Neue Formen der Adaptivität bei der Kreuzapproximation nicht-lokaler Operatoren

Von der Universität Bayreuth
zur Erlangung des Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

von

Maximilian Bauer

aus Weiden

1. Gutachter:	Prof. Dr. Mario Bebendorf
2. Gutachter:	Prof. Dr. Steffen Börm

Tag der Einreichung:	05. Juli 2022
Tag des Kolloquiums:	22. November 2022

Zusammenfassung

Nicht-lokale Operatoren stellen den vorrangigen Anwendungsbereich der in dieser Arbeit vorgestellten Methoden und Strategien dar. Da die Diskretisierung derartiger Operatoren und deren Lösung in linearen Gleichungssystemen sehr speicher- und zeitintensiv ist, werden an dieser Stelle Techniken benötigt, um die Anforderungen an den Speicher und die Rechenzeit zu verringern. In Verbindung mit hierarchischen Matrizen existiert mit der adaptiven Kreuzapproximation (ACA) bereits eine Methode, diskretisierte Operatoren effizient zu behandeln. Jedoch generiert ACA eine im weitesten Sinne universelle Approximation, wodurch hierbei noch redundante oder auch unnötige Informationen entstehen, gespeichert und anschließend verarbeitet werden. In einigen Anwendungen kann dies durchaus vorteilhaft sein, vor allem, wenn der diskrete Operator in vielen verschiedenen Gleichungssystemen Verwendung findet. Im Gegenzug ist bei einmaliger Anwendung des diskretisierten Operators eine Approximation, welche auf das Problem zugeschnitten ist, effizienter als eine universelle Approximation.

In theoretischer Sicht erfordert ACA eine bestimmte Punktauswahl, damit im zugrunde liegenden Interpolationsproblem die eindeutige Lösbarkeit garantiert werden kann. Bei den meisten Problemstellungen liefert diese gut analysierte Punktauswahl vernünftige Ergebnisse. Jedoch existieren auch Beispiele, wie die Anwendung der ACA bei nicht glatten Gebieten, in denen ACA mit dieser Punktauswahl nicht konvergiert.

Die vorliegende Arbeit setzt an den beiden oben beschriebenen Problemen an. Zuerst wird unter Benutzung der Interpolation mittels multivariater bzw. radialer Basisfunktionen die Punktauswahl bei ACA verbessert. Der Vorteil bei diesem Funktionensystem ist, dass radiale Basisfunktionen positiv definit sind und dadurch die eindeutige Lösbarkeit des Interpolationsproblems gewährleistet ist. Die Approximation der Funktion, auf welcher ACA angewendet wird, kann hierbei mittels der Fouriertransformation sichergestellt werden. Somit können die Punkte bei ACA anhand der sogenannten Fülldichte gewählt werden, welche eine bessere Abdeckung der Geometrie ermöglicht.

Des Weiteren wird ACA mit zusätzlichen adaptiven Elementen ausgestattet, um eine spezialisierte Approximation zu erreichen. Hierfür werden Fehlerschätzer und Verfeinerungsstrategien, wie das „Dörfler Marking“, eingeführt, welche die Auswahl derjenigen Blöcke gewährleistet, deren Approximationen den größtmöglichen Genauigkeitsgewinn liefern. Im letzten Schritt wird schließlich die Approximation der Blöcke an das iterative Lösungsverfahren gekoppelt, um so ein hybrides Lösungsverfahren zu erhalten, welches weniger Rechenzeit und Speicheranforderungen benötigt.

Getestet werden die entwickelten Verfahren an verschiedenen numerischen Problemen, wie Randwertproblemen bzgl. der Laplace- oder Lamé-Gleichung, welche mittels der Randelementemethode behandelt werden. Zudem wird die Konvergenz der ACA bei nicht-glatten Geometrien unter Verwendung der Fülldichte anhand numerischer Beispiel gezeigt.

Abstract

Non-local operators represent the primary application of the methods and strategies presented in this thesis. Since the discretization of such operators and their solution in linear systems of equations is very memory and time intensive, techniques are needed at this point to reduce the memory and computation time requirements. In conjunction with hierarchical matrices, adaptive cross approximation (ACA) already exists as a method to handle discretized operators efficiently. However, ACA generates a universal approximation in the broadest sense, whereby redundant or unnecessary information is created, stored and subsequently processed. In some applications this can be quite advantageous, especially if the discrete operator is used in many different equation systems. In turn, when the discretized operator is used once, an approximation which is tailored to the problem is more efficient than a universal approximation.

From a theoretical point of view, ACA requires a certain point selection in order to guarantee unique solvability in the underlying interpolation problem. In most problems, this well-analyzed point selection yields reasonable results. However, examples also exist in which ACA does not converge with this point selection, for instance when dealing with non-smooth geometries.

The present work addresses the two problems described above. First, the point selection for ACA is improved using interpolation by means of multivariate or radial basis functions. The advantage of this function system is that radial basis functions are positive definite and thus the unique solvability of the interpolation problem is guaranteed. The approximation of the function to which ACA is applied can be ensured by means of the Fourier transform. Thus, the points in the ACA method can be selected on the basis of the so-called fill distance, which enables a better coverage of the geometry.

Furthermore, ACA is equipped with additional adaptive elements in order to achieve a specialised approximation. For this purpose, error estimators and refinement strategies, such as „Dörfler marking“, are introduced, which ensure the selection of those blocks whose approximations provide the greatest possible gain in accuracy. Finally, in the last step, the approximation of the blocks is coupled to the iterative solution process in order to obtain a hybrid solution procedure, which requires less computing time and memory.

The developed methods are tested on various numerical problems, such as boundary value problems concerning the Laplace or Lamé equation, which are treated by means of the boundary element method. In addition, the convergence of ACA for non-smooth geometries using the fill distance is shown by numerical examples.

Danksagung

Zunächst möchte ich Herrn Prof. Dr. Mario Bebendorf für die kontinuierliche Betreuung und Ermöglichung meiner Promotion am Lehrstuhl für Wissenschaftliches Rechnen danken. Die zahlreichen fachlichen Gespräche und Diskussionen habe ich stets als motivierend und gewinnbringend empfunden.

Ich danke auch Herrn Prof. Dr. Steffen Börm für die Übernahme des Zweitgutachtens dieser Dissertation.

Des Weiteren geht mein Dank an Herrn Prof. Dr. Kurt Chudej für seine diversen Ratschläge, die nicht selten die Zusammenarbeit an der Universität Bayreuth vereinfacht und erleichtert haben.

Danken möchte ich auch meinen aktuellen und ehemaligen Kolleginnen und Kollegen am Lehrstuhl für Wissenschaftliches Rechnen: Massimo Pinzer-Braese für seinen technischen Support, Anke Bornmann für ihre Unterstützung in verwaltungstechnischen Aufgaben sowie Bernd Feist, Gaby Folger, Thomas Rau, Christina Schwarz und Sandra Aziz für die zahlreichen fachlichen Diskussion und die gute Zusammenarbeit. Ich danke all meinen Freunden, die mich während der Zeit der Promotion unterstützt haben.

Schließlich richte ich meine Worte an meine Familie, die mir immer die wesentlichen Dinge vor Auge gehalten hat. Vielen Dank Oma und Opa für eure Unterstützung! Letztendlich der wichtigste Dank: Vielen Dank Mama für jegliche Unterstützung, die stets ehrlichen Worte und die kleinen Denkanstöße bei wichtigen Entscheidungen.

Inhaltsverzeichnis

1. Einleitung

Die Bedeutung von nicht-lokalen Operatoren spiegelt sich in diversen Anwendungsmöglichkeiten wider. So treten sie in Problemstellungen aus den Ingenieurswissenschaften oder den Naturwissenschaften auf. Klassische Beispiele hierfür sind Integraloperatoren mit singulärem Kern, welche bei der Umformulierung von partiellen Differentialgleichungen, wie der Laplace-Gleichung, der Helmholtz-Gleichung, den Elastizitätsgleichungen/Lamé-Gleichungen oder den Stokes'schen Glei-chungen, als Integralgleichungen auftreten, siehe [65, 70]. Auch der fraktionelle Laplace-Operator, mit dem zum Beispiel Finanz-Modelle bei anormalen Diffusionsprozessen beschrieben werden können, kann als nicht-lokaler Operator aufgefasst werden, siehe [2, 32, 33].

Anschaulich lässt sich die Funktionsweise nicht-lokaler Operatoren anhand eines Gravitationsfelds erklären. Dieses erstreckt sich nicht nur lokal um den Massenschwerpunkt selbst sondern über das gesamte betrachtete Gebiet. Möchte man nun die Anziehungskräfte vieler Massekörper simulieren, kann die Anzahl an zu berechnenden Wechselwirkungen je nach Anzahl an Körpern sehr groß sein. Diese Gedanken lassen sich auf die numerische Behandlung von partiellen Differentialgleichungen anwenden, welche zuvor als Integralgleichung umgeschrieben wurden. Hierbei hat die im Integraloperator enthaltene, meist singuläre Kernfunktion auch Auswirkungen auf das gesamte Rechengebiet. Zwar muss nach Umformulierung der Differentialgleichung in ihre integrale Darstellung nur noch der Rand des Gebiets betrachtet werden, jedoch müssen je nach Genauigkeit der Randdiskretisierung viele Punktinteraktionen berechnet werden, was einen hohen Zeit- und Speicheraufwand zur Folge hat.

Die eben grob beschriebene Technik wird Randintegralmethode gennant (eng. Boundary Element Method, BEM), [65, 70]. Anders als beim Standardverfahren zur numerischen Behandlung von partiellen Differentialgleichungen, der Finiten Elemente Methode (FEM), [29], bei der das gesamte Gebiet diskretisiert werden muss, erfolgt bei BEM durch die ausschließliche Betrachtung des Randes eine Reduzierung der Dimension und der Größe des resultierenden Gleichungssystems. Zu Beginn stellte BEM aber keine wirkliche Alternative zu FEM dar. Der Grund war ein einfaches Abwägungsprinzip. Dadurch, dass bei BEM ein nicht-lokaler Operator diskretisiert wird, folgt im Gleichungssystem eine voll besetzte Systemmatrix. Somit war der Gebrauch von BEM nicht lohnenswert, da die Systemmatrix bei FEM dünn besetzt ist.

Die Situation änderte sich mit der stetigen Entwicklung zu schnellen Randelementmethoden. Methoden wie die schnelle Multipolentwicklung [41], Mosaikzerlegungsmethoden [73], hierarchische Matrizen ($\mathcal{H}$-Matrizen) [44, 45] oder $\mathcal{H}^2$-Matrizen [24, 47] reduzieren die Komplexität durch Approximation des diskretisierten Operators so weit, dass BEM eine Alternative zu FEM darstellt. Beispielhaft kann hier die Beschleunigung der Berechnung der Lamé-Gleichungen durch $\mathcal{H}$-Matrizen angeführt werden, siehe [17]. Zudem konnte sogar gezeigt werden, dass die Inverse einer BEM Systemmatrix mittels hierarchischer Matrizen approximiert werden kann, siehe [36, 37].

Während die schnelle Multipolmethode physikalisch motiviert und auf spezifische Probleme zugeschnitten war, konnten hierarchische Matrizen allgemeiner gehalten werden. Wie der Name bereits vermuten lässt, basieren $\mathcal{H}$-Matrizen auf einer hierarchischen Zerlegung des diskreten Operators in geeignete Blöcke. Jeder dieser Blöcke enthält eine Niedrigrang-Approximation für den jeweiligen originalen Blockeintrag, wobei die gesamte resultierende Matrix nur noch einen logarithmisch-linearen Speicherbedarf aufweist. In Verbindung mit iterativen Lösungsmethoden [64], wie der Methode der konjugierten Gradienten, die viele Matrix-Vektor Multiplikationen erfordern, sind die eben erwähnten Darstellungen zusätzlich vorteilhaft, da hierarchische Matrizen die Möglichkeit anbieten, schnelle

Matrix-Vektor Multiplikationen mit logarithmisch-linearer Komplexität auszuführen [15, 46]. Im Allgemeinen sind hierarchische Matrizen gut geeignet, um nicht-lokale Operatoren mit logarithmisch-linearer Komplexität zu behandeln, siehe [15, 22, 43].

Entscheidend für die Möglichkeit einer effizienten Behandlung nicht-lokaler Operatoren ist die Approximation der zugrunde liegenden Kernfunktionen mit separablen Funktionen. Während explizite Kernapproximationen in den Anfängen der schnellen Methoden zur Behandlung von Integraloperatoren, wie die Approximation des Coulomb Potentials mit sphärischen Funktionen [61], verwendet wurden, sind später auch Entwicklungen mit Hilfe der Taylorreihe [59], oder Techniken basierend auf der Interpolation mit Polynomen [24, 22] verwendet worden. In den letzten Jahren ist in diesem Zusammenhang die adaptive Kreuzapproximation (engl. adaptive cross approximation, ACA) (siehe [11, 21]) populär geworden. Die letztgenannte Methode verwendet nur wenige der ursprünglichen Matrixeinträge von A für ihre datenarme Approximation durch hierarchische Matrizen. Da man normalerweise an Appro-ximationen $\tilde{A}$ von A interessiert ist, die zu einem hinreichend genauen Ergebnis führen, wenn $\tilde{A}$ oder seine Inverse auf beliebige Vektoren angewendet wird, ist es üblich, alle Blöcke der Matrixpartition einheitlich mit einer vorgegebenen Genauigkeit zu approximieren. Die Universalität der Approximation des diskreten Operators A hat einen hohen Preis, denn es werden Informationen erzeugt, die für die Lösung des linearen Systems redundant sind. Wenn die zu betrachtende Größe der Fehler der Lösung x des linearen Systems $Ax = b$ ist, ist die aus der üblichen ACA Methode erhaltene Approximation $\tilde{A}$ möglicherweise nicht die ideale Annäherung an A. In einer solchen Situation können bestimmte Teile des diskreten Operators A wichtiger sein als andere. Umgekehrt können abhängig von der rechten Seite b einige Blöcke für den Fehler von x nicht so wichtig sein.

In dieser Arbeit wird vorgeschlagen, die hierarchische Matrixapproximation auf eine blockadaptive Weise zu konstruieren, d.h. zusätzlich zur Adaptivität der ACA in jedem Block wird eine weitere Ebene der Adaptivität zur Konstruktion der hierarchischen Matrix eingefügt. Diese Variante der ACA wird als BACA bezeichnet. Um eine $\mathcal{H}$-Matrixapproximation zu finden, die besser für den Fehler der Lösung geeignet ist, werden aus der Adaptivität bekannte Techniken, wie die Markierungsstrategie nach Dörfler [34], zusammen mit geeigneten Fehlerschätzern [7, 8, 38, 50] verwendet. Die Strategie von Fehlerschätzern ist im Zusammenhang mit numerischen Methoden für partielle Differential- und Integralgleichungen gut bekannt. Adaptive Verfahren konzentrieren sich hierbei in der Regel auf die Gitterverfeinerung. Für die neue Methode BACA werden ähnliche Strategien genutzt, um sukzessive die blockweisen Niedrigrang-Approximationen zu verbessern, wobei sich das zugrunde liegende Gitter und die hierarchische Blockstruktur nicht ändern. Daher approximieren ACA und BACA den Operator auf demselben Gitter. Obwohl die Ideen dieses Ansatzes im Zusammenhang mit ACA vorgestellt werden, können sie auch auf andere Niedrigrang-Approximationsverfahren angewendet werden.

Im Gegensatz zur Konstruktion der Approximation $\tilde{A}$ von A auf die übliche Weise und der Lösung eines einzigen linearen Systems $\tilde{A}x = b$, wird eine Folge von $\mathcal{H}$-Matrizen A_k zu A erzeugt und jedes lineare System $A_k x_k = b$ für x_k gelöst. Auf den ersten Blick scheint dies aufwändiger zu sein, als wenn $\tilde{A}x = b$ nur einmal gelöst wird. Allerdings kann die Konstruktion der nächsten Approximation x_{k+1} durch die Näherungslösung x_k gesteuert werden und x_{k+1} kann effizient als eine Aktualisierung von x_k berechnet werden. Außerdem kann die Genauigkeit von x_k im iterativen Lösungsverfahren an den Fehlerschätzer angepasst werden. Dies erlaubt es, die Assemblierung der Matrix mit der iterativen Lösung des linearen Systems zu verknüpfen. Neben der Reduzierung des numerischen Aufwands für die Assemblierung und die Speicherung der Matrix, hat dieser Ansatz den praktischen Vorteil, dass die blockweise Genauigkeit der ACA kein erforderlicher Parameter mehr ist. Die Wahl dieses Parameters in der üblichen ACA Approximation ist nicht offensichtlich, da die Beziehung zwischen der blockweisen Genauigkeit und dem Fehler der Lösung in der Regel vom jeweiligen Problem abhängt. Mit dem in dieser Arbeit untersuchten neuen Ansatz wird die Genauigkeit der Approximation automatisch

angepasst.

Ähnliche Ideen können auch auf die Matrix-Vektor Multiplikation $b = Ax$ übertragen und angewendet werden. In diesem Fall ist die zu bewertende Größe nicht die Lösung eines Gleichungssystems sondern der Vektor b. Aufgrund der Art dieses Problems, welches im Gegensatz zur Lösung eines linearen Gleichungssystems nicht von inverser Natur ist, gestaltet sich die Theorie einfacher. Zudem entfällt die Kombination mit iterativen Lösungsverfahren.

Die dritte Thematik betrifft direkt die theoretischen Aspekte der ACA. Die bisherige Konvergenzanalyse der adaptiven Kreuzapproximation zeigt, dass die richtige Auswahl der Pivotpunkte entscheidend ist für die eindeutige Lösbarkeit des zugrunde liegenden Interpolationsproblems [15]. Die in dieser Arbeit vorgestellten neuen a-priori Fehlerabschätzungen, die auf der Interpolation durch radiale Basisfunktionen beruhen, ermöglichen neue Pivotstrategien für ACA. Während bei den früheren Ergebnissen, die auf polynomialen Interpolationsfehlerschätzungen basieren, die Pivots so gewählt werden müssen, dass die Unisolvenz des Polynominterpolationsproblems garantiert ist, zeigen die neuen Resultate, dass nur die Fülldichte der Pivotpunkte entscheidend ist für die Konvergenz der ACA.

Diese Arbeit ist wie folgt aufgebaut. Zuerst werden einige grundlegende Themen, wie die Approximation mit separablen Funktionen, Niedrigrang-Matrizen oder Matrix Partitionierung, zur Approximation von nicht-lokalen Operatoren angesprochen. Des Weiteren sollen im ersten Teil die Interpolation mit multivariaten Funktionen bzw. radialen Basisfunktionen sowie deren Anwendung im Fall der ACA vorgestellt werden. Anschließend folgen die Erweiterungen des ACA bzgl. der Matrix-Vektor Multiplikation und linearer Gleichungssysteme, was zum Ende dieses Abschnitts in die Kombination der erweiterten ACA und iterativer Lösungsverfahren resultiert. Der dritte Teil dieser Arbeit beschäftigt sich mit der Anwendung der neuen Methoden in der Randintegralmethode. Zunächst werden hier die Theorie zu BEM im Fall der Laplace-Gleichungen und der Lamé-Gleichungen erläutert. Am Ende zeigen diverse numerische Tests die Funktionsweise der neuen Methoden.

I. Approximation nicht-lokaler Operatoren

Wie in der Einleitung bereits erwähnt treten nicht-lokale Operatoren in vielen Anwendungsgebieten auf. Die Diskretisierung derartiger Operatoren führt häufig zu voll besetzten Systemmatrizen. Dies bedeutet zugleich einen hohen Speicher- und Rechenaufwand, vor allem, wenn der diskrete Operator in Verbindung mit der Lösung eines linearen Gleichungssystems steht. Im Folgenden werden wir uns mit der Approximation von nicht-lokalen Operatoren beschäftigen, um sowohl die Anforderungen an den Speicherbedarf als auch die Komplexität im Hinblick auf Rechenoperationen für den diskreten Operator zu reduzieren. Wir beginnen mit der Annäherung durch separable Funktionen.

2. Approximation mit separablen Funktionen

Separable Funktionen bilden den Ausgangspunkt unserer Untersuchungen zu nicht-lokalen Operatoren. Diese erlauben es uns, multivariate Funktionen mit Hilfe einer Summe von univariaten Funktionen zu approximieren, siehe auch [15].

Definition 2.1. *Seien $X, Y \subset \mathbb{R}^d$ zwei Gebiete. Eine Funktion $\kappa : X \times Y \to \mathbb{R}$ heißt degeneriert oder separabel, falls $k \in \mathbb{N}$ und Funktionen $u_l : X \to \mathbb{R}$, $v_l : Y \to \mathbb{R}$, $l = 1, ..., k$, existieren, sodass*

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

Die Zahl k wird als Grad der Degeneriertheit bezeichnet.

Im folgenden Abschnitt wird die Problemstellung formuliert werden, bei welcher die Eigenschaft separabler Funktionen vorteilhaft ist.

2.1 Problemstellung

Sei $\mathcal{A}$ ein nicht-lokaler Operator, welcher linear von einer bivariaten Funktion $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ abhänge. Die betrachteten Operatoren haben demnach die Darstellung

$$(\mathcal{A}v)(x) := \int_{\Omega} \kappa(x, y) v(y) \, \mathrm{d}\mu_y, \quad x \in \Omega,$$

wobei $\Omega \subset \mathbb{R}^d$ ein Gebiet, v eine geeignete Funktion und μ das zugehörige Maß bezeichne. Da wir uns im späteren Verlauf mit der Diskretisierung von elliptischen partiellen Differentialgleichungen auseinandersetzen werden, wird der Kern κ zumeist eine singuläre Funktion sein. Obiges Integral bei der Definition des Operators $\mathcal{A}$ ist daher in geeigneter Weise zu interpretieren.

Den Operator $\mathcal{A}$ auszuwerten bzw. mit Diskretisierungen von $\mathcal{A}$ umzugehen, ist aufgrund seiner Eigenschaften mit viel Aufwand verbunden. Unser Ziel ist eine grundlegenden Reduktion des Aufwands, indem eine effiziente Darstellung für die Kernfunktion κ gefunden wird. Der Idealfall ist, wenn κ separabel ist, was im Allgemeinen nicht zutreffen wird. Oft können bivariate jedoch mit separablen Funktionen approximiert werden, d.h.

$$\kappa(x,y) \approx \tilde{\kappa}(x,y) = \sum_{l=1}^{k} u_l(x) v_l(y), \quad x \in X,\, y \in Y.$$

Dementsprechend wird auch der Operator $\mathcal{A}$ approximiert

$$(\mathcal{A}v)(x) \approx (\tilde{A}v)(x) = \int_{\Omega} \tilde{\kappa}(x,y) v(y) \,\mathrm{d}\mu_y.$$

Im Folgenden ist eine Methode angeführt, mit der unter bestimmten Voraussetzungen eine separable Approximation erzeugt werden kann.

2.2 Die Kreuzapproximation

Es gibt viele Möglichkeiten eine separable Approximation zu konstruieren. Die wohl bekannteste Approximation dürfte die des Coulomb Potentials $1/\|x-y\|_2$ mit sphärischen Funktionen sein, welche auch „multipole expansion“ genannt wird, siehe [63]. Auch die Entwicklungen mit Hilfe einer Taylorreihe [61] oder Techniken basierend auf der Interpolation mit Polynomen [24, 22] können dafür verwendet werden. Die Methode der Wahl in unserem Fall wird die Kreuzapproximation sein, welche in Abschnitt 2.2.1 beschrieben wird. Die Existenz von separablen Approximationen kann mit der Annahme, dass die betrachteten Funktionen asymptotisch glatt sind, garantiert werden.

Definition 2.2. *Eine Funktion $\kappa : X \times Y \to \mathbb{R}$ mit $\kappa(\cdot, y) \in C^\infty(X \setminus \{y\})$ für alle $y \in Y$ heißt bezogen auf x asymptotisch glatt, falls Konstanten c und γ existieren, sodass für alle $y \in X$ und alle $\alpha \in \mathbb{N}_0^d$ die Abschätzung*

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

gilt, wobei $p = |\alpha|$.

In den Anwendungsbeispielen in Abschnitt III werden zum Beispiel elliptische Operatoren auftreten, deren Kernfunktionen asymptotisch glatt bzgl. beider Variablen x und y sind. In diesen Fällen wird die Bedingung

$$\min\{\operatorname{diam} X, \operatorname{diam} Y\} \leq \eta \operatorname{dist}(X,Y) \tag{2.1}$$

für ein $0 < \eta < 1$ ausreichend sein, um die asymptotische Glattheit von κ zu gewährleisten. Bedingung (2.1) besagt im Grunde nichts anderes als, dass die beiden Gebiete X und Y weit genug entfernt voneinander sein müssen, vgl. [12, 15].

2.2.1 Konstruktion separabler Funktionen

Wir befinden uns in der Situation, wie sie durch Bedingung (2.1) beschrieben wird. Ausgehend von der Glattheit von κ, welche uns am Ende einen geeigneten Fehler garantieren wird, werden Restriktionen von κ als Basis der Approximation genutzt. Für Punkte $x_l \in X$ und $y_l \in Y$, $l = 1, ..., k$, seien

$$\kappa(x, [y]_k) = \begin{pmatrix} \kappa(x, y_1) \\ \vdots \\ \kappa(x, y_k) \end{pmatrix} \in \mathbb{R}^k \quad \text{und} \quad \kappa([x]_k, y) = \begin{pmatrix} \kappa(x_1, y) \\ \vdots \\ \kappa(x_k, y) \end{pmatrix} \in \mathbb{R}^k.$$

Dann stellt

$$\kappa(x, y) \approx \kappa(x, [y]_k)^T C_k^{-1} \kappa([x]_k, y)$$

mit

$$C_k = \begin{pmatrix} \kappa(x_1, y_1) & \dots & \kappa(x_1, y_k) \\ \vdots & & \vdots \\ \kappa(x_k, y_1) & \dots & \kappa(x_k, y_k) \end{pmatrix} \in \mathbb{R}^{k \times k}$$

eine Approximation an die Kernfunktion κ dar. Zuletzt bezeichne r_k den Restterm, sodass

$$\kappa(x, y) = \kappa(x, [y]_k)^T C_k^{-1} \kappa([x]_k, y) + r_k(x, y)$$

gilt. Durch die Konstruktion von Folgen $\{s_k\}$ und $\{r_k\}$ kann eine Approximation an κ wie folgt generiert werden:

1. Starte mit

$$r_0(x, y) = \kappa(x, y)$$
$$s_0(x, y) = 0.$$

2. Iteriere über $k = 0, 1, ...$ mit

$$r_{k+1}(x, y) = r_k(x, y) - \lambda_{k+1} r_k(x, y_{k+1}) r_k(x_{k+1}, y)$$
$$s_{k+1}(x, y) = s_k(x, y) + \lambda_{k+1} r_k(x, y_{k+1}) r_k(x_{k+1}, y),$$

   wobei $\lambda_{k+1} = 1/r_k(x_{k+1}, y_{k+1})$ und zu beachten ist, dass x_{k+1}, y_{k+1} so gewählt werden, sodass $r_k(x_{k+1}, y_{k+1}) \neq 0$ gilt.

Obiger Iteration folgend, interpoliert s_k (siehe [15]) auf den bereits ausgewählten Punkten x_l bzw. y_l und wir erhalten:

Lemma 2.3. *Für $1 \leq l \leq k$ gelten die folgenden beiden Aussagen:*

(i) $r_k(x, y_l) = 0$ *für alle* $x \in X$,

(ii) $r_k(x_l, y) = 0$ *für alle* $y \in Y$.

Um die Existenz dieser Approximation garantieren zu können, sollten wir wissen, dass die Matrix C_k nicht singulär ist. Sei dazu $C_k^{(l)}(y) \in \mathbb{R}^{k \times k}$ diejenige Matrix, welche durch das Ersetzen der l-ten Spalte von C_k mit dem Vektor $\kappa([x]_k, y)$ entsteht. Dann zeigt Lemma 2.4, dass C_k invertierbar ist, vgl. [15].

Lemma 2.4. *Für $1 \leq l < k$ gelten*

$$\det C_k^{(l)}(y) = r_{k-1}(x_k, y_k) \det C_{k-1}^{(l)}(y) - r_{k-1}(x_k, y) \det C_{k-1}^{(l)}(y_k)$$

und

$$\begin{aligned} \det C_1^{(1)}(y) &= r_0(x_1, y), \\ \det C_k^{(k)}(y) &= r_{k-1}(x_k, y) \det C_{k-1}, \quad k > 1. \end{aligned}$$

Genauer gesagt, heißt das

$$\det C_k = r_0(x_1, y_1) \cdot ... \cdot r_{k-1}(x_k, y_k).$$

Schließlich gilt die folgende Darstellung für die Approximation s_k, siehe [15].

Lemma 2.5. *Die konstruierten Folgen s_k und r_k, $k \geq 0$, erfüllen*

$$s_k(x, y) + r_k(x, y) = \kappa(x, y),$$

wobei für $k \geq 1$ gilt

$$s_k(x, y) = \kappa(x, [y]_k)^T C_k^{-1} \kappa([x]_k, y).$$

Schließlich konnte eine Approximation s_k an die Funktion κ für die Punkte $x_l \in X$ und $y_l \in Y$, $l = 1, ..., k$, konstruiert werden. Nachfolgend bleibt noch zu überprüfen, wie gut κ durch s_k approximiert wird.

2.2.2 Fehleranalyse

Um den Fehler bei der Kreuzapproximation analysieren zu können, muss der Rest r_k abgeschätzt werden. Der Beweis in [15] nutzt hierfür die Bestapproximation in jedem System von Funktionen $\Xi = \{\xi_1, ..., \xi_k\}$. Dort werden qualitative Resultate für ein polynomiales System Ξ präsentiert. Für die Eindeutigkeit des auftretenden Interpolationsproblems in Ξ muss angenommen werden, dass die Vandermonde Matrix $W = [\xi_j(x_i)]_{ij} \in \mathbb{R}^{k \times k}$ nicht singulär ist. Wir halten uns zunächst an die Beweise, wie sie in [15] zu finden sind. In Kapitel zwei werden wir darauf aufbauend eine andere Interpolationstechnik verwenden und ein Funktionensystem angeben, welches die angesprochenen Annahmen nicht benötigt.

Obiger Idee folgend, wird der Interpolationsfehler einer Funktion f durch

$$E_k^{\Xi}[f] := f - \mathcal{I}_k^{\Xi} f$$

definiert, wobei $\mathcal{I}_k^{\Xi}$ unter Benutzung der Lagrange-Funktionen

$$L_l^{\Xi}(x) := \frac{\det W_l(x)}{\det W} \in \Xi, \quad l = 1, ..., k, \tag{2.2}$$

den Interpolationsoperator

$$\mathcal{I}_k^{\Xi} f := \sum_{l=1}^{k} f(x_l) L_l^{\Xi} \tag{2.3}$$

bezeichnet. Dabei entsteht die Matrix $W_l(x)$ dadurch, dass in der Matrix W die l-te Zeile durch den Vektor $[\xi_\mu(x)]_{\mu=1,...,k}$ ersetzt wird. Der in (2.3) definierte Operator ist eine lineare Projektion auf span Ξ, sodass sich der Interpolationsfehler abschätzen lässt durch

$$||E_k^{\Xi}||_\infty \leq \left(1 + \|\mathcal{I}_k^{\Xi}\|\right) \inf_{p \in \text{span } \Xi} \|f - p\|_\infty, \tag{2.4}$$

wobei

$$\|\mathcal{I}_k^\Xi\| := \max\{\|\mathcal{I}_k^\Xi f\|_\infty / \|f\|_\infty \,:\, f \in C(X)\}$$

die übliche Lebesgue-Konstante ist. Damit kann der Fehler der Kreuzapproximation wie folgt dargestellt werden.

Lemma 2.6. *Für $x \in X$ und $y \in Y$ besitzt der Fehler r_k die Darstellung*

$$r_k(x,y) = E_k^\Xi[\kappa_y](x) - \sum_{l=1}^{k} \frac{\det C_k^{(l)}(y)}{\det C_k} E_k^\Xi[\kappa_{y_l}](x),$$

wobei $\kappa_y(x) := \kappa(x,y)$.

Beweis. Bezeichne

$$L^\Xi(x) = \begin{pmatrix} L_1^\Xi(x) \\ \vdots \\ L_k^\Xi(x) \end{pmatrix}$$

den Vektor der Lagrange Funktionen L_l^Ξ, $l = 1, ..., k$, für die Punkte $x_1, ..., x_k$. Mit Lemma 2.5 folgt

$$\begin{aligned} r_k(x,y) &= \kappa(x,y) - \kappa(x,[y]_k)^T C_k^{-1} \kappa([x]_k, y) \\ &= \kappa(x,y) - \kappa([x]_k,y)^T L^\Xi(x) - \left(\kappa(x,[y]_k) - C_k^T L^\Xi(x)\right)^T C_k^{-1} \kappa([x]_k, y) \\ &= E_k^\Xi[\kappa_y](x) - \sum_{l=1}^{k} (C_k^{-1}\kappa([x]_k,y))_l E_k^\Xi[\kappa_{y_l}](x). \end{aligned}$$

Die Behauptung folgt schließlich mit der Cramer'schen Regel. □

Wir haben gesehen, dass sich der Approximationsfehler durch die Wahl der Punkte $y_1, ..., y_k$ beeinflussen lässt. Im Hinblick auf Lemma 2.6 stellt die Wahl der Punkte $y_1, ..., y_k$, sodass Matrizen mit quasi maximalem Volumen erzeugt werden, d.h. die Bedingung

$$|\det C_k^{(l)}(y)| \le c_M |\det C_k|, \quad 1 \le l \le k,\, y \in Y, \tag{2.5}$$

mit einer Konstante $c_M > 1$ erfüllen, ein quasi optimales, jedoch in der Praxis schwer anzuwendendes Kriterium dar. Die Bedingung (2.5) führt uns zu einer Abschätzung der Form

$$|r_k(x,y)| \le (c_M k + 1) \sup_{z \in \{y, y_1, ..., y_k\}} |E_k^\Xi[\kappa_z](x)|.$$

Eine in der Praxis besser handhabbare Bedingung stellt die Wahl der Punkte $y_1, ..., y_k$ durch das Kriterium

$$|r_{k-1}(x_k, y_k)| \ge |r_{k-1}(x_k, y)|, \quad y \in Y, \tag{2.6}$$

dar. Dies führt jedoch auf einen exponentiell wachsenden Faktor von 2^k anstelle des Faktors $c_M k + 1$. Dazu sei angemerkt, dass es sich hierbei um „worst-case" Abschätzungen handelt und diese in der Praxis kaum beobachtbar sind. Genaueres zur alternativen Bedingung (2.6) kann in [15] gefunden werden.

Setzen wir unter Berücksichtigung von Bedingung (2.4) den Interpolationsfehler ein, so erhalten wir

$$|r_k(x,y)| \le (c_M k + 1)(1 + \|\mathcal{I}_k^\Xi\|) \sup_{z \in \{y, y_1, ..., y_k\}} \inf_{p \in \operatorname{span} \Xi} \|\kappa(\cdot, z) - p\|_\infty.$$

Somit ist der Restterm r_k in jedem Funktionensystem Ξ bis auf eine Konstante kleiner als der in Ξ erhaltene Approximationsfehler.

Wie zu Beginn der Fehleranalyse erwähnt, ist die Lösbarkeit des Interpolationsproblems (2.2) und (2.3) an eine strikte Wahl der Punkte x_i, $i = 1, ..., k$, geknüpft. Unter Verwendung eines speziellen Funktionensystems lassen sich die Bedingungen zur Auswahl der Punkte x_i, $i = 1, ..., k$, abschwächen.

3. Interpolation mit radialen Basisfunktionen

Funktionen bzw. Funktionswerte mit radialen Basisfunktionen (RBF) zu interpolieren, bietet, anders als zum Beispiel die Interpolation mit Polynomen, im Hinblick auf unsere Problemstellungen einige Vorteile. So setzen wir die Interpolation mit RBFs im Folgenden dazu ein, um Annahmen, die wir bei der Fehleranalyse der Kreuzapproximation getroffen haben, weglassen zu können und mehr Freiheiten bei der Wahl der Interpolationspunkte zu erlangen. Zunächst gehen wir näher auf das Interpolationsproblem ein, definieren den zugrunde liegenden Funktionenraum und geben Fehlerabschätzungen an. Im Anschluss daran werden die vorgestellten Techniken auf die Kreuzapproximation angewendet und am Beispiel der Singularitätenfunktion des Laplace-Operators erläutert.

3.1 Interpolationsproblem und Notation

Gegeben seien eine Punktmenge $X = \{x_1, ..., x_N\} \subseteq \Omega \subseteq \mathbb{R}^d$, d, $N \in \mathbb{N}$, siehe Abbildung 3.1, und (Funktions-) Werte f_i, $1 \leq i \leq N$. Mit einer geeigneten Kernfunktion Φ, welche im späteren Verlauf dieses Kapitels noch genauer spezifiziert werden soll, dient die Funktion

$$s_{f,X}(x) := \sum_{i=1}^{N} \alpha_i \Phi(x, x_i) \tag{3.1}$$

als Interpolante für die vorgegebenen Werte f_i. Die Koeffizienten α_i werden über die Interpolationsbedingungen

$$s_{f,X}(x_i) = f_i, \quad 1 \leq i \leq N, \tag{3.2}$$

bestimmt, d.h. der Koeffizientenvektor α ist die Lösung des linearen Gleichungssystems

$$A\alpha = f$$

mit $A = (\Phi(x_i, x_j)) \in \mathbb{R}^{N \times N}$ und $f = (f_i)$, $i, j = 1, ..., N$.

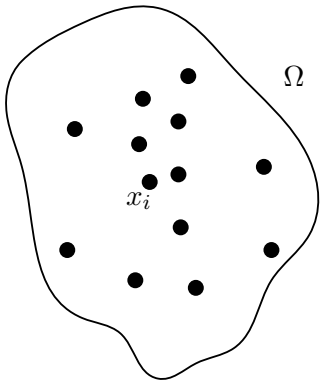

Abb. 3.1: Darstellung einer Punktemenge X im Gebiet Ω.

Ausgangspunkt aller Fehleranalysen ist die Darstellung der Interpolante in ihrer Lagrange-Form. Gegeben sei die Situation, wie sie zu Beginn dieses Abschnitts beschrieben ist. Eine Lösung des Interpolationsproblems (3.1) und (3.2) kann in der Form

$$p(x) := \sum_{i=1}^{N} f_i L_i^{\Phi}(x)$$

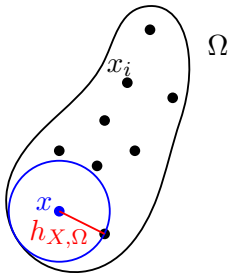

Abb. 3.2: Graphische Darstellung der Fülldichte $h_{X,\Omega}$.

geschrieben werden, wobei $L_i^\Phi(x) = \sum_{j=1}^N \alpha_j^{(i)} \Phi(x, x_j)$ die Lagrange-Funktionen bezeichnen und der Bedingung $L_j^\Phi(x_i) = \delta_{ij}$ genügen, d.h. die Koeffizienten $\alpha^{(i)} \in \mathbb{R}^N$ sind definiert als die Lösungen der linearen Gleichungssysteme $A\alpha^{(i)} = e_i$ mit der Matrix $A := [\Phi(x_i, x_j)]_{ij} \in \mathbb{R}^{N\times N}$. Der Fehler zwischen der Interpolante und der exakten Funktion wird üblicherweise in Termen der Fülldichte

$$h_{X,\Omega} := \sup_{x\in\Omega} \min_{1\le i\le N} \|x - x_i\|_2,$$

siehe Abbildung 3.2, und in der Norm eines geeigneten Funktionenraums gemessen. Dieser Funktionenraum $\mathscr{C}_\Phi$ wird auf Basis einer Variationsformulierung aufgebaut.

3.2 Der Funktionenraum $\mathscr{C}_\Phi$ und dessen Eigenschaften

Madych und Nelson haben in ihren Arbeiten [53, 54] zur multivariaten Interpolation den Funktionenraum $\mathscr{C}_\Phi$ vorgestellt und analysiert. In späteren Artikeln präsentierten sie zudem Fehlerabschätzungen von exponentieller Ordnung. Wiederum einige Jahre später konnte diese Theorie vor allem von Schaback mit radialen Basisfunktionen [30] und Hilberträumen mit reproduzierendem Kern [3] in Verbindung gebracht werden, sodass die Theorie heute eher unter der Interpolation mit radialen Basisfunktionen, siehe z.B. [66, 67, 74], bekannt ist. Im Folgenden werden wir kurz den Funktionenraum $\mathscr{C}_\Phi$ nach Madych und Nelson [55] einführen und die für uns wichtigsten Eigenschaften und Fehlerabschätzungen nennen.

Sei $\Phi : \mathbb{R}^d \to \mathbb{R}$ eine stetige und positiv-definite Funktion. An dieser Stelle benutzen wir die Variationsformulierung einer positiv-definiten Funktion, d.h. es gilt

$$\int_{\mathbb{R}^d\times\mathbb{R}^d} \Phi(x-y)\varphi(x)\overline{\varphi(y)}\,\mathrm{d}x\,\mathrm{d}y > 0$$

für alle $0 \neq \varphi \in C_0^\infty(\mathbb{R}^d)$. Die Fouriertransformierte solcher Funktionen bestimmt ein Maß μ auf $\mathbb{R}^d \setminus \{0\}$, sodass

$$\int \Phi(x)\varphi(x)\,\mathrm{d}x = \int \hat{\varphi}(\xi)\,\mathrm{d}\mu(\xi), \quad \varphi \in C_0^\infty(\mathbb{R}^d).$$

Nach [53, 54, 55] definieren wir den Raum $\mathscr{C}_\Phi$ als die Menge der Funktionen f, welche die Bedingung

$$(f,\varphi)_{L^2}^2 \le c^2 \int_{\mathbb{R}^d\times\mathbb{R}^d} \Phi(x-y)\varphi(x)\overline{\varphi(y)}\,\mathrm{d}x\,\mathrm{d}y \tag{3.3}$$

für eine Konstante $c = c(f) > 0$ und alle $\varphi \in C_0^\infty(\mathbb{R}^d)$ erfüllen. Falls $f \in \mathscr{C}_\Phi$, dann definiert zudem die kleinste Konstante c in (3.3) eine Norm $\|\cdot\|_\Phi$ und $\mathscr{C}_\Phi$ ist ein Hilbertraum.

Bemerkung 3.1. *Bei Interpolationsproblemen wird häufig auch ein diskrete Version der Definition einer positiv-definiten Funktion verwendet, d.h. eine stetige Funktion $\Phi : \mathbb{R}^d \to \mathbb{R}$ heißt positiv-definit, falls für alle $N \in \mathbb{N}$, alle Mengen paarweise verschiedener Punkte $X = \{x_1, ..., x_N\} \subseteq \mathbb{R}^d$ und alle $\alpha \in \mathbb{R}^N \setminus \{0\}$ die Bedingung*

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j \Phi(x_i, x_j) > 0$$

gilt.

Eine Charakterisierung von Funktionen über die Bedingung (3.3) ist zumeist schwierig.An dieser Stelle wird oft die Fouriertransformation zu Rate gezogen. Es gilt, dass Elemente $f \in \mathscr{C}_\Phi$ durch Funktion $g \in L^2_\mu$ mit

$$\hat{f}(\xi)\,\mathrm{d}\xi = g(\xi)\,\mathrm{d}\mu(\xi) \tag{3.4}$$

charakterisiert werden können, siehe [53, 54]. Zudem kann nach [74] die Norm $||\cdot||_\Phi$ über die Fouriertransformation dargetellt werden durch

$$||f||_\Phi = \left(\int_{\mathbb{R}^d} \frac{|\hat{f}|^2}{\hat{\Phi}}(\omega)\, d\omega \right)^{1/2}.$$

Im späteren Verlauf bei der Anwendung der multivariaten Interpolation mit RBFs auf die Singularitätenfunktion des Laplace-Operators wird dieser Zusammenhang nützlich sein.

Zum Ende dieses Abschnitts wollen wir noch eine Fehlerabschätzung, wie sie bei Madych and Nelson [55] zu finden ist, angeben:

Theorem 3.2. *Sei Ω ein Würfel mit Kantenlänge b_0. Angenommen μ erfülle die Bedingung*

$$\int |\xi|^k \, d\mu(\xi) \leq \rho^k k!, \quad k \in \mathbb{N}. \tag{3.5}$$

Dann gibt es ein $0 < \lambda < 1$, sodass für alle $f \in \mathscr{C}_k$ die Interpolante p die Abschätzung

$$|f(x) - p(x)| \leq \lambda^{1/h_{\Omega, X_k}} \|f\|_\Phi$$

erfüllt, wobei $X_k = \{x_1, ..., x_k\} \subset \Omega$.

Die Annahme, dass Ω ein Würfel ist, kann verallgemeinert werden. Theorem 3.2 bleibt gültig, solange Ω als Vereinigung von Rotationen und Translationen eines festen Würfels mit Kantenlänge b_0 ausgedrückt werden kann. Tatsächlich erfüllt jede Kugel im $\mathbb{R}^d$ oder jede Menge mit hinreichend glattem Rand diese Voraussetzung.

Bemerkung 3.3. *Obwohl radiale Basisfunktionen zu einer positiv-definiten Vandermonde-Matrix A führen, könnte ihre numerische Stabilität ein Problem sein. Die Eigenwerte von A hängen signifikant von der Verteilung der Punkte und insbesondere von deren Abständen ab. Ein typisches Maß hierfür ist der Separationsabstand*

$$q_X := \frac{1}{2} \min_{x,y \in X, x \neq y} \|x - y\|_2.$$

In unserem Fall, d.h. für die Gauß-Funktion, kann der kleinste Eigenwert von A abgeschätzt werden durch

$$\lambda_{\min}(A) \geq C(2\beta)^{-d/2} \exp\left(\frac{-40,71d^2}{q_X^2 \beta} \right) q_X^{-d},$$

wobei $C = C(d) > 0$ eine von d abhängige Konstante bezeichnet; siehe [74]. Eines der Hauptziele der hier vorgestellten Verfahren ist eine gleichmäßige Abdeckung des betrachteten Gebiets mit Interpolationspunkten und keine Erzeugung von lokalen Clustern, sodass auch vom numerischen Standpunkt aus ein stabiles Verhalten der Vandermonde-Matrix A erwartet wird.

Bemerkung 3.4. *Wenn die Funktion f auf dem gesamten $\mathbb{R}^d$ definiert ist, kann das vorherige Theorem 3.2 auf glatte Mannigfaltigkeiten $X \subset \mathbb{R}^d$ verallgemeinert werden. Die Erweiterung von Theorem 3.2 auf nicht-glatte Mannigfaltigkeiten funktioniert nicht ohne Weiteres und bedarf weiterer Untersuchungen. Allerdings zeigen numerische Tests, dass die vorgestellte Theorie auch für nicht-glatte Mannigfaltigkeiten vernünftige Ergebnisse liefert. Sei dazu $X = \{(x, y, z) \in [-1, 1]^3 : x = 1\} \cup \{(x, y, z) \in [-1, 1]^3 : z = 1\}$ die Vereinigung zweier Seiten des Würfels $[-1, 1]^3$. Auf verschiedenen Diskretisierungen von X wird die Interpolation der Funktion $f(x, y) = |x - y|^{-1}$ unter Benutzung des Gauß-Kerns $\Phi(x) = \exp(-|x|^2)$ betrachtet. Der Fehler zwischen f und dessen Approximation p wird mit einer Diskretisierung von X bestehend aus 32640 Punkten und zwei unterschiedlichen Punkten $y_1 = (2, 2, 2)^T$ und $y_2 = (5, 5, 5)^T$ aus dem Fernfeld getestet. Dann kann der maximale punktweise Fehler gemessen in $X \times \{y_1\}$ und $X \times \{y_2\}$ in Tabelle 3.1 beobachtet werden.*

$h_{X_k,X}$	0.286	0.133	0.0645	0.0317
Max. Fehler in $X \times \{y_1\}$	4.91e-1	6.73e-2	3.16e-2	1.02e-3
Max. Fehler in $X \times \{y_2\}$	1.31e-1	1.62e-2	7.78e-3	4.03e-4

Tab. 3.1: Maximaler Interpolationsfehler zwischen f und p für verschiedene Fülldichten.

Im Folgenden werden wir die gewonnenen Erkenntnisse auf die Kreuzapproximation und die Singularitätenfunktion des Laplace Operators anwenden.

3.3 Anwendung in der Kreuzapproximation

Damit wir den Fehler der Kreuzapproximation analysieren können, muss der Restterm r_k abgeschätzt werden. Wie wir bisher in Kapitel 2 gesehen haben, benutzt der übliche Beweis die Bestapproximation in jedem beliebigen Funkionensystem $\Xi = \{\xi_1, ..., \xi_k\}$. Dort wurden qualitative Resultate für ein polynomielles System Ξ gezeigt. Für die Existenz einer eindeutigen Lösung beim auftretenden Interpolationsproblem musste angenommen werden, dass die Vandermonde-Matrix $V = [\xi_j(x_i)]_{ij} \in \mathbb{R}^{k \times k}$ nicht singulär ist, siehe Abschnitt 2.2.2. Unser Ziel ist es, unter Verwendung der Interpolation mit radialen Basisfunktionen diese Annahme zu vermeiden. Zudem kann mit der Minimierung der Fülldichte eine Vorschrift angegeben werden, um den nächsten Pivotpunkt x_{k+1} auszuwählen, was zu erhöhten Konvergenzraten führt.

Wir werden die Kreuzapproximation und die Interpolation mit radialen Basisfunktionen beispielhaft auf die Singularitätenfunktion des Laplace-Operators anwenden, d.h. betrachtet werden Funktionen der Form

$$f(x, y) = \frac{1}{|x - y|^\alpha}, \quad \alpha > 0,$$

auf zwei Gebieten X, Y mit

$$\max\{\text{diam } X, \text{diam } Y\} \leq \eta \, \text{dist } (X, Y). \tag{3.6}$$

Damit wir die Funktion f mit radialen Basisfunktionen und Theorem 3.2 interpolieren können, müssen im Vorfeld noch zwei Schritte unternommen werden. Als Erstes wird f außerhalb von $X \times Y$ durch

die Funktion

$$\tilde{f}(x,y) = \begin{cases} \frac{1}{\sigma^\alpha}, & |x-y| \le \sigma \\ \frac{1}{|x-y|^\alpha}, & \sigma < |x-y| \le \sigma+\vartheta \\ -\frac{-|x-y|+\sigma+2\vartheta}{\vartheta(\sigma+\vartheta)^\alpha}, & \sigma+\vartheta < |x-y| \le \sigma+2\vartheta \\ 0, & |x-y| > \sigma+2\vartheta \end{cases}$$

fortgesetzt, wobei $\sigma := \text{dist}\,(X,Y)$ und $\vartheta := \text{diam}\,Y + \text{diam}\,X$. Dabei gilt: $\tilde{f}|_{X\times Y} = f$.
Setze nun $F(t) = \tilde{f}(x,y)$ mit $t := x - y$. Der zweite Schritt ist die Glättung von F mit Hilfe des Glättungskerns $g_m(x) = (m/\pi)^{d/2} e^{-m||x||_2^2}$ für $m \in \mathbb{N}$ und $x \in \mathbb{R}^d$, d.h. $F_m = (F * g_m)$. Da F stetig ist, gilt nach [74](Theorem 5.20), dass F_m für $m \to \infty$ gegen F konvergiert.

Sei $\Phi_m(x,y) = \left(\frac{m}{2\pi}\right)^{d/2} e^{-\frac{m}{2}||x-y||_2^2}$. Wir interpolieren $\tilde{f}$ für ein festes $y \in Y$ mit

$$p_y(x) = \sum_{i=1}^{k} F(t_i) L_i^{\Phi_m}(x), \quad t_i = x_i - y, \tag{3.7}$$

auf dem Datensatz $X_k = \{x_1, ..., x_k\}$, wobei $L_i^{\Phi_m}$, $i = 1, ..., k$, die Lagrange-Funktionen für Φ_m und X_k bezeichnen.

Lemma 3.5. *Sei λ_m aus Theorem 3.2 gegeben. Dann gilt für $x \in X$ und $y \in Y$ die Abschätzung*

$$|F(x-y) - p_y(x)| \le (1 + \Lambda_k^{\Phi_m})||F - F_m||_{L^\infty(\mathbb{R}^d)} + c\lambda_m^{1/h_{X_k,X}}||F||_{L^2(\mathbb{R}^d)}, \tag{3.8}$$

wobei $\Lambda_k^{\Phi_m} := \sup_{x\in X} \sum_{i=1}^k |L_i^{\Phi_m}(x)|$ die Lebesgue-Konstante und $c > 0$ eine Konstante bezeichnet.

Beweis. Mit der Dreiecksungleichung folgt unter Berücksichtigung der geglätteten Funktion F_m und der Interpolante $p_y^{(m)}(x) = \sum_{i=1}^{k} F_m(t_i) L_i^{\Phi_m}(x)$ an F_m, dass die Abschätzung

$$|F(x-y) - p_y(x)| \le |F(x-y) - F_m(x-y)| + |F_m(x-y) - p_y^{(m)}(x)| + |p_y^{(m)}(x) - p_y(x)|$$

gilt. Für den dritten Summanden obiger Abschätzung erhalten wir

$$\begin{aligned} |p_y^{(m)}(x) - p_y(x)| &\le |\sum_{i=1}^{k} F_m(x_i - y) L_i^{\Phi_m}(x) - \sum_{i=1}^{k} F(x_i - y) L_i^{\Phi_m}(x)| \\ &\le \Lambda_k^{\Phi_m} ||F - F_m||_\infty, \end{aligned}$$

wobei $\Lambda_k^{\Phi_m} := \sup_{x\in X} \sum_{i=1}^k |L_i^{\Phi_m}(x)|$.

Um den Fehler $|F_m(x-y) - p_y^{(m)}(x)|$ abzuschätzen, werden die Fouriertransformierten von g_m und Φ_m benötigt:

$$\begin{aligned} \hat{g}_m(\omega) &= (2\pi)^{-d/2} e^{-||\omega||_2^2/4m}, \\ \hat{\Phi}_m(\omega) &= (2\pi)^{-d/2} e^{-||\omega||_2^2/2m}. \end{aligned}$$

Damit erhalten wir

$$\begin{aligned} \|F_m\|_{\Phi_m}^2 &= \int_{\mathbb{R}^d} \frac{(2\pi)^d |\hat{F}\hat{g}_m|^2}{\hat{\Phi}_m}\, d\omega = \int_{\mathbb{R}^d} (2\pi)^d |\hat{F}(\omega)|^2 \frac{\left((2\pi)^{-d/2} e^{-||\omega||_2^2/4m}\right)^2}{(2\pi)^{-d/2} e^{-||\omega||_2^2/2m}}\, d\omega. \\ &= (2\pi)^{d/2} \int_{\mathbb{R}^d} |\hat{F}(\omega)|^2\, d\omega = (2\pi)^{d/2} ||\hat{F}||_2^2 = (2\pi)^{d/2} ||F||_2^2, \end{aligned}$$

wobei der letzte Schritt wegen $F \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ folgt. Demnach beschreibt $|F_m(x-y) - p_y^{(m)}(x)|$ einen Interpolationsfehler, wie er von Theorem 3.2 abgedeckt wird, d.h. es gilt

$$|F_m(x-y) - p_y^{(m)}(x)| \leq c\lambda_m^{1/h_{X_k,X}} \|F\|_2.$$

□

Bemerkung 3.6. *Eine Einschränkung der Funktion F bzw. $\tilde{f}$ auf $X \times Y$ in Abschätzung* (3.8) *liefert ein Resultat für die Funktion f, d.h. es gilt*

$$|f(x-y) - p_y(x)| \leq (1 + \Lambda_k^{\Phi_m})||f - f_m||_{L^\infty(X\times Y)} + \tilde{c}\lambda_m^{1/h_{X_k,X}}||f||_{L^2(X\times Y)},$$

wobei $\tilde{c} = c(1 + \bar{c} + \hat{c})$ eine von σ und α abhängige Konstante bezeichnet mit

$$\bar{c} = \frac{d-2\alpha}{d}\left(\left(\frac{\sigma+\vartheta}{\sigma}\right)^{d-2\alpha} - 1\right)^{-1},$$
$$\hat{c} = \frac{d-2\alpha}{d}\left(\frac{(\sigma+2\vartheta)^d}{(\sigma+\vartheta)^{2\alpha}} - (\sigma+\vartheta)^{d-2\alpha}\right)\left((\sigma+\vartheta)^{d-2\alpha} - \sigma^{d-2\alpha}\right)^{-1}.$$

und $f_m = F_m|_{X\times Y}$.

Da wir erwarten können, dass sich die Fülldichte wie $h_{X_k,X} \sim k^{-1/d}$ verhält, konvergiert der 2. Summand aus Lemma 3.5 exponentiell bezüglich k. Aus Sicht der Approximationstheorie ist das Resultat aus Lemma 3.5 wegen des 1. Summanden nicht optimal. Da sich die Lebesgue-Konstante $\Lambda_k^{\Phi_m}$ in m stabil verhält, ist die Bilanzierung von $\Lambda_k^{\Phi_m}$ gegen $||F - F_m||_{L^\infty(\mathbb{R}^d)}$ für wachsende k und m entscheidend. Aufgrund der hier konstruierten Fortsetzung von f, welche in den Verbindungspunkten nur stetig ist, kann keine hohe Konvergenzgeschwindigkeit von $||F - F_m||_{L^\infty(\mathbb{R}^d)}$ bzgl. m erwartet werden. Die Konstruktion einer glatteren Fortsetzung sollte die Konvergenzgeschwindigkeit bzgl. m erhöhen, wobei das optimale Resultat eine analytische Fortsetzung wäre. Jedoch ist die analytische Fortsetzung von f eine noch offene Fragestellung. Auf der anderen Seite wird die Lebesgue-Konstante $\Lambda_k^{\Phi_m}$ je nach Punktkonfiguration ein Wachstumsverhalten in k zwischen $\log k$ und 2^k oder schlimmer aufweisen. Ferner hängt $\lambda_m \in (0,1)$ von m ab und strebt mit $m \to \infty$ gegen 1.

Für die Anwendung von Lemma 3.5 auf dem Restterm r_k bei der Kreuzapproximation bedeutet dies keinen Abbruch, falls die Genauigkeit für die Wahl eines festen Parameters m bilanziert wird. Hierfür können zwei Gründe genannt werden. Mit der Kreuzapproximation soll ein vorgegebener Fehler erreicht werden, der hauptsächlich durch den Diskretisierungsfehler bestimmt wird. Zudem sind wir an eher kleinen Clustern X interessiert, sodass sich die Zahl k in Grenzen hält. Diese Situation tritt im Übrigen auch bei der ursprünglichen Theorie unter Bezugnahme auf die Polynominterpolation ein. Auch hier ist man eher an kleineren Cluster interessiert.

Die Anwendung des letzten Lemmas auf den Restterm r_k für ein fest gewähltes m führt auf das folgende Resultat, um f auf $X \times Y$ zu interpolieren.

Theorem 3.7. *Für $y \in Y$ bezeichne p_y die Interpolante* (3.7) *basierend auf radialen Basisfunktionen für $f_y := f(\cdot, y) = |\cdot - y|^{-\alpha}$. Generiert die Auswahl der Punkte $y_1, ..., y_k$ Matrizen von quasi maximalem Volumen, d.h. sie erfüllt die Bedingung*

$$|\det C_k^{(i)}(y)| \leq c_M |\det C_k|, \quad 1 \leq i \leq k,\ y \in Y, \tag{3.9}$$

wobei $c_M > 1$ eine Konstante ist, so gilt, dass

$$|r_k(x,y)| \leq (c_M k + 1)\left((1 + \Lambda_k^{\Phi_m})||f - f_m||_{L^\infty(X\times Y)} + \tilde{c}\lambda_m^{1/h_{X_k,X}}||f||_{L^2(X\times Y)}\right), \tag{3.10}$$

wobei $X_k := \{x_1, \ldots, x_k\}$.

Beweis. Aus Abschnitt 1.2.1 wissen wir, dass der Restterm r_k die Darstellung

$$r_k(x,y) = f(x,y) - s_k(x,y),$$

besitzt, wobei $s_k(x,y) = v_k(x)^T C_k^{-1} w_k(y)$ und

$$C_k := \begin{bmatrix} f(x_1,y_1) & \dots & f(x_k,y_k) \\ \vdots & & \vdots \\ f(x_k,y_1) & \dots & f(x_k,y_k) \end{bmatrix} \in \mathbb{R}^{k\times k}, \quad v_k(x) := \kappa(x,[y]_k), \quad w_k(y) = \kappa([x]_k,y).$$

Sei der Vektor der Lagrange Funktionen $L_i^{\Phi_m}$, $i = 1,\dots,k$, bezogen auf die radialen Basisfunktionen Φ_m und die Punkte $x_1,\dots,x_k$ gegeben durch

$$L^{\Phi_m}(x) = \begin{bmatrix} L_1^{\Phi_m}(x) \\ \vdots \\ L_k^{\Phi_m}(x) \end{bmatrix}.$$

Unter Verwendung der Darstellung von r_k erhalten wir

$$\begin{aligned} r_k(x,y) &= f(x,y) - v_k(x)^T C_k^{-1} w_k(y) \\ &= f(x,y) - w_k(y)^T L^{\Phi_m}(x) - \left(v_k(x) - C_k^T L^{\Phi_m}(x)\right)^T C_k^{-1} w_k(y) \\ &= f_y(x) - p_y(x) - \sum_{i=1}^{k} (C_k^{-1} w_k(y))_i \left(f_{y_i}(x) - p_{y_i}(x)\right) \\ &= f_y(x) - p_y(x) - \sum_{i=1}^{k} \frac{\det C_k^{(i)}(y)}{\det C_k} \left(f_{y_i}(x) - p_{y_i}(x)\right), \end{aligned}$$

wobei die letzte Zeile aus der Cramerschen Regel folgt. Die Aussage folgt mit der Dreiecksungleichung und Lemma 3.5. □

Die Bedingung (3.9) ist in der Praxis nur sehr schwer zu überprüfen. Praktikabler ist die Auswahl der Punkte $y_1,\dots,y_k$ bezüglich der Bedingung

$$|r_{k-1}(x_k,y_k)| \geq |r_{k-1}(x_k,y)| \quad \text{für alle } y \in Y.$$

Jedoch führt diese Bedingung auf die schlechtere Fehlerschranke

$$|r_k(x,y)| \leq 2^k \left((1 + \Lambda_k^{\Phi_m}) ||f - f_m||_{L^\infty(X\times Y)} + \tilde{c}\lambda_m^{1/h_{X_k,X}} ||f||_{L^2(X\times Y)} \right).$$

Durch die Wahl des nächsten Punktes x_{k+1} kann die Konvergenz der Approximation auf X kontrolliert werden, indem die Fülldichte $h_{X_{k+1},X}$ von Schritt k zu Schritt $k+1$ minimiert wird. Dieses Minimierungsproblem kann effizient, d.h. in logarithmisch linearer Komplexität, zum Beispiel mit dem „Approximate Nearest Neighbor“ Vefahren gelöst werden, siehe [4, 5, 6].

Bisher haben wir in den Kapiteln 2 und 3 kontinuierliche Funktionen betrachtet. Im späteren Verlauf werden hauptsächlich Diskretisierungen von Funktionen und Operatoren im Fokus stehen, sodass wir uns um entsprechende analoge Techniken kümmern müssen. Dies führt uns in einem ersten Schritt zu Matrizen von niedrigem Rang.

4. Niedrigrang-Matrizen

In diesem Kapitel betrachten wir nicht mehr den Operator selbst, sondern seine diskretisierte Form, was im Fall von nicht-lokalen Operatoren eine voll-besetzte Systemmatrix zur Folge hat. Das Ziel besteht darin, eine effiziente Darstellung der Systemmatrix zu finden. Hierbei wird die Approximation durch Niedrigrang-Matrizen hilfreich sein. Im Fokus liegt daher eine Matrix $A \in \mathbb{R}^{M\times N}$ mit der Darstellung

$$A = \Lambda_1 \mathcal{A} \Lambda_2^*, \tag{4.1}$$

wobei der nicht lokale Operator $\mathcal{A}$ linear von einer bivariaten Funktion $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ abhängt. Im Falle eines beschränkten Gebietes $\Omega \subset \mathbb{R}^d$ mit Lipschitz-Rand dient

$$(\mathcal{A}v)(x) = \int_\Omega \kappa(x,y)v(y)\,\mathrm{d}\mu_y, \quad x \in \Omega,$$

als Prototyp, siehe Kapitel 2. Dabei bezeichnet μ das zugehörige Maß. Die beiden Operatoren $\Lambda_1 : L^2(\Omega) \to \mathbb{R}^N$ und $\Lambda_2 : L^2(\Omega) \to \mathbb{R}^M$ seien linear. Der adjungierte Operator $\Lambda_2^* : \mathbb{R}^M \to L^2(\Omega)$ ist durch

$$(\Lambda_2^* z, f)_{L^2(\Omega)} = z^T(\Lambda_2 f), \quad z \in \mathbb{R}^M, \quad f \in L^2(\Omega),$$

definiert. Die beiden gerade aufgestellten Operatoren Λ_1 und Λ_2 dienen zur Beschreibung von Diskretisierungen. Zwei Beispiele sind:

1. Galerkin Methode: Die Wahl von Funktionen φ_i, $i = 1, ..., N$, und ψ_j, $j = 1, ..., M$, resultiert in der Diskretisierung

$$(\Lambda_1 f)_i = \int_\Omega f(x)\varphi_i(x)\,\mathrm{d}\mu_x \quad \text{und} \quad (\Lambda_2 f)_j = \int_\Omega f(x)\psi_j(x)\,\mathrm{d}\mu_x. \tag{4.2}$$

2. Kollokation: Die Wahl von Punkten y_j, $j = 1, ..., M$, und Funktionen φ_i, $i = 1, ..., N$, führt auf die Diskretisierung

$$(\Lambda_1 f)_i = f(y_i) \quad \text{und} \quad (\Lambda_2 f)_j = \int_\Omega f(x)\varphi_j(x)\,\mathrm{d}\mu_x.$$

Niedrigrang-Matrizen sind stark mit der Approximation von bivariaten Kernfunktionen κ durch degenerierte Funktionen $\tilde{\kappa}$, wie wir es in Kapitel 2 kennengelernt haben, verbunden. Diskretisiert man den approximierten Kern

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

mit

$$a_l = \Lambda_1 u_l \in \mathbb{R}^M \quad \text{und } b_l = \Lambda_2 v_l \in \mathbb{R}^N, \quad l = 1, ..., k,$$

so führt dies automatisch zu einer Matrix

$$\tilde{A} = \Lambda_1 \tilde{\mathcal{A}} \Lambda_2^* = \Lambda_1 \sum_{l=1}^{k} u_l b_l^T = \sum_{l=1}^{k} (\Lambda_1 u_l) b_l^T = \sum_{l=1}^{k} a_l b_l^T$$

von Rang k, wobei $\tilde{\mathcal{A}}$ definiert ist durch $(\tilde{\mathcal{A}}v)(x) := \int_\Omega \tilde{\kappa}(x,y)v(y)\,\mathrm{d}\mu_y$. Die Umkehrung dieser Beziehung ist im allgemeinen nicht richtig. Eine Niedrigrang-Matrix ist wie folgt definiert, siehe auch [15].

Definition 4.1. *Matrizen $A \in \mathbb{R}^{M\times N}$ vom Rang k heißen Niedrigrang-Matrizen, falls*

$$k(M+N) < MN, \quad M, N \in \mathbb{N}.$$

Der Vorteil von Niedrigrang-Matrizen liegt in den reduzierten Speicheranforderungen und den schneller auszuführenden Matrix-Rechenoperationen. Ersteres ist eine Folge der effizienten Darstellungsmöglichkeit. Mit der äußeren Produktform, d.h. es existieren Matrizen $U \in \mathbb{R}^{M\times k}$ und $V \in \mathbb{R}^{N\times k}$ mit

$$A = UV^T = \sum_{l=1}^{k} u_l v_l^T,$$

wobei u_l und v_l, $l = 1, ..., k$, die Spalten von U und V bezeichnen, lässt sich A bzw. die Approximation an A mit $k(M+N)$ anstelle von MN Einheiten speichern. Die äußere Produktform hat auch Auswirkungen auf die Multiplikation einer Niedrigrang-Matrix mit einem Vektor. Hier kann die Anzahl an arithmetischen Operationen von $2MN$ auf $2k(M+N)-k$ reduziert werden, siehe z.B. [15]. Auch einige typische Normen, wie die Frobenius Norm oder die Spektralnorm profitieren von der äußeren Produktdarstellung. Die Frobeniusnorm kann wegen

$$\|A\|_F^2 = \|UV^T\|_F^2 = \sum_{i,j=1}^{k} (u_i^T u_j)(v_i^T v_j)$$

mit $2k^2(M+N)$ Operationen berechnet werden. Bezeichnet $\rho(A)$ den Spektralradius von A, so ist die Anzahl der Rechenoperationen für die Spektralnorm

$$\|A\|_2 = \sqrt{\rho(U^T U V^T V)}$$

von gleicher Größenordnung.

In den im weiteren Verlauf dieser Arbeit entwickelten Algorithmen wird die Addition und Multiplikation von Niedrigrang-Matrizen auftauchen. Eine effiziente Behandlung dieser Matrix-Matrix Rechenoperationen ist demnach essentiell, um effiziente Algorithmen zu erhalten. Daher werfen wir einen kurzen Blick auf die effiziente Addition bzw. Multiplikation von Niedrigrang-Matrizen, siehe auch [15]. Seien also $A = U_A V_A^T \in \mathbb{R}^{M\times r}$ und $B = U_B V_B^T \in \mathbb{R}^{r\times N}$ zwei Matrizen von Rang k_A bzw. k_B in äußerer Produktform. Dann gibt es zwei Möglichkeiten, die Matrix-Matrix-Multiplikation $AB = UV^T$ zu realisieren, welche je nach k_A, k_B, M und N besser oder schlechter geeignet ist. Für die erste Variante setze

$$V := V_B \quad \text{und} \quad U := U_A(V_A^T U_B),$$

was $2k_A k_B(M+r) - k_B(M+k_A)$ arithmetische Operationen erfordert. In der zweiten Variante werden durch das Setzen

$$V := V_B(U_B^T V_A) \quad \text{und} \quad U := U_A$$

insgesamt $2k_A k_B(r+N) - k_A(N+k_B)$ Operationen benötigt. Der Rang des Produkts AB ist hierbei durch $\min\{k_A, k_B\}$ beschränkt.

Bei der Addition zweier Niedrigrang-Matrizen lässt sich der Rang des Resultats $A+B$ nicht exakt angeben. Hier gilt für die Summe zweier Matrizen $A + B = UV^T$ mit $U := [U_A, U_B] \in \mathbb{R}^{M\times(k_A+k_B)}$ und $V := [V_A, V_B] \in \mathbb{R}^{N\times(k_A+k_B)}$, dass

$$\operatorname{rank}(A+B) \le k_A + k_B.$$

Eine Erhöhung des Rangs bei der Addition zweier Matrizen lässt sich leicht nachvollziehen. Jedoch wird in der Praxis aufgrund innerer Abhängigkeiten der Rang der Summe zumeist viel kleiner sein als die Summe der Ränge von A und B.

4.1 Approximation mit Niedrigrang-Matrizen

In den Anwendungen kommen Niedrigrang-Matrizen häufig nur als Approximationen an die eigentliche Matrix vor. Denn viele Matrizen von vollem Rang können oft durch Matrizen von niedrigerem Rang approximiert werden. Die beste Niedrigrang- bzw. Rang-k-Approximation einer Matrix $A \in \mathbb{R}^{M\times N}$, $N \leq M$, ist gegeben durch die Singulärwertzerlegung $A = U\Sigma V^T$ mit $U^TU = I_N = V^TV$ und einer Diagonalmatrix $\Sigma \in \mathbb{R}^{N\times N}$ mit den Einträgen $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_N \geq 0$, siehe [35].

Theorem 4.2. *Sei $A = U\Sigma V^T$ eine Singulärwertzerlegung von $A \in \mathbb{R}^{M\times N}$, $M \geq N$. Dann gilt für $k \in \mathbb{N}$ mit $k \leq N$, dass*

$$\min_{\text{Rang } G\leq k} ||A - G|| = ||A - A_k|| = ||\Sigma - \Sigma_k||,$$

wobei $A_k := U\Sigma V^H$ und $\Sigma_k := \text{diag}(\sigma_1, ..., \sigma_k, 0, ..., 0) \in \mathbb{R}^{n\times n}$ Matrizen vom Rang höchstens k sind.

Ein Vorteil der Singulärwertzerlegung (siehe [35, 15]) hierbei ist, dass wir mit den Abschätzungen

$$\|A - A_k\|_F \leq \sqrt{\sum_{l=k+1}^{N} \sigma_l^2}$$

bzw.

$$\|A - A_k\|_2 \leq \sigma_{k+1},$$

wenn die Spektralnorm betrachtet wird, Informationen über den Fehler der Approximation erhalten. So kann z.B. bei vorgegebener relativer Genauigkeit $\varepsilon > 0$ mit

$$\|A - A_k\|_2 \leq \varepsilon\|A\|_2$$

der kleinstmögliche Rang

$$k(\varepsilon) = \min\{k \in \mathbb{N} \ : \ \sigma_{k+1} \leq \varepsilon\sigma_k\}$$

angegeben werden, um diese Genauigkeit ε zu erreichen.

Da sich der Rest dieser Arbeit vorrangig mit einer anderen Approximationstechnik beschäftigt, werden wir hier nicht näher auf die Singulärwertzerlegung eingehen. Weitere Eigenschaften auch im Hinblick auf Niedrigrang-Matrizen wie zum Beispiel eine approximative Variante der Addition zweier Niedrigrang-Matrizen können in [15] gefunden werden.

4.2 Die adaptive Kreuzapproximation (ACA)

Die Singulärwertzerlegung liefert zwar optimale Resultate bzgl. des Rangs der Approximation bei vorgegebener Genauigkeit, jedoch ist sie aufgrund der hohen Komplexität für hoch bzw. höher dimensionale Problemstellungen nicht geeignet. Die folgende adaptive Kreuzapproximation (ACA) stellt eine effiziente Möglichkeit dar, voll besetzte Matrizen zu approximieren, und ist in gewisser Weise das algebraische Analogon zur Kreuzapproximation aus Kapitel 2. Im Gegensatz dazu wird beim ACA nicht die Kernfunktion des Integraloperators approximiert. Vielmehr wird direkt auf dem diskretisierten Operator, also auf der Matrix selbst gearbeitet, wobei für die Approximation nur wenige Originaleinträge nötig sind. Zudem muss die gesamte Matrix nicht zu Beginn aufgestellt werden. Wie bei der Kreuzapproximation auch muss für die Kernfunktion angenommen werden, dass sie bezüglich mindestens einer Variable asymptotisch glatt ist. Auch wenn die Kernfunktion nicht bekannt sein muss und nur auf der Matrix gearbeitet werden kann, müssen wir an dieser Stelle zumindest wissen, dass die zugrunde liegende Kernfunktion in dieser Klasse von Funktionen liegt. In Abschnitt 4.2.1 soll zunächst der Algorithmus formuliert und anschließend dessen Konvergenz untersucht werden.

4.2.1 Formulierung des Algorithmus

Gegeben ist eine Matrix $A \in \mathbb{R}^{M\times N}$ der Form (4.1). Um zu gewährleisten, dass die asymptotische Glattheit erfüllt ist, betrachten wir einen einzelnen Block $A_b \in \mathbb{R}^{m\times n}$, $m \leq M$, $n \leq N$, von A, sodass die Träger $X = \operatorname{supp} \Lambda_1$ und $Y = \operatorname{supp} \Lambda_2$ die Abstandsbedingung (2.1) erfüllen. Dabei wird ein ähnlicher Ansatz wie bei der Kreuzapproximation in Kapitel 1 verfolgt, siehe z.B. [15].

Wir starten mit $R_0 := A_b$. Zur Berechnung von R_k wird ein Pivotelement (i_k, j_k) in R_{k-1} gesucht und von R_{k-1} ein skaliertes äußeres Produkt subtrahiert, d.h.

$$R_k := R_{k-1} - \frac{1}{(R_{k-1})_{i_k j_k}} (R_{k-1})_{1:m,j_k} (R_{k-1})_{i_k,1:n}.$$

Die Notation $(R_{k-1})_{1:m,j_k}$ bezeichnet die j-te Spalte von R_{k-1}. Der Ausdruck $(R_{k-1})_{i,1:n}$ ist in analoger Art zu verstehen. Dieser Idee folgend konstruieren wir zwei Folgen von Vektoren $u_k \in \mathbb{R}^m$ und $v_k \in \mathbb{R}^n$ für $k = 1, 2, 3, ...$ mit Hilfe des folgenden Algorithmus 1. Als Kriterium für die Terminierung des Algorithmus wird ein Ausdruck bestehend aus einer vorgegebenen Genauigkeit $\varepsilon_{\mathrm{ACA}}$ und dem Parameter η aus der Abstandsbedingung (2.1) verwendet.

Algorithmus 1 Adaptive Kreuzapproximation (engl.: Adaptive Cross Approximation, ACA)

Let $k = 1$; $Z = \emptyset$; $\varepsilon_{\mathrm{ACA}} > 0$.
repeat
  Finde i_k mit einer geeigneten Regel
  $\tilde{v}_k := A_{i_k,1:n}$.
  for $l = 1, ..., k-1$ **do** $\tilde{v}_k := \tilde{v}_k - (u_l)_{i_k} v_l$.
  end for
  $Z := Z \cup \{i_k\}$
  if $\tilde{v}_k \neq 0$ **then**
    $j_k := \operatorname{argmax}_{j\in s} |(\tilde{v}_k)_j|$; $v_k := (\tilde{v}_k)_{j_k}^{-1} \tilde{v}_k$.
    $u_k := A_{1:m,j_k}$.
    for $l = 1, \ldots, k-1$ **do** $u_k := u_k - (v_l)_{j_k} u_l$.
    end for
    $k := k + 1$.
  end if
until $\|u_{k+1}\|_2 \|v_{k+1}\|_2 \leq \frac{\varepsilon_{\mathrm{ACA}}(1-\eta)}{1+\varepsilon_{\mathrm{ACA}}} \|S_k\|_F$ or $Z = t$.

Mit diesen Folgen von Vektoren lässt sich eine eine Matrix

$$S_k := \sum_{l=1}^{k} u_l v_l^T$$

definieren, welche höchstens Rang k besitzt und unter den bisher getroffenen Annahmen eine Approximation an $A_b = R_k + S_k$ mit Fehler $R_k = A_b - S_k$ darstellt. Die rekursive Form der beiden Folgen

$$u_k = (R_{k-1})_{1:m,j_k} = A_{1:m,j_k} - \sum_{l=1}^{k-1} \frac{(\tilde{v}_l)_{j_k}}{(\tilde{v}_l)_{j_l}} u_l$$

und

$$\tilde{v}_k = (R_{k-1})_{i_k,1:n}^T = A_{i_k,1:n}^T - \sum_{l=1}^{k-1} \frac{(u_l)_{i_k}}{(\tilde{v}_l)_{j_l}} \tilde{v}_l$$

zeigt, dass nur wenige Originaleinträge von A nötig sind.

Das in Algorithmus 1 verwendete Abbruchkriterium

$$\|u_{k+1}\|_2\|v_{k+1}\|_2 \leq \frac{\varepsilon_{\mathrm{ACA}}(1-\eta)}{1+\varepsilon_{\mathrm{ACA}}}\|S_k\|_F$$

beruht auf der Annahme

$$\|R_{k+1}\|_F \leq \eta\|R_k\|_F.$$

Denn damit folgt

$$\begin{aligned}\|R_k\|_F &\leq \|R_{k+1}\|_F + \|u_{k+1}v_{k+1}^T\|_F \\ &\leq \eta\|R_k\|_F + \|u_{k+1}\|_2\|v_{k+1}\|_2,\end{aligned}$$

womit wir durch

$$\begin{aligned}\|R_k\|_F &\leq \frac{1}{1-\eta}\|u_{k+1}\|_2\|v_{k+1}\|_2 \\ &\leq \frac{\varepsilon_{\mathrm{ACA}}}{1+\varepsilon_{\mathrm{ACA}}}\|S_k\|_F \\ &\leq \frac{\varepsilon_{\mathrm{ACA}}}{1+\varepsilon_{\mathrm{ACA}}}\left(\|A\|_F + \|R_k\|_F\right)\end{aligned}$$

und

$$\varepsilon_{\mathrm{ACA}}\|A\|_F \geq (1+\varepsilon_{\mathrm{ACA}})\left(\|R_k\|_F - \frac{\varepsilon_{\mathrm{ACA}}}{1+\varepsilon_{\mathrm{ACA}}}\|R_k\|_F\right) = (1+\varepsilon_{\mathrm{ACA}})\,(1+\varepsilon_{\mathrm{ACA}})^{-1}\,\|R_k\|_F = \|R_k\|_F$$

letztendlich einen relativen Approximationsfehler $\varepsilon_{\mathrm{ACA}}$ erhalten.

Während die Wahl von j_k durch die Bedingung

$$|(R_{k-1})_{i_k j_k}| = \max_{j=1,\ldots,n} |(R_{k-1})_{i_k j}|$$

klar festgelegt ist, gibt es bei der Wahl von i_k mehr Möglichkeiten. In [15] wird beispielsweise eine Selektion von i_k diskutiert. Die Menge Z in Algorithmus 1 sammelt hierbei alle verschwindenden Zeilen der Matrix R_k. Die Zeilenindizes i_k müssen so gewählt werden, dass die zum System gehörende Vandermonde-Matrix, in welcher der Approximationsfehler abgeschätzt wird, nicht singulär ist, siehe Kapitel 1. Falls die i_k-te Zeile von R_{k-1} nicht Null ist und daher als v_k genutzt wird (nach einer Skalierung von $\tilde{v}_k$ mit $(\tilde{v}_k)_{j_k}^{-1}$), wird sie auch zu Z hinzugefügt, da die i_k-te Zeile des folgenden Rests R_k auch verschwindet. Es kann gezeigt werden, dass der numerische Aufwand bei ACA von der Ordnung $|Z|^2(|t|+|s|)$ ist und die Anzahle der Schritte $|Z|$ typischerweise logarithmisch von der gewünschten blockweisen Genauigkeit $\varepsilon_{\mathrm{ACA}}$ abhängt.

Wenden wir die Techniken aus Kapitel 3 an, so genügt es, bei der Punktauswahl die Fülldichte des zugrunde liegenden Clusters zu minimieren. Entscheidend für die numerische Stabilität hierbei ist der Separationsabstand der zugrunde liegenden Punktmenge, siehe Bemerkung 3.3.

4.2.2 Fehleranalyse

Um die Konvergenz der ACA zu analysieren, muss der Restterm R_k abgeschätzt werden. Zunächst wird, wie bisher auch, angenommen, dass die betrachtete Kernfunktion κ bzgl. der Variable y asymptotisch glatt ist. Wir halten uns bei der Konvergenzanalyse an die Ausführungen in [15]. Je nachdem

welche Diskretisierung angewendet wird, werden unterschiedliche Abschätzungen erzielt. Am Ende dieses Abschnitts wird dies kurz im Fall von Kollokations- und Galerkin-Matrizen thematisiert.

Aus Gründen der Einfachheit in der folgenden Analyse und ohne Beschränkung der Allgemeinheit seien die Pivot-Indizes i_l und j_l festgelegt als $i_l = j_l = l$, $l = 1, ..., k$. Mit der dadurch ermöglichten Zerlegung der Matrix

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad A_{11} \in \mathbb{R}^{k \times k},$$

besitzt der Rest R_k die folgende Darstellung.

Lemma 4.3. *Es gilt*

$$R_k = A - \begin{pmatrix} A_{11} \\ A_{21} \end{pmatrix} A_{11}^{-1} \begin{pmatrix} A_{11} & A_{12} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & C_k \end{pmatrix},$$

wobei $C_k = A_{22} - A_{21}A_{11}^{-1}A_{12}$ das Schur Komplement von A_{11} in A bezeichnet.

Beweis. Der zweite Teil der Behauptung sowie der erste Teil der Behauptung für $k = 1$ sind klar. Für $k \geq 1$ verwenden wir die Zerlegung

$$A = \begin{pmatrix} A_{11} & w & B \\ v^T & \alpha & y^T \\ C & x & D \end{pmatrix}, \quad A_{11} \in \mathbb{R}^{k \times k},$$

mit den Vektoren $x \in \mathbb{R}^{m-k-1}$, $y \in \mathbb{R}^{n-k-1}$, $v, w \in \mathbb{R}^k$ und $\alpha \in \mathbb{R}$. Der Bildungsvorschrift von R_{k+1} folgend erhalten wir mit dem Pivot Element $\alpha - v^T A_{11}^{-1} w$ und mit $\gamma = (\alpha - v^T A_{11} w)^{-1}$, dass

$$\begin{aligned} R_{k+1} &= R_k - [(R_k)_{i_k j_k}]^{-1} (R_k)_{1:m, j_k} (R_k)_{i_k, 1:n} \\ &= A - \begin{pmatrix} A_{11} & w \\ v^T & \alpha \\ C & x \end{pmatrix} \begin{pmatrix} A_{11}^{-1} + \gamma A_{11}^{-1} w v^T A_{11}^{-1} & -\gamma A_{11}^{-1} w \\ -\gamma v^T A_{11}^{-1} & \gamma \end{pmatrix} \begin{pmatrix} A_{11} & w & B \\ v^T & \alpha & y^T \end{pmatrix} \\ &= A - \begin{pmatrix} A_{11} & w \\ v^T & \alpha \\ C & x \end{pmatrix} \begin{pmatrix} A_{11} & w \\ v^T & \alpha \end{pmatrix}^{-1} \begin{pmatrix} A_{11} & w & B \\ v^T & \alpha & y^T \end{pmatrix}, \end{aligned}$$

wodurch der erste Teil der Behauptung auch für $k \geq 1$ folgt. □

In Algorithmus 1 können nicht nur in den ausgewählten Indices Nullzeilen entstehen. Es kann demnach passieren, dass die eigentliche Anzahl an Nullzeilen k', entstanden durch Pivotisierung oder nicht, größer als die Rangzahl k ist. Wie wir in den nächsten Lemmata sehen werden, bestimmt nicht der eigentlich Rang k die Genauigkeit der Approximation sondern k', da wegen der Bildungsvorschrift der Restterme R_k Nullzeilen erhalten bleiben, d.h. Nullzeilen von R_l sind auch Nullzeilen von R_k für alle $k \geq l$. Nach Umsortierung speichern wir die ausgewählten Indices in $\{1, ..., k\}$ und die zusätzlichen Nullzeilen in $\{k+1, ..., k'\}$, womit sich der untere Teil von A durch

$$A_{21} = \begin{pmatrix} \hat{A}_{21} \\ \check{A}_{21} \end{pmatrix}, \quad A_{22} = \begin{pmatrix} \hat{A}_{22} \\ \check{A}_{22} \end{pmatrix}$$

mit $\hat{A}_{21} \in \mathbb{R}^{(k'-k) \times k}$ und $\hat{A}_{22} \in \mathbb{R}^{(k'-k) \times (n-k)}$ nochmals unterteilen lässt. Um den Fehler weiter analysieren zu können, muss die Matrix C_k näher betrachtet werden, siehe auch [15].

Lemma 4.4. *Sei $X \in \mathbb{R}^{(m-k)\times k'}$ beliebig. Dann gilt:*

$$C_k = \left(A_{22} - X \begin{pmatrix} A_{12} \\ \hat{A}_{22} \end{pmatrix}\right) - \left(A_{21} - X \begin{pmatrix} A_{11} \\ \hat{A}_{21} \end{pmatrix}\right) A_{11}^{-1} A_{12}.$$

Beweis. Zunächst gilt

$$(R_k)_{i,1:n} = 0, \quad i = k+1, ..., k',$$

und mit Lemma 4.3, dass

$$\hat{A}_{22} = \hat{A}_{21} A_{11}^{-1} A_{12}.$$

Durch Addition und Subtraktion von

$$X \begin{bmatrix} A_{12} \\ \hat{A}_{22} \end{bmatrix} = X \begin{bmatrix} A_{11} \\ \hat{A}_{21} \end{bmatrix} A_{11}^{-1} A_{12}$$

von und zu C_k folgt die Behauptung. □

Der folgende Teil der Fehlerabschätzung gestaltet sich ähnlich wie die Abschätzung des Restterms in Kapitel 1. Wir konzentrieren uns zunächst auf den Term $A_{11}^{-1}A_{12}$ in C_k.

Lemma 4.5. *Sei j_k in jedem Schritt so gewählt, dass die Bedingung*

$$|(R_{k-1})_{i_k,j_k}| = \max_{j=1,...,n} |(R_{k-1})_{i_k,j}|.$$

erfüllt ist. Dann gilt für $i = 1, ..., k$ und $j = 1, ..., n-k$, dass

$$|(A_{11}^{-1}A_{12})_{ij}| = \frac{|\det(a_1, ..., a_{i-1}, a'_j, a_{i+1}, ..., a_r)|}{|\det A_{11}|} \leq 2^{k-i},$$

wobei a_l, $l = 1, ..., k$, die Spalten von A_{11} bezeichnen und a'_j die j-te Spalte von A_{12} ist.

Beweis. Der Beweis erfolgt analog zu der Abschätzung bzgl. der Bedingung (1.5). □

Die restlichen Terme der Matrix C_k lassen sich im Bezug auf Interpolationsfehler abschätzen, vgl. Kapitel 2.

Um den Fehler bei ACA endgültig bestimmen zu können, muss ein genauerer Blick auf die gewählte Diskretisierung geworfen werden. Je nach Art der Diskretisierung erhalten wir unterschiedliche Faktoren zusätzlich zum Bestapproximationsfehler. Wir starten mit den Kollokationsmatrizen. Verwenden wir dabei ein allgemeines Funktionensystem Ξ, so sollte, damit das auftretende Interpolationsproblem wohlgestellt ist, die Unisolvenz des Systems Ξ in den Punkten x_i, $i = 1, ..., k'$ erfüllt sein. Dementsprechend nehmen wir an, dass die Bedingung

$$\det\left[\xi_j(x_i)\right]_{i,j=1,...,k'} \neq 0, \tag{4.3}$$

welche unter anderem durch die Wahl der Zeilen i_k realisiert werden kann, erfüllt ist. Werden die Techniken aus Kapitel 3, d.h. Approximation mit Exponentialsummen und Interpolation mittels radialer Basisfunktionen, angewendet, so entfällt die Bedingung (4.3), wobei dies am Ende zu einer spezifischeren Abschätzung führt.

Lemma 4.6. *Sei $(\Lambda_1 f)_i = f(y_i)$, $i = 1, ..., m$. Dann gilt für $i = 1, ..., m$ und $j = 1, ..., n$, dass*

$$|(R_k)_{ij}| \leq 2^k (1 + ||\mathcal{I}_{k'}^{\Xi}||) \max_{j=1,...,n} \inf_{p \in span\ \Xi} ||\mathcal{A}\Lambda_{2,j}^* - p||_\infty,$$

wobei $\Xi := \{\xi_1, ..., \xi_{k'}\}$ ein geeignetes System von Funktionen darstellt.

Beweis. Mit der Wahl von $X_{il} = L_l^{\Xi}(x_i)$ erhalten wir für die Einträge der Matrix $A_{22} - X \begin{pmatrix} A_{12} \\ \hat{A}_{22} \end{pmatrix} \in \mathbb{R}^{m-k \times n-k}$ die Darstellung

$$(A_{22} - X \begin{pmatrix} A_{12} \\ \hat{A}_{22} \end{pmatrix})_{ij} = \mathcal{A}\Lambda_{2,j}^*(x_j) - \sum_{l=1}^{k'} L_l^{\Xi}(y_i)\mathcal{A}\Lambda_{2,j}^*(x_l).$$

Im Hinblick auf (2.4) gilt die Abschätzung

$$|(R_k)_{ij}| \leq 2^k(1 + \|\mathcal{I}_{k'}^{\Xi}\|) \max_{j=1,\dots,N} \inf_{p \in \operatorname{span} \Xi} \|\mathcal{A}\Lambda_{2,j}^* - p\|_\infty,$$

wobei die Abschätzung für die Matrix $A_{21} - X \begin{pmatrix} A_{11} \\ \hat{A}_{21} \end{pmatrix}$ in analoger Weise zu $A_{22} - X \begin{pmatrix} A_{12} \\ \hat{A}_{22} \end{pmatrix}$ folgt. □

Die Galerkin-Diskretisierung, siehe (4.2), ist technisch aufwändiger, da in diesem Fall keine Punktauswertungen sondern höchstens gemittelte Werte vorliegen. Aus diesem Grund werden wir zunächst Funktionen unter einer verallgemeinerten Unisolvenzbedingung konstruieren, welche im Mittel für alle Funktionen $q \in \operatorname{span} \Xi$ verschwinde.

Lemma 4.7. *Sei die Bedingung*

$$\det\, [(\Lambda_1 \xi_j)_i]_{i,j=1,\dots,k'} \neq 0$$

erfüllt. Dann existieren für jedes $i \in \{1, \dots, m\}$ eindeutig definierte Koeffizienten $c_l^{(i)}$, $l = 1, \dots, k'$, sodass

$$\int_\Omega \left(\frac{\varphi_i}{\|\varphi_i\|_{L^1}} - \sum_{l=1}^{k'} c_l^{(i)} \frac{\varphi_l}{\|\varphi_l\|_{L^1}} \right) q \, \mathrm{d}\mu = 0$$

für alle $q \in \operatorname{span} \Xi$ erfüllt ist.

Beweis. Für jede Funktion $q \in \operatorname{span} \Xi$ existieren Koeffizienten α_j, $j = 1, \dots, k'$, sodass

$$q = \sum_{j=1}^{k'} \alpha_j \xi_j$$

gilt. Das lineare Gleichungssystem

$$\sum_{l=1}^{k'} c_l^{(i)} \frac{(\Lambda_1 \xi_j)_l}{\|\varphi_l\|_{L^1}} = \frac{(\Lambda_1 \xi_j)_i}{\|\varphi_i\|_{L^1}}, \quad j = 1, \dots, k',$$

ist dank der Voraussetzung bezogen auf $c_l^{(i)}$, $l = 1, \dots, k'$, für alle $i \in \{1, \dots, m\}$ eindeutig lösbar. Mit der Linearität des Operators Λ_1 folgt die Behauptung. □

Mit Hilfe der in Lemma 4.7 konstruierten Funktionen lässt sich der Fehler der ACA auch für Galerkin-Diskretisierungen abschätzen.

Lemma 4.8. *Im Fall der Galerkin-Diskretisierung sei $\Lambda_{1,i}$ definiert durch*

$$(\Lambda_1 f)_i = \sum_\Omega f(x)\varphi_i(x) \, \mathrm{d}\mu_x, \quad i = 1, \dots, m.$$

Dann gilt für $i = 1, ..., m$ und $j = 1, ..., n$, dass

$$|(R_k)_{ij}| \leq 2^k \left(1 + \sum_{l=1}^{k'} |c_l^{(i)}|\right) \|\varphi_i\|_{L^1} \max_{j=1,...,n} \inf_{p \in \operatorname{span} \Xi} \|\mathcal{A}\Lambda_{2,j}^* - p\|_\infty.$$

gilt.

Beweis. Sei $q_j \in \operatorname{span} \Xi$ mit

$$\|\mathcal{A}\Lambda_{2,j}^* - q_j\|_\infty = \inf_{p \in \operatorname{span} \Xi} \|\mathcal{A}\Lambda_{2,j}^* - p\|_\infty.$$

Dann existieren nach Lemma 4.7 Koeffizienten $c_l^{(i)}$, sodass die Bedingung

$$\int_\Omega \left(\frac{\varphi_i}{\|\varphi_i\|_{L^1}} - \sum_{l=1}^{k'} c_l^{(i)} \frac{\varphi_l}{\|\varphi_l\|_{L^1}}\right) q \, \mathrm{d}\mu = 0$$

erfüllt ist. Definieren wir die Matrix X aus Lemma 4.4 durch die Einträge $X_{il} := \frac{\|\varphi_i\|_{L^1}}{\|\varphi_l\|_{L^1}} c_l^{(i)}$ für alle $i = 1, ..., m - k$ und alle $l = 1, ..., k'$, so erhalten wir

$$\begin{aligned}
\left(A_{22} - X \begin{bmatrix} A_{12} \\ \hat{A}_{22} \end{bmatrix}\right)_{ij} &= \int_\Omega \mathcal{A}\Lambda_{2,j}^*(x)\varphi_i(x) \, \mathrm{d}\mu_x - \sum_{l=1}^{k'} \frac{\|\varphi_i\|_{L^1}}{\|\varphi_l\|_{L^1}} c_l^{(i)} \int_\Omega \mathcal{A}\Lambda_{2,j}^*(x)\varphi_l(x) \, \mathrm{d}\mu_x \\
&= \int_\Omega [\mathcal{A}\Lambda_{2,j}^*(x) - q_j(x)]\varphi_i(x) \, \mathrm{d}\mu_x - \sum_{l=1}^{k'} \frac{\|\varphi_i\|_{L^1}}{\|\varphi_l\|_{L^1}} c_l^{(i)} \int_\Omega [\mathcal{A}\Lambda_{2,j}^*(x) - q_j(x)]\varphi_l(x) \, \mathrm{d}\mu_x \\
&\leq \left(1 + \sum_{l=1}^{k'} |c_l^{(i)}|\right) \|\varphi_i\|_{L^1} \inf_{p \in \operatorname{span} \Xi} \|\mathcal{A}\Lambda_{2,j}^* - p\|_\infty.
\end{aligned}$$

Auf analoge Art folgt dieselbe Abschätzung für den zweiten Teil in Lemma 4.4 und schließlich mit Lemma 4.5 die Behauptung. □

Unter Berücksichtigung von Kapitel 3 erhält man durch die explizite Wahl eines Funktionensystems, mit dem die Voraussetzung von Lemma 4.7 automatisch erfüllt ist, wiederum eine spezifischere Abschätzung in Lemma 4.8.

Kapitel 4 hat die Approximation eines einzelnen Blocks A_b einer Matrix A durch ein Niedrigrang-Matrix thematisiert. Das Ziel jedoch ist, nicht nur eine effiziente Darstellung eines einzelnen Blocks zu finden, sondern auch eine effiziente Darstellung für die gesamte Matrix. Hierzu muss zunächst eine geeignete Partitionierung der Matrix A konstruiert werden. Eine in diesem Sinne „geeignete“ Aufteilung der Matrix A wird Blöcke enthalten, die entweder in ihrer Dimension vergleichbar klein sind oder mit Niedrigrang-Matrizen approximiert werden können.

5. Matrix Partitionierung

Der in Abschnitt 4.2 vorgestellte Algorithmus soll nun auf die gesamte Matrix A angewendet werden. Da ACA aber auf Teilmatrizen bzw. Blöcken A_b von A arbeitet, benötigen wir eine Unterteilung der gesamten Matrix in Blöcke. Die Generierung der Blockstruktur erfolgt in einer hierarchischen Weise mit Clusterbäumen und Block-Clusterbäumen. Die hierarchischen Matrizen können anschließend auf Basis der Partitionierung der Matrix A in Blöcke definiert werden.
Soll ACA auf die Blöcke einer Matrix angewendet werden, welche durch die Diskretisierung einer singulären Kernfunktion ensteht, so ist das nicht bei allen Blöcken möglich, da diese nicht das geforderte Verhalten bzgl. der Singulärwerte des jeweiligen Blockes aufweisen. Eine derartige Problematik kann mit einer Zulässigkeitsbedingung bei der Partitionierung abgefangen werden.

5.1 Partitionierungen und die Zulässigkeitsbedingung

Im Folgenden wird eine Möglichkeit angeführt, Matrixindices $I := \{1, ..., M\} \times J := \{1, ..., N\}$, welche die Zeilen- bzw. Spaltenindices der betrachteten Matrix bezeichnen, in geeigneter Weise zu zerlegen, damit eine Matrix $A \in \mathbb{C}^{M \times N}$ mit Niedrigrang-Matrizen approximiert werden kann, siehe auch [15]. Unser Interesse liegt dabei nicht in der Konstruktion einer optimalen Partition, da dies einen zu großen Aufwand bedeuten würde. Vielmehr wird eine Partition angestrebt, welche mit nahezu linearer Komplexität berechnet werden kann und eine Approximation mit logarithmisch-linearer Komplexität ermöglicht.

Definition 5.1. *Seien $I, J \subset \mathbb{N}$. Eine Teilmenge $P \subset \mathcal{P}(I \times J)$ der Menge aller Teilmengen von $I \times J$ heißt Partition, falls*

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

gilt und falls aus $b_1 \cap b_2 \neq \emptyset$ folgt, dass $b_1 = b_2$ für alle $b_1, b_2 \in P$ gilt. Die Elemente von P werden als (Index-) Blöcke bezeichnet.

Bevor wir uns näher mit der Partitionierung auseinandersetzen, sollte sichergestellt sein, dass die Blöcke der Matrix A, welche in den meisten Anwendungsfällen an das zu Grunde liegende Problem gekoppelt sind, mit einer Niedrigrang-Matrix approximiert werden können. Dies kann mit einer Zulässigkeitsbedingung gewährleistet werden, welche die folgenden drei Eigenschaften voraussetzt:

1. Falls $b \in P$ zulässig ist, so fallen die Singulärwerte von A_b exponentiell ab, wobei A_b die Restriktion von A auf den Block b bezeichnet.

2. Die Zulässigkeitsbedingung kann für jeden Block $b := t \times s \in \mathcal{P}(I \times J), t \subset I$, $s \subset J$, geprüft werden und die benötigten Rechenoperationen sind $\mathcal{O}(|t| + |s|)$.

3. Falls $b \in P$ zulässig ist, so ist auch jede Teilmenge $b' \subset b$ zulässig.

Natürlicherweise wird es auch Blöcke geben, die die Zulässigkeitsbedingung nicht erfüllen, was zum Beispiel bei der Funktion $f(x, y) = |x - y|^{-1}$ der Fall ist, da sie für $x = y$ singulär ist. Um die Gesamtkomplexität nicht zu zerstören, sollten nicht-zulässige Blöcke vergleichsweise klein sein. Daher wird neben der Zulässigkeitsbedingung die Zulässigkeit der Partition definiert.

Definition 5.2. *Eine Partition P heißt zulässig, falls jeder Block $b = t \times s \in P$ entweder zulässig oder klein ist, d.h. die Mächtigkeiten von t und s erfüllen mit einer gegebenen minimalen Dimension $n_{min} \in \mathbb{N}$ die Bedingung*

$$min\{|t|, |s|\} \leq n_{min}.$$

Die Zulässigkeitsbedingung führt je nach Anwendungsgebiet auf andere Bedingungen, welche deutlich leichter zu greifen sind. In unserem Fall betrachten wir Matrizen $A \in \mathbb{R}^{M \times N}$, deren Einträge

$$a_{ij} = (\mathcal{V}\psi_j, \varphi_i), \quad i = 1, ..., N, j = 1, ..., M,$$

aus der Diskretisierung eines nicht-lokalen Operators $\mathcal{V}$, wie der Randintegralformulierung der in Kapitel 9 beschriebenen Probleme, resultieren, wodurch A voll besetzt ist. Auf geeigneten Blöcken $t \times s$ kann die Matrix A durch Niedrigrang-Matrizen approximiert werden, d.h.

$$A_{ts} \approx UV^T, \quad U \in \mathbb{R}^{t \times k}, \; V \in \mathbb{R}^{s \times k},$$

wobei der Rang k im Vergleich zu den Mächtigkeiten $|t|$ und $|s|$ klein ist. Wie in [15] gezeigt wird, ist die Bedingung

$$\min\{\operatorname{diam} X_t, \operatorname{diam} X_s\} \leq \eta \operatorname{dist}(X_t, X_s), \quad \eta > 0, \tag{5.1}$$

geeignet, falls A einen Integraloperator oder die Inverse eines elliptischen partiellen Differentialoperators diskretisiert. Dabei bezeichnen X_t die Vereinigung der Träger $X_i := \operatorname{supp} \varphi_i$, $i \in t$ und

$$\operatorname{diam} X = \sup_{x,y \in X} |x - y| \quad \text{und} \quad \operatorname{dist}(X, Y) = \inf_{x \in X,\, y \in Y} |x - y|$$

den Durchmesser und den Abstand von zwei beschränkten Mengen $X, Y \subset \mathbb{R}^d$. Alle Blöcke, die die Bedingung (5.1) erfüllen, heißen zulässig. Die Abbildungen 5.1 und 5.2 zeigen zulässige Blöcke auf zwei verschiedenen Geometrien.

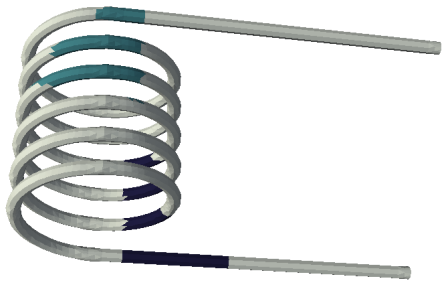

Abb. 5.1: Zwei Cluster X_t und X_s, welche einen zulässigen Block $t \times s$ auf der Geometrie einer elektrischen Spule bilden.

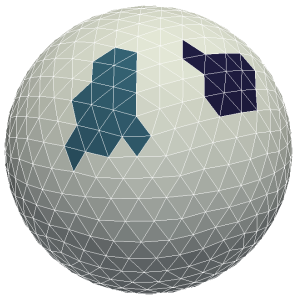

Abb. 5.2: Zwei Cluster X_t und X_s, welche einen zulässigen Block $t \times s$ auf der Einheitssphäre bilden.

Da die Berechnung von Bedingung (5.1) im Allgemeinen $\mathcal{O}(|t| \cdot |s|)$ Operationen benötigt, verletzt die Auswertung dieser Art von Zulässigkeitsbedingung die Voraussetzung 2. Zudem führt die aktuell benötigte Anzahl an Operationen zu keiner Komplexitätsreduktion. Zur Konstruktion einer Bedingung, welche nur $\mathcal{O}(|t| + |s|)$ Operationen erfordert und dennoch (5.1) erfüllt, seien die Mengen X_i für alle $i \in I$ (bzw. X_j für allle $j \in J$) polygonal. Eine derartige Annahme stellt keine starke Einschränkung dar, da wir die Zulässigkeitsbedingung in Verbindung mit Gebietsdiskretisierungen anwenden wollen, welche auf Dreicken, Vierecken oder allgemein auf Polygonen beruhen.
Der Abstand zwischen zwei Clustern X_t und X_s lässt sich im Vorfeld mit Hilfe deren Schwerpunkte m_t und m_s abschätzen. Zusammen mit

$$\rho_t := \sup\{|x - m_t|, x \in X_t\}, \quad t \subset I$$

folgt

$$\operatorname{dist}(X_t, X_s) \geq |m_t - m_s| - \rho_t - \rho_s.$$

Mit $\operatorname{diam} X_t \leq 2\rho_t$ kann die teuer auszuwertende Zulässigkeitsbedingung (5.1) ersetzt werden durch die Bedingung

$$2\min\{\rho_t, \rho_s\} + \eta(\rho_t + \rho_s) \leq \eta|m_t - m_s|, \quad t \subset I,\ s \subset J, \tag{5.2}$$

welche in $\mathcal{O}(|t|+|s|)$ Operationen ausgewertet werden kann. Die letzten Überlegungen sind nochmals geometrisch in Abbildung 5.3 dargestellt.

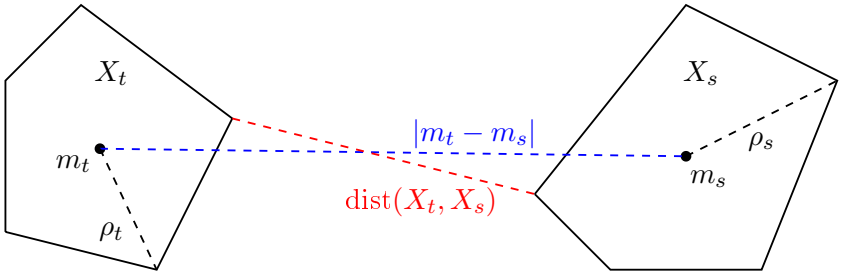

Abb. 5.3: Geometrische Idee zur verbesserten Zulässigkeitsbedingung.

Die Partition der Matrix A erfolgt üblicherweise mittels Clusterbäumen [45, 48], mit denen wir uns im nächsten Abschnitt genauer befassen werden.

5.2 Cluster- und Block-Clusterbäume

Bei der Partitionierung einer Matrix soll keine optimale Partition erreicht werden, da diese im Allgemeinen zu aufwändig sein wird. Daher beschränken wir uns auf eine rekursive Unterteilung der Indexmengen I und J. Die Struktur eines Clusterbaumes gibt einen Leitfaden an, wie zunächst I und J partitioniert werden können. Anschließend werden beide Clusterbäume unter Berücksichtigung der Zulässigkeitsbedingung (5.1) bzw. (5.2) zu einen Block-Clusterbaum verknüpft. Die Menge der Blätter des Block-Clusterbaumes ergibt die gesuchte zulässige Unterteilung. Es kann gezeigt werden, dass die resultierende Partitionierung eine nahezu lineare Komplexität aufweist, siehe auch [15] oder [45, 48].

Definition 5.3. *Ein Baum $T_I = (V, E)$ mit Knoten V und Kanten E heißt Clusterbaum für eine Menge $I \subset \mathbb{N}$, falls die folgenden Bedingungen erfüllt sind:*

(i) I ist die Wurzel von T_I.

(ii) $\emptyset \neq t = \dot{\bigcup}_{t' \in S(t)} t'$ für alle $t \in V \setminus \mathcal{L}(T_I)$.

(iii) Der Grad $\deg t := |S(t)| \geq 2$ jedes Knotens $t \in V \setminus \mathcal{L}(T_I)$ ist von unten beschränkt.

Die Menge an Söhnen $S(t) := \{t' \in V \ : \ (t, t') \in E\}$ von $t \in V$ ist paarweise disjunkt und

$$\mathcal{L}(T_I) := \{t \in V \ : \ S(t) = \emptyset\}$$

bezeichnet die Menge der Blätter von T_I. Die Anzahl an Anwendungen von S auf I, welche notwendig sind, um t zu erhalten, heißt der Level von $t \in T_I$.

Als erste Folgerung lässt sich anmerken, dass jeder Level wiederum eine Partition von I enthält.

Lemma 5.4. *Für alle $t \in T_I$ gilt, dass $t \subset I$ und dass jeder Level*

$$T_I^{(l)} := \{t \in T_I \ : \ \text{Level } t = l\}$$

von T_I zusammen mit $\{t \in \mathcal{L}(T_I) \ : \ \text{Level } t < l\}$ eine Partition von I enthält.

Beweis. Die Behauptung ist eine direkte Folgerung aus (ii) der Definition 5.3. □

Zwei Clusterbäume unterscheiden sich nach Definition 5.3 nur in der Abbildung $t \mapsto S(t)$. Um in unserem Fall die Zulässigkeitsbedingungen zu berücksichtigen, ist S so beschaffen, dass die erzeugten Cluster einen minimalen Durchmesser aufweisen. Durch die spezielle Wahl der Abbildung S und deren rekursive Anwendung zur Konstruktion des Clusterbaumes T_I, kann die Menge der Knoten V mit dem Clusterbaum T_I identifiziert werden.

Eine andauernde rekursive Anwendung von S, solange bis eine Clustergröße von 1 erreicht ist, würde sich letztendlich bei der Darstellung der Matrix bzw. der einzelnen Blöcke der Matrix als äußeres Produkt nicht lohnen. Daher führen wir eine minimale Clustergröße $n_{\min} > 1$ ein. Damit lässt sich zeigen, dass der Speicherbedarf eines Clusterbaumes linear ist, siehe [15].

Lemma 5.5. *Seien $n_{\min} > 1$ und $q := \min_{t \in T_I \setminus \mathcal{L}(T_I)} \deg t \geq 2$. Dann gelten die folgenden beiden Aussagen:*

(i) Für alle $t \in T_I$ ist die Anzahl der Blätter $|\mathcal{L}(T_I)|$ beschränkt durch

$$|\mathcal{L}(T_I)| \leq \frac{|I|}{n_{\min}}.$$

(ii) Die Anzahl der Knoten in T_I ist beschränkt durch

$$|T_I| \leq \frac{q|\mathcal{L}(T_I)| - 1}{q - 1} \leq 2|\mathcal{L}(T_I)| - 1.$$

Beweis. Die Behauptung (i) folgt aus der Definition 5.3 eines Clusterbaums und der Beschränkung der Clustergröße.

Für den zweiten Teil streichen wir den Baum startend bei den Blättern von $T \setminus \mathcal{L}(T)$ in k Schritten Knoten für Knoten zusammen, bis nur noch die Wurzel übrig ist. Sei daher T_l der Clusterbaum nach l-Schritten und q_l der Grad des l-ten Knotens. Dann gelten

$$|T_{l+1}| = |T_l| - q_{l+1}$$

und

$$|\mathcal{L}(T_{l+1})| = |\mathcal{L}(T_l)| - q_{l+1} + 1.$$

Demnach gilt nach k Schritten, dass $|T_k| = 1 = |\mathcal{L}(T_k)|$ und

$$|T_k| = |T| - \sum_{l=0}^{k-1} q_{l+1},$$

$$|\mathcal{L}(T_k)| = |\mathcal{L}(T)| - \sum_{l=0}^{k-1} (q_{l+1} - 1).$$

Da wegen $q_l \geq q$ die Abschätzung

$$k(q-1) \leq |\mathcal{L}(T)| - 1$$

folgt, erhalten wir

$$\begin{aligned}|T| &= |\mathcal{L}(T)| + k \\ &\leq |\mathcal{L}(T)| + \frac{|\mathcal{L}(T)| - 1}{q-1} \\ &= \frac{q|\mathcal{L}(T)| - 1}{q-1}.\end{aligned}$$

Für $q = 2$ folgt sofort die zweite Abschätzung, denn es gilt

$$\frac{q|\mathcal{L}(T)| - 1}{q-1} = 2|\mathcal{L}(T)| - 1.$$

Im Fall $q > 2$ kann durch

$$\begin{aligned}\frac{q|\mathcal{L}(T)| - 1}{q-1} &= \frac{(q-1)(|\mathcal{L}(T)| - 1) + |\mathcal{L}(T)| + q - 2}{q-1} \\ &\leq |\mathcal{L}(T)| - 1 + \frac{|\mathcal{L}(T)| + q - 1}{q-1} \\ &\leq |\mathcal{L}(T)| - 1 + \frac{|\mathcal{L}(T)|(q-1)}{q-1} \\ &= 2|\mathcal{L}(T)| - 1\end{aligned}$$

die zweite Abschätzung gefolgert werden. □

Um den gesamten Speicherbedarf eines Clusterbaumes T_I abschätzen zu können, werden noch Aussagen zur Tiefe eines Baumes hilfreich sein.

Definition 5.6. *Die Tiefe eine Clusterbaumes T_I ist definiert durch*

$$L(T_I) := \max_{t \in T_I} \textit{Level } t + 1.$$

Im schlimmsten Fall, falls sich die Größe der Cluster stark voneinander unterscheidet, kann keine logarithmische Baumtiefe entstehen. Unter der Annahme eines balancierten Baumes, d.h. die Mächtigkeiten der Cluster sind von vergleichbarer Größe, kann eine in I logarithmische Abhängigkeit garantiert werden.

Definition 5.7. *Ein Baum T_I heißt balanciert, falls*

$$R := \min_{t \in T_I \setminus \mathcal{L}(T_I)} \{|t_1|/|t_2|,\, t_1, t_2 \in S(t)\}$$

unabhängig von I durch eine positive Konstante von unten beschränkt ist.

Lemma 5.8. *Sei T_I ein balancierter Clusterbaum mit $q := \min_{t \in T_I \setminus \mathcal{L}(T_I)} \deg t \geq 2$. Dann gilt für die Tiefe des Baumes T_I, dass*

$$L(T_I) \leq \log_\xi(|I|/n_{\min}) + 1 \sim \log|I|$$

und $|t| \leq |I|\xi^{-l}$, wobei l das Level von $t \in T_I$ bezeichnet und $\xi := R(q-1) + 1$.

Beweis. Seien $t \in T_I \setminus \mathcal{L}(T)$ und $t' \in S(t)$. Dann gilt

$$\begin{aligned}\frac{|t|}{|t'|} &= \frac{|t'| + \sum_{t' \neq s \in S(t)} |s|}{|t'|} \\ &= 1 + \sum_{t' \neq s \in S(t)} \frac{|s|}{|t'|} \\ &\geq 1 + (|S(t)| - 1)R \geq 1 + (q-1)R = \xi.\end{aligned}$$

Für den nächsten Schritt sei $e_1, ..., e_{L-1}$ eine Folge von Kanten von der Wurzel $v_1 := I$ zum tiefsten Knoten v_L in T_I und $v_2, ..., v_{L-1}$ die dazwischen liegenden Knoten. Mit

$$\xi |v_{l+1}| \leq |v_l|, \quad l = 1, ..., L-1$$

erhalten wir

$$\xi^{L-1} |v_L| \leq |I|.$$

Damit folgen die Abschätzungen

$$(L-1) \log \xi \leq \log(|I|/|v_L|) \leq \log(|I|/n_{\min})$$

und schließlich die Behauptung durch

$$L \leq \frac{1}{\log \xi} (\log(|I|/n_{\min}) + 1) = \log_\xi(|I|/n_{\min}) + 1 \sim \log |I|.$$

□

Lemma 5.9. *Sei T_I ein Clusterbaum für I. Dann gelten die Abschätzungen:*

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

und

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

Beweis. Jedes Level von T_I besteht aus disjunkten Teilmengen von I. Der zweite Teil folgt durch $\log |t| \leq \log |I|$, $t \in T_I$. □

Mit Lemma 5.8 und Lemma 5.9 folgt, dass der Speicherbedarf eines balancierten Clusterbaums T_I wie $|I| \log |I|$ skaliert.

Bevor wir zu den Block-Clusterbäumen übergehen, betrachten wir beispielhaft die Konstruktion eines Clusterbaums bei der Finite-Elemente-Diskretisierung eines elliptischen Operators. Dazu sei die Menge X_t die Vereinigung der Träger aller Ansatzfunktionen φ_i mit $i \in t$, d.h.

$$X_t = \bigcup_{i \in t} X_i, \quad X_i = \text{supp } \varphi_i.$$

Im Hinblick auf Bedingung (5.1) ist es unser Ziel, den Durchmesser von X_t zu verringern. Zudem soll die benötigte Anzahl an Operationen, um einen Clusterbaum zu generieren, die Gesamtkomplexität nicht erhöhen. Die Konstruktion eines Clusterbaumes T_I auf Basis einer Indexmenge I erfordert einige zusätzliche Annahmen, welche mit den üblichen Anwendungen bzgl. der Diskretisierung mit finiten Elementen übereinstimmen. Wir nehmen die folgenden Voraussetzungen an:

1. Das betrachtete Gebiet $\Omega \subset \mathbb{R}^d$ ist eine m-dimensionale Untermannigfaltigkeit, d.h. es existiert eine Konstante $c_\Omega > 0$, sodass für alle $x \in \Omega$ gilt:

$$\mu(\Omega \cap B_r(x)) \leq c_\Omega r^m, \quad r > 0, \tag{5.3}$$

   wobei $\mu(U)$ das m-dimensionale Maß einer m-dimensionalen Untermannigfaltigkeit $U \subset \mathbb{R}^d$ bezeichnet.

2. Die Mengen X_i, $i \in I$, sind quasi-uniform und von regelmäßiger Form. Demnach gibt es zwei Konstanten $c_U > 0$ und $c_R > 0$ mit

$$\max_{i \in I} \mu(X_i) \leq c_U \min_{i \in I} \mu(X_i) \tag{5.4}$$

   und

$$\mu(X_i) \geq c_R (\text{diam } X_i)^m. \tag{5.5}$$

3. Die Anzahl an Überlappungen der Mengen X_i ist beschränkt, d.h. es existiert eine Zahl $\nu \in \mathbb{N}$, sodass

$$\max_{x \in \text{int } X_i} |\{j \in J \ : \ x \in \text{int } X_j\}| \leq \nu \tag{5.6}$$

   für alle $i \in I$ gilt.

Es gibt verschiedene Methoden, einen Clusterbaum zu erstellen. Da die Clusterbäume bzgl. der numerischen Experimente in dieser Arbeit mit Hilfe der Hauptkomponentenanalyse (engl. principal component analysis, PCA) erzeugt wurden, siehe auch [58], folgt eine kurze Erläuterung dieser Clusterstrategie.

Wir fixieren geeignete Punkte $z_i \in X_i$, $i = 1, ..., n$, und berechnen das Zentrum m_t des Clusters $t \subset I$ mit

$$m_t := \frac{\sum_{i \in t} \mu(X_i) z_i}{\sum_{i \in t} \mu(X_i)}.$$

Der Cluster t wird durch diejenige Hyperebene geteilt, welche durch das Zentrum m_t geht und orthogonal zur Hauptrichtung des Clusters ω_t verläuft. Dabei bezeichnet die Hauptrichtung des Clusters t einen Vektor $\omega_t \in \mathbb{R}^d$, $\|\omega_t\|_2 = 1$, mit

$$\sum_{i \in t} |\omega_t^T (z_i - m_t)|^2 = \max_{\|v\|_2 = 1} \sum_{i \in t} |v^T (z_i - m_t)|^2.$$

Die obige Maximierung ist gleichbedeutend mit der Suche nach dem größten Eigenwert der symmetrischen und positiv definiten Kovarianzmatrix

$$C_t := \sum_{i \in t} (z_i - m_t)(z_i - m_t)^T \in \mathbb{R}^{d \times d},$$

denn es gilt:

$$\begin{aligned} \max_{\|v\|_2 = 1} \sum_{i \in t} |v^T (z_i - m_t)|^2 &= \max_{\|v\|_2 = 1} \sum_{i \in t} v^T (z_i - m_t)(z_i - m_t)^T v \\ &= \max_{\|v\|_2 = 1} v^T C_t v \\ &= \lambda_{\max}(C_t). \end{aligned}$$

Obiges Maximum wird demnach für denjenigen Eigenvektor erreicht, der dem größten Eigenwert von C_t entspricht. Falls die Mengen X_i polygonal sind, so stellen die Punkte z_i die Zentren der X_i dar.

In unserem Fokus sind geometrisch balancierte Bäume, d.h. es gibt Konstanten $c_g > 0$ und $c_G > 0$, sodass für jeden Level $l = 0, ..., L(T_I) - 1$ die beiden Abschätzungen

$$(\text{diam } X_t)^m \leq c_g 2^{-l}$$

und

$$\mu(X_t) \geq 2^{-l} c_G^{-1}$$

für alle $t \in T_I^{(l)}$ erfüllt sind. Derartige Bäume können unter Benutzung der Hauptrichtung ω_t infolge der Definition der Sohnabbildung $S(t) = \{t_1, t_2\}$ durch

$$\begin{aligned} t_1 &:= \{i \in t \ : \ w_t^T (z_i - m_t) > 0\}, \\ t_2 &:= t \setminus t_1 \end{aligned} \tag{5.7}$$

generiert werden. Für quasi-uniforme Gitter sind geometrisch balancierte Bäume wiederum balancierte Bäume:

Lemma 5.10. *Angenommen die Mengen X_i, $i \in I$, sind quasi-uniform. Dann ist ein geometrisch balancierter Clusterbaum balanciert im Sinne von Definition 5.7.*

Beweis. Seien $t_1, t_2 \in T_I$ zwei Cluster desselben Levels l von T_I. Mit der Formregularität, (5.6) und den Eigenschaften eines geometrisch balancierten Baums folgen

$$\begin{aligned} |t_1| \min_{i \in t_1} \mu(X_i) &\leq \sum_{i \in t_1} \mu(X_i) \leq \nu \mu(X_{t_1}) \\ &\leq \nu c_\Omega (\text{diam } X_{t_1})^m \leq \nu c_\Omega c_g 2^{-l} \end{aligned}$$

und

$$|t_2| \max_{i \in t_2} \mu(X_i) \geq \sum_{i \in t_2} \mu(X_i) \geq \mu(X_{t_2}) \geq \frac{2^{-l}}{c_G}.$$

Der Quotient

$$\frac{|t_1|}{|t_2|} \leq \nu c_\Omega c_g c_G \frac{\max_{i \in t_2} \mu(X_i)}{\min_{i \in t_1} \mu(X_i)} \leq \nu c_\Omega c_g c_G c_U$$

liefert die Behauptung. □

Das folgende Lemma zeigt, wie aufwändig die Erstellung eines Clusterbaums unter der Benutzung von Bedingung (5.7) ist und wie viel Speicher er benötigt, siehe [15].

Lemma 5.11. *Die Konstruktion eines Clusterbaumes T_I benötigt mit den in (5.7) beschriebenen Unterteilungen $\mathcal{O}(|I| \log |I|)$ Operationen. Falls die Mengen X_i, $i \in I$, quasi-uniform sind, so ist der entstehende Clusterbaum T_I balanciert. Der Speicherbedarf von T_I beträgt $\mathcal{O}(|I|)$.*

Die Konstruktion eines Clusterbaums T_I reicht für die Partitionierung von $I \times J$ nicht aus. Dafür benötigen wir einen Clusterbaum $T_{I \times J}$, welcher aufgrund seiner Knoten Block-Clusterbaum genannt wird. Seien T_I und T_J zwei Clusterbäume für die Indexmengen I und J mit den zugehörigen Abbildungen S_I und S_J. Ein Block-Clusterbaum $T_{I \times J}$ ist definiert durch die Abbildung

$$S_{I \times J}(t \times s) = \begin{cases} \emptyset, & \text{falls } t \times s \text{ zulässig ist oder } S_I(t) = \emptyset \text{ oder } S_J(s) = \emptyset \\ S_I(t) \times S_J(s), & \text{sonst} \end{cases}.$$

Anhand dieser Definition ist offensichtlich, dass die Tiefe des Clusterbaums $L(T_{I\times J})$ durch die minimale Tiefe der erzeugenden Clusterbäume T_I und T_J beschränkt ist. Zudem bilden die Blätter des Baums $T_{I\times J}$ eine zulässige Partition $P = \mathcal{L}(T_{I\times J})$ unter der Voraussetzung, dass die Partitionen welche durch die Bäume T_I und T_J entstanden sind, zulässig sind. Die folgende Größe stellt bei Block-Clusterbäumen ein Maß für die Komplexität des Baums dar, siehe [40].

Definition 5.12. *Seien T_I und T_J Clusterbäume für die Indexmengen I und J und sei $T_{I\times J}$ ein Block-Clusterbaum für $I \times J$. Für einen Cluster $t \in T_I$ bzw. $s \in T_J$ bezeichnen die Ausdrücke*

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

bzw.

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

die maximale Anzahl an Blöcken $t \times s \in T_{I\times J}$. Die Sparsity-Konstante c_{sp} eines Block-Clusterbaumes $T_{I\times J}$ ist definiert durch

$$c_{\mathrm{sp}}(T_{I\times J}) := \max\{\max_{t\in T_I} c^r_{\mathrm{sp}}(T_{I\times J}, t), \max_{s\in T_J} c^c_{\mathrm{sp}}(T_{I\times J,s})\}.$$

Wichtig ist, die Konstante c_{sp} unabhängig von der Mächtigkeit der Indexmengen I bzw. J abzuschätzen. Bei geometrisch balancierten Bäumen, welche hauptsächlich bei elliptischen Problemen Verwendung finden, hängt die Konstante c_{sp} nur von ν, siehe (5.6), und η ab, siehe [15].

Lemma 5.13. *Angenommen die Clusterbäume T_I und T_J seien geometrisch balanciert. Dann gilt unter Berücksichtigung der Zulässigkeitsbedingung* (5.1)*, dass*

$$c_{\mathrm{sp}} \leq 2\nu c_g c_\Omega c_G \left(2 + \frac{1}{\eta}\right)^m.$$

Beweis. Wir zeigen die Behauptung für c^r_{sp}. Eine Vertauschung der Rollen der gewählten t und s führt auf dieselbe Schranke für c^c_{sp} wie bei c^r_{sp}. Seien $t \in T_I^{(l)}$ und $z_t \in X_t$. Wir definieren die Menge

$$N_\rho := \{s \in T_J^{(l)} \,:\, \max_{x\in X_s} |x - z_t| \leq \rho\}, \quad \rho > 0,$$

welche die Nachbarschaft von t beschreibt. Unter den getroffenen Voraussetzungen lässt sich aus

$$\frac{|N_\rho|}{c_G 2^l} \leq \sum_{s\in N_\rho} \mu(X_s) \leq \nu\mu(X_{N_\rho}) \leq \nu c_\Omega \rho^m$$

folgern, dass die Menge N_ρ höchstens $\nu c_G c_\Omega 2^l \rho^m$ Cluster s vom selben Level l in T_J enthält.

Als nächstes seien $s \in T_J$, sodass $t \times s \in T_{I\times J}$, und t^*, s^* die Vatercluster von t bzw. s. Angenommen, für

$$\rho_0 := \frac{1}{\eta} \min\{\text{diam } X_{t^*}, \text{diam } X_{s^*}\} + \text{diam } X_{t^*} + \text{diam } X_{s^*}$$

gelte die Bedingung

$$\max_{x\in X_s} |x - z_t| \geq \rho_0.$$

Dies hat zur Folge, dass $t^* \times s^*$ zulässig ist, denn es gilt

$$\text{dist } (X_{t^*}, X_{s^*}) \geq \max_{x\in X_s} |x - z_t| - \text{diam } X_{t^*} - \text{diam } X_{s^*} \geq \frac{1}{\eta} \min\{\text{diam } X_{t^*}, \text{diam } X_{s^*}\}.$$

Damit kann der Block-Clusterbaum $T_{I\times J}$ keinen Block $t \times s$ enthalten, wodurch ein Widerspruch entsteht. Die getroffene Annahme muss demnach falsch gewesen sein. Somit erhalten wir

$$\max_{x\in X_s} |x - z_t| < \rho_0 \leq \left(c_g 2^{-(l-1)}\right)^{1/m} \left(2 + \frac{1}{\eta}\right),$$

was schließlich

$$c_{sp}^r \leq 2\nu c_\Omega c_g c_G \left(2 + \frac{1}{\eta}\right)^m.$$

zur Folge hat. □

Damit lassen sich die Anzahl der Blätter des Baums, d.h. die Mächtigkeit der Partition P, und die Anzahl an Operationen, um den Baum zu konstruieren, abschätzen.

Lemma 5.14. *Seien T_I und T_J Clusterbäume für die Indexmengen I und J. Die Anzahl an Blöcken in einer Partition $\mathcal{L}(T_{I\times J})$ und die Anzahl an Knoten in $T_{I\times J}$ erfüllen die Abschätzung*

$$|\mathcal{L}(T_{I\times J})| \leq |T_{I\times J}| \leq 2c_{\mathrm{sp}} \min\{|\mathcal{L}(T_I)|, |\mathcal{L}(T_J)|\}.$$

Falls $|t| \leq n_{\min}$ für alle $t \in T_I \cup T_J$ gilt, so folgt

$$|\mathcal{L}(T_{I\times J})| \leq |T_{I\times J}| \leq \frac{2c_{\mathrm{sp}}}{n_{\min}} \min\{|I|, |J|\}.$$

Beweis. Um die Behauptung zu zeigen, schätzen wir die Mächtigkeit des Block-Clusterbaums $T_{I\times J} = \sum_{t\times s\in T_{I\times J}} 1$ ab. Einerseits erhalten wir unter Benutzung von c_{sp}^r die Schranke

$$\sum_{t\times s\in T_{I\times J}} 1 \leq \sum_{t\in T_I} |\{s \subset J \,:\, t\times s \in T_{I\times J}\}| \leq c_{sp}^r |T_I| \leq c_{sp}|T_I|$$

und andererseits

$$\sum_{t\times s\in T_{I\times J}} 1 \leq c_{sp}^c |T_J| \leq c_{sp}|T_J|,$$

falls c_{sp}^c genutzt wird. Zusammenfassend folgt, dass

$$|T_{I\times J}| \leq c_{sp} \min\{|T_I|, |T_J|\}.$$

Lemma 5.5 führt zu

$$|T_I| \leq 2|\mathcal{L}(T_I)| \quad \text{und} \quad |T_J| \leq 2|\mathcal{L}(T_J)|.$$

Die zweite Aussage ist wiederum eine Konsequenz aus Lemma 5.5. □

Lemma 5.15. *Seien T_I und T_J balancierte Clusterbäume. Mit der zweiten Voraussetzung der Zulässigkeitsbedingung ist die Anzahl der Operationen einen Block-Clusterbaum zu konstruieren von der Ordnung $c_{\mathrm{sp}}(|I|\log|I| + |J|\log|J|)$.*

Beweis. Um die Behauptung zu zeigen müssen wir die Summe $\sum_{t\times s\in T_{I\times J}} c(|t|+|s|)$ genauer betrachten. Mit Lemma 5.9 folgt

$$\begin{aligned}\sum_{t\times s\in T_{I\times J}} c(|t| + |s|) &= c\sum_{t\in T_I} \sum_{\{s\in T_J \,:\, s\in T_{I\times J}\}} |t| + c\sum_{s\in T_J} \sum_{\{t\in T_J \,:\, t\times s\in T_{I\times J}\}} |s| \\ &\leq cc_{sp}\left(\sum_{t\in T_I} |t| + \sum_{s\in T_J} |s|\right) \leq cc_{sp}\left(L(T_I)|I| + L(T_J)|J|\right).\end{aligned}$$

Lemma 5.8 liefert die Behauptung. □

Aufbauend auf den generierten zulässigen Partitionen werden wir zu hierarchischen Matrizen übergehen. Zudem werden deren Eigenschaften und effiziente Rechenoperationen analysiert.

5.3 Hierarchische Matrizen ($\mathcal{H}$-Matrizen)

Hierarchische Matrizen wurden ursprünglich von Hackbusch [44] und Hackbusch und Khoromskij [45, 46] eingeführt. Als Matrix-Approximante stellen sie eine dünnbesetzte Repräsentation einer vollbesetzten Matrix dar, welche mit logarithmisch-linearer Komplexität gespeichert werden kann. Zudem liefern die hierarchische Partitionierung und die Niedrigrang-Struktur die nötigen Bausteine, um schnelle Matrix-Operationen durchzuführen. So kann z.B. gezeigt werden, dass die Matrix-Vektor Mulitplikation in logarithmisch-linearer Komplexität durchführbar ist.
Für die Definition wird der Block-Clusterbaum $T_{I\times J}$ verwendet, von dem angenommen wird, dass bei der Konstruktion die Zulässigkeitsbedingung berücksichtigt wurde.

Definition 5.16. *Die Menge der hierarchsichen Matrizen auf dem Block-Clusterbaum $T_{I\times J}$ mit zulässiger Zerlegung $P := \mathcal{L}(T_{I\times J})$ und Blockrang k ist definiert durch*

$$\mathcal{H}(T_{I\times J}, k) := \{A \in \mathbb{R}^{I\times J} \ : \ \operatorname{rank} A_{ts} \leq k \text{ für alle zulässigen } t\times s \in P\}.$$

Elemente aus $\mathcal{H}(T_{I\times J}, k)$ werden auch $\mathcal{H}$-Matrizen genannt.

Lemma 5.17. *Sei $A \in \mathcal{H}(T_{I\times J}, k)$. Dann gelten die folgenden beiden Aussagen:*

(i) Jede Teilmatrix A_b, $b \in T_{I\times J}$, gehört zu $\mathcal{H}(T_b, k)$.

(ii) Die Matrizen A^T und A^H gehören zu $\mathcal{H}(T_{J\times I}, k)$, falls die Zulässigkeitsbedingung symmetrisch ist, d.h. jeder Block $s\times t$ ist zulässig, falls auch $t\times s$ zulässig ist.

Beweis. Die beiden Aussagen sind eine direkte Folge der Definition von hierarchischen Matrizen. □

Lemma 5.18. *Der Speicherbedarf N_{st} einer hierarchischen Matrix $A \in \mathcal{H}(T_{I\times J}, k)$ ist beschränkt durch*

$$N_{st}(A) \leq c_{\text{sp}} \max\{k, n_{\min}\} \left(L(T_I)|I| + L(T_J)|J|\right).$$

Falls T_I und T_J balancierte Clusterbäume sind, gilt

$$N_{st}(A) \sim \max\{k, n_{\min}\} \left(|I| \log |I| + |J| \log |J|\right).$$

Beweis. Der Speicherbedarf von zulässigen Blöcken $t\times s \in \mathcal{L}(T_{I\times J})$ einer hierarchischen Matrix A beträgt $k(|t| + |s|)$. Für die nicht zulässigen Blöcke werden $|t||s|$ Speichereinheiten benötigt. Wegen $\min\{|t|, |s|\} \leq n_{\min}$ ist das Produkt aus $|t|$ und $|s|$ beschränkt durch

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

Unter Benutzung des Beweises von Lemma 5.15 folgen die beiden Aussagen. □

5.3.1 Matrix-Vektor-Multiplikation für $\mathcal{H}$-Matrizen

In vielen iterativen Verfahren wie den Krylov-Unterraum-Methoden, siehe z.B. [64], kommt die Systemmatrix nur über eine Matrix-Vektor-Multiplikation ins Spiel. Auch die Auswertung der Lösung bei der Randintegralmethode kann mit Hilfe einer Matrix-Vektor-Multiplikation realisiert werden. Da sich jede Multiplikation einer $\mathcal{H}$-Matrix A mit einem Vektor x als Summe der Multiplikation eines Matrixblocks A_{ts} mit dem jeweiligen Anteil x_s des Vektors x darstellen lässt, d.h.

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

kann der Aufwand einer Matrix-Vektor Mulitplikation auf den Aufwand in den einzelnen Blöcken zurückgeführt werden, wobei die Matrizen A_{ts} in obiger Notation für die Aufsummierung in geeigneter Weise zu verstehen sind.

Jeder nicht-zulässige Block $t \times s$, wobei $\min\{|t|, |s|\} \leq n_{\min}$, wird eintragsweise abgespeichert, sodass für eine Multiplikation

$$2|t||s| = 2\min\{|t|, |s|\} \max\{|t|, |s|\} \leq 2n_{\min}(|t| + |s|)$$

arithmetische Operationen anfallen.

Jeder zulässige Block $t \times s$ kann als äußeres Produkt geschrieben werden, sodass gilt:

$$A_{ts}x_s = UV^H x_s, \quad U \in \mathbb{R}^{|t| \times k}, V \in \mathbb{R}^{|s| \times k},$$

wofür $2k(|t| + |s|)$ nötig sind.

Zusammenfassend lässt sich ähnlich zu Lemma 5.18 die Komplexität einer Matrix-Vektor-Multiplikation feststellen, siehe [15].

Lemma 5.19. *Die Anzahl der Operationen N_{MV} einer Matrix-Vektor Multiplikation Ax einer Matrix $A \in \mathcal{H}(T_{I \times J}, k)$ mit einem Vektor $x \in \mathbb{R}^N$ lässt sich abschätzen durch*

$$N_{MV}(A) \leq 2c_{\text{sp}} \max\{k, n_{\min}\} \left(L(T_I)|I| + L(T_J)|J|\right).$$

Falls T_I und T_J balancierte Clusterbäume sind, so gilt

$$N_{MV}(A) \sim \max\{k, n_{\min}\} \left(|I| \log |I| + |J| \log |J|\right).$$

Beweis. Der Beweis funktioniert ähnlich wie der Beweis von Lemma 5.18. □

5.3.2 Norm-Abschätzungen für $\mathcal{H}$-Matrizen

Im späteren Verlauf dieser Arbeit werden die Fehler der Matrixapproximationen hauptsächlich in der Spektralnorm untersucht. Hierzu müssen wir wissen, wie sich die Normen der einzelnen Blöcke bezüglich der Norm der gesamten Matrix verhalten. Zudem wird interessant sein, wie sich lokale Normen auf den Rest der Matrix auswirken. In den folgenden beiden Lemmata sollen diese Fragestellungen geklärt werden, siehe auch [15].

Lemma 5.20. *Betrachtet wird die folgende $r \times r$ Blockmatrix*

$$A = \begin{pmatrix} A_{11} & \dots & A_{1r} \\ \vdots & & \vdots \\ A_{r1} & \dots & A_{rr} \end{pmatrix}$$

mit $A_{ij} \in \mathbb{R}^{m_i \times n_j}$, $i, j = 1, ..., r$. Dann gilt, dass

$$\max_{i,j=1,\dots,r} ||A_{ij}||_2 \leq ||A||_2 \leq \left(\max_{i=1,\dots,r} \sum_{j=1}^{r} ||A_{ij}||_2 \right)^{1/2} \left(\max_{j=1,\dots,r} \sum_{i=1}^{r} ||A_{ij}||_2 \right)^{1/2}.$$

Beweis. Die untere Schranke der Abschätzung ist klar. Für die obere Schranke seien $x = (x_1, ..., x_r)^T \in \mathbb{R}^n$ mit $x_j \in \mathbb{R}^{n_j}$, $j = 1, ..., r$, und $n := \sum_{j=1}^{r} n_j$. Seien zudem $\hat{A} \in \mathbb{R}^{r \times r}$ mit $\hat{a}_{ij} = \|A_{ij}\|_2$ und $\hat{x} \in \mathbb{R}^r$ mit $\hat{x}_j = \|x_j\|_2$, $j = 1, ..., r$. Damit folgt

$$\|Ax\|_2^2 = \sum_{i=1}^{r} \| \sum_{j=1}^{r} A_{ij}x_j \|_2^2 \leq \sum_{i=1}^{r} \left(\sum_{j=1}^{r} \|A_{ij}\|_2 \|x_j\|_2 \right)^2 = \|\hat{A}\hat{x}\|_2^2.$$

Wir erhalten danach die obere Schranke durch

$$\|\hat{A}\hat{x}\|_2^2 \leq \|\hat{A}\|_1\|\hat{A}\|_\infty\|\hat{x}\|_2^2 = \|\hat{A}\|_1\|\hat{A}\|_\infty\|x\|_2^2,$$

denn es gilt $\|\hat{A}\|_2^2 \leq \|\hat{A}\|_1\|\hat{A}\|_\infty$. □

Lemma 5.21. *Sei P die Menge der Blätter eines Clusterbaums $T_{I\times J}$. Dann gelten für $A, B \in \mathcal{H}(T_{I\times j}, k)$ die beiden folgenden Aussagen:*

(i) $\max_{t\times s\in P} ||A_{ts}||_2 \leq ||A||_2 \leq c_{\mathrm{sp}} L(T_{I\times J}) \max_{t\times s\in P} ||A_{ts}||_2$.

(ii) Falls $\max_{t\times s\in P} ||A_{ts}||_2 \leq \max_{t\times s\in P} ||B_{ts}||_2$, *so folgt* $||A||_2 \leq c_{\mathrm{sp}} L(t_{I\times J})||B||_2$.

Beweis. Für den Beweis der ersten Aussage (i) werden die Blöcke der Matrix A in ihre Level aufgeteilt, d.h.

$$(A_l)_b = \begin{cases} A_b, & b \in T_{I\times J}^{(l)} \cap P, \\ 0, & \text{else} \end{cases}$$

und $A = \sum_{l=1}^{L(T_{I\times J})} A_{l-1}$. Damit weist A_l eine verschachtelte Struktur auf, wobei pro Blockzeile bzw. Blockspalte höchstens c_{sp} viele Blöcke existieren. Mit Lemma 5.20 folgt die Abschätzung

$$\begin{aligned} \|A\|_2 \leq \sum_{l=1}^{L(T_{I\times J})} \|A_{l-1}\|_2 &\leq c_{sp} \sum_{l=1}^{L(T_{I\times J})} \max_{b\in T_{I\times J}^{(l-1)}\cap P} \|A_b\|_2 \\ &\leq c_{sp} L(T_{I\times J}) \max_{b\in P} \|A_b\|_2. \end{aligned}$$

Die Aussage (ii) folgt durch

$$\max_{b\in P} \|A_b\|_2 \leq \max_{b\in P} \|B_p\|_2 \leq ||B||_2.$$

□

5.4 Uniforme $\mathcal{H}$- und $\mathcal{H}^2$-Matrizen

Um die betrachteten Matrizen noch effizienter zu approximieren, können uniforme $\mathcal{H}$-Matrizen oder auch $\mathcal{H}^2$-Matrizen angestrebt werden, siehe [44].

Definition 5.22. *Eine Cluster-Basis Φ für eine Rang-Verteilung $(k_t)_{t\in T_I}$ ist eine Familie $\Phi = (\Phi(t))_{t\in T_I}$ von Matrizen $\Phi(t) \in \mathbb{R}^{t\times k_t}$.*

Definition 5.23. *Seien Φ und Ψ Cluster-Basen für T_I und T_J. Eine Matrix $A \in \mathbb{R}^{I\times J}$, die*

$$A|_{ts} = \Phi(t)\, F(t,s)\, \Psi(s)^H \quad \text{für alle } t\times s \in P_{\mathrm{adm}}$$

mit einem $F(t,s) \in \mathbb{R}^{k_t^\Phi \times k_s^\Psi}$ erfüllt, heißt uniforme hierarchische Matrix für Φ und Ψ.

Falls $k_t \leq k$ für alle $t \in T_I$ oder $k_s \leq k$ für alle $s \in T_J$ angenommen wird, so ist der Speicherbedarf für die Kopplungsmatrizen $F(t,s)$ von der Ordnung $k\min\{|I|,|J|\}$. Des Weiteren ist es nicht sinnvoll $k_t > |t|$ zu wählen. Die Cluster-Basen an sich belegen $k[|I|L(T_I) + |J|L(T_J)]$ Speichereinheiten, siehe [47].

Eine rekursive Relation unter den Basisvektoren kann helfen, auch den Speicherbedarf der beiden Basen Φ und Ψ zu verringern. Die zugehörige Struktur bilden die $\mathcal{H}^2$-Matrizen, siehe [47, 22]. Bei der Behandlung von hochfrequenten Helmholtz-Problemen ist diese zusätzliche Hierarchie in den $\mathcal{H}$-Matrizen sogar notwendig, um eine logarithmisch-lineare Komplexität zu erreichen. Des Weiteren wurden in [20] gerichtete bzw. direktionale $\mathcal{H}^2$-Matrizen als Spezialisierung von $\mathcal{H}^2$-Matrizen eingeführt.

Definition 5.24. *Eine Cluster-Basis $U = (U(t))_{t\in T_I}$ heißt geschachtelt, falls es für alle $t \in T_I \setminus \mathcal{L}(T_I)$ Transfermatrizen $T_{t't} \in \mathbb{R}^{k_{t'}\times k_t}$ gibt, sodass für die Restriktion der Matrix $U(t)$ auf die Reihen t' gilt*

$$U(t)|_{t'} = U(t')\,T_{t't} \quad \text{for all } t' \in S_I(t).$$

Um die Komplexität der Speicherung einer verschachtelten Cluster-Basis U abzuschätzen, ist zu beachten, dass die Menge der Blatt-Cluster T_I eine Partition von I darstellt und für jeden Blatt-Cluster $t \in T_I$ müssen höchstens $k|t|$ Einträge gespeichert werden. Dies hat zur Folge, dass $\sum_{t\in T_I} k|t| = k|I|$ Speichereinheiten für die Blattmatrizen $U(t)$, $t \in T_I$, benötigt werden. Der Speicherbedarf für die Transfermatrizen ist ebenfalls von der Größenordnung $k|I|$, siehe [47].

Definition 5.25. *Eine Matrix $A \in \mathbb{R}^{I\times J}$ heißt $\mathcal{H}^2$-Matrix, falls es Cluster-Basen U und V gibt, sodass für $t \times s \in P_{\text{adm}}$ gilt*

$$A|_{ts} = U(t)\,F(t,s)\,V^H(s),$$

wobei $F(t,s) \in \mathbb{R}^{k_t^U \times k_s^V}$ die Kopplungsmatrix beschreibt.

Nach Definition 5.25 beläuft sich der Gesamtspeicherbedarf einer $\mathcal{H}^2$-Matrix auf die Größenordnung $k(|I| + |J|)$.

Bemerkung 5.26. *Weitere Ansätze sind von hybrider Form, in denen die Approximation aus einer Mischung von $\mathcal{H}$- und $\mathcal{H}^2$-Matrizen zusammengesetzt ist. Denn es kann vorteilhaft sein, geschachtelte Basen nur für Cluster t mit einer minimalen Kardinalität $n_{\min}^{\mathcal{H}^2} \geq n_{\min}$ zu betrachten. Blöcke, die aus kleineren Clustern bestehen, werden mit $\mathcal{H}$-Matrizen behandelt.*

Approximationen mit uniformen $\mathcal{H}$- oder $\mathcal{H}^2$-Matrizen sind ohne zusätzliches Wissen über den analytischen Hintergrund des diskretisierten Operators nicht möglich. Hier stellt zum Beispiel die Konstruktion einer geeigneten Cluster-Basis eine entscheidende Rolle dar. Um eine möglichst große Universalität der Methode zu gewährleisten, werden häufig polynomiale Räume verwendet, siehe [25]. Diese Wahl ist zwar aufgrund der besonderen Eigenschaften von Polynomen recht zweckmäßig, jedoch ist sie in der Regel nicht der effizienteste Ansatz. Um zu sehen, warum das so ist, sollte man sich vor Augen halten, dass der dreidimensionale Ansatz, der auf Kugelflächenfunktionen [31] beruht, in einer abgebrochenen Entwicklung mit einer Genauigkeit der Größenordnung von p, Terme in der Anzahl von $k = \mathcal{O}(p^2)$ benötigt, während die Anzahl der Polynomterme für dieselbe Ordnung der Genauigkeit $k = \mathcal{O}(p^3)$ Terme erfordert.

In [10] wird eine Verallgemeinerung der adaptiven Kreuzapproximationsmethode zur kernunabhängige Konstruktion von $\mathcal{H}^2$-Matrizen für Matrizen $A \in \mathbb{R}^{M\times N}$ mit Einträgen der Form

$$a_{ij} = \int_\Omega \int_\Omega K(x,y)\varphi_i(x)\psi_j(y)\,\mathrm{d}y\,\mathrm{d}x, \quad i = 1,\ldots,M,\ j = 1,\ldots,N.$$

vorgeschlagen. Hierbei bezeichnen φ_i und ψ_j lokale Ansatz- und Testfunktionen. Die Kern-Funktion K ist von der Art

$$K(x,y) = \xi(x)\,\zeta(y)\,f(x,y)$$

mit einer singulären Funktion $f(x,y) = |x-y|^{-\alpha}$ und Funktionen ξ und ζ, die jeweils nur von einer der Variablen x und y abhängen, vgl. Kapitel 1. Im Gegensatz zu $\mathcal{H}$-Matrizen, bei denen die Methode auf Blöcke angewandt wird, müssen im Falle von $\mathcal{H}^2$-Matrizen Clusterbasen konstruiert werden. Wenn dies adaptiv geschehen soll, müssen spezielle Eigenschaften des Kerns ausgenutzt werden, um zu gewährleisten, dass der Fehler auch außerhalb des Clusters kontrolliert wird. Der Ansatz in [10] stützt sich auf die Harmonizität des singulären Teils f der Kern-Funktion K. Dazu werden Interpolanten s_k zu den Kernfunktionen f konstruiert, die harmonisch in Bezug auf eine

Variable sind. Das System von Funktionen, in dem die interpolierende Funktion konstruiert wird, wird durch Restriktionen von f definiert. Diese Konstruktion garantiert, dass die Harmonizität von f für seinen Interpolationsfehler erhalten bleibt. Um eine vorgeschriebene Genauigkeit außerhalb des betrachteten Gebiets zu erreichen, genügt es, sie an ihrem Rand zu überprüfen. Dies erlaubt es, s_k auf eine kernunabhängige und adaptive Weise zu generieren. Anschließend wird die interpolierende Funktion s_k dazu verwendet, um eine Quadraturregel zu konstruieren, die bei der Konstruktion von geschachtelten Basen eingesetzt wird. Die in Kapitel 3 gewonnenen Ergebnisse gewährleisten dabei exponentielle Fehlerabschätzungen für s_k bei der Interpolation der Funktion $f(x, y) = |x - y|^{-\alpha}$ für beliebige $\alpha > 0$. Demnach können uniforme $\mathcal{H}$- und $\mathcal{H}^2$-Matrixapproximationen unter Verwendung harmonischer Interpolanten s_k konstruiert werden. Die resultierende Methode kann unter anderem zum Beispiel auf Poisson-Randwertprobleme oder auf fraktionelle Diffusionsprobleme angewendet werden. Für die detaillierte Beschreibung des Verfahrens sei an dieser Stelle auf [10] verwiesen, da das Hauptaugenmerk in dieser Arbeit auf der Generierung von $\mathcal{H}$-Matrixapproximationen liegt.

5.5 Ein numerisches Experiment

Mit dem folgenden numerischen Experiment wollen wir zum Abschluss von Kapitel I ein besseres Verständnis der adaptiven Kreuzapproximation erhalten. Im Blickpunkt liegt das Verhalten der ACA mit einer sich verändernden rechten Seite, wobei die strukturellen Unterschiede innerhalb der rechten Seite zunehmen. Dafür werden die numerischen Lösungen der Randwertprobleme

$$\begin{aligned} \Delta u = 0 \quad &\text{in } \Omega := \{x \in \mathbb{R}^3 \mid ||x||_2 \leq 1\} \\ u = G(x - p_i) \quad &\text{auf } \partial\Omega,\ i = 1, 2, 3, \end{aligned} \tag{5.8}$$

mit $p_1 = (10\ 0\ 0)^T$, $p_2 = (1.5\ 0\ 0)^T$ und $p_3 = (1.1\ 0\ 0)^T$ betrachtet. Die Diskretisierung des Randes der Einheitskugel Ω besteht bei 642 Punkten aus 1280 Dreiecken. Die dafür benötigten Techniken werden später in Kapitel acht und neun behandelt. Eine Anwendung der ACA auf die Systemmatrix, welche in Folge der Randelementmethode ensteht, liefert bei einer minimalen Blockgröße $b_{\min} = 15$ und einer Blockgenauigkeit $\varepsilon_{\text{ACA}} = 10^{-6}$ die in Tabelle 5.1 dargestellten Resultate.

i	$\frac{\|u-u_h\|_2}{\|u\|_2}$	Zeit [s]	Speicher [MB]	Kompressionsrate [%]
1	0.00158	0.309	3.845	61.47
2	0.0332	0.313	3.845	61.47
3	0.264	0.318	3.845	61.47

Tab. 5.1: Numerische Ergebnisse der ACA in drei verschiedenen Fällen.

Zunächst sind wir eher an einer Reduktion des Speicherbedarfs bei der approximierten Matrix und deren Kompressionsrate interessiert als am relativen Fehler der genäherten Lösung. Dennoch sollte der Fehler der approximierten Lösung in einem akzeptablen Bereich liegen. Obwohl wir die ACA auf drei verschiedene lineare Gleichungssysteme angewendet haben, haben wir im Bezug zur Kompression und zum notwendigen Speicherbedarf dieselben Ergebnisse erhalten. Wie die Abbildungen 5.4 und **??** zeigen, werden die Blöcke der Systemmatrix in den drei Fällen p_1, p_2 und p_3 vollkommen identisch approximiert. Die roten Blöcke sind nicht zulässig bzw. es musste jeder einzelne Eintrag berechnet werden. Der Rang der Niedrigrang-Approximation kann in den grünen, zulässigen Blöcken gefunden werden.

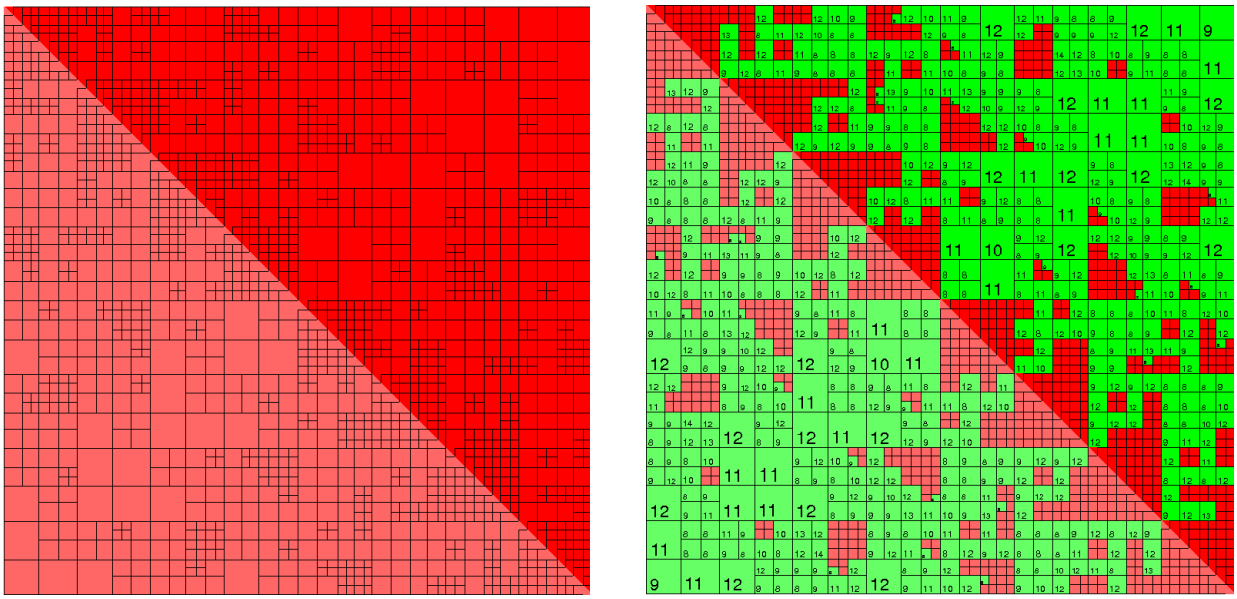

Abb. 5.4: Berechnung/Approximation der Blöcke ohne ACA (links) und in den Fällen p_1, p_2 und p_3 (rechts).

Da ACA ohne Berücksichtigung der rechten Seite auf der Matrix des linearen Gleichungssystems arbeitet, sind diese Ergbenisse nicht besonders überraschend. Sicherlich wird nicht jeder Block denselben Einfluss auf die rechte Seite haben wie ein anderer - zumindest nicht bei rechten Seiten mit größeren strukturellen Unterschieden. Unsere Aufgabe wird es sein, vorrangig diese Blöcke zu approximieren, bei denen der größte Genauigkeitsgewinn erzielt werden kann.
Daher wird die ACA in den folgenden Kapiteln mit Strategien erweitert werden, welche in einem anderen Zusammenhang auch in der adaptiven Randelementmethode [39, 50] zum Einsatz kommen. Zu diesen Techniken zählen eine Art von Fehlerschätzung, das Dörfler-Marking (siehe [34]) und spezielle Verfeinerungsstrategien.

II. Zusätzliche adaptive Elemente bei der adaptiven Kreuzapproximation

Bisher haben wir Verfahren kennengelernt, mit denen diskretisierte nicht-lokale Operatoren effizient, d.h. in logarithmisch-linearer Zeit, behandelt werden können. Auch der Speicherbedarf hängt hierbei nur logarithmisch-linear von der Anzahl der Freiheitsgrade ab. Jedoch ist im ersten Teil auch klar geworden, dass die vorgestellten Verfahren jeden diskreten Operator bzw. jedes Problem vollkommen gleich behandeln. Diese Vorgehensweise kann in einigen Situationen vorteilhaft sein, in anderen aber unnötige Informationen generieren. Wie am Ende von Kapitel 5 bereits kurz angesprochen, werden in den folgenden Kapiteln Techniken vorgestellt, mit denen wir die Approximation der betrachteten Operatoren genauer auf die Problemstellung ausrichten können, um so einerseits die Rechenzeit und andererseits den Speicherbedarf zu reduzieren. Dabei wird der diskrete Operator nicht wie zuvor im Vorfeld einmal komplett bis zu einer gewählten Genauigkeit approximiert, sondern sequentiell in einer adaptiven Prozedur.

6. Operationen auf $\mathcal{H}$-Matrizen

Wir beschäftigen uns im Folgenden mit der mathematischen Operation, welche die Multiplikation einer Matrix mit einem Vektor beschreibt. Durch eine Approximation des nicht-lokalen Operators kann die nötige Anzahl an Rechenoperationen auch hier verringert werden.

6.1 Adaptive Matrix-Vektor Multiplikation (AMVM)

Das Ziel ist die Entwicklung eines approximativen und adaptiven Algorithmus für die Multiplikation einer Matrix $A \in \mathbb{R}^{M \times N}$ mit einem Vektor $x \in \mathbb{R}^N$, d.h.

$$b = Ax,$$

wobei A die Diskretisierung eines nicht-lokalen Operators und b der resultierende Vektor ist. Da die Matrix A voll besetzt ist, wäre das eigentliche Vorgehen bei der Lösung dieser Problematik und den Methoden aus den vorherigen Kapiteln folgend, zuerst die Approximation der Systemmatrix A durch hierarchische Matrizen und die anschließende Multiplikation der Approximation von A mit dem Vektor x. Aufgrund der Konstruktion der Approximation durch ACA können redundante und unnötige Informationen entstehen, aus dem einfachen Grund, dass im ACA Verfahren jeder Matrixblock gleich

behandelt wird und somit mehr Zeit in Anspruch genommen wird. Um die Generierung solcher Informationen zu vermeiden, wird hier eine adaptive Strategie verfolgt. Anstelle des bisherigen Ansatzes, eine einzige Approximation zu erzeugen, konstruieren wir eine Folge von hierarchischen Matrixapproximationen A_k, wobei wir jede Approximation mit dem Vektor x multiplizieren, um eine Folge von Approximationen b_k auch für den resultierenden Vektor b zu erhalten. Die einzelnen Approximationen werden mit Hilfe eines Fehlerschätzers berechnet, der auf der $(h - h/2)$-Strategie, siehe z.B. [7, 8], und der Dörfler-Markierungstechnik [34] basiert. Es ist zu beachten, dass im Gegensatz zum konventionellen Anwendungsbereich solcher Fehlerschätzer hier keine Verfeinerung der Geometrie oder des Gitters berücksichtigt wird. Die zugrunde liegende Gitterstruktur und der zugrunde liegende Block-Clusterbaum werden zu keinem Zeitpunkt verändert.

Eine zuverlässige Schätzung des Fehlers erfordert zunächst die Existenz einer genaueren Approximation $\tilde{A}_k$, d.h. es gelte die sog. Saturierungsannahme

$$\|\hat{b}_k - b\|_2 \leq c_{\text{sat}} \|b_k - b\|_2 \tag{6.1}$$

für ein $0 < c_{\text{sat}} < 1$, wobei $b_k = A_k x$ und $\hat{b}_k = \hat{A}_k x$. Eine natürliche Wahl für $\hat{A}_k$ ist die verbesserte Approximation, die sich aus A_k ergibt, indem eine feste Anzahl von zusätzlichen ACA-Schritten auf jeden zulässigen Block angewendet wird und durch Setzen von $(\hat{A}_k)_{ts} = A_{ts}$ für alle anderen nicht-zulässigen Blöcke $t \times s \in P_{\text{non-adm}}$. Natürlich sieht dieses Verfahren viel komplexer aus als die einmalige Multiplikation einer Matrix mit einem gegebenen Vektor. Der Ansatz hier zielt darauf ab, Eigenschaften des Vektors x in Kombination mit Eigenschaften von A auszunutzen. Betrachten wir als Beispiel den Extremfall, dass $x = 0$ ist. Dann erkennt der adaptive Ansatz, dass es sinnlos ist, eine Approximation von A mit der Genauigkeit ε_{ACA} zu berechnen, während der übliche Ansatz zunächst jeden Block mit dieser Genauigkeit approximieren und dann die Multiplikation durchführen würde. Abhängig von der Kombination von A und x erwarten wir verbesserte Speicheranforderungen und Rechenzeiten.

In Anlehnung an die obigen Ideen wird insbesondere die Assemblierung des diskretisierten nichtlokalen Operators mit der gleichzeitigen Berechnung der Matrix-Vektor-Multiplikation kombiniert. Abbildung 6.1 zeigt eine schematische Darstellung des Verfahrens.

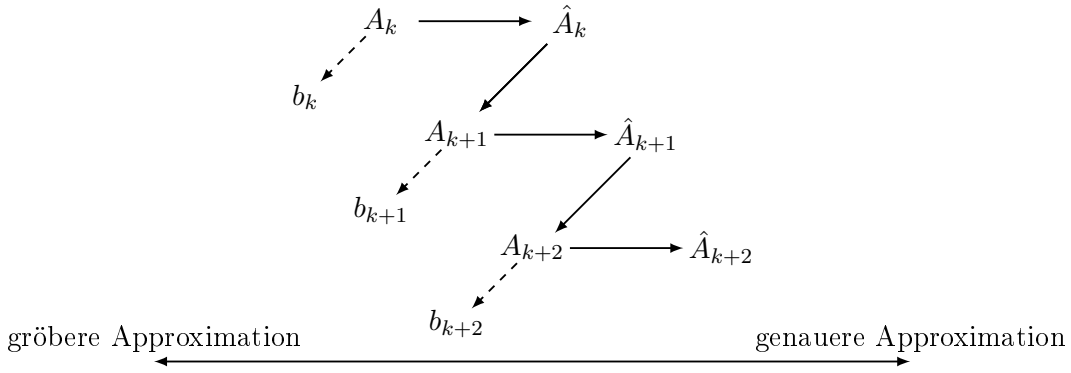

Abb. 6.1: Schematische Darstellung des Verfahrens.

Unter Benutzung des Fehlerschätzers

$$\gamma_k := \|b_k - \hat{b}_k\|_2 = \| \sum_{t \times s \in P} (A_k - \hat{A}_k)_{ts} x_s \|_2,$$

welcher in Bezug auf Blöcke in P lokalisiert ist, ist das Verfahren für die adaptive Matrix-Vektor Multiplikation in Algorithmus 2 zusammengefasst.

Algorithmus 2 Adaptive Matrix-Vektor Multiplikation (engl. adaptive Matrix-Vector Multiplication; AMVM)

1. Starte mit einer groben $\mathcal{H}$-Matrix Approximation A_0 von A und setze $k = 0$.
2. Berechne $b_k = A_k x$ und $\hat{b}_k = \hat{A}_k x$.
3. a) Finde bei gegebenen $0 < \theta < 1$ eine Menge markierter Blöcke $P_k \subset P$ von minimaler Mächtigkeit, sodass
$$\gamma_k - \gamma_k(P_k) \geq \theta \gamma_k, \tag{6.2}$$
wobei $\gamma_k(Q) := \| \sum_{t\times s \in P\setminus Q} (A_k - \hat{A}_k)_{ts} x_s \|_2$ und $\gamma_k = \gamma_k(\emptyset) = \|b_k - \hat{b}_k\|_2$.
   b) Use the following strategy to construct P_k:
      (i) Sortiere die Fehler $|(b_k - \hat{b}_k)_i|$, $i = 1, ..., M$, in absteigender Reihenfolge.
      (ii) Gehe die geordneten Fehler schrittweise von oben nach unten durch und ermittele die zur Zeile i gehörenden Blöcke.
      (iii) Füge jeden Block $t \times s$ zu P_k hinzu, wenn $|[(A_k - \hat{A}_k)_{ts} x_s]_i| \geq (1-\theta)(c_{\mathrm{sp}} LMN)^{-1/2} \gamma_k$ gilt.
      (iv) Erweitere P_k gemäß (ii) und (iii), solange die Bedingung (6.2) nicht erfüllt ist.
4. Setze
$$A_{k+1} = \begin{cases} (\hat{A}_k)_b, & b \in P_k, \\ (A_k)_b, & b \in P \setminus P_k. \end{cases}$$
5. Falls $\gamma_k > \varepsilon_{\mathrm{AMVM}}$ erhöhe k und gehe zu 2.

Auf den ersten Blick benutzt Algorithmus 2 zwei hierarchische Matrizen A_k und $\hat{A}_k$. Da diese beiden Matrizen stark miteinander verbunden sind, genügt es, die feinere Approximation $\hat{A}_k$ zu speichern. Außerdem erlaubt dieser Ansatz, die Struktur des zu multiplizierenden Vektors x bei der Approximation von A zu berücksichtigen. Aufgrund der Auswahlkriterien von P_k in Algorithmus 2 haben Cluster von Nulleinträgen im Vektor x zur Folge, dass die zugehörigen Blöcke gar nicht approximiert werden müssen.

Bemerkung 6.1. *Man beachte, dass der vorherige Algorithmus endet, wenn entweder in Schritt 3 b) (iv) die Bedingung* (6.2) *erfüllt ist oder wenn die Liste der Blöcke ihr Ende erreicht hat. In diesem Fall ist auch* (6.2) *gültig, denn die in Schritt 3 b) iii) verwendete Bedingung impliziert* $|[(A_k - \hat{A}_k)_{ts} x_s]_i| \leq (1-\theta)(c_{\mathrm{sp}} LMN)^{-1/2} \gamma_k$ *für alle Blöcke* $t \times s \in P \setminus P_k$ *und somit*

$$\begin{aligned} \gamma_k(P_k) = \| \sum_{t\times s \in P\setminus P_k} (A_k - \hat{A}_k)_{ts} x_s \|_2 &= \left(\sum_{i=1}^{M} | \left[\sum_{t\times s \in P\setminus P_k} (A_k - \hat{A}_k)_{ts} x_s \right]_i |^2 \right)^{1/2} \\ &\leq \left(\sum_{i=1}^{M} c_{\mathrm{sp}} L \sum_{t\times s \in P\setminus P_k} | \left[(A_k - \hat{A}_k)_{ts} x_s \right]_i |^2 \right)^{1/2} \\ &\leq \left(\sum_{i=1}^{M} \sum_{t\times s \in P\setminus P_k,\, i\in t} (1-\theta)^2 (MN)^{-1} \gamma_k^2 \right)^{1/2} \leq (1-\theta)\gamma_k. \end{aligned}$$

Der neu eingeführte Algorithmus soll nun genauer untersucht werden. Zunächst betrachten wir mit der Zuverlässigkeit und Effizienz zwei grundlegende Eigenschaften eines jeden Fehlerschätzers.

Lemma 6.2. *Sei Annahme* (6.1) *erfüllt. Dann ist der Fehlerschätzer γ_k zuverlässig, dass heißt, es gilt die Abschätzung*

$$\|b_k - b\|_2 \leq c_{rel}\,\gamma_k,$$

wobei $c_{rel} := \frac{1}{1-c_{\mathrm{sat}}}$.

Beweis. Mit der Saturierungsannahme folgt

$$\|b_k - b\|_2 \leq \|b_k - \hat{b}_k\|_2 + \|\hat{b}_k - b\|_2 \leq \gamma_k + c_{\mathrm{sat}}\|b_k - b\|_2$$

und somit

$$\|b_k - b\|_2 \leq c_{\mathrm{rel}}\gamma_k, \quad c_{\mathrm{rel}} := \frac{1}{1 - c_{\mathrm{sat}}}.$$

□

Lemma 6.3. *Sei Annahme* (6.1) *erfüllt. Dann ist γ_k effizient, d.h. es gilt die Abschätzung*

$$c_{\mathrm{eff}}\gamma_k \leq \|b_k - b\|_2,$$

wobei $c_{eff} := \frac{1}{1+c_{\mathrm{sat}}}$.

Beweis. Es folgt wiederum mit der Saturierungsannahme, dass

$$\gamma_k = \|b_k - \hat{b}_k\|_2 \leq \|b_k - b\|_2 + \|b - \hat{b}_k\|_2 \leq (1 + c_{\mathrm{sat}})\|b_k - b\|_2$$

and damit

$$c_{\mathrm{eff}}\gamma_k \leq \|b_k - b\|_2, \quad c_{\mathrm{eff}} := \frac{1}{1 + c_{\mathrm{sat}}}.$$

□

Im Folgenden wird die Konvergenz des Verfahrens zur adaptiven Matrix-Vektor Multiplikation untersucht.

6.2 Konvergenz der adaptiven Matrix-Vektor-Multiplikation

Vor allem die Zuverlässigkeit von γ_k wird hilfreich sein, um die Konvergenz der adaptiven Matrix-Vektor-Multiplikation zu zeigen. Dafür müssen wir zunächst wissen, wie das Konvergenzverhalten des Fehlerschätzers γ_k selbst ist. Wir starten mit der Untersuchung des Fehlers $\hat{e}_k := \|\hat{b}_k - \hat{b}_{k+1}\|_2^2$.

Lemma 6.4. *Seien $\hat{b}_k = \hat{A}_k x$ und $\hat{b}_{k+1} = \hat{A}_{k+1}x$. Dann konvergiert der Fehler $\hat{e}_k = \|\hat{b}_k - \hat{b}_{k+1}\|_2^2$ gegen Null für $k \to \infty$.*

Beweis. Für $\hat{e}_k$ gilt:

$$\hat{e}_k = \|\hat{A}_k x - \hat{A}_{k+1}x\|_2^2 \leq \|\hat{A}_k - \hat{A}_{k+1}\|_2^2\|x\|_2^2,$$

wobei das Konvergenzverhalten von $\hat{e}_k$ aus der Konvergenz des ACA folgt. Denn hier gilt entweder $\lim_{k\to\infty}(A_k)_b = A_b$ oder $b \notin M_k$ ab einem Index $k_0 \in \mathbb{N}$, was $(A_k)_b = (A_{k+1})_b$ für $k \geq k_0$ impliziert. Damit gilt

$$\lim_{k\to\infty} A_k = A_* \in \mathbb{R}^{N\times N}$$

und

$$\lim_{k\to\infty} \|\hat{A}_{k+1} - \hat{A}_k\| = 0.$$

□

Die Konvergenz des Fehlerschätzers γ_k ist eine Konsequenz des sogenannten Fehlerschätzerreduktionsprinzips (6.3) (siehe [8, 38] für die Anwendung dieses Konzepts in der adaptiven Randelementmethode).

Lemma 6.5. *(Fehlerschätzerreduktion) Seien $s > 1$ und $1 - \frac{1}{\sqrt{s}} < \theta < 1$ gegeben. Dann erfüllt der Fehlerschätzer γ_k das folgende Reduktionsprinzip*

$$\gamma_{k+1}^2 \leq c_1\gamma_k^2 + c_2\hat{e}_k^2, \tag{6.3}$$

wobei $c_1 = c_1(s) = \frac{1}{s} < 1$ und $c_2 = c_2(s,\theta) = \frac{1}{1-s(1-\theta)^2}$ und $\hat{e}_k$ gegen Null konvergiert. Zudem gilt

$$\lim_{k\to\infty} \gamma_k = 0.$$

Beweis. Um eine Reduktion des Fehlerschätzers zu zeigen, betrachten wir γ_{k+1} genauer. Mit der Youngschen Ungleichung für $\delta > 0$ folgt:

$$\begin{aligned}
\gamma_{k+1}^2 &= \|b_{k+1} - \hat{b}_{k+1}\|^2 \\
&= \|b_{k+1} - \hat{b}_k + \hat{b}_k - \hat{b}_{k+1}\|^2 \leq \left(\|b_{k+1} - \hat{b}_k\| + \|\hat{b}_k - \hat{b}_{k+1}\|\right)^2 \\
&\leq (1+\delta)\underbrace{\|b_{k+1} - \hat{b}_k\|_2^2}_{=:e} + (1+\delta^{-1})\hat{e}_k
\end{aligned}$$

Falls die ersten Summanden in die ausgewählten und nicht ausgewählten Blöcke aufgeteilt werden, erhalten wir die folgende Fehlerschätzerreduktion:

$$\begin{aligned}
e = \|b_{k+1} - \hat{b}_k\| &= \|\sum_{t\times s\in P} (A_{k+1} - \hat{A}_k)_{ts} x_s\| \\
&= \|\sum_{t\times s\in P_k} (A_{k+1} - \hat{A}_k)_{ts} x_s + \sum_{t\times s\in P\setminus P_k} (A_{k+1} - \hat{A}_k)_{ts} x_s\| \\
&= \|\sum_{t\times s\in P\setminus P_k} (A_k - \hat{A}_k)_{ts} x_s\| = \gamma_k(P_k) \leq (1-\theta)\gamma_k.
\end{aligned}$$

Mit der Wahl $\delta = \frac{1-s(1-\theta)^2}{s(1-\theta)^2}$, wobei $\delta > 0$ aufgrund der Voraussetzung erfüllt ist, folgt, dass

$$\begin{aligned}
\gamma_{k+1}^2 &= (1+\delta)(1-\theta)^2\gamma_k^2 + (1+\delta^{-1})\hat{e}_k^2 \\
&= \left(1 + \frac{1-s(1-\theta)^2}{s(1-\theta)^2}\right)(1-\theta)^2\gamma_k^2 + \left(1 + \frac{s(1-\theta)^2}{1-s(1-\theta)^2}\right)\hat{e}_k^2 \\
&= \frac{1}{s}\gamma_k^2 + \frac{1}{1-s(1-\theta)^2}\hat{e}_k^2.
\end{aligned}$$

Mit $c_1(s) := \frac{1}{s} < 1$ und $c_2(s,\theta) := \frac{1}{1-s(1-\theta)^2}$ folgt die Behauptung.
Der zweite Teil der Behauptung wird mit Lemma 6.4 und dem Fehlerschätzerreduktionsprinzip,

welches in [8] eingeführt wurde, bewiesen. Sei $\hat{E} > 0$ eine Zahl mit $\hat{e}_k \leq \hat{E}$ für alle k. Mit der Fehlerschätzerreduktion folgt:

$$\begin{aligned}\gamma_{k+1}^2 &\leq c_1\gamma_k^2 + c_2\hat{e}_k \leq c_1(c_1\gamma_{k-1}^2 + c_2\hat{e}_{k-1}) + c_2\hat{e}_k \\ &\vdots \\ &\leq c_1^{k+1}\gamma_0^2 + c_2\sum_{i=0}^{k} c_1^{k-i}\hat{e}_k \\ &\leq c_1^{k+1}\gamma_0^2 + c_2\hat{E}\sum_{l=0}^{k} c_1^l \leq \gamma_0^2 + \frac{c_2\hat{E}}{1-c_1}.\end{aligned}$$

Demnach ist die Folge $\{\gamma_k\}_{k\in\mathbb{N}_0}$ beschränkt und es lässt sich $M := \limsup_{k\to\infty} \eta_k^2$ definieren. Eine weitere Anwendung der Fehlerschätzerreduktion führt zusammen mit Lemma 6.4 zu

$$E = \limsup_{k\to\infty} \gamma_{k+1}^2 \leq c_1 \limsup_{k\to\infty} \gamma_k^2 + c_2 \limsup_{k\to\infty} \hat{e}_k = c_1 E.$$

Damit ist $E = 0$ und wir erhalten

$$0 \leq \liminf_{k\to\infty} \gamma_k \leq \limsup_{k\to\infty} \gamma_k = 0.$$

Schließlich folgt

$$\lim_{k\to\infty} \gamma_k = 0.$$

□

Zusammen mit der Zuverlässigkeit des Fehlerschätzers lässt sich die Konvergenz der adaptiven Matrix-Vektor Multiplikation zeigen.

Theorem 6.6. *Seien die Voraussetzungen von Lemma 6.5 erfüllt. Dann konvergiert der Fehler $\|b_k - b\|_2$ der durch Algorithmus 2 konstruierten Folge $\{b_k\}_{k\in\mathbb{N}}$ gegen Null.*

Beweis. Mit Lemma 6.2 and Lemma 6.5 erhalten wir

$$\|b_k - b\|_2 \leq c_{\text{rel}}\gamma_k \to 0, \quad k \to 0.$$

□

Da die Approximation der Matrix A stark auf den zu multiplizierenden Vektor x zugeschnitten ist, macht diese Art von Approximation nur Sinn, wenn sich x nicht ändert. Eine typische Situation ist zum Beispiel die Konstruktion der rechten Seite bei der Randintegralmethode. Diese wird später in Kapitel 9 nochmals aufgegriffen.

Auch bei iterativen Lösungsverfahren wie dem Verfahren der konjugierten Gradienten treten Matrix-Vektor-Multiplikationen auf. Da sich in diesem Fall der zu multiplizierende Vektor in jedem Iterationsschritt ändert, verursacht AMVM keine Vorteile, wohl eher noch einen erhöhten Zeitaufwand. Soll also eine adaptive Approximation der Systemmatrix auch bei der iterativen Lösung eines linearen Gleichungssystem zum Einsatz kommen, müssen wir uns einen etwas anderen Ansatz überlegen. Die folgenden beiden Kapitel zeigen, wie auch in diesem Fall eine adaptive Approximation erzeugt werden kann.

7. Zusätzliche adaptive Methoden für die adaptive Kreuzapproximation

Nachdem wir uns im letzten Kapitel mit der Approximation einer Matrix bei der Matrix-Vektor-Multiplikation beschäftigt haben, konzentrieren wir uns im sechsten Kapitel auf die Approximation einer Systemmatrix bei einem Gleichungssystem $Ax = b$. Die Matrix A wird dabei vorrangig aus der Finite-Elemente-Diskretisierung eines in integraler Form dargestellten Differentialoperators resultieren. Später in Kapitel acht und neun werden wir auf diese Thematik nochmals zurückkommen und die betrachteten Problemstellungen formulieren. Für den Moment werden die üblichen Voraussetzungen an das ACA-Verfahren angenommen. Mit der ACA existiert bereits ein Algorithmus, mit dem derartige Matrizen angenähert werden können, ohne jemals die volle Matrix aufstellen zu müssen. In Abschnitt 5.4 haben wir jedoch auch gesehen, dass sich die Approximation trotz unterschiedlicher rechter Seiten b nicht ändert. Daher soll im Folgenden eine Abhängigkeit zwischen der ACA und der rechten Seite des diskretisierten Problems erzeugt werden. Ein derartiges Vorgehen ist hauptsächlich bei Anwendungen vorteilhaft, in denen eine feste rechte Seite betrachtet wird. Bei sich ändernden rechten Seiten hat die Universalität der Approximation bei der ACA mehr Vorteile.

Im nachfolgenden Abschnitt 7.1 wird eine Erweiterung der adaptiven Kreuzapproximation vorgestellt. Diese werde mit exACA (engl.: extended adaptive cross approximation) bezeichnet. Anschließend soll im siebten Kapitel die neue Methdoe exACA mit einem iterativen Lösungsverfahren kombiniert werden, was den finalen Stand der Erweiterung der ACA darstellt. Die Kombination aus Lösungsverfahren und exACA werde BACA (engl.: block-adaptive cross approximation) genannt.

7.1 Residuale block-basierte Fehlerschätzer und die erweiterte adaptive Kreuzapproximation

Der übliche Weg, diskretisierte Probleme mit voll besetzter Matrix zu behandeln, ist die Konstruktion einer hierarchischen Matrixapproximation $\tilde{A}$, sodass jeder Block $\tilde{A}_b$, $b \in P$, die Bedingung

$$\|A_b - \tilde{A}_b\|_F \leq \varepsilon \|A_b\|_F, \quad \text{für alle } b \in P, \tag{7.1}$$

erfüllt. Wegen der Unabhängigkeit der Blöcke kann dies sogar parallel erfolgen, siehe [19]. Hierarchische Matrixapproximationen bzw. Niedrigrang-Approximationen können auf viele unterschiedliche Arten erzeugt werden. Die Methode der Wahl in unserem Fall ist die ACA. Während (7.1) lokale Eigenschaften der Approximation garantiert, sind die globalen Auswirkungen dieser Bedingung schwer abzuschätzen. Nichtsdestotrotz impliziert die Abschätzung (7.1) die globale Eigenschaft

$$\|A - \tilde{A}\|_F \leq \varepsilon \|A\|_F.$$

Da die Eigenvektoren, welche zu kleinen Eigenwerten von A bzw. $\tilde{A}$ gehören, in der Regel weniger genau als ε sind, müssen geeignete Techniken eingesetzt werden, um die spektrale Äquivalenz von A und $\tilde{A}$ zu gewährleisten, siehe [16]. Zudem kann (7.1), ohne dass zusätzliche Stabilisierungsmethoden angewendet werden (siehe [18]), die positive Definitheit der Matrix A nicht für $\tilde{A}$ erhalten. Demnach wird die Niedrigrang-Approximation üblicherweise ohne Rücksicht auf die Bedeutung des jeweiligen Blocks auf globale Eigenschaften der Matrix konstruiert.

Die von der ACA erworbene Matrixapproximation $\tilde{A}$ muss nicht zwingend die beste Approximation an A sein, falls die rechte Seite oder die Norm des Fehlers der zugehörigen Lösung das Maß ist. Um hierarchische Matrixapproximationen zu finden, welche für die jeweilige Problemstellung besser geeignet sind, wenden wir aus der Theorie der Adaptivität bekannte Techniken mit geeigneten Fehlerschätzern an. Die Strategie, Fehler zu schätzen, ist in Zusammenhang mit numerischen Methoden für partielle Differentialgleichungen, d.h. die adaptive Finite Elemente Methode (z.B. [29]), oder

Integralgleichungen, d.h. die adaptive Randintegralmethode (z.B. [39, 50]), gut bekannt. Inzwischen gibt es zahlreiche a posteriori Fehlerschätzer. Die Methode hier basiert wie auch in Kapitel 5 auf der in [38] eingeführten $(h-h/2)$ Version. Die dort zu findenden adaptiven Methoden konzentrieren sich vor allem auf die Verfeinerung des Gitters. Bei der im Folgenden vorgestellten Erweiterung der ACA werden ähnliche Strategien genutzt, um sukzessive die blockweisen Niedrigrang-Approximationen zu verbessern, wobei sich das zugrunde liegende Gitter und die zugrunde liegende Blockstruktur P nicht ändern. Dementsprechend approximieren das ACA-Verfahren und dessen Erweiterung den Operator auf demselben Gitter.

Im Gegensatz dazu, die Matrixapproximation $\tilde{A}$ von A auf die übliche Art mit ACA zu konstruieren und ein einziges lineares Gleichungssystem $\tilde{A}x = b$ zu lösen, generieren wir eine Folge von $\mathcal{H}$-Matrixapproximationen A_k von A und lösen jedes lineare Gleichungssystem $A_k x_k = b$ für x_k. Auf den ersten Blick sieht dieses Vorgehen teurer aus, als die Gleichung $\tilde{A}x = b$ nur einmal zu lösen. Die Approximation von A_{k+1} kann jedoch durch die approximierte Lösung x_k gesteuert und x_{k+1} kann als Update von x_k berechnet werden. Zudem muss die Genauigkeit von x_k für kleine k nicht hoch sein. Dieses Vorgehen ermöglicht die Adaptierung an den geschätzten Fehler.

Die folgende Methode basiert auf residualen Fehlerschätzern und folgt einer ähnlichen Idee wie in Kapitel 5. Dazu bezeichne $\hat{A}_k$ eine bessere Approximation an A als A_k, d.h. wir nehmen an, dass die Saturierungsbedingung

$$\|\hat{A}_k x_k - A x_k\|_2 \leq c_{\text{sat}} \|A_k x_k - A x_k\|_2 \tag{7.2}$$

für $0 < c_{\text{sat}} < 1$ erfüllt ist. Eine natürliche Wahl von $\hat{A}_k$ ist die bessere Approximation, welche aus A_k resultiert, indem eine feste Anzahl an zusätzlichen ACA-Schritten auf jeden zulässigen Block angewendet wird und $(\hat{A}_k)_b = A_b$ gesetzt wird für alle nicht zulässigen Blöcke $b \in P_{\text{non-adm}}$. Als Fehlerschätzer wählen wir

$$\eta_k^2(P) := \sum_{t \times s \in P} \|(A_k - \hat{A}_k)_{ts}(x_k)_s\|_2^2, \tag{7.3}$$

wobei P unsere bisherige Partition der Matrixindices in Blöcke bezeichnet. Damit lautet der Algorithmus zur erweiterten adaptiven Kreuzapproximation:

Algorithmus 3 Erweiterte adaptive Kreuzapproximation (engl. extended adaptive cross approximation; exACA)

1. Starte mit einer groben $\mathcal{H}$-Matrixapproximation A_0 von A und setze $k = 0$.
2. Löse das lineare Gleichungssystem $A_k x_k = b$ für x_k.
3. Finde bei gegebenen $0 < \theta < 1$ eine Menge markierter Blöcke $M_k \subset P$ mit minimaler Mächtigkeit, sodass
$$\eta_k(M_k) \geq \theta \, \eta_k, \tag{7.4}$$
erfüllt ist.
4. Verbessere A_k, indem dieselben Schritte auf alle Blöcke $(A_k)_{ts}$ mit $t \times s \in M_k$ angewendet werden, wie zur Erzeugung von $(\hat{A}_k)_{ts}$ angewendet wurden. Bei Blöcken $t \times s \in P \setminus M_k$ setze $(A_{k+1})_{ts} = (A_k)_{ts}$, d.h. setze
$$A_{k+1} = \begin{cases} (\hat{A}_k)_{ts}, & t \times s \in M_k, \\ (A_k)_{ts}, & t \times s \in P \setminus M_k. \end{cases}$$
5. Falls $\eta_k > \varepsilon_{\text{exACA}}$, erhöhe $k := k + 1$ und gehe zu 2.

Die ACA-Schritte, welche zur Konstruktion von A_{k+1} benötigt werden, können ohne großen Aufwand von $\hat{A}_k$ übernommen werden, und $\hat{A}_{k+1}$ kann durch die Verbesserung der Blöcke in M_k aus $\hat{A}_k$ berechnet werden. Daher müssen nur zu Beginn alle zulässigen Blöcke in ihrer Genauigkeit erhöht werden. Nach dem ersten Schritt arbeitet exACA nur noch auf den im jeweiligen Schritt ausgewählten Blöcken.

Die ursprüngliche ACA - beschrieben in Algorithmus 1 - enthält mit dem Abbruchparamter $\varepsilon_{\mathrm{ACA}}$ ein Abbruchkriterium, welches die gewünschte Genauigkeit der Niedrigrang-Approximation für den entsprechenden Block beschreibt. Während die Angabe einer Fehlerschranke $\varepsilon_{\mathrm{ACA}}$, damit die Lösung des linearen Gleichungssystems einer vorgeschriebenen Genauigkeit genügt, schwierig ist, gibt der Parameter $\varepsilon_{\mathrm{exACA}}$, wie wir in Lemma 7.1 sehen werden, eine obere Schranke an den residualen Fehler $\|b - Ax_k\|_2$ von x_k vor.

Neben der Fähigkeit des Schätzers η_k den Fehler auf den einzelnen Blöcken zu lokalisieren, wollen wir einige Eigenschaften wie die Zuverlässigkeit und die Effizienz des Fehlerschätzers untersuchen. Die im folgenden Lemma gezeigte Zuverlässigkeit von η_k wird maßgeblich für die Konvergenz der erweiterten adaptiven Kreuzapproximation verantwortlich sein.

Lemma 7.1. *Für zwei Matrixapproximation A_k und $\hat{A}_k$ an A und für die Lösung x_k von $A_k x_k = b$ gelte die Saturierungsbedingung* (7.2)*. Dann schätzt η_k zuverlässig den Fehler $\|b - Ax_k\|_2$, d.h. es gilt*

$$\|b - Ax_k\|_2 \leq \frac{\sqrt{c_{\mathrm{sp}} L}}{1 - c_{\mathrm{sat}}} \eta_k.$$

Beweis. Zunächst sei angemerkt, dass

$$\left(\sum_{i=1}^{n} a_i\right)^2 \leq n \sum_{i=1}^{n} a_i^2 \quad \text{für alle } a_i \in \mathbb{R},\ i = 1, \ldots, n,$$

nach der Cauchy-Schwarzschen-Ungleichung erfüllt ist.
Um die geforderte Abschätzung zu zeigen, nutzen wir die Aufteilung von $A \in \mathcal{H}(P, k)$ in eine Summe von Level-Matrizen $A^{(\ell)}$, d.h.

$$A = \sum_{\ell=1}^{L} A^{(\ell)}.$$

Damit erhalten wir:

$$\begin{aligned}
\|(A_k - \hat{A}_k)x_k\|_2^2 &\leq \left(\sum_{\ell=1}^{L} \|(A_k^{(\ell)} - \hat{A}_k^{(\ell)})x_k\|_2\right)^2 \leq L \sum_{\ell=1}^{L} \|(A_k^{(\ell)} - \hat{A}_k^{(\ell)})x_k\|_2^2 \\
&= L \sum_{\ell=1}^{L} \sum_{t \in T^{(\ell)}} \| \sum_{s : t \times s \in P} (A_k - \hat{A}_k)_{ts} (x_k)_s \|_2^2 \\
&\leq L \sum_{\ell=1}^{L} \sum_{t \in T^{(\ell)}} \left(\sum_{s : t \times s \in P} \|(A_k - \hat{A}_k)_{ts} (x_k)_s\|_2 \right)^2 \\
&\leq c_{\mathrm{sp}} L \sum_{\ell=1}^{L} \sum_{t \in T^{(\ell)}} \sum_{s : t \times s \in P} \|(A_k - \hat{A}_k)_{ts} (x_k)_s\|_2^2 \\
&= c_{\mathrm{sp}} L \sum_{t \times s \in P} \|(A_k - \hat{A}_k)_{ts} (x_k)_s\|_2^2 \\
&= c_{\mathrm{sp}} L \, \eta_k^2.
\end{aligned}$$

Mit der Saturierungsbedingung (7.2) und

$$\|b - Ax_k\|_2 = \|A_k x_k - Ax_k\|_2 \leq \|(A_k - \hat{A}_k)x_k\|_2 + \|\hat{A}_k x_k - Ax_k\|_2 \leq \sqrt{c_{\text{sp}} L}\, \eta_k + c_{\text{sat}} \|b - Ax_k\|_2$$

folgt die Abschätzung

$$\|b - Ax_k\|_2 \leq \frac{\sqrt{c_{\text{sp}} L}}{1 - c_{\text{sat}}}\, \eta_k.$$

□

Das nächste Resultat beschreibt nicht direkt die Effizienz von η_k, d.h. $\eta_k \leq c\|b - Ax_k\|_2$, wobei $c > 0$ eine Konstante ist. Der Beweis der Effizienz ist noch eine offene Fragestellung. Mit $\|(A_k - \hat{A}_k)x_k\|_2$ kann jedoch ein leicht berechenbarer Ausdruck angegeben werden, welcher als untere Schranke für $\|b - Ax_k\|_2$ dient.

Lemma 7.2. *Es gelte die Saturierungsannahme* (7.2)*. Dann gilt die Abschätzung*

$$\|b - Ax_k\|_2 \geq \frac{1}{1 + c_{\text{sat}}} \|(A_k - \hat{A}_k)x_k\|_2.$$

Beweis. Mit der Saturierungsannahme (7.2) und der Annahme, dass $A_k x_k = b$ exakt gelöst wird, erhalten wir

$$\begin{aligned} \|(A_k - \hat{A}_k)x_k\|_2 &\leq \|A_k x_k - Ax_k\|_2 + \|Ax_k - \hat{A}_k x_k\|_2 \leq (1 + c_{\text{sat}})\|A_k x_k - Ax_k\|_2 \\ &\leq (1 + c_{\text{sat}}) \left(\|b - A_k x_k\|_2 + \|b - Ax_k\|_2\right) \\ &\leq (1 + c_{\text{sat}})\|b - Ax_k\|_2. \end{aligned}$$

Damit folgt

$$\|b - Ax_k\|_2 \geq \frac{1}{1 + c_{\text{sat}}} \|(A_k - \hat{A}_k)x_k\|_2.$$

□

Bemerkung 7.3. *Um die Effizienz des Fehlerschätzers η_k zu zeigen, müsste die Abschätzung*

$$\|(A_k - \hat{A}_k)x_k\|_2 = \|\sum_{t \times s \in P} (A_k - \hat{A}_k)_{ts} (x_k)_2\|_2 \geq \sum_{t \times s \in P} \|(A_k - \hat{A}_k)_{ts} (x_k)_s\|_2$$

in Form einer umgekehrten Dreiecksungleichung nachgewiesen werden. Aufgrund der Problemstellung und der Struktur der Matrix A sollte eine derartige Abschätzung möglich sein. Bisher konnten die inneren Abhängigkeiten in A nicht weit genug aufgeschlüsselt werden. Vermutlich spielt hier die zugrunde liegende Problemstellung eine entscheidende Rolle. Ein weiterer Ansatz orientiert sich an Gebietszerlegungsmethoden, wobei diese wiederum auf dem betrachteten Gebiet und nicht auf der Matrix selbst arbeiten. Dort bedient man sich der Annahme einer stabilen Gebietszerlegung, welches ein ähnliches Problem wie hier löst, siehe [72].

Die exACA-Methode lässt nur dieselbe (genauere) Blockapproximation zu, welche auch bei der Schätzung des Fehlers verwendet wurde. In manchen Situationen würde man eher nur einen weiteren Rang benutzen, um die genauere Approximation zu erzeugen, aber die Blockapproximation schließlich um zwei Ränge verbessern. Als nächsten Schritt, bevor die Konvergenz des Verfahrens analysiert wird, wollen wir daher noch eine relaxierte Version der erweiterten ACA (rexACA; engl.: relaxed extendend adaptive cross approximation) entwickeln, mit der die beiden verwendeten Blockapproximationen von

A_k und $\hat{A}_k$ verschieden voneinander behandelt werden können. Wir führen mit $0 < \omega < 1$ einen weiteren Parameter ein und fordern, dass bei der Verbesserung ausgewählter Blöcke die Bedingung

$$\|(A_{k+1} - \hat{A}_{k+1})_{ts}(x_k)_2\|_2 \leq \omega \|(A_k - \hat{A}_k)_{ts}(x_k)_s\|_2 \tag{7.5}$$

für alle $t \times s \in M_k$ erfüllt ist. Fügen wir Bedingung (7.5) im vierten Schritt des Algorithmus 3 ein, so folgt die relaxierte Form des exACA in Algorithmus 4.

Algorithmus 4 Relaxierte erweiterte adaptive Kreuzapproximation (engl. relaxed extended adaptive cross approximation; rexACA)

1. Starte mit einer groben $\mathcal{H}$-Matrixapproximation A_0 von A und setze $k = 0$.
2. Löse das lineare Gleichungssystem $A_k x_k = b$ für x_k.
3. Finde bei gegebenen $0 < \theta < 1$ eine Menge markierter Blöcke $M_k \subset P$ mit minimaler Mächtigkeit, sodass
$$\eta_k(M_k) \geq \theta \, \eta_k, \tag{7.6}$$
erfüllt ist.
4. Verbessere A_k, indem weitere Schritte des ACA auf alle Blöcke $(A_k)_{ts}$ mit $t \times s \in M_k$ angewendet werden, sodass für das Ergebnis $(A_{k+1})_{ts}$ gilt, dass
$$\|(A_{k+1} - \hat{A}_{k+1})_{ts}(x_k)_s\|_2 \leq \omega \|(A_k - \hat{A}_k)_{ts}(x_k)_s\|_2 \tag{7.7}$$
ist, wobei $0 < \omega < 1$ gegeben ist. Bei Blöcken $t \times s \in P \setminus M_k$ setze $(A_{k+1})_{ts} = (A_k)_{ts}$.
5. Falls $\eta_k > \varepsilon_{\text{rexACA}}$, erhöhe $k := k + 1$ und gehe zu 2.

Der Vorteil des rexACA im Gegensatz zum exACA besteht darin, dass wir verschiedene Strategien im Zusammenhang mit der genaueren Approximation und der Verbesserung von A_k zu A_{k+1} verwenden können. Zudem erhält man durch Bedingung (7.7) mehr Kontrolle über den Prozess selbst. Auf der anderen Seite muss hier eine zusätzliche Bedingung überprüft werden, was zusätzlichen Aufwand bedeutet und die Rechenzeit erhöht. Wie auch beim exACA gibt der Parameter $\varepsilon_{\text{rexACA}}$ eine obere Schranke an den residualen Fehler $\|b - Ax_k\|$ von x_k vor.

In den numerischen Tests zu den neu entwickelten Methoden, welche in Kapitel 11 zu finden sind, werden wir uns auf den äquivalenten Fehler der Lösung der eigentlichen Problemstellung beziehen, siehe nachfolgende Bemerkung 7.4.

Bemerkung 7.4. *Bezeichne $\mathcal{A} : V \to V'$ den Operator, welcher durch A diskretisiert wird, und u_h die Galerkin-Approximation der exakten Lösung von $\mathcal{A}u = f$, dann gilt für den residualen Fehler $\|f - \mathcal{A}u_h\|_{V'}$, dass*

$$c_B \|u - u_h\|_V \leq \|f - \mathcal{A}u_h\|_{V'} = \sup_{\varphi \in V} \frac{|a(u - u_h, \varphi)|}{\|\varphi\|_V} \leq c_S \|u - u_h\|_V$$

erfüllt ist, vorausgesetzt, die zu $\mathcal{A}$ gehörende Bilinearform $a : V \times V \to \mathbb{R}$ ist stetig, d.h.

$$|a(u, v)| \leq c_S \|u\|_V \|v\|_V, \quad \text{für alle } u, v \in V,$$

und genügt einer inf-sup-Bedingung

$$\inf_{u \in V} \sup_{v \in V} \frac{|a(u, v)|}{\|u\|_V \|v\|_V} \geq c_B.$$

Damit ist der Fehler der Lösung $\|u - u_h\|_V$ äquivalent zum residualen Fehler $\|f - \mathcal{A}u_h\|_{V'}$. Anstelle der Dualnorm wird im Diskreten häufig die euklidische Norm verwendet. Dies hat den Vorteil, dass dann die Konditionszahl mit in die Abschätzung eingeht.

7.2 Konvergenzanalyse

Um die Konvergenz der beiden Algorithmen 3 und 4 nachweisen zu können, muss gezeigt werden, dass der residuale Fehler gegen Null konvergiert, d.h.

$$\|b - Ax_k\|_2 \to 0, \quad k \to \infty.$$

Unter Benutzung der Zuverlässigkeit von η_k genügt es hierfür, die Konvergenz des Fehlerschätzers zu zeigen. Diese kann wiederum mit Hilfe eines Fehlerschätzerreduktionspronzips (eingeführt in [8]) hergeleitet werden. In unserem Fall folgt dieses Prinzip aus der Konvergenz des ACA, denn hier gilt (wie in Lemma 6.4) entweder $\lim_{k\to\infty}(A_k)_b = A_b$ oder $b \notin M_k$ ab einem Index $k_0 \in \mathbb{N}$, was $(A_k)_b = (A_{k+1})_b$ für $k \geq k_0$ impliziert. Damit gilt

$$\lim_{k\to\infty} A_k = A_* \in \mathbb{R}^{N\times N} \tag{7.8}$$

und

$$\lim_{k\to\infty} \|A_{k+1} - A_k\| = 0 = \lim_{k\to\infty} \|\hat{A}_{k+1} - \hat{A}_k\|.$$

Wie im nächsten Lemma zu sehen ist, ist das genutzte Dörfler Marking (7.6) ein weiterer Grund für die Konvergenz des Fehlerschätzers.

Lemma 7.5. *Angenommen A_* ist invertierbar. Dann gilt für den exACA aus Algorithmus 3 das Fehlerschätzerreduktionsprinzip*

$$\eta_{k+1}^2 \leq q\,\eta_k^2 + z_k, \tag{7.9}$$

wobei $q := 1 - \frac{1}{2}\theta^2 < 1$ und z_k konvergiert gegen Null. Zudem folgt: $\lim_{k\to\infty}\eta_k = 0$.

Beweis. Die Young'sche Ungleichung und die Dreiecksungleichung führen zu:

$$\begin{aligned}
\eta_{k+1}^2 &= \sum_{t\times s\in P} \|(A_{k+1} - \hat{A}_{k+1})_{ts}(x_{k+1})_s\|_2^2 \\
&\leq \sum_{t\times s\in P} \Big(\|(A_{k+1} - \hat{A}_k)_{ts}(x_k)_s\|_2 + \|(\hat{A}_k - \hat{A}_{k+1})_{ts}(x_k)_s\|_2 + \|(\hat{A}_{k+1})_{ts}(x_k - x_{k+1})_s\|_2\Big)^2 \\
&\leq (1+\delta) \sum_{t\times s\in P} \|(A_{k+1} - \hat{A}_k)_{ts}(x_k)_s\|_2^2 + \\
&\quad (1+1/\delta) \sum_{t\times s\in P} \Big(\|(\hat{A}_k - \hat{A}_{k+1})_{ts}(x_k)_s\|_2 + \|(\hat{A}_{k+1})_{ts}(x_k - x_{k+1})_s\|_2\Big)^2
\end{aligned}$$

für alle $\delta > 0$. Mit $A_k x_k = b = A_{k+1}x_{k+1}$ erhalten wir

$$\begin{aligned}
\sum_{t\times s\in P} \|(A_{k+1} - \hat{A}_{k+1})_{ts}(x_k)_s\|_2^2 &= \sum_{t\times s\in M_k} \|(A_{k+1} - \hat{A}_{k+1})_{ts}(x_k)_s\|_2^2 + \sum_{t\times s\in P\setminus M_k} \|(A_{k+1} - \hat{A}_{k+1})_{ts}(x_k)_s\|_2^2 \\
&\leq 0 + \sum_{t\times s\in P\setminus M_k} \|(A_k - \hat{A}_k)_{ts}(x_k)_s\|_2^2 \\
&= \eta_k^2(P \setminus M_k) = \eta_k^2 - \eta_k^2(M_k) \\
&\leq (1-\theta^2)\eta_k^2,
\end{aligned}$$

wobei wir (7.6) verwendet haben. Die Wahl $\delta := \frac{1}{2}\theta^2/(1-\theta^2)$, die Young'sche Ungleichung (mit $\rho > 0$) und

$$\begin{aligned}
z_k &= (1+1/\delta) \sum_{t\times s\in P} \Big(\|(\hat{A}_k - \hat{A}_{k+1})_{ts}(x_k)_s\|_2 + \|(\hat{A}_{k+1})_{ts}(x_k - x_{k+1})_s\|_2 \Big)^2 \\
&\leq (1+1/\delta) \left[(1+\rho) \sum_{t\times s\in P} \|(\hat{A}_k - \hat{A}_{k+1})_{ts}(x_k)_s\|_2^2 + (1+1/\rho) \sum_{t\times s\in P} \|(\hat{A}_{k+1})_{ts}(x_k - x_{k+1})_s\|_2^2 \right] \\
&\leq c_{\mathrm{sp}} L (1+1/\delta) \left[(1+\rho) \max_{t\times s\in P} \|(\hat{A}_k - \hat{A}_{k+1})_{ts}\|_2^2 \|x_k\|_2^2 + (1+1/\rho) \max_{t\times s\in P} \|(\hat{A}_{k+1})_{ts}\|_2^2 \|x_k - x_{k+1}\|_2^2 \right] \\
&\leq c_{\mathrm{sp}} L (1+1/\delta) \left[(1+\rho) \|(\hat{A}_k - \hat{A}_{k+1})\|_2^2 \|x_k\|_2^2 + (1+1/\rho) \|\hat{A}_{k+1}\|_2^2 \|x_k - x_{k+1}\|_2^2 \right] \\
&\leq c_{\mathrm{sp}} L (1+1/\delta) \left[(1+\rho) \|(\hat{A}_k - \hat{A}_{k+1})\|_2^2 \|x_k\|_2^2 + (1+1/\rho) \|\hat{A}_{k+1}\|_2^2 \|A_{k+1}^{-1}\|_2^2 \|A_{k+1}x_{k+1} - A_{k+1}x_k\|_2^2 \right] \\
&\leq c_{\mathrm{sp}} L (1+1/\delta) \left[(1+\rho) \|(\hat{A}_k - \hat{A}_{k+1})\|_2^2 \|x_k\|_2^2 + (1+1/\rho) \|\hat{A}_{k+1}\|_2^2 \|A_{k+1}^{-1}\|_2^2 \|A_k x_k - A_{k+1}x_k\|_2^2 \right] \\
&\leq c_{\mathrm{sp}} L (1+1/\delta) \left[(1+\rho) \|(\hat{A}_k - \hat{A}_{k+1})\|_2^2 \|x_k\|_2^2 + (1+1/\rho) \|\hat{A}_{k+1}\|_2^2 \|A_{k+1}^{-1}\|_2^2 \|A_k - A_{k+1}\|_2^2 \|x_k\|_2^2 \right]
\end{aligned}$$

liefern die erste Aussage der Behauptung, da wegen (7.8) und $A_k^{-1} \to A_*^{-1}$ für $k \to \infty$ die Folge $\{z_k\}_{k\in\mathbb{N}}$ gegen Null konvergiert und die Folge $\{x_k\}_{k\in\mathbb{N}}$ wegen

$$\|x_k\|_2 \leq \|A_k^{-1}\|_2 \|b\|_2$$

beschränkt ist.

Der zweite Teil der Behauptung folgt mit der gerade hergeleiteten Fehlerreduktion. Sei $z > 0$ eine Zahl mit $z_k \leq z$ für alle $k \in \mathbb{N}$. Dann folgt mit (7.9) die Abschätzung

$$\begin{aligned}
\eta_{k+1}^2 &\leq q\eta_k^2 + z_k \leq q^2\eta_{k-1}^2 + qz_{k-1} + z_k \leq \cdots \leq q^{k+1}\eta_0^2 + \sum_{i=0}^{k} q^{k-i} z_i \\
&\leq q^{k+1}\eta_0^2 + z \sum_{l=0}^{k} q^l \leq \eta_0^2 + \frac{z}{1-q}.
\end{aligned}$$

Damit ist die Folge $\{\eta_k\}_{k\in N}$ beschränkt und wir können die Zahl $M := \limsup_{k\to\infty} \eta_k^2$ definieren. Eine weitere Benutzung von (7.9) liefert

$$M = \limsup_{k\to\infty} \eta_{k+1}^2 \leq q \limsup_{k\to\infty} \eta_k^2 + \limsup_{k\to 0} z_k = qM.$$

Damit gilt $M = 0$ und wir erhalten

$$0 \leq \liminf_{k\to\infty} \eta_k \leq \limsup_{k\to\infty} \eta_k = 0.$$

Schließlich folgt

$$\lim_{k\to\infty} \eta_k = 0.$$

□

Der Nachweis der Konvergenz des rexACA aus Algorithmus 4 erfolgt nach demselben Prinzip. Hier erhalten wir jedoch bezogen auf die Konvergenz des Fehlerschätzers eine in Abhängigkeit von ω verminderte Konvergenzrate. Dies hat einen der folgenden beiden Gründe:

1. Falls zur Konstruktion der genaueren Approximation $\hat{A}_k$ mehr ACA-Schritte genutzt wurden als bei Bedingung (7.7) zur Erzeugung von A_{k+1}, so schöpfen wir nicht das volle Potential der genaueren Approximation aus und wir verlieren an Konvergenzgeschwindigkeit.

oder

2. Ist das Umgekehrte der Fall, d.h. zur Erzeugung von A_{k+1} werden mehr ACA-Schritte genutzt als bei der Generierung von $\hat{A}_{k+1}$, so ist aufgrund der unterschiedlichen Anzahl an ACA-Schritten nicht genau klar, was passiert. Somit kann theoretisch nur eine verminderte Konvergenzrate gezeigt werden.

Theorem 7.6. *Angenommen A_* ist invertierbar. Dann gilt für den rexACA aus Algorithmus 4 das Fehlerschätzerreduktionsprinzip*

$$\eta_{k+1}^2 \leq q\,\eta_k^2 + z_k,$$

wobei $q := 1 - \frac{1}{2}(1-\omega^2)\theta^2 < 1$ und z_k konvergiert gegen Null. Zudem folgt: $\lim_{k\to\infty} \eta_k = 0$.

Beweis. Die Young'sche Ungleichung und die Dreiecksungleichung führen zu:

$$\begin{aligned}
\eta_{k+1}^2 &= \sum_{t\times s\in P} \|(A_{k+1} - \hat{A}_{k+1})_{ts}(x_{k+1})_s\|_2^2 \\
&\leq \sum_{t\times s\in P} \Big(\|(A_{k+1} - \hat{A}_{k+1})_{ts}(x_k)_s\|_2 + \|(A_{k+1} - \hat{A}_{k+1})_{ts}(x_{k+1} - x_k)_s\|_2\Big)^2 \\
&\leq (1+\delta) \sum_{t\times s\in P} \|(A_{k+1} - \hat{A}_{k+1})_{ts}(x_k)_s\|_2^2 + (1+1/\delta) \sum_{t\times s\in P} \|(A_{k+1} - \hat{A}_{k+1})_{ts}(x_{k+1} - x_k)_s\|_2^2
\end{aligned}$$

für alle $\delta > 0$. Mit (7.7) erhalten wir

$$\begin{aligned}
\sum_{t\times s\in P} \|(A_{k+1} - \hat{A}_{k+1})_{ts}(x_k)_s\|_2^2 &= \sum_{t\times s\in M_k} \|(A_{k+1} - \hat{A}_{k+1})_{ts}(x_k)_s\|_2^2 + \sum_{t\times s\in P\setminus M_k} \|(A_{k+1} - \hat{A}_{k+1})_{ts}(x_k)_s\|_2^2 \\
&\leq \omega^2 \sum_{t\times s\in M_k} \|(A_k - \hat{A}_k)_{ts}(x_k)_s\|_2^2 + \sum_{t\times s\in P\setminus M_k} \|(A_k - \hat{A}_k)_{ts}(x_k)_s\|_2^2 \\
&= \omega^2\,\eta_k^2(M_k) + \eta_k^2(P\setminus M_k) = \eta_k^2 - (1-\omega^2)\eta_k^2(M_k) \\
&\leq [1 - (1-\omega^2)\theta^2]\eta_k^2,
\end{aligned}$$

wobei wir (7.6) verwendet haben. Die Wahl $\delta := \frac{1}{2}(1-\alpha^2)\theta^2/[1-(1-\alpha^2)\theta^2]$ und

$$\begin{aligned}
z_k &= (1+1/\delta) \sum_{t\times s\in P} \|(A_{k+1} - \hat{A}_{k+1})_{ts}(x_{k+1} - x_k)_s\|_2^2 \\
&\leq c_{\mathrm{sp}} L(1+1/\delta) \max_{t\times s\in P} \|(A_{k+1} - \hat{A}_{k+1})_{ts}\|_2^2 \|x_{k+1} - x_k\|_2^2 \\
&\leq c_{\mathrm{sp}} L(1+1/\delta) \|A_{k+1} - \hat{A}_{k+1}\|_2^2 \|x_{k+1} - x_k\|_2^2 \\
&\leq c_{\mathrm{sp}} L(1+1/\delta) \|A_{k+1} - \hat{A}_{k+1}\|_2^2 \|A_{k+1}^{-1}\|_2^2 \|A_{k+1}x_{k+1} - A_{k+1}x_k\|_2^2 \\
&= c_{\mathrm{sp}} L(1+1/\delta) \|A_{k+1} - \hat{A}_{k+1}\|_2^2 \|A_{k+1}^{-1}\|_2^2 \|A_k x_k - A_{k+1}x_k\|_2^2 \\
&\leq c_{\mathrm{sp}} L(1+1/\delta) \|A_{k+1} - \hat{A}_{k+1}\|_2^2 \|A_{k+1}^{-1}\|_2^2 \|A_k - A_{k+1}\|_2^2 \|x_k\|_2^2
\end{aligned}$$

liefert die erste Aussage der Behauptung.

Der zweite Teil der Behauptung folgt wie im Beweis von Lemma 7.5. □

In den Algorithmen 3 und 4 wurde vorausgesetzt, dass die einzelnen linearen Gleichungssysteme exakt gelöst werden. Unter Verwendung direkter Verfahren, siehe [59], zur Lösung linearer Gleichungssysteme erhalten wir auch eine exakte Lösung, falls hierbei bei numerischen Berechnungen von Rundungsfehlern abgesehen wird. Derartige Methoden weisen aber eine erhöhte Komplexität auf, sodass sie bei größeren Problemen in der Praxis unbrauchbar sind. Zudem sind direkte Verfahren bei hierarchischen Matrizen nicht etabliert bzw. werden hier hauptsächlich in unvollständiger Form zur Konstruktion von Vorkonditionierern verwendet, siehe [13].
Iterative Verfahren, siehe [64], hingegen sind von geringerer Komplexität, lösen aber das lineare Gleichungssystem auch nur approximativ. In der Regel wird hier eine Fehlertoleranz vorgegeben. Den Fehler, der hierbei entsteht, können wir ausnutzen, da die Matrixapproximation bei der ACA auch nur von der Größenordnung der vorgegebenen Fehlertoleranz sein muss. Daher ist es sinnvoll, die Approximation der Systemmatrix mit der iterativen Lösung des linearen Gleichgungssystems zu verbinden, um so einen Mehraufwand bei einem der beiden Schritte einzusparen. Die Kombination zwischen Matrixapproximation und iterativer Lösung soll im achten Kapitel thematisiert werden.

8. Verknüpfung iterativer Löser mit der Matrixapproximation

Am Ende vieler Verfahren, die auf der Diskretisierung von Operatoren beruhen, tritt ein lineares Gleichungssystem der Form $Ax = f$ auf, um das Ausgangsproblem numerisch zu lösen. Die Systemmatrix A ist dabei zumeist symmetrisch und positiv-definit. Für die hart erarbeitete logarithmisch-lineare Komplexität bieten sich Verfahren an, welche dieser Komplexität nicht entgegenwirken. Beispielhaft sei an dieser Stelle das Verfahren der konjugierten Gradienten genannt. Auf den ersten Blick scheint dies eine schlechte Wahl zu sein, da das CG-Verfahren nur für dünnbesetzte Matrizen die geforderte Komplexität einhalten kann. Da wir im Falle hierarchischer Matrizen auf eine schnelle Matrix-Vektor-Multiplikation von logarithmisch-linearer Komplexität zurückgreifen können, kann die Methode der konjugierten Gradienten auch effizient in Verbindung mit $\mathcal{H}$-Matrizen eingesetzt werden.

Im Folgenden wollen wir das Verfahren mit den Techniken aus Kapitel 7 verknüpfen, um so den Speicherbedarf und die Rechenzeit noch weiter zu verringern. Wir starten mit einer kurzen Einführung in das CG-Verfahren gefolgt von einer Anpassung der Methoden aus Kapitel 7 an das iterative Lösungsverfahren. Zuvor sei noch angemerkt, dass auch andere Methoden wie GMRES [59, 64] möglich sind und das CG-Verfahren hier nur beispielhaft angeführt ist.

Da bei einem der Anwendungsbeispiele in Kapitel III indefinite Systemmatrizen auftreten, soll in Abschnitt 8.2 mit dem Bramble-Pasciak CG-Verfahren eine spezielle Variante des Verfahrens der konjugiertem Gradienten, die auf Probleme mit Sattelpunktcharakter zugeschnitten ist, kurz diskutiert werden.

8.1 Die Methode der konjugierten Gradienten

Die Methode der konjugierten Gradienten ist ein auf symmetrische und positiv-definite Matrizen ausgelegtes Verfahren. Die Herleitung und Konstruktion dieses iterativen Verfahrens kann in [59, 64, 68] nachgelesen werden.

Algorithmus 5 CG-Verfahren

Eingabe: Matrix $A \in \mathbb{R}^{N\times N}$ sym., pos. def., rechte Seite $b \in \mathbb{R}^N$, Genauigkeit $\varepsilon_{\mathrm{CG}} > 0$
Ausgabe: Lösungsvektor $x \in \mathbb{R}^N$ von $Ax = b$

Wähle $x^{(0)}$
Setze $r^{(0)} = Ax^{(0)} - b$, $p^{(1)} = -r^{(0)}$, $k = 0$
do
  k++
  if $k > 1$ **then**
    $h_{k-1} = \frac{\langle r^{(k-1)}, r^{(k-1)}\rangle}{\langle r^{(k-2)}, r^{(k-2)}\rangle}$
    $p^{(k)} = -r^{(k-1)} + h_{k-1}p^{(k-1)}$
  end if
  $z = Ap^{(k)}$
  $q_k = \frac{\langle r^{(k-1)}, r^{(k-1)}\rangle}{\langle q^{(k)}, z\rangle}$
  $x^{(k)} = x^{(k-1)} + q_k p^{(k)}$
  $r^{(k)} = r^{(k-1)} + q_k z$
while $\left\|r^{(k)}\right\|_2 \leq \varepsilon_{\mathrm{CG}}$

Nach Algorithmus 5 erhält man pro Iterationsschritt eine Matrix-Vektor-Multiplikation, zwei Skalarprodukte und drei skalare Multiplikationen von Vektoren. Für den Rechenaufwand bedeutet dies $(N+5)N$ multiplikative Operationen. Angenommen γN, $\gamma \ll N$, bezeichne die Anzahl der Elemente

in A, die ungleich Null sind, so führt dies bei dünnbesetzten Matrizen auf einen Rechenaufwand pro CG-Schritt von $(\gamma + 5)N$ multiplikativen Operationen. In unserem Fall, bei der Verwendung der Multiplikation mit Niedrigrang-Matrizen, erhalten wir eine logarithmisch-lineare Komplexität.

Die Methode der konjugierten Gradienten wurde als endlicher Prozess konstruiert. Um dies zu zeigen, sehen wir uns zunächst einige Eigenschaften an, siehe [68].

Lemma 8.1. *Die Residuenvektoren $r^{(k)}$ bilden ein Orthonormalsystem. Die Richtungsvektoren $p^{(k)}$ sind paarweise konjugiert und für $k \geq 2$ gelten die folgenden Bedingungen:*

(i) $\langle r^{(k-1)}, r^{(j)} \rangle = 0,\ j = 0, 1, 2, ..., k-2,$

(ii) $\langle r^{(k-1)}, p^{(j)} \rangle = 0,\ j = 0, 1, 2, ..., k-2,$

(iii) $\langle p^{(k)}, Ap^{(j)} \rangle = 0,\ j = 0, 1, 2, ..., k-1.$

Mit Hilfe von Lemma 8.1 folgt, dass das CG-Verfahren terminiert.

Lemma 8.2. *Die Methode der konjugierten Gradienten liefert die Lösung eines Gleichungssystems in N Unbekannten nach höchstens N Schritten.*

Lemma 8.2 stellt in erster Linie ein theoretisches Resultat dar. Numerisch betrachtet kann es vorkommen, dass der Algorithmus nach N Schritten noch nicht am Ende ist. Denn in der Praxis wird bei den Orthogonalitätsrelationen $\langle r^{(k-1)}, r^{(j)} \rangle$, $j = 0, 1, ..., k-2$, immer ein kleiner Fehler entstehen, welcher sich je nach Konditionszahl von A anders auswirken kann. Daher kann es durchaus angebracht sein, das Verfahren nach N Schritten fortzusetzen und das Abbruchkriterium aus Algorithmus 5 entscheiden zu lassen. In erster Linie hängt die Konvergenz von der Konditionszahl der Matrix A ab. Ein entsprechendes Resultat lässt sich für den Fehler $e^{(k)} = x^{(k)} - x$ in der Energienorm $\|e\|_A := \sqrt{\langle e, Ae \rangle}$ formulieren, siehe zum Beispiel [68]. Dabei ist entscheidend, dass die Unterräume $S_k := \text{span}\{r^{(0)}, r^{(1)}, ..., r^{(k-1)}\}$ Krylov-Unterräume bilden.

Lemma 8.3. *Im CG-Algorithmus gilt für den Fehler $e^{(k)} = x^{(k)} - x$ in der Energienorm die Abschätzung*

$$\|e^{(k)}\|_A \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \|e^{(0)}\|_A. \tag{8.1}$$

Die obere Schranke (8.1) kann dazu verwendet werden, um eine Mindestanzahl an Schritten k anzugeben, welche für die Lösung des linearen Gleichungssystems $Ax = b$ nötig ist. Sei dazu $\varepsilon > 0$ die relative, zu erreichende Genauigkeit, d.h. $\frac{\|e^{(k)}\|_A}{\|e^{(0)}\|_A} \leq \varepsilon$. Dann folgt als untere Schranke für die Anzahl an Iterationen, dass

$$k \geq \frac{1}{2} \sqrt{\kappa(A)} \ln \left(\frac{2}{\varepsilon} \right) + 1.$$

Demnach ist die Effizienz des CG-Verfahrens an die Konditionszahl der Systemmatrix A gekoppelt. Daher werden bei dieser Methode eher kleine Konditionszahlen bevorzugt. Durch eine entsprechende Vorkonditionierung kann die Konditionszahl problemabhängig verkleinert werden. Da im weiteren Verlauf keine Vorkonditionierung verwenden wird, sei auf [64, 59, 68] verwiesen.

Bei den später in Kapitel III betrachteten Problemen wird das Verfahren der konjugierten Gradienten alleine nicht ausreichen, da hier Systemmatrizen auftreten werden, die indefinit sind. Bramble und Pasciak haben in [28] eine Transformation angegeben, unter welcher das konjugierte Gradientenverfahren auch in indefiniten linearen Gleichungssystemen angewendet werden kann.

8.2 Das CG-Verfahren nach Bramble und Pasciak

Betrachtet wird nun ein lineares Gleichungssystem der Form

$$\begin{pmatrix} A & -B \\ B^T & C \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}, \tag{8.2}$$

wobei $A \in \mathbb{R}^{M\times M}$ symmetrisch und positiv-definit, $C \in \mathbb{R}^{N\times N}$ symmetrisch und positiv-semidefinit und $B \in \mathbb{R}^{M\times N}$. Diese Art von linearem Gleichungssystem ist aufgrund der genannten Eigenschaften der Blockmatrizen A, B und C für das CG-Verfahren ungeeignet. Um das Verfahren der konjugierten Gradienten dennoch anwenden zu können, wird nachfolgend ein Vorgehen skizziert, welches ein Gleichungssystem der Form (8.2) in ein symmetrisches und positiv definites Gleichungssystem transformiert. Wir halten uns dabei an die Ausführungen in [28] und [70]. Zu Beginn wird vorausgesetzt, dass eine zu A spektraläquivalente Matrix P_A existiert, d.h. gibt Konstanten $c_2^A \geq c_1^A > 0$ mit

$$c_1^A(P_A x, x) \leq (Ax, x) \leq c_2^A(P_A x, x), \quad x \in \mathbb{R}^M.$$

Die Matrix P_A wird in den meisten Fällen als eine Vorkonditionierung für die Matrix A gewählt. Beispiele hierfür können in [13, 14, 43, 57] gefunden werden. Für die Konstruktion dieser Transformation wird zusätzlich $c_1^A > 1$ vorgeschrieben. Mit dieser Forderung ist die Matrix $A - P_A$ positiv-definit und somit invertierbar, denn es gilt die Abschätzung

$$((A - P_A)x, x) \geq (c_1^A - 1)(P_A x, x)$$

für alle $x \in \mathbb{R}^M$. Dies hat zur Folge, dass auch die beiden Matrizen

$$AP_A^{-1} - I = (A - P_A)P_A^{-1} \quad \text{und} \quad T = \begin{pmatrix} AP_A^{-1} - I & 0 \\ -B^T P_A^{-1} & I \end{pmatrix}$$

invertierbar sind, sodass eine geeignete Transformation des linearen Gleichungssystems (8.2) mit der Matrix T möglich ist. Die Anwendung von T auf das linearen Gleichungssystem (8.2) ergibt

$$M \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = T \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}, \tag{8.3}$$

wobei

$$\begin{aligned} M &= \begin{pmatrix} AP_A^{-1} - I & 0 \\ -B^T P_A^{-1} & I \end{pmatrix} \begin{pmatrix} A & -B \\ B^T & C \end{pmatrix} \\ &= \begin{pmatrix} AP_A^{-1}A - A & (I - AP_A^{-1})B \\ B^T(I - P_A^{-1}A) & C + B^T P_A^{-1} B \end{pmatrix} \end{aligned}$$

symmetrisch ist. Wie in [28] oder [70] gezeigt wird, ist die Matrix M positiv-definit. Dies erlaubt uns die Anwendung des CG-Verfahrens auf das transformierte lineare Gleichungssystem (8.3), um so eine Lösung des Ausgangssystems (8.2) zu bestimmen.

In den weiteren Abschnitten aus Kapitel acht sollen die iterativen Lösungsverfahren mit der erweiterten adaptiven Kreuzapproximation kombiniert und der resultierende Algorithmus analysiert werden.

8.3 Anpassung der erweiterten ACA an residuale Fehler

In Kapitel 7 ist die Annahme getroffen worden, dass wir das in jedem Schritt enthaltene lineare Gleichungssystem exakt lösen. Bei iterativen Verfahren wird eine derartige Annahme schnell an ihre Grenzen kommen. Zudem macht es wenig Sinn, viel Zeit beim Lösen aufzuwenden, um möglichst genau zu rechnen, wenn die restlichen Approximationen viel gröber sind. Damit wir diese Überlegungen in der erweiterten ACA berücksichtigen können, werden wir den zweiten Schritt von Algorithmus 3 weiter ausbauen. Sei dazu ein Parameter $\alpha \geq 0$ gegeben. Dann wird das CG-Verfahren solange auf das lineare Gleichungssystem $A_k x_k = b$ angewendet, bis der residuale Fehler die Bedingung

$$\|b - A_k x_k\|_2 \leq \alpha \|(A_k - \hat{A}_k) x_k\|_2$$

erfüllt. Da im restlichen Verlauf des Verfahrens die Einträge in den einzelnen Blöcken durch sukzessive Rang-Updates entstehen, kann der Vektor x_{k-1} als Startvektor für den iterativen Löser verwendet werden, um so weitere Rechenzeit zu sparen. Die Kombination aus ACA und iterativen Lösungsverfahren ergibt den folgenden Algorithmus.

Algorithmus 6 Block-adaptive Kreuzapproximation (engl. block-adaptive cross approximation; BACA)

1. Starte mit einer groben $\mathcal{H}$-Matrix Approximation A_0 von A und setze $k = 0$.
2. Wende bei gegebenem $\alpha \geq 0$ das CG-Verfahren solange auf das lineare Gleichungssystem $A_k x_k = b$ an, bis der residuale Fehler die Bedingung

   $$\|b - A_k x_k\|_2 \leq \alpha \|(A_k - \hat{A}_k) x_k\|_2$$

   erfüllt (benutze x_{k-1} als Startvektor für den iterativen Löser; $x_{-1} := 0$).
3. Finde bei gegebenem $0 < \theta < 1$ eine Menge markierter Blöcke $M_k \subset P_{\text{adm}}$ von minimaler Mächtigkeit, sodass

   $$\eta_k(M_k) \geq \theta \, \eta_k \tag{8.4}$$

   gilt, wobei $\eta_k^2(M) := \sum_{t \times s \in M} \|(A_k - \hat{A}_k)_{ts} (x_k)_s\|_2^2$ und $\eta_k := \eta_k(P_{\text{adm}})$.
4. Setze

   $$A_{k+1} = \begin{cases} (\hat{A}_k)_b, & b \in M_k, \\ (A_k)_b, & b \in P \setminus M_k. \end{cases}$$

5. Falls $\eta_k > \varepsilon_{\text{BACA}}$, erhöhe k und gehe zu Schritt 2.

Eine Änderung im Schritt zwei von Algorithmus 3 hat auch Auswirkungen auf die Eigenschaften des verwendeten Fehlerschätzers. Im Folgenden gilt es zu ermitteln, wie der Fehlerschätzer η_k vom neu eingeführten Parameter $\alpha > 0$ und dem residualen Fehler

$$\delta_k := \|b - A_k x_k\|_2$$

der approximierten Lösung x_k von $A_k x_k = b$ abhängt.

Lemma 8.4. *Sei die Saturationsannahme* (7.2) *erfüllt. Dann ist der Fehlerschätzer η_k zuverlässig, d.h. es gilt*

$$\|b - A x_k\|_2 \leq \frac{1 + \alpha(1 + c_{\text{sat}})}{1 - c_{\text{sat}}} \|(A_k - \hat{A}_k) x_k\|_2 \leq \sqrt{c_{\text{sp}} L} \, \frac{1 + \alpha(1 + c_{\text{sat}})}{1 - c_{\text{sat}}} \, \eta_k.$$

Beweis. Wir nutzen wiederum die Zerlegung von $A \in \mathcal{H}(P,k)$ in eine Summe von Level-Matrizen $A^{(\ell)}$, welche diejenigen Blöcke $b \in P$ von A enthalten, die zum ℓ-ten Level des Blockclusterbaumes $T_{I\times I}$ gehören

$$A = \sum_{\ell=1}^{L} A^{(\ell)}.$$

Mit $(\sum_{i=1}^n a_i)^2 \le n \sum_{i=1}^n a_i^2$ für alle $a_i \in \mathbb{R}$, $i = 1,\dots,n$, erhalten wir

$$\begin{aligned}
\|(A_k - \hat{A}_k)x_k\|_2^2 &\le \left(\sum_{\ell=1}^{L} \|(A_k - \hat{A}_k)^{(\ell)}x_k\|_2\right)^2 \le L\sum_{\ell=1}^{L} \|(A_k - \hat{A}_k)^{(\ell)}x_k\|_2^2 \\
&= L\sum_{\ell=1}^{L} \sum_{t\in T_I^{(\ell)}} \| \sum_{s:t\times s\in P} (A_k - \hat{A}_k)_{ts}(x_k)_s\|_2^2 \\
&\le L\sum_{\ell=1}^{L} \sum_{t\in T_I^{(\ell)}} \left(\sum_{s:t\times s\in P} \|(A_k - \hat{A}_k)_{ts}(x_k)_s\|_2\right)^2 \\
&\le c_{\mathrm{sp}} L\sum_{\ell=1}^{L} \sum_{t\in T_I^{(\ell)}} \sum_{s:t\times s\in P} \|(A_k - \hat{A}_k)_{ts}(x_k)_s\|_2^2 \\
&= c_{\mathrm{sp}} L \sum_{t\times s\in P} \|(A_k - \hat{A}_k)_{ts}(x_k)_s\|_2^2 = c_{\mathrm{sp}} L\, \eta_k^2.
\end{aligned}$$

Die Aussage des Lemmas folgt aus

$$\begin{aligned}
\|b - Ax_k\|_2 &\le \|b - A_k x_k\|_2 + \|(A_k - \hat{A}_k)x_k\|_2 + \|\hat{A}_k x_k - Ax_k\|_2 \\
&\le (1+\alpha)\|(A_k - \hat{A}_k)x_k\|_2 + c_{\mathrm{sat}}\|A_k x_k - Ax_k\|_2 \\
&\le (1+\alpha)\|(A_k - \hat{A}_k)x_k\|_2 + c_{\mathrm{sat}}(\alpha\|(A_k - \hat{A}_k)x_k\|_2 + \|b - Ax_k\|_2).
\end{aligned}$$

□

Obwohl die Effizienz des Fehlerschätzers η_k, d.h. $\eta_k \lesssim \|b - Ax_k\|_2$, in numerische Tests beobachtet werden kann, ist dessen Beweis noch eine offene Fragestellung. Nichtsdestoweniger kann der berechenbare Ausdruck $\|(A_k - \hat{A}_k)x_k\|_2$ als untere Schranke für $\|b - Ax_k\|_2$ dienen, falls eine obere Schranke an α angenommen wird.

Lemma 8.5. *Sei die Zulässigkeitsbedingung* (7.2) *erfüllt und* $\alpha \le 1/2$. *Dann gilt die Abschätzung*

$$\|b - Ax_k\|_2 \ge \frac{1 - \alpha(1 + c_{\mathrm{sat}})}{1 + c_{\mathrm{sat}}}\|(A_k - \hat{A}_k)x_k\|_2.$$

Beweis. Mit der Zulässigkeitsbedingung (7.2) folgt

$$\begin{aligned}
\|(A_k - \hat{A}_k)x_k\|_2 &\le \|A_k x_k - Ax_k\|_2 + \|Ax_k - \hat{A}_k x_k\|_2 \le (1 + c_{\mathrm{sat}})\|Ax_k - A_k x_k\|_2 \\
&\le (1 + c_{\mathrm{sat}})(\|b - Ax_k\|_2 + \|b - A_k x_k\|_2) \\
&\le (1 + c_{\mathrm{sat}})\|b - Ax_k\|_2 + (1 + c_{\mathrm{sat}})\alpha\|(A_k - \hat{A}_k)x_k\|_2.
\end{aligned}$$

□

Im anschließenden Kapitel wird die Zuverlässigkeit des Fehlerschätzers η_k hilfreich sein, um die Konvergenz des BACA zu zeigen.

8.4 Konvergenz der residualen BACA

Bevor die Konvergenz der BACA gezeigt wird, setzen wir uns mit der Konvergenz des Fehlerschätzers η_k auseinander, welche eine Konsequenz der Dörfler-Marking-Strategie (8.4) ist. Dabei sei angemerkt, dass mit der Konvergenz der ACA entweder $\lim_{k\to\infty}(A_k)_b = A_b$ oder $b \notin M_k$ ab einem bestimmten Index $k_0 \in \mathbb{N}$ gilt, was $(A_k)_b = (A_{k+1})_b$ für $k \geq k_0$ zur Folge hat. Damit gilt $\lim_{k\to\infty} A_k = A_* \in \mathbb{R}^{N\times N}$ und

$$\lim_{k\to\infty} \|A_{k+1} - A_k\| = 0 = \lim_{k\to\infty} \|\hat{A}_{k+1} - \hat{A}_k\|. \tag{8.5}$$

Unter Beachtung der eben genannten Konvergenzeigenschaften von A_k kann das folgende Fehlerschätzer-Reduktionsprinzip gezeigt werden.

Lemma 8.6. *Angenommen A_* ist invertierbar und α ist hinreichend klein. Dann gilt*

$$\eta_{k+1}^2 \leq q\,\eta_k^2 + z_k, \tag{8.6}$$

wobei z_k eine Folge bezeichnet, die gegen Null konvergiert, und $q < 1$. Zudem konvergiert η_k gegen Null, d.h. $\lim_{k\to\infty}\eta_k = 0$.

Beweis. Nutzen wir die Definitionen von A_{k+1} und η_{k+1}^2, so folgt

$$\sum_{t\times s\in P} \|(A_{k+1}-\hat{A}_{k+1})_{ts}(x_{k+1})_s\|_2^2 = \sum_{t\times s\in M_k} \|(\hat{A}_k-\hat{A}_{k+1})_{ts}(x_{k+1})_s\|_2^2 + \sum_{t\times s\in P\setminus M_k} \|(A_k-\hat{A}_k)_{ts}(x_{k+1})_s\|_2^2.$$

Mit der Youngschen Ungleichung und (8.4) erhalten wir für den zweiten Term

$$\begin{aligned}
\sum_{t\times s\in P\setminus M_k} \|(A_k - \hat{A}_k)_{ts}(x_{k+1})_s\|_2^2 &\leq (1+\varepsilon)\,\eta_k^2(P\setminus M_k) + (1+1/\varepsilon)\sum_{t\times s\in P\setminus M_k} \|(A_k - \hat{A}_k)_{ts}(x_{k+1}-x_k)_s\|_2^2 \\
&= (1+\varepsilon)[\eta_k^2 - \eta_k^2(M_k)] + (1+1/\varepsilon)\sum_{t\times s\in P\setminus M_k} \|(A_k - \hat{A}_k)_{ts}(x_{k+1}-x_k)_s\|_2^2 \\
&\leq (1+\varepsilon)(1-\theta^2)\,\eta_k^2 + (1+1/\varepsilon)\sum_{t\times s\in P\setminus M_k} \|(A_k - \hat{A}_k)_{ts}(x_{k+1}-x_k)_s\|_2^2
\end{aligned}$$

für alle $\varepsilon > 0$. Die letzte Summe kann abgeschätzt werden durch

$$\begin{aligned}
\sum_{t\times s\in P\setminus M_k} \|(A_k - \hat{A}_k)_{ts}(x_{k+1}-x_k)_s\|_2^2 &= \sum_{s\in T_I}\sum_{t:t\times s\in P\setminus M_k} \|(A_k - \hat{A}_k)_{ts}(x_{k+1}-x_k)_s\|_2^2 \\
&\leq \max_{t\times s\in P}\|(A_k - \hat{A}_k)_{ts}\|_2^2 \sum_{s\in T_I}\sum_{t:t\times s\in P\setminus M_k} \|(x_{k+1}-x_k)_s\|_2^2 \\
&\leq c_{\mathrm{sp}} L\|x_{k+1}-x_k\|_2^2 \max_{t\times s\in P}\|(A_k - \hat{A}_k)_{ts}\|_2^2 \\
&\leq c_{\mathrm{sp}} L\|A_{k+1}^{-1}\|_2^2\,\|A_{k+1}x_{k+1} - A_{k+1}x_k\|_2^2 \max_{t\times s\in P}\|(A_k - \hat{A}_k)_{ts}\|_2^2 \\
&\leq 3c_{\mathrm{sp}} L\|A_{k+1}^{-1}\|_2^2 \Big(\|A_{k+1}x_{k+1} - b\|_2^2 + \|(A_{k+1}-A_k)x_k\|_2^2 + \|A_kx_k - b\|_2^2\Big)\|A_k - \hat{A}_k\|_2^2.
\end{aligned}$$

Mit $\|b - A_kx_k\|_2^2 \leq \alpha^2\|(A_k - \hat{A}_k)x_k\|_2^2 \leq \alpha^2 c_{\mathrm{sp}} L\,\eta_k^2$ folgt mit der Wahl $\varepsilon := \frac{1}{2}\theta^2/(1-\theta^2)$ die Abschätzung

$$\eta_{k+1}^2 \leq (1 - \frac{1}{2}\theta^2)\,\eta_k^2 + \gamma[\eta_{k+1}^2 + \eta_k^2] + z_k, \tag{8.7}$$

wobei $\gamma := 3\alpha^2 \frac{2-\theta^2}{\theta^2} \left(c_{\text{sp}} L \|A_k - \hat{A}_k\|_2 \|A_{k+1}^{-1}\|_2\right)^2$ und

$$z_k := \sum_{t\times s \in M_k} \|(\hat{A}_k - \hat{A}_{k+1})_{ts}(x_{k+1})_s\|_2^2 + 3(1+1/\varepsilon)c_{\text{sp}} L \|A_k - \hat{A}_k\|_2^2 \, \|A_{k+1}^{-1}\|_2^2 \|A_{k+1} - A_k\|_2^2 \|x_k\|_2^2$$

gegen 0 konvergiert. Dies wiederum ist eine Konsequenz aus $A_k^{-1} \to A_*^{-1}$ für $k \to \infty$ und (8.5). Zu beachten ist, dass $\{x_k\}_{k\in\mathbb{N}}$ wegen

$$\begin{aligned}\|x_k\|_2 &\leq \|A_k^{-1}\|_2 \left(\|b\|_2 + \|A_k x_k - b\|_2\right) \leq \|A_k^{-1}\|_2 \left(\|b\|_2 + \alpha \|(A_k - \hat{A}_k)x_k\|_2\right) \\ &\leq \|A_k^{-1}\|_2 \|b\|_2 + \alpha \|A_k^{-1}\|_2 (\|A_k\|_2 + \|\hat{A}_k\|_2)\|x_k\|_2\end{aligned}$$

und einem hinreichend kleiem α beschränkt ist. Die Wahl von α, sodass $\gamma < \theta^2/4$ erfüllt ist, führt mit (8.7) und

$$q := \frac{1 - \frac{1}{2}\theta^2 + \gamma}{1-\gamma} < 1$$

zum ersten Teil der Aussage.

Beim Beweis der zweiten Aussage halten wir uns an die Idee des Fehlerschätzer-Reduktionsprinzip, welches in [8] vorgestellt wurde. Sei $z > 0$ eine Zahl mit $z_k \leq z$ für alle k. Mit der Fehlerschätzer-Reduktion (8.6) folgt

$$\begin{aligned}\eta_{k+1}^2 &\leq q\,\eta_k^2 + z_k \leq q^2\,\eta_{k-1}^2 + q\,z_{k-1} + z_k \leq \ldots \leq q^{k+1}\eta_0^2 + \sum_{i=0}^{k} q^{k-i} z_i \\ &\leq q^{k+1}\eta_0^2 + z\sum_{l=0}^{k} q^l \leq \eta_0^2 + \frac{z}{1-q}.\end{aligned}$$

Damit ist die Folge $\{\eta_k\}_{k\in\mathbb{N}_0}$ beschränkt und wir definieren $M := \limsup_{k\to\infty} \eta_k^2$. Benutzen wir ein weiteres Mal die Fehlerschätzer-Reduktion (8.6), so führt das zu

$$M = \limsup_{k\to\infty} \eta_{k+1}^2 \leq q \limsup_{k\to\infty} \eta_k^2 + \limsup_{k\to\infty} z_k = qM.$$

Schließlich folgt $M = 0$ und wir erhalten

$$0 \leq \liminf_{k\to\infty} \eta_k \leq \limsup_{k\to\infty} \eta_k = 0$$

und letztendlich $\lim_{k\to\infty} \eta_k = 0$. □

Die Konvergenz der BACA ist nun eine Folge der Zuverlässigkeit von η_k.

Theorem 8.7. *Die Residuen $r_k := b - Ax_k$ der Folge $\{x_k\}_{k\in\mathbb{N}}$, welche durch Algorithmus 6 konstruiert wurde, konvergiert gegen Null.*

Beweis. Mit Lemma 8.4 und Lemma 8.6 erhalten wir

$$\|b - Ax_k\|_2 \leq \sqrt{c_{\text{sp}} L}\, \frac{1 + \alpha(1 + c_{\text{sat}})}{1 - c_{\text{sat}}}\, \eta_k \to 0.$$

□

Die Analyse der entwickelten Algorithmen ist mit Theorem 8.7 abgeschlossen. Im folgenden dritten Teil wird die Funktionsweise der neuen Methoden in der Praxis untersucht. Wir konzentrieren uns auf die numerische Lösung von partiellen Differentialgleichungen mit Hilfe der Randelemente Methode.

III. Anwendung der Kreuzapproximation bei der Randintegralmethode

Der dritte Teil bildet den Abschluss der Erweiterung der adaptiven Kreuzapproximation mit zusätzlichen adaptiven Elementen. Hier soll mit der Randintegralmethode ein Anwendungsgebiet der ACA bzw. BACA vorgestellt werden. Im weiteren Verlauf schildern wir einige numerische Resultate der BACA und vergleichen sie mit denen der ACA. Zudem wird die Funktionsweise und die Qualität der im zweiten Teil dieser Arbeit behandelten Fehlerschätzer anhand numerischer Beispiele genauer betrachtet. Dabei konzentrieren wir uns auf die numerische Lösung der Laplace-Gleichung und der Elastizitätsgleichungen. Zu Beginn dieses dritten Teils werden die mathematischen Grundlagen zu den betrachteten Problemen in Bezug auf die Randelementmethode kurz eingeführt.

9. Nicht-lokale Operatoren bei Randintegralgleichungen

Wir konzentrieren uns auf die numerische Lösung von partiellen Differentialgleichungen in Zusammenhang mit der Randintegralmethode (engl. Boundary Element Method, BEM). Solche Probleme, die durch BEM gelöst werden können, besitzen oft Fundamentallösungen mit Singularitäten. Das Integral über diese singulären bzw. schwach singulären Kerne führt uns zu den nicht-lokalen Operatoren. Zuerst kümmern wir uns um die beiden Problemstellungen, die genauer betrachtet werden sollen, und anschließend um die notwendigen Funktionenräume, welche in der Regel Sobolev-Räume sind.

9.1 Problemstellungen

Als Problemstellung, auf welche wir die in den vorherigen Kapiteln angesprochenen Techniken anwenden wollen, seien hier mit der Laplace-Gleichung und den Lamé-Gleichungen zwei partielle Differentialgleichungen nachfolgend aufgeführt. Erstere ist Bestandteil vieler physikalischer oder ingenieurwissenschaftlicher Anwendungen. Mit der zweiten Gleichung lässt sich sehr gut lineare Elastizität beschreiben.

9.1.1 Die Laplace-Gleichung

Bei der Laplace-Gleichung

$$-\Delta u = 0$$

handelt es sich um den homogenen Spezialfall der Poisson-Gleichung

$$-\Delta u = f, \quad f : \Omega \to \mathbb{R}, \quad \Omega \subset \mathbb{R}^d.$$

Mit dem Laplace-Operator

$$\Delta u := \sum_{i=1}^{d} \partial_i^2 u$$

lassen sich physikalische Phänomene wie elektrostatische Potentiale oder Gravitationspotentiale sehr gut modellieren, siehe [52]. Zudem kann die Laplace-Gleichung in der Fluiddynamik dazu verwendet werden, einfache Strömungen [69] wie z.B. stationäre, inkompressible, wirbelfreie Strömungen im zwei Dimensionen ohne Benutzung der Navier-Stokes-Gleichungen zu beschreiben. Im Allgemeinen stellt der Laplace-Operator einen Bestandteil vieler partieller Differentialgleichungen dar.

9.1.2 Lineare Elastizität

Die Theorie der linearen Elastizität versucht, Verformungen von Körpern, die nach der Krafteinwirkung wieder in ihre ursprüngliche Form zurückkehren, mathematisch zu beschreiben. siehe [49, 70, 71]. Das Ziel ist hier die Bestimmung eines Verschiebungsfeldes $u(x)$ für alle x in einem beschränkten Gebiet $\Omega \subset \mathbb{R}^3$, so dass für einen elastischen Körper die Gleichgewichtsgleichungen

$$-\sum_{j=1}^{3} \frac{\partial}{\partial x_j} \sigma_{ij}(u, x) = f_i(x) \quad \text{for } x \in \Omega,\ i = 1, 2, 3, \tag{9.1}$$

erfüllt sind, wobei ein reversibles, isotropes und homogenes Materialverhalten angenommen wird. Mit Hilfe des Hooke'schen Gesetzes werden die Komponenten des Spannungstensors

$$\sigma_{ij}(u, x) = \frac{E\nu}{(1+\nu)(1-2\nu)} \delta_{ij} \sum_{k=1}^{3} e_{kk}(u, x) + \frac{E}{1+\nu} e_{ij}(u, x) \quad \text{for } x \in \Omega,\ i = 1, 2, 3,$$

mit dem Dehnungstensor $e_{ij}(u, x)$ verknüpft, welcher unter der Annahme kleiner Deformationen die Form

$$e_{ij} = \frac{1}{2} \left[\frac{\partial}{\partial x_i} u_j(x) + \frac{\partial}{\partial x_j} u_i(x) \right] \quad \text{für } x \in \Omega,\ i, j = 1, 2, 3,$$

hat. Dabei bezeichnen $E > 0$ das Young'sche Modul und $\nu \in (0, 1/2)$ die Poisson Zahl. Nach einigen Transformationen, siehe z.B. [70], erhalten wir das Navier-System

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

mit den Lamé-Konstanten

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

Typische Randbedingungen in der Festkörpermechanik sind eine Mischung aus mehreren Dirichlet-Bedingungen, welche feste Begrenzungen beschreiben, und Neumann-Randbedingungen, welche freie Lagerungen beschreiben. Aus Gründen der Einfachheit wählen wir zunächst eine Dirichlet-Bedingung

$$\gamma_0^{\text{int}} u(x) = g_D(x) \quad \text{für } x \in \Gamma_D$$

und eine Neumann-Randbedingung

$$\gamma_1^{\text{int}} u(x) = g_N(x) \quad \text{für } x \in \Gamma_N,$$

wobei $\partial\Omega := \Gamma = \overline{\Gamma}_D \cup \overline{\Gamma}_N$, $\Gamma_D \cap \Gamma_N = \emptyset$. Zusätzlich nehmen wir an, dass der Dirichlet-Rand positives Maß besitzt, d.h.

$$\int_{\Gamma_D} \mathrm{d}s > 0,$$

um die eindeutige Lösbarkeit des betrachteten Problems zu gewährleisten.

Untersucht man das allgemeine inhomogene Neumann Randwertproblem

$$\begin{aligned} -\mu\Delta u(x) - (\lambda+\mu)\text{grad div } u(x) &= f(x) \quad \text{für } x \in \Omega, \\ \gamma_1^{\text{int}} u(x) &= g_N(x) \quad \text{für } x \in \Omega, \end{aligned}$$

so ist dessen Lösung unter der zusätzlichen Lösbarkeitsbedingung

$$\int_\Omega v_k(x)^T f(x)\, dx + \int_\Gamma \gamma_0^{\text{int}} v_k(x)^T g_N(x)\, ds_x = 0, \quad \text{für alle } v_k \in \mathcal{R}$$

nur bis auf die Lösung des homogenen Neumann Problems, die Starrkörperbewegungen, beschrieben durch die Menge

$$\mathcal{R} = \text{span}\left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -x_2 \\ x_1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -x_3 \\ x_2 \end{pmatrix}, \begin{pmatrix} x_3 \\ 0 \\ -x_1 \end{pmatrix} \right\}$$

eindeutig bestimmt. Bei reinen Neumann Randwertproblemen können für die Eindeutigkeit zusätzliche nodale bzw. skalierende Bedingungen gefordert werden, siehe [70].

9.2 Funktionenräume

Zunächst nehmen wir an, dass $\Omega \subset \mathbb{R}^d$ ein offenes und beschränktes Gebiet darstellt. Um später Spur- bzw. Fortsetzungsoperatoren definieren zu können, werden noch stärkere Voraussetzungen nötig sein. Für $p \in [1, \infty]$ bezeichnet der Raum $L^p(\Omega) := \{u : \Omega \to \mathbb{R} \mid u \text{ messbar}, ||u||_{L^p(\Omega)} < \infty\}$ mit

$$||u||_{L^p(\Omega)} := \begin{cases} \left(\int_\Omega |u|^p \right)^{1/p} & , \ 1 \le p < \infty \\ \inf\limits_{M \subset \Omega,\ |M|=0} \sup\limits_{x \in \Omega \setminus M} |u(x)| & , \ p = \infty \end{cases}$$

den Raum der messbaren Funktionen, welche zur p-ten Potenz integrierbar sind. Im Fall $p = 2$ bildet der Raum $L^2(\Omega)$ in Verbindung mit dem Skalarprodukt

$$\langle u, v \rangle_{L^2(\Omega)} := \int_\Omega uv\ dx, \quad u, v \in L^2(\Omega),$$

einen Hilbert-Raum. Zudem wird über das L^2-Skalarprodukt durch

$$||u||_{L^2(\Omega)} := \langle u, u \rangle_{L^2(\Omega)}^{1/2}$$

eine Norm induziert.

9.2.1 Sobolev-Räume

Unter dem Sobolev-Raum der Ordnung $k \in \mathbb{N}_0$ versteht man den Raum

$$W^{k,p}(\Omega) := \{u \in L^p(\Omega) \mid D^\alpha u \in L^p(\Omega) \text{ für alle } |\alpha| \leq k\},$$

wobei $D^\alpha u$ die schwache Ableitung von u vom Grad α bezeichnet, d.h. es gibt $v_\alpha = D^\alpha u \in L^1_{loc}(\Omega)$ mit

$$\int_\Omega v_\alpha \varphi \, dx = (-1)^{|\alpha|} \int_\Omega u \, \partial^\alpha \varphi \, dx \quad \text{für alle } \varphi \in C_0^\infty(\Omega).$$

Ausgestattet mit den Normen

$$||u||_{W^{k,p}(\Omega)} := \left(\sum_{|\alpha| \leq k} ||D^\alpha u||^p_{L^p(\Omega)} \right)^{1/p} \text{ für } 1 \leq p < \infty \text{ und } ||u||_{W^{k,\infty}(\Omega)} := \sum_{|\alpha| \leq k} ||D^\alpha u||_{L^\infty(\Omega)}$$

sind die Sobolev-Räume $W^{k,p}(\Omega)$ Banach-Räume.
Im Zusammenhang mit partiellen Differentialgleichungen ist der Fall $p = 2$ von größerer Bedeutung, da die Sobolev-Räume $H^k(\Omega) := W^{k,2}(\Omega)$ mit dem Skalarprodukt

$$\langle u, v \rangle_{H^k(\Omega)} := \sum_{|\alpha| \leq k} \langle D^\alpha u, D^\alpha v \rangle_{L^2(\Omega)}, \quad u, v \in H^k(\Omega),$$

wiederum Hilbert-Räume bilden. Des Weiteren stimmt die über das Skalarprodukt induzierte Norm mit der $W^{k,2}(\Omega)$-Norm überein.

Da wir später Randintegralgleichungen betrachten wollen, wird auch eine Definition bzw. Konstruktion von Sobolev-Räumen auf Manigfaltigkeiten und Rändern benötigt. Wir halten uns hierbei an die Konstruktion aus [70]. Sei dafür $\{\varphi_i\}_{i=1}^N$ eine Zerlegung der Eins mit den Eigenschaften

1. $\varphi_i \in C_0^\infty(\mathbb{R}^d)$,
2. $\sum_{i=1}^N \varphi_i(x) = 1, \quad x \in \Gamma$,
3. $\varphi_i(x) = 0, \quad x \in \Gamma \setminus \Gamma_i$,

auf einer geeigneten und stückweisen Parametrisierung

$$\Gamma_i := \{x \in \mathbb{R}^d \, : \, x = \chi_i(\xi) \text{ für } \xi \in \mathcal{T}_i \subset \mathbb{R}^{d-1}\}$$

mit $\partial\Omega = \Gamma = \bigcup_{i=1}^N \Gamma_i$. Damit lässt sich ein entsprechender Sobolevraum $H^s(\Gamma)$ auf dem Rand Γ definieren.

Definition 9.1. *Für $0 \leq s \leq k$ sei die Sobolevnorm $\|\cdot\|_{H^s_\chi(\Gamma)}$ definiert durch*

$$\|v\|_{H^s_\chi(\Gamma)} := \left(\sum_{i=1}^N \|\tilde{v}_i\|^2_{H^s(\mathcal{T}_i)} \right)^{1/2},$$

wobei

$$\tilde{v}_i(\xi) := \varphi_i(\chi_i(\xi)) v(\chi_i(\xi)) = \varphi_i(x) v(x) = v_i(x), \quad \xi \in \mathcal{T}_i \subset \mathbb{R}^{d-1}, \, i = 1, ..., N,$$

mit $v_i(x) := \varphi_i(x) v(x)$, $x \in \Gamma$, die lokalen Funktionen beschreibt.

In Abhängigkeit der Ordnung der zu betrachtenden Ableitung müssen wir eine Bedingung an die lokalen Parametrisierungen vorschreiben, d.h. für $|s| \leq k$ die Annahme $\chi_i \in C^{k-1,1}(\mathcal{T}_i)$. Eine erste zu $\|\cdot\|_{H^s_\chi(\Gamma)}$ äquivalente Norm ist im folgenden Lemma gegeben, siehe [70].

Lemma 9.2. *Für $s = 0$ ist eine in $H^0_\chi(\Gamma)$ äquivalente Norm durch*

$$\|v\|_{L^2(\Gamma)} := \left(\int_\Gamma |v(x)|^2 \, ds_x \right)^{1/2}$$

gegeben.

Beweis. Zunächst gelten

$$\|v\|^2_{H^0_\chi(\Gamma)} = \sum_{i=1}^{J} \int_{\mathcal{T}_i} [\varphi_i(\chi_i(\xi)) v(\chi_i(\xi))]^2 \, d\xi$$

und

$$\|v\|^2_{L^2(\Gamma)} = \int_\Gamma [v(x)]^2 \, ds_x = \sum_{i=1}^{J} \int_{\Gamma_i} [\varphi_i(x) v(x)]^2 \, ds_x.$$

Unter Benutzung der lokalen Parametrisierung erhalten wir

$$\|v\|^2_{L^2(\Gamma)} = \sum_{i=1}^{J} \int_{\mathcal{T}_i} [\varphi_i(\chi_i(\xi)) v(\chi_i(\xi))]^2 \mathrm{det} \chi_i(\xi) \, d\xi,$$

wodurch die Behauptung folgt. Dabei hängen die auftretenden Konstanten bei der Normäquivalenz von der gewählten Parametrisierung und den gewählten Abschneidefunktionen ab. □

Eine weitere äquivalente Norm ist für $s \in (0,1)$ durch die Sobolev-Slobodeckij Norm

$$\|v\|_{H^s(\Gamma)} := \left(\|v\|_{L_2(\Gamma)} + \int_\Gamma \int_\Gamma \frac{(v(x) - v(y))^2}{|x-y|^{d-1+2s}} \, \mathrm{d}s_x \, \mathrm{d}s_y \right)^{1/2}.$$

gegeben. Als letztes werden Sobolev-Räume für negative Exponenten $s < 0$ benötigt. Unter Berücksichtigung der Dualitätspaarung

$$\langle u, v \rangle_\Gamma := \int_\Gamma u(x) v(x) \, \mathrm{d}s_x$$

ist der Sobolevraum $H^s(\Gamma)$ für $s < 0$ definiert als der Dualraum von $H^{-s}(\Gamma)$, d.h.

$$H^s(\Gamma) := (H^{-s}(\Gamma))'.$$

Die zu $H^s(\Gamma)$, $s < 0$, gehörende Norm lautet

$$\|u\|_{H^s(\Gamma)} := \sup_{0 \neq v \in H^{-s}(\Gamma)} \frac{\langle u, v \rangle_\Gamma}{\|v\|_{H^{-s}(\Gamma)}}.$$

Ränder, welche nur stückweise glatt sind, benötigen eine gesonderte Behandlung. Da dies an dieser Stelle zu weit führen würde, sei für derartige Probleme auf [70] verwiesen.

9.2.2 Eigenschaften von Sobolev-Räumen

In diesem Abschnitt werden einige Eigenschaften von Sobolev-Räumen angegeben, welche bei der späteren Formulierung und Analyse von Randwertproblemen notwendig sind. Wir starten mit dem Sobolev'schen Einbettungssatz aus [29] bzw. [56].

Lemma 9.3. *Sei $\Omega \subset \mathbb{R}^d$ ein beschränktes Gebiet mit Lipschitz-Rand $\partial\Omega$ und sei $d \leq k$ für $p = 1$ bzw. $d/p < k$ im Fall $p > 1$. Für $u \in W^{k,p}(\Omega)$ gilt:*

(i) $u \in C(\Omega)$,

(ii) $\|u\|_{L^\infty(\Omega)} \leq c\|u\|_{W^{k,p}(\Omega)}$ *für alle* $u \in W^{k,p}(\Omega)$.

Äquivalente Normen in $W^{k,p}(\Omega)$ können mit Hilfe des Normierungssatzes von Sobolev angegeben bzw. konstruiert werden. Eine Folgerung dieses Theorems ist z.B. die Poincarésche Ungleichung.

Lemma 9.4. *Sei Ω ein beschränktes Lipschitz-Gebiet und $F : W^{1,p}(\Omega) \to \mathbb{R}$ ein stetiges, positiv-homogenes Funktional, d.h.*

$$F(\lambda u) = \lambda F(u), \quad \text{für alle } \lambda > 0,\, u \in W^{1,p}(\Omega),$$

sodass für alle $q = const.$ gilt

$$F(q) = 0 \quad \Rightarrow \quad q = 0.$$

Dann gibt es für jedes $p \in [1, \infty)$ eine Konstante $c > 0$ mit

$$\|u\|_{W^{1,p}(\Omega)} \leq c\left(|u|_{W^{1,p}(\Omega)} + |F(u)|\right), \quad \text{für alle } u \in W^{1,p}(\Omega).$$

Ist F zusätzlich linear, so definiert $|\cdot|_{W^{1,p}} + |F|$ eine zu $\|\cdot\|_{W^{1,p}(\Omega)}$ äquivalente Norm auf $W^{1,p}(\Omega)$.

Beweis. Angenommen eine derartige Konstante existiere nicht. Dann gibt es für jedes $n \in \mathbb{N}$ ein $u_n \in W^{1,p}(\Omega)$ mit

$$\|u_n\|_{W^{1,p}(\Omega)} > n\left(|u_n|_{W^{1,p}(\Omega)} + |F(u_n)|\right).$$

Für $v_n := u_n/\|u_n\|_{W^{1,p}(\Omega)}$ gilt dann $\|v_n\|_{W^{1,p}(\Omega)} = 1$ für alle $n \in \mathbb{N}$ und

$$\lim_{n\to\infty} |v_n|_{W^{1,p}(\Omega)} + |F(v_n)| = 0.$$

Wegen des Rellichschen Kompakheitssatzes, siehe [75], ist $W^{1,p}(\Omega)$ kompakt in $L^p(\Omega)$ eingebettet. Daher existiert eine in $L^p(\Omega)$ konvergente Teilfolge $\{v_{n_i}\}_{i\in\mathbb{N}}$ von $\{v_n\}_{n\in N}$. Insbesondere ist $\{v_{n_i}\}_{i\in\mathbb{N}}$ eine Cauchy-Folge in $L^p(\Omega)$. Wegen

$$|v_{n_i}|_{W^{1,p}(\Omega)} \to 0, \quad i \to \infty,$$

und

$$\|v_{n_i} - v_{n_j}\|^p_{W^{1,p}(\Omega)} \leq \|v_{n_i} - v_{n_j}\|^p_{L^p(\Omega)} + \left(|v_{n_i}|_{W^{1,p}(\Omega)} + |v_{n_j}|_{W^{1,p}(\Omega)}\right)^p$$

folgt, dass $\{v_{n_i}\}_{i\in\mathbb{N}}$ eine Cauchy-Folge in $W^{1,p}(\Omega)$ ist. Aufgrund der Vollständigkeit von $W^{1,p}(\Omega)$ konvergiert $\{v_{n_i}\}_{i\in\mathbb{N}}$ gegen ein $v^* \in W^{1,p}(\Omega)$. Infolge der Stetigkeit gilt

$$\|v^*\|_{W^{1,p}(\Omega)} = 1, \quad |v^*|_{W^{1,p}(\Omega)} = 0, \quad F(v^*) = 0.$$

Zudem gilt $v^* = const.$ und daher folgt nach Voraussetzung aus $F(v^*) = 0$, dass $v^* = 0$ ist, was einen Widerspruch darstellt. □

Die nächsten beiden Sätze - der Spursatz und der inverse Spursatz - geben die Relation zwischen einer Funktion u und deren innerer Spur $\gamma_0^{\text{int}} u$ bzw. die Relation zwischen den zugehörigen Räumen an, siehe z.B. [1]. Die Inverse ist hier in Form einer Linksinversen aufzufassen.

Lemma 9.5. *Sei $\Omega \subset \mathbb{R}^d$ ein $C^{k-1,1}$- Gebiet. Für $1/2 < s \leq k$ ist der innere Spuroperator*

$$\gamma_0^{\text{int}} : H^s(\Omega) \to H^{s-1/2}(\Gamma)$$

beschränkt durch

$$\|\gamma_0^{\text{int}}\|_{H^{s-1/2}(\Gamma)} \leq c_T \|v\|_{H^s(\Omega)}, \quad c_T > 0,$$

für alle $v \in H^s(\Omega)$.

Lemma 9.6. *Sei $\Omega \subset \mathbb{R}^d$ ein $C^{k-1,1}$- Gebiet. Für $1/2 < s \leq k$ besitzt der innere Spuroperator γ_0^{int} einen steigen inversen Operator*

$$\mathcal{E} : H^{s-1/2}(\Gamma) \to H^s(\Omega)$$

mit den beiden Eigenschaften

1. *$\gamma_0^{\text{int}} \mathcal{E} v = v$ für alle $v \in H^{s-1/2}(\Gamma)$,*
2. *$\|\mathcal{E} v\|_{H^s(\Omega)} \leq c_{IT} \|H^{s-1/2}(\Gamma)\|$, $c_{IT} > 0$, für alle $v \in H^{s-1/2}(\Gamma)$.*

Als Konsequenz dieser beiden Sätze kann der Sobolevraum $H^s(\Gamma)$ mit einer entsprechenden Norm auch als Spurraum $H^{s+1/2}(\Omega)$ aufgefasst werden.

9.3 Randintegralformulierung der Laplace-Gleichung

Beim ersten Problem, welches wir numerisch lösen wollen, handelt es sich, wie in Abschnitt 9.1 bereits erwähnt, um die Laplace-Gleichung mit Dirichlet-Randdaten auf einem Gebiet $\Omega \subset \mathbb{R}^d$, $d \in \mathbb{N}$, mit hinreichend glattem Rand $\Gamma := \partial\Omega$, d.h.

$$\begin{aligned} -\Delta u(x) &= 0, \quad x \in \Omega \\ \gamma_0^{\text{int}} u(x) &= g_D(x), \quad x \in \Gamma \end{aligned} \tag{9.2}$$

für das innere Randwertproblem bzw.

$$\begin{aligned} -\Delta u(x) &= 0, \quad x \in \Omega^c := \mathbb{R}^d \setminus \overline{\Omega} \\ \gamma_0^{\text{ext}} u(x) &= g_D(x), \quad x \in \Gamma \end{aligned} \tag{9.3}$$

für das äußere Randwertproblem. Um später für (9.3) die Lösbarkeit zu garantieren, muss z.B. durch

$$|u(x) - u_0| \leq \frac{C}{|x|}, \quad x \to \infty, \quad C > 0, \tag{9.4}$$

mit einer gegebenen Zahl $u_0 \in \mathbb{R}$ eine geeignete Bedingung im Unendlichen angenommen werden. Derartige Bedingungen sind typisch für Außenraumprobleme und beispielsweise in einer etwas anderen Form auch bei der Helmholtz-Gleichung im Fall akustischer Streuprobleme zu finden, siehe [56] für die Laplace-Gleichung bzw. [27] für die Helmholtz-Gleichung.

9.3.1 Variationelle Formulierung und Diskussion der Lösbarkeit

Der grundsätzliche Start, um die genannten Probleme zu lösen, wird die schwache Formulierung bzw. variationelle Formulierung des jeweiligen Problems sein. Dieses Vorgehen soll kurz im Allgemeinen angesprochen werden.

Wir sind also an der Lösung $u \in V$ der Operatorgleichung

$$\mathcal{A}u = f \tag{9.5}$$

für ein $f \in V'$ interessiert. Dabei sei V ein Hilbertraum mit Skalarprodukt $\langle \cdot, \cdot \rangle_V$ und induzierter Norm $\| \cdot \|_V := \sqrt{\langle \cdot, \cdot \rangle_V}$. Der Raum V' bezeichne den Dualraum von V bzgl. der Dualitätspaarung $\langle \cdot, \cdot \rangle$. Der lineare Operator $\mathcal{A} : V \to V'$ erfülle die folgenden Bedingungen

(i) $\langle \mathcal{A}u, v \rangle = \langle u, \mathcal{A}v \rangle \quad$ für alle $u, v \in V$ (selbstadjungiert)

(ii) $\|Av\|_{V'} \leq c_{\mathcal{A}} \|v\|_V, c_{\mathcal{A}} > 0$, für alle $v \in V$. (beschränkt)

Jede Lösung von (9.5) ist auch eine Lösung des folgenden Problems

$$\langle Au, v \rangle = \langle f, v \rangle \quad \text{für alle } v \in V.$$

Die Gegenrichtung kann hierbei leicht unter Verwendung der Norm des Dualraums gezeigt werden. Mit Hilfe von $\mathcal{A}$ lässt sich auch eine Bilinearform

$$\begin{aligned} &a(u, v) : V \times V \to \mathbb{R}, \\ &a(u, v) := \langle Au, v \rangle \quad \text{für alle } u, v \in V \end{aligned}$$

definieren. Die Beziehung der Bilinearform zum Operator ist in Lemma 9.7 zusammengefasst, siehe auch [70].

Lemma 9.7. *Sei $a(\cdot, \cdot) : V \times V \to \mathbb{R}$ eine beschränkte Bilinearform mit*

$$|a(u, v)| \leq c_{\mathcal{A}} \|u\|_V \|v\|_V \quad u, v \in V.$$

Dann existiert für alle $u \in V$ ein Element $\mathcal{A}u \in V'$ mit

$$\langle \mathcal{A}u, v \rangle = a(u, v) \quad \text{für alle } v \in V.$$

Zudem ist $\mathcal{A} : V \to V'$ linear und beschränkt, d.h. es gilt

$$\|\mathcal{A}u\|_{V'} \leq c_{\mathcal{A}} \|u\|_V \quad \text{für alle } v \in V.$$

Beweis. Sei $\langle f_u, v \rangle := a(u, v)$ für ein $u \in V$. Dann ist $\langle f_u, \cdot \rangle$ eine beschränkte Linearform in V mit $f_u \in V'$. Dann definiert $u \mapsto f_u$ einen linearen Operator $\mathcal{A} : V \to V'$ mit $\mathcal{A}u = f_u$. Weiterhin gilt:

$$\begin{aligned} \|\mathcal{A}u\|_{V'} = \|f_u\|_{V'} &= \sup_{0 \neq v \in V} \frac{|\langle f_u, v \rangle|}{\|v\|_V} \\ &= \sup_{0 \neq v \in V} \frac{|a(u, v)|}{\|v\|_V} \\ &\leq c_{\mathcal{A}} \|u\|_V \end{aligned}$$

□

Der folgende Rieszsche Darstellungssatz (siehe [29]) bildet den ersten Schritt zur Analyse der Lösbarkeit von (9.5). Er erlaubt es stetige, lineare Funktionale mit Elementen von Hilberträumen zu identifizieren.

Lemma 9.8. *Sei V ein Hilbertraum. Dann existiert zu jedem stetigen linearen Funktional $\varphi \in V'$ genau ein $u \in V$ mit $\|u\|_V = \|\varphi\|_{V'}$, sodass*

$$\varphi(v) = \langle v, u\rangle \quad \text{für alle } v \in V.$$

Für ein gegebenes $u \in V$ ist die Abbildung φ definiert durch $v \mapsto \langle v, u\rangle$ ein stetiges lineares Funktional mit $\|\varphi\|_{V'} = \|u\|_V$.

Beweis. Der zweite Teil des Satzes, was der Umkehrung der Aussage des ersten Teils entspricht, ist klar. Um den ersten Teil des Satzes zu zeigen, definieren wir uns einen Unterraum von V. Die Menge $N := \{v \in V : \varphi(v) = 0\}$ ist ein abgeschlossener Teilraum von V, da φ stetig ist. Sei nun $\{v_n\}_{n\in\mathbb{N}}$ eine Folge in N mit $v_n \to v \in V$. Dann folgt:

$$|\varphi(v)| = |\varphi(v_n) + \varphi(v_n - v)| \leq \|\varphi\|_{V'}\|v_n - v\|_V \to 0, \quad \text{für } n \to \infty.$$

Im Folgenden unterscheiden wir zwei Fälle:

1. Sei $N = V$
   Dann können wir $u := 0$ wählen und es gilt $\|u\|_V = \|\varphi\|_{V'}$.

2. Sei $N \neq V$.
   Da N ein abgeschlossener Unterraum von V ist, existiert die Zerlegung $V = N \oplus N^\perp$. Dann gibt es ein $y \in N^\perp$ mit $y \neq 0$ und $\varphi(y) \neq 0$. Es folgt für $v \in V$, dass

   $$v - y\frac{\varphi(v)}{\varphi(y)} \in N$$

   gilt. Wählen wir nun

   $$u := \frac{\overline{\varphi(y)}}{\|y\|_V^2} y$$

   so folgt

   $$\varphi(v) = \langle v, u\rangle.$$

   Somit bleibt die Eindeutigkeit noch zu zeigen. Sei dafür $u' \in V$ eine weitere Lösung von $\langle v, u'\rangle = \varphi(v)$. Dann erhalten wir

   $$\langle v, u - u'\rangle = 0 \quad \text{für alle } v \in V \quad \Rightarrow \quad \|u - u'\|_V = 0.$$

   Aus

   $$\|\varphi\|_{V'} = \sup_{\|v\|_V=1} |\varphi(v)| = \sup_{\|v\|_V=1} |\langle v, u\rangle| \leq \|u\|_V$$

   und

   $$\|u\|_V = \frac{\varphi(y)}{\|y\|} \leq \|\varphi\|_{V'}$$

   folgt schließlich, dass die beiden Normen $\|u\|_V$ und $\|\varphi\|_{V'}$ übereinstimmen.

□

Um die eindeutige Lösbarkeit der Operatorgleichung (9.5) garantieren zu können, benötigen wir weitere Annahmen an den Operator $\mathcal{A}$ bzw. die Bilinearform $a(\cdot,\cdot)$.

Definition 9.9. *Sei $(V, \|\cdot\|_V)$ ein normierter Raum.*

1. *Eine Bilinearform $a(\cdot,\cdot)$ heißt stetig über V, falls ein $c_S > 0$ existiert mit*

$$|a(u,v)| \leq c_S \|u\|_V \|v\|_V \quad \text{für alle } u, v \in V.$$

2. *Eine Bilinearform $a(\cdot,\cdot)$ heißt koerziv über V, falls ein $c_K > 0$ existiert mit*

$$a(v,v) \geq c_K \|v\|_V^2 \quad \text{für alle } v \in V.$$

3. *Ein Operator $\mathcal{A} : V \to V'$ heißt koerziv oder V-elliptisch, falls ein $c_K > 0$ existiert mit*

$$\langle Av, v\rangle \geq c_K \|v\|_V^2 \quad \text{für alle } v \in V.$$

Das nachfolgende Lemma von Lax-Milgram liefert die eindeutige Existenz einer Lösung des Problems (9.5).

Lemma 9.10. *Sei V ein Banachraum und $a : V \times V \to \mathbb{R}$ eine stetige und koerzive Bilinearform. Dann existiert zu jedem $f \in V'$ ein eindeutig bestimmtes $u \in V$, welches eine Lösung von (9.5) ist und der Abschätzung*

$$\|u\|_V \leq \frac{1}{c_K} \|f\|_{V'}.$$

genügt.

Beweis. Um die Aussage dieses Satzes zu zeigen, werden wir den Banachschen Fixpunktsatz, siehe [75, 76], und Lemma 9.8 verwenden. Nach Lemma 9.8 existiert ein bijektiver Operator $\mathcal{R} : V' \to V$ mit $\|\mathcal{R}\varphi\|_V = \|\varphi\|_{V'}$. Sei $J := I - \delta\mathcal{R}\mathcal{A} : V \to V$ mit $\delta > 0$. Dann gilt

$$u = u - \delta\mathcal{R}(\mathcal{A}u - f) = Ju + \delta\mathcal{R}f.$$

Mit der Stetigkeit und Elliptizität von $\mathcal{A}$ erhalten wir

$$\langle \mathcal{R}\mathcal{A}v, v\rangle_V = \langle Av, v\rangle \geq c_K \|v\|_V^2$$

und

$$\|\mathcal{R}\mathcal{A}v\|_V = \|\mathcal{A}v\|_{V'} \leq c_{\mathcal{A}} \|v\|_V$$

Damit folgt

$$\begin{aligned} \|Jv\|_V^2 &= \|(I - \delta\mathcal{R}\mathcal{A})v\|_V^2 \\ &= \|v\|_V^2 - 2\delta\langle \mathcal{R}\mathcal{A}v, v\rangle_V + \delta^2 \|\mathcal{R}\mathcal{A}v\|_V^2 \\ &\leq (1 - 2\delta c_K + \delta^2 c_{\mathcal{A}}^2)\|v\|_V^2. \end{aligned}$$

Der Operator ist nun für $\delta \in (0, 2c_K/c_{\mathcal{A}}^2)$ eine Kontraktion in V. Somit folgt die eindeutige Lösbarkeit von (9.5) mit dem Banachschen Fixpunktsatz. Die noch zu zeigenden Abschätzung ist eine Konsequenz aus

$$c_K \|u\|_V^2 \leq \langle \mathcal{A}u, u\rangle = \langle f, u\rangle \leq \|f\|_{V'} \|u\|_V,$$

wobei $u \in V$ die eindeutige Lösung von (9.5) bezeichnet. □

9.3.2 Fundamentallösung

Die numerische sowie analytische Lösung der Laplace-Gleichung als Randintegralgleichung erfordert zuerst die Existenz einer Fundamentallösung, d.h. es existiert eine Funktion $S(x,y)$, welche die distributionelle Lösung der Gleichung

$$-\Delta_y S(x,y) = \delta_0(y-x), \quad x,y \in \mathbb{R}^d$$

darstellt. Die Fundamentallösung des Laplace-Operators ist gegeben durch

$$S(x) := \begin{cases} -\frac{1}{2\pi}\ln|x|, & d=2, \\ \frac{1}{(d-2)\omega_d}|x|^{2-d}, & d>2 \end{cases}$$

für $x \in R^d \setminus \{0\}$, wobei ω_d die Oberfläche der d-dimensionalen Einheitskugel bezeichnet. Die Darstellung der Fundamentallösung S kann z.B. für $d=3$ mit Hilfe der Fouriertransformation und unter Verwendung von Kugelkoordinaten hergeleitet werden, siehe [70]. Die Lösung von (9.2) kann anschließend mit Hilfe der Darstellungsformel

$$\begin{aligned} u(x) &= \int_\Gamma S(x,y)\gamma_1^{\text{int}}u(y)\,\mathrm{d}s_y - \int_\Gamma \gamma_{1,y}^{\text{int}}S(x,y)\gamma_0^{\text{int}}u(y)\,\mathrm{d}s_y, \quad x \in \Omega \\ &= \int_\Gamma S(x,y)\gamma_1^{\text{int}}u(y)\,\mathrm{d}s_y - \int_\Gamma \gamma_{1,y}^{\text{int}}S(x,y)g_D(y)\,\mathrm{d}s_y \end{aligned} \tag{9.6}$$

angegeben werden, welche aus den Greenschen Formeln resultiert. Die Aufgabe dabei ist die Bestimmung der noch unbekannten Neumann-Daten $\gamma_1^{\text{int}}u := \partial_\nu u|_\Gamma \in H^{-1/2}(\Gamma)$, wobei ∂_ν die Ableitung bzgl. des Normalenvektors ν bezeichnet.

Im Fall des externen Randwertproblems ist die Lösung unter Berücksichtigung von Bedingung (9.4) für $x \in \Omega^c$ durch die Darstellungsformel

$$\begin{aligned} u(x) &= u_0 - \int_\Gamma S(x,y)\gamma_1^{\text{ext}}u(y)\,\mathrm{d}s_y + \int_\Gamma \gamma_{1,y}^{\text{ext}}S(x,y)\gamma_0^{\text{ext}}u(y)\,\mathrm{d}s_y \\ &= u_0 - \int_\Gamma S(x,y)\gamma_1^{\text{ext}}u(y)\,\mathrm{d}s_y + \int_\Gamma \gamma_{1,y}^{\text{ext}}S(x,y)g_D(y)\,\mathrm{d}s_y \end{aligned} \tag{9.7}$$

gegeben, wobei wiederum die fehlenden Neumann-Daten zu berechnen sind.

9.3.3 Randintegraloperatoren für das reine Dirichlet Randwertproblem

Bevor wir die Randintegralgleichung zum Randwertproblem der Laplace-Gleichung formulieren, werden die nötigen Operatoren eingeführt und deren Eigenschaften kurz diskutiert. Wir starten mit dem Einfachschichtpotential.

Definition 9.11. *Sei $\varphi \in H^{-1/2}(\Gamma)$ eine gegebene Dichtefunktion. Dann ist das Einfachschichtpotential*

$$\tilde{V} : H^{-1/2}(\Gamma) \to H^1(\Omega)$$

gegeben durch

$$(\tilde{V}\varphi)(x) := \int_\Gamma S(x,y)\varphi(y)\,\mathrm{d}s_y, \quad x \in \Omega \cup \Omega^c.$$

Aufgrund der Eigenschaften der Fundamentallösung S_Δ ist leicht ersichtlich, dass $\tilde{V}\varphi$ die Laplace-Gleichung löst, d.h. es gilt

$$-\Delta(\tilde{V}\varphi)(x) = 0, \quad \text{für } x \in \Omega \cup \Omega^c.$$

Zudem kann gezeigt werden, dass $\tilde{V}$ durch

$$\|\tilde{V}\varphi\|_{H^1(\Omega)} \leq c\|\varphi\|_{H^{-1/2}(\Gamma)}$$

für Funktionen $\varphi \in H^{-1/2}(\Gamma)$ und eine Konstante $c > 0$ ein beschränkter Operator ist, siehe [70]. Demnach wissen wir, dass $\tilde{V}\varphi$ in $H^1(\Omega)$ liegt und die Anwendung der Spuroperatoren γ_0^{int} bzw. γ_0^{ext} auf $\tilde{V}$ wohldefiniert ist.

Definition 9.12. *Sei $\varphi \in H^{-1/2}(\Gamma)$. Dann ist der Einfachschichtoperator bzgl. der inneren Spur definiert durch*

$$V^{\text{int}} = \gamma_0^{\text{int}}\tilde{V} : H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma)$$

und bzgl. der äußeren Spur durch

$$V^{\text{ext}} = \gamma_0^{\text{ext}}\tilde{V} : H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma).$$

Nach [70] können die beiden Operatoren V^{int} und V^{ext} als schwach singuläres Oberflächenintegral dargestellt werden.

Lemma 9.13. *Sei $\varphi \in L^\infty(\Gamma)$. Dann besitzen die beiden Einfachschichtoperatoren die Darstellung*

$$(V^{\text{int}}\varphi)(x) = \int_\Gamma S(x,y)\varphi(y)\,\mathrm{d}s_y, \quad x \in \Gamma,$$

und

$$(V^{\text{ext}}\varphi) := \lim_{\tilde{x}\to x}(\tilde{V}\varphi)(\tilde{x}), \quad \tilde{x} \in \Omega^c, \quad x \in \Gamma.$$

Für Randpunkte $x \in \Gamma$ stimmen auch die innere und äußere Spur überein, d.h. es gilt

$$(V^{\text{int}}\varphi)(x) - (V^{\text{ext}}\varphi)(x) = \gamma_0^{\text{int}}(\tilde{V}\varphi)(x) - \gamma_0^{\text{ext}}(\tilde{V}\varphi)(x) = 0, \quad x \in \Gamma.$$

Um die Randintegralformulierung bei der Laplace-Gleichung vervollständigen zu können, wird als nächstes noch das Doppelschichtpotential benötigt. Bei der Definition und den Eigenschaften gehen wir analog zum Einfachschichtpotential vor.

Definition 9.14. *Sei $\psi \in H^{1/2}(\Gamma)$ eine gegebene Dichtefunktion. Das Doppelschichtpotential ist definiert als*

$$(K\psi)(x) := \int_\Gamma \gamma_{1,y}^{\text{int}} S(x,y)\psi(y)\,\mathrm{d}s_y, \quad x \in \Omega \cup \Omega^c.$$

Wiederum lässt sich aus den Eigenschaften der Fundamentallösung S folgern, dass $w(x) = (K\psi)(x)$ die Gleichung

$$-\Delta w(x) = 0$$

für alle $x \in \Omega \cup \Omega^c$ löst. Zudem ist für $\psi \in H^{1/2}(\Gamma)$ der Operator K beschränkt, d.h. es existiert eine Konstante $c > 0$, sodass die Abschätzung

$$\|K\psi\|_{H^1(\Omega)} \leq c\|\psi\|_{H^{1/2}(\Gamma)}$$

erfüllt ist. Damit ist auch die Anwendung des Spuroperators γ_0^{int} bzw. γ_0^{ext} auf K wohldefiniert und $\gamma_0^{\text{int}}K : H^{1/2}(\Gamma) \to H^{1/2}(\Gamma)$ bzw. $\gamma_0^{\text{ext}}K : H^{1/2}(\Gamma) \to H^{1/2}(\Gamma)$ definiert einen linearen, beschränkten Operator. Sie besitzen die folgenden Darstellungen, siehe auch [70].

Lemma 9.15. *Für Funktionen $\psi \in H^{1/2}(\Gamma)$ gilt die Darstellung*

$$(K^{\text{int}}\psi)(x) := \gamma_0^{\text{int}}(K\psi)(x) = (-1+\sigma(x))\psi(x) + (\mathcal{K}\psi)(x), \quad x \in \Gamma,$$

bzw.

$$(K^{\text{ext}}\psi)(x) := \gamma_0^{\text{ext}}(K\psi)(x) = \sigma(x)\psi(x) + (\mathcal{K}\psi)(x), \quad x \in \Gamma,$$

wobei

$$\sigma(x) := \lim_{\varepsilon\to 0} \frac{1}{2(d-1)\pi}\frac{1}{\varepsilon^{d-1}} \int\limits_{y\in\Omega:|y-x|=\epsilon} \mathrm{d}s_y, \quad x \in \Gamma,$$

und der Operator

$$(\mathcal{K}\psi)(x) := \lim_{\epsilon\to 0} \int\limits_{y\in\Gamma:|y-x|\geq\varepsilon} \gamma_{1,y}^{\text{int}} S(x,y)\psi(y)\,\mathrm{d}s_y, \quad x \in \Gamma,$$

ein weiteres Doppelschichtpotential definiert.

In diesem Fall addiert sich nicht die Sprungbedingung des Doppelschichtpotentials K zu Null, sondern der Sprung der Konormalenableitung, d.h. es gilt:

$$(K^{\text{ext}}\psi)(x) - (K^{\text{int}}\psi)(x) = \psi(x), \quad x \in \Gamma$$

und

$$\gamma_1^{\text{ext}}(K\psi)(x) - \gamma_1^{\text{int}}(K\psi)(x) = 0, \quad x \in \Gamma.$$

Da wir bei der Laplace-Gleichung nur das reine Dirichletproblem betrachten, werden zu diesem Zeitpunkt keine weiteren Operatoren benötigt. Bei gemischten Randwertproblemen muss zusätzlich noch der hypersinguläre Operator betrachtet werden, siehe Kapitel 9.4.

9.3.4 Randintegralgleichung der Laplace Gleichung

Der Darstellungsformel (9.6) folgend, müssen die fehlenden Cauchydaten $(\gamma_0^{\text{int}}u, \gamma_1^{\text{int}}u)$ ermittelt werden. Unter Berücksichtigung der Operatoren aus Kapitel 9.3.2 gilt für $x \in \Gamma$, dass

$$\begin{aligned}\gamma_0^{\text{int}}u(x) &= (V^{\text{int}}\gamma_1^{\text{int}}u)(x) - (K^{\text{int}}\gamma_0^{\text{int}}u)(x)\\ &= (V^{\text{int}}\gamma_1^{\text{int}}u)(x) + (1-\sigma(x))\gamma_0^{\text{int}}u(x) - (K\gamma_0^{\text{int}}u)(x).\end{aligned} \tag{9.8}$$

Da beim reinem Dirichlet-Randwertproblem (9.2) durch $\gamma_0^{\text{int}}u(x) = g_D(x)$, $x \in \Gamma$, ein Ausdruck der Cauchydaten bereits gegeben ist, reicht die Gleichung (9.8) aus, um die noch fehlende Konormalenableitung $\gamma_1^{\text{int}}u \in H^{-1/2}(\Gamma)$ zu bestimmen. Eine Umstellung von (9.8) liefert mit

$$(V^{\text{int}}\gamma_1^{\text{int}}u)(x) = \sigma(x)g_D(x) + (K^{\text{int}}g_D)(x), \quad x \in \Gamma, \tag{9.9}$$

eine Fredholmsche Randintegralgleichung erster Art, welche aufgrund der $H^{-1/2}(\Gamma)$-Elliptizität, siehe [70], und der Beschränktheit von V^{int} in $H^{1/2}(\Gamma)$ infolge von Lemma 9.10 eine eindeutige und beschränkte Lösung $\gamma_1^{\text{int}} \in H^{-1/2}(\Gamma)$ besitzt, d.h. es gilt

$$\|\gamma_1^{\text{int}}u\|_{H^{-1/2}(\Gamma)} \leq \frac{c_{2,W}}{c_{1,V}}\|g_D\|_{H^{1/2}(\Gamma)}.$$

Den Ausgangspunkt der numerischen Berechnung bildet die zu Gleichung (9.9) äquivalente variationelle Formulierung: finde $\gamma_1^{\text{int}} \in H^{-1/2}(\Gamma)$, sodass

$$\langle V^{\text{int}}\gamma_1^{\text{int}}u, \varphi\rangle_\Gamma = \langle\left(\frac{1}{2}I + K^{\text{int}}\right)g_D, \varphi\rangle_\Gamma$$

für alle $\varphi \in H^{-1/2}(\Gamma)$ erfüllt ist, wobei aufgrund der Definition von σ sich für fast alle $x \in \Gamma$, $\sigma(x) = \frac{1}{2}$ ergibt. Die Betrachtung der variationellen Formulierung ist an dieser Stelle möglich, denn aufgrund der Abbildungseigenschaften der Operatoren V^{int} und K^{int} gilt, dass

$$\begin{aligned} 0 &= \|V^{\text{int}}\gamma_1^{\text{int}}u - \left(\frac{1}{2}I + K^{\text{int}}\right) g_D\| \\ &= \sup_{0\neq\varphi\in H^{-1/2}(\Gamma)} \frac{\langle V^{\text{int}}\gamma_1^{\text{int}}u - \left(\frac{1}{2}I + K^{\text{int}}\right) g_D, \varphi\rangle_\Gamma}{\|\varphi\|_{H^{-1/2}(\Gamma)}}. \end{aligned}$$

Neben der Verwendung der Darstellung von $\gamma_0^{\text{int}}u$ in (9.8) kann auch eine Darstellung für die konormalen Ableitung $\gamma_1^{\text{int}}u$ angegeben werden, wobei hierfür der adjungierte Doppelschichtoperator und der hypersinguläre Operator, siehe Abschnitt 9.4.2, nötig sind. Dies führt auf eine Fredholmsche Randintegralgleichung der zweiten Art, siehe [70], und soll an dieser Stelle nicht weiter verfolgt werden.

Um die fehlende äußere konormalen Ableitung $\gamma_1^{\text{ext}}u \in H^{-1/2}(\Gamma)$ beim äußeren Dirichlet-Randwertproblem zu bestimmen, können wir analog zum inneren Problem vorgehen. Die Anwendung des äußeren Spuroperators auf die Darstellungsformel (9.7) liefert

$$\begin{aligned} \gamma_0^{\text{ext}}u(x) &= u_0 - (V^{\text{ext}}\gamma_1^{\text{ext}}u)(x) + (K^{\text{ext}}\gamma_0^{\text{ext}}u)(x) \\ &= u_0 - (V^{\text{ext}}\gamma_1^{\text{ext}}u)(x) + \frac{1}{2}\gamma_0^{\text{ext}}u(x) + (K\gamma_0^{\text{ext}}u)(x) \end{aligned}$$

für $x \in \Gamma$. Nach einer Umstellung erhalten wir die gesuchte konormalen Ableitung als Lösung der Gleichung

$$(V^{\text{ext}}\gamma_1^{\text{ext}}u)(x) = -\frac{1}{2}g_D(x) + (Kg_D)(x) + u_0, \quad x \in \Gamma,$$

dessen eindeutige Lösbarkeit wiederum eine Folge des Lemmas von Lax-Milgram und den Eigenschaften des Operators V^{ext} ist, bzw. als Lösung der Gleichung in variationeller Form

$$\langle V^{\text{ext}}\gamma_1^{\text{ext}}u, \varphi\rangle_\Gamma = \langle\left(-\frac{1}{2}I + K\right) g_D, \varphi\rangle_\Gamma$$

für alle $\varphi \in H^{-1/2}(\Gamma)$.

9.4 Randintegralformulierung der linearen Elastizität

Wie in Abschnitt 9.1 bereits erwähnt, interessieren wir uns beim zweiten Problem für die Verformungen von elastischen Materialien und die dabei auftretenden Spannungen im Inneren. Während wir in Kapitel 9.3 nur reine Dirichletprobleme in Betracht gezogen haben, stehen hier auch Randwertaufgaben mit gemischten Randwerten im Fokus. Demnach suchen wir die Lösung des Problems

$$\begin{aligned} -\mu\Delta u(x) - (\lambda+\mu)\text{grad div } u(x) = f(x), \quad x \in \Omega \subset \mathbb{R}^d, \quad d = 2,3, \\ \gamma_0^{\text{int}}u(x) = g_D(x), \quad x \in \Gamma_D \\ \gamma_1^{\text{int}}u(x) = g_N(x), \quad x \in \Gamma_N \end{aligned}$$

mit $\Gamma = \overline{\Gamma}_D \cup \overline{\Gamma}_N$ und

$$\int_{\Gamma_D} \mathrm{d}s > 0,$$

siehe auch Kapitel 9.1. Da auch diese Gleichung als Randintegralgleichung mittels der Randelementmethode gelöst werden soll, wird wiederum mit der Existenz einer Fundamentallösung gestartet.

9.4.1 Fundamentallösung der Lamé-Gleichung

Die Fundamentallösung der linearen Elastizität ist gegeben durch die Lösungen $v^k(x,y)$ der Gleichung

$$\int_\Omega \sum_{i,j=1}^{d} \frac{\partial}{\partial y_j}\sigma_{i,j}(v^k(x,y),y)u_i(y)\,\mathrm{d}y = u_k(x), \quad x \in \Omega, k = 1,...,d,$$

oder anders ausgedrückt der Gleichung

$$-\mu\Delta_z v^k(z) - (\lambda+\mu)\mathrm{grad}_z\ \mathrm{div}_z v^k(z) = \delta_0 e^k, \quad z := y - x \in \mathbb{R}^d,\ k = 1,...,d,$$

wobei $e^k \in \mathbb{R}$ den Einheitsvektor mit $e^k_l = \delta_{kl}$, $k,l = 1,...,d$, bezeichnet. Wie [70] zeigt, ist diese Lösung für $d = 2$ gegeben durch den Kelvin'schen Lösungstensor $S(x,y) = (v^1, v^2)$ mit den Komponenten

$$S_{kl}(x,y) = \frac{1}{4\pi}\frac{1}{E}\frac{1+\nu}{1-\nu}\left[(4\nu-3)\log|x-y|\delta_{kl} + \frac{(y_k-x_k)(y_l-x_l)}{|x-y|^2}\right].$$

In drei Dimensionen besitzt $S(x,y) = (v^1, v^2, v^3)$ die Darstellung

$$S_{kl}(x,y) = \frac{1}{4(d-1)\pi}\frac{1}{E}\frac{1+\nu}{1-\nu}\left[(3-4\nu)\frac{\delta_{kl}}{|x-y|} + \frac{(y_k-x_k)(y_l-x_l)}{|x-y|^3}\right].$$

Demnach ist die Lösung des Problems der linearen Elastizität für $x \in \Omega$ und $k = 1,...,d$ gegeben durch die Repräsentationsformel

$$u_k(x) = \int_\Gamma S_k(x,y)^T\gamma_1^{\mathrm{int}}u(y)\,ds_y - \int_\Gamma u(y)^T\gamma_{1,y}^{\mathrm{int}}S_k(x,y)\,ds_y + \int_\Omega f(y)^T S_k(x,y)dy.$$

9.4.2 Zusätzliche Randintegraloperatoren für gemischte Randwertprobleme

Die beim gemischten Randwertproblem auftretenden Neumann-Randdaten können mit Hilfe weiterer Operatoren berücksichtigt werden. Dabei übernimmt der hypersinguläre Operator die reinen Neumann-Werte und ein adjungierter Doppelschichtoperator die Interaktion zwischen den beiden Randtypen. Letzterer ist, wie folgt, definiert, siehe [70].

Definition 9.16. *Sei $w \in H^{-1/2}(\Gamma)$. Dann ist das adjungierte Doppelschichtpotential definiert durch*

$$(\mathcal{K}'w)(x) := \lim_{\epsilon\to 0} \int_{y\in\Gamma\,:\,|y-x|\geq\epsilon} \gamma_{1,x}^{\mathrm{int}}S(x,y)w(y)\,\mathrm{d}s_y.$$

Damit können wir die innere konormalen Ableitung

$$\gamma_1^{\mathrm{int}}\tilde{V} \,:\, H^{-1/2}(\Gamma) \to H^{-1/2}(\Gamma),$$

welche im übrigen, wie [70] zeigt, ein beschränkter Operator ist, zusammen mit

$$\sigma(x) = \lim_{\epsilon\to 0}\frac{1}{2(d-1)}\frac{1}{\epsilon^{d-1}}\int_{y\in\Gamma\,:\,|y-x|=\epsilon} \mathrm{d}s_y \quad \text{für } x \in \Gamma$$

genauer charakterisieren.

Lemma 9.17. *Sei $w \in H^{-1/2}(\Gamma)$. Dann besitzt die konormalen Ableitung $\gamma_1^{\text{int}}\tilde{V}$ die Darstellung*

$$\gamma_1^{\text{int}}(\tilde{V}w)(x) = \sigma(x)w(x) + (\mathcal{K}'w)(x) \quad \text{für } x \in \Gamma$$

bzw.

$$\langle \gamma_1^{\text{int}}\tilde{V}w, v\rangle_\Gamma = \langle \sigma w + \mathcal{K}'w, v\rangle_\Gamma \quad \text{für } v \in H^{1/2}(\Gamma)$$

im $H^{-1/2}(\Gamma)$-Sinne.

Beweis. Dieser eher technische Beweis kann in [70] (Lemma 6.8) nachgeschlagen werden. □

Wie schon in Abschnitt 9.3.4 gilt auch hier $\sigma(x) = 1/2$ für fast alle $x \in \Gamma$, falls Γ zumindest in einer Umgebung von x differenzierbar ist. Des Weiteren stellt der Operator $\mathcal{K}'$, welcher linear und beschränkt ist, den zu K adjungierten Operator dar. Fortan bezeichnen wir $\mathcal{K}'$ mit K'. Auf ähnliche Weise, wie bei der inneren konormalen Ableitung, folgt die äußere konormalen Ableitung zu

$$\gamma_1^{\text{ext}}(\tilde{V}w)(x) = (\sigma(x) - 1)w(x) + (K'w)(x) \quad \text{für } x \in \Gamma.$$

Für die Sprungbedingung der beiden konormalen Ableitungen gilt schließlich im Sinne von $H^{-1/2}(\Gamma)$-Funktionen die Gleichung

$$\gamma_1^{\text{ext}}(\tilde{V}w)(x) - \gamma_1^{\text{int}}(\tilde{V}w)(x) = -w(x)$$

für $x \in \Gamma$.

Als letztes Bauteil benötigen wir noch den hypersingulären Operator. Dieser steht in Zusammenhang mit der konormalen Ableitung des Doppelschichtoperators $\gamma_1^{\text{int}}K$.

Definition 9.18. *Sei $v \in H^{1/2}(\Gamma)$. Dann ist der hypersinguläre Operator definiert durch*

$$(Dv)(x) := -\gamma_1^{\text{int}}(Kv)(x) = -\lim_{\tilde{x}\to x} n_x \cdot \nabla_{\tilde{x}}(Kv)(\tilde{x}) \quad \text{für } x \in \Gamma,$$

wobei $\tilde{x} \in \Omega$.

Aufgrund der Eigenschaften des Spuroperators γ_1 und des Operators K ist D ein beschränkter Operator. Weitere Darstellungen für D, die in der Praxis handhabbarer sind, können in [70] gefunden werden. Für die spätere Analyse der Lösbarkeit der untersuchten Probleme ist an dieser Stelle wichtig, dass D elliptisch ist. Leider kann die Elliptizität nicht auf dem gesamten Raum $H^{1/2}(\Gamma)$ gezeigt werden. Jedoch können wir einen geeigneten Unterraum angeben, auf welchem sich die Elliptizität von D zeigen lässt, siehe [70]. Sei hierfür zunächst

$$H^{-1/2}_{\text{const}}(\Gamma) := \{w \in H^{-1/2}(\Gamma) \,:\, \langle w, 1\rangle_\Gamma = 0\}$$

der Raum von $H^{-1/2}(\Gamma)$-Funktionen, welche orthogonal zu den konstanten Funktionen sind. Wählen wir ein $w_{\text{const}} \in H^{-1/2}(\Gamma)$ mit $(Vw_{\text{const}})(x) = \lambda$ für $x \in \Gamma$, $\lambda > 0$ und $\langle w_{\text{const}}, 1\rangle = 1$, so können wir den Raum

$$H^{1/2}_{\text{const}}(\Gamma) := \{v \in H^{1/2}(\Gamma) \,:\, \langle v, w_{\text{const}}\rangle_\Gamma = 0\},$$

angeben, auf welchem die Elliptizität von D garantiert werden kann.

Lemma 9.19. *Der hypersinguläre Operator D ist $H^{1/2}_{\text{const}}(\Gamma)$-elliptisch, d.h. es gilt die Abschätzung*

$$\langle Dv, v\rangle_\Gamma \geq c_K \|v\|^2_{H^{1/2}(\Gamma)}, \quad \text{für } v \in H^{1/2}_{\text{const}}(\Gamma)$$

mit einer Konstante $c_k > 0$.

Beweis. Siehe [70]. □

Die Einschränkung von D auf den Raum $H^{-1/2}_{\text{const}}(\Gamma)$ erscheint auf den ersten Blick etwas abstrakt. Im Fall der Lamé-Gleichungen werden wir im nächsten Kapitel sehen, dass der hypersinguläre Operator nur auf die sogenannten Starrkörperbewegungen bezogen elliptisch ist.

9.4.3 Randintegralgleichung der Lamé-Gleichung

Sei $\Omega \subset \mathbb{R}^3$ ein Lipschitz-Gebiet mit Dirichlet-Rand Γ_D und Neumann-Rand Γ_N. Die Lösung der Gleichungen der linearen Elastizität kann infolge der Repräsentationsformel als

$$u(x) = \tilde{V}\gamma_1 u - \tilde{K}\gamma_0 u$$

geschrieben werden, wobei

$$\tilde{V}h(x) := \int_\Gamma S(x,y)h(y)\,\mathrm{d}s_y, \quad h \in [H^{-1/2}(\Gamma)]^3$$

das Einfachschichtpotential und

$$\tilde{K}g(x) := \int_\Gamma \gamma_{1,y} S(x,y)g(y)\,\mathrm{d}s_y, \quad g \in [H^{1/2}(\Gamma)]^3,$$

das Doppelschichtpotential bezeichnen. Die Fundamentallösung $S(x,y) := S(x-y)$ der linearen Elastizität ist gegeben durch den Kelvin'schen Lösungstensor

$$S_K(x)[ij] := \left(\frac{1}{8\pi}\frac{1}{E}\frac{1+\nu}{1-\nu}\left[\frac{3-4\nu}{|x|}\delta_{ij} + \frac{x_i x_j}{|x|^3}\right]\right)_{ij}, \quad i,j = 1,2,3,$$

für $x \in \mathbb{R}^3 \setminus \{0\}$, vgl. auch Kapitel 8.4.1.

Unter Benutzung der Spuroperatoren γ_0^{int} und γ_1^{int} erhalten wir, wie in Kapitel 9.3, den Einfachschichtoperator

$$V := \gamma_0^{\text{int}}\tilde{V} : [H^{-1/2}(\Gamma)]^3 \to [H^{1/2}(\Gamma)]^3$$

und den Doppelschichtoperator

$$\begin{aligned} K &: [H^{1/2}(\Gamma)]^3 \to [H^{1/2}(\Gamma)]^3, \\ (Kg)(x) &= \lim_{\varepsilon\to 0} \int_{\Gamma\setminus B_\varepsilon(x)} \gamma_{1,y} S(x,y)g(y)\,\mathrm{d}s_y, \quad x \in \Gamma, \quad g \in [H^{1/2}(\Gamma)]^3, \end{aligned}$$

bzw. den adjungierten Doppelschichtoperator

$$\begin{aligned} K' &: [H^{-1/2}(\Gamma)]^3 \to [H^{-1/2}(\Gamma)]^3, \\ (K'f)(x) &= \lim_{\varepsilon\to 0} \int_{\Gamma\setminus B_\varepsilon(x)} \gamma_{1,x} S(x,y)g(y)\,\mathrm{d}s_y, \quad x \in \Gamma, \quad f \in [H^{-1/2}(\Gamma)]^3, \end{aligned}$$

auch im vektorwertigen Fall, siehe [56]. Als letztes bleibt noch der hypersinguläre Operator übrig. Nach [56] besitzt er die Darstellung

$$D := -\gamma_1^{\text{int}}\tilde{K} : [H^{1/2}(\Gamma)]^3 \to [H^{-1/2}(\Gamma)]^3.$$

Die folgenden Eigenschaften, welche in [56] oder auch [70] zu finden sind, werden am Ende zusammen mit dem Satz von Lax-Milgram für die Lösbarkeit der Gleichungen der linearen Elastizität entscheidend sein.

Lemma 9.20. *Die Operatoren V, K, K' und D erfüllen die folgenden Eigenschaften:*

1. *Die Operatoren sind stetig, d.h. für $g \in [H^{1/2}(\Gamma)]^3$ und $f \in [H^{-1/2}(\Gamma)]^3$ gelten die Abschätzungen*

   (i) $\|Vf\|_{[H^{1/2}(\Omega)]^3} \leq c_S^V \|f\|_{[H^{-1/2}(\Omega)]^3}$,

   (ii) $\|Kg\|_{[H^{1/2}(\Omega)]^3} \leq c_S^K \|g\|_{[H^{1/2}(\Omega)]^3}$,

   (iii) $\|K'f\|_{[H^{-1/2}(\Omega)]^3} \leq c_S^{K'} \|f\|_{[H^{-1/2}(\Omega)]^3}$,

   (iv) $\|Dg\|_{[H^{-1/2}(\Omega)]^3} \leq c_S^D \|g\|_{[H^{1/2}(\Omega)]^3}$.

   für Konstanten $c_S^V, c_S^K, c_S^{K'}, c_S^D > 0$.

2. *Der Operator V ist $[H^{1/2}(\Gamma)]^3$-elliptisch, d.h. für alle $f \in [H^{1/2}(\Gamma)]^3$ gibt es eine Konstante $c_K^V > 0$, sodass die Abschätzung*

   $$\langle Vf, f\rangle \geq c_K^V \|f\|_{[H^{-1/2}(\Gamma)]^3}$$

   gilt.

3. *Im Raum der Starrkörperbewegungen $\mathcal{R}$, siehe Abschnitt 8.1.2, ist der Operator D elliptisch bzgl. dem Raum $[H^{1/2}(\Gamma)]^3$ d.h. es gibt eine Konstante $c_K^D > 0$ mit*

   $$\langle Dg, g\rangle \geq c_K^D \|g\|^2_{[H^{1/2}(\Gamma)]^3}$$

   für alle $g \in [H^{1/2}(\Gamma)]^3$ mit $\langle g, r\rangle = 0$, $r \in \mathcal{R}$.

Beweis. Siehe [56]. □

Bis jetzt haben wir nur Operatoren auf dem gesamten Rand Γ kennengelernt. Um gemischte Randwertprobleme sinnvoll lösen zu können, müssen die Randintegraloperatoren auf den entsprechenden Rändern Γ_D und Γ_N definiert werden. Für die Definitionen auf dem Dirichlet-Rand Γ_D haben wir

$$V_{DD} : [\tilde{H}^{-1/2}(\Gamma_D)]^3 \to [H^{1/2}(\Gamma_D)]^3, \quad V_{DD} f := V\tilde{f}|_{\Gamma_D},$$

wobei

$$\tilde{H}^{-1/2}(\Gamma_D) = [H^{1/2}(\Gamma_D)]'$$

und $f := \tilde{f}|_{\Gamma_D}$ mit $\tilde{f} \in [H^{-1/2}(\Gamma)]^3$ und supp $\tilde{f} \subset \Gamma_D$. Unter Verwendung der Fortsetzung $\tilde{g} \in [H^{1/2}(\Gamma)]^3$ einer Funktion $g \in \tilde{[}H^{1/2}(\Gamma_D)]^3$ und $\tilde{H}^{1/2}(\Gamma_N) := \{v = \tilde{v}|_{\Gamma_N} : \tilde{v} \in H^{1/2}(\Gamma), \text{supp } \tilde{v} \subset \Gamma_N\}$, besitzt der Operator des reinen Neumann-Problems die Darstellung

$$D_{NN} : [\tilde{H}^{1/2}(\Gamma_N)]^3 \to [H^{-1/2}(\Gamma_N)]^3, \quad D_{NN} g := D\tilde{g}|_{\Gamma_N}.$$

mit $H^{-1/2}(\Gamma_N) = [\tilde{H}^{1/2}(\Gamma_N)]'$. Übrig bleiben noch die beiden Operatoren, welche die Interaktion zwischen den Dirichlet- und Neumann-Daten beschreiben. Der Operator

$$K_{ND} : [\tilde{H}^{1/2}(\Gamma_N)]^3 \to [H^{1/2}(\Gamma_D)]^3, \quad K_{ND} g := K\tilde{g}|_{\Gamma_D}$$

beschreibt den Doppelschicht Operator für den Neumann-Rand und

$$K'_{DN} : [\tilde{H}^{-1/2}(\Gamma_D)]^3 \to [H^{-1/2}(\Gamma_N)]^3, \quad K'_{DN} f := K'\tilde{f}|_{\Gamma_N}$$

den adjungierten Doppelschichtoperator für den Dirichlet-Rand. Die Funktionen $\tilde{g} \in [H^{1/2}(\Gamma)]^3$ und $\tilde{f} \in [H^{-1/2}(\Gamma)]^3$ bezeichnen dabei wiederum die Fortsetzungen der Funktionen $g \in \tilde{[}H^{1/2}(\Gamma_N)]^3$ und $f \in \tilde{[}H^{-1/2}(\Gamma_D)]^3$ durch Null. Die gerade auf den Teilrändern definierten Operatoren erben im Allgemeinen die Eigenschaften der Operatoren V, K, K' und D bzgl. der Stetigkeit und Koerzivität. Diese Eigenschaften sind im folgenden Lemma nochmals für die Operatoren auf den Teilrändern zusammengefasst.

Lemma 9.21. *Es gelten die folgenden Eigenschaften:*

1. *Der Operator V_{DD} ist stetig und $[\tilde{H}^{-1/2}(\Gamma)]^3$-elliptisch.*
2. *Der Operator D_{NN} ist stetig und $[\tilde{H}^{1/2}(\Gamma)]^3$-elliptisch.*
3. *Die Operatoren K_{ND} und K'_{ND} sind stetig.*

Beweis. Die Aussagen von Lemma 9.21 sind direkte Folgerungen der Definitionen und Elliptizität der einzelnen Operatoren. □

Damit besitzt das Randwertproblem

$$\begin{aligned} -\sum_{j=1}^{3} \frac{\partial}{\partial x_j}\sigma_{ij}(u,x) &= 0, \quad x \in \Omega \\ u(x) &= g_D(x), \quad x \in \Gamma_D \\ \sum_{j=1} \frac{\partial}{\partial x_j}\sigma_{ij}(u,x)n_j(x) &= g_N, \quad x \in \Gamma_N \end{aligned}$$

die Lösung

$$v = \tilde{V}(\tilde{g}_N + \tilde{t}) - \tilde{W}(\tilde{g}_D + \tilde{u}), \tag{9.10}$$

wobei $\tilde{u} \in [H^{1/2}(\Gamma)]^3$ und $\tilde{t} \in [H^{-1/2}(\Gamma)]^3$ die Fortsetzung der Funktionen $u \in [\tilde{H}^{1/2}(\Gamma_N)]^3$ und $t \in [\tilde{H}^{-1/2}(\Gamma_D)]^3$ durch Null bezeichnen, welche die Lösungen der Integralgleichungen

$$\begin{aligned} V_{DD}t - K_{ND}u &= \left(\frac{1}{2}I + K\right)\tilde{g}_D|_{\Gamma_D} - V\tilde{g}_N|_{\Gamma_D} \\ K'_{DN}t + D_{NN}u &= -D\tilde{g}_D|_{\Gamma_N} + \left(\frac{1}{2}I - K'\right)\tilde{g}_N|_{\Gamma_N} \end{aligned} \tag{9.11}$$

sind. In (9.10) wurden die gegebenen Dirichlet- und Neumann-Daten $g_D \in [H^{1/2}(\Gamma_N)]^3$ und $g_N \in [H^{-1/2}(\Gamma_D)]^3$ zu den Funktionen $\tilde{g}_D \in [H^{1/2}(\Gamma)]^3$ und $\tilde{g}_N \in [H^{-1/2}(\Gamma)]^3$ fortgesetzt.

Die Hauptaufgabe besteht demnach darin, die Integralgleichungen (9.11) zu lösen. Mit der Bilinearform

$$a_{\text{Lamé}}((\tau,\varphi),(\tau',\varphi')) := \langle V_{DD}\tau,\tau'\rangle_{\Gamma_D} - \langle K_{ND}\varphi,\tau'\rangle_{\Gamma_D} + \langle K'_{DN}\tau,\varphi'\rangle_{\Gamma_N} + \langle D_{NN}\varphi,\varphi'\rangle_{\Gamma_N} \tag{9.12}$$

für $(\tau,\varphi),(\tau',\varphi') \in [\tilde{H}^{-1/2}(\Gamma)]^3 \times [\tilde{H}^{1/2}(\Gamma)]^3$ und der Linearform

$$f((\tau',\varphi')) := \langle\left(\frac{1}{2}I + K\right)\tilde{g}_D - V\tilde{g}_N,\tau'\rangle_{\Gamma_D} + \langle\left(\frac{1}{2}I - K'\right)\tilde{g}_N - D\tilde{g}_D,\varphi'\rangle_{\Gamma_N}$$

für $(\tau',\varphi') \in [\tilde{H}^{-1/2}(\Gamma)]^3 \times [\tilde{H}^{1/2}(\Gamma)]^3$, bedeutet dies in variationeller Form: Finde $(t,u) \in [\tilde{H}^{-1/2}(\Gamma)]^3 \times [\tilde{H}^{1/2}(\Gamma)]^3$, sodass

$$a_{\text{Lamé}}((t,u),(t',u')) = f((t',u')) \tag{9.13}$$

für alle $(t',u') \in [\tilde{H}^{-1/2}(\Gamma)]^3 \times [\tilde{H}^{1/2}(\Gamma)]^3$ gilt. Unter Verwendung der Eigenschaften der Operatoren V_{DD}, K_{ND}, K'_{DN} und D_{NN} und dem Satz von Lax-Milgram kann die Existenz einer Lösung (t,u) von (9.11) und letztendlich die Lösung der Gleichungen der linearen Elastizität garantiert werden.

Lemma 9.22. *Das System von Integralgleichungen* (9.11) *bzw. die dazu äquivalente Variationsgleichung* (9.13) *besitzt eine eindeutige Lösung $(t,u) \in [\tilde{H}^{-1/2}(\Gamma)]^3 \times [\tilde{H}^{1/2}(\Gamma)]^3$.*

Beweis. Die Stetigkeit der Bilienarform $a_{\text{Lamé}}$ und der Linearform f ist eine Folge der Stetigkeit der darin enthaltenen Operatoren. Aus

$$\langle K_{ND}u,t\rangle_{\Gamma_D} = \langle K\tilde{u},\tilde{t}\rangle_{\Gamma} = \langle \tilde{u},K'\tilde{t}\rangle_{\Gamma} = \langle u,K'_{DN}t\rangle_{\Gamma_N}, \quad \text{für } (t,u)\in[\tilde{H}^{-1/2}(\Gamma)]^3\times[\tilde{H}^{1/2}(\Gamma)]^3$$

und den Eigenschaften der Operatoren V_{DD} und D_{NN} können wir die Koerzivität der Bilinearform $a_{\text{Lamé}}$ folgern, denn es gilt:

$$\begin{aligned} a_{\text{Lamé}}((t,u),(t,u)) &= \langle V_{DD}t,t\rangle_{\Gamma_D} - \langle K_{ND}u,t\rangle_{\Gamma_D} + \langle u,K'_{DN}t\rangle_{\Gamma_N} + \langle D_{NN}u,u\rangle_{\Gamma_N} \\ &= \langle V_{DD}t,t\rangle_{\Gamma_D} + \langle D_{NN}u,u\rangle_{\Gamma_N} \\ &\geq c_K^V\|t\|_{[H^{-1/2}(\Gamma_D)]^3} + c_K^D\|u\|_{[H^{1/2}(\Gamma_N)]^3} \\ &\geq \min(c_K^V,c_K^D)\|(t,u)\|^2_{[\tilde{H}^{-1/2}(\Gamma_D)]^3\times[\tilde{H}^{1/2}(\Gamma_N)]^3}, \end{aligned}$$

wobei hier wie in Lemma 9.20 die Starrkörperbewegungen auszuschließen sind. Mit dem Lemma von Lax-Milgram 9.10 folgt schließlich die Existenz einer eindeutigen Lösung. □

Eine stabile numerische Behandlung der gerade definierten Operatoren ist nur möglich, wenn die Singularitäten der Kernfunktionen nicht zu stark sind. Da der Operator V in diese Kategorie fällt, kann hier eine stabile numerische Berechnung erwartet werden. Für die Operatoren K und D sollten aber schwach singuläre Darstellungen gefunden werden. Mit Hilfe der Rand-Differentialoperatoren

$$T_{ij}(\partial_x,n(x)) := n_j(x)\frac{\partial}{\partial x_j} - n_i(x)\frac{\partial}{\partial x_i}, \quad i,j=1,2,3, \quad x\in\Gamma$$

und

$$\begin{aligned} \frac{\partial}{\partial S_1}(x) &:= T_{32}(\partial_x,n(x)), \quad \frac{\partial}{\partial S_2}(x) := T_{13}(\partial_x,n(x)), \\ \frac{\partial}{\partial S_3}(x) &:= T_{12}(\partial_x,n(x)), \end{aligned}$$

kann der Doppelschichtoperator umgeschrieben werden zu

$$Ku(x) = \frac{1}{4\pi}\int_\Gamma \frac{\partial}{\partial n_y}\frac{1}{|x-y|}u(y)\,\mathrm{d}s_y - \frac{1}{4\pi}\int_\Gamma \frac{1}{|x-y|}Tu(y)\,\mathrm{d}s_y + 2\mu VTu(x)$$

für $u\in[H^{1/2}(\Gamma)]^3$. Für den Operator D gilt in Termen schwach singulärer Anteile die folgende Darstellung:

$$\begin{aligned} \langle Du,v\rangle_\Gamma = {} & \frac{\mu}{4\pi}\int_\Gamma\int_\Gamma \frac{1}{|x-y|}\left(\sum_{k=1}^3 \frac{\partial}{\partial S_k}u(y)\cdot\frac{\partial}{\partial S_k}v(x)\right)\mathrm{d}s_x\,\mathrm{d}s_y \\ & + \frac{\mu}{2\pi}\int_\Gamma\int_\Gamma (T(\partial_x,n(x))v(x))^T\frac{I}{|x-y|}(T(\partial_y,n(y))u(y))^T\,\mathrm{d}s_x\,\mathrm{d}s_y \\ & - 4\mu^2\langle VTu,Tv\rangle_\Gamma \\ & + \frac{\mu}{4\pi}\int_\Gamma\int_\Gamma \sum_{i,j,k=1}^3 T_{kj}(\partial_x,n(x)))v_j(x)\frac{1}{|x-y|}T_{ki}(\partial_y,n(y))v_i(x)\,\mathrm{d}s_x\,\mathrm{d}s_y, \end{aligned}$$

siehe [42].
Da nun alle für die Randintegralformulierung der Lamé-Gleichungen notwendigen Operatoren definiert wurden, können wir im nächsten Schritt zu der Diskretisierung der Operatoren und der Beschreibung der Randelementmethode übergehen.

10. Approximation von Randintegralgleichungen - Die Randelementmethode

Das Ziel ist die adaptive Berechnung einer numerischen Lösung von Randwertproblemen bezüglich der Laplace-Gleichung oder linearen Elastizität auf Basis von Randintegralgleichungen, wie sie in Kapitel 9 formuliert wurden. Wir verwenden dazu die Randelementmethode (engl. Boundary Element Method, BEM) und starten mit der Diskretisierung des Randes Γ.

10.1 Diskretisierung und Ansatzräume

Den Ausgangspunkt stellt eine zulässige Triangulierung $\mathcal{T}_h$ (siehe z.B. [65]) der Oberfläche des betrachteten Gebiets $\Omega \subset \mathbb{R}^d$ in $M \in \mathbb{N}$ reguläre Dreiecke τ_k, $k = 1, ..., M$, und $N \in \mathbb{N}$ Knoten n_l, $l = 1, ..., N$ dar. Hierbei heißt eine Triangulierung zulässig, falls disjunkte Dreiecke nur eine gemeinsame Kante oder einen gemeinsamen Knoten haben, siehe Abbildung 10.1.

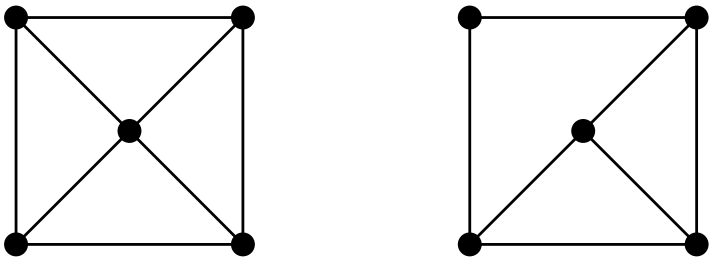

Abb. 10.1: Zulässige Triangulierung links, unzulässige Triangulierung rechts.

Auf der Triangulierung $\mathcal{T}_h$ sei der Raum der konstanten Funktionen $\mathcal{S}_h^0(\Gamma)$ gegeben durch die Basis

$$\varphi_k(x) = \begin{cases} 1, & x \in \tau_k \\ 0, & \text{sonst} \end{cases}, \quad k = 1, ..., M,$$

welche für die Diskretisierung der Einfachschichtpotentiale verwendet wird. Die übrigen Operatoren werden mit den Funktionen

$$\psi_k(x) := \begin{cases} 1, & x = n_k \\ 0, & x = n_j \neq n_k \\ \text{linear}, & \text{sonst} \end{cases}, \quad k = 1, ..., N,$$

diskretisiert, welche eine Basis des Raums der global stetigen und stückweise linearen Funktionen $\mathcal{S}_h^1(\Gamma)$ bilden. Im Folgenden werden wir die resultierenden Gleichungssysteme für die beiden betrachteten Gleichungen angeben.

10.1.1 Diskretisierung der Laplace-Gleichung

Um die Laplace-Gleichung numerisch zu lösen, wird eine Lösung der Form

$$u_h(x) = \sum_{j=1}^{M} \alpha_j \varphi_j, \quad \alpha_j \in \mathbb{R}, \quad j = 1, ..., M,$$

gesucht. Zusammen mit einer stückweisen linearen Approximation der rechten Seite, d.h.

$$g_h(x) = \sum_{l=1}^{N} g_l \psi_l(x), \quad g_l,\, l = 1, ..., N,$$

besitzt das diskretisierte Variationsproblem bzgl. der Laplace-Gleichung die Darstellung

$$\langle\left(\frac{1}{2}I + K\right) g_h, \varphi_i\rangle_{L^2} = \langle V u_h, \varphi_i\rangle_{L^2}, \quad i = 1, ..., M. \tag{10.1}$$

Mit

$$a_{ij} := \int_{\partial\Omega}\int_{\partial\Omega} S(x-y)\varphi_j(y)\varphi_i(x)\,\mathrm{d}s_y\,\mathrm{d}s_x, \quad i,j = 1, ..., M,$$

und

$$b_i := \sum_{l=1}^{N} g_l\langle\left(\frac{1}{2}I + K\right)\psi_l, \varphi_i\rangle_{L^2}, \quad i = 1, ..., M,$$

folgt das lineare Gleichungssystem

$$A\alpha = b, \quad A \in \mathbb{R}^{M\times M}, \quad b \in \mathbb{R}^M$$

für die obige Diskretisierung (10.1).

10.1.2 Diskretisierung der Lame Gleichungen

Im Fall der linearen Elastizität wird eine Lösung der Form

$$t_h(x) = \sum_{i=1}^{M}\begin{pmatrix} t_i^{(1)} \\ t_i^{(2)} \\ t_i^{(3)} \end{pmatrix}\varphi_i(x) \quad \text{und} \quad u_h(x) = \sum_{j=1}^{N}\begin{pmatrix} u_j^{(1)} \\ u_j^{(2)} \\ u_j^{(3)} \end{pmatrix}\psi_j(x)$$

gesucht. Die Koeffizientenvektoren $u = [u_j^{(1)}, u_j^{(2)}, u_j^{(3)}]_{j=1}^N$ und $t = [t_i^{(1)}, t_i^{(2)}, t_i^{(3)}]_{i=1}^M$ sind die Lösungen des linearen Gleichungssystems

$$\begin{pmatrix} V_{DD,h} & -K_{ND,h} \\ K_{ND,h}^T & D_{NN,h} \end{pmatrix}\begin{pmatrix} t \\ u \end{pmatrix} = \begin{pmatrix} -V & \frac{1}{2}M + K \\ \frac{1}{2}M^T - K^T & -D \end{pmatrix}\begin{pmatrix} \tilde{g}_N \\ \tilde{g}_D \end{pmatrix} =: \begin{pmatrix} f_N \\ f_D \end{pmatrix} \tag{10.2}$$

mit den Einträgen

$$V_{DD,h}[ij] = \langle V_{DD}\varphi_j, \varphi_i\rangle_{\Gamma_D}, \quad K_{ND,h}[jk] = \langle K_{ND}\psi_k, \varphi_j\rangle_{\Gamma_D}, \quad D_{NN,h}[kl] = \langle D_{NN}\psi_l, \psi_k\rangle_{\Gamma_N}$$

für $i, j = 1, ..., M$ und $k, l = 1, ..., N$.

Nachfolgend wollen wir noch Darstellungen für die Operatoren $V_{DD,h}$, $K_{ND,h}$ und $D_{NN,h}$ angeben, welche für numerische Berechnungen vorteilhafter sind. Unter der Benutzung des Kelvin'schen Lösungstensors und einem geeigneten Raum zur Diskretisierung der Operatoren wie der Raum $[\mathcal{S}^0(\Gamma)]^3$ lässt sich die Steifigkeitsmatrix des Einfachschichtpotentials durch

$$V_h = \frac{1}{2}\frac{1}{E}\frac{1+\nu}{1-\nu}\left((3-4\nu)\begin{pmatrix} V_{\Delta,h} & 0 & 0 \\ 0 & V_{\Delta,h} & 0 \\ 0 & 0 & V_{\Delta,h} \end{pmatrix} + \begin{pmatrix} V_{11} & V_{12} & V_{13} \\ V_{12} & V_{22} & V_{23} \\ V_{13} & V_{23} & V_{33} \end{pmatrix}\right), \tag{10.3}$$

darstellen, wobei

$$V_{\Delta,h,ij} = \frac{1}{4}\int_{\tau_j}\int_{\tau_i}\frac{1}{|x-y|}\,\mathrm{d}s_y\,\mathrm{d}s_x$$

und

$$V_{kl,ij} = \frac{1}{4}\int_{\tau_j}\int_{\tau_i} \frac{(x_k - y_k)(x_l - y_l)}{|x-y|^3}\,\mathrm{d}s_y\,\mathrm{d}s_x$$

$M \times M$ Untermatrizen für $k,l = 1,2,3$ und $i,j = 1,...,M$ sind. Mit Hilfe des Raums $[\mathcal{S}^1(\Gamma)]^3$ besitzt der Operator K_h die Darstellung

$$K_h = \begin{pmatrix} K_{\Delta,h} & 0 & 0 \\ 0 & K_{\Delta,h} & 0 \\ 0 & 0 & K_{\Delta,h} \end{pmatrix} - \begin{pmatrix} V_{\Delta,h} & 0 & 0 \\ 0 & V_{\Delta,h} & 0 \\ 0 & 0 & V_{\Delta,h} \end{pmatrix} T_h + \frac{E}{1+\nu} V_h T_h, \tag{10.4}$$

mit

$$K_{\Delta,h,ij} = \frac{1}{4\pi} \sum_{\tau \in \mathrm{supp}(\psi_j)} \int_{\tau}\int_{\tau_i} \frac{(x-y)^T n(y)}{|x-y|^3}\psi_j(y)\,\mathrm{d}s_y\,\mathrm{d}s_x, \quad i = 1,...,M,\ j = 1,...,N,$$

und

$$T_h := \begin{pmatrix} 0 & T_{12,h} & T_{13,h} \\ -T_{12,h} & 0 & T_{23,h} \\ -T_{13,h} & -T_{23,h} & 0 \end{pmatrix}, \quad T_{kl,h}[ij] := T_{kl}(\partial_x, n(\hat{x}))\psi_j(\hat{x}),\ \hat{x} \in \tau_i,$$

für $k,l \in \{1,2,3\}$, $i = 1,...,M$, $j = 1,...,N$. Zuletzt kann der Operator D_h geschrieben werden als

$$\begin{aligned} D_h = &\sum_{k=1}^{3} \frac{\mu}{4\pi} S_{k,h}^T \begin{pmatrix} V_{\Delta,h} & 0 & 0 \\ 0 & V_{\Delta,h} & 0 \\ 0 & 0 & V_{\Delta,h} \end{pmatrix} S_{k,h} + \frac{\mu}{2\pi} T_h^T \begin{pmatrix} V_{\Delta,h} & 0 & 0 \\ 0 & V_{\Delta,h} & 0 \\ 0 & 0 & V_{\Delta,h} \end{pmatrix} T_h \\ &+ 4\mu^2 T_h^T V_h T_h + \frac{\mu}{4\pi}\hat{D}_h \end{aligned} \tag{10.5}$$

mit

$$D'_h := \begin{pmatrix} D'_{11,h} & D'_{12,h} & D'_{13,h} \\ D'_{21,h} & D'_{22,h} & D'_{23,h} \\ D'_{31,h} & D'_{32,h} & D'_{33,h} \end{pmatrix}, \quad D'_{ij,h} = \sum_{k=1}^{3} T_{kj,h}^T V_{\Delta,h} T_{ki,h}$$

und

$$S_{1,h} := \begin{pmatrix} T_{32,h} & 0 & 0 \\ 0 & T_{32,h} & 0 \\ 0 & 0 & T_{32,h} \end{pmatrix}, \quad S_{2,h} := \begin{pmatrix} T_{13,h} & 0 & 0 \\ 0 & T_{13,h} & 0 \\ 0 & 0 & T_{13,h} \end{pmatrix}$$

$$S_{3,h} := \begin{pmatrix} T_{21,h} & 0 & 0 \\ 0 & T_{21,h} & 0 \\ 0 & 0 & T_{21,h} \end{pmatrix}.$$

Die Verwendung spezifischer Restriktionsoperatoren, die in [62] definiert sind, erlaubt die Darstellung der diskretisierten Operatoren V_h, K_h und D_h in Bezug auf die entsprechenden Grenzen, was zu den Operatoren $V_{DD,h}$, $K_{ND,h}$ und $D_{NN,h}$ führt.

Trotz der neuen Darstellung enthalten die Operatoren der Lamé-Gleichungen, wie im übrigen auch die Operatoren der Laplace-Gleichung, singuläre bzw. schwach singuläre Integrale. Die numerische Auswertung derartiger Integrale ist teuer und sehr technisch, sodass es von Vorteil ist, je weniger Auswertungen berechnet werden müssen. Da die Beschreibung der nötigen Integrationstechniken aufwendig ist, sei an dieser Stelle auf das Buch von Sauter und Schwab [65] verwiesen.

10.2 Approximationssätze und Fehleranalyse

Die im Kapitel 10.1 betrachteten Techniken stellen Approximationsmethoden dar, d.h. wir können zunächst nur mit einer Annäherung u_h an die gesuchte Lösung u (bzw. t_h an t) rechnen. Im Folgenden wird eine Fehleranalyse für die verschiedenen Approximationstechniken angeführt, welche die approximative Lösung u_h in Verbindung zu u setzt. Wir starten mit einem ersten allgemeinen Resultat, dem Bramble-Hilbert-Lemma, um die Approximation durch finite Elemente abzuschätzen:

Lemma 10.1. *Sei $\Omega \subset \mathbb{R}^d$ ein Lipschitz-Gebiet und $F : H^k(\Omega) \to \mathbb{R}$ ein stetiges, sublineares Funktional, d.h. es gelten*

(i) $|F(u)| \leq c_1 \|v\|_{H^k(\Omega)}$,

(ii) $|F(u+v)| \leq c_2(|F(u)| + |F(v)|)$,

für $u, v \in H^k(\Omega)$ und $c_1, c_2 > 0$. Falls F auf dem Raum Π^d_{k-1} der d-dim. Polynome vom Grad höchstens $k-1$ verschwindet, so folgt

$$|F(v)| \leq c|v|_{H^k(\Omega)}$$

für alle $v \in H^k(\Omega)$ und $c > 0$.

In den nächsten Schritten wollen wir die Analyse des entstehenden Fehler (vgl. [70]) weiter vertiefen und betrachten die folgenden drei Approximationen genauer in der Galerkin-Formulierung:

1. Die Approximation der Lösung.
2. Die Approximation der Linearform.
3. Die Approximation des Operators.

Bevor wir an dieser Stelle weitermachen soll nochmals genauer auf die Notation eingegangen und geklärt werden, was unter der Galerkin-Formulierung verstanden wird. Die Problemstellung ist die Suche einer Lösung $u \in V$ der Variationsgleichung

$$\langle Au, v\rangle = \langle f, v\rangle \quad \text{für alle } v \in V, \tag{10.6}$$

wobei $f \in V'$ und $A : V \to V'$ ein beschränkter und V-elliptischer Operator ist, d.h. es gelten für $v \in V$ die Eigenschaften:

(i) Es gibt $c_1^A > 0$ mit $\langle Av, v\rangle \geq c_1^A \|v\|_V^2$.

(ii) Es gibt $c_2^A > 0$ mit $\|Av\|_{V'} \leq c_2^A \|v\|_V$.

Darauf aufbauend lautet das im Galerkin-Sinne approximierte Problem: Finde

$$u_h := \sum_{k=1}^{M} u_k \varphi_k \in V_h := \operatorname{span}\{\varphi_k\}_{k=1}^M \subset V,$$

sodass die Variationsgleichung

$$\langle Au_h, v_h\rangle = \langle f, v_h\rangle \tag{10.7}$$

für alle $v_h \in V_h$ erfüllt ist. Dabei bezeichnen φ_k, $k = 1, ..., M$, geeignete konforme Ansatzfunktionen, wie sie z.B. in Kapitel 10.1 verwendet wurden. Insbesondere entspricht die in Abschnitt 10.1 benutzte

Approximation genau der hier vorgestellten Galerkin-Formulierung. Eine wichtige Eigenschaft dieser Approximation ist die sogenannte Galerkin-Orthogonalität

$$\langle A(u-u_h), v_h\rangle = 0 \quad \text{für alle } v_h \in V_h,$$

welche durch $V_h \subset V$ und die Wahl $v = v_h$ in (10.6) folgt. Da (10.7) für alle $v_h \in V_h$ gilt, können wir für v_h die Darstellung

$$v_h := \sum_{l=1}^{M} v_l \varphi_l$$

in V_h verwenden. Damit folgt im Fall von (10.7) das lineare Gleichungssystem

$$Av = f \tag{10.8}$$

mit

$$A_{ij} = \langle A\varphi_k, \varphi_l\rangle \quad \text{und} \quad f_l = \langle f, \varphi_l\rangle, \quad k,l = 1,...,M,$$

wobei die Aufgabe, die Berechnung des Koeffizientenvektors $v \in \mathbb{R}$ ist. Als Folge davon kann die eindeutige Lösbarkeit des linearen Gleichungssystems (10.8) von der eindeutigen Lösbarkeit des Variationsproblems (10.6) abgeleitet werden.

Das nachfolgend aufgeführte Céa-Lemma stellt eine Verbindung zwischen der eigentlichen Lösung und der approximierten Lösung her.

Lemma 10.2. *Sei $A : V \to V'$ ein beschränkter und V-elliptischer Operator. Für die eindeutige Lösung $u_h \in V_h$ des Variationsproblems* (10.7) *gelten sowohl die Stabilitätsabschätzung*

$$\|u_h\|_V \leq \frac{1}{c_1^A}\|f\|_{V'}$$

als auch die Fehlerabschätzung

$$\|u-u_h\|_V \leq \frac{c_2^A}{c_1^A} \inf_{v_h \in V_h} \|u-v_h\|_V.$$

Infolge des Céa-Lemmas wird die Konvergenz der approximativen Lösung u_h für $h \to 0$ an die Approximationseigenschaften des endlich-dimensionalen Raums V_h geknüpft. Konvergiert V_h gegen V für $h \to 0$, so konvergiert auch u_h gegen u für $h \to 0$. Damit hätten wir den ersten Punkt - die Approximation der Lösung - geklärt. Als nächstes kümmern wir uns um die Approximation der Linearform auf der rechten Seite der Gleichung.

Wie in Kapitel 8 thematisiert worden ist, entsteht die rechte Seite aus der Anwendung eines Operators auf die gegebenen Daten. Dieser Operator sei nun mit $F : W \to V'$ bezeichnet und die Daten mit $g \in W$, was in dem Variationsproblem: Finde $u \in V$ mit

$$\langle Au, v\rangle = \langle Fg, v\rangle, \quad \text{für alle } v \in V,$$

resultiert. Setzen wir dieselbe Approximation wie eben und eine geeignete Approximation der Daten, d.h.

$$g_h := \sum_{l=1}^{N} g_l \psi_l \in W_h := \operatorname{span}\{\psi_l\}_{l=1}^{N} \subset W$$

an, so erhalten wir das gestörte Variationsproblem

$$\langle A\tilde{u}_h, v_h\rangle = \langle Fg_h, v_h\rangle, \quad \text{für alle } v_h \in V_h, \tag{10.9}$$

in welchem die gestörte Lösung $\tilde{u}_h \in V_h$ gesucht ist. Auf Basis von (10.9) kann wiederum wie zuvor ein lineares Gleichungssystem

$$A_h \tilde{u} = F_h G$$

mit

$$A_{h,ij} = \langle A\varphi_j, \varphi_i \rangle, \quad F_{h,il} = \langle B\psi_l, \varphi_i \rangle, \quad i,j = 1,...,M, \quad l = 1,...,N,$$

und $G = (g_1, ..., g_N)^T$ gefolgert werden, dessen eindeutige Lösbarkeit durch die Eigenschaften von A gegeben ist. Das folgende Lemma von Strang gibt eine obere Schranke für den Fehler bei der gestörten Lösung $\tilde{u}_h$ an.

Lemma 10.3. *Sei $A : V \to V'$ ein beschränkter linearer und V-elliptischer Operator. Sei $u \in V$ die eindeutige Lösung des stetigen Variationsproblem* (10.6) *und $u_h \in V_h$ die eindeutige Lösung des Galerkin-Variationsproblems* (10.7). *Dann gilt für die eindeutige Lösung $\tilde{u}_h \in V_h$ des gestörten Variationsproblems* (10.9) *die Abschätzung*

$$\|u - \tilde{u}_h\|_V \leq \frac{1}{c_1^A} \left\{ c_2^A \inf_{v_h \in V_h} \|u - v_h\|_V + c_2^B \|g - g_h\|_W \right\}.$$

Beweis. Unter Ausnutzung der V-Elliptizität auf die Subtraktion des gestörten Problems vom Galerkin-Problem folgt:

$$\begin{aligned} \|u_h - \tilde{u}_h\|_V^2 &\leq \frac{1}{c_1^A} \langle A(u_h - \tilde{u}_h), u_h - \tilde{u}_h \rangle \\ &= \langle B(g - g_h), u_h - \tilde{u}_h \rangle \\ &\leq \|B(g - g_h)\|_{V'} \|u_h - \tilde{u}_h\|_V \\ &\leq c_2^B \|g - g_h\|_W \|u_h - \tilde{u}_h\|_V. \end{aligned}$$

Damit erhalten wir

$$\|u_h - \tilde{u}_h\|_V \leq \frac{c_2^B}{c_1^A} \|g - g_h\|_W,$$

wodurch schließlich mit der Dreiecksungleichung und dem Céa-Lemma die Behauptung folgt. □

Wie in Kapitel 10.1 bereits erwähnt, werden wir die auftretenden singulären bzw. schwach singulären Integrale im Allgemeinen nicht analytisch berechnen, d.h. wir werden in diesem Fall auf eine numerische Integration zurückgreifen müssen und eine Approximation des Operators in Betracht ziehen. Auch dieses Vorgehen liefert uns einen Fehler, welchen es abzuschätzen gilt. Sei demnach $\tilde{A} : V \to V'$ der approximierte lineare Operator mit

$$\|\tilde{A}v\|_{V'} \leq c_2^{\tilde{A}} \|v\|_V, \quad v \in V.$$

Dann ist die Lösung $\tilde{u}_h \in V_h$ des gestörten Problems

$$\langle \tilde{A}\tilde{u}_h, v_h \rangle = \langle f, v_h \rangle, \quad \text{für alle } v_h \in V_h, \tag{10.10}$$

gesucht. Da es sich bei $\tilde{A}$ um eine Approximation an A handelt, können wir nicht davon ausgehen, dass $\tilde{A}$ alle Eigenschaften von A erbt. Im folgenden Strang-Lemma, das eine Fehlerabschätzung bezüglich der Lösung des gestörten Problems (10.10) angibt, müssen wir daher eine Art diskrete Stabilität fordern, um überhaupt eine eindeutige Lösung garantieren zu können.

Lemma 10.4. *Angenommen der approximative Operator $\tilde{A} : V \to V'$ ist V_h-elliptisch, d.h. es gibt eine Konstante $\tilde{c}_1^A > 0$ mit*

$$\langle \tilde{A} v_h, v_h \rangle_\Gamma \geq \tilde{c}_1^A \|v_h\|_{V_h}^2 \quad \text{für alle } v_h \in V_h.$$

Dann existiert eine eindeutige Lösung $\tilde{u}_h \in V_h$ des gestörten Problems (10.9) *und es gilt die Abschätzung*

$$\|u - u_h\|_{V_h} \leq \left(1 + \frac{1}{\tilde{c}_1^A}(c_2^A + \tilde{c}_2^A)\right) \frac{c_2^A}{c_1^A} \inf_{v_h \in V_h} \|u - v_h\|_{V_h} + \frac{1}{\tilde{c}_1^A} \|(A - \tilde{A})u\|_{V'}.$$

Beweis. Klar ist, dass das gestörte Problem mit der zusätzlichen Annahme der V_h-Elliptizität eindeutig lösbar ist. Sei also u_h die Lösung des Galerkin-Problems (10.7), dann gilt

$$\begin{aligned}
\|u_h - \tilde{u}_h\|_V^2 &\leq \frac{1}{\tilde{c}_1^A} \langle \tilde{A}(u_h - \tilde{u}_h), u_h - \tilde{u}_h \rangle \\
&= \frac{1}{\tilde{c}_1^A} \langle \tilde{A}(u_h - \tilde{u}_h) + Au_h - Au_h, u_h - \tilde{u}_h \rangle \\
&= \frac{1}{\tilde{c}_1^A} \langle (\tilde{A} - A)u_h, u_h - \tilde{u}_h \rangle \\
&\leq \frac{1}{\tilde{c}_1^A} \|(\tilde{A} - A)u_h\|_{V'} \|u_h - \tilde{u}_h\|_V,
\end{aligned}$$

denn

$$\langle Au_h - \tilde{A}\tilde{u}_h, v_h \rangle = 0, \quad \text{für alle } v_h \in V_h.$$

Damit folgt unter Verwendung der Dreiecksungleichung die Abschätzung

$$\begin{aligned}
\|u_h - \tilde{u}_h\|_V &\leq \frac{1}{c_1^{\tilde{A}}} \|(A - \tilde{A})u_h\|_{V'} \\
&\leq \frac{1}{\tilde{c}_1^A} \|(A - \tilde{A})u\|_{V'} + \frac{1}{\tilde{c}_1^A} \|(A - \tilde{A})(u - u_h)\|_{V'} \\
&\leq \frac{1}{\tilde{c}_1^A} \|(A - \tilde{A})u\|_{V'} + \frac{1}{\tilde{c}_1^A} \|A(u - u_h)\|_{V'} + \|\tilde{A}(u - u_h)\|_{V'} \\
&\leq \frac{1}{\tilde{c}_1^A} \|(A - \tilde{A})u\|_{V'} + \frac{1}{\tilde{c}_1^A} \left(c_2^A + c_2^{\tilde{A}}\right) \|u - u_h\|_V.
\end{aligned}$$

Das Céa-Lemma und eine weitere Anwendung der Dreiecksungleichung in $\|u - \tilde{u}_h\|_V$ liefert die Behauptung. □

Bevor nach der Analyse der einzelnen Approximationsfehler zu den numerischen Tests übergegangen wird, werfen wir einen Blick auf die Anwendung von AMVM und BACA im Fall der linearen Elastizität.

11. Anwendung der neuen Methoden bei der linearen Elastizität

Die block-adaptive Kreuzapproximation ist zunächst für die Laplace-Gleichung mit reinen Dirichlet-Randbedingungen entwickelt worden, wobei die Verfeinerungsstrategie, der Fehlerschätzer und die Auswahl der Blöcke speziell auf die Struktur dieses einfacheren Problems ausgelegt worden sind. Bei komplexeren Problemen wie gemischten Randwertproblemen oder vektorwertigen Gleichungen bedarf es einer Anpassung der eben genannten adaptiven Elemente. Am Beispiel der Lamé-Gleichungen sollen diese Änderungen erläutert werden.

11.1 Anpassung der BACA an die lineare Elastizität

Um die in Kapitel 8 eingeführte BACA auch im Fall der linearen Elastizität einsetzen zu können, werden die im Folgenden beschriebenen Anpassungen vorgenommen.

Wir betrachten die numerische Lösung des linearen Gleichungssystems

$$Ax = f$$

mit einer Matrix A und einer rechten Seite f, wie sie im Fall einer Randintegralapproximation der linearen Elastizität entstehen, d.h.

$$A = \begin{pmatrix} V_{DD,h} & -K_{ND,h} \\ K_{ND,h}^T & D_{NN,h} \end{pmatrix}, \quad f = \begin{pmatrix} f_N \\ f_D \end{pmatrix} \quad \text{und } x = \begin{pmatrix} t \\ u \end{pmatrix}.$$

Jede der vier Teilmatrizen von A besteht wiederum aus neun Teilmatrizen mit dem zugehörigen Blockclusterbaum $T_{I\times J}$. Wir sammeln alle möglichen Blöcke in der Menge $\mathcal{P}$ und nennen die Menge aller zulässigen Blöcke $\mathcal{P}_{\text{adm}}$. Demnach ist $\mathcal{P}$ die Vereinigung aller zulässigen Partitionen des Randes und der zugehörigen Operatoren, d.h. $\mathcal{P} := \mathcal{P}_V \cup \mathcal{P}_K \cup \mathcal{P}_D$, wobei $\mathcal{P}_V$, $\mathcal{P}_K$ und $\mathcal{P}_D$ aus allen Blöcken der Diskretisierung des Einfachschichtoperators V_h, des diskretisierten Doppelschichtoperators K_h sowie des diskretisierten hypersingulären Operators D_h bestehen. Entsprechend bezeichnen auch $c_{\text{sp},V}$, $c_{\text{sp},K}$ und $c_{\text{sp},D}$ die „Sparsity"-Konstanten der zu den Operatoren gehörenden Block-Clusterbäume.

Die Matrizen

$$A_k = \begin{pmatrix} V_{DD,k} & -K_{ND,k} \\ K_{ND,k}^T & D_{NN,k} \end{pmatrix}$$

bezeichnen die konstruierten Approximationen, wobei eigentlich die Teilmatrizen $V_{DD,h}$, $K_{ND,h}$ und $D_{NN,h}$ approximiert werden. Zudem sei

$$\hat{A}_k = \begin{pmatrix} \hat{V}_{DD,k} & -\hat{K}_{ND,k} \\ \hat{K}_{ND,k}^T & \hat{D}_{NN,k} \end{pmatrix}$$

eine bessere Approximation an A als A_k. Dabei nehmen wir an, dass die Saturationsannahme

$$\|\tilde{A}_k x_k - A x_k\|_2 \leq c_{\text{sat}} \|A_k x_k - A x_k\|_2 \tag{11.1}$$

für $0 < c_{\text{sat}} < 1$ erfüllt ist, wobei x_k die Lösung des linearen Gleichungssystems $A_k x_k = b$ bezeichnet. Ein mögliches Vorgehen bei der Konstruktion von $\hat{A}_k$ ist wiederum, eine feste Anzahl an ACA-Schritten bei jedem zulässigen Block von A_k hinzuzufügen und $(\hat{A}_k)_{t\times s} = A_{t\times s}$ für alle nicht zulässigen Blöcke $t \times s \in \mathcal{P}_{\text{non-adm}} := \mathcal{P} \setminus \mathcal{P}_{\text{adm}}$ zu setzen.

Um Informationen über den Fehler der Approximation zu erhalten, verwenden wir den Fehlerschätzer

$$\mathcal{E}_k^2 := \sum_{t\times s \in \mathcal{P}} \|(A_k - \hat{A}_k)_{ts} (x_k)_s\|_2^2.$$

Die Anpassung der in Kapitel 8 eingeführten BACA an die lineare Elastizität erfordert zwei weitere Erweiterungen bzw. Änderungen. Der erste Unterschied liegt in der Lösung des auftretenden Gleichungssystems, da dies strukturell gesehen ein Sattelpunktproblem ist. Wir nutzen daher im Algorithmus anstelle der CG-Methode den Bramble-Pasciak-CG, welcher speziell auf derartige Gleichungssysteme zugeschnitten ist, siehe [28]. Die größere Änderung zur Laplace-Gleichung ist die Auswahl der zu verfeinernden Blöcke. Das Vorgehen ist, wie folgt:

Wir nutzen die Darstellung der diskretisierten Operatoren (10.3), (10.4) und (10.5), wobei die festen Matrizen T_h, $S_{1,h}$, $S_{2,h}$ und $S_{3,h}$ während des gesamten Prozesses unberührt bleiben. Da die Teilmatrix $V_{\Delta,h}$ in jedem der Operatoren V_h, K_h und D_h meist mehrfach enthalten ist, wird die Verfeinerung von $V_{\Delta,h}$ zuerst ausgeführt. Danach folgen die Verbesserung der Teilmatrizen V_{ij}, $i,j = 1,2,3$, $K_{\Delta,h}$ und $\tilde{D}_h$ in dieser Reihenfolge. Die Blöcke werden nur besser approximiert, falls sie auch durch den Fehlerschätzer und das Dörfler-Marking ausgewählt wurden.

Im Bezug auf die gerade erklärte Vorgehensweise bei der Approximation der Blöcke wird in unserem Fall die genauere Approximation $\hat{A}_k$ durch die Anwendung einer festen Anzahl an ACA-Schritte auf jeden zulässigen Block von $V_{\Delta,h}$, V_{ij}, $i,j = 1,2,3$, $K_{\Delta,h}$ und $\tilde{D}_h$ erzeugt. Nochmals sei angemerkt, dass keine Approximationen für die festen Matrizen T_h, $S_{i,h}$, $i = 1,2,3$, berechnet werden. Alle Änderungen sind im Algorithmus 7 zusammengefasst.

Algorithmus 7 BACA für lineare Elastizität

1. Starte mit einer groben $\mathcal{H}$-Matrix-Approximation A_0 von A und setze $k = 0$.
2. Wende bei $\alpha \geq 0$ den Bramble-Pasciak-CG auf das lineare System $A_k x_k = b$ an, bis das Residuum folgende Bedingung erfüllt

$$\|b - A_k x_k\|_2 \leq \alpha \|(A_k - \hat{A}_k) x_k\|_2 \tag{11.2}$$

   (verwende x_{k-1} als Startvektor; $x_{-1} := 0$).
3. Finde bei gegebenen $0 < \theta < 1$, eine Menge markierter Blöcke $M_k \subset P_{\text{adm}}$ mit minimaler Kardinalität, so dass

$$\mathcal{E}_k(M_k) \geq \theta \, \mathcal{E}_k, \tag{11.3}$$

   wobei $\mathcal{E}_k^2(M) := \sum_{t \times s \in M} \|(A_k - \hat{A}_k)_{ts}(x_k)_s\|_2^2$ und $\mathcal{E}_k := \mathcal{E}_k(P_{\text{adm}})$.
4. Passe die Genauigkeit der Matrixapproximation in der folgenden Reihenfolge an:
   (i) Falls Blöcke im Operator D_{NN} ausgewählt werden, setze

$$D_{NN,k+1} = \hat{D}_{NN,k}.$$

   (ii) Falls Blöcke im Operator K_{ND} ausgewählt werden, setze

$$V_{DD,k+1} = \hat{V}_{DD,k}$$

   und $(K_{\Delta,h,k+1})_b = (\hat{K}_{\Delta,h,k})$ für alle im Operator $K_{\Delta,h}$ ausgewählten Blöcke b.
   (iii) Falls nur Blöcke im Operator V_{DD} ausgewählt werden, setze $(V_{\Delta,h,k+1})_b = (\hat{V}_{\Delta,h,k})_b$ oder $(V_{ij,k+1})_b = (\hat{V}_{ij,k})_b$ für alle ausgewählten Blöcke b.
   Alle Blöcke, die nicht selektiert werden, bleiben in ihrer Approximation unverändert.
5. Falls $\mathcal{E}_k > \varepsilon_{\text{BACA}}$ erhöhe k und gehe zu 2.

Nach Algorithmus 7 bzw. der verwendeten Verfeinerungsstrategie wird der Operator $V_{\Delta,h}$ am genauesten approximiert.

Die Anpassung der AMVM kann auf analoge Art und Weise erfolgen. Als Bezugsgröße wird in diesem Fall die rechte Seite f des betrachteten Gleichungssystem verwendet.

Die nächsten Schritte sind die Anpassung der Konvergenzanalyse auf die eben getroffenen Änderungen. Wir starten mit der Zuverlässigkeit des Fehlerschätzers, welche aus der Saturationsannahme folgt.

Lemma 11.1. *Sei die Saturationsannahme* (11.1) *erfüllt. Dann ist der Fehlerschätzer $\mathcal{E}_k$ zuverlässig, d.h. es gilt die Abschätzung*

$$\|b - Ax_k\|_2 \leq \frac{1+\alpha(1+c_{\text{sat}})}{1-c_{\text{sat}}}\|(A_k - \hat{A}_k)x_k\|_2 \leq \sqrt{27\mathcal{C}_{\text{sp}}\mathcal{L}}\,\frac{1+\alpha(1+c_{\text{sat}})}{1-c_{\text{sat}}}\mathcal{E}_k,$$

wobei $\mathcal{L}$ die maximale Tiefe der verwendeten Clusterbäume ist und $\mathcal{C}_{\text{sp}} := \max\{C_{\text{sp}V}, C_{\text{sp},K}, C_{\text{sp}D}\}$.

Beweis. Die erste Behauptung folgt mit der Bedingung (11.2) und der Saturationsannahme aus

$$\begin{aligned}
\|b - Ax_k\|_2 &\leq \|b - A_k x_k\|_2 + \|(A_k - \hat{A}_k)x_k\|_2 + \|\hat{A}_k x_k - Ax_k\|_2 \\
&\leq (\alpha+1)\|(A_k - \hat{A}_k)x_k\|_2 + c_{\text{sat}}\|A_k x_k - Ax_k\|_2 \\
&\leq (\alpha + 1 + c_{\text{sat}}\alpha)\|(A_k - \hat{A}_k)x_k\|_2 + c_{\text{sat}}\|b - Ax_k\|_2.
\end{aligned}$$

Die zweite Ungleichung ergibt sich aus der Zerlegung von $A = \sum_{l=1}^{\mathcal{L}} A^{(l)}$ in eine Summe von Levelmatrizen $A^{(l)}$. Da es mehrere Cluster-Bäume und eine maximale Tiefe $\mathcal{L}$ gibt, werden auf einer bestimmten Ebene keine weiteren Untermatrizen mehr existieren. In diesem Fall verwendet man die Null-Matrix für die verbleibenden Level. Somit erhalten wir

$$\begin{aligned}
\|(A_k - \hat{A}_k)x_k\|_2^2 &\leq \left(\sum_{l=1}^{\mathcal{L}} \|(A_k - \hat{A}_k)^{(l)}x_k\|_2\right)^2 \leq \mathcal{L}\sum_{l=1}^{\mathcal{L}} \|(A_k - \hat{A}_k)^{(l)}x_k\|_2^2 \\
&= \mathcal{L}\sum_{l=1}^{\mathcal{L}} \sum_{t\in T_I^{(l)}} \| \sum_{s:t\times s\in P} (A_k - \hat{A}_k)_{ts}(x_k)_s\|_2^2 \\
&\leq \mathcal{L}\sum_{l=1}^{\mathcal{L}} \sum_{t\in T_I^{(l)}} \left(\sum_{s:t\times s\in P} \|(A_k - \hat{A}_k)_{ts}(x_k)_s\|_2\right)^2 \\
&\leq 27\mathcal{C}_{\text{sp}}\mathcal{L}\sum_{l=1}^{\mathcal{L}} \sum_{t\in T_I^{(l)}} \sum_{s:t\times s\in P} \|(A_k - \hat{A}_k)_{ts}(x_k)_s\|_2^2 \\
&= 27\mathcal{C}_{\text{sp}}\mathcal{L} \sum_{t\times s\in P} \|(A_k - \hat{A}_k)_{ts}(x_k)_s\|_2^2 \\
&= 27\mathcal{C}_{\text{sp}}\mathcal{L}\mathcal{E}_k^2.
\end{aligned}$$

□

Der einzige Unterschied bei der Zuverlässigkeit des Fehlerschätzers $\mathcal{E}_k$ gegenüber dem Fehlerschätzer η_k bei der Laplace-Gleichung ist die Anzahl an Blöcken in der Systemmatrix A. Demnach erhalten wir für die Zuverlässigkeit bei der linearen Elastizität eine etwas höhere Konstante. Bei der Effizienz des Fehlerschätzers bzw. einer unteren Schranke für den Ausdruck $\|b - Ax_k\|_2$ ergeben sich keine

Unterschiede zum Laplace-Fall, sodass Lemma 8.5 gültig bleibt. Die Konvergenz des mit Algorithmus 7 beschriebenen Verfahrens kann analog zu Lemma 8.6 aus Kapitel 8 nachgewiesen werden. Im Fall der linearen Elastizität erhöht sich allerdings die Konstante bei γ geringfügig. Um die Theorie an dieser Stelle zu vervollständigen, wird der Konvergenzbeweis nochmals angefügt.

Lemma 11.2. *Angenommen A_* ist invertierbar und α ist hinreichend klein. Dann gilt*

$$\mathcal{E}_{k+1}^2 \leq q\,\mathcal{E}_k^2 + z_k, \tag{11.4}$$

wobei z_k eine Folge bezeichnet, die gegen Null konvergiert, und $q < 1$. Zudem konvergiert $\mathcal{E}_k$ gegen Null, d.h. $\lim_{k\to\infty} \mathcal{E}_k = 0$.

Beweis. Nutzen wir die Definitionen von A_{k+1} und $\mathcal{E}_{k+1}^2$, so folgt

$$\sum_{t\times s\in P} \|(A_{k+1}-\hat{A}_{k+1})_{ts}(x_{k+1})_s\|_2^2 = \sum_{t\times s\in M_k} \|(\hat{A}_k-\hat{A}_{k+1})_{ts}(x_{k+1})_s\|_2^2 + \sum_{t\times s\in P\setminus M_k} \|(A_k-\hat{A}_k)_{ts}(x_{k+1})_s\|_2^2.$$

Mit der Youngschen Ungleichung und (11.3) erhalten wir für den zweiten Term

$$\begin{aligned}
\sum_{t\times s\in P\setminus M_k} \|(A_k-\hat{A}_k)_{ts}(x_{k+1})_s\|_2^2 &\leq (1+\epsilon)\,\mathcal{E}_k^2(P\setminus M_k) + (1+1/\epsilon) \sum_{t\times s\in P\setminus M_k} \|(A_k-\hat{A}_k)_{ts}(x_{k+1}-x_k)_s\|_2^2 \\
&= (1+\epsilon)[\mathcal{E}_k^2 - \mathcal{E}_k^2(M_k)] + (1+1/\epsilon) \sum_{t\times s\notin M_k} \|(A_k-\hat{A}_k)_{ts}(x_{k+1}-x_k)_s\|_2^2 \\
&\leq (1+\epsilon)(1-\theta^2)\,\mathcal{E}_k^2 + (1+1/\epsilon) \sum_{t\times s\notin M_k} \|(A_k-\hat{A}_k)_{ts}(x_{k+1}-x_k)_s\|_2^2
\end{aligned}$$

für alle $\epsilon > 0$. Die letzte Summe kann abgeschätzt werden durch

$$\begin{aligned}
\sum_{t\times s\notin M_k} \|(A_k-\hat{A}_k)_{ts}(x_{k+1}-x_k)_s\|_2^2 &= \sum_{s\in T_I} \sum_{t:t\times s\notin M_k} \|(A_k-\hat{A}_k)_{ts}(x_{k+1}-x_k)_s\|_2^2 \\
&\leq \max_{t\times s\in P} \|(A_k-\hat{A}_k)_{ts}\|_2^2 \sum_{s\in T_I} \sum_{t:t\times s\notin M_k} \|(x_{k+1}-x_k)_s\|_2^2 \\
&\leq c\,c_{\mathrm{sp}}\mathcal{L}\|x_{k+1}-x_k\|_2^2 \max_{t\times s\in P} \|(A_k-\hat{A}_k)_{ts}\|_2^2 \\
&\leq c\,c_{\mathrm{sp}}\mathcal{L}\|A_{k+1}^{-1}\|_2^2\,\|A_{k+1}x_{k+1}-A_{k+1}x_k\|_2^2 \max_{t\times s\in P} \|(A_k-\hat{A}_k)_{ts}\|_2^2 \\
&\leq c\,c_{\mathrm{sp}}\mathcal{L}\|A_{k+1}^{-1}\|_2^2 \left(\|A_{k+1}x_{k+1}-b\|_2^2 + \|(A_{k+1}-A_k)x_k\|_2^2 + \|A_kx_k-b\|_2^2\right)\|A_k-\hat{A}_k\|_2^2,
\end{aligned}$$

wobei die Konstante von Schritt 2 zu Schritt 3 aus der Struktur der Matrix A_k resultiert, vgl. den Beweis von Lemma 11.1. Mit $\|b-A_kx_k\|_2^2 \leq \alpha^2\|(A_k-\hat{A}_k)x_k\|_2^2 \leq \alpha^2 c_{\mathrm{sp}}\mathcal{L}\,\mathcal{E}_k^2$ folgt mit der Wahl $\epsilon := \frac{1}{2}\theta^2/(1-\theta^2)$ die Abschätzung

$$\mathcal{E}_{k+1}^2 \leq (1-\frac{1}{2}\theta^2)\,\mathcal{E}_k^2 + \gamma[\mathcal{E}_{k+1}^2 + \mathcal{E}_k^2] + z_k, \tag{11.5}$$

wobei $\gamma := c\,\alpha^2\frac{2-\theta^2}{\theta^2}\left(c_{\mathrm{sp}}\mathcal{L}\|A_k-\hat{A}_k\|_2\|A_{k+1}^{-1}\|_2\right)^2$ und

$$z_k := \sum_{t\times s\in M_k} \|(\hat{A}_k-\hat{A}_{k+1})_{ts}(x_{k+1})_s\|_2^2 + 3(1+1/\epsilon)c_{\mathrm{sp}}L\|A_k-\hat{A}_k\|_2^2\,\|A_{k+1}^{-1}\|_2^2\|A_{k+1}-A_k\|_2^2\|x_k\|_2^2$$

gegen Null konvergiert. Dies wiederum ist eine Konsequenz aus $A_k^{-1} \to A_*^{-1}$ für $k \to \infty$ und (8.5). Zu beachten ist, dass $\{x_k\}_{k\in\mathbb{N}}$ wegen

$$\begin{aligned}\|x_k\|_2 &\leq \|A_k^{-1}\|_2 \left(\|b\|_2 + \|A_k x_k - b\|_2\right) \leq \|A_k^{-1}\|_2 \left(\|b\|_2 + \alpha\|(A_k - \hat{A}_k)x_k\|_2\right) \\ &\leq \|A_k^{-1}\|_2\|b\|_2 + \alpha\|A_k^{-1}\|_2(\|A_k\|_2 + \|\hat{A}_k\|_2)\|x_k\|_2\end{aligned}$$

und einem hinreichend kleinem α beschränkt ist. Die Wahl von α, sodass $\gamma < \theta^2/4$ erfüllt ist, führt mit (11.5) und

$$q := \frac{1 - \frac{1}{2}\theta^2 + \gamma}{1 - \gamma} < 1$$

zum ersten Teil der Aussage.

Beim Beweis der zweiten Aussage halten wir uns an die Idee des Fehlerschätzer-Reduktionsprinzip, welches in [8] vorgestellt wurde. Sei $z > 0$ eine Zahl mit $z_k \leq z$ für alle k. Mit der Fehlerschätzer-Reduktion (11.4) folgt

$$\begin{aligned}\mathcal{E}_{k+1}^2 &\leq q\,\mathcal{E}_k^2 + z_k \leq q^2\,\mathcal{E}_{k-1}^2 + q\,z_{k-1} + z_k \leq \ldots \leq q^{k+1}\mathcal{E}_0^2 + \sum_{i=0}^{k} q^{k-i} z_i \\ &\leq q^{k+1}\mathcal{E}_0^2 + z\sum_{l=0}^{k} q^l \leq \mathcal{E}_0^2 + \frac{z}{1-q}.\end{aligned}$$

Damit ist die Folge $\{\mathcal{E}_k\}_{k\in\mathbb{N}_0}$ beschränkt und wir definieren $M := \limsup_{k\to\infty} \mathcal{E}_k^2$. Benutzen wir ein weiteres Mal die Fehlerschätzer-Reduktion (11.4), so führt das zu

$$M = \limsup_{k\to\infty} \mathcal{E}_{k+1}^2 \leq q \limsup_{k\to\infty} \mathcal{E}_k^2 + \limsup_{k\to\infty} z_k = qM.$$

Schließlich folgt $M = 0$ und wir erhalten

$$0 \leq \liminf_{k\to\infty} \mathcal{E}_k \leq \limsup_{k\to\infty} \mathcal{E}_k = 0$$

und letztendlich $\lim_{k\to\infty} \mathcal{E}_k = 0$. □

Die Konvergenz der BACA im Fall der linearen Elastizität ist nun eine Folge der Zuverlässigkeit von $\mathcal{E}_k$.

Theorem 11.3. *Die Residuen $r_k := b - Ax_k$ der Folge $\{x_k\}_{k\in\mathbb{N}}$, welche durch Algorithmus 7 konstruiert wurde, konvergiert gegen Null.*

Beweis. Mit Lemma 11.1 und Lemma 11.2 erhalten wir

$$\|b - Ax_k\|_2 \leq \sqrt{27 c_{\mathrm{sp}}\mathcal{L}}\,\frac{1 + \alpha(1 + c_{\mathrm{sat}})}{1 - c_{\mathrm{sat}}}\,\mathcal{E}_k \to 0.$$

□

11.2 Berechnung von Spannungen im Inneren - Kollokationsmatrizen

Angenommen, wir haben in einem vorherigen Schritt die Randdaten (t_h, w_h), d.h.

$$t_h = \sum_{i=1}^{M} t_i \varphi_i \quad \text{und} \quad w_h = \sum_{j=1}^{N} w_j \psi_j$$

mit den Koeffizientenvektoren $t, w \in \mathbb{R}^3$ berechnet, dann kann die Lösung u_h in Ω durch

$$u_h(x) = \sum_{j=1}^{M} t_j \int_{\partial\Omega} S(x,y)\varphi_j(y)\,\mathrm{d}s_y - \sum_{k=1}^{N} w_k \int_{\partial\Omega} \gamma_{1,y}^{\mathrm{int}} S(x,y)\psi_k(y)\,\mathrm{d}s_y,$$

für alle $x \in \Omega$ ausgewertet werden. Die Spannungen $\sigma(u_h, x)$ können zudem unter Benutzung der Ableitungen

$$\partial_{x_i} u_h(x) = \sum_{j=1}^{M} t_j \int_{\partial\Omega} \partial_{x_i} S(x,y)\varphi_j(y)\,\mathrm{d}s_y - \sum_{k=1}^{N} w_k \int_{\partial\Omega} \partial_{x_i}(\gamma_{1,y}^{\mathrm{int}} S(x,y))\psi_k(y)\,\mathrm{d}s_y, \tag{11.6}$$

für $i = 1, 2, 3$ zusammen mit dem Hook'schen Gesetz angegeben werden. Falls das Ziel ist, die Verformungen und Spannungen an mehreren Punkten $x_1, \ldots, x_l$, $l \in \mathbb{N}$, zu analysieren, so kann dies als vektorwertige Gleichung

$$v := \tilde{V}_h t - \tilde{W}_h u$$

mit $v = [u_h(x_i)]_{i=1,\ldots,l}$ angesehen werden. Da die beiden diskretisierten Operatoren

$$\tilde{V}_h := \left[\int_{\partial\Omega} S(x_i,y)\varphi_j(y)\,\mathrm{d}s_y\right]_{ij} \quad \text{and} \quad \tilde{W}_h := \left[\int_{\partial\Omega} \gamma_{1,y}^{\mathrm{int}} S(x_i,y)\psi_k(y)\,\mathrm{d}s_y\right]_{ik}$$

mit $i = 1, \ldots, l$, $j = 1, \ldots, M$, und $k = 1, \ldots, N$ vom Kollokationstyp sind, können wir die Berechnung der Verformungen und Spannungen mit der in Kapitel 6 eingeführten adaptiven Matrix-Vektor Multiplikation beschleunigen. Die Auswertung der Spannungen unter Benutzung der Ableitungen $\partial_{x_i} u_h$ $i = 1, 2, 3$, kann in ähnlicher Form unter Verwendung von (11.6) durchgeführt werden.

Nach der Einführung und Analyse der neuen adaptive Algorithmen, folgen im letzten Kapitel die numerischen Tests zur Laplace Gleichung und der linearen Elastizität.

12. Numerische Resultate

Neben numerischen Tests zur Laplace- und Lamé-Gleichung wollen wir uns auch mit der neuen Punktauswahl bei der ACA, siehe Kapitel 3, beschäftigen. Im nachfolgenden Abschnitt ist ein Problem angeführt, bei dem mit der üblichen Punktauswahl keine Konvergenz beim ACA beobachtet werden kann. Derartige Schachmuster-Probleme können zum Beispiel bei der Diskretisierung des Doppelschichtpotentials auftreten.

Neben der eigenen Implementierung der vorgestellten Algorithmen wurde die Bibliothek „A$\mathcal{H}$med“ (siehe [15]) verwendet. Die Berechnungen sind im Fall der Laplace-Gleichung auf einem Computer mit Intel(R) Core(TM) i5-6500 CPU 3.20 GHz und im Fall der linearen Elastizität auf einem PC mit Intel(R) Core(TM) i7-6700HQ CPU 2.60 GHz durchgeführt worden.

12.1 ACA unter Berücksichtigung der Fülldichte

Wir wenden die ACA zusammen mit der Pivotstrategie basierend auf der Fülldichte bzgl. x an, um einen einzelnen Block $A \in \mathbb{R}^{N\times N}$ mit den Einträgen

$$a_{ij} = \frac{(x_i - y_j)\cdot n_{y_j}}{|x_i - y_j|^3}, \quad i,j = 1,\dots,N,$$

zu approximieren. Die Punkte x_i werden aus der Menge $D_1 \cup D_2$ gewählt und die Punkte y_j aus der Menge $D_3 \cup D_4$. Der Vektor n_{y_j} bezeichnet den Einheitsnormalenvektor in y_j auf dem Rand des Gebiets aus Abb. 12.1. Die zwei kleinsten Seitenlängen dieses Gebiets sind 1 und der Abstand zwischen $D_1 \cup D_2$ und $D_3 \cup D_4$ ist 9. Ein ähnliches Problem wurde in früheren Artikeln schon präsentiert und diskutiert, siehe [23, 15].

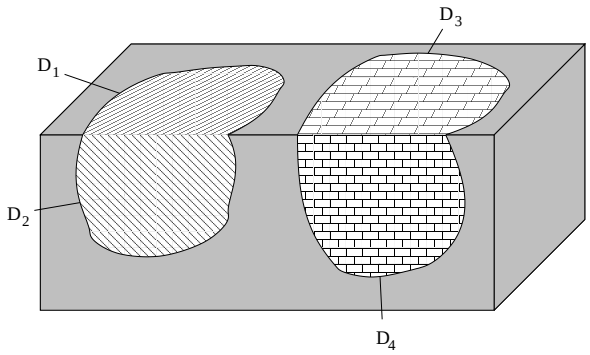

Abb. 12.1: Mengen auf der Oberfläche , vgl. [15].

Falls die Punkte x_i, $i = 1,\dots,N$, and y_j, $j = 1,\dots,N$, so geordnet sind, dass die ersten Punkte in D_1 bzw. D_3 liegen, dann hat A die Struktur

$$A = \begin{bmatrix} 0 & A_{12} \\ A_{21} & 0 \end{bmatrix}.$$

Wie zuvor bereits erwähnt wurde, siehe auch [15], konvergiert der ACA mit der üblichen Punkt- bzw. Pivotauswahl nicht, da die Pivotpunkte in einem der Blöcke A_{12} oder A_{21} verweilen, während der andere Block überhaupt nicht approximiert wird. Die neue Pivotstrategie, welche die Pivots nach der Fülldichte auswählt, führt zu der gewünschten Konvergenz wie Abb. 12.2 zeigt.

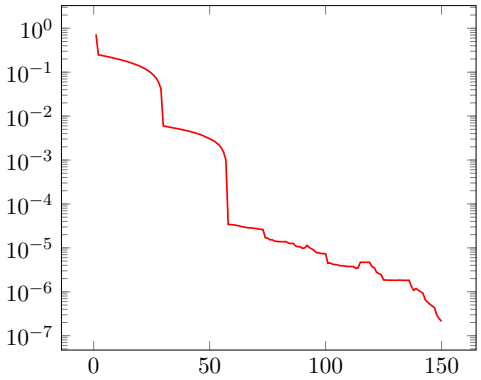

Abb. 12.2: Fehler gegen Rang der Approximation basierend auf der Fülldichte.

12.2 Die Laplace Gleichung und AMVM

Betrachtet wird die numerische Lösung eines Randwertproblems für die Laplace-Gleichung, d.h.

$$-\Delta u(x) = 0, \quad x \in \Omega, \quad u(x) = g(x), \quad x \in \Gamma,$$

mit Hilfe der Randelementmethode. Der Randintegralformulierung folgend hat das Doppelschichtpotential die Darstellung

$$\mathcal{K}\xi(x) := \int_\Gamma \xi(y)\partial_{n(y)} S(x-y)\,\mathrm{d}s_y,$$

wobei

$$S(r) = \begin{cases} -\frac{1}{2\pi}\ln|r|, & d=2, \\ \frac{1}{(d-2)\omega_d}\frac{1}{|r|^{d-2}}, & d>2, \end{cases} \qquad r \in \mathbb{R}^d \setminus \{0\},$$

die Fundamentallösung der Laplace-Gleichung bezeichnet. Unter Verwendung einer Diskretisierung bestehend aus stückweisen konstanten Funktionen ψ_i^0, $i = 1, ..., N$, auf einer Partitionierung $\mathcal{T}$ von Γ in N reguläre Dreiecke und einer stückweisen linearen Approximation für die Randdaten g, d.h.

$$g \approx g_h \in \mathcal{P}_1(\mathcal{T}) = \mathrm{span}\{\psi_1^1, ..., \psi_M^1\},$$

konzentrieren wir uns auf die Multiplikation des diskretisierten Doppelschichtpotentials $K \in \mathbb{R}^{M\times N}$, $k_{ij} = (\mathcal{K}\psi_j^1, \psi_i^0)$ mit dem gegebenem Datenvektor g. In den folgenden numerische Beispielen werden die von der ACA und AMVM benötigten Rechenzeiten und Speicheranforderungen bei gleichbleibenden relativen Fehler $e_h = \|u-u_h\|_{L^2}/\|u\|_{L^2}$ miteinander verglichen. Die Blöcke der Matrix werden bei der ACA bei einer minimalen Blockgröße $b_{\min} = 15$ und einer Genauigkeit $\varepsilon_{\mathrm{ACA}} = 10^{-6}$ approximiert. Die Genauigkeit von AMVM liegt bei $\varepsilon_{\mathrm{AMVM}} = 10^{-6}$.

Als erstes Beispiel wird der Datenvektor

$$g = (\underbrace{1, ..., 1}_{10\,mal}, 0, ..., 0)^T$$

auf einer Triangulierung von $\Omega = B_1(0) \subset \mathbb{R}^3$ in 1280 Dreiecke und 642 Punkte betrachtet. Für die Multiplikation des Doppelschichtoperators mit g benötigt das ACA-Verfahren 0.71s. Die adaptive Matrix-Vektor Multiplikation erfolgt mit 0.44s in nur 60% der Zeit des ACA. Dieses Beispiel ist durch die großen strukturellen Unterschiede im Datenvektor sehr auf die Funktionsweise der AMVM

zugeschnitten. Im nächsten numerischen Test wird ein Datenvektor verwendet der zwar noch strukturelle Unterschiede aufweist, diese aber deutlich geringer sind. Wir setzen

$$g(x) = S(x - p), \quad p = (1.5, 0, 0)^T$$

und betrachten wiederum die Multiplikation des Doppelschichtoperators mit g. Tabelle 12.1 zeigt die numerischen Resultate der AMVM im Vergleich zur ACA mit $\theta = 0.7$.

N	M	start rank	AMVM $\|b - b_k\|_2$	AMVM e_h	AMVM Zeit	AMVM Speicher	ACA e_h	ACA Zeit	ACA Speicher
1 280	642	7	1.73e-06	0.033	0.61s	4.08 MB	0.033	0.71s	5.03 MB
5 120	2 562	8	1.53e-06	0.010	3.21s	27.46 MB	0.010	4.65s	37.86 MB
20 480	10 242	9	2.26e-06	0.003	18.9s	199.12 MB	0.003	28.47s	257.54 MB

Tab. 12.1: Numerische Resultate der AMVM im Vergleich zu ACA beim Doppelschichpotential.

Mit demselben relativen Fehler e_h der approximierten Lösung u_h, benötigt die AMVM weniger Speicher und weniger Rechenzeit im Vergleich zur ACA, um die Matrix K mit dem gegebenen Datenvektor g zu multiplizieren. Ein Blick auf den Fehlerschätzer γ_k, siehe Abbildung 12.3, zeigt, dass γ_k den Fehler $\|b - b_k\|$ zuverlässig und effizient anzeigt.

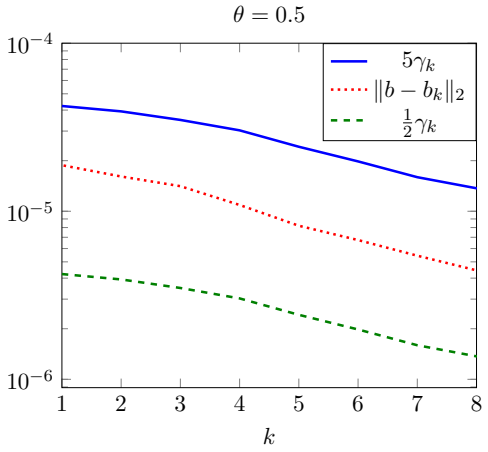

Abb. 12.3: Quality of the error estimator γ_k.

Die Beobachtungen demonstrieren, dass der neu entwickelte adaptive Algorithmus zur approximativen Matrix-Vektor-Multiplikation bei einfacheren Problemen wie der Laplace-Gleichung gut funktioniert. Im nächsten Schritt soll die neue Methode bei komplizierteren Problemen wie der linearen Elastizität zum Einsatz kommen. Bevor wir uns darum kümmern, bleiben wir im nächsten Abschnitt zunächst bei der Laplace-Gleichung und betrachten die neue Methode BACA genauer.

12.3 BACA bei der Laplace Gleichung

Die numerischen Berechnungen für die Laplace-Gleichung sind in drei Teile untergliedert. Zuerst untersuchen wir den Speicherbedarf bei der BACA. Danach wird die Qualität des verwendeten Fehlerschätzers diskutiert. Zuletzt behandelt der dritte Teil die Beschleunigung der Matrixapproximation

und die Lösung des Randwertproblems im Vergleich zur ACA. Um diese Resultate mit den Ergebnissen, welche wir durch die ACA erhalten haben, vergleichen zu können, wollen wir denselben Fehler für u_h erzielen. Im Folgenden gibt

$$e_h := \frac{\|u - u_h\|_{L^2}}{\|u\|_{L^2}}$$

den relativen Fehler der Lösung u_h an. Die einzelnen Approximationsschritte werden in beiden Fällen ohne Parallelisierung ausgeführt. Dabei sei angemerkt, dass wegen der Lokalität des Fehlerschätzers die BACA parallelisiert werden kann. Um das auftretende lineare Gleichungssystem zu lösen, benutzen wir das in Kapitel 8.1 beschriebene CG-Verfahren ohne Vorkonditionierung. Im Fall der BACA wird die Genauigkeit des iterativen Lösers an die Größe von $\|(A_k - \hat{A}_k)x_k\|_2$ angepasst, siehe Lemma 8.4. Bei allen numerischen Untersuchungen auf Basis der ACA beträgt die Genauigkeit des CG-Verfahrens 10^{-8}. Als numerische Beispiele betrachten wir eine Familie von Randwertproblemen, bei welchen die Singularität auf der Rechte Seite zum Rand des Berechnungsgebiets Ω bewegt wird, d.h.

$$-\Delta u(x) = 0 \qquad \text{in } \Omega := B_1(0), \tag{12.1a}$$

$$u(x) = S(x - p_i) \qquad \text{auf } \partial\Omega,\ i = 1, 2, 3, 4, \tag{12.1b}$$

für $p_i = (x_i, 0, 0)^T$ mit $x_1 = 10.0$, $x_2 = 1.5$, $x_3 = 1.1$, $x_4 = 1.05$. In diesen Tests werden die folgenden Parameter verwendet: die minimale Blockgröße $b_{\min} = 15$, die blockweise Genauigkeit $\varepsilon_{\text{ACA}} = 10^{-6}$ der ACA, und der Zulässigkeitsparameter $\beta = 0.8$.

12.3.1 Kompressionsraten

Wir beginnen mit einer uniformen Unterteilung der Einheitskugel in 642 Punkte und 1280 Dreiecke für das Randwertproblem (12.1). An dieser Stelle sei nochmals an das erste numerische Experiment erinnert. Da das ACA-Verfahren die rechte Seite nicht berücksichtigt, hängt auch die Approximation nicht von der Wahl von p_i ab. Daher erhalten wir das linke Bild von Abbildung 12.4, nachdem die ACA auf den diskreten Integraloperator angewendet wurde in jedem der Fälle $i = 1, ..., 4$. Tabelle 12.2 zeigt die Resultate des BACA-Algorithmus für einen Parameter $\theta = 0.9$. In diesem Fall sind die Ränge von $\hat{A}_k$ der Matrix A_k um zwei ACA-Schritte voraus. Der Parameter α von Lemma 8.4 wird auf 100 gesetzt. Da weniger genaue Resultate erwartet werden können, wenn die Singularität näher an den Rand bewegt wird, vgl. die Spalte e_h in Tabelle 12.2, müssen unterschiedliche Fehlerschranken $\varepsilon_{\text{BACA}}$ verwendet werden. Verglichen mit der Approximation, die wir durch die ACA erhalten haben, benötigen wir weniger Speicheranforderungen für die Matrixapproximation $\hat{A}_k$. Die in Tabelle 12.2

	BACA						ACA		
	blockweiser Rang A_0	$\varepsilon_{\text{BACA}}$	$\|b - Ax_k\|_2$	e_h	Speicher (MB)	Rel. Sp. (%)	e_h	Speicher (MB)	Rel. Sp. (%)
p_1	6	1e-08	1.89e-08	0.002	3.43	54.9	0.002	3.85	61.5
p_2	4	5e-06	6.08e-06	0.033	2.86	45.7	0.033	3.85	61.5
p_3	3	1e-04	1.48e-04	0.264	2.40	38.4	0.264	3.85	61.5
p_4	2	5e-04	6.40e-04	0.951	2.10	33.5	0.951	3.85	61.5

Tab. 12.2: Speicherbedarf der Approximation konstruiert durch BACA für vier Positionen der Singularität verglichen mit ACA, wobei der Fehler e_h auf dem gleichen Level behalten wird.

festgestellte Speicherreduzierung sowie der geringere relative Speicher (der Speicher für die Approximation dividiert durch den Speicher für A) sind auch aus Abbildung 12.4, in welcher die zugehörige

Approximation $\hat{A}_k$ von A zusammen mit den blockweisen Rängen (grüne Blöcke) gezeigt ist. Die darin enthaltenen roten Blöcke werden Eintrag für Eintrag konstruiert ohne Rücksicht darauf, ob sie zulässig sind oder nicht. Rote Blöcke müssen demnach komplett berechnet werden, um die geforderte Genauigkeit zu erreichen. Beispiel (12.1) wurde gewählt, um verschiedene rechte Seiten zu unter-

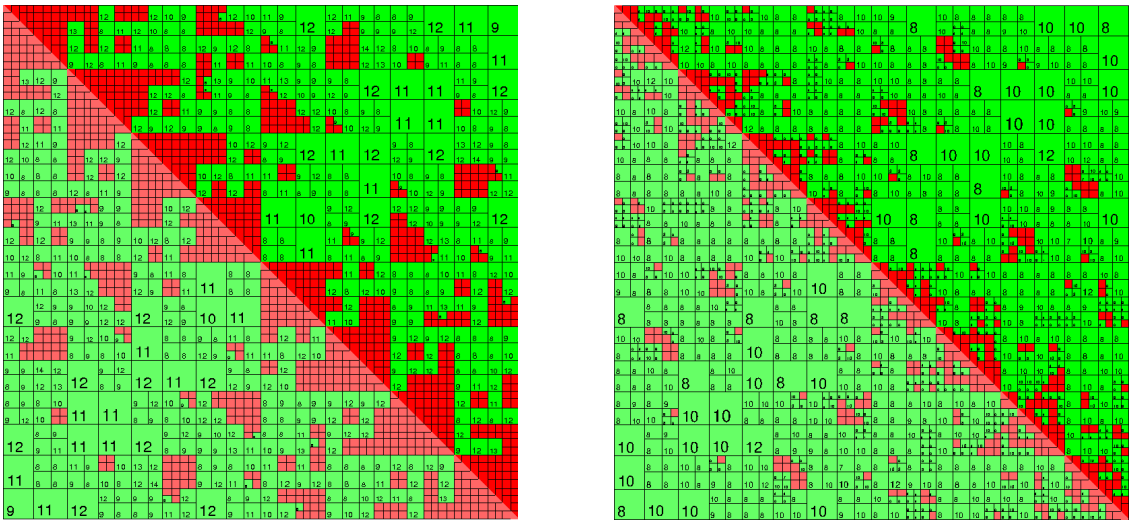

Abb. 12.4: Approximation der Blöcke mit ACA (links) und mit BACA im Fall p_1 (rechts). Die Blöcke mit Nummern enthalten ihre Ränge.

suchen, die zu strukturellen Unterschieden in der jeweiligen Lösung führen. Dabei ist zu beachten, dass der Einfluss der Singularität auf die Lösung umso stärker ist, je kleiner der Abstand von p_i zum Rand von Ω wird. Daher sind für p_i nahe des Randes bestimmte Teile des diskreten Integraloperators wichtiger für die Genauigkeit der Lösung als andere. Selbst für einen relativ großen Abstand, d.h. $p_1 = (10, 0, 0)^T$, sind kleinere Ränge und damit eine bessere Kompressionsrate als bei der ACA zu beobachten; siehe Abb. 12.4. Dieser Effekt ist noch stärker, wenn sich p_i dem Rand annähert. Die durchschnittlichen Ränge (die Summe der Ränge aller Blöcke dividiert durch die Anzahl der Blöcke) können in Tabelle 12.3 betrachtet werden. So zeigen Tabelle 12.2 und Abbildung 12.4 an, dass der

	ACA	BACA			
Matrix		p_1	p_2	p_3	p_4
A_k	12.54	7.44	5.00	3.71	3.08
$\hat{A}_k$	—	8.52	6.62	5.38	5.01

Tab. 12.3: Durchschnittliche Ränge der Matrizen $\hat{A}_k$ und A_k konstruiert durch die BACA für $p_1, \ldots, p_4$ verglichen mit der ACA.

vorgeschlagene Fehlerschätzer diejenigen Matrixblöcke erkennt, die für das jeweilige Problem wichtig sind. Daher ist der BACA-Algorithmus in der Lage, eine $\mathcal{H}$-Matrixapproximation zu konstruieren, die besonders dazu geeignet ist, ein lineares System für eine bestimmte rechte Seite zu lösen.

Die verbesserten Speicheranforderungen sind auch bei feineren Triangulationen des Randes zu beobachten. Tabelle 12.4 zeigt die Ergebnisse für das Randwertproblem (12.1) im Fall $i = 3$ für mehrere verschiedene Freiheitsgrade N. Neben dem relativen Speicher und den Speicheranforderungen wird die Anzahl der berechneten Einträge für die Konstruktion von $\hat{A}_k$ angegeben. Da erwartet wird, dass die Lösung u_h umso genauer ist, je größer N ist, muss die Genauigkeit $\varepsilon_{\text{BACA}}$ des

Fehlerschätzers an N angepasst werden. Alle anderen Parameter bleiben unverändert. Im Vergleich dazu sind die Ergebnisse der ACA, angewendet auf das Randwertproblem (12.1) im Fall $i = 3$, in Tabelle 12.5 zu sehen.

N	$\varepsilon_{\text{BACA}}$	$\|b - Ax_k\|_2$	e_h	Anzahl berechneter Einträge	Speicher (MB)	Rel. Sp. (%)
1 280	1e-04	4.14e-04	0.264	3.22e05	2.48	39.7
7 168	1e-05	2.31e-05	0.077	3.32e06	25.47	13.0
28 672	1e-06	2.56e-06	0.023	2.10e07	160.98	5.1
114 688	1e-06	2.13e-06	0.007	8.76e07	837.92	1.7

Tab. 12.4: Anzahl der berechneten Einträge, Speicher und relativer Speicher für die von der BACA konstruierten Approximation für p_3.

N	e_h	Anzahl berechneter Einträge	Speicher (MB)	Rel. Sp. (%)
1 280	0.264	5.01e05	3.85	61.5
7 168	0.077	4.91e06	37.53	19.8
28 672	0.023	2.66e07	203.70	6.5
114 688	0.007	1.35e08	1034.56	2.1

Tab. 12.5: Anzahl der berechneten Einträge, Speicher und relativer Speicher für die von der ACA konstruierten Approximation für p_3.

Aus den eben genannten Gründen benötigt der BACA deutlich weniger originale Matrixeinträge als die übliche Konstruktion der $\mathcal{H}$-Matrixapproximation mittels ACA, um ungefähr den gleichen relativen Fehler von u_h zu erhalten.

12.3.2 Verhalten des Fehlerschätzers

In den folgenden Tests validieren wir den in (7.3) eingeführten Fehlerschätzer η_k. Zuerst wird die Aussage bzgl. der Zuverlässigkeit von Lemma 8.4 zusammen mit der unteren Schranke $\|(A_k - \hat{A}_k)x_k\|_2$ von Lemma 8.5 untersucht. Auch hier wird die mit der BACA kombinierte Randelementmethode mit dem in (12.1) beschriebenen Problem und einer Triangulation mit 14338 Punkten und 28672 Dreiecken getestet. Die Genauigkeit $\varepsilon_{\text{BACA}}$ des Fehlerschätzers wird auf 10^{-7} gesetzt. Wir starten im BACA mit einer groben Approximation A_0, die durch Anwendung von vier ACA-Schritten auf jeden Block erzeugt wurde. Tabelle 12.6 und Abbildung 12.5 enthalten die Ergebnisse für $\alpha = 1/2$ und drei Verfeinerungsparameter θ. Die Ränge von $\hat{A}_k$ liegen um drei ACA-Schritte vor A_k.

Der vorgeschlagene Fehlerschätzer η_k schätzt das Residuum $\|b - Ax_k\|_2$ in geeigneter Weise. Der Ausdruck $\frac{1}{2}\|(A_k - \hat{A}_k)x_k\|_2$ kann als untere Schranke für das Residuum dienen. Tabelle 12.6 zeigt, dass die Konvergenz in Bezug auf k umso schneller ist, je näher der Verfeinerungsparameter θ bei 1 liegt. Ein größerer Parameter θ in (8.4) führt zu einer größeren Menge M_k markierter Blöcke, was die Generierung redundanter Informationen in A_{k+1} fördert. Umgekehrt führen kleine Verfeinerungsparameter θ gewöhnlich zu Matrixapproximationen mit geringerem Speicherbedarf. Die numerischen Ergebnisse bestätigen also die theoretischen Erkenntnisse von Lemma 8.4 und Lemma 8.5. Der vorgeschlagene Fehlerschätzer η_k ist zuverlässig und $\frac{1}{2}\|(A_k - \hat{A}_k)x_k\|_2$ kann als untere Grenze für den residualen Fehler verwendet werden.

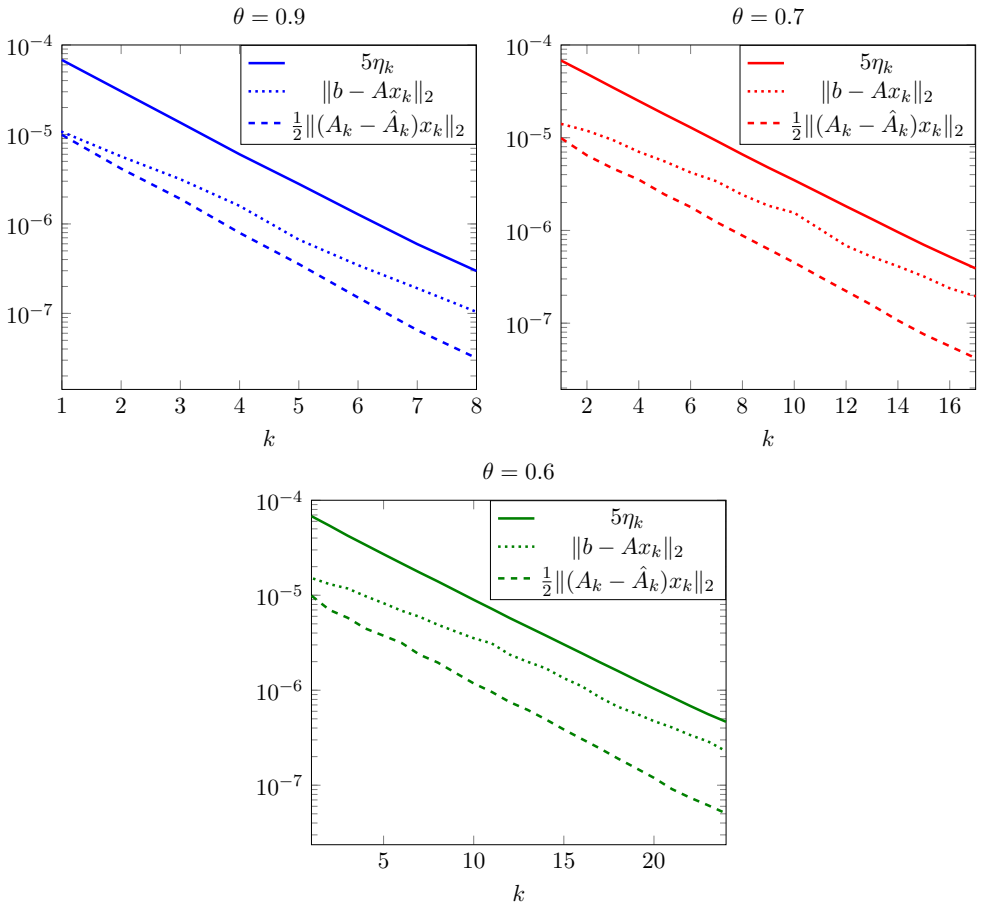

Abb. 12.5: Verhalten des Fehlerschätzers, des Residuum und der unteren Schranke $\|(A_k - \hat{A}_k)x_k\|_2$ für mehrere Parameter θ.

12.3.3 Beschleunigung der numerischen Berechnung

Die bisher erzielten Ergebnisse stellen nur einen Teil der Ziele dar, die wir mit der Erweiterung der ACA verfolgen. Wie bereits in der Einleitung erwähnt, sind wir auch an einer Beschleunigung der Matrixapproximation und an einer Beschleunigung des Lösungsprozesses interessiert. Da die Berechnung der Matrixeinträge bei weitem der zeitaufwendigste Teil der Randelementmethode ist und wir gesehen haben, dass der BACA-Algorithmus weniger Einträge benötigt, kann mit verbesserten Rechenzeiten gerechnet werden.

Wir betrachten erneut die Familie der Randwertprobleme (12.1). Die Ränge der Matrixapproximationen $\hat{A}_k$ liegen zwei ACA-Schritte vor A_k und wir wählen $\alpha = 100$. Das Verhältnis zwischen der von der BACA benötigten Zeit und der benötigten Zeit von ACA kann in Abbildung 12.6 und 12.7 für zwei verschiedene Triangulationen beobachtet werden. Die beiden Abbildungen 12.6 und 12.7 zeigen, dass wir selbst im Fall von p_1, in welchem die strukturellen Unterschiede auf der rechten Seite gering sind, einen geringeren Zeitverbrauch erzielen. Wenn die Singularität näher an den Rand von Ω gerückt wird, beschleunigt sich das BACA-Verfahren kontinuierlich.

Einer der größten Vorteile der ACA ist seine logarithmisch-lineare Komplexität. Um dieses Verhalten zu beobachten, wird die mit 642 Punkten und 1280 Dreiecken diskretisierte Kugel betrachtet. Wir behalten die resultierende Geometrie bei und erhöhen die Anzahl der Dreiecke N für das Randwertproblem mit p_1. Die Zeitspalten von Tabelle 12.7 zeigen die erwartete Komplexität der ACA und auch

	η_k			$\|b - Ax_k\|_2$		
k	$\theta = 0.6$	$\theta = 0.7$	$\theta = 0.9$	$\theta = 0.6$	$\theta = 0.7$	$\theta = 0.9$
1	1.36e-05	1.36e-05	1.36e-05	1.52e-05	1.41e-05	1.07e-05
2	1.08e-05	9.77e-06	6.07e-06	1.31e-05	1.19e-05	5.68e-06
3	8.49e-06	6.96e-06	2.71e-06	1.18e-05	9.45e-06	3.16e-06
4	6.78e-06	4.98e-06	1.20e-06	9.78e-06	7.05e-06	1.59e-06
5	5.41e-06	3.57e-06	5.60e-07	8.24e-06	5.55e-06	6.68e-07
6	4.33e-06	2.58e-06	2.56e-07	6.87e-06	4.22e-06	3.46e-07
7	3.47e-06	1.85e-06	1.19e-07	5.93e-06	3.39e-06	1.91e-07
8	2.80e-06	1.32e-06	5.96e-08	4.90e-06	2.42e-06	1.04e-07
9	2.24e-06	9.52e-07	—	4.15e-06	1.85e-06	—
10	1.79e-06	6.96e-07	—	3.53e-06	1.55e-06	—
11	1.44e-06	5.04e-07	—	3.11e-06	1.03e-06	—
12	1.15e-06	3.64e-07	—	2.37e-06	6.84e-07	—
13	9.31e-07	2.65e-07	—	1.99e-06	5.19e-07	—
14	7.53e-07	1.92e-07	—	1.70e-06	4.11e-07	—
15	6.07e-07	1.40e-07	—	1.34e-06	3.19e-07	—

Tab. 12.6: Numerische Ergebnisse für η_k verglichen mit dem Residuum für verschiedene θ.

die des neuen BACA-Verfahrens. Der Wachstumsfaktor von N beträgt vier und der Wachstumsfaktor der Berechnungszeit liegt zwischen fünf und sechs.

	BACA			ACA	
N	$\|b - Ax_k\|_2$	e_h	Zeit (s)	e_h	Zeit (s)
1 280	1.58e-08	0.002	0.27	0.002	0.31
5 120	1.49e-08	0.002	1.46	0.002	1.83
20 480	1.08e-08	0.003	7.34	0.003	10.20
81 920	4.22e-09	0.004	38.15	0.004	54.15

Tab. 12.7: Zeitverbrauch der BACA für p_1 und $\varepsilon_{\text{BACA}} = 5 \cdot 10^{-8}$, verglichen mit ACA, wobei e_h auf dem gleichen Niveau gehalten wird.

12.3.4 Auswirkungen auf zum Teil zu stark verfeinerte Gebiete

In den folgenden Tests werden ACA und BACA auf Triangulationen angewendet, die teilweise zu stark verfeinert sind. Die Oberfläche des Ellipsoiden $\Omega = \{x \in \mathbb{R}^3 \ : \ x_1^2 + x_2^2 + x_3^2/9 = 1\}$ ist in der mittleren Region höher aufgelöst als auf der übrigen Geometrie; siehe Abbildung 12.8.

Die numerischen Ergebnisse für die BACA bezogen auf das Ellipsoid (p_2, $\varepsilon_{\text{BACA}} = 10^{-6}$) mit einer wachsenden Anzahl von Dreiecken sind in Tabelle 12.8 enthalten. Die Genauigkeit der Blockapproximation für die ACA beträgt $\varepsilon_{\text{ACA}} = 10^{-6}$. Die minimale Blockgröße ist ebenfalls in Tabelle 12.8 angegeben, die wir mit zunehmender Größe des Problems erhöhen. Um die Ergebnisse vergleichen zu können, halten wir den relativen Fehler auf dem gleichen Niveau.

Ein Blick in die Zeitspalten der Tabellen 12.8 und 12.9 zeigt, dass das BACA-Verfahren signifikant schneller ist als die ACA. Für das größte Problem ist die BACA mehr als doppelt so schnell wie ACA. Der Grund für die Beschleunigung liegt darin, dass die Lösung nicht von der lokalen Überverfeinerung profitiert. Während der auf der ACA basierende Löser dies nicht erkennen kann, führt

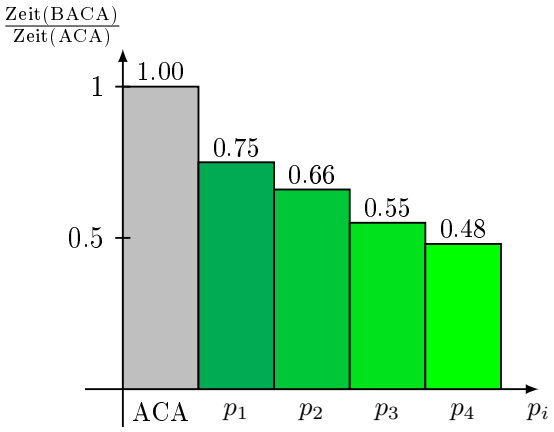

Abb. 12.6: Zeitverhältnis (BACA/ACA) für eine Triangulation mit 7 168 Dreiecken.

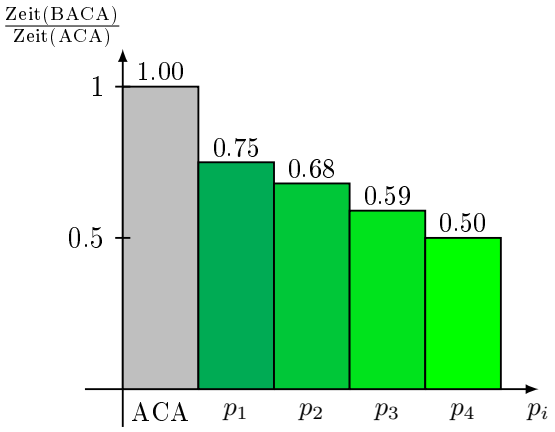

Abb. 12.7: Zeitverhältnis (BACA/ACA) für eine Triangulation mit 28 672 Dreiecken.

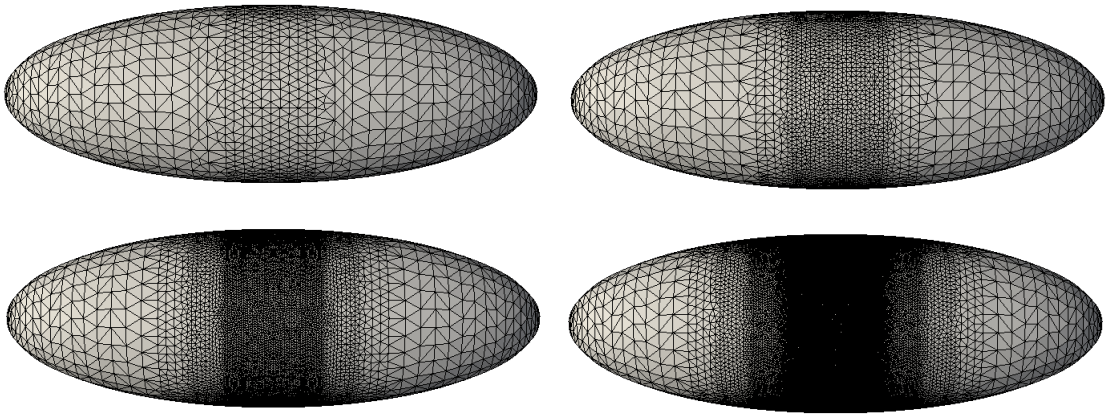

Abb. 12.8: Gitter auf dem betrachteten Ellipsoiden (gröbstes Level oben links, feinstes Level unten rechts).

die in BACA verwendete Kombination aus Löser und Fehlerschätzer zu einer geringeren Anzahl von CG-Iterationen und einer geringeren Rechenzeit. Für das größte Problem ist die Anzahl der erforderlichen CG-Iterationen für die Lösung auf der Grundlage von ACA mehr als doppelt so groß wie für die Lösung auf der Grundlage von BACA. Wir erhalten keine signifikanten Unterschiede in den Speicheranforderungen nach der Anwendung der beiden Algorithmen. Daher resultiert die beobachtete Beschleunigung eher aus der Kopplung des iterativen Lösers und des Fehlerschätzers als aus der Reduzierung der Speicheranforderungen. Zusätzlich zu diesen Resultaten ist eine abnehmende Anzahl der erforderlichen Iterationen k bei einer wachsenden Anzahl von Unbekannten N zu beobachten.

Das Verhältnis der für ACA und für BACA benötigten Zeit ist in Abb. 12.9 ersichtlich. Die Kurven stabilisieren sich bei einer bestimmten Konstante, die von der Geometrie und der Anzahl und Lage der verfeinerten Bereiche abzuhängen scheint.

N	$b_{\min}$	$\varepsilon_{\text{BACA}}$	$\|b - Ax_k\|_2$	e_h	k	Zeit (s)	CG-Iterationen	Speicher (MB)
3 452	15	1e-06	1.39e-06	0.034	10	1.22	134	11.61
8 574	30	1e-06	1.16e-06	0.023	10	4.89	273	42.10
30 642	60	1e-06	1.04e-06	0.012	6	34.80	500	227.07
125 948	120	1e-06	1.21e-06	0.006	6	569.36	1119	1608.32

Tab. 12.8: Numerische Ergebnisse der BACA für die vier in Abbildung 12.8 gezeigten Gitter.

N	$b_{\min}$	e_h	Zeit (s)	CG-Iterationen	Speicher (MB)
3 452	15	0.034	1.54	196	13.67
8 574	30	0.023	6.91	371	48.37
30 642	60	0.012	66.63	909	257.59
125 948	120	0.006	1206.28	2828	1674.62

Tab. 12.9: Numerische Ergebnisse der ACA für die vier in Abbildung 12.8 gezeigten Gitter.

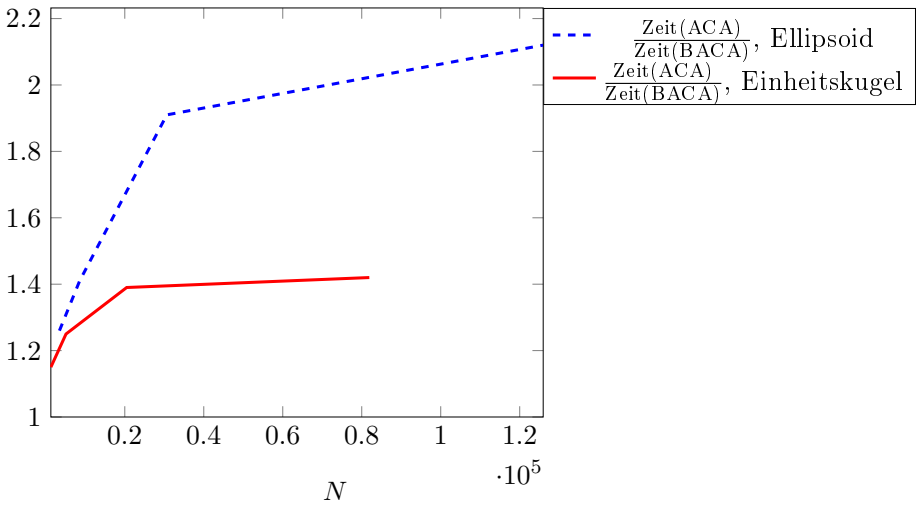

Abb. 12.9: Das Verhältnis der für BACA und ACA benötigten Zeit für zwei Geometrien.

12.4 Elastizitätsgleichungen und die Anwendung von BACA und AMVM

Die folgenden numerischen Experimente sind in zwei Teile unterteilt. In beiden Fällen wird die numerische Lösung der Lamé-Gleichungen

$$-\mu\,\Delta u(x) - (\lambda + \mu)\,\text{grad div}\; u(x) = 0, \quad x \in \Omega, \tag{12.2}$$

mit den Lamé-Konstanten

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

und $E = 1.0$ (N/mm), $\nu = 0.3$ berechnet. Der erste Teil befasst sich mit der Qualität des Fehlerschätzers in AMVM und der numerischen Leistung von AMVM im Vergleich zur Multiplikation mit

einer Approximation aus der ACA. Anschließend wird die numerische Leistung der BACA, welche an die lineare Elastizität angepasst ist, im Vergleich zur ACA untersucht. Jegliche Approximationsschritte in den Verfahren werden dabei ohne Parallelisierung durchgeführt.

12.4.1 Qualität von AMVM für lineare Elastizität

Die qualitativen Untersuchungen der AMVM werden auf drei verschiedenen Diskretisierungen des Einheitswürfels $\Omega = [-1, 1]^3$ durchgeführt, die aus 488, 1946 und 7778 Punkten bestehen, siehe Abbildung 12.10.

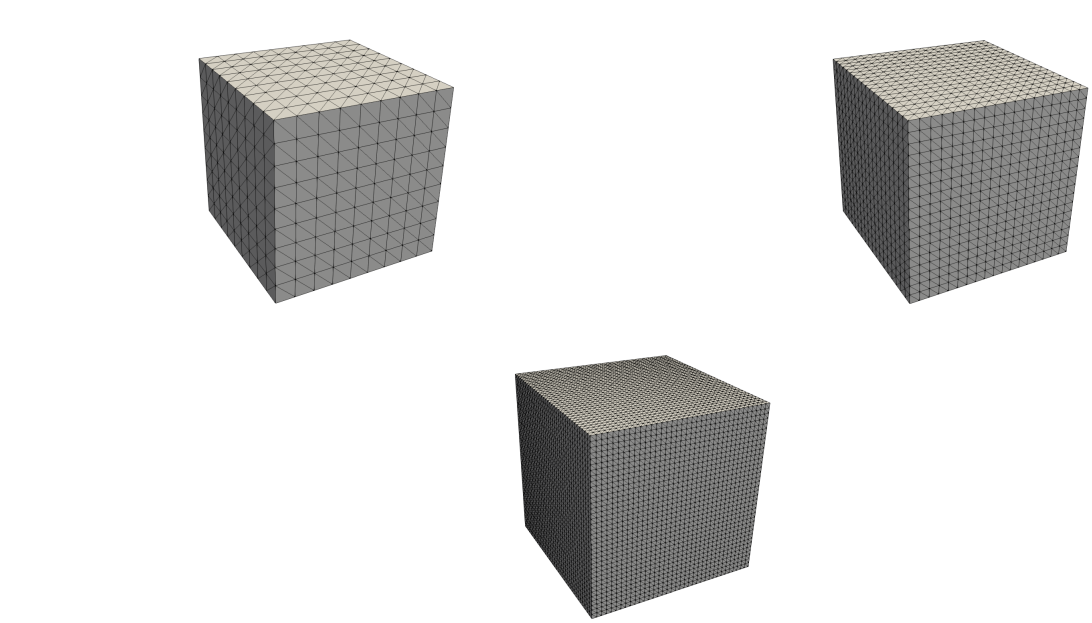

Abb. 12.10: Gitter auf dem betrachteten Würfel Ω (gröbstes Level oben links, feinstes Level unten Mitte).

Die folgenden Randbedingungen werden gewählt

$$\gamma_0 u(x) = g_D(x) := S(x - p) \quad \text{für } x \in \Gamma_D = \{x \in \Omega : x_1 = 1 \text{ oder } x_2 = -1 \text{ oder } x_3 = 1\}$$

und

$$\gamma_1 u(x) = g_N(x) := \frac{\partial}{\partial n} S(x - p)$$

für $x \in \Gamma_N = \{x \in \Omega : x_1 = -1 \text{ oder } x_2 = 1 \text{ oder } x_3 = -1\}$ mit $p = (5.0, 5.0, 5.0)^T$. Wir vergleichen die Berechnungszeit und den Speicherbedarf der beiden Methoden AMVM und ACA bei der Berechnung der rechten Seite von (10.2). Die Approximation der letzteren wird mit b_{AMVM} bzw. b_{ACA} bezeichnet.

Die blockweise Genauigkeit der ACA wird mit $\varepsilon_{\text{ACA}} = 10^{-6}$ und der Zulässigkeitsparameter mit $\beta = 0,8$ gewählt. Die Ergebnisse von ACA sind in den Tabellen 12.10 und 12.11 dargestellt. Sie sind das Resultat einer BEM-Implementierung für die Lamé-Gleichungen, welche als Teil einer Masterarbeit [60] am Lehrstuhl für Wissenschaftliches Rechnen entstanden ist.

N	$b_{\min}$	$\|b - b_{\mathrm{ACA}}\|_2$	Zeit (Approximation)
488	15	3.45e-7	6.9 s
1 946	20	4.43e-7	39.6 s
7 778	30	2.53e-7	248.1 s

Tab. 12.10: Fehler und benötigte Zeit bei der Berechnung der rechten Seite von (10.2) mittels ACA.

	$V_{\Delta,h}$		V_{11}		V_{12}		V_{13}		V_{22}	
N	MB	%	MB	%	MB	%	MB	%	MB	%
488	3.0	83.5	3.5	95.6	3.5	96.5	3.5	96.4	3.4	95.5
1 946	19.7	34.1	23.9	41.4	23.4	40.5	23.3	40.4	23.8	41.3
7 778	115.6	12.5	137.6	14.9	132.7	14.4	131.3	14.2	137.7	14.9

	V_{23}		V_{33}		K_{Δ}	
N	MB	%	MB	%	MB	%
488	3.5	96.1	3.5	95.6	3.6	99.6
1 946	23.3	40.4	23.8	41.3	31.1	53.9
7 778	131.4	14.2	136.2	14.8	201.1	21.8

Tab. 12.11: Speicheranforderungen für die durch die ACA konstruierten Approximationen.

Die Anwendung der adaptiven Matrix-Vektor Multiplikation (AMVM) auf das zu Beginn des Abschnitts beschriebene lineare Elastizitätsproblem liefert für $\theta = 0.7$ die in den Tabellen 12.12 und 12.13 dargestellten Ergebnisse. Der Fehler $\|b - b_{\mathrm{AMVM}}\|_2$ wurde in der gleichen Größenordnung gehalten wie $\|b - b_{\mathrm{ACA}}\|_2$ in den vorherigen Tests. Auf allen drei Diskretisierungen des Würfels Ω konnte eine Reduzierung der Rechenzeit erreicht werden. Der Speicherbedarf der Operatoren $V_{\Delta,h}$, V_{11}, V_{12}, V_{13}, V_{22}, V_{23} und V_{33} erweist sich als geringfügig niedriger als die entsprechenden Approximationen mittels ACA. Der größte Vorteil ergibt sich für den Operator $K_{\Delta,h}$.

N	$b_{\min}$	$\|b - b_{\mathrm{AMVM}}\|_2$	Zeit (Approximation)
488	15	6.34e-7	5.2 s
1 946	20	6.85e-7	29.5 s
7 778	30	4.63e-7	188.8 s

Tab. 12.12: Fehler und benötigte Zeit bei der Berechnung der rechten Seite von (10.2) mittels AMVM.

	$V_{\Delta,h}$		V_{11}		V_{12}		V_{13}		V_{22}	
N	MB	%	MB	%	MB	%	MB	%	MB	%
488	2.9	79.1	3.0	84.2	3.0	84.2	3.0	84.4	3.0	84.2
1 946	19.3	33.5	21.3	37.0	20.9	36.2	20.8	36.1	21.3	36.9
7 778	113.7	12.3	120.3	13.0	116.6	12.6	115.5	12.5	120.6	13.1

	V_{23}		V_{33}		K_Δ	
N	MB	%	MB	%	MB	%
488	3.0	84.3	3.0	84.2	2.2	60.8
1 946	20.8	36.1	21.3	36.9	17.2	29.9
7 778	115.7	12.5	119.4	14.8	145.4	15.8

Tab. 12.13: Speicheranforderungen für die durch die AMVM konstruierten Approximationen.

Bevor wir uns einem realistischeren Problem zuwenden, werfen wir einen genaueren Blick auf die Zuverlässigkeit und Effizienz des Fehlerschätzers. Wir stellen die Ergebnisse vor, die im Falle einer Diskretisierung mit 488 Punkten entstehen. Der iterative Approximationsprozess wird mit einer Rang-2-Approximation gestartet.

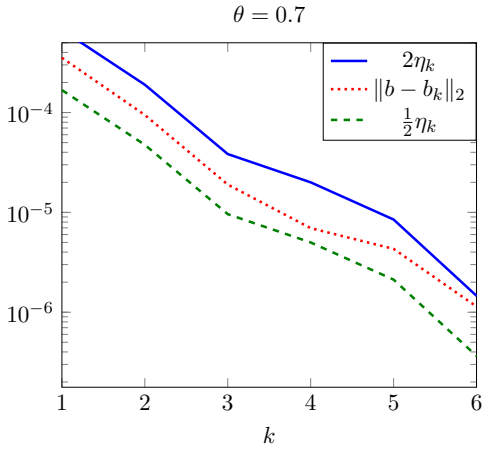

Abb. 12.11: Qualität des Fehlerschätzers $\mathcal{E}_k$ im Fall der AMVM.

Abbildung 12.11 zeigt, dass der Fehlerschätzer $\mathcal{E}_k$ den Fehler $\|b - b_k\|_2$ der rechten Seite zuverlässig und effizient schätzt, was die die theoretischen Ergebnisse der adaptiven Matrix-Vektor Multiplikation, die in Abschnitt 10 vorgestellt wurden, bestätigt.

12.4.2 Belastung eines Doppel-T Trägers in z-Richtung

Die folgenden Experimente konzentrieren sich auf die numerische Lösung der Lamé-Gleichungen für drei Diskretisierungen der in Abb. 12.12 dargestellten Geometrie. Der Balken hat eine Länge, Höhe und Breite von 2 mit einem zentralen Teil mit einer Höhe und Breite von 1. Abbildung 12.13 zeigt die Zuordnung der Randelemente zu Dirichlet- und Neumann-Komponenten. Die blaue Fläche des Balkens mit einer Kraft von 0,1 (N) belastet, wobei der Dirichlet-Rand durch die grüne Fläche dargestellt ist. Auf dem restlichen Teil des Randes, d. h. auf dem grauen Bereich in Abb. 12.13, werden homogene Neumann-Randbedingungen ($\gamma_1^{\text{int}} u(x) = 0$) vorgeschrieben. Die rechte Seite des

zu berechnenden Gleichungssystems erhält man durch Multiplikation der gegebenen Randdaten mit den jeweiligen diskretisierten Operatoren V_h, K_h und D_h; vgl. (10.2).

Wir vergleichen die Näherungslösung, die sich aus der Approximation der Koeffizientenmatrix mittels BACA bzw. ACA ergibt.

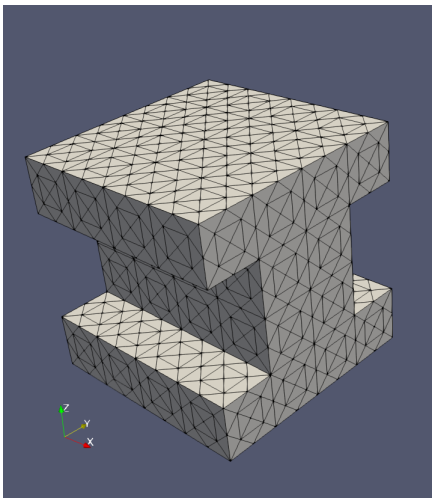

Abb. 12.12: Diskretisierung der Oberfläche eines Doppel-T Trägers in Dreiecke.

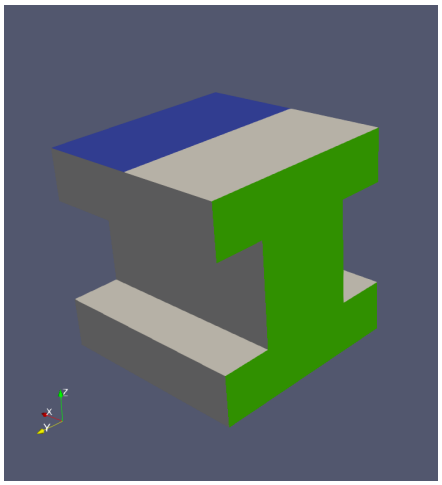

Abb. 12.13: Dirichlet-Rand in grün und belasteter Neumann Teilrand in blau dargestellt.

Die Verformungen des Trägers unter Belastung in z-Richtung sind in Abbildung 12.14 dargestellt. Die maximalen absoluten Differenzen zwischen den mit der ACA und der BACA erzeugten Verformungen in x-, y- und z-Richtung betragen 1.2e-04, 2.4e-04 und 3.3e-04. Die beiden Methoden ACA und BACA liefern demnach vergleichbare Ergebnisse bei den Verformungen.

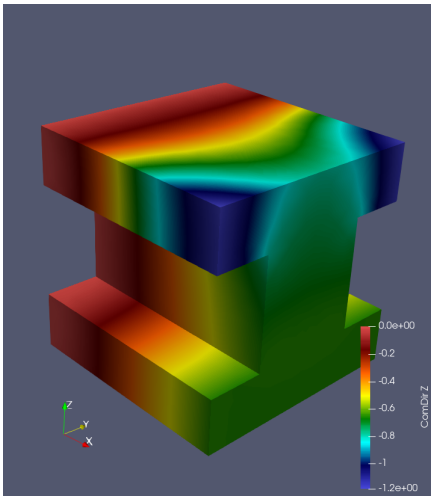

Abb. 12.14: Verschiebung unter der Belastung in z-Richtung nach Anwendung des ACA auf Gitter 1.

Die für die ACA in Abschnitt 11.4 verwendeten Parameter bleiben unverändert. Zusätzlich verwenden wir $\varepsilon_{\text{BPCG}} = 10^{-5}$ während der iterativen Lösung mittels der konjugierten Gradienten Methode nach Bramble-Pasciak [28]. Die Ergebnisse für ACA sind in Tabelle 12.14 dargestellt.

		$V_{\Delta,h}$		V_{11}		V_{12}		V_{13}		V_{22}
N	M	MB	%	MB	%	MB	%	MB	%	MB
1 664	834	6.9	65.2	8.3	78.3	8.4	79.8	8.3	78.7	8.4
6 656	3 330	44.4	26.2	56.3	33.3	56.5	33.4	54.6	32.3	56.3
26 624	13 314	238.7	8.8	306.5	11.3	300.8	11.1	285.3	10.6	305.1

		V_{22}	V_{23}		V_{33}		K_Δ		Zeit gesamt
N	M	%	MB	%	MB	%	MB	%	
1 664	834	79.7	8.4	79.4	8.2	77.2	9.5	90.1	40.3 s
6 656	3 330	33.3	54.6	32.3	54.0	31.9	75.2	44.9	426.7 s
26 624	13 314	11.3	283.7	10.5	287.2	10.6	497.3	18.4	3 714.7 s

Tab. 12.14: Speicherbedarf der mit der ACA konstruierten Approximationen und Zeitaufwand für die Lösung des Problems.

Für BACA müssen andere Parameter gewählt werden. Die adaptive Anpassung der Fehlertoleranz im Bramble-Pasciak CG erfolgt gemäß der Bedingung (11.2) mit $\alpha = 10$. Der Anfangswert der Genauigkeit beim Bramble-Pasciak-CG ist 10^{-1}. Die Genauigkeit $\varepsilon_{\text{BACA}}$ beträgt 10^{-4} und $\theta = 0.8$. Die Anfangsapproximationen der jeweiligen V-Operatoren erhält man durch Anwendung von 8 (für die beiden gröberen Gitter) und 10 (für das feinste Gitter) ACA-Schritten. Für den Operator K beträgt die entsprechende Anzahl an Schritten 4 und 6. Die Lösung der Lamé-Gleichungen mit Hilfe der BACA mit diesen Parametern führt zu den in Tabelle 12.15 angegebenen Werten.

		$V_{\Delta,h}$		V_{11}		V_{12}		V_{13}		V_{22}
N	M	MB	%	MB	%	MB	%	MB	%	MB
1 664	834	6.5	61.0	6.8	64.2	6.8	64.4	6.8	63.9	6.8
6 656	3 330	40.1	23.7	41.5	24.5	41.2	24.4	40.2	23.8	41.5
26 624	13 314	220.3	8.1	228.9	8.5	223.8	8.3	213.2	7.9	228.8

		V_{22}	V_{23}		V_{33}		K_Δ		Zeit gesamt
N	M	%	MB	%	MB	%	MB	%	
1 664	834	64.5	6.8	64.0	6.8	63.9	4.7	44.4	30.5 s
6 656	3 330	24.6	40.2	23.8	40.5	24.0	27.3	16.1	215.8 s
26 624	13 314	8.5	212.7	7.9	218.3	8.1	180.4	6.7	2 070.4 s

Tab. 12.15: Speicher, relativer Speicher für die mit der BACA konstruierten Approximationen und Zeitaufwand für die Lösung des Problems nach Anwendung der BACA im Fall der Lamé-Gleichungen.

Im Vergleich zu den mit der ACA erzielten Ergebnissen lassen sich bei der Anwendung der BACA auf den Operator $V_{\Delta,h}$ kaum signifikante Unterschiede feststellen. Die V-Operatoren benötigen nur etwa 70-80% des Speicherplatzes, der für der mit der ACA erzeugten Approximationen. Stärkere Vorteile können für den K_Δ-Operator erzielt werden. Hier benötigt die Approximation mit BACA nur 50% (für das gröbste Gitter) und 36% (für die beiden feinsten Gitter) des Speichers, der im Fall der ACA benötigt wird. Tabelle 12.15 zeigt auch die Vorteile der BACA in Bezug auf die Berechnungszeit. Während auf dem gröbsten Gitter 75% der von der ACA benötigten Zeit verbraucht wird, kann die

Zeit für das zweitfeinste Gitter auf 51% und für das feinste betrachtete Gitter auf 56% reduziert werden.

13. Abschließende Bemerkungen

Das Ziel dieser Arbeit war die Untersuchung und Erweiterung der adaptiven Kreuzapproximation (ACA) bezüglich der Anwendung auf diskretisierte nicht-lokale Operatoren, welche in der Regel voll besetzte Matrizen darstellen. In Verbindung mit der Lösung eines linearen Gleichungssystems, was z.B. bei der Benutzung der Randelementmethode auftritt, ergibt sich hier ein hoher Kostenaufwand. Die ACA stellt eine Methode dar, um diesen Aufwand zu reduzieren, jedoch bezieht sie keine weiteren problemspezifischen Informationen in die Approximation mit ein, d.h. ACA approximiert den diskreten Operator universell ohne z.B. die rechte Seite mit in Betracht zu ziehen. Ein ähnliches Problem tritt nicht nur bei der Lösung eines linearen Gleichungssystems auf, sondern auch bei der Matrix-Vektor-Multiplikation, falls die Matrix aufgrund der Diskretisierung eines nicht-lokalen Operators entstanden ist. Des Weiteren kann es vorkommen, dass die Approximation des Doppelschichtoperators innerhalb der Randintegralmethode, welcher ein essentieller Bestandteil der Randintegralformulierung bei partiellen Differentialgleichung ist, bei bestimmten Gebietsdiskretisierungen nicht konvergiert (Schachmusterproblem), siehe Kapitel 12. Im Hinblick auf die eben dargestellten Probleme konnten in dieser Arbeit die folgenden drei Lösungsansätze entwickelt und analysiert werden:

1. Lösung des Schachmusterproblems
   Der Ansatz bei dieser Problemstellung ist es, das der adaptiven Kreuzapproximation zugrunde liegende Interpolationsproblem in einer anderen Funktionsbasis zu betrachten. Anders als bei der zuvor genutzten Polynominterpolation, kann bei Singularitätenfunktionen wie sie z.B. beim Laplace-Operator auftreten, die Interpolation mittels radialer Basisfunktionen genutzt werden. Vorteilhaft hierbei ist, dass durch die positive Definitheit von radialen Basisfunktion die Lösbarkeit des Interpolationsproblems nicht mehr durch Pivotisierung garantiert werden muss. Als Konsequenz daraus kann die Fülldichte, welche ein grundlegender Bestandteil der Fehleranalyse der Interpolation mit radialen Basisfunktionen darstellt, als Kriterium bei der adaptiven Auswahl der nächsten Punkte genutzt werden. Aufgrund dieser gebietsabdeckenden Punktauswahl kann der Fall, dass die ACA nur noch Punkte aus einem bestimmten Teil des betrachteten Gebiets nutzt und somit nicht konvergiert, nicht mehr eintreten.

2. Adaptive Matrix-Vektor-Multiplikation
   Dieses neue Verfahren zielt auf die simultane Approximation einer Matrix und der Berechnung eines Matrix-Vektor-Produkts ab. Unter Benutzung eines block-zeilen basierten Fehlerschätzers und spezieller Verfeinerungs- und Blockauswahlstrategien können diejenigen Blöcke detektiert werden, welche für die betrachtete Matrix-Vektor Multiplikation die meisten Informationen liefern. Hierbei konnte die Zuverlässigkeit und Effizient des neu entwickelten Fehlerschätzers theoretisch analysiert und dessen Qualität in numerischen Beispielen beobachtet werden. Des Weiteren zeigten die numerischen Tests eine signifikante Reduktion des Speicherbedarfs der Matrix sowie der Berechnungszeit.

3. Block-adaptive Kreuzapproximation
   Ähnlich zur adaptiven Matrix-Vektor Multiplikation werden auch hier zwei Lösungsschritte miteinander kombiniert. Im Fokus stehen die simultane Approximation einer Matrix und der Lösung des dazugehörigen Gleichungssystems. Mit den neu eingeführten block-basierten Fehlerschätzern und zusätzlichen Blockauswahlverfahren können diejenigen Blöcke gefunden werden, welche bei genauerer Approximation eine genauere Lösung liefern. In der Theorie konnte die Zuverlässigkeit, welche essentiell für die Konvergenz des gesamten Verfahrens ist, bewiesen werden. Für die Effizienz konnte jedoch nur eine untere Schranke angegeben werden. Numerische

Beispiele zeigen die Qualität des entwickelten Fehlerschätzers. Je nach Problemstellung haben sich sehr deutliche Speicher- und Rechenzeitreduktionen ergeben.

Die einzelnen Punkte geben Raum für weitere Untersuchungen. Im Bezug auf 1. ist die Stabilität der adaptiven Kreuzapproximation noch nicht geklärt. Bei den anderen beiden Punkten wurde bislang nur die Adaptivität in den Matrixblöcken in Betracht gezogen, ohne das zugrunde liegende Gebiet weiter zu verfeinern. Da sich die einzelnen Blöcke der Matrix direkt im Gebiet widerspiegeln bzw. umgekehrt, würde sich eine derartige Vorgehensweise anbieten. Aus Sicht der hierarchischen Matrixapproximation hätte eine zusätzliche adaptive Gebietsverfeinerung keinen allzu großen Mehraufwand, da in der Matrixapproximation lediglich Blockzeilen und Blockspalten angepasst werden müssten und nicht die komplette Matrix neu aufgestellt werden muss. Die Analyse der verwendeten Fehlerschätzer hingegen wird sich aufgrund der zusätzlichen Abhängigkeit des Gitters bzw. der Gitterweite schwieriger gestalten. Da schon im aktuellen Fall die Effizienz des Fehlerschätzers im Allgemeinen nicht komplett gezeigt werden konnte, stellt sich natürlich auch die Frage, ob das bei einer adaptiven Gebietsverfeinerung überhaupt möglich sein wird.

Abgesehen von den in dieser Arbeit betrachteten Gleichungen wird es interessant sein, zu sehen, wie sich die neuen Methoden bei anderen Problemstellungen wie z.B. der Helmhotz-Gleichung oder den Stokes-Gleichungen verhalten werden. Zumindest im Fall der Stokes-Gleichungen wird ähnlich zu den hier behandelten Lamé-Gleichungen eine kleine Anpassung in der Verfeinerungsstrategie nötig sein. Aufgrund der Darstellung von fraktionellen Diffusionsproblemen als singuläre bzw. schwach singuläre Oberflächenintegrale ist auch eine Anwendung in diesem Bereich möglich. Da jeder Matrixeintrag eines diskretisierten fraktionellen Laplace-Operators sehr teuer ist, sollten die neuen Methoden eine enorme Kosten- bzw. Zeitersparnis liefern.

Anhang A: Daten zu den Grafiken

A.1 Daten zu Abbildung 12.3

k	γ_k	$\|b - b_k\|_2$
1	8.46e-06	1.88e-05
2	7.86e-06	1.61e-05
3	6.99e-06	1.41e-05
4	6.060e-06	1.09e-05
5	4.84e-06	8.20e-06
6	3.96e-06	6.74e-06
7	3.19e-06	5.44e-06
8	2.74e-06	4.45e-06
9	2.29e-06	3.55e-06
10	1.82e-06	2.38e-06
11	1.41e-06	1.82e-06
12	1.11e-06	1.53e-06
13	9.41e-07	1.19e-06

Tab. A.1: Qualität des Fehlerschätzers γ_k.

A.3 Daten zu Abbildung 12.9

N	$\frac{\text{Zeit ACA}}{\text{Zeit BACA}}$
3 452	1.26
8 574	1.41
30 642	1.91
125 948	2.12

Tab. A.2: Das Verhätnis der für BACA und ACA benötigten Zeit im Fall des Ellipsoids.

N	$\frac{\text{Zeit ACA}}{\text{Zeit BACA}}$
1 280	1.15
5 120	1.25
20 480	1.39
81 920	1.42

Tab. A.3: Das Verhätnis der für BACA und ACA benötigten Zeit im Fall der Einheitskugel.

A.2 Daten zu Abbildung 12.11

k	γ_k	$\|b - b_k\|_2$
1	3.36e-04	3.54e-04
2	9.50e-05	9.46e-05
3	1.92e-05	1.90e-05
4	1.00e-05	6.95e-06
5	4.24e-06	4.31e-06
6	7.30e-07	1.14e-06

Tab. A.4: Qualität des Fehlerschätzers $\mathcal{E}_k$.

Literaturverzeichnis

[1] R.A. Adams, Sobolev Spaces, Academic Press, New York, 1975.

[2] M. Ainsworth, C. Glusa, Aspects of an adaptive finite element method for the fractional Laplacian: a priori error estimates, efficient implementation an multigrid solver, Comput. Methods Appl. Mech. Eng. 327 (2017), pp. 4-35.

[3] N. Aronszajn, Theory of reproducing kernels, Trans. Am. Math. Soc. (68), pp. 337 - 404, 1950.

[4] S. Arya and D. M. Mount, Approximate nearest neighbor searching, Proc. 4th Ann. ACM-SIAM Symposium on Discrete Algorithms, pp. 271–280, New York, ACM Press, 1993.

[5] S. Arya and D. M. Mount, Approximate range searching, Proc. 11th Annual ACM Symp. on Computational Geometry, pp. 172–181, New York, ACM Press, 1995.

[6] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, An optimal algorithm for approximate nearest neighbor searching, J. ACM, 45: 891–923, 1998.

[7] M. Aurada, M. Feischl, T. Führer, M. Karkulik, D. Praetorius, Efficiency and optimality of some weighted-residual error estimator for adaptive 2D boundary element methods, Comput. Methods Appl. Math. 13(3), 2013, pp. 305 - 332.

[8] A. Aurada, S. Ferraz-Leite, D. Praetorius, Estimator reduction and convergence of adaptive BEM, Applied Numerical Mathematics 62 (2012), pp. 787-801.

[9] H. Bateman and A. Erdélyi, Tables of integral transforms, Volume 2, Bateman Manuscript Project, McGraw-Hill, New York, USA, 1954.

[10] M. Bauer, M. Bebendorf und B. Feist, Kernel-independent adaptive construction of $\mathcal{H}^2$-matrix approximations, Numer. Math., 150:1–32, 2022.

[11] M. Bebendorf, Approximation of boundary element matrices, Numer. Math. 86 (2000), pp. 565 - 589.

[12] M. Bebendorf, Efficient inversion of Galerkin matrices of general second-order elliptic differential operators with nonsmooth coefficients, Math. Comp. 74 (2005), pp. 1179 - 1199.

[13] M. Bebendorf, Hierarchical LU decomposition based preconditioners for BEM, Computing 74 (2005), pp. 225 - 247.

[14] M. Bebendorf, Approximate inverse preconditioning of finite element discretizations of elliptic operators with nonsmooth coefficients, SIAM J. Anal. Appl. 27 (2006), pp. 909 - 929.

[15] M. Bebendorf, Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems, Volume 63 of Lecture Notes in Computational Science and Engineering (LNCSE), Springer, Berlin(2008).

[16] M. Bebendorf, M. Bollhöfer, and M. Bratsch, On the spectral equivalence of hierarchical matrix preconditioners for elliptic problems, Math. Comp. 85(302), pp.2839–2861, 2016.

[17] M. Bebendorf, R. Grzhibovskis, Accelerating Galerkin BEM for linear elasticity using adaptive cross approximation, Mathematical Methods in the Applied Sciences 29 (2006), pp. 1721 - 1747.

[18] M. Bebendorf and W. Hackbusch, Stabilised rounded addition of hierarchical matrices, Num. Lin. Alg. Appl., 14(5):407–423, 2007.

[19] M. Bebendorf, R. Kriemann, Fast parallel solution of boundary integral equations and related problems, Computing and Visualization 8(3-4), 2005, pp. 121 - 135.

[20] M. Bebendorf, C. Kuske, and R. Venn, Wideband nested cross approximation for Helmholtz problems, Numer. Math., 130:1–34, 2015.

[21] M. Bebendorf, S. Rjasanow, Adaptive low-rank approximation of collocation matrices, Computing 70 (2003), pp. 1 - 24.

[22] S. Börm, Efficient Numerical Methods for non-local operators, Tracts in Mathematics 14. EMS, 2010.

[23] S. Börm and L. Grasedyck, Hybrid cross approximation of integral operators, Numer. Math. 205, pp. 221–249, 2005.

[24] S. Börm, W. Hackbusch, $\mathcal{H}^2$-matrix approximation of integral operators by interpolation, Applied Numerical Mathematics, 43:139-143, 2002.

[25] S. Börm, M. Löhndorf, and J. M. Melenk, Approximation of integral operators by variable-order interpolation, Numer. Math., 99(4):605–643, 2005.

[26] D. Braess and W. Hackbusch. On the efficient computation of high-dimensional integrals and the approximation by exponential sums, In Ronald A. DeVore and Angela Kunoth, eds., Multiscale, nonlinear and adaptive approximation, pages 39–74. Springer, Berlin, 2009.

[27] H. Brakhage, P. Werner, Über das Dirichletsche Außenraumproblem für die Helmholtzsche Schwingungsgleichung, Arch. Math. 16 (1965), pp. 325-329.

[28] J. H. Bramble, J. E. Pasciak, A Preconditioning Technique for Indefinite Systems Resulting from Mixed Approximations of Ellipitic Problems, Mathematics of Computation 50.181 (1988), pp. 1-17.

[29] S.C. Brenner, L.R. Scott, The Mathematical Theory of Finite Element Methods, 3rd Edition, Texts in Applied Mathematics 15, Springer, New York, 2010.

[30] M.D. Buhmann, Radial basis functions, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2003.

[31] H. Cheng, L. Greengard, and V. Rokhlin, A fast adaptive multipole algorithm in three dimensions, J. Comput. Phys., 155(2):468–498, 1999.

[32] E. Di Nezza, G. Palatucci, E. Valdinoci, Hitchhiker's guide to the fractional Sobolev spaces, Bull. Sci. math. 136 (2012), pp. 521-573.

[33] M. D'Elia, M. Gunzburger, The fractional Laplacian operator on bounded domains as a special case of the nonlocal diffusion operator, Comput. Math. Appl. 66(7), 2013, pp. 1245-1260.

[34] W. Dörfler, A convergent adaptive algorithm for Poisson's equation, SIAM J. Numer. Anal. 33 (1996), pp. 1106-1124.

[35] G. Eckart, G. Young, The approximation of one matrix by another of lower rank, Psychometrica (1), pp. 211 - 218, 1936.

[36] M. Faustmann, J. M. Melenk, and D. Praetorius, Existence of $\mathcal{H}$-matrix approximants to the inverse of BEM matrices: The simple-layer operator, Math. Comp., 85(297): 119-152, 2016.

[37] M. Faustmann, J. M. Melenk, and D. Praetorius, Existence of $\mathcal{H}$-matrix approximants to the inverse of BEM matrices: The hyper-singular integral operator, IMA J. Numer. Anal., 37(3): 1211-1244, 2017.

[38] S. Ferraz-Leite, D. Praetorius, Simple a posteriori error estimators for the h-version of the boundary element method, Computing 83 (2008), pp. 135 - 162.

[39] T. Gantumur, Adaptive boundary element methods with convergence rates, Numer. Math. 124(3), 2013, pp. 471 - 516.

[40] L. Grasedyck, W. Hackbusch, Constructions and arithmetics of $\mathcal{H}$-matrices, Computing 70 (2003), pp. 295 - 334.

[41] L.F. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73(2), 1987, pp. 325 - 348.

[42] H. Han, The boundary integro-differential equations of three dimensional Neumann problem in linear elasticity, Numer. Math. 68, 1994, pp. 269-281.

[43] W. Hackbusch, Hierarchical Matrices: Algorithms and Analysis, Springer Series in Computational Mathematics 49, Springer, Berlin, 2015.

[44] W. Hackbusch, A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices, Computing 62 (1999), pp. 89-108.

[45] W. Hackbusch, B.N. Khoromskij, A sparse $\mathcal{H}$-matrix arithmetic. Part II: Application to multi - dimensional problems, Computing 64 (2000), pp. 21-47.

[46] W. Hackbusch, B.N. Khoromskij, A sparse $\mathcal{H}$-matrix arithmetic: general complexity estimates, J. Comput. Appl. Math. 125(1-2), 2000, pp. 479-501. Numerical analysis 2000, Vol. VI, Ordinary differential equations and integral equations.

[47] W. Hackbusch, B. N. Khoromskij, and S. A. Sauter, On $\mathcal{H}^2$-matrices, In H.-J. Bungartz, R. H. W. Hoppe, and Ch. Zenger, eds., Lectures on Applied Mathematics, pages 9–29. Springer-Verlag, Berlin, 2000.

[48] W. Hackbusch, Z.P. Nowak, On the fast matrix multiplication in the boundary element method by panel clustering, Numer. Math. 54(4), 1989, pp. 463-491.

[49] R. B. Hetnarski, J. Ignaczak, Mathematical Theory of Elasticity, Taylor & Francis, 2004.

[50] M. Karkulik, G. Of, D. Praetorius, Convergence of adaptive 3D BEM for weakly singular integral equations based on isotropic mesh-refinement, Numerical Methods for Partial Differential Equations 29 (2013), pp. 2081-2106.

[51] R. Kress, Linear Integral Equations, Third Edition, Applied Mathmatical Sciences, Springer, New York, 2014.

[52] N. S. Landkof, Foundations of Modern Potential Theory, Springer, Berlin, Heidelberg, 1972.

[53] W.R. Madych, S.A. Nelson, Multivariate Interpolation and conditionally positive definite functions, Approximation Theory Appl. (4), pp. 77 - 89, 1988.

[54] W.R. Madych, S.A. Nelson, Multivariate Interpolation and conditionally positive definite functions II, Math. Comput. (54), pp. 211 - 230, 1990.

[55] W.R. Madych, S.A. Nelson, Bounds on multivariate polynomials and exponential error estimates for multiquadric interpolation, J. Approx. Theory (70), pp. 94 - 114, 1992.

[56] W. McLean, Strongly Elliptic Systems and Boundary Integral Equations, Cambridge University Press, New York, 2000.

[57] W. McLean, O. Steinbach, Boundary element preconditioners for a hypersingular integral equation on an interval, Advances in Computational Mathematics (1999), pp. 271–286.

[58] K. Pearson, On lines and planes of closest fit to systems of points in space, Philosophical Magazine, 2(6), pp. 559 - 572, 1901.

[59] A. Quarteroni, R. Sacco, F. Saleri, Numerical Mathematics, second edition, Texts in Applied Mathematics, Springer, Berlin, Heidelberg, 2007.

[60] T. Rau, Schnelle Methoden zur Approximation der Elastizitäts-Gleichungen, Masterarbeit, Lehrstuhl für Wissenschaftliches Rechnen, Universität Bayreuth, 2019.

[61] V. C. Raykar, C. Yang, R. Duraiswami, N. Gumerov, Fast computation of sums of Gaussians in high dimension, Technical report, Department of Computer Science and Institute for Advanced Computer Studies, University if Maryland, College Park, 2005.

[62] S. Rjasanow, O. Steinbach, The Fast Solution of Boundary Integral Equations, Mathematical and Analytical Techniques with Applications to Engineering, Springer, New York, 2007.

[63] V. Rokhlin, Rapid solution of integral equations of classical potential theory, J. Comput. Phys., 60(2), pp. 187 - 207, 1985.

[64] Y. Saad, Iterative methods for sparse linear systems, Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2003.

[65] S. A. Sauter, C. Schwab, Boundary Element Methods, Springer Series in Computational Mathematics, Springer, Berlin, 2011.

[66] R. Schabak, Native Hilbert spaces for radial basis functions I, In M.W. Müller et al., eds., New Developments in Approximation Theory, 2nd International Dortmund Meeting (IDoMat '98), Germany, February 23-27, 1998, volume 132 of Int. Ser. Numer. Math., pp. 255-282, Basel, Birkhäuser Verlag, 1999.

[67] R. Schabak, A unified theory of radial basis functions: native Hilbert spaces for radial basis functions II, J. Comput. Appl. Math. (121), pp. 165-177, 2000.

[68] H. R. Schwarz, N. Köckler, Numerische Mathematik, 8. Auflage, Vieweg+Teubner, Wiesbaden, 2011.

[69] J. H. Spurk, N. Aksel, Strömungslehre, Einführung in die Theorie der Strömungen, 8. Auflage, Springer, Heidelberg, 2010.

[70] O. Steinbach, Numerical Approximation Methods for Elliptic Boundary Value Problems, Springer, New York, 2008.

[71] P. P. Teodrescu, Treatise on Classical Elasticity. Theory and Related Problems, Springer, 2013.

[72] A. Toselli, O. B. Widlund, Domain Decomposition Methods - Algorithms and Theory, Springer, Berlin, Heidelberg, 2005.

[73] E.E. Tyrtyshnikov, Mosaic-skeleton approximations, Calcolo 33(1-2), pp. 47 - 57. Toeplitz matrices: structures algorithms and applications (Cortona, 1996).

[74] H. Wendland. Scattered Data Approximation. Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, UK, 2005.

[75] J. Wloka, Funktionalanalysis und Anwendungen, Walter de Gruyter, Berlin, 1971.

[76] K. Yosida, Functional Analysis, Springer, Berlin Heidelberg, 1980.

Tabellenverzeichnis

Abbildungsverzeichnis

Publikationen

1. M. Bauer, M. Bebendorf, Block-Adaptive Cross Approximation of Discrete Integral Operators, Computational Methods in Applied Mathematics 21(1), 2021, pp. 13-29.

2. M. Bauer, M. Bebendorf, Adaptive $\mathcal{H}$-Matrix Computations in Linear Elasticity, Preprint.

3. M. Bauer, M. Bebendorf und B. Feist, Kernel-independent adaptive construction of $\mathcal{H}^2$-matrix approximations, Numer. Math., 150:1–32, 2022.

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die von mir angegebenen Quellen und Hilfsmittel verwendet habe.

Weiterhin erkläre ich, dass die Hilfe von gewerblichen Promotionsberatern bzw. Promotionsvermittlern oder ähnlichen Dienstleistern weder bisher in Anspruch genommen habe, noch künftig in Anspruch nehmen werde.

Zusätzlich erkläre ich hiermit, dass ich keinerlei frühere Promotionsversuche unternommen habe.

Ort, Datum	Maximilian Bauer