IWR Heidelberg

Preprint 92 - 10

1992

1

Die nichtüberlappende Gebietszerlegungsmethode zur Parallelisierung und Vorkonditionierung des CG-Verfahrens - Algorithmen und numerische Vergleiche -

Gundolf Haase
TU Chemnitz, FB Mathematik
PSF 964, O-9010 Chemnitz

Zusammenfassung

Die Entwicklung paralleler Algorithmen zur Lösung partieller Differentialgleichungen auf Mehrprozessorrechnern mit verteiltem Speicher ist gegenwärtig eine zentrale Fragestellung in der Numerischen Mathematik und Angewandten Informatik. Stand anfangs die reine Parallelisierbarkeit meist bekannter Algorithmen im Mittelpunkt des Interesses, fallen nunmehr Fragen der Effizienz der Konstruktion neuer paralleler Lösungsstrategien, der Parallelisierung und der Anwendbarkeit der neu entwickelten Verfahren auf praktische Aufgabenstellungen stärker ins Gewicht. Verschiedene Varianten von Gebietszerlegungsmethoden nehmen gegenwärtig einen zentralen Platz bei der Entwicklung hocheffizienter neuer paralleler Verfahren ein. Die vorliegende Arbeit versteht sich daher als Einführung in eine Variante der nichtüberlappenden Gebietszerlegungstechnik und in die Implementierung der entsprechenden Algorithmen auf einem MIMD-Rechner mit verteiltem Speicher (distributed memory) und Botschaftenaustausch (message passing). Die auf einem Transputersystem für ausgewählte Modellbeispiele durchgeführten Experimente dienen in erster Linie zur numerischen Unterlegung der theoretischen Resultate. Die erhaltenen Rechenzeiten zeigen andererseits die zu erwartende Konkurrenzfähigkeit der untersuchten Gebietszerlegungstechniken gegenüber der globalen Parallelisierung etablierter Verfahren, wie z.B. des Multigrid-Verfahrens, in harten praktischen Anwendungen.

1 Einleitung

Die Entwicklung paralleler Algorithmen zur Lösung partieller Differentialgleichungen auf Mehrprozessorrechnern mit verteiltem Speicher erlangte durch die Verfügbarkeit der

*Diese Arbeit entstand im Wintersemester 1991 während eines 3-monatigen Aufenthalts am IWR Heidelberg. Mein Dank für die erhaltene Unterstützung gilt insbesondere Prof. G. Wittum und Herrn W. Weiler.

entsprechenden Rechentechnik (u.a. INMOS-Transputer) eine zentrale Bedeutung in der numerischen Mathematik und Angewandten Informatik. Ausgehend von verschiedenen Denkansätzen sind gegenwärtig eine Reihe von Parallelisierungstechniken für Mehrprozessorrechner mit verteiltem Speicher erkennbar (vgl. z.B. [15, 8, 9, 25]) Die ältesten Ansätze gehen auf Arbeiten von H. Schwarz (1869,[36]) zür überlappenden Gebietszerlegungsmethode und auf die in den 60-er Jahren entwickelte klassische Finite-Elemente-Substrukturtechnik [33] zurück. Diese Arbeiten entstanden ohne jeglichen Parallelisierungshintergrund. Die Ideen dieser Arbeiten initialisierten allerdings Mitte der 80-er Jahre die Entwicklung einer Vielzahl neuer numerischer Algorithmen zur Lösung partieller Differentialgleichungen auf der Basis der überlappenden und nichtüberlappenden Gebietszerlegung [11, 12, 6]. Die F.E. Substrukturtechnik basiert auf einer nichtüberlappenden Gebietszerlegung und auf der Zurückführung des globalen Problems auf ein Schur-Komplementproblem zur Bestimmung der Unbekannten auf den Koppelrändern. Die neuen Herangehensweisen nutzen diese Ideen vor allem zur Konstruktion paralleler Vorkonditionierungsoperatoren aus, die dann in parallelen Varianten des CG-Verfahrens bzw. CG-like-Verfahren zur Verbesserung der Konvergenzgeschwindigkeit dieser Verfahren genutzt werden [1, 3, 4, 6, 10, 11, 12, 21, 20]. Ziel dieser Technik ist stets eine Entkopplung des Gesamtproblems in eine Anzahl lokaler Probleme.

Eine völlig andere Herangehensweise ist die der Parallelisierung globaler Algorithmen (Block-Gauß-Seidel [**?**], unvolsständige Zerlegungen [2], Multigrid [29], ...). Hier kommt es vorrangig darauf an, die Arbeit so auf die Prozessoren zu verteilen, daß eine gleichmäßige Auslastung der Prozessoren bei minimalem Datenaustausch erreicht wird. Detaillierte Vergleiche obiger 2 Parallelisierungsvarianten sind in [34] und [13] zu finden.

Die neueste Richtung der Entwicklung paralleler Algorithmen wird durch die Multilevel Methoden eingeschlagen. Diese Verfahren sind eine Weiterentwicklung der Multigrididee [23] bzw. des Ansatzes der hierarchischen Basen [40]. Die Implementierung dieser Verfahren geschieht einerseits über Gebietszerlegung [32] und andererseits über Parallelisierung des globalen Verfahrens [7]. Weitere vergleichende Betrachtungen sind in Xu [39] und Smith [37] zu finden.

Die vorliegende Arbeit versteht sich als Einführung in eine Variante der nichtüberlappenden Gebietszerlegungstechnik und in die Implementierung der entsprechenden Algorithmen auf einem MIMD-Rechner mit verteiltem Speicher (distributed memory) und Botschaftenaustausch (message passing). Die auf einem Transputersystem für ausgewählte Modellbeispiele durchgeführten Experimente dienen in erster Linie zur numerischen Unterlegung der theoretischen Resultate. Die erhaltenen Rechenzeiten zeigen andererseits die zu erwartende Konkurrenzfähigkeit der untersuchten Gebietszerlegungstechniken gegenüber der globalen Parallelisierung etablierter Verfahren, wie z.B. des Multigrid-Verfahrens, in harten praktischen Anwendungen.

Nachdem im Kapitel 2 die Modellprobleme und die Dirichletsche Gebietszerlegungsmethode (DD) eingeführt wurden, behandelt Kapitel 3 die algorithmischen Aspekte und diskutiert verschiedene Varianten der zur Dirichletschen DD-Methode gehörenden Vorkonditionierung. Wie bereits in vorangegangen Arbeiten [19, 18, 21, 17, 20] gezeigt wurde, läßt sich die Dirichletsche DD-Methode als Additive Schwarzsche Methode (ASM) interpretieren. Die mit der

ASM verbundene Basistransformation wird in Kapitel 4 näher untersucht. Kapitel 5 enthält vergleichende Betrachtungen bei der Verwendung von verschiedenen Techniken (Multigrid, MAF, Frequenzfiltermethode) in der Basistransformation. Kapitel 4 und 5 sind mit Testrechnungen unterlegt. Schließlich werden in Kapitel 6 Schlußfolgerungen aus den Überlegungen und numerischen Beispielen der vorangehenden Kapitel gezogen und Ausblicke auf weitere Möglichkeiten der Gebietszerlegungsmethode gegeben. Gleichzeitig enthält dieses Kapitel eine Effizienzuntersuchung.

2 Zusammenstellung einiger Grundlagen

2.1 Das Modellproblem und die Modellgebiete

Alle Beispiele nehmen auf die Randwertaufgabe

$$\left.\begin{array}{rcll} -\nabla\,(mat(x)\,\nabla\,u(x)) & = & \mathbf{1} & x \text{ in } \Omega \\ u(x) & = & 0 & x \text{ auf } \partial\Omega \end{array}\right\} \tag{2.1}$$

Bezug, wobei $u(x)$ die gesuchte Lösung im Gebiet Ω ist und $mat(x)$ die Materialkoeffizienten beschreibt. Falls $mat(x) \geq \text{const.} > 0$ für fast alle x aus Ω , ergibt sich aus (2.1) das symmetrische, $\mathbf{V}_0$-elliptische und $\mathbf{V}_0$-beschränkte Variationsproblem der Form (2.2):

$$\text{Finde} \quad u \in \mathbf{V}_0 : \quad a(u,v) \;= < F, v > \quad \forall v \in \mathbf{V}_0 \quad . \tag{2.2}$$

Nach einer F.E.-Triangularisierung von Ω und der entsprechenden Diskretisierung geht die Variationsformulierung (2.2) in das Gleichungssystem

$$K\,\underline{u} \;=\; \underline{f} \tag{2.3}$$

mit der symmetrischen und positiv definiten Matrix K über.

Betrachten wir nun das Modellgebiet 1 (= Bsp. 1) welches sich aus $\overline{\Omega} = \overline{\Omega}_1 \cup \overline{\Omega}_2$ mit $\overline{\Omega}_1 = [0,1] \times [0,1]$ und $\overline{\Omega}_2 = [1,2] \times [0,1]$ zusammensetzt. Bild 1 illustriert die Unterteilung der Knoten in innere Knoten und in Knoten auf dem Koppelrand Γ_C. Im folgenden werden die entsprechenden Teilvektoren mit "I" und "C" indiziert. Die Numerierung geschieht derart, daß zuerst die Koppelrandknoten und danach die inneren Knoten zuerst in Ω_1 und dann in Ω_2 numeriert werden. Somit wird z.B. der Lösungsvektor $\underline{u}$ als $\begin{pmatrix}\underline{u}_C\\ \underline{u}_I\end{pmatrix}$ geschrieben. In analoger Weise wird (2.3) in die Blockform

$$\begin{pmatrix} K_C & K_{CI} \\ K_{IC} & K_I \end{pmatrix} \begin{pmatrix}\underline{u}_C\\ \underline{u}_I\end{pmatrix} = \begin{pmatrix}\underline{f}_C\\ \underline{f}_I\end{pmatrix} \quad , \tag{2.4}$$

überführt. Die Matrix K_I besitzt die Blockdiagonalgestalt

$$K_I = diag\,(K_{I,i})_{i=1,2,\ldots,p} \quad , \tag{2.5}$$

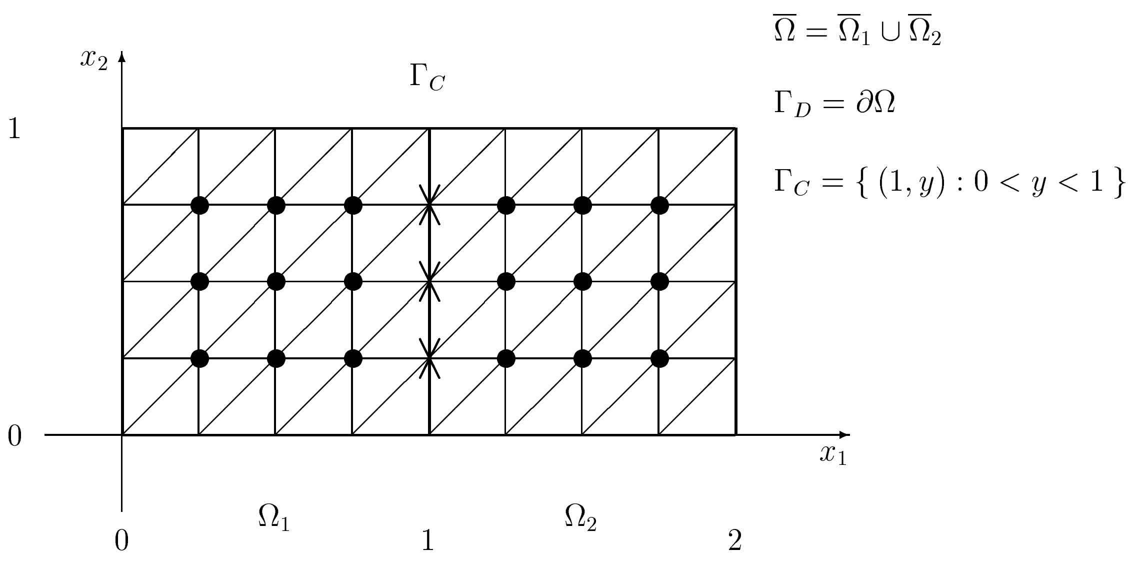


Abbildung 1: Gebietszerlegung und Triangularisierung (2 Gitter) des Modellgebietes Ω ($N_C = 3$, $N_I = 18$, $N_{I,1} = N_{I,2} = 9$)

($p =$ Anzahl der Teilgebiete, in Bsp. 1 ist $p = 2$) wobei die Diagonalblöcke $K_{I,i}$ die partielle Differentialgleichung (2.1) mit $\Omega := \Omega_i$ und $\partial\Omega := \partial\Omega_i$ approximieren. Die Rechteckmatrizen K_{IC} und $K_{CI} = K_{IC}^T$ erfüllen eine Kopplungsfunktion zwischen den Koppelknoten und den inneren Knoten und zerfallen analog zu K_I in rechteckige Blockmatrizen $K_{IC,i}$. Diese Matrixstrukturen sind nicht vom Modellgebiet 1 abhängig. Etwas anders verhält es sich mit der Matrix K_C, die im Modellgebiet 1 aus einem einzigen Block besteht, jedoch im Falle streifenweiser Zerlegung des Gebietes $[0, 2] \times [0, 1]$ eine Blockdiagonalstruktur (Blockanzahl = Anzahl Streifen -1) aufweist. Noch komplizierter wird die Struktur von K_C bei dem Modellgebiet 2, (siehe Bild 2) da zusätzlich noch Crosspoints auftreten,

d.h. Punkte in Teilgebietsecken. Wenn diese Crosspoints bei der Numerierung zuerst berücksichtigt werden, ergibt sich folgende Struktur von K_C :

$$K_C = \begin{pmatrix} K_V & K_{VE} \\ K_{VE}^T & K_E \end{pmatrix} . \tag{2.6}$$

Diese Struktur von K_C legt eine getrennte Behandlung der reinen Koppelränder (“E“) (ohne Crosspoints) und den Crosspoints (“V“) nahe, insbesondere werden die Blockstrukturen von K_E und K_{VE} ausgenutzt. Weiteres dazu ist in Abschnitt 3.2 zu finden.

Im Modellgebiet 1 wird stets $mat(x) \equiv 1\ \forall x \in \Omega$ angenommen, während die Materialkoeffizienten in Modellgebiet 2 nur innerhalb der Teilgebiete Ω_i konstant sind, sich ansonsten von Teilgebiet zu Teilgebiet unterscheiden können.

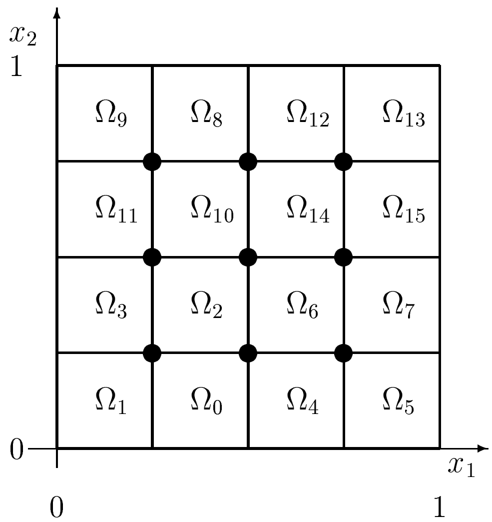


Abbildung 2: Gebietszerlegung von Modellgebiet 2 in 16 Teilgebiete (Gebietsnumerierung derart, daß im Hypercube minimale Kommunikationswege auftreten)

2.2 Die einfache DD- und die ASM-DD-Vorkonditionierung

In den letzten Jahren hat die Parallelrechentechnik eine rasante Entwicklung durchlaufen. Mehrprozessorrechner mit einer beträchtlichen Anzahl an Prozessoren sind z.B. in Form von Transputersystemen kostengünstig auf dem Markt zu erwerben. Im krassen Gegensatz dazu steht die Entwicklung der Software, insbesondere der Applikationssoftware. Die Schaffung paralleler Algorithmen in der Finite-Elemente (F.E.) Simulation ist dabei sicherlich ein Schwerpunkt in der angewandten mathematischen Forschung. Gebietszerlegungstechniken eröffnen hierbei eine Vielzahl von Möglichkeiten zur Parallelisierung von F.E. Software. Die nichtüberlappende Gebietszerlegung ist in der FEM unter dem Namen F.E. Substrukturtechnik wohlbekannt und wurde seit Anfang der 60-er Jahre zur effizienten Erstellung von F.E. Modellen und zur Lösung des F.E. Gleichungssystems auf Rechnern mit beschränktem Hauptspeicher genutzt (siehe z.B. [33]). Hierbei wird die Matrix K aus (2.4) faktorisiert und man erhält

$$K = \begin{pmatrix} I_C & K_{CI}K_I^{-1} \\ O & I_I \end{pmatrix} * \begin{pmatrix} S_C & O \\ O & K_I \end{pmatrix} * \begin{pmatrix} I_C & O \\ K_I^{-1}K_{IC} & I_I \end{pmatrix}, \tag{2.7}$$

wobei $S_C = K_C - K_{CI}K_I^{-1}K_{IC}$ das Schur-Komplement darstellt. Wir benutzen diese Faktorisierung nicht zum Lösen des Gleichungssystems (2.3) sondern nur zur Vorkonditionierung im cg-Algorithmus (siehe Kapitel 3). Indem wir die Operatoren K_I und S_C durch spektraläquivalente, symmetrische und positiv definite Operatoren C_I und C_C ersetzen, d.h. es existieren positive Konstanten $\underline{\gamma}_I$, $\overline{\gamma}_I$, $\underline{\gamma}_C$ und $\overline{\gamma}_C$, so daß die Spektraläquivalenzungleichungen

$$\underline{\gamma}_I C_I \le K_I \le \overline{\gamma}_I C_I \quad \text{und} \quad \underline{\gamma}_C C_C \le S_C \le \overline{\gamma}_C C_C \tag{2.8}$$

gelten, erhält man die einfache DD-Vorkonditionierung

$$C = \begin{pmatrix} I_C & K_{CI}C_I^{-1} \\ O & I_I \end{pmatrix} * \begin{pmatrix} C_C & O \\ O & C_I \end{pmatrix} * \begin{pmatrix} I_C & O \\ C_I^{-1}K_{IC} & I_I \end{pmatrix}. \tag{2.9}$$

Dieser Vorkonditionierungsoperator C ist spektraläquivalent zu K [20]. Operatorgleichungen der Form(2.8) werden hier und im weiteren immer im Sinne des Euklidischen Skalarproduktes verstanden.

Schaut man sich (2.9) genauer an, kann man die beiden Dreiecksmatrizen als Basistransformation der ursprünglichen Basis des Lösungsraumes (d.h. des von den F.E.-Basisfunktionen aufgespannten Raumes) in die approximative diskrete harmonische Basis auffassen und kann diese Vorkonditionierung von der Additiven Schwarzschen Methode (ASM) ableiten (genaueres siehe [36, 19, 18, 21, 17, 20]). Somit muß man eigentlich die Basistransformation (die Dreiecksmatrizen) von der Vorkonditionierung in der neuen Basis (der Diagonalmatrix) unterscheiden. Mittels dieser Überlegungen gelangt man zur ASM-DD-Vorkonditionierung

$$C = \begin{pmatrix} I_C & K_{CI}B_I^{-T} \\ O & I_I \end{pmatrix} * \begin{pmatrix} C_C & O \\ O & C_I \end{pmatrix} * \begin{pmatrix} I_C & O \\ B_I^{-1}K_{IC} & I_I \end{pmatrix}, \tag{2.10}$$

wobei die Operatoren K_I und $S_C + T_C$ durch spektraläquivalente, symmetrische und positiv definite Operatoren C_I und C_C ersetzt werden, d.h. es existieren positive Konstanten $\underline{\gamma}_I$, $\overline{\gamma}_I$, $\underline{\gamma}_C$ und $\overline{\gamma}_C$ so daß die Spektraläquivalenzungleichungen

$$\underline{\gamma}_I C_I \leq K_I \leq \overline{\gamma}_I C_I \quad \text{und} \quad \underline{\gamma}_C C_C \leq S_C + T_C \leq \overline{\gamma}_C C_C. \tag{2.11}$$

erfüllt werden. Die Konstanten $\underline{\gamma}_I$, $\overline{\gamma}_I$, $\underline{\gamma}_C$ und $\overline{\gamma}_C$ in (2.11) sind verschieden von den in (2.8) auftretenden Konstanten gleichen Namens. Der neue Operator $T_C = K_{CI}(K_I^{-1} - B_I^{-T})K_I(K_I^{-1} - B_I^{-1})K_{IC}$ ist offenbar aufs engste mit der Basistransformationsmatrix B_I verknüpft, welche nur nichtsingulär zu sein hat. Diese relativ freie Wahl von B_I wird in Kapitel 4 Gegenstand der Untersuchungen sein. In [19] wurde gezeigt, daß der Vorkonditionierungsoperator C aus (2.10) unter den Voraussetzungen (2.11) spektraläquivalent zu K ist und die Konditionszahl $\kappa(C^{-1}K)$ des vorkonditionierten Systems sich durch

$$\kappa(C^{-1}K) \leq \left(\overline{\gamma}_1/\underline{\gamma}_1\right)\left(\sqrt{\mu} + \sqrt{1+\mu}\right)^2 \tag{2.12}$$

abschätzen läßt, wobei $\mu = \varrho(S_C^{-1}T_C)$ der Spektralradius von $S_C^{-1}T_C$ ist, sowie $\overline{\gamma}_1 = max\{\overline{\gamma}_C, \overline{\gamma}_I\}$ und $\underline{\gamma}_1 = min\left\{\underline{\gamma}_C, \underline{\gamma}_I\right\}$. Mehr Details sind u.a. in [17, 20] zu finden.

Da offensichtlich die einfache DD-Vorkonditionierung nur ein Spezialfall der ASM-DD-Vorkonditionierung ist, beschränken sich die weiteren Ausführungen auf letztere.

3 Algorithmische Aspekte

3.1 Eine parallelisierte Version des CG-Verfahrens

Die in diesem Abschnitt beschriebene Parallelisierungsidee geht auf K.H. Law [28] zurück und wurde später für verschiedene Vorkonditionierungstechniken von A. Meyer, U. Langer und G. Haase weiterentwickelt [17, 19, 21, 20, 31]. Die auf der Gebietszerlegungsmethode beruhende Parallelisierung des CG-Verfahrens eignet sich besonders für eine Implementierung auf Mehrprozessorrechnern mit lokalem Speicher und Botschaftenaustausch über Links. Wenn es um Fragen des Kommunikationsaufwandes geht, wird stets eine Hypercubestruktur zugrunde gelegt.

Entsprechend den p Teilgebieten $\overline{\Omega}_i$ $(i = 1, 2, \ldots, p)$ verteilen wir alle Matrizen und Vektoren, die im CG-Algorithmus vorkommen, auf die p Prozessoren P_i $(i = 1, 2, \ldots, p)$ unseres Mehrprozessorrechners. Für die Vektoren nutzen wir zwei Typen der Verteilung, die wir überlappend (Typ I) und addierend (Typ II) nennen wollen:

Typ I : $\underline{\mathbf{u}}, \underline{\mathbf{w}}$ und $\underline{\mathbf{s}}$ werden im Prozessor P_i $(\hat{=} \overline{\Omega}_i)$ als $\underline{\mathbf{u}}_i = A_i \underline{\mathbf{u}}$, $\underline{\mathbf{w}}_i = A_i \underline{\mathbf{w}}$ und $\underline{\mathbf{s}}_i = A_i \underline{\mathbf{s}}$ gespeichert;

Typ II : $\underline{r}, \underline{v}, \underline{f}$ werden in P_i als $\underline{r}_i, \underline{v}_i, \underline{f}_i$ gespeichert, so daß $\underline{r} = \sum\limits_{i=1}^{p} A_i^T \underline{r}_i$ etc. gilt,

wobei A_i die Superelementzusammenhangsmatrix von $\overline{\Omega}_i$ ist, d.h. die Matrix A_i ist eine Boolesche Matrix der Dimension $N_i \times N$ und bildet einen Vektor $\underline{g} \in \mathbf{R}^N$, der alle Knotenparameter enthält, in einen Vektor $\underline{g}_i \in \mathbf{R}^{N_i}$ ab, der nur die zu $\overline{\Omega}_i$ gehörenden Knotenparameter beinhaltet. Mit den eingeführten Booleschen Matrizen A_i kann nun die Steifigkeitsmatrix K in der Form

$$K = \sum_{i=1}^{p} A_i^T K_i A_i \tag{3.1}$$

dargestellt werden, wobei K_i die zum Superelement $\overline{\Omega}_i$ gehörende Superelementsteifigkeitsmatrix ist. In Analogie dazu benutzen wir die Notationen

$$K_{IC,i} = (K_i)_{IC} \ , \ A_{C,i} = (A_i)_C \quad \text{etc.}$$

Mit der oben eingeführten Datenverteilung und den entsprechenden Bezeichnungen können wir nun die parallelisierte Version des cg-Algorithmus wie folgt aufschreiben.

For all $i = 1, 2, \ldots, p$ do in parallel
0. Startschritt Bestimme eine Anfangsnäherung $\underline{\mathbf{u}}_i$, z.B., $\underline{\mathbf{u}}_i = 0$ $\underline{r}_i = \underline{f}_i - K_i \underline{\mathbf{u}}_i$ $\underline{\mathbf{w}} = C^{-1} \sum A_i^T \underline{r}_i$ $\underline{\mathbf{s}}_i = \underline{\mathbf{w}}_i \quad (\underline{\mathbf{w}}_i = A_i \underline{\mathbf{w}})$ $\sigma_i = \underline{\mathbf{w}}_i^T \underline{r}_i$ $\sigma = \sigma^0 = \sum \sigma_i$
Iteration
1. $\check{\underline{v}}_i = K_i \check{\underline{\mathbf{s}}}_i$ $\delta_i = \check{\underline{v}}_i^T \check{\underline{\mathbf{s}}}_i$ $\delta = \sum \delta_i$ $\alpha = \check{\sigma} \,/\, \delta$ 2. $\underline{\mathbf{u}}_i = \check{\underline{\mathbf{u}}}_i + \alpha \check{\underline{\mathbf{s}}}_i$ $\underline{r}_i = \check{\underline{r}}_i - \alpha \check{\underline{v}}_i$ 3. $\underline{\mathbf{w}} = C^{-1} \sum A_i^T \underline{r}_i$ 4. $\sigma_i = \underline{\mathbf{w}}_i^T \underline{r}_i$ $\sigma = \sum \sigma_i$ $\beta = \sigma / \check{\sigma}$ 5. $\underline{\mathbf{s}}_i = \underline{\mathbf{w}}_i + \beta \check{\underline{\mathbf{s}}}_i$ 6. $\sigma \leq \varepsilon^2 * \sigma^0$? $\xrightarrow{\text{nein}}$ goto 1. $\downarrow$ ja stop

Im oben aufgeschriebenen Algorithmus markiert das Symbol " ˇ " die vorangegangenen Iterierten. Das Summenzeichen $\sum = \sum\limits_{i=1}^{p}$ bedeutet eine Aufsummierung bzw. Typenumwandlung, die immer mit Transferoperationen (Kommunikation !) verbunden ist.

Selbst im Falle $C = I$, dh. ohne Vorkonditionierung, erfordert der Schritt 3

$$\underline{\mathbf{w}} = C^{-1} \sum A_i^T \underline{r}_i \tag{3.2}$$

bzw. der entsprechende Startschritt eine Umwandlung eines Vektors vom Typ II in einen Vektor vom Typ I und ist folglich mit einem lokalen Datenaustausch vom Umfang $O(h^{-1})$ Daten über benachbarte Teilgebietskoppelränder hinweg verbunden. Von einem Vorkonditionierungsoperator C werden wir deshalb unter anderem verlangen, daß der Kommunikationsaufwand gegenüber dem nichtvorkonditionierten Fall $C = I$ nicht wesentlich erhöht wird. Akzeptiert wird eine zusätzliche globale Kommunikation von Crosspoint-Daten über den Crosspoints. Alle in dieser Arbeit vorgestellten Vorkonditionierungsoperatoren entsprechen dieser Forderung.

Im Schritt 6 des oben aufgeschriebenen Algorithmus wird der Genauigkeitstest in der $KC^{-1}K$-energetischen Norm $\| \cdot \|_{KC^{-1}K} = (KC^{-1}K\cdot, \cdot)^{0.5}$ durchgeführt. Für gute Vorkonditionierungsoperatoren C, wie wir sie im Auge haben, ist dieser Test dem in der numerischen

Praxis noch oft anzutreffenden Defektnormtest (= Test in der K^2-energetischen Norm) unbedingt vorzuziehen (vgl. dazu [14]). In der Theorie des CG-Verfahrens wird der relative Fehler jedoch in der K-energetischen Norm abgeschätzt (siehe z.B. [35]). Wenn die Spektraläquivalenzungleichungen

$$\underline{\gamma}\, C \;\leq\; K \;\leq\; \overline{\gamma}\, C \tag{3.3}$$

mit gewissen positiven Spektraläquivalenzkonstanten $\underline{\gamma}$ und $\overline{\gamma}$ gelten (was laut [17] für den DD-Vorkonditionierer erfüllt ist), dann erhält man die Fehlerabschätzung

$$\| \underline{\mathbf{u}} - \underline{\mathbf{u}}^k \|_K \;\leq\; q_k \; \| \underline{\mathbf{u}} - \underline{\mathbf{u}}^0 \|_K \tag{3.4}$$

wobei $\underline{\mathbf{u}} = \left(\underline{\mathbf{u}}_C^T \,,\, \underline{\mathbf{u}}_I^T\right)^T$ die exakte Lösung des F.E. Gleichungssystems (2.3) ist, $\underline{\mathbf{u}}^k$ bezeichnet die k-te Iterierte des CG-Algorithmus, $\| \cdot \|_K = (\cdot,\cdot)_K^{0.5} = (K\cdot,\cdot)^{0.5}$ ist die K-energetische Norm, $\xi = \underline{\gamma}\,/\,\overline{\gamma}$, $\rho = (1-\xi^{0.5})\,/\,(1+\xi^{0.5})$, und $q_k = 2\rho^k\,/\,(1+\rho^{2k})$. Folglich wird der Anfangsfehler $\underline{\mathbf{u}} - \underline{\mathbf{u}}^0$ durch $k = \sharp \ln(\varepsilon^{-1} + (\varepsilon^{-2}-1)^{0.5})\,/\,\ln(\rho^{-1})\,\sharp$ Iterationen mindestens um den Faktor ε $(0 < \varepsilon < 1)$ reduziert. Hierbei bezeichnet $\sharp x \sharp$ die kleinste ganze Zahl, die größer oder gleich x ist.

3.2 Der Vorkonditionierungsschritt im parallelisierten cg-Algorithmus

In diesem Abschnitt werfen wir einen Blick auf die Möglichkeiten, den Vorkonditionierungsschritt im cg-Algorithmus (d.h. Gleichung (3.2)) entsprechend der DD-Vorkonditionierung (2.10) aus Abschnitt 2.2 zu implementieren. Entsprechend der Notation aus Abschnitt 2.1 ergibt sich für die Vorkonditionierungsgleichung $C\underline{\mathbf{w}} = \underline{r}$ mit (2.10) das Aussehen

$$\begin{pmatrix} I_C & K_{CI}B_I^{-T} \\ O & I_I \end{pmatrix} \begin{pmatrix} C_C & O \\ O & C_I \end{pmatrix} \begin{pmatrix} I_C & O \\ B_I^{-1}K_{IC} & I_I \end{pmatrix} * \begin{pmatrix} \underline{w}_C \\ \underline{w}_I \end{pmatrix} = \begin{pmatrix} \underline{r}_C \\ \underline{r}_I \end{pmatrix}, \tag{3.5}$$

Aus (3.5) leitet sich folgender Algorithmus 1 für den Vorkonditionierungsschritt her (offensichtlich ist dieser Schritt für jeden Prozessor P_i, $i = 1, \ldots p$ gleich).

Algorithmus 1 Abk.: C=? , B=?

$$\left.\begin{array}{lrcl} 1) & \underline{\mathbf{w}}_C & = & C_C^{-1} \sum A_{C,i}^T \left(\underline{r}_{C,i} - K_{CI,i}B_{I,i}^{-T}\underline{r}_{I,i}\right) \\ 2) & \underline{\mathbf{w}}_{I,i} & = & C_{I,i}^{-1}\underline{r}_{I,i} - B_{I,i}^{-1}K_{IC,i}\underline{\mathbf{w}}_{C,i} \end{array}\right\} \tag{3.6}$$

Die Schreibweise C =? , B =? beinhaltet die freie Wählbarkeit der Operatoren $B_{I,i}$ und $C_{I,i}$. Die Operatoren C_I und C_C haben die Spektraläquivalenzungleichungen (2.11) zu erfüllen und müssen symmetrisch und positiv definit sein. Die einzige Bedingung an den Operator B_I ist seine Nichtsingularität. In allen Algorithmen wird C_C entsprechend den Vorschlägen von Bramble, Pasciak, Schatz [6] gewählt, d.h. entsprechend der Struktur von K_C in (2.6). Konkret erfolgt wiederum eine Basistransformation, diesmal in die hierarchische Basis der Crosspoints (= Grobgitterpunkte), analog zu der in (2.10). Die Transformation der reinen Koppelrandknoten auf die Crosspoints läßt sich somit über lineare Abbildungen parallel realisieren. Die Matrix der reinen Koppelrandknoten K_E wird durch eine Blockdiagonale Matrix C_E approximiert. Hierfür wurde der Vorschlag von Dryja [11]

$$C_E \;=\; \text{Blockdiag}\{C_{E,i}\}_{i=1,\dots,r} \;=\; \text{Blockdiag}\{2mat_i\,\text{Tridiag}(-1,2.-1)^{0.5}\}_{i=1,\dots,r}$$

entsprechend den Materialsprüngen an den Koppelrändern modifiziert (r = Anzahl der Koppelränder). Nach der Typumwandlung von $\underline{r}_C = (\underline{r}_V^T, \underline{r}_E^T)^T$ (Kommunikation) werden die Gleichungssysteme $C_{E,i}\underline{\mathbf{w}}_{E,i} = \underline{\mathbf{r}}_{E,i}$ parallel mittels Fouriertransformation gelöst. Das den Crosspoints entsprechende Grobgittersystem $C_V\underline{\mathbf{w}}_V = \underline{\mathbf{r}}_V$ ist global. Als Vorkonditionierer C_V für das Grobgittersystem wird die aus dem Fünfpunktdifferenzenstern entstehende Matrix verwendet. Wegen der kleinen Dimension ($<\ p$ (Prozessoranzahl)) kann diese Matrix nach einmaliger globaler Assemblierung in jedem Prozessor gespeichert werden. Somit löst jeder Prozessor das gleiche Grobgittersystem und erhält den Vektor $\underline{\mathbf{w}}_V$ der Werte aller Crosspoints, aus denen er sich die ihn betreffenden herausgreift. Das Lösen des Grobgittersystems erfolgt mittels einer Iteration bestehend aus einem lexikographisch vorwärtigen Gauß-Seidel-Iterationsschritt gefolgt von einem lexikographisch rückwärtigen Gauß-Seidel-Iterationsschritt, da bei diesem Verfahren die Grobgittermatrix nicht geändert wird. Die vorherige Kommunikation zum Erhalt des globalen Vektors $\underline{r}_V$ erfordert im Hypercube nur $\ln_2(p)$ Schritte. Für den Fall massiver Parallelisierung ($p >> 100$) ist diese Herangehensweise unbefriedigend und wird weitere Überlegungen erfordern.

Der Fall $C_I = B_I$ ändert nichts an den Voraussetzungen an den Operator C_I, spart aber einen Rechenschritt bei der Anwendung der Operatoren ein. Da dies einen nicht unerheblichen Einfluß auf die Rechenzeit des cg-Algorithmus hat, soll dieser Fall nicht unberücksichtigt bleiben.

Algorithmus 2 Abk.: C=B=?

$$\left.\begin{array}{rlcl} 1) & \underline{\mathbf{w}}_C & = & C_C^{-1} \sum A_{C,i}^T \left(\underline{r}_{C,i} - K_{CI,i}C_{I,i}^{-1}\underline{r}_{I,i}\right) \\ 2) & \underline{\mathbf{w}}_{I,i} & = & C_{I,i}^{-1}\left(\underline{r}_{I,i} - K_{IC,i}\underline{\mathbf{w}}_{C,i}\right) \end{array}\right\} \tag{3.7}$$

Wir definieren nun B_I und C_I implizit mittels geeigneter Iterationsverfahren und erhalten

$$B_I = K_I\,\left(I_I - \overline{M}_I\right)^{-1}\ ,\ B_I^T = K_I\,\left(I_I - \overset{*}{\overline{M}}_I\right)^{-1} \quad \text{und} \quad C_I = K_I\,\left(I_I - M_I\right)^{-1}, \tag{3.8}$$

wobei $\overline{M}_I$ und M_I die Iterationsoperatoren der verwendeten Iterationsverfahren sind. Mit $\overset{*}{\overline{M}}_I$ bezeichnen wir den zu $\overline{M}_I$ bzgl. des K_I-energetischen Skalarprodukts adjungierten Operator. In dieser Notation kann ein Iterationsoperator mehrere Schritte eines Iterationsverfahrens bezeichnen. Ausgehend von den Anforderungen an die Operatoren B_I und C_I müssen die Iterationsoperatoren $\overline{M}_I$ und M_I die Bedingungen (siehe [17])

1. M_I ist selbstadjungiert im K_I-energetischen Skalarprodukt,

2.
$$\mu \;=\; \rho(S_C^{-1}T_C) \;=\; \sup_{\underline{v}\in \overset{*}{V}_C \mathrm{R}^{N_C}\setminus\{O\}} \left\| \begin{pmatrix} O & O \\ O & \overline{M_I} \end{pmatrix} \underline{v} \right\|_K^2 \Big/ \parallel \underline{v} \parallel_K^2 \;=\; \left\| \begin{pmatrix} O & O \\ O & \overline{M_I} \end{pmatrix}\Big|_{\overset{*}{V}_C \mathrm{R}^{N_C}} \right\|_K^2 \;\le \overline{\eta}^2\;, \quad (3.9)$$

3.
$$\parallel M_I \parallel_{K_I} \;=\; \varrho(M_I) \;\le\; \eta < 1. \quad (3.10)$$

erfüllen. Hier und in (3.13) stellt $\overset{*}{V}_C \;=\; \begin{pmatrix} I_C \\ -K_I^{-1}K_{IC} \end{pmatrix}$ die Basistransformation in die exakte diskrete harmonische Basis dar.

Vergleicht man die Algorithmen 1 und 2, so benötigt der erste Algorithmus 3 Iterationsschritte gegenüber 2 Iterationsschritten im Algorithmus 2. Die naheliegende Frage, warum dann nicht von vornherein der zweite Algorithmus implementiert wird, läßt sich folgendermaßen beantworten: Wie aus (3.6), (3.8) und (3.9) ersichtlich, wirkt der Iterationsoperator $\overline{M}_I$ auf den Unterraum $K_I^{-1}K_{IC}\mathrm{R}^{N_I}$ des Raumes R^{N_I}. Somit besteht der Vorteil von Algorithmus 1 darin, zur Definition von B_I ein Iterationsverfahren zu verwenden, das auf diesem Unterraum schnell konvergiert. Wie man einen solchen Iterationsoperator für das Modellproblem im Modellgebiet 1 konstruieren kann wird in Abschnitt 4.2 gezeigt. Das Verhalten von $\overline{M}_I$ im gesamten Raum R^{N_I} ist nicht von Interesse, d.h. das zu $\overline{M}_I$ gehörende Iterationsverfahren muß in diesem Raum nicht notwendig konvergieren.

Natürlich wäre es erstrebenswert einen Algorithmus zu haben, der die Vorteile von Algorithmus 1 ($\overline{M}_I$ erfaßt Unterraumeigenschaften) und Algorithmus 2 (nur 2 Iterationen) miteinander verknüpft. Tatsächlich ist ein solcher Algorithmus mit Hilfe von Iterationsverfahren möglich. Seien die Operatoren B_I, B_I^T und C_I aus (3.5) mittels

$$B_I = K_I\left(I_I - \overline{M}_I\right)^{-1}\;,\; B_I^T = K_I\left(I_I - \overset{*}{\overline{M}}_I\right)^{-1} \quad \text{und} \quad C_I = K_I\left(I_I - \overline{M}_I\,\overset{*}{\overline{M}}_I\right)^{-1} \quad (3.11)$$

definiert, so ergibt sich folgender Algorithmus

Algorithmus 3	Abk.: C=C(B), B=?

$$\left.\begin{aligned} 1)\quad \underline{\tilde{\mathbf{w}}}_{I,i} &= \left(I_{I,i} - \overset{*}{\overline{M}}_{I,i}\right) K_{I,i}^{-1}\underline{r_{I,i}} \\ 2)\quad \underline{\mathbf{w}}_C &= C_C^{-1} \sum A_{C,i}^T \left(\underline{r}_{C,i} - K_{CI,i}\underline{\tilde{\mathbf{w}}}_{I,i}\right) \\ 3)\quad \underline{\mathbf{w}}_{I,i} &= \left(I_{I,i} - \overline{M}_{I,i}\right) K_{I,i}^{-1}\left(\underline{r}_{I,i} - K_{IC,i}\underline{\mathbf{w}}_C\right) + \overline{M_{I,i}}\underline{\tilde{\mathbf{w}}}_{I,i} \end{aligned}\right\} \quad (3.12)$$

Die Schreibweise $C = C(B)$ drückt die Darstellung des Operators C durch die Wahl des Operators B aus. Schritt 1 in obigem Algorithmus bedeutet die Anwendung eines Iterationsschrittes mit dem Iterationsoperator $\overset{*}{\overline{M}}_{I,i}$ auf das Gleichungssystem

$$K_{I,i}\underline{\tilde{\mathbf{w}}}_{I,i} = \underline{r}_{I,i}$$

mit der Startlösung $\underline{\tilde{\mathbf{w}}}_{I,i}^0 = 0$. In Schritt 3 hingegen wird auf das Gleichungssytem

$$K_{I,i}\underline{\mathbf{w}}_{I,i} = \underline{r}_{I,i} - K_{IC,i}\underline{\tilde{\mathbf{w}}}_{I,i}$$

das Iterationsverfahren mit dem Iterationsoperator $\overline{M}_{I,i}$ bei einer Startlösung $\underline{\mathbf{w}}_{I,i}^0 = \underline{\tilde{\mathbf{w}}}_{I,i}$ angewandt.

Da die Symmetrie von C_I in (3.11) offensichtlich ist, ergeben sich nunmehr (in Analogie zu Algorithmus 1 an den Iterationsoperator $\overline{M_I}$ folgende Anforderungen :

1.
$$\mu = \rho(S_C^{-1}T_C) = \left\| \begin{pmatrix} O & O \\ O & \overline{M_I} \end{pmatrix}\Big|_{\overset{*}{V}_C \mathrm{R}^{N_C}} \right\|_K^2 \leq \overline{\eta}^2 , \tag{3.13}$$

2.
$$\| \, \overline{M}_I \, \overset{*}{\overline{M}}_I \, \|_{K_I} = \varrho(\overline{M}_I \, \overset{*}{\overline{M}}_I) \leq \eta = \delta^2 < 1. \tag{3.14}$$

Auf den ersten Blick haben wir wegen Bedingung 2 nicht viel gegenüber Algorithmus 2 gewonnen, da jetzt der Iterationsoperator $\overline{M_I}$ der Basistransformation B_I auch ein Kontraktionsoperator auf dem Gesamtraum $\mathrm{R}^{\mathrm{N_I}}$ sein muß. Jedoch führt der Vorkonditionierer C_I nunmehr 2 Iterationen auf diesem Raum aus, so daß sich η näher an 0 als in den Algorithmen 1 und 2 ergibt. Der größte Vorteil des neuen Algorithmus besteht aber darin, daß wir nur noch ein Iterationsverfahren benötigen, welches durch einen nichtnotwendig selbstadjungierten Iterationsoperator $\overline{M_I}$ definiert wird. Weiteres dazu im Abschnitt 5.

Ein weiterer Algorithmus zur Implementation des Vorkonditionierungschrittes $C_{I,i}\underline{\mathbf{w}}_{I,i} = \underline{r}_{I,i}$ basiert auf der symmetrischen Multiplikativen Schwarzschen Methode (MSM) angewandt auf die DD-Methode. Der hierbei auftretende große Vorteil besteht, neben der Anwendbarkeit der gesamten Theorie zur MSM, in der schnellen Konvergenz der MSM und damit des CG-Algorithmus. Die Herleitung des Algorithmus, die entsprechenden theoretischen Resultate sowie einige Testrechnungen sind in [17] und [22] zu finden. Hier sei nur der entsprechende Algorithmus angegeben. Entsprechend der Herleitung in [17] und [22] ergeben sich die Operatoren B_I und C_I als

$$\left.\begin{array}{rcll} B_I & = & K_I\,(I_I - M_I\overline{M}_I)^{-1} & \text{und} \\ C_I & = & K_I\,(I_I - M_I^2)^{-1} & . \end{array}\right\} \tag{3.15}$$

Die Operatoren B_I und C_I aus (3.15) in den Algorithmus 1 (3.6) eingesetzt und deren Struktur ausgenutzt führt zu Algorithmus 4.

Algorithmus 4

$$\left.\begin{array}{lrcl}
1) & \underline{y}_{I,i} & = & (I_{I,i} - M_{I,i})\, K_{I,i}^{-1} \underline{r}_{I,i} \\
2) & \underline{\hat{w}}_{I,i} & = & B_{I,i}^{-T} \underline{r}_{I,i} \;= \overset{*}{\overline{M}}_{I,i} \underline{y}_{I,i} + (I_{I,i} - \overset{*}{\overline{M}}_{I,i})\, K_{I,i}^{-1} \underline{r}_{I,i} \\
3) & \underline{w}_{C} & = & C_C^{-1} \sum A_{C,i}^{T} \left(\underline{r}_{C,i} - K_{CI,i} \underline{\hat{w}}_{I,i}\right) \\
4) & \underline{z}_{I,i} & = & (I_{I,i} - \overline{M}_{I,i})\, K_{I,i}^{-1} \left(-K_{IC,i} \underline{w}_{C,i}\right) \\
5) & \underline{\tilde{r}}_{I,i} & = & \underline{r}_{I,i} - K_{IC,i} \underline{w}_{C,i} \\
6) & \underline{\tilde{w}}_{I,i} & = & \underline{y}_{I,i} + \underline{z}_{I,i} \\
7) & \underline{w}_{I,i} & = & M_{I,i} \underline{\tilde{w}}_{I,i} + (I_{I,i} - M_{I,i})\, K_{I,i}^{-1} \underline{\tilde{r}}_{I,i}
\end{array}\right\} \qquad (3.16)$$

Die Gleichung 1) ist gleichbedeutend mit der Anwendung eines Iterationsverfahrens, das durch den Iterationsoperator $M_{I,i}$ definiert wird, auf das Gleichungssystem

$$K_{I,i}\, \underline{y}_{I,i} \;=\; \underline{r}_{I,i} \qquad (3.17)$$

mit dem Nullvektor $\underline{y}_{I,i}^{0} = 0$ als Startvektor, während 2) die Anwendung eines durch $\overset{*}{\overline{M}}_{I,i}$ definierten Iterationsverfahrens auf die gleiche Gleichung (3.17) aber mit der Startnäherung $\underline{y}_{I,i}$ bedeutet. In Analogie dazu sind die Beziehungen 4) und 7) jeweils äquivalent zur Anwendung des Iterationsoperators $\overline{M}_{I,i}$ auf die Gleichung

$$K_{I,i}\, \underline{z}_{I,i} \;=\; -K_{IC,i} \underline{\hat{w}}_{C,i}$$

mit der Startnäherung $\underline{z}_{I,i}^{0} = 0$ bzw. von $M_{I,i}$ auf die Gleichung

$$K_{I,i}\, \underline{w}_{I,i} \;=\; \underline{\tilde{r}}_{I,i}$$

mit der Startlösung $\underline{\tilde{w}}_{I,i}$.

3.3 Verwendete Iterationstechniken

In den Beispielen und den vergleichenden Betrachtungen wurden die folgenden Iterationstechniken für das Innere der Teilgebiete verwendet (weiteres dazu in Hackbusch [24]). Die Angaben beziehen sich auf den Iterationsoperator M_I, gelten aber natürlich genauso für den Operator $\overline{M}_I$. Falls die Numerierung der Knoten einen Einfluß auf die Eigenschaften des Operators hat, ist grundsätzlich von einer zeilenweisen Numerierung auszugehen. Ein Stern über einem Operator G bedeutet stets den zu G im K_I-energetischen Skalarprodukt adjungierten Operator. Bei der Implementation dieser Operatoren wurden die Eigenschaften des Modellproblems (Poisson-Gleichung) und der Teilgebiete (äquidistant vernetzte Einheitsquadrate) ausgenutzt. Die Angaben der Gitteranzahlen l entsprechen $2^l - 1$ Knoten auf einem Koppelrandstück (d.h. Diskretisierungsparameter $h = 2^{-l}$) und damit $N_I = (2^l - 1)^2$ Freiheitsgraden im Teilgebietsinnern.

I) Multigridoperator

Der verwendete Multigridoperator wird mittels

$$\begin{aligned} M_{I,2} &= S_{I,2,post}^{\nu^{post}} \left(I_{I,2} - I_{I,1}^{2} K_{I,1}^{-1} I_{I,2}^{1} K_{I,2} \right) S_{I,2,pre}^{\nu^{pre}} \\ M_{I,q} &= S_{I,q,post}^{\nu^{post}} \left(I_{I,q} - I_{I,q-1}^{q} \left(I_{I,q-1} - M_{I,q-1}^{\gamma} \right) K_{I,q-1}^{-1} I_{I,q}^{q-1} K_{I,q} \right) S_{I,q,pre}^{\nu^{pre}} \qquad q = 3 \ldots l \end{aligned} \tag{3.18}$$

definiert (siehe auch Hackbusch [24]). Der Interpolationsoperator $I_{I,q-1}^{q}$ realisiert die lineare Interpolation entsprechend der gleichmäßigen Dreiecksvernetzung. Der entsprechende Restriktionsoperator $I_{I,q}^{q-1}$ ist so definiert, daß $I_{I,q}^{q-1} = (I_{I,q-1}^{q})^{T}$ gilt. Der Parameter γ ist im folgenden entweder 1 (V-zyklus) oder 2 (W-Zyklus). Die Vor- bzw. Nachglättungsoperatoren $S_{I,q,pre}$ und $S_{I,q,post}$ (mit ν^{pre} und ν^{post} als Anzahl der Vor- und Nachglättungen) werden wie folgt definiert.

Ia) Multigrid mit ω-Jacobi-Glätter : Die benutzte Glättung wird über

$$S_{I,q,pre} = \overset{*}{S}_{I,q,post} = I_{I,q} - \omega D_{I,q}^{-1} K_{I,q}, \tag{3.19}$$

mit $D_{I,q} = diag(K_{I,q})$ definiert, wobei der Parameter ω variiert werden wird (üblicherweise ist $\omega = 2/3$).

Abkürzung : V12(ω) bedeutet einen V-Zyklus mit einem Vor- und zwei Nachglättungsschritten mittels der ω-Jacobi-Iteration.

Ib) Multigrid mit Gauß-Seidel-Glätter : Bei diesem Glätter ist die Numerierung von Bedeutung. Der lexikographisch vorwärts arbeitende Gauß-Seidel-Glätter, angewandt auf die Gleichung $K_{I,q} x_{I,q} = f_{I,q}$, wird mittels

$$(I_{I,q} + D_{I,q}^{-1} L_{I,q}) x_{I,q}^{i+1} = (I_{I,q} - D_{I,q}^{-1} U_{I,q}) x_{I,q}^{i} + D_{I,q}^{-1} f_{I,q} \tag{3.20}$$

erklärt, wobei $L_{I,q}$, $U_{I,q}$ die untere bzw. obere Dreiecksmatrix (ohne Diagonale) und $D_{I,q}$ die Digonale von $K_{I,q}$ darstellen. Der dazu adjungierte Operator ist der lexikographisch rückwärts arbeitende Gauß-Seidel-Glätter.

Abkürzung : W21↑↑↓ bedeutet einen W-Zyklus mit zwei lexikographisch vorwärtigen Gauß-Seidel-Iterationen als Vorglätter und einer lexikographisch rückwärtigen Gauß-Seidel-Iteration als Nachglätter.

II) Zweigitteroperator

Dieser Operator dient vorrangig zur Analyse, um theoretische Ergebnisse zu erzielen. Sein Aussehen

$$M_{I,l} = S_{I,l,post}^{\nu^{post}} \left(I_{I,l} - I_{I,l-1}^{q} K_{I,l-1}^{-1} I_{I,l}^{l-1} K_{I,l} \right) S_{I,l,pre}^{\nu^{pre}} \tag{3.21}$$

ist dem Aussehen des Mehrgitteroperators für das zweitgröbste Gitter aus (3.18) äquivalent. Die Komponenten sind die gleichen wie beim Mehrgitteroperator, jedoch wird als Glätter stets der ω-Jacobi-Glätter benutzt. Die Implementation für Testzwecke erfolgte derart, daß anstelle des exakten Lösens des Grobgittersystems $K_{I,l-1} x_{I,l-1} = f_{I,l-1}$ 3 W22↑↑↓↓-Iterationen auf dem $(l-1)$-ten Gitter angewandt wurden, was de facto einem exakten Lösen entspricht.

Abkürzung : TG11(ω) bedeutet eine Zweigitteriteration mit je einer ω-Jacobi-Iteration als Vor- bzw. Nachglättung.

III) MAF

Dieses Iterationsverfahren gehört zur großen Gruppe der unvollständigen Faktorisierungen. Die Grundidee besteht darin, zu einer gegebenen Matrix $K_I = L_I + D_I + U_I$ eine Diagonalmatrix T_I zu finden, so daß für einen Testvektor $\underline{e}$ folgende Gleichung erfüllt ist

$$(T_I + L_I)T_I^{-1}(T_I + U_I)\,\underline{e} \;=\; K_I\underline{e}. \tag{3.22}$$

In (3.22) bezeichnen L_I und U_I die untere bzw. obere Dreiecksmatrix, sowie D_I die Diagonale von K_I. Dies bedeutet nichts anderes, als daß der Basistransformationsoperator B_I über

$$B_I \;=\; (T_I + L_I)T_I^{-1}(T_I + U_I) \tag{3.23}$$

ausgedrückt wird. Die Diagonalmatrix T_I läßt sich über die Rekursionsformel

$$T_I\underline{e} \;=\; D_I\underline{e} - L_I D_I^{-1} U_I\underline{e} \tag{3.24}$$

bestimmen. Näheres dazu kann in [30] nachgelesen werden. In den Beispielrechnungen wird stets $\underline{e} = \mathbf{1}$ als Testvektor verwendet.
Abkürzung : MAF

IV) Frequenzfiltermethode

Dieses von G. Wittum [38] entwickelte Iterationsverfahren gehört ebenfalls zur Gruppe der unvollständigen Faktorisierungen. Im Unterschied zu obiger MAF wird von der, durch die zeilenweise Numerierung entstandenen, tridiagonalen Blockstruktur der Matrix K_I ausgegangen wobei die Dimension der Blöcke gleich $nx - 1$ (nx - Unterteilungsanzahl in x-Richtung) ist. Die Diagonalblöcke $D_{I,j}$ von K_I haben eine tridiagonale Struktur, während die Nebendiagonalblöcke $L_{I,j}$ und $U_{I,j} = L_{I,j}^T$ Diagonalmatrizen sind ($j = 1 \ldots ny$, mit ny ist Anzahl der Unterteilungen in y-Richtung). Gesucht wird nun eine Blockdiagonalmatrix T_I deren einzelne Blöcke $T_{I,j}$ tridiagonal sein sollen und welche die Gleichung (3.22) für das Testvektorenpaar (die Vektoren haben die gleiche Blockstruktur wie die Matrizen) $\underline{e}_1 = \{\underline{e}^{\nu_1}\}_{j=1\ldots ny}$ und $\underline{e}_2 = \{\underline{e}^{\nu_2}\}_{j=1\ldots ny}$ erfüllt. Damit die Blöcke der Testvektoren $\underline{e}_k$ alle gleich gewählt werden können, müssen die Blocktestvektoren $\underline{e}^{\nu_k}$ ($k = 1, 2$) die Eigenvektoren der Matrix K_I in x-Richtung sein. Somit werden die Testvektoren über

$$\underline{e}_i^{\nu_k} \;=\; \sin(\nu_k \pi \frac{i}{nx}) \qquad i = 1 \ldots nx - 1\,, \quad k = 1, 2 \tag{3.25}$$

berechnet. Der oft verwendete Faktor $\sqrt{(2/nx)}$ vor den Eigenfunktionen kann hier ohne weiteres weggelassen werden, da er in (3.22) rausfällt. Die Parameter ν_k ($k = 1, 2$) werden als Testfrequenzen bezeichnet. Die entscheidende Idee ist nunmehr, daß man sich nicht auf ein einziges Paar solcher Testfrequenzen beschränkt, sondern obige Bestimmung der Blockmatrix

T_I für mehrere Testfrequenzenpaare durchführt. Hierbei wird stets $\nu_2 = \nu_1 + 1$ gewählt, weshalb im weiteren die Notation $\nu := \nu_1$ das Paar (ν_1, ν_2) repräsentieren soll. Die durch ν eindeutig bestimmte Matrix $T_{I,\nu}$ definiert ihrerseits den Iterationsoperator

$$S_{I,\nu} := I_I - (U_I + T_{I,\nu})^{-1} T_{I,\nu} (L_I + T_{I,\nu})^{-1} K_I. \tag{3.26}$$

Für den Fall eines Testfrequenzpaares gilt $B_I = K_I(I_I - S_{I,\nu})^{-1}$ und es ergibt sich Gleichung (3.23). Bei mehreren Testfrequenzpaaren müssen die entsprechenden Iterationsoperatoren $S_{I,\nu}$ hintereinander angewandt werden. Wenn man die geordnete Indexmenge der Testfrequenzenpaare ν mit ω_ν bezeichnet, läßt sich der gesuchte Iterationsoperator M_I als

$$M_I = \prod_{\nu \in \omega_\nu} S_{I,\nu} \quad , \tag{3.27}$$

mit $S_{I,\nu}$ aus (3.26) schreiben. Nach Wittum [38] können die Testfrequenzen der Indexmenge ω_ν wie folgt gewählt werden :

$$\left.\begin{array}{rcll} \nu^1 &=& 1 & \\ \nu^i &=& \nu^{i-1} + 2 \quad \text{falls } [\alpha\nu^{i-1}] \le \nu^{i-1} & \qquad i < nx \\ &=& [\alpha\nu^{i-1}] \quad \text{sonst.} & \end{array}\right\} \tag{3.28}$$

Üblicherweise wird $\alpha = 2$ gewählt, jedoch sind auch andere Parameter möglich. Bei den Testrechnungen wurde α zwischen 2 und 4 gewählt, bzw. nur einige Testfrequenzen aus der in (3.28) entstehenden Indexmenge ω_ν benutzt. Der zu M_I (im K_I-energetischen Skalarprodukt) adjungierte Operator $\overset{*}{M}_I$ ergibt sich, wenn man anstelle der zeilenweisen eine spaltenweise Numerierung benutzt und die geordnete Indexmenge ω_ν in umgekehrter Reihenfolge in (3.27) anwendet. In den hier auftretenden Testbeispielen wurde stets das Einheitsquadrat als Teilgebiet gewählt, so daß $nx = ny$ gilt.

Abkürzungen : FFM($\alpha = 3$) bedeutet die Anwendung des Iterationsoperators M_I aus (3.27), wobei die Testfrequenzen aus (3.28) mit $\alpha = 3$ gewählt wurden. FFM(1,3,6) bedeutet, daß nur die Testfrequenzen $(1, 3, 6)$ als Indexmenge ω_ν benutzt werden. $F\overset{*}{F}M(1,3,6)$ stellt den zu $FFM(1,3,6)$ adjungierten Operator dar (d.h. spaltenweise numeriert und umgekehrte Reihenfolge der Testfrequenzen).

4 Der Basistransformationsoperator

4.1 Der Einfluß der Basistransformation auf die Konvergenz des cg-Verfahrens

Die Konditionszahl $\kappa(C^{-1}K)$ aus (2.12) verhält sich im Falle exakter Basistransformation und exaktem inneren Löser (d.h. $B_I = C_I = K_I$) mit der Wahl für C_C nach Bramble, Pasciak, Schatz [6] wie $O(\,(1 + \log(H/h))^2$ mit $H \approx aver.diam(\Omega_i)\,)$. Der innere Löser C_I kann mittels einer geeigneten Technik (z.B. Multigrid, Frequenzfiltermethode) so gewählt werden, daß die Spektraläquivalenzkonstanten $\overline{\gamma}_I$ und $\underline{\gamma}_I$ aus (2.11) unabhängig vom Diskretisierungsparameter h sind. Somit kann eine weitere h-Abhängigkeit der Konditionszahl $\kappa(C^{-1}K)$ nur noch durch die Basistransformationsmatrix B_I hineinkommen.

Um den Einfluß der Basistransformation auf die Konditionszahl $\kappa(C^{-1}K)$ zu dokumentieren, wurden die folgenden Testrechnungen mit den Algorithmen 1 (3.6) und 2 (3.7) durchgeführt. Der cg-Algorithmus stoppte bei einem relativen Fehler kleiner als $1 * 10^{-6}$, gemessen in der $\|\cdot\|_{KC^{-1}K}$-Norm. Als Testbeispiel diente das Modellproblem (2.1) im Modellgebiet 2 mit 16 Teilgebieten.

Gitter l	C=B B=3W22	C=V11 B=3W22	C=V11 B=W22	C=V11 B=W11	C=B B=V11	C=V11 B=V22	C=B B=V22
3	11	11	12	14	14	12	12
4	11	12	13	15	18	14	13
5	12	13	14	19	24	15	15
6	14	15	16	25	33	18	18
7	15	16	18	33	52	22	21

Tabelle 1 Modellgebiet 2 mit 16 Prozessoren; Vergleich der Iterationszahlen für verschiedene Multigridoperatoren (GS↑↓-Glättung)

Die erste Spalte in Tabelle 1 zeigt das Verhalten der Iterationszahlen bei de facto exakter Basistransformation B_I und exaktem inneren Löser C_I. Diese Ergebnisse stimmen mit den Ergebnissen in [6] überein. An Hand der zweiten Spalte, wo ein ganz einfacher Multigridzyklus als innerer Löser C_I verwendet wurde, wird die untergeordnete Bedeutung von C_I für die Konvergenzrate $\kappa(C^{-1}K)$ deutlich. Daher wird in den restlichen Spalten (außer der letzten Spalte) die Basistransformation B_I variiert. Aus den Spalten 3 und 6 geht hervor, daß ein einfacher Multigridschritt mit je 2 Vor- und Nachglättungsschritten leicht erhöhte Iterationszahlen liefert. Jedoch steigen die Iterationszahlen in den Spalten 4 und 5 bei nur einem Vor- und einem Nachglättungsschritt sehr stark an, und lassen eine h-Abhängigkeit deutlich zu Tage treten. Die letzte Spalte dient zur Komplettierung in Bezug auf die Rechenzeiten, welche in Tabelle 2 zu finden sind.

Gitter l	C=B B=3W22	C=V11 B=3W22	C=V11 B=W22	C=V11 B=W11	C=B B=V11	C=V11 B=V22	C=B B=V22
3	3.6	3.6	3.5	3.9	3.8	3.4	3.3
4	6.4	7.8	5.5	5.9	6.3	5.9	4.9
5	19.5	22.4	13.5	15.8	15.4	12.8	11.1
6	81.6	103.9	48.9	62.9	57.6	49.5	36.2
7	341.5	391.0	204.6	301.1	210.3	203.4	157.8

Tabelle 2 Modellgebiet 2 mit 16 Prozessoren; Vergleich der Rechenzeiten in Sekunden für verschiedene Multigridoperatoren (GS↑↓-Glättung)

Auf Grund der geringen Iterationszahlen und des geringen arithmetischen Kostenaufwandes für die Basistransformation sind die Verfahren der Spalten 3 und 6 am schnellsten bei Implementation von Algorithmus 1 (3.6). Wie aus der letzten Spalte ersichtlich, ist Algorithmus 2 (3.7) durch den großen Vorteil des Wegfalls einer Multigriditeration pro cg-Schritt nochmals schneller. Inwieweit Algorithmus 3 (3.12) eine weitere Beschleunigung bringen kann, wird in Abschnitt 5 ersichtlich. Tabelle 1 zeigt, daß eine Verbesserung der Konditionszahlen und eventuell der Rechenzeiten über den Basistransformationsoperator B_I zu erfolgen hat. Der nächste Abschnitt versucht dazu einige Wege aufzuzeigen.

4.2 Zur Konstruktion von Basistransformationsoperatoren

Wie schon aus der Abschätzung (2.12) für die Konditionszahl des vorkonditionierten Operators $C^{-1}K$ (C aus (2.10)) ersichtlich, geht die Basistransformation B_I über den Parameter $\mu = \varrho(S_C^{-1}T_C)$ darin ein. Das bedeutet, daß $T_C = K_{CI}(K_I^{-1} - B_I^{-T})K_I(K_I^{-1} - B_I^{-1})K_{IC}$ möglichst nahe dem Nulloperator sein sollte. Dies wird erreicht, wenn

$$B_I^{-1}K_{IC}\underline{u}_C \;\approx\; K_I^{-1}K_{IC}\underline{u}_C \qquad \forall \underline{u}_C \in \mathrm{R}^{\mathrm{N_C}} \tag{4.1}$$

erfüllt ist, d.h. B_I muß K_I auf dem Unterraum $K_{IC}\mathrm{R}^{\mathrm{N_C}}$ gut approximieren. Eine Idee wäre die Konstruktion einer unvollständigen Faktorisierung analog zur Frequenzfiltermethode (siehe Abschnitt 3.3 und Wittum [38]) mit geeignet gewählten Testvektoren aus dem Unterraum $K_{IC}\mathrm{R}^{\mathrm{N_C}}$. Dazu liegen noch keine Erfahrungen vor, jedoch sind die Ergebnisse von Tabelle 4 aus Kapitel 5 ermutigend.

Wird der in (3.8) definierte Basistransformationsoperator B_I in die Definitionsgleichung für $\mu = \varrho(S_C^{-1}T_C)$ eingesetzt, ergibt sich

$$\mu \;=\; \sup_{\underline{u}_C\in\mathrm{R}^{\mathrm{N_C}}\setminus\mathrm{O}} \frac{\|\,\overline{M}_I K_I^{-1}K_{IC}\,\underline{u}_C\,\|_{K_I}^2}{\|\,\underline{u}_C\,\|_{S_C}^2} \;=\; \sup_{\underline{u}_C\in\mathrm{R}^{\mathrm{N_C}}\setminus\mathrm{O}} \frac{\sum\limits_{i=1}^{p}\|\,\overline{M}_{I,i}K_{I,i}^{-1}K_{IC,i}\,\underline{u}_C\,\|_{K_{I,i}}^2}{\|\,\underline{u}_C\,\|_{S_C}^2}\;. \tag{4.2}$$

Führt man die neue Größe

$$\delta_i \;:=\; \sup_{\underline{u}_C\in\mathrm{R}^{\mathrm{N_C}}\setminus\mathrm{O}} \frac{\|\,\overline{M}_{I,i}K_{I,i}^{-1}K_{IC,i}\,\underline{u}_C\,\|_{K_{I,i}}^2}{\|\,\underline{u}_C\,\|_{S_C}^2} \tag{4.3}$$

ein, läßt sich μ mit

$$\mu \;\le\; \sum_{i=1}^{p}\delta_i \tag{4.4}$$

abschätzen, da δ_i in jedem Teilgebiet i ($i = 1\ldots p$) berechnet werden kann.

Zum Zwecke der Abschätzung von δ_i aus (4.3) für das Modellproblem im Modellgebiet 1 führen wir eine Fourierreihenentwicklung der in (4.3) auftretenden Ausdrücke durch. Die verwendeten Indizes k und i entsprechen der x-Richtung, die Indizes l und j der y-Richtung

($n = nx = ny = 1/h$ - Anzahl der Unterteilungen). Aus den Eigenvektoren $\mu_k(i)$ und Eigenwerten λ_k des Drei-Punkte-Differenzenoperators der 1D-Laplace-Gleichung

$$\mu_k(i) \;=\; \sqrt{2h}\sin(k\pi ih) \quad \text{und} \quad \lambda_k \;=\; 4\left(\sin(k\frac{h}{2}\pi)\right)^2 \tag{4.5}$$

ergeben sich die Eigenwerte und Eigenvektoren der Matrix K_I als

$$\mu_{k,l}(i,j) \;=\; \mu_k(i)\mu_l(j) \quad \text{und} \quad \lambda_{k,l} \;=\; \lambda_k + \lambda_l \,. \tag{4.6}$$

Im weiteren bezeichnen Eigenwerte ohne Angabe der zugehörigen Matrix stets die in (4.5) und (4.6) eingeführten Eigenwerte. Die Fourierreihenentwicklung für $\underline{u}_C$ (Koppelrand in y-Richtung) hat das Aussehen

$$u_C(j) \;=\; \sum_{l=1}^{n-1} \alpha_l\,\mu_l(j) \qquad \text{mit} \qquad \alpha_k \;=\; \left\langle \underline{u}_C \,,\, \underline{\mu_l} \right\rangle \;=\; \sum_{j=1}^{n-1} u_C(j)\mu_l(j)\,. \tag{4.7}$$

Da die, zu den Eigenvektoren in (4.5) gehörenden, Eigenwerte $\lambda_l(S_C)$ von S_C für $h \longrightarrow 0$ gegen

$$\sqrt{\lambda_l^2 + 4\lambda_l} \tag{4.8}$$

konvergieren, ist dies eine hinreichende Näherung für die Eigenwerte des Schur-Komlements (siehe auch Golub [16]). Der Nenner in (4.3) läßt sich mit Hilfe von (4.7) und (4.8) als

$$\|\, \underline{u}_C \,\|_{S_C}^2 = \langle S_C\underline{u}_C \,,\, \underline{u}_C\rangle \;\approx\; \sum_{l=1}^{n-1} \sqrt{\lambda_l^2 + 4\lambda_l}\;\alpha_l^2 \tag{4.9}$$

schreiben.

Zur Entwicklung des Zählers in (4.3) betrachten wir zuerst die Fourierreihenentwicklung eines Vektors $\underline{y}_I$ des Unterraumes $K_{IC}\mathrm{R}^{\mathrm{N_C}}$. Verwendet werden nur Gleichung (4.7) und die Struktur von K_{IC}.

$$\begin{aligned}
\underline{y}_I \;=\; K_{IC}\underline{u}_C \;&=\; y_I(i,j) \;=\; \begin{cases} \;\;0 & i = \overline{1, n-2} \quad j = \overline{1, n-1} \\ -u_j & i = n-1 \quad\;\; j = \overline{1, n-1} \end{cases} \\
&=\; \sum_{l=1}^{n-1}\sum_{k=1}^{n-1} \beta_{k,l}\,\mu_{k,l}(i,j) \\
&=\; \sum_{l=1}^{n-1}\sum_{k=1}^{n-1} \overbrace{\left[\sum_{j=1}^{n-1} \underbrace{\sum_{i=1}^{n-1} y_I(i,j)\,\mu_k(i)}_{y_I(n-1,j)\mu_k(n-1) \,=\, -u_j\mu_k(n-1)} \mu_l(j)\right]}^{\beta_{k,l} = \left\langle \underline{y}_I \,,\, \mu_{k,l}(i,j)\right\rangle} \mu_{k,l}(i,j)
\end{aligned}$$

$$= \sum_{l=1}^{n-1}\sum_{k=1}^{n-1}\left[-\mu_k(n-1)\underbrace{\sum_{j=1}^{n-1} u_j\,\mu_l(j)}_{\alpha_l}\right]\mu_{k,l}(i,j)$$

$$\underline{y}_I \;=\; K_{IC}\underline{u}_C \;=\; \sum_{l=1}^{n-1}\sum_{k=1}^{n-1}\left(-\mu_k(n-1)\,\alpha_l\right)\,\mu_{k,l}(i,j) \qquad . \tag{4.10}$$

Entsprechend (4.6) und (4.10) ergibt sich

$$\underline{z}_I \;=\; K_I^{-1}K_{IC}\underline{u}_C \;=\; K_I^{-1}\underline{y}_I \;=\; \sum_{l=1}^{n-1}\sum_{k=1}^{n-1}\left(\frac{-\mu_k(n-1)}{\lambda_{k,l}}\,\alpha_l\right)\mu_{k,l}(i,j) \quad , \tag{4.11}$$

und der Zähler in (4.3) läßt sich als

$$\|\,\overline{M}_I\underline{z}_I\,\|^2_{K_I} \;=\; \langle K_I M_I\underline{z}_I\,,\,M_I\underline{z}_I\rangle \;=\; \sum_{l=1}^{n-1}\sum_{k=1}^{n-1}\varrho^2_{k,l}(\overline{M}_I)\,\frac{\mu_k^2(n-1)}{\lambda_{k,l}}\,\alpha_l^2 \tag{4.12}$$

schreiben. Die Größe $\varrho_{k,l}(\overline{M}_I)$ bezeichne den Dämpfungsfaktor des Operators $\overline{M}_I$ bezüglich des Eigenvektors $\mu_{k,l}$, d.h. bezüglich der Frequenz (k,l) des Fehlerspektrums. Die Gleichungen (4.9) und (4.12) werden in (4.3) eingesetzt und mit Hilfe der Beziehung

$$\max_{\underline{\alpha}}\frac{\sum\limits_{l=1}^{n-1} a_l\,\alpha_l^2}{\sum\limits_{l=1}^{n-1} b_l\,\alpha_l^2} \;\le\; \max_{l=\overline{1,n-1}}\frac{a_l}{b_l}$$

abgeschätzt. Dadurch erhält man folgende δ_i-Abschätzung für das Modellbeispiel 1 :

$$\delta_i \;\le\; \max_{l=\overline{1,n-1}}\sum_{k=1}^{n-1}\varrho^2_{k,l}(\overline{M}_I)\,\underbrace{\frac{\mu_k^2(n-1)}{\lambda_{k,l}}\,\frac{1}{\sqrt{\lambda_l^2+4\lambda_l}}}_{fak_{k,l}} \;=\; \overline{\delta} \tag{4.13}$$

Der in (4.13) eingeführte Faktor $fak_{k,l}$ bleibt für jede Wahl des Operators $\overline{M}_I$ gleich. Um die Anforderungen an die Dämpfungsfaktoren $\varrho_{k,l}(\overline{M}_I)$ des Operators $\overline{M}_I$ (und damit auch die Forderungen an den Operator) angeben zu können, wählen wir zuerst $\overline{M}_I = I_I$ und damit $\varrho_{k,l}(\overline{M}_I) = 1 \;\; \forall k,l = \overline{1,n-1}$. Der dadurch für $\overline{\delta}$ entstandene Ausdruck wird mit Hilfe der Beziehung

$$\mu_k^2(n-1) \;=\; \mu_k^2(1) \;=\; 8h\sin^2(k\pi\frac{h}{2})\cos^2(k\pi\frac{h}{2}) \;=\; \frac{h}{2}\,\lambda_k\,(4-\lambda_k)\;,$$

zu

$$\overline{\delta}(I_I) \;=\; \max_{l=\overline{1,n-1}}\sum_{k=1}^{n-1}\frac{\frac{h}{2}\,\lambda_k\,(4-\lambda_k)}{(\lambda_k+\lambda_l)\sqrt{\lambda_l^2+4\lambda_l}} \;=\; \sum_{k=1}^{n-1}\frac{\frac{h}{2}\,\lambda_k\,(4-\lambda_k)}{(\lambda_k+\lambda_1)\sqrt{\lambda_1^2+4\lambda_1}} \tag{4.14}$$

umgeformt, da die Summe für $l = 1$ offensichlich ihr Maximum annimmt. Somit muß ein Operator $\overline{M}_I$ insbesondere die in l-Richtung kleinen Frequenzen reduzieren. Genau diese Eigenschaft hat die am Anfang des Abschnittes erwähnte Frequenzfiltermethode bei einer Numerierung parallel zu Koppelrand. Die entsprechenden numerischen Erfahrungen sind in Kapitel 5 zu finden. Darüber hinaus wird diese Möglichkeit Gegenstand künftiger Untersuchungen sein.

Eine zusätzliche Variante wäre, daß man versucht, die Dämpfungsfaktoren $\varrho_{k,l=1}$ zu verbessern. Zur analytischen Konstruktion eines entsprechenden Operators wählen wir zunächst den ω-Jacobi-Iterationsoperator aus Abschnitt 3.3 (3.19)

$$\begin{aligned} \overline{S}_I &= I_I - \omega D_I^{-1} K_I \qquad \text{mit} \\ \varrho_{k,l} &= |\lambda_{k,l}(\overline{S}_I)| \;=\; |1 - \frac{\omega}{4}(\lambda_k + \lambda_l)| \end{aligned} \tag{4.15}$$

als Dämpfungsfaktor (= Betrag des Eigenwertes von $\overline{S}_I$). Betrachtet man die interessierenden Dämpfungsfaktoren, ergibt sich der optimale Relaxationsparameter ω

$$\omega_{opt} \;=\; \omega(k) \;=\; \frac{4}{\lambda_k + \lambda_1} \tag{4.16}$$

in Abhängigkeit der Frequenz in k-Richtung. Offensichtlich ist ω_{opt} stets ein Überrelaxationsparameter ($\omega(n-1) = 1 \leq \omega_{opt}$).

Interessant werden diese Überlegungen beim Einsatz des Zweigitterverfahrens mit ω-Jacobi-Glättung (sieh Abschnitt 3.3). Eine Analyse der interessierenden Dämpfungsfaktoren $\varrho_{k,1}$ für das Zweigitterverfahren (siehe W. Hackbusch [23]) zeigt deren monotonen Anstieg mit k. Dem steht das monotone Fallen von $fak_{k,1}$ aus (4.13) gegenüber, so daß das Maximum der Summanden in (4.13) in der Mitte des Frequenzspektrums von k zu erwarten ist. In numerischen Tests wurde $\omega_{opt} \approx \omega(0.6n) \approx 1.53$ ermittelt.

Gitter	3	4	5	6	7
B=TG11($\omega = 2/3$)	8	9	12	14	17
B=TG11($\omega = \omega(0.6n)$)	8	10	10	10	10
B=V11($\omega = 2/3$)	9	10	14	19	27
B=V11($\omega = \omega(0.6n)$)	9	10	11	11	11

Tabelle 3 Einfluß des Relaxationsparameters ω auf die Konvergenz des cg-Algorithmus im Modellgebiet 1, Alg. 1 mit C=V11$\uparrow\downarrow$

Tabelle 3 zeigt deutlich den Unterschied zwischen der Verwendung eines Operators $\overline{M}_I$ der im Raum $\mathrm{R}^{\mathrm{N_I}}$ Kontraktionsoperator ist ($\omega = 2/3$), und einem dem Raum $K_{IC}\mathrm{R}^{\mathrm{N_I}}$ angepassten Operator ($\omega = \omega(0.6n)$) in der Basistransformation B_I. Besonders bedeutsam ist der große Gewinn an Konvergenzgeschwindigkeit bei Verwendung eines Multigrid-V-Zyklus, da dieser arithmetisch wesentlich billiger ist als das Zweigitterverfahren. Wie sofort ersichtlich, ist der Operator $\overline{M}_I$ auf dem gesamten Raum $\mathrm{R}^{\mathrm{N_I}}$ kein Kontraktionsoperator mehr, da die Fehleranteile der Frequenzen (k,l) mit $\lambda_{k,l} > \frac{4}{\omega(0.6n)} \approx 2.61$ sogar vergrößert werden. Demzufolge wird Algorithmus 3 mit diesen unterraumangepassten Operator B_I keine Konvergenzverbesserung des cg-Algorithmus erzielen können. Im Gegenteil, einige numerische Tests zeigten ein drastisches Ansteigen der cg-Iterationen.

Mit diesem Ausblick (bei Modellgebiet 2 mit 16 Teilgebieten wird es schwerer !) bezüglich der Konstruktion unterraumangepasster Basistransformationsoperatoren B_I für Algorithmus 1 schließt Abschnitt 4.

5 Diskussion weiterer numerischer Ergebnisse

Alle Beispiele beziehen sich auf das Modellproblem (2.1) im jeweiligen Modellgebiet. Verwendet wurde der in Abschnitt 3 vorgestellte cg-Agorithmus, das Abbruchkriterium ist ein in der $KC^{-1}K$-Norm gemessener relativer Fehler $\leq 10^{-6}$. Falls nicht anders angegeben, gilt $mat(x) = 1\ \forall x \in \Omega$. Die Anzahl der verwendeten Prozessoren entspricht stets der Anzahl der Teilgebiete.

Modellgebiet 1 : Die Frequenzfiltermethode in der Basistransformation

In diesem Beispiel wird der Basistransformationsoperator B_I mittels der Frequenzfiltermethode aus Abschnitt 3.3 ausgedrückt. Da der Koppelrand in y-Richtung liegt, wird die Methode mit spaltenweiser Numerierung verwendet. Zum Einsatz kamen die Algorithmen 1 und 2.

	C=FFM($\alpha=2$)$\cdot$ $\overset{*}{FFM}(\alpha=2)$			C=C(B)	
l	B=$\overset{*}{FFM}$(1,3,6)	B=$\overset{*}{FFM}$(1,3)	B=$\overset{*}{FFM}$(1)	B=$\overset{*}{FFM}$(1,3,6)	B=$\overset{*}{FFM}$(1,3)
3	5 0.05	5 0.019	5 0.034	5 0.071	5 0.026
4	5 0.045	5 0.045	6 0.076	5 0.045	5 0.054
5	4 0.028	5 0.040	7 0.115	4 0.031	6 0.093
6	4 0.022	5 0.046	9 0.207	5 0.057	8 0.173
7	4 0.024	5 0.063	12 0.303	7 0.128	12 0.292

Tabelle 4 Modellgebiet 1 mit 2 Prozessoren; Iterationszahlen und Konvergenzraten; Variierung der Testvektoren in der Frequenzfiltermethode als Basistransformation

Diese Rechnungen bestätigen die Vermutung aus Abschnitt 4.2, daß der Basistransformationsoperator B_I vor allem die niedrigen Frequenzen in x-Richtung dämpfen muß. Wie man deutlich erkennt, reichen schon drei Testfrequenzpaare für (im betrachteten Bereich) konstante Konvergenzraten aus. Auch zwei Testfrequenzpaare liefern nur ein leichtes Ansteigen der Konvergenzrate, wohingegen nur ein Testfrequenzpaar offensichtlich zu wenig ist. Die ersten 3 Spalten enthalten als inneren Löser C_I einen de facto exakten Löser, was natürlich arithmetisch teuer ist. Interessant werden die Untersuchungen bei Verwendung von Algorithmus 3 (3.12). Schaut man sich die beiden rechten Spalten an, ist ein Anstieg der Konvergenzrate mit wachsendem Diskretisierungsparameter $h = 2^{-l}$ zu verzeichnen. Der Grund dafür liegt in der Tatsache, daß das Frequenzfilterverfahren in einem bestimmten Frequenzstreifenbereich eines Testfrequenzpaares gute Fehlerdämpfungen erzielt, während im restlichen Bereich der Dämfungsfaktor nahe 1 ist. Da in Algorithmus 3 der Iterationoperator $\overline{M}_I$ der Basistransformation B_I auch als Iterationsoperator im gesamten Raum $\mathrm{R}^{\mathrm{N}_\mathrm{I}}$ fungiert, macht sich die schlechte Dämfung der hohen Frequenzen bemerkbar. Mit wachsendem h steigt die Anzahl der schlecht gedämpften Frequenzen und somit die Konvergenzrate des gesamten Verfahrens.

Modellgebiet 2 : Springende Materialkoeffizienten

Dieses Beispiel vergleicht den Fall springender, aber im Teilgebiet konstanter, Materialkoeffizienten mit dem bisher behandelten Fall konstanter Koeffizienten im gesamten Gebiet. Die Materialkoeffizienten in Bild 3 wurden nach einem Beispiel von Bramble et al [6] gewählt.

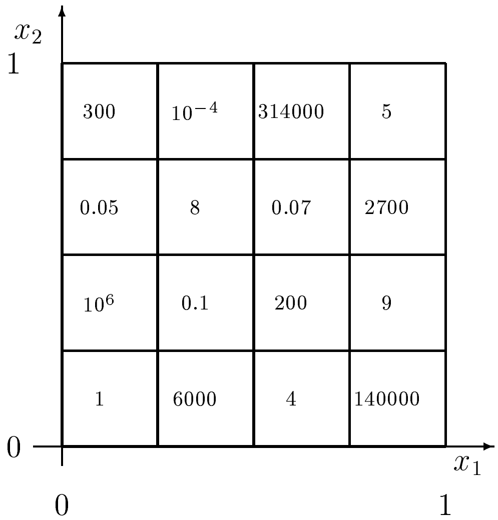


Abbildung 3: Materialkoeffizienten (Bramble-Bsp.) im Modellgebiet 2

Die notwendige Koeffizientenmittlung erfolgte über die Formel

$$mat(x_C) \;=\; \frac{1}{\#\{i \,:\, x_C \in \overline{\Omega}_i\}} \sum_{i\,:\,x_C \in \overline{\Omega}_i} mat(\Omega_i) \qquad i = \overline{1,p}\,, \tag{5.1}$$

d.h. die Summe der Materialkennzahlen angrenzender Teilgebiete geteilt durch die Anzahl der angrenzenden Teilgebiete. Die Ergebnisse sind in Tabelle 5 zusammengefaßt. Algorithmus 3 wurde verwendet.

Gitter	3	4	5	6	7
B=3W22↑↑↓↓	10	12	14	16	18
mat → Bild 3	0.24	0.31	0.37	0.42	0.46
B=V22↑↑↓↓	11	13	15	17	21
mat → Bild 3	0.26	0.33	0.39	0.44	0.51
B=V22↑↑↓↓	12	13	15	18	21
mat ≡ 1	0.31	0.33	0.37	0.43	0.51

cg-Iterationen
durchschnittl. cg-Rate

Tabelle 5 Modellgebiet 4 mit 16 Prozessoren; Iterationszahlen und Konvergenzraten; Springende Koeffizienten; $C = C(B)$

Ein Vergleich der zweiten und dritten Zeile zeigt, daß die großen Materialsprünge keine Verschlechterung der Iterationsraten bewirken. Die Rechenzeit für das Bsp. $B = V22 \uparrow\uparrow\downarrow\downarrow$ ist in Tabelle 7 Zeile 1 ersichtlich. Im Falle komplizierterer Teilgebiete muß in Formel (5.1) die Wichtung der einzelnen Materialkoeffizienten verändert werden, dazu dürfte die Einbeziehung des Teilgebietswinkels am Crosspoint ausreichen. Für die einfachen Koppelränder hat Formel (5.1) weiterhin Bestand. Zeile 1 dient dem Vergleich mit dem Originalbeispiel von Bramble [6], da dieser direkte Löser verwendete.

Modellgebiet 2 : Variierung des Parameters α in der Frequenzfiltermethode als Basistransformation

Wie schon in Abschnitt 3.3 erwähnt, kann der Multiplikator α zur Wahl der Testfrequenzenpaare variiert werden. Da die Frequenzfiltermethode nunmehr auf dem Unterraum $K_{IC}\mathrm{R}^{\mathrm{N_I}}$ wirken soll, kann eine anderere Wahl als die von Wittum in [38] vorgeschlagene durchaus sinnvoll sein. Entscheidendes Kriterium ist die Rechenzeit des verwendeten Algorithmus 3.

Gitter	3	4	5	6	7
	11	11	12	14	14
$\alpha = 2$	3.7	5.2	22.5	107.7	511.6
	11	11	13	14	16
$\alpha = 3$	3.3	5.7	20.6	76.5	402.2
	11	12	13	16	> 30
$\alpha = 4$	3.3	6.3	16.7	86.7	—

cg-Iterationen
CPU-Zeit [sec]

Tabelle 6 Modellgebiet 2 mit 16 Prozessoren; Variierung des Parameters in der Frequenzfiltermethode

Mit $\alpha = 2$ werden die gleichen Iterationszahlen wie bei exaktem Löser (siehe Tabelle 1 Spalte 1) erreicht. Eine nur unwesentliche Erhöhung der Iterationszahlen bei $\alpha = 3$ geht mit einer deutlichen Einsparung an Rechenzeit einher. Der enorme Anstieg der Iterationszahl bei $\alpha = 4$ und 7 Gittern erklärt sich aus der Tatsache, daß die hohen Frequenzen bei dieser Gitterschrittweite ($h = 1/128$) schlecht gedämpft werden, da das höchstfrequente Testfrequenzenpaar das Paar (64,65) ist.

Modellgebiet 2 : Vergleich verschiedener Basistransformationsoperatoren B_I

Hier werden die jeweils schnellsten Varianten verschiedener Techniken in der Basistransformation B_I verglichen. Zum Einsatz kamen der Multigrid-V-Zyklus mit Gauß-Seidel-Glätter, die Frequenzfiltermethode und die MAF. Letztere aus dem Grund, daß eine MAF-Iteration arithmetisch billig ist. Die beiden anderen Verfahren wurden in vorangehenden Abschnitten bereits genauer untersucht.

C=C(B), B=	3	4	5	6	7
V22↑↑↓↓	12	13	15	18	21
	0.31	0.33	0.37	0.43	0.51
	3.7	5.2	11.3	37.9	154.8
FFM($\alpha = 3$)	11	11	13	14	16
	0.25	0.29	0.34	0.37	0.42
	3.3	5.7	20.6	76.5	402.2
MAF	22	31	49	76	117
	0.53	0.64	0.75	0.83	0.89
	5.7	9.5	23.4	83.7	404.0

cg-Iterationen
Iterationsrate
CPU-Zeit [sec]

Tabelle 7 Modellgebiet 2 mit 16 Prozessoren; Vergleich verschiedener Basistransformationen (jeweils schnellste Version)

Der verwendete Multigridalgorithmus in Zeile 1 ist der schnellste von allen. Wegen des verwendeten Algorithmus 3 ($C = C(B)$) ist er geringfügig schneller als bei Verwendung von Algorithmus 2 ($C = B$) (siehe Rechenzeiten Tabelle 2 in Abschnitt 4.1). Gleichzeitig ist das für die Bramble-Pasciak-Schatz Vorkonditionierung typische Anwachsen der Iterationsraten mit $O(\log h^{-1})$ erkennbar. Der Vorteil geringen Aufwandes der MAF pro Iteration wird durch die mit $O(\sqrt{H/h})$ wachsenden Iterationszahlen wieder aufgehoben, obwohl die bei diesem Verfahren benötigte Diagonalmatrix nur einmal berechnet wird [30]. Die von der Frequenzfiltermethode erzielten Iterationszahlen unterscheiden sich nur unwesentlich von denen bei Verwendung direkter Löser. Leider muß man sich bei diesem Verfahren zwischen Skylla und Charybdis entscheiden. Entweder man spart Rechenzeit, indem man das für jedes Testfrequenzpaar zu lösende tridiagonale Gleichungssystem einmal faktorisiert und abspeichert. Oder man will größere Beispiele rechnen und löst diese Gleichungssysteme immer wieder. Hier wurde Letzteres implementiert, der Rechenzeitgewinn der anderen Variante dürfte bei ca. 25 % liegen. Eine weitere Wertung der Rechenzeiten erfolgt im Abschnitt 6.

6 Schlußfolgerungen

Die in Tabelle 7 erzielten Zeiten lassen die Verwendung von Multigridoperatoren in der Basistransformation B_I als angeraten erscheinen. Jedoch muß diese Einschätzung in Hinblick auf andere Probleme als das Modellproblem (2.1) relativiert werden. Wie G. Wittum [38] zeigt, erbringt das Multigridverfahren im Falle anisotroper Materialien ($-\epsilon\,\partial_x^2 u - \partial_y^2 u = f$) sehr schlechte Konvergenzraten wenn ϵ sehr klein wird. Somit ist der im Moment zu beobachtende Rechenzeitgewinn gegenüber der Frequenfiltermethode von geringer Bedeutung. Wesentlich wichtiger wird die Entwicklung von Basistransformationsoperatoren B_I sein, die speziell auf den Unterraum $K_{IC}\mathrm{R}^{\mathrm{N_I}}$ wirken. Für das Modellgebiet 1 konnte ein solcher Operator konstruiert werden, für allgemeine Teilgebiete mit Crosspoints (Modellgebiet 2) ist das noch offen. Vielleicht erfüllt ein solcher Operator automatisch die entsprechenden Bedingung (3.14) auf dem Gesamtraum $\mathrm{R}^{\mathrm{N_I}}$, so daß Algorithmus 3 anstelle von Algorithmus 1 benutzt werden kann. Gleichzeitig könnte der die Basistransformation B_I in Algorithmus 1 definierende Operator $\overline{M}_I$ in Algorithmus 4 (3.16) eingesetzt werden. Eine weitere Idee wäre ein unterraumangepasster Glätter für das Multigridverfahren in Analogie zu Abschnitt 4.2 .

Der Algorithmus 3 hat seinen großen Vorteil darin, daß der Basistransformationsoperator B_I nichtsymmetrisch sein kann und sich die Symmetrie des inneren Lösers C_I zwangsläufig ergibt. Da die oben angeführte Frequenzfiltermethode einen nichtsymmetrischen Operator liefert, werden erst durch diesen neuen Algorithmus konkurrenzfähige Rechenzeiten erzielt.

Zum Zwecke vergleichender Betrachtungen zu Speedup und Effizienz ist weniger der Vergleich von einem Prozessor zu p Prozessoren als vielmehr der Vergleich von q zu p Prozessoren von Bedeutung, da die hier ermittelte Effizienz direkt in Kosten-Nutzen Betrachtungen verwandt werden kann. Die entsprechende Definition des Speedups von q auf p Prozessoren lautet

$$S(q,p) \quad = \quad \frac{\text{Zeit pro Freiheitsgrad}(q)}{\text{Zeit pro Freiheitsgrad}(p)} \quad = \quad \frac{T(q)}{N(q)} \times \frac{N(p)}{T(p)} \quad , \tag{6.1}$$

wobei $T(q)$ die CPU-Zeit des parallelen Algorithmus und $N(q)$ die Gesamtanzahl der Freiheitsgrade auf q Prozessoren darstellen. Die entsprechende Effizienz ergibt sich durch

$$E(p,q) \quad = \quad \frac{S(q,p)}{S_{theor.}(q,p)} \quad = \quad \frac{q}{p} \times S(q,p) \quad . \tag{6.2}$$

Es lassen sich die folgenden 3 Effizienzvergleiche anstellen (es wird stets auf die Randwertaufgabe (2.1) Bezug genommen).

Effizienzvergleich 1 : $h =$ **const.**

- lokale Diskretisierung konstant
- variable Prozessoranzahl p

D.h. diese Effizienzmessung gibt den praktisch relavanten Fall an, daß der lokale Speicher eines Prozessors zum größten Teil genutzt wird. Grundlage der Effizienzberechnung ist folgende Tabelle.

Gitter	# Prozessoren Gebiet			
	2	4	8	16
6	14.9	21.3	35.3	38.3
7	53.9	88.9	141.4	157.8

Tabelle 8 CPU-Zeit in sec.; $B = C = V22 \uparrow\uparrow\downarrow\downarrow$

Dies ergibt folgende Tabelle von Speedup und Effizienz bei Verdopplung der Prozessoranzahl.

Gitter	S(2,4)	S(4,8)	S(8,16)	E(2,4)	E(4,8)	E(8,16)
6	1.40	1.21	1.84	0.70	0.60	0.92
7	1.21	1.25	1.79	0.60	0.62	0.90

Tabelle 9 Speedup und Effizienz

Die gute Effizienz bei Verdopplung der Prozessorenanzahl von 8 auf 16 im Gegensatz zur Verdopplung von 4 auf 8 Prozessoren läßt sich wie folgt erklären. Bei der Einteilung des Einheitsquadrates in 4 Teilquadrate entsteht ein Crosspoint, gegenüber 3 Crosspoints bei 8 Teilgebieten und 9 Crosspoints bei 16 Teilquadraten (siehe Bild 2). Im Fall von 4 Teilgebieten reduziert sich das Grobgittersystem auf eine Gleichung mit einer Unbekannten. In den anderen Fällen kann das Grobgittersystem basierend auf dem Differenzenstern nur eine Näherung des tatsächlich entstehenden Grobgittersystems sein. Da sich diese unangenehme Eigenschaft des Grobgittervorkonditionierers von 8 zu 16 Teilgebieten nicht wesentlich ändert, wird die gute Effizienz erklärlich. Es ist anzunehmen, daß die Effizienz beim Vergleich von 16 zu 64 Teilgebieten eine mindestens genauso gute Effizienz (≥ 0.9) liefern wird. Leider konnte ich entsprechende Rechnungen nicht durchführen, da das Betriebssystem Helios (Parsytec) nicht einmal die Konfigurierung eines Hypercubes mit 32 Prozessoren zuließ. Dieser Engpass muß wahrscheinlich mittels Standaloneprogrammierung überwunden werden, wobei der Hypercube mittels Doppelknoten aufgebaut wird.

Effizienzvergleich 2 : $H =$ const.

- globale Diskretisierung konstant

- variable Prozessoranzahl p

D.h. diese Effizienzmessung gibt den Fall an, daß der lokale Speicher eines Prozessors mit wachsender Prozessoranzahl immer schlechter ausgenutzt wird. Die folgende Tabelle 10 gibt Rechenzeiten, Speedup und Effizienz für eine Vervierfachung der Prozessoranzahl an.

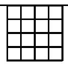
Gebiet	$H = \frac{1}{32}$	$H = \frac{1}{64}$	$H = \frac{1}{128}$	$H = \frac{1}{256}$
	1.73	5.47	21.3	88.9
	3.34	4.92	11.08	38.3
S(4,16)	0.52	1.11	1.92	2.32
E(4,16)	0.13	0.28	0.48	0.58

Tabelle 10 CPU-Zeit in sec.; $B = C = V22 \uparrow\uparrow\downarrow\downarrow$

Tabelle 10 dokumentiert das Wachsen der Effizienz mit der lokalen Problemgröße. Diese Effizienzmessung hat aber meines Erachtens nicht die praktische Relevanz der Effizienzmessung 1. Gleichzeitig gelten die Bemerkungen der Effizienzmessung 1, so daß bei Verwendung von 64 Prozessoren hier eine deutliche Effizienzverbesserung zu erwarten ist.

Effizienzvergleich 3 : $H =$ **const.** $p =$ **const.**

- globale Diskretisierung konstant
- konstante Prozessoranzahl
- variable lokale Problemgröße h

D.h. diese Effizienzmessung untersucht das Verhalten bei Halbierung der lokalen Problemgröße h, in diesem Falle zwischen den Gittern 6 und 7. Die Rechenzeiten können den Spalten der Tabelle 8 entnommen werden. Die Effizienz bei Halbierung der lokalen Schrittweite h bei Verwendung von 6 bzw. 7 Gittern lag bei allen verwendeten Modellgebieten um 1.0, d.h. eine Vervierfachung der Freiheitsgrade erbrachte fast genau eine Vervierfachung der Rechenzeit in den betrachteten Beispielen.

Die herausragende Rechenzeit von 155 Sekunden für $511^2 = 261\,121$ Freiheitsgrade beim Modellproblem (2.1) auf 16 T800-Prozessoren mit selbstprüfendem Speicher (das bedeutet ca. 20% mehr Rechenzeit !), erzielt auf dem Supercluster des IWR-Heidelberg, zeigt deutlich die Konkurrenzfähigkeit der Gebietszerlegung gegenüber globalisierten Verfahren. Der Zeitanteil für notwendige Kommunikationen im cg-Algorithmus blieb stets unter 5%. Die sich unter Umständen sehr negativ auf globale Verfahren auswirkende Materialsprünge sind bereits ohne Mehraufwand in der Gebietszerlegungsmethode integriert, falls man die Teilgebiete so legt, daß Materialgrenzen stets Gebietsgrenzen sind.

Eine weitere Möglichkeit parallele Algorithmen zu konstruieren, ist ein Kompromiß von Gebietszerlegung und parallelisierten globalen Verfahren. Dazu unterteilt man das Ausgangsgebiet Ω in Gebiete Ω_i gleicher Materialzahlen und einfacher Gestalt (Gebietszerlegung). Diesen

Gebieten Ω_i wird jeweils ein Teilcluster von p_i Prozessoren aus der Gesamtanzahl p der Prozessoren zugeordnet ($p = \sum_i p_i \;, \quad i = \overline{1, \#\text{Teilgebiete}}$). Die in den Teilgebieten Ω_i zu lösenden Gleichungssysteme $K_{I,i}\underline{u}_{I,i} = \underline{f}_{I,i}$ werden mittels parallelisierter globaler Verfahren in den einzelnen Teilgebieten Ω_i gelöst. Somit haben wir 2 Parallelisierungsebenen, die globale DD-Parallelisierung und die lokale (im Teilgebiet Ω_i) Parallelisierung mittels parallelisierter (im Teilgebiet Ω_i) globaler Verfahren. Ein weiterer Vorteil dieser Strategie besteht sicherlich darin, daß bei größeren Prozessoranzahlen ein dynamischer Lastenausgleich etwa bei lokaler Netzverfeinerung flexibler realisiert werden kann.

Der Einsatz von FEM und BEM in verschiedenen Gebieten einer sich nicht überlappenden Gebietszerlegung und die Kopplung beider Methoden über den Koppelrand Γ_C ist in einer Reihe praktischer Anwendungen außerordentlich vorteilhaft. Dazu zählt zum Beispiel die Berechnung elektromagnetischer Felder in der Elektrotechnik und insbesondere im Elektromaschinenbau [26]. In [27] wird von U. Langer eine parallelisierte und vorkonditionierte Variante des Bramble/Pasciak CG's [5] zur Auflösung der symmmetrischen und indefiniten gekoppelten FE/BE-DD-Gleichungen vorgeschlagen. Vom Standpunkt der parallelen Implementierung auf einem MIMD-Rechner mit verteiltem Speicher (z.B. Transputer) hat der in [27] diskutierte Algorithmus ähnlich exzellente Eigenschaften wie der in der vorliegenden Arbeit untersuchte parallele PCG-Algorithmus zur Lösung symmetrischer und positiv definiter FE-Gleichungen. Damit sind die hier erzielten Resultate und Erfahrungen für die Auflösung gekoppelter FE/BE-DD-Gleichungen sowohl vom Standpunkt der Vorkonditionierung als auch der Parallelisierung voll nutzbar.

Literatur

[1] O. Axelsson and P. S. Vassilevski. Construction of variable-step preconditioners for inner-outer iteration methods. In *Proceedings of the IMACS*, 1991. Brussels, April 2-4, 1991.

[2] P. Bastian and G. Horton. Parallelization of robust multigrid methods : ILU factorization and frequency decomposition method. *SIAM J.Sci.Stat.Comput.*, 12(6):1457–1470, 1991.

[3] B. E. Bjørstad, R. Moe, and M. Skogen. Parallel domain decomposition and iterative refinement algorithms. In *[25]*, pages 28–46, 1991.

[4] C. Börgers. The Neumann-Dirichlet domain decomposition method with inexact solvers on the subdomains. *Numer. Math*, 55(2):123–136, 1989.

[5] J. H. Bramble and J. E. Pasciak. A preconditioning technique for indefinite systems resulting from mixed approximation of elliptic problems. *Math. Comp.*, 50(181):1–17, 1988.

[6] J. H. Bramble, J. E. Pasciak, and A. H. Schatz. The construction of preconditioners for elliptic problems by substructuring. i-iv. *Math. Comp.*, 47, 49, 51, 53(175, 179):103–234, 1–16, 415–430, 1–24, 1986, 1987, 1988, 1989.

[7] J. H. Bramble, J. E. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Math. Comp.*, 1990.

[8] T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, editors. *Second International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, 1989. SIAM. Los Angeles, California, January 14-16, 1988.

[9] T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, editors. *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, 1990. SIAM. Houston, March 20-22, 1989.

[10] T. F. Chan and D. Resasco. A framework for the analysis and construction of domain decomposition preconditioners. In *[15]*, pages 217–230, 1988.

[11] M. Dryja. A capacitance matrix method for Dirichlet problems on polygonal regions. *Numer. Math.*, 39(1):51–64, 1982.

[12] M. Dryja and O. B. Widlund. An additive variant of Schwarz alternating method for the case of many subregions. Technical Report 339, Department of Computer Science, Courant Institute, New York, 1987. also Ultracomputer Note 131.

[13] U. Gärtel and R. Ressel. Parallel multigrid : Grid partitioning versus domain decomposition. Technical Report 599, Arbeitspapiere der GMD, 1991.

[14] G. Globisch and U. Langer. On the use of multigrid preconditioners in a multigrid software package. In S. Hengst, editor, *Proceedings of the "4-th Multigrid Seminar"held at Unterwirbach, GDR, May 2-6, 1989*, pages 105–134, Berlin, 1990. Academy of Science. Report-Nr. R-MATH-03/90.

[15] R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, editors. *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, 1988. SIAM. Paris, January 1987.

[16] G. H. Golub and D.Mayers. The use of preconditiong over irregular subregions. In R. Glowinski and J. L. Lions, editors, *Computing Methods in Applied Sciences and Engineering VI*, pages 3–14, Amsterdam, 1984. North-Holland.

[17] G. Haase and U. Langer. The non-overlapping domain decomposition multiplicative Schwarz method. Preprint 206, TU Chemnitz, 1991.

[18] G. Haase and U. Langer. On the use of multigrid preconditioners in the domain decomposition method. In *[25]*, pages 101–110, 1991.

[19] G. Haase, U. Langer, and A. Meyer. A new approach to the Dirichlet decomposition method. In S. Hengst, editor, *Proceedings of the "5-th Multigrid Seminar"held at Eberswalde, GDR, May 14-18, 1990*, pages 1–59, Berlin, 1990. Academy of Science. Report-Nr. R-MATH-09/90.

[20] G. Haase, U. Langer, and A. Meyer. The approximate dirichlet decomposition method; i,ii. *Computing*, 47:137–167, 1991.

[21] G. Haase, U. Langer, and A. Meyer. Domain decomposition preconditioners with inexact subdomain solvers. Preprint 192, TU Chemnitz, 1991. Appears in the "Journal of Numerical Linear Algebra with Applications".

[22] G. Haase and U. L. und Arnd Meyer. Parallelisierung und Vorkonditionierung des CG-Verfahrens durch Gebietszerlegung. 1991. erscheint im Tagungsbericht der GAMM-Tagung, Juni 1991, Heidelberg.

[23] W. Hackbusch. *Multi-Grid Methods and Applications.* Springer, Berlin, 1980.

[24] W. Hackbusch. *Iterative Lösung großer schwachbesetzter Gleichungssysteme.* Teubner, Stuttgart, 1991.

[25] W. Hackbusch, editor. *Parallel Algorithms for Partial Differential Equations*, Braunschweig, 1991. Vieweg. Proceedings of the Sixth GAMM-Seminar, Kiel, January 19-21, 1990.

[26] B. Heise. Mehrgitter-Newton-Verfahren zur berechnung nichtlinearer magnetischer felder. *Wissenschaftliche Schriftenreihe*, 4, 1991.

[27] U. Langer. Parallel iterative solution of symmetric coupled fe/be- equations via domain decomposition. Preprint, TU Chemnitz, 1992. Eingereicht.

[28] K. H. Law. A parallel finite element solution method. *Computer and Structures*, 23(6):845–858, 1989.

[29] O. A. McBryan and U. Trottenberg. Multigrid methods on parallel computers. Technical Report 370, Arbeitspapiere der GMD, 1989.

[30] A. Meyer. On the o(h^{-1})-property of MAF. Preprint 45, TU Karl-Marx-Stadt, 1987.

[31] A. Meyer. A parallel preconditioned conjugate gradient method using domain decomposition and inexact solvers on each subdomain. *Computing*, 45:217–234, 1990.

[32] P. Oswald. Preconditioners for discretization of the biharmonic equation by rectangular finite elements. Forschungsergebnisse der Math. Fakultät Math/91/03, FSU Jena, 1991.

[33] J. S. Przemieniecki. Matrix structural analysis of substructures. *AIAA J.*, 1:138–147, 1963.

[34] K. Ressel. Gebietszerlegung und Mehrgitterverfahren : Varianten, numerische Untersuchungen und Aspekte der Parallelisierung. Diplomarbeit, Math. Inst. der Univ. Köln, 1990.

[35] A. A. Samarski and E. S. Nikolaev. *Numerical methods for grid equations. Vol. II : Iterative Methods*. Birkhäuser-Verlag, Basel-Boston-Berlin, 1989.

[36] H. A. Schwarz. *Über einige Abbildungsaufgaben. Gesammelte Mathemathische Abhandlungen*, volume 2. Springer, Bonn, 1890. First published in "Vierteljahresschrift der Naturforschenden Gesellschaft in Zürich"1870, v.15, pp 272-286.

[37] B. F. Smith. Domain decomposition algorithms for the partial differential equations of linear elasticity. Technical Report 517, Argonne National Laboratory, Univ. of Chicago, 1990.

[38] G. Wittum. *Filternde Zerlegungen*. Teubner, Stuttgart, 1992.

[39] J. Xu. Iterative methods by space decomposition and subspace correction: A unifying approach. Technical Report AM67, Department of Mathematics, Penn State University, 1990.

[40] H. Yserentant. Two preconditioners on the multi-level-splitting of finite element spaces. *Numer. Math.*, 58:163–184, 1990.

A Der Eigenwert des Schur-Komplements im Modellproblem

Ausgehend vom Modellproblem (2.1) im Modellgebiet 1 (siehe Bild 1) wurde im Abschnitt 3.2 eine Fourierreihenentwicklung durchgeführt. Die folgenden Ausführungen eines neuen Zugangs zum Erhalt der Eigenwerte des Schur-Komplements beziehen sich auf Ergebnisse dieses Abschnittes. Insbesondere gelten die Gleichungen für die Eigenwerte $\lambda_{k,l}$ und Eigenvektoren $\mu_{k,l}$ (4.5) und (4.6) der Matrix K_I. Ohne Beschränkung der Allgemeinheit gelte $mat(x) \equiv 1$.

Wegen der Struktur des Schur-Komplements

$$\begin{aligned} S_C &= S_{C,1} + S_{C,2} \\ &= K_{C,1} - K_{CI,1} K_{I,1}^{-1} K_{IC,1} + K_{C,2} - K_{CI,2} K_{I,2}^{-1} K_{IC,2} \end{aligned} \tag{A.1}$$

im Modellgebiet 1 und der völligen Gleichheit der beiden Teilgebiete kann (A.1) in

$$S_C \;=\; K_C - 2 * K_{CI,1} K_{I,1}^{-1} K_{IC,1} \tag{A.2}$$

überführt werden. Zur Bestimmung der Eigenwerte von S_C betrachten wir das diskrete Eigenwertproblem

$$\begin{aligned} S_C \mu_l &= \lambda_l(S_C)\, \mu_l \quad \text{, d.h.} \\ \left(K_C - 2 * K_{CI,1} K_{I,1}^{-1} K_{IC,1}\right) \mu_l &= \left(\lambda_l(K_C) + 2\, \underbrace{\lambda_l(K_{CI,1} K_{I,1}^{-1} K_{IC,1})}_{\gamma_l}\right) \mu_l \quad l = \overline{1, n-1}\,. \end{aligned} \tag{A.3}$$

Die Eigenwerte von K_C ergeben sich nach der Darstellung

$$K_C \;=\; \text{Tridiag}\{-1, 4, -1\} \;=\; \text{Diag}\{2\} \;+\; \text{Tridiag}\{-1, 2, -1\}$$

als

$$\lambda_l(K_C) \;=\; 2 \;+\; \lambda_l \quad , \tag{A.4}$$

mit λ_l aus (4.5).

Nach der Fourierreihenentwicklung (4.7) von $\underline{u}_C \in \mathrm{R}^{\mathrm{N_C}}$ ergeben sich die Eigenwerte γ_l aus der Beziehung

$$\left(K_{CI,1} K_{I,1}^{-1} K_{IC,1}\right) \sum_{l=1}^{n-1} \alpha_l\, \mu_l(j) \;=\; \sum_{l=1}^{n-1} \gamma_l\, \alpha_l\, \mu_l(j) \quad j = \overline{1, n-1}\,. \tag{A.5}$$

Man erkennt, daß der Ausdruck $K_{I,1}^{-1} K_{IC,1} \sum_{l=1}^{n-1} \alpha_l\, \mu_l$ bereits in Gleichung (4.11) entwickelt wurde. Die dort entstandene Gleichung

$$\underline{z}_I(i,j) \;=\; K_{I,1}^{-1} K_{IC,1} \underline{u}_C \;=\; \sum_{l=1}^{n-1} \sum_{k=1}^{n-1} \left(\frac{-\mu_k(n-1)}{\lambda_{k,l}}\, \alpha_l \right) \mu_{k,l}(i,j)$$

wird nun weiterbearbeitet zu

$$\begin{aligned} \underline{w}_C(j) &= K_{CI,1} z_I(i,j) \;=\; -z_I(n-1,j) \\ &= -\sum_{l=1}^{n-1}\sum_{k=1}^{n-1}\left(\frac{-\mu_k(n-1)}{\lambda_{k,l}}\,\alpha_l\right)\,\mu_{k,l}(n-1,j) \\ &= \sum_{l=1}^{n-1}\underbrace{\left[\sum_{k=1}^{n-1}\left(\frac{\mu_k^2(n-1)}{\lambda_{k,l}}\right)\right]}_{=:S}\,\alpha_l\,\mu_l(j) \qquad j=\overline{1,n-1}\,. \end{aligned} \tag{A.6}$$

Unter Ausnutzung von $\mu_k^2(i) = 8h\sin^2(k\pi\frac{h}{2})\cos^2(k\pi\frac{h}{2})$ und $\lambda_{k,l} = 4\sin^2(k\pi\frac{h}{2}) + \lambda_l$ ergibt sich

$$S \;=\; 2h\sum_{k=1}^{n-1}\frac{\sin^2(k\pi\frac{h}{2})\left(1-\sin^2(k\pi\frac{h}{2})\right)}{\sin^2(k\pi\frac{h}{2}) + \underbrace{\lambda_l/4}_{=:C}} \qquad . \tag{A.7}$$

Mit Einführung der Größe $\triangle x = \pi h/2$ ändert sich (A.7) in

$$S \;=\; \frac{4}{\pi}\sum_{k=1}^{n-1}\frac{\sin^2(k\triangle x)\left(1-\sin^2(k\triangle x)\right)}{\sin^2(k\triangle x)+C}\,\triangle x \tag{A.8}$$

und es bietet sich der Übergang von der Summe zum Integral an. Für $h \longrightarrow 0$, d.h. immer feinere Vernetzung, geht die Summe S aus (A.8) in das Integral S_∞ über. Dieses Integral läßt sich wie folgt berechnen :

$$\begin{aligned} S_\infty &= \frac{4}{\pi}\int_0^{\pi/2}\frac{\sin^2(x)\left(1-\sin^2(x)\right)}{\sin^2(x)+C}\,dx \\ &= \frac{4}{\pi}\left\{-\int_0^{\pi/2}\sin^2(x)dx \;+(1+C)\int_0^{\pi/2}dx \;-C(1+C)\int_0^{\pi/2}\frac{dx}{\sin^2(x)+C}\right\} \\ &= \frac{4}{\pi}\left\{-\frac{\pi}{4} \;+(1+C)\frac{\pi}{2} \;-\sqrt{C^2+C}\,\frac{\pi}{2}\right\} \;=\; -1\;+2C\;-2\sqrt{C^2+C} \\ S_\infty &= 1+\frac{1}{2}\lambda_l - \frac{1}{2}\sqrt{\lambda_l^2+4\lambda_l} \;=:\; \gamma_l \end{aligned} \tag{A.9}$$

Somit ergeben sich die Eigenwerte des Schur-Komplements aus (4.8) für $h \longrightarrow 0$, mittels (A.4) und (A.9) als

$$\lambda_l(S_C) \;=\; \sqrt{\lambda_l^2+4\lambda_l} \quad , \tag{A.10}$$

und decken sich mit den von Golub [16] angegebenen. Gleichzeitig erkennt man, daß der von Dryja [11] angegebene Vorkonditionierer für den Koppelrand die Ungleichungen

$$\frac{1}{\sqrt{2}}\,\lambda_l(S_C) \;<\; \lambda_l(C_C) \;=\; 2\sqrt{\lambda_l} \;<\; \lambda_l(S_C) \tag{A.11}$$

erfüllt, da sich (A.10) in $2\sqrt{\lambda_l}\sqrt{\sin^2(l\pi h/2)+1}$ umformen läßt. Eine genauere Bestimmung der Eigenwerte des Schur-Komplements ohne die Bedingung $h \longrightarrow 0$ ist für streifenweise Zerlegungen in Chan/Resasco [10] zu finden.

B μ-Abschätzung für den Zweigitteroperator

In Abschnitt 3.2 wurde mit (4.13) eine allgemeine Abschätzung für das Verhalten von δ aus (4.3) und damit für μ aus (4.2) im Modellgebiet 1 gegeben. Der Summand in (4.13) spaltet sich in einen vom Operator $\overline{M}_I$ unabhängigen Teil $fak_{k,l}$ und den Dämfungsfaktor $\varrho^2_{k,l}(\overline{M}_I)$ auf. Die Berechnung dieses Dämpfungsfaktors wird im folgenden für die Wahl des Zweigitteroperators aus (3.21) mit ω-Jacobi-Glätter (3.19) angegeben. Alle angegebenen Eigenvektoren $\mu_{k,l}(i,j)$ und Eigenwerte $\lambda_{k,l}$ beziehen sich auf die in (4.6) definierten. Die Grundlagen der Berechnung sind bei W. Hackbusch [23] zu finden.

Sei die Fourierreihenentwicklung eines Vektors $\underline{z}_I$ durch

$$z_I(i,j) \;=\; \sum_{l=1}^{n-1}\sum_{l=1}^{n-1} \beta_{k,l}\,\mu_{k,l}(i,j) \qquad i,j=\overline{1,n-1} \tag{B.1}$$

gegeben. Um die Dämpfungsfaktoren $\varrho_{k,l}(\overline{M}_I)$ des Zweigitteroperators zu bestimmen, betrachten wir unter Einbeziehung von (B.1) die Gleichung

$$\|\,\overline{M}_I\underline{z}_I\,\|^2 \;=\; (\overline{M}_I\underline{z}_I\,,\,\overline{M}_I\underline{z}_I) \;=\; \sum_{l=1}^{n-1}\sum_{k=1}^{n-1} \varrho^2_{k,l}(\overline{M}_I)\beta^2_{k,l}\;. \tag{B.2}$$

Der Zweigitteroperator aus (3.21) wird etwas verändert als

$$\overline{M}_I \;=\; \underbrace{(I_h-\frac{\omega}{4}K_h)^\nu}_{S_h}\;\underbrace{\left(I_h-I^h_H K_H^{-1} I^H_h K_h\right)}_{C_h}\;\underbrace{(I_h-\frac{\omega}{4}K_h)^\nu}_{S_h} \tag{B.3}$$

geschrieben, mit den Eigenwerten des Glätters S_h

$$G_{k,l} \;:=\; \lambda_{k,l}(S_h) \;=\; \left(1-\frac{\omega}{4}\lambda_{k,l}\right)^\nu\;. \tag{B.4}$$

Die Fourieranalyse des Zweigitteroperators $S_hC_hS_h$ geschieht nach Hackbusch [23]. Folgende Abkürzungen werden verwendet (analog für Indizee l) :

$$k' \;=\; n-k\;,\; s_k \;=\; \sin^2(k\pi h/2)\;,\; c_k \;=\; \cos^2(k\pi h/2) \tag{B.5}$$

Offensichtlich gelten die Beziehungen (analog für Indizee l)

$$s_{k'} \;=\; c_k \quad \text{und} \quad c_{k'} \;=\; s_k\;. \tag{B.6}$$

Mittels einer Basistransformationsmatrix $Q_h = \{\mu_{k,l}\}_{k,l=\overline{1,n-1}}$, wobei die Eigenvektoren eines Indizeequadrupel $[k,l] = \{(k,l),(k',l),(k,l'),(k',l')\}$ in benachbarten Spalten plaziert sind, wird der Operator $S_h C_h S_h$ in einen blockdiagonalen Operator $\hat{S}_h \hat{C}_h \hat{S}_h$ überführt. Infolge dieser Basistransformation wird die zu betrachtende Gleichung

$$S_h C_h S_h \underline{z}_I(i,j) = \sum_{l=1}^{n-1} \sum_{k=1}^{n-1} \gamma_{k,l}\, \mu_{k,l}(i,j) \qquad i,j = \overline{1,n-1} \tag{B.7}$$

in

$$\hat{S}_h \hat{C}_h \hat{S}_h \{\beta_{k,l}\}_{k,l=\overline{1,n-1}} = \{\gamma_{k,l}\}_{k,l=\overline{1,n-1}} \tag{B.8}$$

umgewandelt. Die Vektoren β und γ besitzen die gleiche Blockstruktur wie die Matrizen $\hat{S}_h$ und $\hat{C}_h$. Während die Blöcke $\hat{S}_h^{[k,l]}$ des Operators $\hat{S}_h$ die Form

$$\hat{S}_h^{[k,l]} = \operatorname{diag}\{G_{k,l},\, G_{k',l},\, G_{k,l'},\, G_{k',l'}\} \tag{B.9}$$

besitzen, haben die einzelnen Blöcke $\hat{C}_h^{[k,l]}$ des Operators $\hat{C}_h$ die Form

$$\begin{aligned} C_h^{[k,l]} &= I_h^{[k,l]} - c \left[L_H^{[k,l]}\right]^{-1} c^T L_h^{[k,l]} \\ \text{mit} & \\ c &= [c_k c_l,\, -s_k c_l,\, -c_k s_l,\, s_k s_l]^T \\ \left[L_H^{[k,l]}\right]^{-1} &= \tfrac{1}{4(s_k c_k + s_l c_l)} =: R \\ L_h^{[k,l]} &= \operatorname{diag}\{\lambda_{k,l},\, \lambda_{k',l},\, \lambda_{k,l'},\, \lambda_{k',l'}\} \ . \end{aligned} \tag{B.10}$$

Gleichung (B.8) blockweise, aber ohne die Nachglättung betrachtet, liefert die Fourierkoeffizienten

$$\begin{aligned} \tilde{\gamma}^{[k,l]} &= C_h^{[k,l]} S_h^{[k,l]} \beta^{[k,l]} \\ &= G_{k,l}\beta^{[k,l]} - c\, \underbrace{G_{k,l} R c^T L_h^{[k,l]} \beta^{[k,l]}}_{s \in \mathrm{R}} \ . \end{aligned} \tag{B.11}$$

Bemerkenswert ist, daß die Größe s für alle Indexpaare des Quadrupels $[k,l]$ gleich ist. Die Abkürzungen in (B.11) durch die Ausdrücke in (B.10) ersetzt, (B.5) angewandt und die Abkürzung

$$A(x,y) := G_{x,y} R c_x c_y \lambda_{x,y} \tag{B.12}$$

eingeführt, ergibt für s die Beziehung

$$s = A(k,l)\beta_{k,l} - A(k',l)\beta_{k',l} - A(k,l')\beta_{k,l'} + A(k',l')\beta_{k',l'} \ . \tag{B.13}$$

Schreibt man c mittels (B.5) in $c = [c_k c_l,\, -c_{k'} c_l,\, -c_k c_{l'},\, c_{k'} c_{l'}]^T$ um, sieht man, daß alle Einträge (k,l) des Quadrupels $[k,l]$ das gleiche Aussehen

$$\tilde{\gamma}_{k,l} = G_{k,l}\beta_{k,l} - c_k c_l \left[A(k,l)\beta_{k,l} - A(k',l)\beta_{k',l} - A(k,l')\beta_{k,l'} + A(k',l')\beta_{k',l'}\right] \tag{B.14}$$

besitzen. Schließlich erhält man für die Fourierkoeffizienten von (B.8) den Ausdruck

$$\gamma_{k,l} \;=\; G_{k,l}^2\beta_{k,l} - c_k c_l G_{k,l}\left[A(k,l)\beta_{k,l} - A(k',l)\beta_{k',l} - A(k,l')\beta_{k,l'} + A(k',l')\beta_{k',l'}\right] \tag{B.15}$$

für alle Indexpaare $(k,l) \in [k,l]$. Im Falle von $k = k' = n/2$ oder (und) $l = l' = n/2$ reduziert sich Gleichung (B.15) auf $\gamma_{k,l} \;=\; G_{k,l}^2\beta_{k,l}$.

Nach Ausführung der Rücktransformation Q_h^{-1} und der formalen Berechnung der linken Seite von (B.2) ergibt sich mit

$$\|\overline{M}_I \underline{z}_I\| = \sum_{l=1}^{n-1}\sum_{l=1}^{n-1} G_{k,l}^2 \left\{G_{k,l}\beta_{k,l} - c_k c_l\left[A(k,l)\beta_{k,l} - A(k',l)\beta_{k',l} - A(k,l')\beta_{k,l'} + A(k',l')\beta_{k',l'}\right]\right\}^2 \tag{B.16}$$

ein Ausdruck, der sich nur über eine grobe Abschätzung in die Form der rechten Seite von (B.2) überführen läßt. Eine besserer Weg ist der folgende.

Sei $\underline{w}_I \;=\; \overline{M}_I \underline{z}_I$ der nach der Rücktransformation Q_h^{-1} enstandene Vektor, mit der entsprechenden Fourierreihenentwicklung

$$w_I(i,j) = \sum_{l=1}^{n-1}\sum_{l=1}^{n-1} \gamma_{k,l}\mu_{k,l}(i,j) \qquad i,j = \overline{1, n-1}\ . \tag{B.17}$$

Beachtet man die Struktur von $\gamma_{k,l}$ in (B.15), d.h. die Durchmischung der Fourierkoeffizienten $\beta_{k,l}$ innerhalb des Indexquadrupels $[k,l]$, bietet sich eine Umformung von (B.17) in

$$w_I(i,j) = \sum_{l=1}^{n/2}\sum_{l=1}^{n/2} \underbrace{\sum_{(k,l)\in[k,l]} \gamma_{k,l}\,\mu_{k,l}(i,j)}_{w_I^{[k,l]}(i,j)} \qquad i,j = \overline{1, n-1} \tag{B.18}$$

mit anschließender Analyse der Summe $w_I^{[k,l]}(i,j)$ über dem Indexquadrupel $[k,l]$ an. Ordnet man diese Summe nicht nach den Eigenwerten $\mu_{k,l}$, sondern nach den Koeffizienten $\beta_{k,l}$ der ursprünglichen Fourierreihenentwicklung (B.1) und führt die Abkürzung

$$B_{k,l} \;:=\; G_{k,l} c_k c_l \tag{B.19}$$

ein, ergibt sich

$$w_I^{[k,l]}(i,j) \;=\; \sum_{(k,l)\in[k,l]} \left\{G_{k,l}^2\,\mu_{k,l} - A(k,l)\left[B_{k,l}\,\mu_{k,l} - B_{k',l}\,\mu_{k',l} - B_{k,l'}\,\mu_{k,l'} + B_{k',l'}\,\mu_{k',l'}\right]\right\}\beta_{k,l}\ . \tag{B.20}$$

Die leicht nachzuprüfende Beziehung

$$\mu_{k'}(i) \;=\; (-1)^i\,\mu_k(i) \tag{B.21}$$

ausnutzend, kann (B.20) zu

$$w_I^{[k,l]}(i,j) \; = \qquad \text{(B.22)}$$
$$\sum_{(k,l)\in[k,l]} \underbrace{\left\{G_{k,l}^2\,\mu_{k,l} - A(k,l)\left[B_{k,l} - (-1)^i B_{k',l} - (-1)^j B_{k,l'} + (-1)^{i+j} B_{k',l'}\right]\right\}}_{\Gamma_{k,l}(i,j)} \beta_{k,l}\,\mu_{k,l}(i,j)\;.$$

umgeschrieben werden. Somit läßt sich der Vektor $\underline{w}_I$ durch

$$w_I(i,j) = \sum_{l=1}^{n}\sum_{l=1}^{n} \Gamma_{k,l}(i,j)\,\beta_{k,l}\,\mu_{k,l}(i,j) \qquad i,j = \overline{1,n-1} \qquad \text{(B.23)}$$

ausdrücken.

Da $\Gamma_{k,l}(i,j)$ vom Ort (i,j) derart abhängt, daß zwischen geraden und ungeraden Indizees i und j unterschieden werden muß, zerfällt das interessierende Euklidische Skalarprodukt in 4 Teile

$$(\underline{w}_I,\, \underline{w}_I) \; = \|\, \overline{M}_I \underline{z}_I \,\| = \; S_1 + S_2 + S_3 + S_3 \;\;. \qquad \text{(B.24)}$$

Entsprechend (B.22) werden diese Teilskalarprodukte wie folgt berechnet :

$$\begin{aligned}
\sum_{i-\text{odd}}\sum_{j-\text{odd}} \Gamma_{k,l}(i,j)\,\mu_{k,l}(i,j)\,\beta_{k,l} &= \frac{1}{4}\left\{G_{k,l}^2 - A(k,l)\left[B_{k,l} - B_{k',l} - B_{k,l'} + B_{k',l'}\right]\right\}^2 \beta_{k,l}^2 \\
\sum_{i-\text{odd}}\sum_{j-\text{even}} \Gamma_{k,l}(i,j)\,\mu_{k,l}(i,j)\,\beta_{k,l} &= \frac{1}{4}\left\{G_{k,l}^2 - A(k,l)\left[B_{k,l} - B_{k',l} + B_{k,l'} - B_{k',l'}\right]\right\}^2 \beta_{k,l}^2 \\
\sum_{i-\text{even}}\sum_{j-\text{odd}} \Gamma_{k,l}(i,j)\,\mu_{k,l}(i,j)\,\beta_{k,l} &= \frac{1}{4}\left\{G_{k,l}^2 - A(k,l)\left[B_{k,l} + B_{k',l} - B_{k,l'} - B_{k',l'}\right]\right\}^2 \beta_{k,l}^2 \\
\sum_{i-\text{even}}\sum_{j-\text{even}} \Gamma_{k,l}(i,j)\,\mu_{k,l}(i,j)\,\beta_{k,l} &= \frac{1}{4}\left\{G_{k,l}^2 - A(k,l)\left[B_{k,l} + B_{k',l} + B_{k,l'} + B_{k',l'}\right]\right\}^2 \beta_{k,l}^2
\end{aligned}$$

Die Teilskalarprodukte unterscheiden sich nur in den Vorzeichen einzelner Terme. Da eine Nebenrechnung (u, v, x, y, z stellen obige Terme in den S_i dar) die Vereinfachung

$$\begin{aligned}
&(u-v+x+y-z)^2 + (u-v+x-y+z)^2 + (u-v-x+y+z)^2 + (u-v-x-y-z)^2 \\
&\qquad = \; 4u^2 + 4v^2 + 4x^2 + 4y^2 + 4z^4 - 8uv \; = \; 4\left((u-v)^2 + x^2 + y^2 + z^2\right)
\end{aligned}$$

liefert, kann schließlich für (B.2) die Gleichung

$$\|\, \overline{M}_I \underline{z}_I \,\| = \qquad \text{(B.25)}$$
$$\sum_{l=1}^{n-1}\sum_{l=1}^{n-1} \underbrace{\left\{G_{k,l}^4\left(1 - Rc_k^2c_l^2\lambda_{k,l}\right)^2 + G_{k,l}^2R^2c_k^2c_l^2\lambda_{k,l}^2\left[G_{k',l}^2 s_k^2 c_l^2 + G_{k,l'}^2 c_k^2 s_l^2 + G_{k',l'}^2 s_k^2 s_l^2\right]\right\}\beta_{k,l}^2}_{\varrho_{k,l}^2(\overline{M}_I)}$$

geschrieben werden. Die in (B.25) vorkommenden Abkürzungen wurden in (B.4), (B.5) und (4.6) eingeführt.

Damit sind die Dämpfungsfaktoren $\varrho_{k,l}(\overline{M}_I)$ des Zweigitteroperators mit ω-Jacobi-Gätter bestimmt. Den Ausdruck für $\varrho^2_{k,l}(\overline{M}_I)$ aus (B.25) in die Ungleichung (4.13) eingesetzt und damit μ in (4.4) abgeschätzt, liefert eine obere Schranke für $\mu = \varrho(S_C^{-1}T_C)$. Die numerische Auswertung der resultierenden μ-Abschätzung mit $\omega = 2/3$ und $\nu = 1$ zeigt Tabelle 11.

#Gitter	3	4	5	6	7	8	9	10	11	12	13	14
μ	0.09	0.20	0.42	0.85	1.71	3.43	6.85	13.71	27.41	54.82	109.7	219.3

Tabelle 11 μ-Abschätzung für TG11($\omega = 2/3$) im Modellgebiet 1

Aus Tabelle 11 wird ersichtlich, daß $\mu = O(h^{-1})$ ist, und sich die Konditionszahl $\kappa(C^{-1}K)$ im Modellproblem somit ebenfalls wie $O(h^{-1})$ verhält.

C μ-Abschätzung für die Frequenzfiltermethode

In Abschnitt 3.2 wurde mit (4.13) eine allgemeine Abschätzung für das Verhalten von δ aus (4.3) und damit für μ aus (4.2) im Modellgebiet 1 gegeben. Der Summand in (4.13) spaltet sich in einen vom Operator $\overline{M}_I$ unabhängigen Teil $fak_{k,l}$ und den Dämfungsfaktor $\varrho^2_{k,l}(\overline{M}_I)$ auf. Die Berechnung dieses Dämpfungsfaktors wird im folgenden für die Wahl der Frequenzfiltermethode (3.25) - (3.28) mit einer spaltenweisen Numerierung (parallel zum Koppelrand) im Modellgebiet 1 angegeben. Die Grundlagen der Berechnung sind in G. Wittum [38] zu finden.

Da der Frequenzfilteroperator $S_{I,\nu}$ aus (3.26) für das Modellproblem (2.1) positive reelle Eigenwerte besitzt [38], gilt wegen(3.27)

$$\varrho_{k,l}(\overline{M}_I) \;=\; \prod_{\nu\in\omega_\nu} \lambda^{(\nu)}_{k,l}(FFM) \qquad k,l = \overline{1, n-1} \quad . \tag{C.1}$$

Die Berechnung der Eigenwerte $\lambda^{\nu}_{k,l}(FFM)$ für das Testvektorenpaar $(\nu,\ \nu{+}1)$ erfolgt über den in [38] angegebenen Formelsatz für das Tridiagonale Schema. Den Ausdruck für $\varrho_{k,l}(\overline{M}_I)$ aus (C.1) in die Ungleichung (4.13) eingesetzt und damit μ in (4.4) abgeschätzt, liefert eine obere Schranke für $\mu = \varrho(S_C^{-1}T_C)$. Die numerische Auswertung der resultierenden μ-Abschätzung mit $\alpha = 2$ in der Frequenzfiltermethode zeigt Tabelle 12.

#Gitter	6	7	8	9	10	11	12	13	14	15	16
$\mu = x * 10^{-7}$	0.043	0.181	0.357	0.512	0.692	0.905	1.165	1.344	1.546	1.662	1.78

Tabelle 12 μ-Abschätzung für FFM($\alpha = 2$) im Modellgebiet 1

Die Tabelle 12 dokumentiert zweierlei. Erstens ist der Wert von μ so klein, daß die Frequenzfiltermethode auf diesem Unterraum schon fast ein exakter Löser ist. Zweitens scheint der

Wert von μ mit wachsender Gitteranzahl (d.h. $h \longrightarrow 0$) gegen eine h-unabhängige Schranke zu konvergieren, da der Quotient $\mu(l) \,/\, \mu(l+1)$ monoton fallend gegen 1 strebt. Könnte man dies analytisch beweisen, wäre die Frequenzfiltermethode ein Iterationsverfahren welches die Börgers-Vermutung ($\ln(h^{-1})$ Rechenschritte nötig für h-Unabhängigkeit von μ) widerlegen kann. Dieser analytische Beweis steht noch aus. Höhere Gitterzahlen konnten in der numerischen μ-Abschätzung nicht erreicht werden.