

## Zusammenfassung der Arbeit

In vielen Wissenschaften führt die Modellbildung auf partielle Differentialgleichungen. So werden vor allem in der Physik (Schwingungen, Wärmeleitung, Wellenausbreitung, Diffusion, Schrödinger-Gleichung), aber auch in vielen anderen Disziplinen wie der Ökonomie (Black-Scholes-Modell zur Bewertung von Finanzoptionen), der Biologie (Populationsdynamik) und der Medizin (Zellwachstum) die Lösungen partieller Differentialgleichungen benötigt. Durch eine schnelle Lösung von partiellen Differentialgleichungen bietet sich die Möglichkeit der Simulation und Optimierung der modellierten Prozesse am Rechner. Dabei lassen sich optimale Lösungen am Computermodell einfacher finden als in einem Versuch mit Prototypen.

Im Gegensatz zu gewöhnlichen Differentialgleichungen existiert selbst für lineare partielle Differentialgleichungen mit konstanten Koeffizienten keine allgemeine Lösungstheorie. Analytische Lösungen partieller Differentialgleichungen sind nur auf einfachen Gebieten bekannt. Daher muss die Lösung numerisch bestimmt werden.

Nichtlineare partielle Differentialgleichungen können mit Hilfe des Newton-Verfahrens gelöst werden. Dazu muss in jedem Schritt des Newton-Verfahrens die Lösung einer linearen partiellen Differentialgleichung berechnet werden. Deshalb beschränken wir uns auf die Behandlung linearer partieller Differentialgleichungen.

Nach einem Überblick über die verschiedenen partiellen Differentialgleichungstypen wird die Finite-Elemente-Methode zur numerische Lösung elliptischer partieller Differentialgleichungen kurz erläutert. Zur praktischen Berechnung von Lösungen werden elliptische partielle Differentialgleichungen diskretisiert. Die dabei entstehenden linearen Gleichungssysteme sind umso größer, je feiner diskretisiert wird, besitzen pro Zeile und Spalte nur wenige von Null verschiedene Einträge und sind schlecht konditioniert. In der Praxis werden die linearen Gleichungssysteme mit direkten Verfahren, Mehrgitter-Verfahren oder iterativen Methoden gelöst. Daneben befasst sich die Arbeit mit Vorkonditionierern, weil sie die von der Kondition der Matrix abhängige Konvergenz iterativer Verfahren verbessern können. Die Vorkonditionierung dient nicht nur zur Steigerung der Effizienz iterativer Löser, sondern auch zur Verbesserung der Robustheit.

Da viele der in der Arbeit auftretenden algorithmischen Probleme auf Graphen zurückgeführt werden können, behandelt das zweite Kapitel die benötigten graphentheoretischen Begriffe und führt zwei Algorithmen zur Bestimmung von Abständen in Graphen ein. Dies sind die Breitensuche zur Bestimmung von Abständen in ungewichteten Graphen sowie der Algorithmus von Dijkstra zur Bestimmung von Abständen in gewichteten Graphen.

Aufgrund der Größe der linearen Gleichungssysteme werden zur Lösung immer häufiger iterative Verfahren in Verbindung mit Vorkonditionierern eingesetzt. Mit der Technik der hierarchischen Matrizen ( $\mathcal{H}$ -Matrizen) lassen sich effiziente Vorkonditionierer konstruieren, die eine schnelle Konvergenz iterativer Verfahren sichern.  $\mathcal{H}$ -Matrizen sind zur Vorkonditionierung besonders geeignet, da sich der Vorkonditionierer in linear-logarithmischer Komplexität erstellen und anwenden lässt. Kapitel 4 ist der Einführung hierarchischer Matrizen gewidmet. Diese basieren auf einer Partition der Matrix in Blöcke und der Niedrigrang-Approximation der Blöcke. Da nicht für jeden Block einer (Finite-Elemente-) Matrix  $A \in \mathbb{R}^{I \times I}$  ( $I$  eine Indexmenge) eine effiziente Niedrigrang-Approximation berechnet werden kann, ist es für die Laufzeit der arithmetischen Operationen hierarchischer Matrizen besonders wichtig, eine günstige Partition zu generieren. Ob ein Block zur Niedrigrang-Approximation

geeignet ist, wird durch die Auswertung einer anwendungsabhängigen Bedingung bestimmt. Für Finite-Elemente-Probleme werden häufig die Bedingung

$$\min\{\text{diam } X_{t_1}, \text{diam } X_{t_2}\} \leq \eta \text{dist}(X_{t_1}, X_{t_2}). \quad (1)$$

oder ähnliche Bedingungen eingesetzt. Dabei sind  $X_i = \text{supp } \varphi_i$  die Träger der  $i$ -ten Finite-Elemente-Basisfunktion und  $X_{t_k} = \bigcup_{i \in t_k} X_i$ ,  $k = 1, 2$ , disjunkte Teilgebiete des Berechnungsgebiets. Der Parameter  $\eta$  hängt vom vorliegenden Problem ab. In (1) wird folglich die Beziehung zwischen den Durchmessern der Gebiete  $X_{t_1}$ ,  $X_{t_2}$  und ihrem Abstand geprüft.

Bisher wird die Partition der Matrix auf Basis geometrischer Informationen vorgenommen. Auf Grund der zunehmenden geometrischen Komplexität der zu modellierenden Bauteile oder adaptiver Verfeinerungen werden verstärkt rein algebraische Verfahren eingesetzt. Die im fünften Kapitel eingeführte algebraische Konstruktion hierarchischer Matrizen folgt dieser aktuellen Entwicklung. Dazu wird eine algebraische Variante der Zulässigkeitsbedingung (1) motiviert,

$$\min\{\text{diam } t_1, \text{diam } t_2\} \leq \eta \text{dist}(t_1, t_2). \quad (2)$$

Hierbei werden  $t_1, t_2 \subset I$  als Menge von Knoten im Matrixgraphen aufgefasst. Der Abstand und die Durchmesser werden im Matrixgraphen berechnet. Auf Basis von (2) wird die Existenz der Faktoren  $L$  und  $U$  der  $LU$ -Zerlegung im hierarchischen Format für gut konditionierte Matrizen bewiesen. Die numerischen Experimente zeigen aber, dass nicht nur gut konditionierte Probleme gelöst werden können. Nach dem Existenzbeweis für die  $LU$ -Zerlegung im hierarchischen Format wird eine konkrete algorithmische Realisierung der Partition angegeben. Diese nutzt lediglich die Besetzungsstruktur der Matrix aus. Die Besetzungsstruktur wird dazu als Matrixgraph interpretiert. Zur Auswertung der Zulässigkeitsbedingung werden verschiedene Approximationsalgorithmen für die Näherung des Abstands aus (2) im Matrixgraphen vorgestellt. Die neue Methode ermöglicht somit die Behandlung von Problemen, ohne auf die Kenntnis der Geometrie angewiesen zu sein.

Im sechsten Kapitel werden hierarchische Vorkonditionierer, basierend auf geometrischer bzw. algebraischer Matrixpartition, verglichen. Es zeigt sich, dass bei ähnlicher Wirkung des Vorkonditionierers die algebraische Konstruktion sowohl bzgl. der Zeit als auch der Speicherbelegung effizienter ist. Weiterhin werden die Laufzeit sowie der Speicherverbrauch der neuen Methode mit denen eines direkten Verfahrens (PARDISO) und denen eines mit Algebraischen Mehrgitter-Verfahren (AMG) vorkonditionierten CG-Verfahrens bzgl. der Asymptotik und der Robustheit verglichen. Hierarchische Vorkonditionierer erweisen sich dabei als die speichereffizienteste Methode. Das AMG-Verfahren ist asymptotisch für ein zu lösendes lineares System das schnellste Verfahren. Sind mehrere rechte Seiten zu lösen, lohnt sich auf Grund der leicht durch Anpassen der blockweisen Genauigkeit steuerbaren Qualität der hierarchischen Vorkonditionierer der Einsatz dieser. Zudem ist das CG-Verfahren mit hierarchischer Vorkonditionierung deutlich robuster als das AMG vorkonditionierte CG-Verfahren.

Die numerischen Experimente zur Parallelisierbarkeit der Konstruktion des hierarchischen Vorkonditionierers zeigen für kleine Prozessoranzahlen eine sehr gute parallele Effizienz. Diese sinkt für 12 Prozessoren auf ca. 50% ab.

Der Vergleich von AMG-vorkonditioniertem und hierarchisch vorkonditioniertem CG-Verfahren innerhalb des Newton-Verfahrens ergibt für die Gesamt-Rechenzeiten der Experimente Vorteile für das hierarchisch vorkonditionierte CG-Verfahren.