

Michael Bader

Robuste, parallele Mehrgitterverfahren
für die Konvektions-Diffusions-Gleichung



Herbert Utz Verlag · Wissenschaft
München

Die Deutsche Bibliothek – CIP-Einheitsaufnahme
Ein Titeldatensatz für diese Publikation ist
bei Der Deutschen Bibliothek erhältlich

Zugleich: Dissertation, München, Techn. Univ., 2001

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, der Entnahme von Abbildungen, der Wiedergabe auf photomechanischem oder ähnlichem Wege und der Speicherung in Datenverarbeitungsanlagen bleiben – auch bei nur auszugsweiser Verwendung – vorbehalten.

Copyright © Herbert Utz Verlag GmbH 2001

ISBN 3-8316-0040-6

Printed in Germany

Herbert Utz Verlag GmbH, München

Tel.: 089/277791-00 – Fax: 089/277791-01

Inhaltsverzeichnis

1. Numerische Simulation von Strömungen	1
1.1. Numerische Lösung von Poisson- und Konvektions-Diffusions-Gleichungen	2
1.1.1. Bedeutung der Poisson- und Konvektions-Diffusions-Gleichung	2
1.1.2. Diskretisierung zum linearen Gleichungssystem	3
1.2. Schnelles Lösen linearer Gleichungssysteme	4
1.2.1. Geschichtliches	4
1.2.2. Bedeutung schneller, robuster und paralleler Verfahren	6
1.3. Entwicklung schneller iterativer Löser auf Basis der rek. Substrukturierung	7
1.3.1. Aufgabenstellung	7
1.3.2. Gewählter Lösungsansatz	7
1.3.3. Überblick über diese Arbeit	8
2. Multilevelansätze	11
2.1. Mehrgitterverfahren	11
2.2. Hierarchische Basen und Erzeugendensysteme	15
2.2.1. Knotenbasen	15
2.2.2. Hierarchische Basen	17
2.2.3. Hierarchische Erzeugendensysteme	19
2.2.4. Durchführung der Hierarchisierung	20
2.3. Erzeugendensysteme und Mehrgitterverfahren	23
2.4. Mehrgitterverfahren für Konvektions-Diffusions-Gleichungen	25
2.4.1. Anwendung von Standard-Mehrgitterverfahren auf die Konvektions-Diffusions-Gleichung	25
2.4.2. Ansätze für robuste Mehrgitterverfahren für die Konvektions-Diffusions-Gleichung	27
3. Rekursive Substrukturierung	29
3.1. Direkte Löser auf Basis der rekursiven Substrukturierung	30
3.1.1. Gebietszerlegung	31
3.1.2. Aufstellen und Zusammensetzen der lokalen Gleichungssysteme	33

- 3.1.3. Kondensation 38
- 3.1.4. Rekursive Lösung 41
- 3.1.5. Rechenzeit, Speicheraufwand und Parallelität 41
- 3.2. Iterative Verfahren 44
 - 3.2.1. Vorkonditionierung des Gleichungssystems 45
 - 3.2.2. Vorkonditionierung des Schurkomplements 46
 - 3.2.3. Teilelimination 47
 - 3.2.4. Relaxationszyklen 50
 - 3.2.5. Grundform des iterativen Algorithmus 51
- 3.3. Rekursive Substrukturierung auf Erzeugendensystemen 52
 - 3.3.1. Zerlegung der Teilgebiete und Akkumulation der Gleichungssysteme 52
 - 3.3.2. Verflechtung von Substrukturierung und Hierarchisierung 55
 - 3.3.3. Teilelimination auf Erzeugendensystemen 57
 - 3.3.4. Der Algorithmus 58
- 4. Lösen der Konvektions-Diffusions-Gleichung 63**
 - 4.1. Rekursive Substrukturierung mit teilweiser Elimination von Kopplungen 63
 - 4.1.1. Überlegungen zur Wahl eines idealen Grobgitters 63
 - 4.1.2. Rekursive Substrukturierung mit Elimination der wichtigsten Kopplungen 65
 - 4.1.3. Realisierung der Eliminationsstrategie 67
 - 4.2. Durch Teilelimination zum Mehrgitterverfahren 70
 - 4.2.1. Teilelimination als Basistransformation 70
 - 4.2.2. Teilelimination als Gitterauswahl – Stabile Grobgitterdiskretisierungen auf Erzeugendensystemen 71
- 5. Implementierung des Verfahrens 73**
 - 5.1. Behandlung beliebiger Gebiete und innerer Randbedingungen 73
 - 5.2. Einsatz des Verfahrens als Vorkonditionierer 75
 - 5.3. Datenstrukturen für die effiziente Implementierung des Verfahrens 76
 - 5.3.1. Substrukturierungsbaum 76
 - 5.3.2. Persistent und temporär benötigte Variablen 77
 - 5.3.3. Effizienter Aufbau der Systemmatrizen 79
 - 5.3.4. Ausnutzung der Blockstruktur der Eliminationsmatrizen 79
 - 5.4. Parallelisierung 80
 - 5.4.1. Parallelisierung des Substrukturierungsbaumes 81
 - 5.4.2. Parallelisierung und Vorkonditionierung 81

6. Numerische Ergebnisse	85
6.1. Einfluss der Eliminationsstrategie auf die Robustheit	86
6.1.1. Rekursive Substrukturierung ohne angepasste Eliminationsstrategie	87
6.1.2. Rekursive Substrukturierung mit angepasster Eliminationsstrategie	89
6.1.3. Einfluss von Strömungshindernissen	92
6.2. Performance	95
6.2.1. Bedarf an Rechenzeit und Speicherplatz	95
6.2.2. Parallele Effizienz	100
7. Spezialfälle: Poisson-Gleichung und stark konvektionsdominierte Strömungen	105
7.1. Lösen der Poisson-Gleichung	105
7.1.1. Mehrgitterverfahren für die Poisson-Gleichung	105
7.1.2. Diskussion möglicher Eliminationsstrategien für die Poisson-Gleichung	106
7.1.3. Numerische Ergebnisse	106
7.2. Stark konvektionsdominierte Strömungen	111
7.2.1. Einsetzbarkeit der bisherigen Eliminationsstrategie	111
7.2.2. Eliminationsstrategien für stark konvektionsdominierte Strömungen	113
7.2.3. Numerische Ergebnisse	114
8. Zusammenfassung und Ausblick	119
8.1. Welche Vorgaben wurden erreicht?	119
8.2. Einordnung des vorgestellten Verfahrens	120
8.3. Erweiterung des vorgestellten Verfahrens	121
A. Anhang	125
A.1. Verwendete Diskretisierungs-Schemata	125
A.1.1. Finite Differenzen	125
A.1.2. Finite Elemente	128
A.2. Verwendete Iterationsverfahren	131
A.2.1. Richardson-Iteration	131
A.2.2. CG-Verfahren	132
A.2.3. BiCG-STAB-Verfahren	135
Literaturverzeichnis	137

1. Numerische Simulation von Strömungen

„πάντα ῥεῖ – *alles fließt*“

Der berühmte, auf Heraklit zurückgehende¹ Ausspruch „alles fließt“ ist eigentlich ein philosophischer Leitspruch und bezieht sich vor allem auf das Weltgeschehen und nicht auf Naturwissenschaft oder gar Technik. Nichtsdestotrotz sind Strömungsvorgänge gerade in Naturwissenschaft und Technik nahezu allgegenwärtig. Strömungsphänomene beschäftigen Physiker wie Ingenieure, Meteorologen wie Mediziner, Astronomen und Biologen. In all diesen Disziplinen existieren Anwendungsgebiete, in denen die Vorhersage von bestimmten Strömungsverhältnissen durch Berechnung oder Simulation gefordert ist. Zur Reduzierung des Treibstoffverbrauchs von Kraftfahrzeugen oder Flugzeugen etwa streben wir nach Verbesserung der aerodynamischen Eigenschaften, aber auch der Verbrennungsvorgang selbst wird durch die Strömungen der Verbrennungsgase in den Motoren und Turbinen wesentlich bestimmt. Während die Medizin und Biologie vielleicht die mikroskopischen Strömungen in Gefäßen und Zellen untersucht, wird in der Astronomie auch die Jahrmillionen dauernde Entwicklung von Sternsystemen durch Strömungsvorgänge modelliert. Nicht zuletzt ist es vielleicht die tägliche Wettervorhersage, die uns die globale Bedeutung von Strömungsphänomenen vor Augen führt.

Strömungsvorgänge werden mathematisch beschrieben durch Systeme von partiellen Differentialgleichungen. Als vielleicht bekanntestes solches System bestehen etwa die Navier-Stokes-Gleichungen aus fünf Differentialgleichungen, je eine für die drei Geschwindigkeitsrichtungen, sowie eine für den Druck und eine für die Dichte des Fluids. Die Beschreibung von Temperatur, Stoffkonzentrationen, Turbulenzvorgängen und manches weitere können diese Differentialgleichungssysteme weiter vergrößern. Mathematisch exakte Lösungen für solche komplizierten Modelle zu erhalten ist schlicht hoffnungslos. Der Weg zur erfolgreichen Simulation führt daher über die numerische Berechnung von Näherungslösungen.

¹Der Wahlspruch „alles fließt“ wird allgemein Heraklit von Ephesos (544-483 v.Chr.) zugesprochen, taucht aber in keinem seiner erhaltenen Textfragmente wörtlich auf. Als Zitat findet sich πάντα ῥεῖ erstmals bei Plato.

1.1. Numerische Lösung von Poisson- und Konvektions-Diffusions-Gleichungen

1.1.1. Bedeutung der Poisson- und Konvektions-Diffusions-Gleichung

Die vorliegende Arbeit beschäftigt sich mit der Entwicklung robuster Löser für die Konvektions-Diffusions-Gleichung und, als Spezialfall, für die Poisson-Gleichung.

Die (stationäre) Konvektions-Diffusions-Gleichung

$$-\Delta u + v \cdot \nabla u = f \quad (1.1)$$

beschreibt in der Strömungsmechanik die Verteilung einer physikalischen Größe u – z.B. Wärme, Stoffkonzentration, Impuls, etc. – in einem zeitlich stationären, aber bewegten Fluid, dessen Strömung durch das ortsabhängige Vektorfeld v bestimmt wird. Dementsprechend wird angenommen, dass weder zeitliche Veränderungen der Fluideigenschaften oder der äußeren Einflüsse, noch der Strömungen innerhalb des Fluids, die Verteilung der Größe u beeinflussen. Ebenso wird angenommen, dass die Verteilung der Größe u selbst einen zeitlich konstanten Zustand angenommen hat. Die Verteilung der Größe u wird dann zum einen bestimmt durch den Transport der Größe durch Konvektion, also die Tatsache, dass sie mit der Strömung „mitgerissen“ wird. In Gleichung 1.1 wird dies durch den *konvektiven Term* $v \cdot \nabla u$ modelliert. Zum anderen wird die physikalische Größe durch Diffusion verbreitet, also durch mikroskopische Bewegungen der Fluidpartikel. Dies wird durch den *Diffusions-Term* Δu beschrieben. Äußere Einflüsse, die auf die Ausbreitung einwirken, z.B. Wärmequellen, sind in der Gleichung durch die rechte Seite f wiedergegeben.

Die Poisson-Gleichung

$$-\Delta u = f \quad (1.2)$$

beschreibt dagegen die Verteilung einer physikalischen Größe u in einem stationären, ruhenden Fluid ($v \equiv 0$). Der Transport der physikalischen Größe u im Fluid geschieht hier also allein durch Diffusion. Die Verteilung der Größe ist dann außer von den äußeren Einflüssen f nur noch von der Stärke der Diffusion abhängig. Auf einen expliziten Diffusionskoeffizienten wurde in den beiden Gleichungen 1.1 und 1.2 jedoch verzichtet. Stattdessen sind v und f als relative Größen zu verstehen.

Sowohl die Konvektions-Diffusions-Gleichung als auch die Poisson-Gleichung machen für die Strömungssimulation stark vereinfachende Annahmen über die Eigenschaften der betrachteten Strömung. Der Grund für die Beschränkung auf diese einfachen Modellgleichungen liegt darin, dass beide Gleichungen bei der zeitlichen Diskretisierung der Navier-Stokes-Gleichungen als Teilprobleme auftreten. Dies kann ebenso in ihrer kontinuierlichen wie in ihrer diskretisierten Form der Fall sein. Ein

explizites Zeitschrittverfahren etwa führt in der Regel auf die Lösung einer diskretisierten Poisson-Gleichung für den Druck [1, 15, 50]. Implizite oder semi-implizite Verfahren führen dagegen entsprechend zu einer diskretisierten zeitabhängigen Konvektions-Diffusions-Gleichung [67]. Für den Grenzfall großer Zeitschritte ergibt sich jedoch wieder der in dieser Arbeit betrachtete stationäre Fall.

1.1.2. Diskretisierung zum linearen Gleichungssystem

Für partielle Differentialgleichungen, wie sie die Konvektions-Diffusions-Gleichung und die Poisson-Gleichung darstellen, lassen sich nur in den allerseltensten Fällen exakte Lösungen angeben. Insbesondere für realitätsnahe Strömungsgebiete, etwa solche mit kompliziert geformten Hindernissen, ist fast ausschließlich die näherungsweise, numerische Berechnung auf dem Computer angezeigt.

Dazu wird das kontinuierliche Problem durch eine Diskretisierung in ein endlich dimensionales Problem überführt. Als Diskretisierung können etwa Finite-Differenzen-, Finite-Volumen- oder Finite-Elemente-Verfahren zum Einsatz kommen. Gemeinsam ist den drei Verfahren die Verwendung eines Diskretisierungsgitters.

Die vorliegende Arbeit wird sich ausschließlich mit regulären, kartesischen Gittern auf einem quadratischen **Berechnungsgebiet** $\Omega = [0, 1] \times [0, 1]$ befassen. Ein entsprechendes **Diskretisierungsgitter** Ω_h mit der Gitterweite $h := 2^{-p}$ ist dann definiert als

$$\Omega_h := \{x_{ij} = (i \cdot h, j \cdot h) : i, j = 0, \dots, 2^p\}. \quad (1.3)$$

Dabei darf das „physikalische“ Gebiet durchaus unregelmäßig sein. Es wird dann von dem quadratischen Berechnungsgebiet eingeschlossen, wobei entsprechend innere Ränder ergänzt werden. Die Wahl der Rechtecksgitter und die Beschränkung der Zahl der Unbekannten pro Dimension auf Zweierpotenzen ergeben sich aus der gewünschten Verwandtschaft des Verfahrens zu Geometriebeschreibungen, die auf Quadtree- oder Octree-Strukturen beruhen. Ebenso ist die später verwendete Gebietszerlegungsstrategie darauf abgestimmt.

Der zweite Schritt der Diskretisierung ist die Überführung der partiellen Differentialgleichung in ein lineares Gleichungssystem

$$A_h u_h = f_h, \quad (1.4)$$

dessen Lösungsvektor u_h die zu berechnenden Werte von u an den Gitterpunkten x_{ij} enthält. Das Aufstellen des linearen Gleichungssystems unterscheidet sich je nach verwendetem Diskretisierungsverfahren:

Beim **Finite-Differenzen-Verfahren** wird für jede Unbekannte des Gitters eine Gleichung aufgestellt, die sich aus dem Übergang von den exakten Ableitungen zu den Differenzenformeln an den Gitterpunkten ergibt.¹

¹Ausführliche Beschreibung des Finite-Differenzen-Verfahrens im Anhang A.1.1.

Das *Finite-Volumen-Verfahren* ermittelt die einzelnen Gleichungen aus der Betrachtung der Flüsse und Erhaltungsgleichungen in den Gitterzellen [49].

Beim *Finite-Elemente-Verfahren* schließlich wird eine Lösung der schwachen Formulierung der partiellen Differentialgleichung gesucht. Mit der zugehörigen Bilinearform

$$a: V \times V \rightarrow \mathbb{R}, \quad a(\varphi, u) = \begin{cases} \int -\nabla\varphi \cdot \nabla u + \varphi(v \cdot \nabla u) \, dx & \text{(Konv.-Diff.-Gleichung)} \\ \int -\nabla\varphi \cdot \nabla u \, dx & \text{(Poisson-Gleichung)} \end{cases} \quad (1.5)$$

wird die Differentialgleichung dazu in eine Variationsgleichung überführt:

$$\forall \varphi \in V: \quad a(\varphi, u) = (\varphi, f) = \int \varphi f \, dx. \quad (1.6)$$

In einer *Ritz-Galerkin-Diskretisierung*² wird der Funktionsraum V auf einen endlich dimensionalen Suchraum V_h beschränkt, in dem dann mit Hilfe von Testfunktionen $\varphi_i \in V_h$ die beste Näherungslösung $u \in V_h$ ermittelt wird. Für jede Testfunktion ergibt sich eine Zeile des Gleichungssystems. Im Folgenden werden diejenigen Teile des Algorithmus, die von der Art der Diskretisierung abhängen, jeweils für die Finiten Elemente erläutert.

Generell beschränkt sich diese Arbeit auf Diskretisierungstechniken, die zu Gleichungssystemen führen, die in den einzelnen Gleichungen jeweils nur direkt benachbarte Gitterpunkte miteinander koppeln. Diese Einschränkung ergibt sich aus der später verwendeten Art der Substrukturierung des Berechnungsgebietes in Teilgebiete. Insbesondere werden aus diesem Grund Diskretisierungen höherer Ordnung nicht berücksichtigt.

1.2. Schnelles Lösen linearer Gleichungssysteme

1.2.1. Geschichtliches

Unabhängig von der Art der zeitlichen oder örtlichen Diskretisierung muss zur numerischen Lösung der auftretenden partiellen Differentialgleichungen stets ein Großteil der insgesamt benötigten Rechenzeit für die Lösung der aus der Diskretisierung entstandenen linearen Gleichungssysteme aufgewendet werden. Daher wurden von Beginn der Strömungssimulation an effiziente Verfahren zur Lösung der typischerweise auftretenden Gleichungssysteme gesucht.

²Ausführliche Beschreibung der Ritz-Galerkin-Diskretisierung und des Finite-Elemente-Verfahrens im Anhang A.1.2.

Wegen der Dünnbesetztheit der Gleichungssysteme sind naive, direkte Verfahren, wie etwa die Gauß-Elimination, nicht gut zur Lösung geeignet. Der durch die Eliminationsoperationen verursachte *fill-in* in den Gleichungssystemen treibt den Speicherplatz inakzeptabel in die Höhe. Auch der Bedarf an Rechenzeit wäre bei diesem Vorgehen viel zu hoch. Immerhin lässt sich durch geschickte Reihenfolge der Eliminationen und Anordnung der Unbekannten auch mit direkten Verfahren eine halbwegs akzeptable Performance erzielen. Bei gewöhnlicher, zeilenweiser Nummerierung der Unbekannten kann im zweidimensionalen Fall mit Bandlösern eine Rechenzeitkomplexität von $\mathcal{O}(N^2)$ bei einem Speicherplatzbedarf von $\mathcal{O}(N^{3/2})$ erreicht werden. Eine weitere Reduktion, zumindest für eine gewisse Klasse von Gleichungssystemen bzw. zugrunde liegender Diskretisierungen, ist durch das Verfahren der *nested dissection* [22] möglich. Durch geschickte Umnummerierung und eine *divide&conquer*-Strategie wird dabei die Rechenzeitkomplexität auf $\mathcal{O}(N^{3/2})$ und der Speicherplatzbedarf auf $\mathcal{O}(N \log N)$ gedrückt. Dabei kommt diesem Verfahren entgegen, dass der Hauptaufwand im *setup* liegt, also in der Umnummerierung und *divide&conquer*-Elimination. Die eigentliche Lösung kann danach sogar in $\mathcal{O}(N \log N)$ Operationen berechnet werden. In vielen Anwendungen ändert sich aber nur die rechte Seite der Gleichungssysteme. Bei gleichbleibender Systemmatrix ist dann jedes derartige Gleichungssystem in $\mathcal{O}(N \log N)$ lösbar.

Mehr als die Rechenzeit ist jedoch der erhöhte Speicherplatzbedarf in numerischen Simulationsrechnungen besonders schmerzhaft. Während eine längere Wartezeit auf die Ergebnisse unter Umständen noch verkraftet werden kann, bildet Mangel an Speicherplatz schnell eine unüberwindbare Grenze für die Größe des diskretisierten Problems und damit für die Genauigkeit der Simulationsrechnung.

Schon früh wurde daher eine Verlagerung auf iterative Verfahren deutlich. Die ersten und einfachsten Iterationsverfahren, z.B. die Gauß-Seidel- und Jacobi-Iteration, liefern praktisch ohne zusätzlichen Speicherbedarf eine Rechenzeitkomplexität von $\mathcal{O}(N^2)$. Die einzelnen Iterationsschritte sind dabei zwar von linearem Aufwand, jedoch verschlechtern sich die Konvergenzraten für die typischen zu lösenden Gleichungssysteme mit einer Erhöhung der Problemgröße derart, dass auch die Zahl der benötigten Iterationsschritte linear mit der Zahl der Unbekannten wächst. Einen Fortschritt in dieser Richtung brachten die SOR-Verfahren (*successive over-relaxation*), die etwa im Fall der Poisson-Gleichung bei optimaler Wahl eines Überrelaxationsfaktors die Rechenzeitkomplexität auf $\mathcal{O}(N^{3/2})$ reduzieren. Dadurch wird ohne den Nachteil eines zusätzlichen Speicheraufwandes eine mit den besten direkten Methoden vergleichbare Komplexität erreicht.

Der Durchbruch der iterativen Verfahren zur effizienten Lösung partieller Differentialgleichungen gelang mit der Entwicklung der sogenannten *Mehrgitterverfahren*. Zumindest für bestimmte Problemklassen können diese eine sowohl bzgl. Rechenzeit als auch bzgl. Speicherplatzbedarf optimale Komplexität von $\mathcal{O}(N)$ erreichen. Für das Modellproblem der Poisson-Gleichung auf dem Einheitsquadrat benötigen die schnellsten Mehrgitterverfahren heute im Wesentlichen nur noch eine einzige Mehr-

gitteriteration. Mit diesem minimalen Rechenaufwand, der weniger als 10 Iterationen des Gauß-Seidel-Verfahrens entspricht, kann die partielle Differentialgleichung bis auf Diskretisierungsgenauigkeit gelöst werden. Diese sogenannte *textbook multigrid convergence* ist allerdings noch bei weitem nicht für alle Problemklassen erreicht [10].

1.2.2. Bedeutung schneller, robuster und paralleler Verfahren

Komplizierte Hindernisgeometrien, starke Konvektion oder zirkulierende Strömungen stellen hohe Anforderung an die Entwicklung effizienter Mehrgitterlöser für Anwendungen in der Strömungssimulation. Für viele solche Fälle lässt sich die genannte optimale $\mathcal{O}(N)$ -Komplexität mit den bisherigen Mehrgitterverfahren nicht erzielen. Neben den Wunsch nach Robustheit und guter Effizienz über eine große Klasse von Problemen hinweg, gesellen sich Forderungen wie etwa nach guter Parallelisierbarkeit des Verfahrens, die die Entwicklung zusätzlich erschweren.

Im Zeitalter der Hoch- und Höchstleistungsrechner mag man freilich versucht sein, nicht mehr nach optimaler Komplexität zu streben, sondern einfach die nächste Rechnergeneration abzuwarten. Das folgende Gedankenexperiment soll verdeutlichen, dass nicht *obwohl*, sondern gerade *weil* die verfügbare Hardware immer leistungsfähiger wird, stets die optimale Komplexität des Algorithmus angestrebt werden sollte.

Nach *Moore's Law* [46] verdoppelt sich die verfügbare Rechenkapazität, also insbesondere Rechengeschwindigkeit und Speicherkapazität, etwa alle 18 Monate. Folglich wird man alle 18 Monate aufgrund des dann in doppelter Menge vorhandenen Speicherplatzes ein Problem mit doppelt so vielen Unbekannten (also z.B. entsprechend höherer Auflösung) in Angriff nehmen können. Bei einem Verfahren der Rechenzeitkomplexität $\mathcal{O}(N)$ wird das berechnete Ergebnis in derselben Zeit wie bisher erhalten, da sich auch die Rechengeschwindigkeit entsprechend verdoppelt hat. Bei schlechterer Rechenzeitkomplexität erhöht sich dagegen die Wartezeit auf die Ergebnisse deutlich. Da man generell bestrebt sein wird, stets das größtmögliche Problem zu rechnen – und oft ist gerade der verfügbare Speicherplatz die härtere Grenze – so wird klar, dass man bei einer stärker als $\mathcal{O}(N)$ wachsenden Rechenzeit die Steigerung der Rechnerkapazität nicht linear in eine Steigerung der Problemkapazität umsetzen kann.

Neben der sich ständig verdoppelnden Rechenkapazität gibt es noch eine zweite Beobachtung, aus der sich Forderungen an leistungsfähige Verfahren ableiten lassen. Die Prozessorgeschwindigkeit, die Speicherzugriffszeiten und die Kommunikationsgeschwindigkeit zwischen Prozessoren in Parallelrechnern entwickeln sich keinesfalls gleichmäßig. Vielmehr existiert zur Zeit ein Trend, dass die Prozessorgeschwindigkeit deutlich schneller wächst als die des Speichers und der Kommunikation. Künftige Hochleistungsrechner – und damit hochparallele Rechner – werden also nur dann optimal ausgenutzt werden können, wenn die Algorithmen mit hoher par-

alle Effizienz auch auf Architekturen mit verteiltem Speicher und verhältnismäßig langsamer Kommunikation laufen [40].

1.3. Entwicklung schneller iterativer Löser auf Basis der rekursiven Substrukturierung

1.3.1. Aufgabenstellung

Die vorliegende Arbeit konzentriert sich auf die folgenden drei wesentlichen Anforderungen, die ein „idealer“ Algorithmus erfüllen sollte:

- Als erstes sollte er eine **optimale Komplexität** sowohl bzgl. benötigter Rechenzeit, als auch bzgl. erforderlichem Speicherplatz aufweisen. Rechenzeit und Speicherbedarf des entwickelten Algorithmus sollen für Konvektions-Diffusions-Gleichungen nicht stärker als linear mit der Zahl der Unbekannten wachsen.
- Zweites Ziel ist die **gute Parallelisierbarkeit** des Algorithmus. Der Algorithmus soll auch auf Systemen, die nicht primär als Parallelrechner ausgelegt sind, etwa auf vernetzten Arbeitsplatzrechnern, gute parallele Effizienz aufweisen. Dabei soll die parallele Version gegenüber der seriellen keine Modifikationen des Algorithmus erforderlich machen.
- Vor allem aber soll sich der Algorithmus durch **Robustheit** für eine möglichst umfangreiche Klasse von Problemen auszeichnen. Seine optimalen Komplexitätseigenschaften soll er möglichst unabhängig von der Strömungsgeometrie und -geschwindigkeit liefern können. Dabei beschränkt sich diese Arbeit in puncto Geschwindigkeit zunächst auf den Fall einer im Verhältnis zur Auflösung des Diskretisierungsgitters mäßigen Dominanz der Konvektion über die Diffusion. Die Konvektion darf zwar prinzipiell beliebig stark werden, im Gegenzug soll dann aber auch das Diskretisierungsgitter fein genug sein, um das physikalische Problem weiterhin korrekt zu beschreiben. Das Gitter wird insbesondere als so fein angenommen, dass keine zusätzliche Diffusion zur Stabilisierung der Diskretisierung eingesetzt werden muss. Der Fall der noch stärkeren Konvektion erfordert ein modifiziertes Vorgehen, für das lediglich einige mögliche Ansätze diskutiert werden.

1.3.2. Gewählter Lösungsansatz

Die Forderung nach **optimalen Komplexitätsschranken** für Rechenzeit und Speicherplatz soll durch das Arbeiten auf hierarchischen Basen und Erzeugendensysteme