ij}^V\right|^2 \\
&= \tilde{c}^2\big(1+\frac{2}{\pi}\log p\big)^{2d}\big(\frac{\gamma\eta}{4}\big)^{2p} \|A^V\|_F^2.
\end{aligned}
$$

□

Die Aussagen für den Nyström- und Kollokationsfall folgen analog. Wenn wir B^U und B^V aufgestellt haben, verfahren wir genau so wie beim Einfachschichtpotential. Da ihre Berechnungen in derselben Komplexitätsklasse liegen, können wir die Aufwandsabschätzungen aus [1] auf das Doppelschichtpotential erweitern.

4.3 Der Algorithmus und seine Komplexität

Wenn man die unterschiedlichen Varianten für das Erstellen der jeweiligen Matrix B bei der Implementierung unterscheidet, dient der folgende Algorithmus der Erzeugung der rekomprimierten Matrix X für alle drei Diskretisierungsverfahren, den Einfach-

sowie den Doppelschichtpotentialoperator und sowohl für die Matrix A^U als auch für A^V. Er ist hier mit der Bezeichnung $A = A^U$ formuliert. Damit er für $A = A^V$ gilt, müssen nur m durch n ersetzt und einige Matrizen komplex konjugiert werden. Die Größe k' der Matrix B nimmt bei der Matrix B^V im Fall des Doppelschichtpotentials den Wert $k' = (d+1)p^d$ und sonst $k' = p^d$ an.

Eingabe: $A \in \mathbb{C}^{m\times k}$, gewünschte Genauigkeit ε und k'

1: Matrix $B \in \mathbb{C}^{m\times k'}$ aufstellen
2: Singulärwertzerlegung durchführen $B = U_B \Sigma V_B^H$
3: Matrix $X^{ls} := V_B \Sigma^+ U_B^H A \in \mathbb{C}^{k'\times k}$ berechnen
4: QR-Zerlegung $\Pi B = QR$
5: finde das kleinste $0 < r_B < k'$, s. d. die $k' - r_B$ unteren Zeilen $\| \begin{bmatrix} 0 & R_{22} \end{bmatrix} X^{ls} \|_F \leq \varepsilon \|A\|_F$ erfüllen
6: berechne $Q^H A \in \mathbb{C}^{m\times k}$
7: finde das kleinste $0 < r < r_B$, s. d. die $k - r$ hinteren Spalten $\|Q_2^H A\|_F \leq 2\varepsilon \|A\|_F$ erfüllen
8: berechne $X \in \mathbb{C}^{r\times k}$, das $\hat{R}X = Q_1^H A$ löst

Ausgabe: Rang r, Permutation Π_r und Koeffizientenmatrix X

Algorithmus 4.1: RACA

In [1] wird des Weiteren beschrieben, wie die Matrix-Vektor-Multiplikation einer mit RACA komprimierten Matrix durch ein Verfahren, ähnlich dem Clenshaw-Algorithmus, durchgeführt werden kann, ohne dass man die Matrix B explizit berechnen muss. Für den Clenshaw-Algorithmus sei auf [13] und für die abgewandelte Form in zwei Dimensionen auf [1] verwiesen.
Insgesamt ergeben sich die Aufwandsabschätzungen in Tabelle 4.1 für einen zulässigen $(m \times n)$-Block A_{ts}.

Kompression	benötigter Speicher	Matrix-Vektor-Multiplikation	Berechnungszeit
keine	mn	mn	mn
ACA	$k(m+n)$	$k(m+n)$	$k^2(m+n)$
RACA	kk'	$k'(m+n+k)$	$(k^2+k'^2)(m+n)$

Tabelle 4.1: Vergleich der Komplexitätsklassen der Kompressionsverfahren

4.4 Numerische Ergebnisse

Für numerische Ergebnisse beim Einfachschichtpotential sei auf [1] verwiesen. Beim Doppelschichtpotential sind allgemein schlechtere Werte zu erwarten. Da die $\mathscr{H}$-Matrix nicht symmetrisch ist, müssen wir sie komplett speichern und die Ausgangsdimension von B ist um den Faktor $d+1$, also im $\mathbb{R}^3$ vier mal größer. Die Algorithmen wurden in C++ implementiert. Für elementare, algebraische Operationen haben wir Routinen aus den Fortranbibliotheken BLAS[1] und LAPACK[2] verwendet. Zum Speichern und Verwalten der Hierarchischen Matrizen wurde die Bibliothek AHMED[3] benutzt.
Alle Experimente wurden auf einem Rechner mit einer CPU aus zwei Intel Xeon Quad-Core X5482, einer Taktrate von 3.2 GHz und einem Arbeitsspeicher von 64 GB, durchgeführt. Die Algorithmen sind sehr effektiv parallelisierbar, siehe [7], die Zeitangaben sind allerdings für die sequentielle Implemetation angegeben.

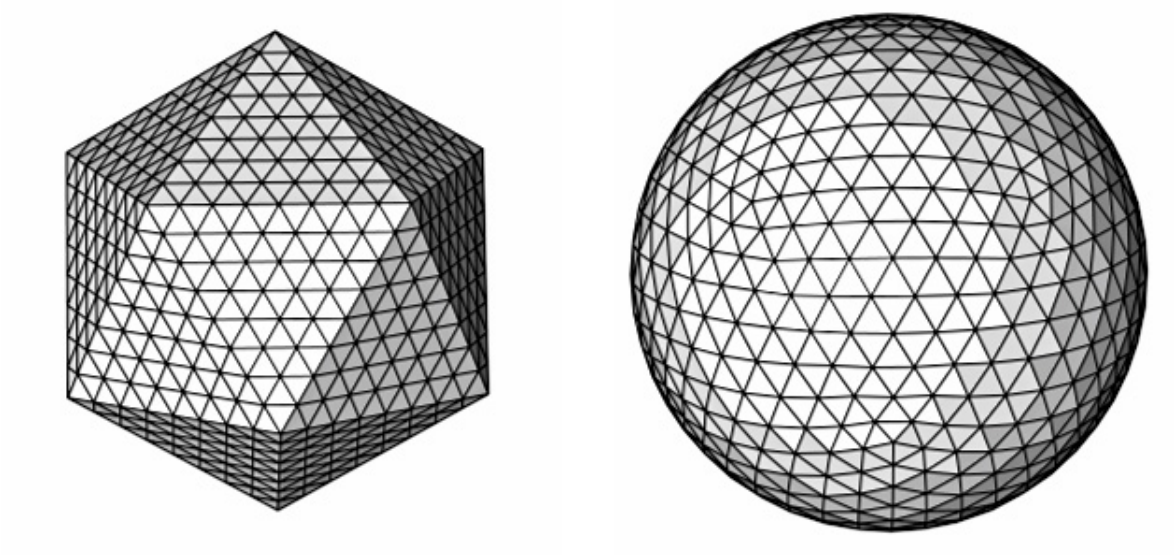

Abbildung 4.5: Erzeugung einer geodätischen Kuppel aus einem Ikosaeder mit $N = 1280$ Freiheitsgraden (Quelle: http://www.3d-meier.de/)

Die Werte in Tabelle 4.2 sind Ergebnisse der Approximation des Doppelschichtpotentials des Laplaceoperators durch ACA und RACA auf einer Kugel. Diese wird durch eine geodätische Kuppel über einem Ikosaeder approximiert. Dabei werden die 20 Dreiecksseiten des Ikosaeders in mehrere gleichseitige Dreiecke zerlegt und auf die Oberfläche der umfassenden Kugel projiziert, zur Veranschaulichung

[1] http://www.netlib.org/blas/
[2] http://www.netlib.org/lapack/
[3] http://bebendorf.ins.uni-bonn.de/AHMED.html

dient Abbildung 4.5. Hier ist eine geodätische Kuppel mit acht Dreiecken an jeder Kante einer ursprünglichen Dreiecksfläche dargestellt, was der kleinsten Auflösung mit $N = 1280$ Freiheitsgraden entspricht. Die höheren haben 16 und 32 Dreiecke pro Ikosaederkante und ähneln mehr und mehr der Kugel.

Tabelle 4.2 vergleicht den Speicherbedarf von ACA und RACA mit dem der unkomprimierten Matrix und ist wie folgt aufgebaut. Die Zeilen entsprechen den drei verschiedenen Auflösungen. In der ersten Spalte ist die Anzahl der Dreiecke in der Geometrie und damit die Zahl der Freiheitsgrade N angegeben. Die Spalten 2 und 5 enthalten den relativen Speicherbedarf in Bezug auf die komplette Matrix mit N^2 Einträgen in Prozent. In den Spalten 3 und 6 befindet sich der Speicheraufwand pro Freiheitsgrad in KB und schließlich in den Spalten 4 und 7 die Zeit zur Berechnung der Kompression in Sekunden.
Wir haben mit dem Kollokationsverfahren diskretisiert. Bei der Matrix A^V wurde die Gaußquadratur auf Dreiecken mit $g = 7$ Knoten aus Kapitel 1 verwendet. Für den Parameter aus der Fernfeldbedingung haben wir $\eta = 1.1$ gewählt, die Tschebyscheffpolynome sind vom Grad $p = 4$ und die Approximationsgenauigkeit liegt bei $\varepsilon = 0.001$.

	ACA			RACA		
N	rel. Speicher	Speicher/N	Zeit	rel. Speicher	Speicher/N	Zeit
1280	25.69%	2.57	0.31	19.21%	1.92	0.76
5120	8.98%	3.59	1.74	5.38%	2.15	4.56
20480	2.89%	4.63	8.72	1.47%	2.36	24.42

Tabelle 4.2: Ergebnisse der Kompressionsverfahren auf der Kugel

Die Zeit zur Berechnung steigt wie erwartet quadratisch an. Das ist jedoch akzeptabel, da die Approximation nur ein Mal durchgeführt werden muss.

In Abbildung 4.6 wurde der Speicherbedarf der unkomprimierten Matrix des durch ACA und des durch RACA erzeugten Approximanten am Beispiel der Kugel im Vergleich dargestellt.
Da der benötigte Speicher bei der unkomprimierten Matrix mit quadratisch und damit zu stark steigt und ACA schon sehr gute Ergebnisse liefert, haben wir zur Verdeutlichung der Speicherersparnis durch RACA eine logarithmierte Darstellung der Daten gewählt.

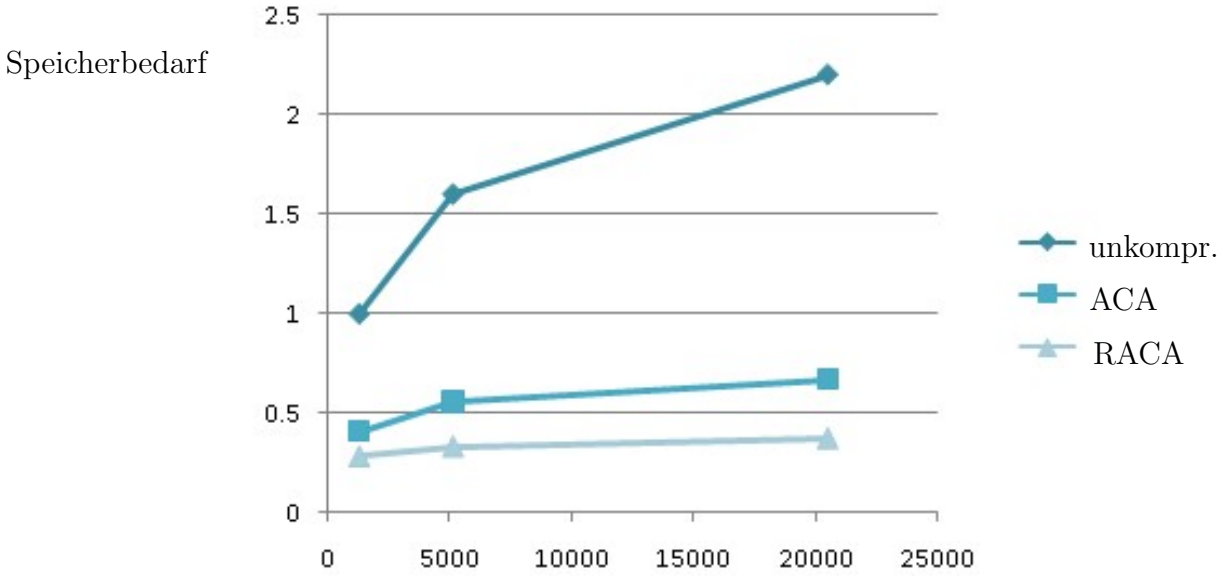

Abbildung 4.6: logarithmierter Speicherbedarf der Kompressionsalgorithmen auf der Kugel

Kapitel 5

Kompression der Steifigkeitsmatrix beim Helmholtzoperator

Ausgehend von der dreidimensionalen Wellengleichung aus der Akustik

$$-\Delta u + \frac{\partial^2 u}{\partial t^2} = 0,$$

erhalten wir durch Fouriertransformation nach t den zeitunabhängigen Helmholtzdifferentialoperator

$$\mathscr{L}u = -\Delta u - \kappa^2 u.$$

Dabei ist κ die Wellenzahl und $\Delta = \nabla^2$ der Laplaceoperator.
Die Singularitätenfunktion

$$S(x,y) = \frac{e^{i\kappa\|x-y\|}}{\|x-y\|} \tag{5.1}$$

des Helmholtzoperators ist, da er ein elliptischer Differentialoperator der Form (2.4) ist, asymptotisch glatt bezüglich y, aus Symmetriegründen natürlich auch bezüglich x. Das heißt, es existieren Konstanten c und γ, so dass für alle Multiindizes α, alle $x \in \mathbb{R}^n$ und alle $y \in \mathbb{R}^n \setminus \{x\}$,

$$|\partial_y^\alpha S(x,y)| \leq c|\alpha|!\gamma^{|\alpha|}\frac{|S(x,y)|}{\|x-y\|^{|\alpha|}} \tag{5.2}$$

gilt, vergleiche Definition 2.1.6. Daraus folgt, dass sie lokal durch eine degenerierte Kernfunktion approximiert und somit blockweise als Niedrigrangmatrix gespeichert werden kann.

Allerdings ist γ linear von der Frequenz, also von der Wellenzahl κ abhängig, was zur Folge hat, dass die Abschätzungen für hohe Frequenzen wertlos werden. Die Singularitätenfunktion oszilliert zu stark. Es kann zwar ein degenerierter Kern gefunden werden, der $S(x, y)$ approximiert, doch der Grad der Degeneriertheit wächst mit steigender Frequenz. Dies überträgt sich direkt auf die Anzahl k der Spalten von Uund V, die letztendlich gespeichert werden müssen. Für Details siehe [7] und [14].

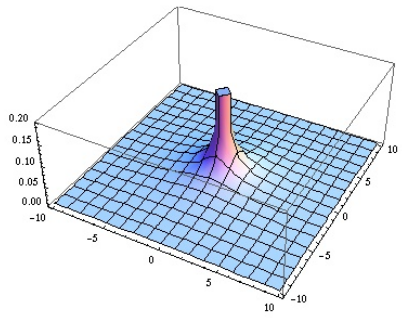

Abbildung 5.1: Singularitätenfunktion des Laplaceoperators

In Abbildung 5.1 ist zum Vergleich die Singularitätenfunktion des Laplaceoperators

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

für $x = 0, y \in \mathbb{R}^3, y_1, y_2 \in [-10, 10], y_3 = 0$ dargestellt.

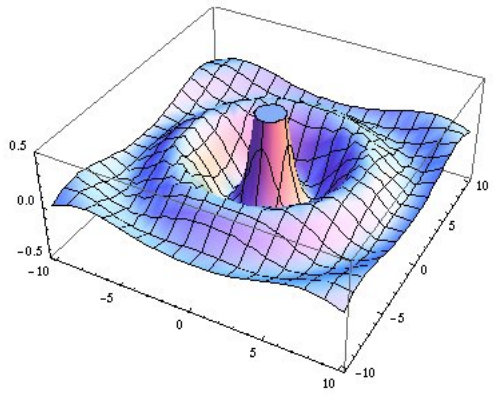

Abbildung 5.2: Singularitätenfunktion des Helmholtzoperators mit $\kappa = 1$

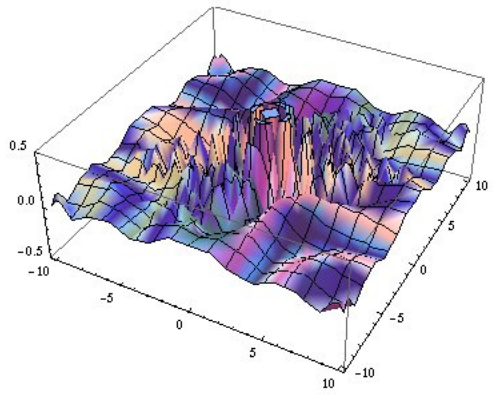

Abbildung 5.3: Singularitätenfunktion des Helmholtzoperators mit $\kappa = 5$

Hier kann man sehr gut erkennen, warum die Zerlegung des Raumes mit Hilfe der

Fernfeldbedingung zu approximierbaren Blöcken führt. Die Matrixeinträge in einem zulässigen Block unterscheiden sich kaum voneinander, da $S(x, y)$ nur in der Nähe der Singularität $x = y$ hohe Werte annimmt und sonst sehr glatt verläuft.

Die Singularitätenfunktion des Helmholtzoperators (5.1) verhält sich völlig anders. In Abbildung 5.2 wurde ihr Realteil im gleichen Bereich für $\kappa = 1$ und in Abbildung 5.3 für $\kappa = 5$ dargestellt. Man kann sich gut vorstellen, dass eine derartig oszillierende Funktion nur sehr schwer approximierbar ist. Wenn man aber reale Problemstellungen betrachten möchte, handelt es sich im Allgemeinen um noch viel höhere Frequenzen, zum Beispiel im hörbaren Bereich von 20 bis 20000 Hz in der Akustik.

5.1 Einschränkung der Zulässigkeit

Bisher gibt es noch keine zufriedenstellende Lösung für das Problem der Approximation des Helmholtzoperators im Hochfrequenzbereich. Es gibt jedoch die Möglichkeit, die Singularitätenfunktion teilweise zu glätten. Dies erlaubt es uns, $S(x, y)$ immerhin in einem kleinen Gebiet annähernd zu beschreiben. Da wir die zu Grunde liegende Geometrie sowieso in zulässige Bereiche zerlegen wollten, bereitet es auch nicht zu große Umstände. Diese Bereiche sehen jetzt nur etwas anders aus.

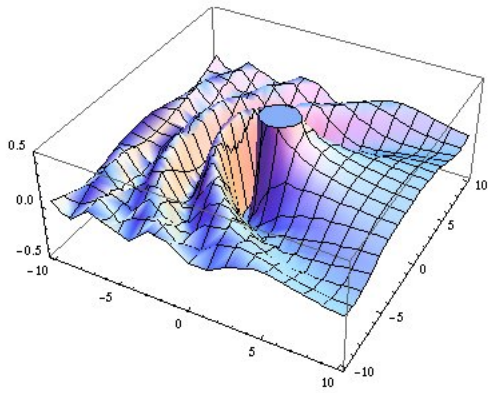

Abbildung 5.4: Singularitätenfunktion des Helmholtzoperators mit $\kappa = 1$ entlang der x-Achse geglättet

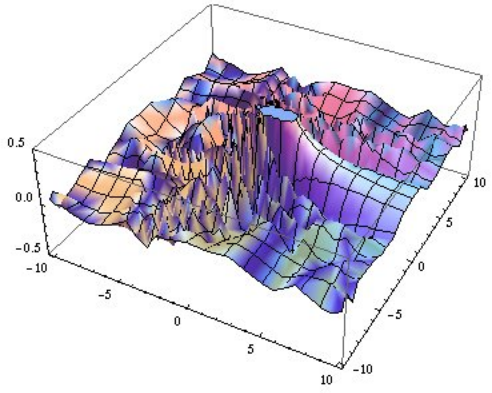

Abbildung 5.5: Singularitätenfunktion des Helmholtzoperators mit $\kappa = 5$ entlang der x-Achse geglättet

Abbildung 5.4 zeigt den Realteil der mit

$$e^{i\kappa\langle x-y,e_1\rangle} = e^{i\kappa(x-y)(1,0,0)^T} = e^{i\kappa(x_1-y_1)}$$

multiplizierten Singularitätenfunktion für $x, y \in \mathbb{R}^3, x = (0,0,0)^T, y_1, y_2 \in [-10,10], y_3 = 0$ mit $\kappa = 1$ und Abbildung 5.5 mit $\kappa = 5$.

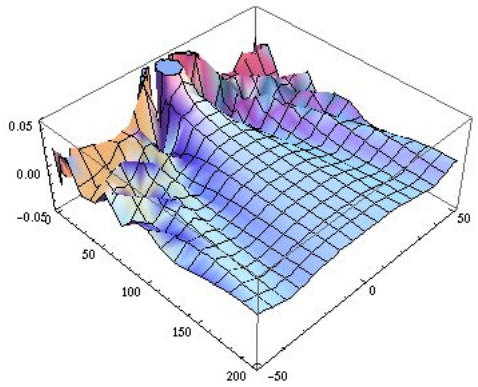

Abbildung 5.6: Singularitätenfunktion des Helmholtzoperators mit $\kappa = 1$ entlang der x-Achse geglättet

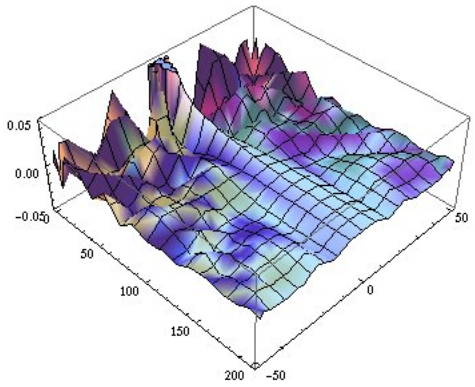

Abbildung 5.7: Singularitätenfunktion des Helmholtzoperators mit $\kappa = 5$ entlang der x-Achse geglättet

In Richtung der Glättung, hier also $(1,0,0)^T$, entsteht ein nur leicht gewölbter Rücken, der gut approximiert werden kann. Die Abbildungen 5.6 und 5.7 verdeutlichen dies. Hier wurde $x = 0, y \in \mathbb{R}^3, y_1 \in [0,200], y_2 \in [-50,50], y_3 = 0$ betrachtet. Alle übrigen Bereiche bleiben stark oszillierend. Es fällt auf, dass dieser glatte Rücken schmaler wird, wenn κ wächst.

Wir betrachten für die Partitionierung die Hochfrequenzfernfeldbedingung

$$\kappa \min\{(\operatorname{diam}(X_t))^2, (\operatorname{diam}(Y_s))^2\} \leq \beta \operatorname{dist}(X_t, Y_s) \tag{5.3}$$

zusammen mit der Winkelbedingung

$$\exists e \in S^{d-1} \forall x \in X_t, y \in Y_s : \sin \measuredangle(y - x, e) < \frac{1}{c\kappa \min\{\operatorname{diam}(X_t), \operatorname{diam}(Y_s)\}}. \tag{5.4}$$

Hier sind S^{d-1} die Einheitssphäre und $c, \beta > 0$ Parameter.
Wir werden sehen, dass die Bedingungen (5.3) und (5.4) in den Blöcken, die sie erfüllen, die Möglichkeit garantieren, die Singularitätenfunktion unabhängig von der Frequenz zu glätten. Die Überprüfung ist allerdings zu komplex. Wir arbeiten deshalb mit den Zulässigkeitsbedingungen

$$\kappa r_t^2 \leq \beta \operatorname{dist}(m_t, Y_s) \tag{5.5}$$

und

$$\exists e \in S^{d-1} \forall y \in Y_s : \sin \sphericalangle(y - m_t, e) < \frac{1}{c\kappa r_t}, \tag{5.6}$$

die mit $\mathcal{O}(|t| + |s|)$ Operationen prüfbar sind. Dabei sind r_t der Radius und m_t der Mittelpunkt des Clusters X_t.
Die Beschränkung des maximalen Winkels zwischen den Punkten aus X_t und Y_s geht mit der Idee konform, dass kreisförmige Wellen auf einem schmalen Winkel durch ebene Wellen angenähert werden können, siehe [15].

Lemma 5.1.1. *Sei $Y_s \subset \mathbb{R}^3$ ein Cluster, das die Zulässigkeitsbedingungen (5.5) und (5.6) mit einem Cluster X_t erfüllt. Weiter existiere eine Konstante $\beta' > 0$ mit $\beta' \operatorname{dist}(X_t, Y_s) \leq \kappa(\operatorname{diam}(X_t))^2$ und es gelte $\kappa \operatorname{diam}(X_t) \geq \frac{4}{3}\beta + \frac{1}{3}\sqrt{3 + (4\beta + 3)^2}$ sowie $\langle y - x, e\rangle > 0, \forall x \in X_t, y \in Y_s$. Dann gibt es eine Zerlegung*

$$\|x - y\| = \langle y - x, e\rangle + g(x, y)$$

für $x \in X_t, y \in Y_s$, so dass $\kappa|g|$ bezüglich κ beschränkt ist.

Der Beweis und weitere Informationen sind in [14] zu finden. Lemma 5.1.1 verallgemeinert die Beobachtungen aus den obigen Abbildungen auf beliebige Richtungen e sowie Frequenzen κ und garantiert die Existenz einer Zerlegung

$$S(x, y) = \frac{e^{i\kappa\|x-y\|}}{\|x - y\|} = \frac{e^{i\kappa(\langle y-x,e\rangle + g(x,y))}}{\|x - y\|} = e^{i\kappa\langle y-x,e\rangle}\hat{u}(x, y) \tag{5.7}$$

mit

$$\hat{u}(x, y) = \frac{e^{i\kappa g(x,y)}}{\|x - y\|}.$$

Da $\kappa|g|$ bezüglich κ beschränkt ist, stellt $\hat{u}(x, y)$ eine asymptotisch glatte Funktion auf $X_t \times Y_s$ dar, wobei die Konstanten in (5.2) gänzlich von κ unabhängig sind. Der Grad der Degeneriertheit des Approximanten an die Kernfunktion und damit der nötige Rang k des zu speichernden Blocks wird somit ebenfalls nicht von der Frequenz beeinflusst.

Das genaue Aussehen von $g(x, y)$ sowie seine Berechnung sind kompliziert. Wir können $\hat{u}(x, y)$ jedoch einfach ohne g aus dem Zusammenhang (5.7) gewinnen:

$$\hat{u}(x, y) = \frac{S(x, y)}{e^{i\kappa\langle y-x,e\rangle}} = \frac{e^{i\kappa(\|x-y\| - \langle y-x,e\rangle)}}{\|x - y\|} = \frac{e^{i\kappa(\|x-y\| + \langle x-y,e\rangle)}}{\|x - y\|}.$$

Wir werden also nicht $S(x,y)$, sondern ausschließlich $\hat{u}(x,y) = S(x,y)e^{i\kappa\langle x-y,e\rangle}$ interpolieren, da es in den zulässigen Bereichen nicht oszilliert und somit approximierbar ist. Wann immer Funktionsauswertungen der tatsächlichen Kernfunktion $S(x,y)$ gebraucht werden, muss das Interpolationspolynom zu $\hat{u}(x,y)$ ausgewertet und das Ergebnis anschließend mit $e^{i\kappa\langle y-x,e\rangle}$ multipliziert werden.

Mit der Zerlegung der Singularitätenfunktion (5.7) haben wir die Approximierbarkeit des Einfachschichtpotentials sichergestellt. Analog lässt sich das Doppelschichtpotential

$$(\mathcal{K}u)(x) = \int_\Gamma \frac{\langle x-y, n_y\rangle\, e^{i\kappa|x-y|}(1-i\kappa\,|x-y|)}{|x-y|^3} u(y)\,\mathrm{d}s_y$$

stückweise durch

$$\begin{aligned}\frac{e^{i\kappa|x-y|}(1-i\kappa\,|x-y|)}{|x-y|^3} &= \frac{e^{i\kappa(\langle y-x,e\rangle+g(x,y))}(1-i\kappa\,|x-y|)}{|x-y|^3} \\ &= e^{i\kappa\langle y-x,e\rangle}\frac{e^{i\kappa g(x,y)}(1-i\kappa\,|x-y|)}{|x-y|^3}\end{aligned} \tag{5.8}$$

glätten.

5.2 Partitionierung

Wie wir gesehen haben, kann ein Block $b = t \times s$ der Steifigkeitsmatrix eines elliptischen Differentialoperators im Hochfrequenzfall zulässig genannt werden, wenn die Cluster X_t und Y_s sowohl die Hochfrequenzfernfeld- (5.5) als auch die Winkelbedingung (5.6) erfüllen.

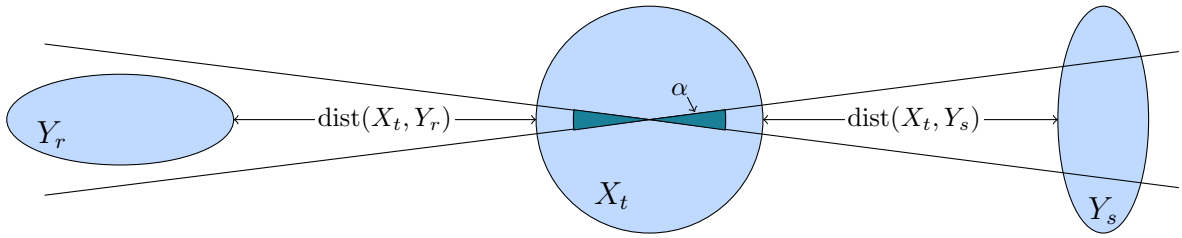

Abbildung 5.8: Das Cluster Y_r erfüllt die Winkelbedingung mit X_t, Y_s dagegen nicht.

Ein Beispiel für zulässige und nicht zulässige Cluster wurde in Abbildung 5.8 dargestellt. Dabei soll für die dunkelblau gezeichneten Winkel α gelten: $\sin(\alpha) = (c\kappa r_t)^{-1}$.

Leider hat die Aufteilung nach dem Winkel zur Folge, dass sehr viel mehr Blockcluster entstehen, als bei der einfachen Fernfeldbedingung, so dass wir über einen Zerlegungsalgorithmus nachdenken müssen, der die Anzahl der Blöcke minimiert. Ein solcher Algorithmus wurde in (3.5) vorgestellt, er sucht absteigend im Clusterbaum zu jedem Cluster die größte Menge zulässiger Dreiecke und fasst sie zu einem Blockcluster zusammen. Dadurch erhalten wir im Allgemeinen keine zusammenhängenden, allerdings minimal viele Blöcke.

Dieser Algorithmus kann Grundlage für den sein, der unser Problem bearbeitet, wenn man den zur Fernfeldbedingung (5.5) zulässigen Bereich weiter in Teile zerlegt, die jeweils die Winkelbedingung (5.6) erfüllen.

Die rechte Seite der Winkelbedingung wird, da κ im Nenner steht, im Hochfrequenzfall sehr klein werden. Dies bedeutet, dass auch eine sehr feine Partition notwendig sein wird, was zwangsläufig zu einer riesigen Zahl von Blöcken führt.
Hinzu kommt, dass man im Gegensatz zur reinen Fernfeld- (3.3) bei der Winkelbedingung (5.6) nicht von einer zulässigen Indexmenge $z(t)$ sprechen kann, da es im Allgemeinen zu verschiedenen Richtungen $e \in S^{d-1}$ auch verschiedene zulässige Cluster gibt. Auf der rechten Seite in Abbildung 5.9 sind sieben zulässige Bereiche erkennbar. Abhängig von maximaler Winkelgröße und Dimension können mehrere Millionen winkelzulässige Cluster auf ein fernfeldzulässiges kommen.

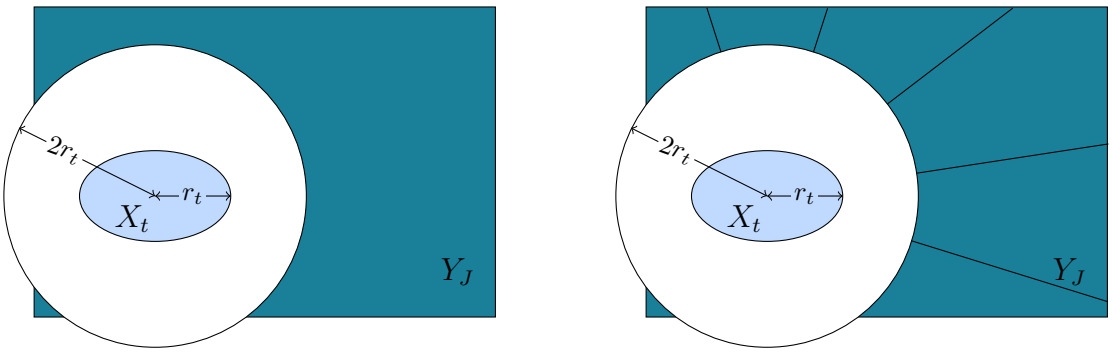

Abbildung 5.9: Beispiel für maximale zulässige Bereiche $z(t)$, dunkelgrau gekennzeichnet, links mit der Fernfeld- und rechts mit der Winkelbedingung, $\eta = 1$

Des Weiteren kann man in den Abbildungen 5.6, 5.7 und 5.9 erkennen, dass die optimale Form, die ein Cluster in einem möglichen zweiten Clusterbaum zur Indexmenge J haben sollte, ein schmaler, langer, $d-$dimensionaler Kegelstumpf ist. Wie schmal und in welche Richtung orientiert er letztendlich sein muss, um tatsächlich zusammen mit einer Indexmenge t die Zulässigkeitsbedingung zu erfüllen, hängt vom Cluster X_t, dessen Lage und dessen Größe ab. Insbesondere sind diese Eigenschaften in Betrachtung verschiedener $t_1, t_2 \subset I$ völlig unterschiedlich und stehen einander teilweise sogar diametral gegenüber. Wir könnten mit der ersten Partitionierungsvariante (3.4) aus Kapitel 3 die Informationen über die Form der Cluster Y_s also nicht nutzen und uns bliebe nichts anderes übrig, als viele kleine Cluster zu erzeugen.

Aufgrund der zahlreichen Vorteile entscheiden wir uns für die zweite Art der Blockclusterbaumerzeugung (3.5), beziehungsweise für eine davon leicht abgewandelte Form.
Die Zusammensetzung der einzelnen Cluster ist nicht eindeutig festgelegt. Es bietet sich daher aus Komplexitätsgründen an, den Raum um das gegebene X_t vorab in geeignete, möglicherweise unendlich lange Kegel, beziehungsweise Pyramiden zu zerlegen, so dass alle Träger der Testfunktionen, die in einem dieser Bereiche liegen, automatisch die reine Winkelbedingung (5.6) zusammen mit X_t erfüllen. Das jeweilige $e \in S^{d-1}$ entspricht damit den Höhen der Pyramiden. So kann jeder Index in J, der die Fernfeldbedingung (5.5) erfüllt, ganz einfach seinem nach der Winkelbedingung maximal zulässigen Cluster zugeordnet werden.
Im Allgemeinen bleiben viele der Bereiche leer, dennoch ist die Anzahl der zulässigen Cluster nicht zu unterschätzen. Im 2D-Fall gibt es zu jedem Knoten $t \in T_I$ bis zu

$$c_2 \kappa r_t$$

und im Dreidimensionalen sogar

$$c_3 \kappa^2 r_t^2 \tag{5.9}$$

solcher Pyramiden, wobei $c_2, c_3 \in \mathbb{R}$ von c abhängige Konstanten sind. Wir betrachten den 3D-Fall genauer. Man stelle sich im $\mathbb{R}^3$ vor, wie in Kugelkoordinaten der Raum entlang φ und θ in gleich große Teile zerlegt wird und jede Pyramide aus dem kartesischen Produkt je eines φ- und eines θ- Teils entstehen, r beliebig. Ihre enorme Zahl im Hochfrequenzbereich unterstreicht noch einmal die Notwendigkeit der zweiten Partitionierungsart (3.5).
Um mehrere zulässige Bereiche berücksichtigen zu können, wandeln wir sie allerdings etwas ab. Seien $e_i \in S^{d-1}, i = 1, ..., w$, die Höhen der vorab gewählten Pyramiden, $w = \lceil c_3 \kappa^2 r_t^2 \rceil$ deren Anzahl,

$$z_i(t) = \{j \in J : t \times j \text{ zulässig nach (5.5) und (5.6) mit } e = e_i\}$$

die Menge der Indizes aus dem Fernfeld, die zusammen mit e_i keinen größeren Winkel als $\arcsin((c\kappa r_t)^{-1})$ einschließen. Weiter seien $t_1, ..., t_n \in S(T)$ die Söhne und $n(t) := s \setminus \bigcup_{i=1}^{w} z_i(t)$ das Nahfeld von t, also die zu t nicht zulässigen Indizes, dann hat die Zerlegung folgende Form:

$$S_{I\times J}(t \times s) = \begin{cases} \emptyset, \text{ für } t \times s \text{ zulässig} \vee |s| < n_{min}, \\ \{t \times (z_1(t) \cap s)), ..., t \times (z_w(t) \cap s)), t \times n(t)\}, \text{ für } S(t) = \emptyset \vee |n(t)| < n_{min}, \\ \{t \times (z_1(t) \cap s)), ..., t \times (z_w(t) \cap s)), t_1 \times n(t), ..., t_n \times n(t)\}, \text{ sonst.} \end{cases}$$

Wir nennen sie Hochfrequenzpartitionierung.

Die Beweise zur Zeitkomplexitätsabschätzung, siehe Tabelle 3.1, aus [7] lassen sich analog auf die neue Variante der Blockclusterbaumkonstruktion übertragen. Wir benötigen $\mathcal{O}(n \log n)$ Operationen für die Berechnung der Partitionierung. Leider nimmt $c_{sp}(T_I)$ hier völlig andere Ausmaße an. Wenn wir nicht nur die Abhängigkeit von n sondern auch von κ betrachten, ist $c_{sp}(T_I, \kappa)$ nicht einmal mehr eine Konstante. Aus (5.9) folgt für den 3D-Fall, dass es zu jeder Indexmenge enorm viele zulässige und wieder einen nicht zulässigen Teil geben kann:

$$c_{sp}(T_I, \kappa) = \max_{t \in T_I} |\{s \in J : t \times s \in T_{I\times J}| \leq \max_{t \in T_I} c_3 \kappa^2 r_t^2 + 1$$

und das liefert die folgende Speicherkomplexität.

Satz 5.2.1. *Sei $T_{I\times J}$ ein Blockclusterbaum, der die Zerlegung der Oberfläche einer dreidimensionalen Geometrie nach der Hochfrequenzpartitionierung beinhaltet. Für den zu Grunde liegenden Clusterbaum T_I existiere eine Konstante C die für alle $t_1, t_2 \in T_I^l$ einer Ebene*

$$r_{t_1} \leq C r_{t_2}$$

erfüllt. Dann ist die Anzahl der Blätter $|\mathscr{L}(T_{I\times J})|$ dieses Blockclusterbaums von der Ordnung $\mathcal{O}(n \log n)$.

Beweis: Die zweidimensionale Oberfläche der 3D-Geometrie hat eine feste Größe g. Wenn man sie mit Kugeln vom Radius r überdecken möchte, benötigt man

$$\frac{g}{r^2}$$

viele. Zu jedem Cluster mit Radius r aus dem Clusterbaum T_I können bis zu

$$c_3 \kappa^2 r^2$$

zulässige Bereiche gefunden werden. Sei r^l_{min} der kleinste und r^l_{max} der größte Radius, den ein Cluster in der l-ten Ebene T^l_I des Clusterbaums hat. Dann gilt nach Voraussetzung

$$r^l_{max} \leq C r^l_{min}.$$

Darum gibt es in einer gegebenen Ebene l maximal

$$\frac{g}{(r^l_{min})^2}$$

Cluster und zu jedem Cluster aus dieser Ebene nicht mehr als

$$c_3(\kappa r^l_{max})^2$$

zulässige Bereiche. Dies ergibt für jede der $\mathcal{O}(\log n)$ Ebenen von T_I höchstens

$$\frac{g c_3(\kappa r^l_{max})^2}{(r^l_{min})^2} \leq \frac{g c_3(\kappa C r^l_{min})^2}{(r^l_{min})^2} = g c_3(\kappa C)^2$$

Blätter im Blockclusterbaum und schließlich $\mathcal{O}(\kappa^2 \log n)$ viele für T_I insgesamt.
Für die Randelementmethode muss $\kappa h \sim 1$ gelten. Die Oberfläche der Geometrie wurde beim Triangulieren in n Dreiecke mit Flächeninhalt der Größenordnung $\mathcal{O}(h^2)$ zerlegt, deshalb gilt weiter $nh^2 \sim g$. Daraus folgt $k^2h^2 \in \mathcal{O}(1)$, $nh^2 \in \mathcal{O}(1)$ und somit $k^2 \sim n$. Die Anzahl der Blöcke beträgt also

$$\mathcal{O}(\kappa^2 \log n) = \mathcal{O}(n \log n).$$

□

5.3 Berechnung der Einträge

5.3.1 Kompression mit ACA

Das Verfahren der Adaptive Cross Approximation kann wie gewohnt, auf die bereits partitionierte Geometrie angewendet werden. Da die asymptotische Glattheit von $\hat{u}$ unabhängig von κ auf den zulässigen Blöcken garantiert ist, liefert es gute Approximanten an die tatsächliche Matrix. Es sei an Kapitel 3 erinnert.

Für die neue Partitionierungsvariante und ACA gilt weiterhin die Abschätzung aus Satz 3.3.2, dass die Matrix-Vektor-Multiplikation in $\mathcal{O}(n \log n)$ Schritten durchführbar ist.

Die Berechnung der Einträge eines Blocks mit ACA benötigt die Auswertung der geglätteten Kernfunktion für k Zeilen und Spalten also $\mathcal{O}(k(|t| + |s|))$. Für jeden Block müssen letztlich auch $k(|t| + |s|)$ Einträge gespeichert werden. Was schließlich, da $|t|, |s|$ linear von $|I| = n$ abhängen, zusammen mit Satz 5.2.1 zu einer Zeit- und Speicherkomplexitätsklasse von $\mathcal{O}(n^2 \log n)$ führt.

Zeitlich ist der quadratische Aufwand verschmerzbar, da die Approximation nur ein Mal durchgeführt werden muss. Aber für den Speicher ist das inakzeptabel. Wenn wir für jeden Block nur eine konstante Anzahl von Einträgen speichern müssten, hätten wir das gewünschte $\mathcal{O}(n \log n)$. Um dies zu erreichen, gibt es verschiedene Möglichkeiten. Es sei auf [14] für eine Variante mit ACA auf $\mathscr{H}^2$-Matrizen verwiesen. Wir betrachten die Approximation durch gemeinsame Basen am Beispiel der Tschebyscheffpolynome.

5.3.2 Approximation durch gemeinsame Basen

Nach Zerlegung der Kernfunktion des Helmholtzoperators müssen wir nur noch die Funktion $\hat{u}$ approximieren. Auf den zulässigen Bereichen ist sie asymptotisch glatt bezüglich beider Variablen mit Konstanten unabhängig von der Frequenz κ. Dort kann sie also interpoliert werden. Wegen ihrer guten numerischen Eigenschaften benutzen wir auch hier Tschebyscheffpolynome. Es sei jedoch darauf hingewiesen, dass jede andere Basis an ihrer Stelle denkbar wäre.
Wir betrachten stellvertretend den per Nyströmverfahren diskretisierten Einfachschichtpotentialoperator und müssen demnach nur einen Approximanten für die Matrix $\hat{U} \in \mathbb{C}^{m \times n}$,

$$\hat{u}_{ij} = \frac{e^{i\kappa g(x_i, y_j)}}{\|x_i - y_j\|},$$

finden. Alle Ergebnisse lassen sich jedoch analog auf die anderen Diskretisierungsvarianten und das wie in (5.8) geglättete Doppelschichtpotential übertragen.

In Kapitel 3 wurde die Interpolation zum Erzeugen degenerierter Kerne vorgestellt. Es war ausreichend entlang einer Variable zu interpolieren um die Degeneriertheit zu garantieren. Damit wir nur eine konstante Anzahl an Einträgen speichern müssen, werden wir jetzt bezüglich x und y interpolieren.
Mit den Bezeichnungen der Tschebyscheffinterpolation auf zwei Variablen aus Kapitel

1 ist

$$\boldsymbol{\mathcal{I}}_p\hat{u}(x,y) = \boldsymbol{\mathcal{I}}_p^x\boldsymbol{\mathcal{I}}_p^y\hat{u}(x,y) = \sum_{\boldsymbol{\alpha},\boldsymbol{\beta}\in\mathbb{N}^d,\alpha_\nu,\beta_\nu<p} c_{\boldsymbol{\alpha}\boldsymbol{\beta}} \prod_{\nu=1}^{d} T_{\alpha_\nu}(\xi_\nu)\cdot T_{\beta_\nu}(\epsilon_\nu),$$

,

mit

$$c_{\boldsymbol{\alpha}\boldsymbol{\beta}} = \prod_{\nu=1}^{d} \frac{(2-\delta(\alpha_\nu))(2-\delta(\beta_\nu))}{p^2} \sum_{\boldsymbol{k},\boldsymbol{l}\in\mathbb{N}^d,k_\nu,l_\nu<p} \hat{u}(\mathring{x}_{\boldsymbol{k}},\mathring{y}_{\boldsymbol{l}}) \prod_{\nu=1}^{d} \cos\left(\frac{2k_\nu+1}{2p}\alpha_\nu\pi\right)\cos\left(\frac{2l_\nu+1}{2p}\beta_\nu\pi\right).$$

Die p^{2d} Koeffizienten können als Koeffizientenmatrix $C \in \mathbb{C}^{p^d\times p^d}$ interpretiert werden. Wir erhalten $\hat{U} \approx \tilde{U} = B^x C B^y$ mit $B^x \in \mathbb{C}^{n\times p^d}$, $B^y \in \mathbb{C}^{p^d\times m}$,

$$\begin{aligned} b^x_{i\boldsymbol{\alpha}} &= \prod_{\nu=1}^{d} T_{\alpha_\nu}(\xi_{i_\nu}), \\ b^y_{\boldsymbol{\beta}j} &= \prod_{\nu=1}^{d} T_{\beta_\nu}(\epsilon_{j_\nu}). \end{aligned}$$

Es genügt die Koeffizienten $C \in \mathbb{C}^{p^d\times p^d}$ des Polynoms zur gegebenen Basis zu kennen, um es eindeutig zu beschreiben. Dies bedeutet, dass für jeden zulässigen Block nur konstant viele Einträge gespeichert werden müssen. Nach Satz 5.2.1 folgt, dass der Speicheraufwand für die ganze Matrix in der von uns angestrebten Komplexitätsklasse $\mathcal{O}(n\log n)$ liegt.

Die Matrizen B^x und B^y können mit Hilfe der Rekursionsrelation (1.2) sehr schnell aufgestellt werden und müssen demnach auch nicht gespeichert werden, um die Matrix-Vektor-Multiplikation zu ermöglichen. Ihre Berechnung erfolgt analog zu denen in Kapitel 4 in $\mathcal{O}(np^d)$ Schritten, was die Komplexität von $\mathcal{O}(n\log n)$ diser algebraischen Operation nicht verändert.

Für die Einträge eines Blocks benötigen wir jeweils $\mathcal{O}(|t|+|s|)$ Operationen zur Bestimmung der Glättungsrichtung e, der transformierten Koordinaten und der Intervallgrenzen. Außerdem müssen wir p^{2d} Koeffizienten berechnen, von denen jeder einzelne mit d^2p^{2d} Schritten ermittelt werden kann. Da der Polynomgrad p und die Dimension d als konstant betrachtet werden und $|t|, |s|$ linear von n abhängen, ergeben die $\mathcal{O}((|t|+|s|)d^2p^{4d}) = \mathcal{O}(n)$ Operationen pro Block zusammen mit Satz 5.2.1 eine Gesamtzeitkomplexität in der Klasse von $\mathcal{O}(n^2\log n)$.

Die Fehlerabschätzung für die Tschebyscheffpolynominterpolation asymptotisch glatter Funktionen aus Satz 3.4.2 benötigte ein Gebiet, welches ein kartesisches Produkt von

Intervallen ist. Damit die Pyramiden, in welche die Geometrie zerlegt werden muss, diese Bedingung erfüllen, ist eine Transformation auf Kugelkoordinaten sinnvoll.

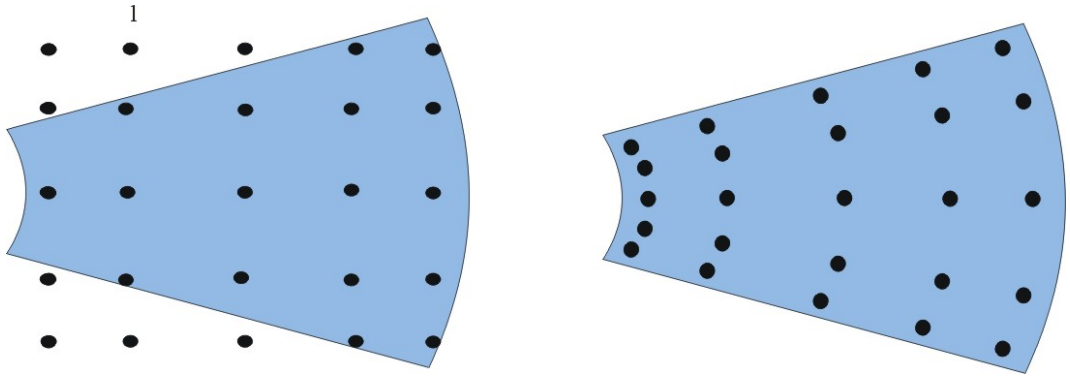

Abbildung 5.10: Beispiel für die Verteilung der Tensortschebyscheffknoten auf einem zulässigen Bereich, links in kartesischen und rechts in Polarkoordinaten

In Abbildung 5.10 links ist die Verteilung der zweidimensionalen Tensortschebyscheffknoten auf einem typischen hochfrequenzfernfeld- und winkelzulässigen Gebiet dargestellt.
Es ist offensichtlich, dass hier keine Fehlerabschätzung halten kann, da die Knoten teilweise in ungeglätteten, stark oszillierenden Bereichen liegen, in denen die Kernfunktion unvorhersehbare Werte annimmt. Auf der rechten Seite von Abbildung 5.10 ist die Anordnung der Knoten in Polarkoordinaten dargestellt. Sie befinden sich nur im glatten Bereich.

5.3.3 Numerische Ergebnisse

Die folgenden Werte sind Ergebnis numerischer Experimente auf demselben Rechner und der selben triangulierten Kugel wie in Kapitel 4. Wir haben wieder drei verschiedene Auflösungen. Tabelle 5.1 enthält die Ergebnisse der Approximation mit ACA und ist wie folgt aufgebaut. In der ersten Spalte ist die Anzahl der Freiheitsgrade N angegeben. Spalte 2 enthält den relativen Speicherbedarf in Bezug auf die komplette Matrix mit N^2 Einträgen in Prozent. In Spalte 3 befindet sich der Speicheraufwand pro Freiheitsgrad in KB und schließlich in Spalte 4 die Zeit zur Berechnung der Kompression in Sekunden. Hierzu sei angemerkt, dass die Implementierung nicht zeitoptimiert ist.

Wir haben das Einfachschichtpotential des Helmholtzoperators mit dem Nyströmverfahren diskretisiert. Der Parameter aus der Fernfeldbedingung beträgt $\beta = 1.1$, der aus der Winkelbedingung $c = 2$ und die Approximationsgenauigkeit $\varepsilon = 0.001$. Um bei der Randelementmethode Konvergenz zu garantieren muss $\kappa h \lesssim 1$ gelten, wobei h die Größe der Dreiecke ist. Bei den Experimenten in Tabelle 5.1 wurde $\kappa h = 0.3$ gewählt.

N	κ	rel. Speicher	Speicher/N	Zeit
1280	3.15%	99.95	19.98	0.64
5120	6.28%	77.93	62.34	5.21
20480	12.55%	41.17	131.75	37.37

Tabelle 5.1: Ergebnisse von ACA auf der Kugel

Die erzielten Ergebnisse genügen wie erwartet nicht unseren Forderungen. Zwar steigt der Speicherbedarf nicht so stark wie bei der unkomprimierten Matrix mit der Anzahl der Freiheitsgrade, doch offensichtlich auch nicht quasilinear.

Die Triangulationen der Kugel sowie die Blockclusterbäume für die $\mathscr{H}$-Matrizen aus Tabelle 5.2 stimmen identisch mit denen aus Tabelle 5.1 überein. Lediglich bei der Bestimmung der Einträge wurde statt mit ACA mit Tschebyscheffbasen gearbeitet. In der ersten Spalte ist die Anzahl der Freiheitsgrade in den drei verschiedenen Auflösungen und in der zweiten die Frequenz, also κ gegeben. Es gilt auch hier $\kappa h = 0.3$. Anschließend folgen drei gleich aufgebaute Tabellenblöcke, einer für jeden der Polynomgrade eins, zwei und drei.
Die erste Spalte jedes Blocks enthält den relativen Speicherbedarf im Verhältnis zu dem der unkomprimierten Matrix in Prozent. Der Speicher pro Freiheitsgrad in KB befindet sich jeweils in der zweiten. In Spalte drei jedes Blocks ist der durchschnittliche, relative Fehler an den Dreiecksmittelpunkten, die bei unserer Wahl der Diskretisierung sowohl Kollokationspunkte als auch Stützstellen für die Quadraturformel beim Nyströmverfahren darstellen, aufgeführt. Jeder Block endet mit der Angabe der Zeit in Sekunden, die für die Berechnung der Koeffizienten benötigt wurde.
Da die Matrixblöcke nach der Partitionierung voneinander unabhängig sind und diese nur einen Bruchteil der Rechenzeit ausmacht, können Einsparungen sehr effektiv durch Parallelisierung erzielt werden. Alle Zeitangaben sind aber Ergebnis sequentieller Berechnung.

		1				2				3			
N	κ	S_{rel}	S/N	F_{rel}	Zeit	S_{rel}	S/N	F_{rel}	Zeit	S_{rel}	S/N	F_{rel}	Zeit
1280	3.2	21%	4.2	0.45	0.07	34%	6.74	0.19	4.5	93%	18.6	0.02	80.9
5120	6.3	6.7%	5.36	0.47	0.41	9.6%	7.71	0.28	17.9	34%	27.4	0.08	1509
20480	12.6	1.7%	5.36	0.54	2.01	2.2%	7.1	0.33	55.1	7.5%	23.9	0.1	5767

Tabelle 5.2: Ergebnisse der Tschebyscheffinterpolation auf der Kugel

Gegenüber dem ACA-Verfahren sind deutliche Verbesserungen beim Speicherbedarf zu erkennen.

Abbildung 5.11 vergleicht den Speicherbedarf der unkomprimierten Matrix mit dem von ACA mit $\varepsilon = 0.001$ und der Approximation mit gemeinsamen Basen mit dem Polynomgrad $p = 3$.

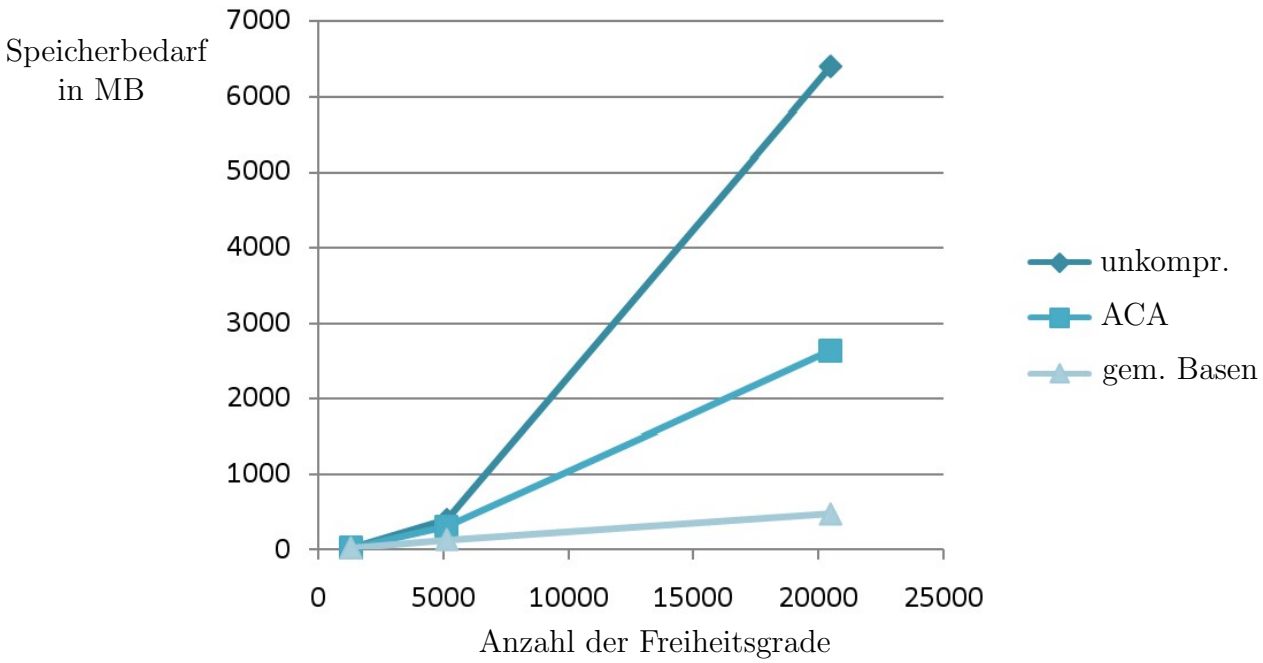

Abbildung 5.11: Speicherbedarf der Kompressionsalgorithmen auf der Kugel in MB

In Abbildung 5.12 sind zur Verdeutlichung für dieselben Kompressionsarten der Speicher pro Freiheitsgrad angegeben, der bei der Approximation mit gemeinsamen Basen quasi konstant ist.

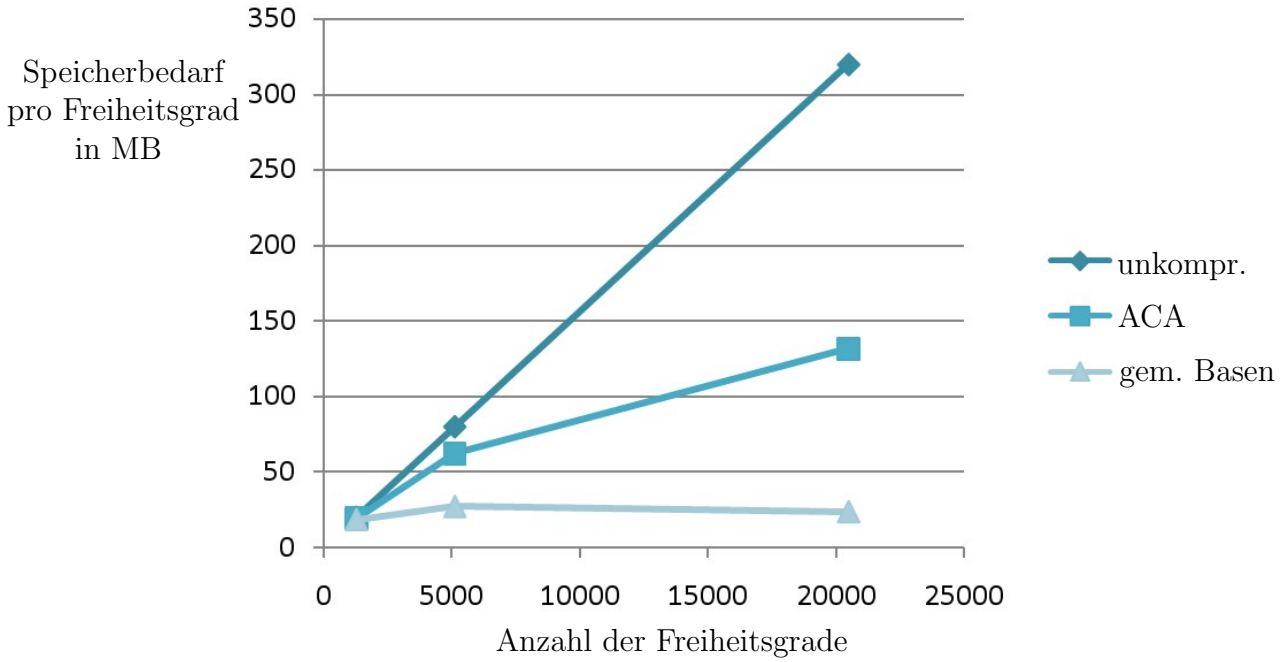

Abbildung 5.12: Speicherbedarf pro Freiheitsgrad der Kompressionsalgorithmen auf der Kugel in MB

Die Ergebnisse spiegeln exakt unsere theoretischen Überlegungen wider.

Kritisch ist das Anwachsen des Approximationsfehlers bei steigender Frequenz zu betrachten. Dies ist angesichts der theoretischen Fehlerabschätzung verschmerzbar. Außerdem ist das Wachsen der Fehler eher gering. Ausgehend von der kleinsten Auflösung steigt der Fehler durch Vervierfachung der Frequenz deutlich weniger an, als durch Verringerung des Polynomgrades um eins.

Kapitel 6

Zusammenfassung

In dieser Arbeit wurden verschiedene Methoden zur Steigerung der Effizienz bereits existierender Verfahren zur Erzeugung Hierarchischer Matrizen vorgestellt. Sie dienen der Kompression von Steifigkeitsmatrizen elliptischer Differentialoperatoren, so dass auf den Approximanten algebraische Operationen durchführbar bleiben.

Unsere Algorithmen bauen auf der Randelementmethode auf, die ein Randwertproblem in ein lineares Gleichungssystem überführt. Da diese Systeme bei praktischen Anwendungen enorm viele Freiheitsgrade haben, ist für ihre Speicherung und Lösung ein effizientes Verfahren von großer Bedeutung.

Hierarchische Matrizen ermöglichen die Behandlung dieser Matrizen mit fast linearem Speicheraufwand. Auch die Matrix-Vektor-Multiplikation ist in logarithmisch-linearer Zeit durchführbar. Viele iterative Verfahren zum Lösen linearer Gleichungssysteme können damit sogar schneller auf einer $\mathscr{H}$ - Matrix arbeiten als auf einer unkomprimierten.
Die Einträge können effizient durch den Adaptive Cross Approximation Algorithmus aus originalen berechnet werden.

Das Verfahren Recompression for Adaptive Cross Approximation wurde in Kapitel 4 vorgestellt. Erstmals wurde detailliert beschrieben, wie es auf Doppelschichtpotentialoperatoren angewendet werden kann. Es wurde gezeigt, dass der zusätzliche Speicher- und Berechnungsaufwand nichts an der Komplexitätsklasse ändert.

Kapitel 5 beschreibt Ideen zur Behandlung des Helmholtzoperators. Obwohl er als elliptische Differentialoperator eine asymptotisch glatte Singularitätenfunktion besitzt, unterscheidet er sich massiv von den meisten anderen. Im Hochfrequenzfall werden alle

Abschätzungen, die die Konvergenz der Einträge berechnenden Verfahren garantieren sollen, unbeschränkt und damit wertlos. Neue Partitionierungsvarianten und partielle Glättung ermöglichen den Nachweis der Konvergenz unabhängig von der Frequenz.
Wir haben gesehen, dass die Hochfrequenzzerlegung als Grundlage für ACA zu einer für uns inakzeptablen Speicherkomplexität führt. Eine mögliche Lösung wurde in der Interpolation der Kernfunktion mit Tschebyscheffpolynomen gefunden.

Zur Bestätigung der Theorien wurden die vorgestellten Verfahren an Standardbeispielen getestet.

Kapitel 7

Ausblick

Zur Weiterführung dieser Arbeit könnte man Versuche machen, bei denen RACA auf dem Helmholtzoperator arbeitet. Zumindest die Speicherkomplexität sollte unsere Anforderungen erfüllen.

Es wäre denkbar mit anderen gemeinsamen Basen als denen der Tschebyscheffpolynome zu arbeiten.

Natürlich bietet es sich an weitere Berechnungen an anderen Geometrien durchzuführen und die momentanen Quellcodes zu optimieren.

In Kapitel 5 wurde nur kurz darauf eingegangen, dass die Partitionierung und Glättung theoretisch auch für das Doppelschichtpotential und für die anderen Diskretisierungsverfahren funktioniert. Zur Kontrolle könnten dazu numerische Experimente durchgeführt werden.

Das Testen der Algorithmen beinhaltete bisher nur die Kompression der Steifigkeitsmatrizen. Es wäre wünschenswert einmal ein praktisch relevantes Beispiel mit tatsächlichen Randwerten zu lösen. Ein Vergleich mit gemessenen Werten in Zusammenarbeit mit Physikern oder Forschungsgruppen anderer Fachbereiche, die mit elliptischen Differentialoperatoren arbeiten, wäre aebenfalls interessant.

Literaturverzeichnis

[1] Mario Bebendorf and S. Kunis. Recompression techniques for adaptive cross approximation. *Journal of Integral Equations and Applications*, 2009.

[2] Robert Plato. *Numerische Mathematik kompakt: Grundlagenwissen für Studium und Praxis*. Vieweg, 2004.

[3] Gisela Engeln-Müllgres and Fritz Reutter. *Numerik Algorithmen*. VDI Verlag, 1996.

[4] Olaf Steinbach. *Numerische Näherungsverfahren für elliptische Randwertprobleme*. B. G. Teubner, 2003.

[5] Sarah Engleder. Stabilisierte Randintegralgleichungen für äussere Randwertprobleme der Helmholtz-Gleichung. *http://www.numerik.math.tu-graz.ac.at/berichte/Bericht1006.pdf*, 2006.

[6] William Charles Hector McLean. *Strongly elliptic systems and boundary integral equations*. Cambridge University Press, 2000.

[7] Mario Bebendorf. *Hierarchical Matrices*. Springer, 2008.

[8] Stefan Sauter and Christoph Schwab. *Randelementmethoden*. B. G. Teubner, 2004.

[9] Wolfgang Hackbusch. A sparse matrix arithmetic based on H-matrices. Part I: Introduction to H-matrices. *Computing 62, Seiten 89-108*, 1999.

[10] Wolfgang Hackbusch and Boris Khoromskij. A sparse H-matrix arithmetic. II. Application to multi-dimensional problems. *Computing 64, Seiten 21-47*, 2000.

[11] Gerhard Kockläuner. *Multivariate Datenanalyse: am Beispiel des statistischen Programmpakets SPSS*. Vieweg, 2000.

[12] Sergey A. Goreinov and Eugene E. Tyrtyshnikov andNikolai L. Zamarashkin. A theory of pseudo-skeleton approximations. *Linear Algebra and its Applications 261, Seiten 1-21*, 1997.

[13] Martin Hanke-Bourgeois. *Grundlagen der numerischen Mathematik und des wissenschaftlichen Rechnens*. Teubner, 2006.

[14] Mario Bebendorf and Sergej Rjasanow. Adaptive Cross Approximation with High Frequency Clustering. *technical report*, 2010.

[15] Eric Darve. The Fast Multipole Method: Numerical Implementation. *Journal of Computational Physics 160, Seiten 195-240*, 2000.

[16] Michael Bratsch. Effiziente Simulation von stationären mikromagnetischen Phänomenen. *http://lips.informatik.uni-leipzig.de/files/Diplomarbeit-Bratsch.pdf*, 2009.

[17] Christoph Erath. Randelementemethode, eine Einführung. *http://www.mathematik.uni-ulm.de/numerik/staff/erath/material/Lecturenotes.pdf*, 2007.

[18] Samuel Ferraz-Leite. A posteriori Fehlerschätzer für die Symmsche Integralgleichung in 3D. *http://www.asc.tuwien.ac.at/˜dirk/download/thesis/ferrazleite.pdf*, 2007.

[19] Lothar Gaul and Matthias Fischer. Boundary Element Methods. *http://www.iam.uni-stuttgart.de/bem/bem_pages/bem_script.html*, 2002.

[20] Christian Großmann and Hans-Görg Roos. *Numerische Behandlung partieller Differentialgleichungen*. B. G. Teubner, 2005.

[21] Klaus Höllig and Barbara Wohlmuth. Mathematik Online Kurs Numerik. *http://mo.mathematik.uni-stuttgart.de/kurse/kurs5*, 2006.

[22] Martin Hanke-Bourgeois. *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens*. B. G. Teubner, 2006.

[23] Armin Iske. Numerische Mathematik 2. *http://www.math.uni-hamburg.de/home/iske/vorlesungen/numerik2/folien/n.pdf*, 2007.

[24] James Edward Rickett. Spectral factorization of wavefields and wave operators. *http://sepwww.stanford.edu/public/docs/sep109/paper.pdf*, 2001.

Eidesstattliche Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann.

Leipzig, 29. April 2010 _____

Unterschrift