
Die Frequenzzzerlegungsmethode als robustes Mehrgitterverfahren: Implementierung und Parallelisierung

von

Peter Bastian, geb. 7.3.64 in Nürnberg

Danksagung

Ich möchte besonders Graham Horton für die unzähligen Diskussionen danken, die wir zusammen geführt haben. Weiter möchte ich mich bei Gabriel Wittum und Georg Scheuerer für ihre Hinweise und ihre große Anteilnahme am Fortgang der Arbeit bedanken. Nicht zuletzt gilt mein Dank auch meinem Betreuer Herrn J. Volkert, der die Behandlung des Themas erst ermöglichte.

Inhaltsverzeichnis

1	Einführung	1
2	Standardmehrgitteriteration	5
2.1	Testprobleme	5
2.2	Diskretisierung	7
2.3	Mehrgitterverfahren	8
2.4	Fourieranalyse des Standardverfahrens	10
2.5	Definition der Robustheit	15
3	Die Frequenzerlegungsmethode für periodische Randbedingungen	17
3.1	Mehrfache Grobgitterkorrektur	17
3.2	Notwendige Grobgitterkorrekturen	21
3.3	Komplexität	24
3.4	Fourieranalyse	26
3.5	Numerische Ergebnisse	27
4	Die Frequenzerlegungsmethode für Dirichlet und von Neumann Randbedingungen	31
4.1	Die groben Gitter	31
4.2	Berechnung des Galerkin Produktes	34
4.3	Numerische Ergebnisse	35
4.4	Probleme mit der Frequenzerlegungsmethode	45
4.5	Eine Modifikation	46
5	Parallelisierung	51

5.1	Zur Parallelisierung von Mehrgitterverfahren	51
5.2	Parallele Frequenzerlegungsmethode	55
5.3	DIRMU Implementierung	63
5.4	Speedups	65
6	Zusammenfassung und Ausblick	69

Tabellenverzeichnis

2.1	Konvergenzraten für ein Standard-Mehrgitterverfahren angewandt auf die anisotrope Gleichung	10
3.1	Zweigiterraten für die Frequenzzzerlegungsmethode (ged. Jacobi Glätter mit $\omega = \frac{1}{2}$, $h = \frac{1}{16}$, $\nu_1 = 2$, $\nu_2 = 0$) angewandt auf die anisotrope Gleichung	27
3.2	Mehrgitterraten für die Frequenzzzerlegungsmethode (ged. Jacobi Glätter mit $\omega = \frac{1}{2}$, $h = \frac{1}{2^{l+1}}$, $\nu_1 = 2$, $\nu_2 = 0$, $\gamma = 1$) angewandt auf die anisotrope Gleichung mit periodischen Randbedingungen	29
3.3	Mehrgitterraten für die Frequenzzzerlegungsmethode (red-black Gauß-Seidel Glätter, $h = \frac{1}{2^{l+1}}$, $\nu_1 = 2$, $\nu_2 = 0$, $\gamma = 1$) angewandt auf die anisotrope Gleichung mit periodischen Randbedingungen	29
3.4	Mehrgitterraten für die Frequenzzzerlegungsmethode (ged. Jacobi Glätter $\omega = \frac{1}{2}$, $h = \frac{1}{2^{l+1}}$, $\nu_1 = 2$, $\nu_2 = 0$) mit verschiedenen Zyklusformen, angewandt auf die anisotrope Gleichung mit periodischen Randbedingungen	30
4.1	Mehrgitterraten für die Frequenzzzerlegungsmethode (ged. Jacobi Glätter $\omega = \frac{1}{2}$, $h = \frac{1}{2^{l+1}}$, $\nu_1 = 2$, $\nu_2 = 0$) angewandt auf die anisotrope Gleichung mit Dirichlet Randbedingungen	41
4.2	Mehrgitterraten für die Frequenzzzerlegungsmethode (red-black Gauß-Seidel Glätter, $h = \frac{1}{2^{l+1}}$, $\nu_1 = 2$, $\nu_2 = 0$) angewandt auf die anisotrope Gleichung mit Dirichlet Randbedingungen	41
4.3	Mehrgitterraten für die Frequenzzzerlegungsmethode (red-black Gauß-Seidel Glätter, $h = \frac{1}{32}$, $\nu_1 = 2$, $\nu_2 = 0$) angewandt auf die anisotrope Gleichung mit variablen Koeffizienten	42
4.4	Mehrgitterraten für die Frequenzzzerlegungsmethode (red-black Gauß-Seidel Glätter, $h = \frac{1}{32}$, $\nu_1 = 2$, $\nu_2 = 0$) angewandt auf die anisotrope Gleichung mit gemischten Randbedingungen	42

4.5	Mehrgitterraten für die Frequenzzzerlegungsmethode (red-black Gauß-Seidel Glätter, $h = \frac{1}{32}$, $\nu_1 = 2$, $\nu_2 = 0$) angewandt auf die gedrehte, anisotrope Gleichung mit Dirichlet Randbedingungen	43
4.6	Mehrgitterraten für die Frequenzzzerlegungsmethode (red-black Gauß-Seidel Glätter, $h = \frac{1}{32}$, $\nu_1 = 2$, $\nu_2 = 0$) mit diagonalanisotroper Diskretisierung	43
4.7	Mehrgitterraten für die Frequenzzzerlegungsmethode (red-black Gauß-Seidel Glätter, $h = \frac{1}{32}$, $\nu_1 = 2$, $\nu_2 = 0$) angewandt auf die gescherte, anisotrope Gleichung mit Dirichlet Randbedingungen	44
4.8	normierte Rechenzeiten für verschiedene Zyklusformen (red-black Gauß-Seidel Glätter, $h = \frac{1}{32}$, $\nu_1 = 2$, $\nu_2 = 0$) für die anisotrope Gleichung mit Dirichlet Randbedingungen	44
4.9	Zweigiterraten für den modifizierten Standardansatz (ged. Jacobi Glätter mit $\omega = \frac{1}{2}$, $h = \frac{1}{16}$, $\nu_1 = 2$, $\nu_2 = 0$) angewandt auf die anisotrope Gleichung	49
5.1	Parallelisierungsgrad von verschiedenen Glättungsalgorithmen	53
5.2	Normierte, parallele Rechenzeiten für verschiedene Verfahren angewandt auf die anisotrope Gleichung, $h=1/64$, $p = 8,16$	66
5.3	Speedup für den $\gamma = \langle 1, 1, 1, 1 \rangle$ Zyklus	67
5.4	Speedup für den $\gamma = \langle 1, 2, 2, 2 \rangle$ Zyklus	67
5.5	Speedup für den $\gamma = \langle 2, 2, 2, 2 \rangle$ Zyklus	68

Abbildungsverzeichnis

2.1	Gitter für die Diskretisierung der anisotropen Gleichung mit periodischen Randbedingungen	8
3.1	grobe Gitter bei der Frequenzzzerlegungsmethode mit periodischen Randbedingungen	18
3.2	Fehlerfunktion bei Anwendung von p_{10} auf eine konstante 1 Funktion . .	19
3.3	Baum der notwendigen Grobgitterkorrekturen	22
4.1	grobe Gitter bei der Frequenzzzerlegungsmethode mit Dirichlet Randbedingungen	33
4.2	Gitterausschnitt zur Erläuterung des Galerkin Produktes	35
4.3	Gitter zum gescherten Problem	39
5.1	Aufteilung der Gitterpunkte auf die Prozessorkonfiguration	54
5.2	Transportanforderung bei acht Prozessoren pro Zeile	57
5.3	Ziehharmonikakonstruktion für acht Prozessoren	58
5.4	Transportanforderungen bei der Ziehharmonikakonstruktion	59
5.5	Aufteilung der Teilbäume einer Typ I Verzweigung	60
5.6	Abspeicherung eines Gitters im Privat- und Multiportspeicher	64

Kapitel 1

Einführung

In vielen Bereichen der Physik und der Ingenieurwissenschaften treten partielle Differentialgleichungen auf. Im allgemeinen wird eine Funktion u in mehreren Variablen in einem Gebiet gesucht, wobei eine Beziehung zwischen partiellen Ableitungen der Funktion u gegeben ist, einschließlich gewisser Bedingungen am Rand des Gebietes. Ein Beispiel ist etwa die bekannte Laplace Gleichung $\Delta u = 0$, die die Temperaturverteilung in einer Metallplatte beschreibt. Als Randbedingung kann z. B. die Temperatur am Rand vorgegeben sein.

Da für die meisten Gleichungen keine analytische Lösung existiert, setzt man numerische Verfahren zu deren Lösung ein. Um eine Behandlung im Rechner zu ermöglichen wird dazu zunächst aus dem kontinuierlichen Problem ein diskretes Problem abgeleitet. Das diskrete Problem ist in der Regel ein lineares Gleichungssystem mit sehr vielen Unbekannten, falls eine hohe Genauigkeit der Lösung erwünscht ist. Die Matrix des linearen Gleichungssystems ist sehr dünn besetzt, die Anzahl der Elemente ist proportional zur Anzahl der Unbekannten. Zur Lösung der linearen Gleichungssysteme haben sich iterative Verfahren (siehe vor allem [13]) durchgesetzt, da diese in der Lage sind die Nullstruktur der Matrix auszunutzen. Seit dem Verfahren von Jacobi (1845) wurden wesentlich verbesserte Methoden entwickelt, die schließlich zum Mehrgitterverfahren führten. Die wichtigste Eigenschaft des Mehrgitterverfahrens ist, daß die Anzahl der Iterationsschritte, die zur Lösung benötigt werden, nicht mehr von der Anzahl der Unbekannten (Genauigkeit!) abhängt. Damit ist das Mehrgitterverfahren ein optimales Verfahren (der Aufwand ist proportional zur Anzahl der Unbekannten).

Die schnelle Konvergenz der Mehrgitterverfahren ist allerdings nur für die diskreten Probleme, die aus einer eingeschränkten Klasse partieller Differentialgleichungen entstehen, gesichert (sog. elliptische Differentialgleichungen). In der Praxis treten häufig elliptische Differentialgleichungen auf, die von einem Parameter ϵ abhängen und die für $\epsilon \rightarrow 0$ nicht mehr elliptisch sind. (singulär gestörte Probleme, siehe [6, Kapitel 10]).

Im numerischen Lösungsverfahren kann diese Typänderung eine starke Verschlechterung

der Konvergenzrate (Faktor um den der Fehler pro Iteration reduziert wird) bewirken, bis hin zur Divergenz. Behält ein Verfahren für alle Werte des Parameters seine guten Konvergenzeigenschaften, wird es *robust für dieses Problem* genannt (für eine formale Definition der Robustheit siehe Ende Kapitel 2).

Ein weiterer Gesichtspunkt bei der numerischen Lösung partieller Differentialgleichungen ist der Einsatz von Parallelrechnern. Im Rahmen dieser Arbeit sollen Multiprozessoren betrachtet werden, die aus einer mittleren Anzahl (Größenordnung 16 bis 256) von leistungsfähigen Einzelprozessoren bestehen. Dabei kann ein solcher Einzelprozessor wieder ein Vektorrechner sein, wie etwa im SUPRENUM System.

In den Arbeiten zur Parallelisierung numerischer Verfahren wird häufig der Algorithmus verwendet, mit dem die höchste Auslastung erzielt wird ("MFLOP Rate") ohne dabei die Gesamteffizienz des Verfahrens zu betrachten. In Kapitel 5 wird deshalb die normierte, parallele Rechenzeit (Def. siehe dort) als objektives Kriterium zum Vergleich von iterativen, numerischen Methoden auf Parallelrechnern verwendet.

Ziel dieser Arbeit war es, ein robustes Mehrgitterverfahren, die Frequenzzzerlegungsmethode von Hackbusch [8], auf einem Parallelrechner zu implementieren und nachzuweisen, daß die Gesamteffizienz dieses Verfahrens für bestimmte singular gestörte Probleme besser ist, als die herkömmlicher Methoden für Parallelrechner.

æ

Kapitel 2

Standardmehrgitteriteration

Zunächst werden einige einfache partielle Differentialgleichungen vorgestellt, an denen für die Praxis relevante Probleme isoliert studiert werden können. Nach der Beschreibung des Diskretisierungsprozesses für die anisotrope Gleichung wird der Standardmehrgitteralgorithmus zur Lösung des entstandenen linearen Gleichungssystems vorgestellt. Anhand praktischer Versuche wird dessen Versagen für die anisotrope Gleichung gezeigt und theoretisch mit Hilfe der Fourieranalyse erklärt.

2.1 Testprobleme

Im folgenden werden zwei skalare, lineare, partielle Differentialgleichungen beschrieben, an denen zwei Probleme studiert werden können, die bei der Lösung der strömungsmechanischen Grundgleichungen, den sog. Navier-Stokes'schen Gleichungen, häufig auftreten. Es sind dies die anisotrope Gleichung und die Konvektions-Diffusionsgleichung.

Die anisotrope Gleichung ist eine einfache Modifikation der Poisson-Gleichung, mit unterschiedlichen Faktoren vor den zweiten Ableitungen. Man erhält die anisotrope Gleichung, wenn man die Poisson-Gleichung auf einem rechteckigem Gebiet lösen will und dieses mit Hilfe einer Koordinatentransformation auf das Einheitsquadrat zurückführt. Bei der Lösung der Navier-Stokes'schen Gleichungen wird der Druck durch eine Poisson-Gleichung beschrieben und die Lösung in krummlinigen Gebieten wird häufig durch eine Koordinatentransformation auf das Einheitsrechteck umgangen. Genau diese Situation soll mit der anisotropen Gleichung modelliert werden.

Wir betrachten die anisotrope Gleichung mit zwei verschiedenen Randbedingungen: periodische Randbedingungen bzw. Dirichlet Randbedingungen. Wobei erstere für die theoretischen Untersuchungen in diesem Kapitel günstiger und letztere für die Praxis relevanter sind.

Problem 2.1 Anisotrope Gleichung mit Dirichlet Randbedingungen.

$$-\alpha \frac{\partial^2}{\partial x^2} u(x, y) - \beta \frac{\partial^2}{\partial y^2} u(x, y) = f(x, y) \quad (2.1)$$

im Gebiet $\Omega = (0, 1) \times (0, 1)$, sowie $u(x, y) = b(x, y)$ auf dem Rand $\partial\Omega$ und $\alpha, \beta \geq 0$.

Problem 2.2 Anisotrope Gleichung mit periodischen Randbedingungen.

$$-\alpha \frac{\partial^2}{\partial x^2} u(x, y) - \beta \frac{\partial^2}{\partial y^2} u(x, y) + \frac{1}{4} u(x, y) = f(x, y) \quad (2.2)$$

im Gebiet $\Omega = (-1, 1) \times (-1, 1)$ und $\alpha, \beta \geq 0$, sowie $u(-1, y) = u(1, y), u(x, -1) = u(x, 1), u_x(-1, y) = u_x(1, y), u_y(x, -1) = u_y(x, 1)$.

Die Addition von $\frac{1}{4}u(x, y)$ ist nötig, damit die Lösung der Differentialgleichung eindeutig bestimmt ist, denn ohne diesen Summanden wäre mit $u(x, y)$ auch jedes $u(x, y) + C, C \in \mathfrak{R}$ (\mathfrak{R} , Menge der reellen Zahlen) eine Lösung. Man beachte auch den unterschiedlichen Definitionsbereich gegenüber den Dirichlet-Randbedingungen.

Die Konvektions-Diffusionsgleichung beschreibt z. B. die Temperaturverteilung in einer Flüssigkeit bei vorgegebenem Strömungsfeld. Dabei werden die ersten Ableitungen als konvektive und die zweiten Ableitungen als diffusive Terme bezeichnet. Der Parameter ϵ beschreibt die Zähigkeit der Flüssigkeit. Ein kleines ϵ (hohe Reynoldszahl) steht für ein dünnflüssiges Medium wie etwa Wasser. Da die einzelnen Impulsgleichungen der Navier-Stokes'schen Gleichungen die Form der Konvektions-Diffusionsgleichung haben ist diese Gleichung ein geeignetes Testproblem.

Problem 2.3 Konvektions-Diffusionsgleichung mit Dirichlet Randbedingungen.

$$-\epsilon \left(\frac{\partial^2}{\partial x^2} u(x, y) + \frac{\partial^2}{\partial y^2} u(x, y) \right) + c_1 \frac{\partial}{\partial x} u(x, y) + c_2 \frac{\partial}{\partial y} u(x, y) = f(x, y) \quad (2.3)$$

im Gebiet $\Omega = (0, 1) \times (0, 1)$, sowie $u(x, y) = b(x, y)$ auf dem Rand $\partial\Omega$ und $\epsilon \geq 0$.

Die beiden vorgestellten Probleme sind partielle Differentialgleichungen, die von Parametern abhängen. In der anisotropen Gleichung sind dies α und β und in der Konvektions-Diffusionsgleichung die Zahl ϵ . Die anisotrope Gleichung ist für $\alpha, \beta > 0$ elliptisch, aber für $\alpha = 0$ oder $\beta = 0$ nicht mehr. Diese Tatsache wirkt sich auf das Konvergenzverhalten von einigen Standardverfahren sehr nachteilig aus, wie im Rest dieses Kapitels sowohl theoretisch als auch praktisch gezeigt wird. Die Frequenzzzerlegungsmethode wird sich als robustes Verfahren für die anisotrope Gleichung erweisen, d. h. die Konvergenzrate ist relativ unabhängig von der Wahl der Parameter α und β .

Bei der Konvektions-Diffusionsgleichung bereitet der Fall $\epsilon \rightarrow 0$ mit vielen Standardverfahren Schwierigkeiten. Da allerdings mit der Frequenzzzerlegungsmethode, wie sie von Hackbusch in [8] vorgestellt wurde, dieses Problem noch nicht gelöst werden kann, betrachten wir für den Rest dieser Arbeit nur die anisotrope Gleichung. Ein für beide Testprobleme robustes Verfahren ist das Mehrgitterverfahren mit ILU-Glättung, dessen Parallelisierung in [12] beschrieben ist.

2.2 Diskretisierung

Um die Differentialgleichung 2.2 numerisch zu lösen, wird der Definitionsbereich Ω durch ein regelmäßiges Gitter der Feinheit $h_l = 1/N_l$, $N_l = 2^{l+1}$ ersetzt. Der Index l bestimmt dabei eine ganze Folge von immer feineren Gittern, die im Mehrgitterverfahren benötigt werden. Die zweiten Ableitungen werden durch finite Differenzen approximiert, und die Lösung $u(x, y)$ nur noch an den diskreten Gitterpunkten $\Omega_l = \{(ih_l, jh_l) | 1 - N_l \leq i, j \leq N_l\}$ berechnet. Das entstehende Gleichungssystem hat $4N_l^2$ Unbekannte auf Ebene l , die für $l = 1$ in Bild 2.1 eingezeichnet sind.

Die Standard-Diskretisierung mit dem 5-Punkt Stern für Gleichung 2.2 auf der Ebene l (siehe z. B. [6, S. 202])

$$L_l = h_l^{-2} \begin{bmatrix} & & -\beta & & \\ -\alpha & & 2(\alpha + \beta) + \frac{1}{4}h_l^2 & & -\alpha \\ & & -\beta & & \end{bmatrix} \quad (2.4)$$

liefert das lineare Gleichungssystem (siehe [7, Kap. 4.2])

$$L_l u_l = f_l \quad (2.5)$$

Bezeichnet \mathfrak{R} die Menge der reellen Zahlen, so sind u_l und f_l Elemente des Vektorraumes $\mathfrak{R}^{4N_l^2}$ sowie L_l eine $4N_l^2 \times 4N_l^2$ Matrix mit folgender Gestalt:

$$L_l = h_l^{-2} \begin{bmatrix} T_l & -\beta I & & & -\beta I \\ -\beta I & T_l & -\beta I & & \\ & \ddots & \ddots & \ddots & \\ & & -\beta I & T_l & -\beta I \\ -\beta I & & & -\beta I & T_l \end{bmatrix} \quad (2.6)$$

Abbildung 2.1: Gitter für die Diskretisierung der anisotropen Gleichung mit periodischen Randbedingungen

$$T_l = \begin{bmatrix} 2(\alpha + \beta) + \frac{1}{4}h_l^2 & & -\alpha & & & & -\alpha \\ & -\alpha & 2(\alpha + \beta) + \frac{1}{4}h_l^2 & & -\alpha & & \\ & & & \ddots & & \ddots & \\ & & & & -\alpha & 2(\alpha + \beta) + \frac{1}{4}h_l^2 & -\alpha \\ -\alpha & & & & & -\alpha & 2(\alpha + \beta) + \frac{1}{4}h_l^2 \end{bmatrix} \quad (2.7)$$

Dabei ist I die $2N_l \times 2N_l$ Einheitsmatrix und T_l eine $2N_l \times 2N_l$ Matrix.

2.3 Mehrgitterverfahren

Zur Einführung in Mehrgitterverfahren siehe [6] oder [10]. Im folgenden sind die wesentlichen Begriffe zusammengefaßt.

Wenn u_l^n die n-te Näherung der exakten Lösung u_l des linearen Gleichungssystemes 2.5 bezeichnet, so ist $v_l^n = u_l - u_l^n$ der Fehler in der n-ten Näherung. Die Lösung der *Defektgleichung*

$$L_l v_l^n = f_l - L_l u_l^n = d_l^n \quad (2.8)$$

ist dann gleichbedeutend zu dem gegebenenem Problem, da die exakte Lösung nach $u_l = u_l^n + v_l^n$ berechnet werden kann. Die Größe d_l^n wird als *Defekt* bezeichnet. Statt der exakten Lösung von 2.8 kann man folgende Näherung betrachten: Der Defekt wird auf

ein gröberes Gitter projiziert (*restringiert*) und dort wird nun mit weniger Aufwand (da weniger Unbekannte) die *Grobgittergleichung*

$$L_{l-1}v_{l-1} = rd_l^n \quad (2.9)$$

gelöst. Dabei ist L_{l-1} geeignet zu wählen. Der Fehler v_{l-1} wird auf das feine Gitter interpoliert (*prolongiert*) und zur Korrektur der alten Lösung verwendet : $u_l^{n+1} = u_l^n + pv_{l-1}$. Die Schritte Defektbildung, Lösen der Grobgittergleichung und Korrigieren der Lösung werden als *Grobgitterkorrektur* bezeichnet. Kombiniert man einen Grobgitterschritt mit mehreren (ν_1) Schritten eines klassischen Iterationsverfahrens, wie etwa dem gedämpften Jacobi oder dem red-black Gauß-Seidel Verfahren, so erhält man das *Zweigitterverfahren*. Ersetzt man die exakte Lösung von 2.9 durch eine γ -fache Anwendung des selben Algorithmus auf die Grobgittergleichung und setzt diesen Prozess rekursiv bis zur Ebene 0 fort, so erhält man das *Mehrgitterverfahren*. Auf Ebene 0 wird ein exakter Löser (Gauß Elimination) für die 2×2 Unbekannten verwendet.

Speziell betrachten wir folgende Wahl der Komponenten als Standard-Mehrgitterverfahren für das lineare Gleichungssystem 2.5:

- ν_1 Glättungsschritte mit einem klassischen Iterationsverfahren (z. B. gedämpfter Jacobi oder red- black Gauß-Seidel) $\mathcal{S}_l^{\nu_1}$.

- Neun Punkt Prolongations- und Restriktionsoperatoren:

$$p = \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, r = p^* = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (2.10)$$

- Galerkin Grobgittergleichung

$$L_{l-1} = rL_l p \quad (2.11)$$

Der Mehrgitteralgorithmus lautet dann in algorithmischer Form:

Algorithmus 2.1 *Standard-Mehrgitterverfahren.*

```

PROCEDURE mgc (l : INTEGER; u, f : grid);
BEGIN
  IF l = 0 THEN u := L_0^{-1} f
  ELSE
    u := S_l^{\nu_1}(u, f); d := r(f - L_l u); v := 0;
    FOR j := 1 TO \gamma DO mgc(l - 1, v, d); END;
    u := u + pv;
  END;
END;
```

Wendet man Algorithmus 2.1 mit gedämpften Jacobi-Glätter $\gamma = 1$ und $\nu_1 = 2$ auf das Gleichungssystem 2.5 an, so erhält man die in Tabelle 2.1 angegebenen Konvergenzraten. Die Konvergenzraten wurden durch Mittelung über die Zyklen 11 bis 20 ermittelt. Schon ab $\alpha \approx 10\beta$ ist ein deutlicher Abfall in der Konvergenzrate zu erkennen. Die Ursache für dieses Verhalten wird im nächsten Abschnitt erläutert.

α	1	1	1	1	1	1	1	1	1	0.1	0.01
β	1	0.9	0.7	0.5	0.1	0.01	0.0001	0.000001	1	1	
ρ	0.533	0.544	0.592	0.652	0.871	0.974	0.986	0.986	0.871	0.974	

Tabelle 2.1: Konvergenzraten für ein Standard-Mehrgitterverfahren angewandt auf die anisotrope Gleichung

2.4 Fourieranalyse des Standardverfahrens

Mit Hilfe der Fourieranalyse kann man den Spektralradius (betragsgrößter Eigenwert einer Matrix, asymptotische Konvergenzrate) bzw. die Spektralnorm (siehe unten) von Zweigitteriterationen theoretisch exakt bestimmen. Voraussetzung hierfür ist ein Differenzenstern mit konstanten Koeffizienten. Ist er außerdem symmetrisch, kann man Dirichlet Randbedingungen untersuchen, sonst muß man sich auf periodische Randbedingungen beschränken. Da bei der Frequenzerlegungsmethode mit Dirichlet Randbedingungen die Koeffizienten auf dem groben Gitter nicht mehr konstant sind, betrachten wir ausschließlich periodische Randbedingungen. Die Notation wurde weitestgehend aus [6, Kap. 2.4, Kap. 8] und [8] übernommen.

Wir beginnen mit der allgemeinen Form eines *linearen Iterationsverfahrens* zur Lösung eines linearen Gleichungssystems $L_l u_l = f_l$ (der Index l bezieht sich auf die im Diskretisierungsprozeß verwendete Gitterweite h_l). Die neue diskrete Näherungslösung u_l^{n+1} wird aus der alten Lösung u_l^n durch eine lineare Beziehung berechnet:

$$u_l^{n+1} = M_l u_l^n + K_l f_l \quad (2.12)$$

M_l heißt *Iterationsmatrix* des Verfahrens. Der Fehler v_l^n in der n -ten Näherungslösung ist dann $v_l^n = u_l - u_l^n$, wobei u_l die exakte Lösung des linearen Gleichungssystems bezeichnet. Für den Fehler gilt die einfache Beziehung

$$v_l^{n+1} = M_l v_l^n \quad (2.13)$$

(siehe [13, S. 59]). Ist der Spektralradius $\rho(M_l) < 1$ und $(I - M)$ nicht singulär, so konvergiert das Verfahren gegen die eindeutige Lösung u_l ([13, Theorem 1.4]). Gilt sogar $\|M_l\| < 1$ so konvergiert das Verfahren wegen

$$|v_l^{n+1}| \leq \|M_l\| |v_l^n| \quad (2.14)$$

monoton, d. h. der Fehler wird in jeder Iteration echt kleiner. Dabei bezeichnet $|\cdot|$ die euklidische Norm und $\|\cdot\|$ die zur euklidischen Norm verträgliche Matrixnorm $\|M_l\| =$

$\sup_{x \neq 0} \frac{|M_l x|}{|x|}$ (Spektralnorm).

Die Zweigitteriteration (ein Spezialfall der obigen Mehrgitteriteration, in dem die Grobgittergleichung $L_{l-1}v_{l-1} = d_{l-1}$ exakt gelöst wird) läßt sich in der Form 2.12 darstellen. Für die Iterationsmatrix der Zweigitteriteration mit m Glättungsschritten gilt:

$$M_l^{(m)} = C_l S_l^m = (I - pL_{l-1}^{-1}rL_l)S_l^m \quad (2.15)$$

wobei der Faktor C_l die Grobgitterkorrektur und S_l den Glätter beschreibt. r und p sind Matrizen der entsprechenden Größe, die die Abbildung der feinen Werte auf grobe und umgekehrt beschreiben.

Unter den in der Einführung zu diesem Abschnitt genannten Voraussetzungen kennt man die Eigenvektoren der Matrix L_l . Für periodische Randbedingungen wie in Gleichung 2.2 und dem Diskretisierungsprozess wie in 2.2 lauten die normierten Eigenvektoren

$$(e_l^{\nu\mu})_{k,j} = \frac{1}{2} h_l e^{i\pi h_l (\nu k + \mu j)} \quad 1 - N_l \leq \nu, \mu, k, j \leq N_l \quad (2.16)$$

wobei $(e_l^{\nu\mu})_{k,j}$ dasjenige Element des Vektors $e_l^{\nu\mu}$ bezeichnen soll, das dem Gitterpunkt (k, j) entspricht. Da $e^{i\pi(\nu x + \mu y)}$ eine periodische Funktion in den Ortsvariablen x und y ist, kann man ν und μ als *Frequenzen* interpretieren.

Genauer betrachtet man den Bereich der Indizes $\mathcal{L} = (1 - N_{l-1}, N_{l-1})$ als *niederfrequent* und den Bereich $\mathcal{H} = (1 - N_l, -N_{l-1}) \cup (N_{l-1} + 1, N_l)$ als *hochfrequent*. Jedem niederfrequenten Vektor $e_l^{\nu\mu}$, $\nu, \mu \in \mathcal{L}$ ordnet man durch

$$\nu' = \begin{cases} \nu + N_l & \nu \leq 0 \\ \nu - N_l & \nu > 0 \end{cases}, \quad \mu' = \begin{cases} \mu + N_l & \mu \leq 0 \\ \mu - N_l & \mu > 0 \end{cases}, \quad \nu, \mu \in \mathcal{L} \quad (2.17)$$

jeweils einen *in x hochfrequenten* Vektor $e_l^{\nu'\mu}$, *in y hochfrequenten* Vektor $e_l^{\nu\mu'}$ sowie einen *in beiden Richtungen hochfrequenten* Vektor $e_l^{\nu'\mu'}$ zu.

Wir bezeichnen außerdem $V_{00} = \text{span}\{e_l^{\nu\mu} | \nu, \mu \in \mathcal{L}\}$ als Bereich der *niederfrequenten Fehler*, $V_{10} = \text{span}\{e_l^{\nu\mu} | \nu \in \mathcal{H}, \mu \in \mathcal{L}\}$ als Bereich der *in x hochfrequenten Fehler*, $V_{01} = \text{span}\{e_l^{\nu\mu} | \nu \in \mathcal{L}, \mu \in \mathcal{H}\}$ als Bereich der *in y hochfrequenten Fehler* sowie $V_{11} = \text{span}\{e_l^{\nu\mu} | \nu, \mu \in \mathcal{H}\}$ als Bereich der *in beiden Richtungen hochfrequenten Fehler*.

Der Grundgedanke der Fourieranalyse ist es die Iterationsmatrix $M_l^{(m)}$ des Zweigitterverfahrens durch Transformation mit einer geeigneten Matrix Q_l auf eine Gestalt zu bringen, in der man $\rho(M_l^{(m)})$ bzw. $\|M_l^{(m)}\|$ einfach bestimmen kann. Wählt man für Q_l eine bestimmte Anordnung der Eigenvektoren von L_l :

$$Q_l = [\dots, e_l^{\nu\mu}, e_l^{\nu'\mu}, e_l^{\nu\mu'}, e_l^{\nu'\mu'}, \dots] \quad \nu, \mu \in \mathcal{L} \quad (2.18)$$

und die Komponenten der Iteration wie in Abschnitt 2.3 so gilt:

$$\hat{M}_l^{(m)} = Q_l^{-1} M_l^{(m)} Q_l \quad (2.19)$$

mit $\hat{M}_l^{(m)}$ einer Blockdiagonalmatrix. Genauer ist

$$\begin{aligned} \hat{M}_l^{(m)} &= \begin{bmatrix} M^{1-N_{l-1}, 1-N_{l-1}} & & & & \\ & \ddots & & & \\ & & M^{\nu\mu} & & \\ & & & \ddots & \\ & & & & M^{N_{l-1}, N_{l-1}} \end{bmatrix} \\ &= \text{blockdiag}\{\dots, M^{\nu\mu}, \dots\}, \nu, \mu \in \mathcal{L} \end{aligned} \quad (2.20)$$

mit 4×4 Matrizen $M^{\nu\mu}$. In dieser Form lassen sich die gesuchten Größen wegen [6, Theorem 8.1.7] bestimmen:

$$\rho(M_l^{(m)}) = \rho(\hat{M}_l^{(m)}) = \max\{\rho(M^{\nu\mu}) | \nu, \mu \in \mathcal{L}\} \quad (2.21)$$

$$\|M_l^{(m)}\| = \|\hat{M}_l^{(m)}\| = \max\{\|M^{\nu\mu}\| | \nu, \mu \in \mathcal{L}\} \quad (2.22)$$

Die Analyse einer $4N_l^2 \times 4N_l^2$ Matrix wird damit auf die Analyse von 4×4 Matrizen reduziert. Auch diese werden meist noch mit einem numerischen Eigenwertlöser ausgewertet.

In 2.18 ist die Reihenfolge der Eigenwerte wesentlich. So wurde jeweils ein niederfrequenter Vektor $e_l^{\nu\mu}$ mit seinen drei zugeordneten hochfrequenten Vektoren $e_l^{\nu'\mu}, e_l^{\nu\mu'}, e_l^{\nu'\mu'}$ zusammen angeordnet. Wegen der Blockgestalt von $\hat{M}_l^{(m)}$ gilt, daß jeder Vektor $v \in E_l^{\nu\mu} = \text{span}\{e_l^{\nu\mu}, e_l^{\nu'\mu}, e_l^{\nu\mu'}, e_l^{\nu'\mu'}\}$ durch die Matrix $M_l^{(m)}$ wieder nach $E_l^{\nu\mu}$ abgebildet wird. $E_l^{\nu\mu}$ bildet einen *invarianten* Unterraum unter der Abbildung $M_l^{(m)}$. Wie der Unterraum $E_l^{\nu\mu}$ abgebildet wird, beschreibt die 4×4 Matrix $M^{\nu\mu}$.

Zum Beweis der Blockgestalt der Matrix $\hat{M}_l^{(m)}$ setzt man 2.15 in 2.19 ein und erweitert zu:

$$\begin{aligned} Q_l^{-1} M_l^{(m)} Q_l &= (I - Q_l^{-1} p Q_{l-1} \quad Q_{l-1}^{-1} L_{l-1}^{-1} Q_{l-1} \quad Q_{l-1}^{-1} r Q_l \quad Q_l^{-1} L_l Q_l) \quad Q_l^{-1} S_l^m Q_l \\ &= (I - \hat{p} \hat{L}_{l-1}^{-1} \hat{r} \hat{L}_l) \hat{S}_l^m \end{aligned} \quad (2.23)$$

Nun bleibt die Diagonalgestalt der Faktoren \hat{p} , \hat{L}_{l-1}^{-1} , \hat{r} , \hat{L}_l und \hat{S}_l zu zeigen. Dies erfolgt unter Benutzung von $Q_l^{-1} = Q_l^H$ (hermitesche Matrix) durch konsequentes einsetzen und

zusammenfassen (Hinweis: in Q_{l-1} sind die Eigenvektoren lexikographisch erst in ν , dann in μ geordnet). Wegen der Länge der Rechnungen werden im Folgenden nur die Ergebnisse bei Anwendung des Zweigitterverfahrens mit den Komponenten aus Abschnitt 2.3 auf die anisotrope Gleichung 2.2 angegeben.

Für das Produkt $Q_l^{-1}L_lQ_l$ gilt:

$$\hat{L}_l = Q_l^{-1}L_lQ_l = \text{blockdiag}\{\dots, L_l^{\nu\mu}, \dots\}, \quad \nu, \mu \in \mathcal{L} \quad (2.24)$$

mit

$$L_l^{\nu\mu} = \frac{1}{4}I + 4h_l^{-2} \begin{bmatrix} \alpha s_\nu^2 + \beta s_\mu^2 & 0 & 0 & 0 \\ 0 & \alpha c_\nu^2 + \beta s_\mu^2 & 0 & 0 \\ 0 & 0 & \alpha s_\nu^2 + \beta c_\mu^2 & 0 \\ 0 & 0 & 0 & \alpha c_\nu^2 + \beta c_\mu^2 \end{bmatrix} \quad (2.25)$$

wobei $s_\nu^2 = \sin^2\left(\frac{\nu\pi h_l}{2}\right)$, $c_\nu^2 = \cos^2\left(\frac{\nu\pi h_l}{2}\right)$ und s_μ^2, c_μ^2 entsprechend definiert sind. \hat{L}_l hat natürlich Diagonalgestalt, da $e_l^{\nu\mu}$ die Eigenvektoren von \hat{L}_l sind.

Als Glätter ist das gedämpfte Jacobi Verfahren am einfachsten zu analysieren. Beim Jacobi Verfahren wird die Matrix des zu lösenden Gleichungssystems zerlegt in

$$L_l = D - R \quad (2.26)$$

mit der Hauptdiagonalen D von L_l . Die Iterationsvorschrift lautet dann:

$$u_l^{n+1} = \omega D^{-1}Ru_l^n + (1 - \omega)u_l^n + \omega D^{-1}f_l \quad (2.27)$$

Die Iterationsmatrix des gedämpften Jacobi Verfahrens ist damit $S_l = \omega D^{-1}R + (1 - \omega)I$. Im Falle konstanter Diagonalelemente $d_{i,i}$ läßt sie sich umformen in

$$S_l = I - \frac{\omega}{d_{i,i}}L_l \quad (2.28)$$

und hat damit die gleichen Eigenvektoren wie L_l . Mit 2.24 folgt somit:

$$\hat{S}_l = Q_l^{-1}S_lQ_l = \text{blockdiag}\{\dots, S_l^{\nu\mu}, \dots\}, \quad \nu, \mu \in \mathcal{L} \quad (2.29)$$

$$S_l^{\nu\mu} = I - \frac{\omega h_l^2}{2(\alpha + \beta) + \frac{1}{4}h_l^2} L_l^{\nu\mu} \quad (2.30)$$

$$= \left(1 - \frac{\omega}{8(\alpha + \beta)h_l^{-2} + 1}\right) I \quad (2.31)$$

$$-\frac{4\omega}{2(\alpha + \beta) + \frac{1}{4}h_l^2} \begin{bmatrix} \alpha s_\nu^2 + \beta s_\mu^2 & 0 & 0 & 0 \\ 0 & \alpha c_\nu^2 + \beta s_\mu^2 & 0 & 0 \\ 0 & 0 & \alpha s_\nu^2 + \beta c_\mu^2 & 0 \\ 0 & 0 & 0 & \alpha c_\nu^2 + \beta c_\mu^2 \end{bmatrix}$$

Die Matrix r , die den Übergang $\Omega_l \rightarrow \Omega_{l-1}$ beschreibt, ist eine $N_l^2 \times 4N_l^2$ Matrix. Entsprechend sind die $r^{\nu\mu}$ Blöcke der Größe 1×4 .

$$\hat{r} = Q_{l-1}^{-1} r Q_l = \text{blockdiag}\{\dots, r^{\nu\mu}, \dots\}, \quad \nu, \mu \in \mathcal{L} \quad (2.32)$$

$$r^{\nu\mu} = \frac{1}{2} [c_\nu^2 c_\mu^2 \quad s_\nu^2 c_\mu^2 \quad c_\nu^2 s_\mu^2 \quad s_\nu^2 s_\mu^2] \quad (2.33)$$

wegen $p = 4r^T$ gilt auch $\hat{p} = 4\hat{r}^T$ und damit

$$\hat{p} = Q_l^{-1} p Q_{l-1} = \text{blockdiag}\{\dots, p^{\nu\mu}, \dots\}, \quad \nu, \mu \in \mathcal{L} \quad (2.34)$$

$$p^{\nu\mu} = 2 \begin{bmatrix} c_\nu^2 c_\mu^2 \\ s_\nu^2 c_\mu^2 \\ c_\nu^2 s_\mu^2 \\ s_\nu^2 s_\mu^2 \end{bmatrix} \quad (2.35)$$

Die Grobgittermatrix wird durch den Galerkinansatz bestimmt:

$$\hat{L}_{l-1} = \hat{r} \hat{L}_l \hat{p} = \text{blockdiag}\{\dots, L_{l-1}^{\nu\mu}, \dots\}, \quad \nu, \mu \in \mathcal{L} \quad (2.36)$$

$$\begin{aligned} L_{l-1}^{\nu\mu} &= r^{\nu\mu} L_l^{\nu\mu} p^{\nu\mu} \\ &= c_\nu^4 c_\mu^4 \left[4h_l^{-2} (\alpha s_\nu^2 + \beta s_\mu^2) + \frac{1}{4} \right] + s_\nu^4 c_\mu^4 \left[4h_l^{-2} (\alpha c_\nu^2 + \beta s_\mu^2) + \frac{1}{4} \right] \\ &\quad + c_\nu^4 s_\mu^4 \left[4h_l^{-2} (\alpha s_\nu^2 + \beta c_\mu^2) + \frac{1}{4} \right] + s_\nu^4 s_\mu^4 \left[4h_l^{-2} (\alpha c_\nu^2 + \beta c_\mu^2) + \frac{1}{4} \right] \end{aligned} \quad (2.37)$$

Multipliziert man $\hat{M}_l = (I - \hat{p} \hat{L}_{l-1}^{-1} \hat{r} \hat{L}_l) \hat{S}_l^m$ zusammen, erkennt man für die Blockmatrizen:

$$M^{\nu\mu} = \left(I_{4 \times 4} - p^{\nu\mu} (L_{l-1}^{\nu\mu})^{-1} r^{\nu\mu} L_l^{\nu\mu} \right) (S_l^{\nu\mu})^m \quad \nu, \mu \in \mathcal{L} \quad (2.38)$$

Da $L_{l-1}^{\nu\mu}$ ein Skalar ist, läßt sich die Inversion leicht durchführen. Für jeden der N_l^2 Blöcke $M^{\nu\mu}$ sind dann die Eigenwerte (evtl. numerisch) zu berechnen, um den Spektralradius der Iterationsmatrix M_l angeben zu können.

Mit der Fourieranalyse kann jetzt das Fehlverhalten der Standardmehrgitteriteration in Tabelle 2.1 erklärt werden. Nach der Mehrgitteridee sollen niederfrequente Fehler

$v \in V_{00}$ durch die Grobgitterkorrektur und hochfrequente Fehler $v \in V_{10} \cup V_{01} \cup V_{11}$ durch den Glätter beseitigt werden. Die Wirkung des Glätters wird durch $S_l^{\nu\mu}$ beschrieben. Betrachten wir $S_l^{\nu\mu}$ für große l :

$$\lim_{l \rightarrow \infty} S_l^{\nu\mu} = S^{\nu\mu} = I - \frac{4\omega}{2(\alpha + \beta)} \begin{bmatrix} \alpha s_\nu^2 + \beta s_\mu^2 & 0 & 0 & 0 \\ 0 & \alpha c_\nu^2 + \beta s_\mu^2 & 0 & 0 \\ 0 & 0 & \alpha s_\nu^2 + \beta c_\mu^2 & 0 \\ 0 & 0 & 0 & \alpha c_\nu^2 + \beta c_\mu^2 \end{bmatrix} \quad (2.39)$$

im anisotropen Fall $\alpha = 1, \beta = 0$ sowie $\omega = \frac{1}{2}$ vereinfacht sich dies zu

$$(S^{\nu\mu})^m = \begin{bmatrix} c_\nu^2 & 0 & 0 & 0 \\ 0 & s_\nu^2 & 0 & 0 \\ 0 & 0 & c_\nu^2 & 0 \\ 0 & 0 & 0 & s_\nu^2 \end{bmatrix}^m \quad (2.40)$$

Für $1 - N_{l-1} \leq \nu \leq N_{l-1}$ ist $s_\nu^2 \leq \frac{1}{2}$ und $\frac{1}{2} \leq c_\nu^2 \leq 1$. Demnach werden Fehler $v \in V_{10} \cup V_{11}$ durch den Faktor s_ν^{2m} gut gedämpft aber die Fehler $v \in V_{00} \cup V_{01}$ wegen dem Faktor c_ν^{2m} kaum gedämpft, da $c_\nu \rightarrow 1$ für kleine ν . Die Fehler aus dem Bereich V_{00} werden durch die Grobgitterkorrektur erfaßt, die Fehler $v \in V_{01}$ hingegen werden *weder durch die Grobgitterkorrektur noch vom Glätter erfaßt*! Im Fall $\beta = 1$ und $\alpha = 0$ würde der Jacobi Glätter im Bereich V_{10} versagen. Diese Aussagen gelten genauso für punktweisen Gauß-Seidel Glätter.

2.5 Definition der Robustheit

Nach diesen theoretischen Betrachtungen können wir den Begriff der Robustheit formal definieren. Die Definition ist Wittum [15, S.2] entnommen:

Definition 2.1 Sei $M_l(\alpha, \beta)$ die Iterationsmatrix eines Mehrgitterverfahrens zur Lösung der anisotropen Gleichung 2.2. Ist $\rho_l(\alpha, \beta)$ der Spektralradius von $M_l(\alpha, \beta)$, so heißt das Verfahren robust für die anisotrope Gleichung, genau dann wenn

$$\rho_l(\alpha, \beta) \leq \rho < 1 \quad \forall \alpha, \beta \geq 0, l \in \mathbf{N} \quad (2.41)$$

In dieser Definition wird verlangt, daß die Konvergenzrate des Verfahrens unabhängig von den Parametern α, β und der Gitterweite h echt kleiner als eins beschränkt bleibt.

Entsprechend den beiden Hauptkomponenten des Mehrgitterverfahrens, Glättung und Grobgitterkorrektur, gibt es prinzipiell zwei Ansatzpunkte Robustheit zu erreichen: Einsatz von komplizierteren Glättern wie etwa dem ILU Verfahren oder eine kompliziertere Grobgitterkorrektur wie bei der ab dem nächsten Kapitel beschriebenen Frequenzzzerlegungsmethode. Die Robustheit des Standardmehrgitterverfahrens mit ILU-Glättung für die anisotrope Gleichung wurde von Wittum in [15] bewiesen. Eine weitere Möglichkeit ist der Einsatz eines Linienrelaxationsverfahrens als Glätter. Auch diese Kombination ist robust für die anisotrope Gleichung. æ

Kapitel 3

Die Frequenzzzerlegungsmethode für periodische Randbedingungen

Zunächst werden die neuartigen Korrekturen vorgestellt und dann erklärt auf welche man verzichten kann. Die Erweiterung der Fourieranalyse für die neuen Korrekturen bildet den nächsten Abschnitt. Nach den Komplexitätsabschätzungen für den Algorithmus werden die praktischen Ergebnisse vorgestellt.

3.1 Mehrfache Grobgitterkorrektur

Im Frequenzzzerlegungsverfahren nach Hackbusch [8] sollen nicht nur die niederfrequenten Fehler $v \in V_{00}$ auf größeren Gittern beseitigt werden, sondern Fehler aus allen vier Bereichen. Dazu wird für jeden Bereich $V_\iota, \iota \in I = \{00, 10, 01, 11\}$, eine Grobgitterkorrektur bestehend aus

- einem groben Gitter Ω_{l-1}^ι
- Prolongation p_ι
- Restriktion r_ι
- Grobgittergleichung : $L_{l-1}^\iota = r_\iota L_l p_\iota$

definiert. Die Komponenten für $\iota = 00$ sind identisch mit denjenigen aus Abschnitt 2.3, sie sind aber der Vollständigkeit wegen noch einmal aufgelistet. Die groben Gitter entstehen durch *verschieben* des Gitters Ω_{l-1}^{00} (siehe Abb. 3.1) :

$$\Omega_{l-1}^{00} = \{(kh_{l-1}, jh_{l-1}) | 1 - N_{l-1} \leq k, j \leq N_{l-1}\} \quad (3.1)$$

Abbildung 3.1: grobe Gitter bei der Frequenzerlegungsmethode mit periodischen Randbedingungen

$$\Omega_{l-1}^{10} = \{(x - h_l, y) | x, y \in \Omega_{l-1}^{00}\} \quad (3.2)$$

$$\Omega_{l-1}^{01} = \{(x, y - h_l) | x, y \in \Omega_{l-1}^{00}\} \quad (3.3)$$

$$\Omega_{l-1}^{11} = \{(x - h_l, y - h_l) | x, y \in \Omega_{l-1}^{00}\} \quad (3.4)$$

Als Prolongationen werden verwendet:

$$\begin{aligned} p_{00} &= \frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, & p_{10} &= \frac{1}{4} \begin{bmatrix} -1 & 2 & -1 \\ -2 & 4 & -2 \\ -1 & 2 & -1 \end{bmatrix} \\ p_{01} &= \frac{1}{4} \begin{bmatrix} -1 & -2 & -1 \\ 2 & 4 & 2 \\ -1 & -2 & -1 \end{bmatrix}, & p_{11} &= \frac{1}{4} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \end{aligned} \quad (3.5)$$

Für die Restriktionen gilt wie üblich

$$r_l = \frac{1}{4} p_l^T \quad (3.6)$$

Mit diesen Prolongationen gelingt es, hochfrequente Fehler auf das feine Gitter zu projizieren. Beispielsweise ergibt die Anwendung von p_{10} auf ein konstantes 1 Gitter die in Abbildung 3.2 dargestellte Fehlerfunktion aus V_{10} .

Damit mit den angegebenen Korrekturen alle möglichen Fehlerfunktionen korrigiert werden, müssen durch die Prolongationen auch alle erreicht werden können:

Abbildung 3.2: Fehlerfunktion bei Anwendung von p_{10} auf eine konstante 1 Funktion

$$\bigcup_{\iota \in I} \text{range}(p_\iota) = \mathfrak{R}^{4N_\iota^2} \quad (3.7)$$

Durch das Verschieben der Gitter in obiger Weise wird erreicht, daß die Bildräume von paarweise verschiedenen Prolongationen orthogonal zueinander sind:

$$p_\iota^T p_\kappa = 0 \quad \forall \iota \neq \kappa \quad (3.8)$$

Durch ein Dimensionsargument zeigt man dann 3.7 (für genaue Beweise siehe [8]). Gl. 3.8 ist äquivalent zu

$$r_\iota p_\kappa = 0 \quad \forall \iota \neq \kappa \quad (3.9)$$

Dies bedeutet: ein Fehler der Form $p_\kappa v_\kappa$ kann nur auf Gitter Ω_{i-1}^κ dargestellt werden, auf den anderen nicht.

Die Zweigitteriteration besteht nun, entsprechend dem Vorgehen beim Mehrgitterverfahren, aus

- ν_1 Glättungsschritte mit einem gedämpften Jacobi oder punkweisen Gauß-Seidel Glätter.
- Defektberechnung $d_i = f_i - L_i u_i$
- Lösen der vier Grobgittergleichungen $L_{i-1}^\iota v_\iota = r_\iota d_i$

- Korrektur $u_{l-1} = u_{l-1} + \sum_{i \in I} p_i v_i$
- Nachkorrektur durch weitere ν_2 Glättungsschritte

Ausgehend von der Zweigitteridee erhält man durch rekursive Anwendung des Algorithmus statt der exakten Lösung der Grobgittergleichungen die Mehrgitterversion. Die vier Grobgittergleichungen L_{l-1}^l erzeugen im nächsten Schritt 16 Grobgittergleichungen $L_{l-2}^{l^k}$. Die verschobenen Gitter werden rekursiv konstruiert: Sei ein Gitter

$$\Omega_r = \{(x, y) | x = x_1 + (k-1)h_r, y = y_1 + (j-1)h_r, -1 < x, y \leq 1\} \quad (3.10)$$

auf Ebene r gegeben. (x_1, y_1) ist die Position des Gitterpunktes $(1, 1)$. Dann sind die von diesem Gitter erzeugten vier größeren Gitter:

$$\Omega_{r-1}^l = \{(x, y) | x = x_1(l) + (k-1)h_{r-1}, y = y_1(l) + (j-1)h_{r-1}, -1 < x, y \leq 1\} \quad (3.11)$$

mit

$$(x_1(l), y_1(l)) = \begin{cases} (x_1 + h_r, y_1 + h_r) & l = 00 \\ (x_1, y_1 + h_r) & l = 10 \\ (x_1 + h_r, y_1) & l = 01 \\ (x_1, y_1) & l = 11 \end{cases} \quad (3.12)$$

In dieser Notation gilt für das feinste Gitter Ω_l :

$$\Omega_l = \{(x, y) | x = h_l + (k-1)h_l, y = h_l + (j-1)h_l, -1 < x, y \leq 1\} \quad (3.13)$$

Die Berechnung der vier Grobgitterkorrekturen kann unabhängig voneinander erfolgen. Hierin liegt einer der Ansätze zur Parallelisierung (siehe Kapitel 5). Außerdem werden die Grobgittermatrizen aus Effizienzgründen in einer Initialisierungsphase ausgerechnet. Im Falle variabler Koeffizienten entsteht dadurch allerdings ein erheblicher Speicheraufwand!

Algorithmus 3.1 *Frequenzzzerlegungsverfahren, Version 1. Es ist $I_0 \subseteq \{00, 10, 01, 11\}$ die Menge der eingeschalteten Korrekturen.*

```

PROCEDURE fdm1 (L : matrix; u, f : grid; l : INTEGER, I_0 : set);
BEGIN
  IF l = 0 THEN u := L-1f
  ELSE
    u := Slv1(L, u, f);
    d := f - Lu; v := 0;
    FOR  $\iota \in I_0$  DO
      v $\iota-1$  := 0
      FOR j := 1 TO  $\gamma$  DO fdm1(r $\iota$ Lp $\iota$ , v $\iota-1$ , r $\iota$ d, l - 1, I_0); END;
      v := v + p $\iota$ v $\iota-1$ ;
    END;
    u := u + v;
    u := Slv2(L, u, f);
  END;
END;

```

3.2 Notwendige Grobgitterkorrekturen

Im Abschnitt 2.4 wurde gezeigt, daß die Standardmehrgitteriteration für die anisotrope Gleichung im Fall $\alpha = 1, \beta = 0$ versagt, weil der Glätter die in y hochfrequenten Fehler aus V_{01} nicht erfaßt. Bei Anwendung von Algorithmus 3.1 auf die anisotrope Gleichung braucht deswegen zusätzlich zur 00-Korrektur nur noch die 01-Korrektur eingeschaltet zu werden um Robustheit zu erreichen. Die Korrekturen $\iota \in \{10, 11\}$ können bei $\alpha = 1, \beta = 0$ als unnötige Korrekturen betrachtet werden, bei $\alpha = 0, \beta = 1$ sind entsprechend alle $\iota \in \{01, 11\}$ unnötig. Es ist aber zu betonen, daß diese unnötigen Korrekturen die Konvergenzrate *nicht verschlechtern* falls sie trotzdem zugeschaltet werden.

Um zu entscheiden, welche Korrekturen für ein gegebenes Glätter-Problem Paar nötig und welche unnötig sind, ist der Glättungsprozeß sehr genau zu studieren. Es wäre deshalb vorteilhafter ein automatisches Kriterium zu besitzen das unnötige Korrekturen ausschaltet. Betrachten wir hierzu die Bildung der Grobgittergleichungen: Die Matrix L_l auf der höchsten Ebene erzeugt die vier Matrizen $L_{l-1}^{00}, L_{l-1}^{10}, L_{l-1}^{01}$ und L_{l-1}^{11} eine Ebene darunter. Diese wiederum erzeugen 16 weitere Matrizen $L_{l-2}^{\iota\kappa} = r_\kappa r_\iota L_l p_\iota p_\kappa$. Hackbusch zeigt in [8], daß die Matrizen $L_{l-2}^{\iota\kappa}$ keine konsistenten Diskretisierungen der Differentialgleichung sind, falls $\iota \neq \kappa$ und $\iota, \kappa \neq 00$ gilt. Deshalb können alle Grobgittermatrizen $L_{l-k}^{\iota_1 \iota_2 \dots \iota_k}$ weggelassen werden, für die gilt

$$\exists \iota, \kappa \in \{\iota_1, \iota_2, \dots, \iota_k\} : \iota \neq \kappa \wedge \iota, \kappa \neq 00. \quad (3.14)$$

Der Baum mit Verzweigungsgrad vier geht damit über in den *Baum der notwendigen Grobgitterkorrekturen* aus Abbildung 3.3, der im wesentlichen nur noch den Verzweigungsgrad zwei besitzt. Hat man auf Ebene k insgesamt n Grobgittergleichungen, so sind es auf Ebene $k - 1$ $2n + 2$, da nur der am weitesten links stehende Knoten vier

Abbildung 3.3: Baum der notwendigen Grobgitterkorrekturen

Söhne hat. Per Induktion zeigt man, daß es auf Ebene k

$$3 \cdot 2^{l-k} - 2 \quad (3.15)$$

Grobgittergleichungen gibt. Dieses Resultat wird im Abschnitt über Komplexität benötigt.

Mit einem zusätzlichen Parameter $\iota \in I_0$ realisieren wir den Frequenzerlegungsalgorithmus der nur noch die notwendigen Grobgitterkorrekturen bearbeitet.

Algorithmus 3.2 *Frequenzerlegungsverfahren, Version 2 mit notwendigen Grobgitterkorrekturen. Beim Aufruf des Algorithmus auf Ebene l ist $\iota = 00$ anzugeben. Durch Vorgabe der Menge $I_0 \subseteq \{00, 10, 01, 11\}$ können vom Problem her unnötige Korrekturen abgeschaltet werden.*

```

PROCEDURE fdm2 (L : matrix; u, f : grid; l : INTEGER;  $\iota$  : freq,I0 : set);
BEGIN
  IF l = 0 THEN u := L-1f
  ELSE
    u := Sl $\iota$ (L, u, f);
    d := f - Lu; v := 0;
    IF  $\iota$  = 00 THEN
      FOR  $\kappa \in I_0$  DO
        vl-1 := 0
        FOR j := 1 TO  $\gamma$  DO fdm2(r $\kappa$ Lp $\kappa$ , vl-1, r $\kappa$ d, l - 1,  $\kappa$ , I0); END;
        v := v + p $\kappa$ vl-1;
      END
    ELSE
      vl-1 := 0
      FOR j := 1 TO  $\gamma$  DO fdm2(r00Lp00, vl-1, r00d, l - 1,  $\iota$ , I0); END;
      v := v + p00vl-1;
      vl-1 := 0
      FOR j := 1 TO  $\gamma$  DO fdm2(r $\iota$ Lp $\iota$ , vl-1, r $\iota$ d, l - 1,  $\iota$ , I0); END;
      v := v + p $\iota$ vl-1;
    END;
    u := u + v;
    u := Sl $\iota$ (L, u, f);
  END;
END;
```

Das Kriterium für die notwendigen Grobgitterkorrekturen läßt sich auf drei Dimensionen erweitern (siehe [8]). Dort gilt dann, daß $L_{l-k}^{\iota_1 \iota_2 \dots \iota_k}$ unnötig ist, wenn

$$\exists \iota, \kappa, \lambda \in \{\iota_1, \iota_2, \dots, \iota_k\} : \iota, \kappa, \lambda \text{ paarweise verschieden} \wedge \iota, \kappa, \lambda \neq 000. \quad (3.16)$$

In drei Dimensionen gibt es acht verschiedene Korrekturen, d. h. es gilt:

$$\iota, \kappa, \lambda \in \{000, 100, 010, 001, 101, 110, 011, 111\}$$

Die Gitter sind dann in x,y und z Richtung verschoben.

3.3 Komplexität

Der Aufwand für einen Standardmehrgitterzyklus ist für vernünftige Werte von γ ($\gamma \leq 2$) $O(n_l)$, d. h. proportional zur Anzahl der Unbekannten $n_l = 4N_l^2$ auf dem feinsten Gitter. Dieses Resultat werden wir zunächst herleiten und dann die Analyse auf die Frequenzzerlegungsmethode übertragen.

Wir gehen davon aus, der Aufwand für die einzelnen Komponenten auf einer Ebene k sei proportional zur Anzahl der Gitterpunkte n_k :

Anweisung	Anzahl arithmetischer Operationen	
$u_k := \mathcal{S}_k^1(u_k, f_k)$	$\leq C_S n_k, k > 0$	(3.17)
$d_k := f_l - L_k u_k$	$\leq C_D n_k, k > 0$	
$d_{k-1} := r d_k$	$\leq C_R n_k, k > 0$	
$u_k := u_k + p v_{k-1}$	$\leq C_S n_k, k > 0$	
$u_0 := L_0^{-1} f_0$	$\leq C_0$	

Wir benötigen noch eine obere Schranke für das Verhältnis $\frac{n_{l-1}}{n_l}$:

$$C_H = \sup_{l>1} \frac{n_{l-1}}{n_l} \quad (3.18)$$

Bei *standard coarsening*, d. h. $h_{l-1} = 2h_l$ ist $C_H = 2^{-d}$ mit d der Dimension des Problems. Für den Gesamtaufwand $C_l(\nu, \gamma)$ eines Standardmehrgitterzyklus auf Ebene l gilt dann:

$$C_l(\nu, \gamma) \leq \left[\sum_{k=1}^l \gamma^{l-k} (\nu C_S + C_D + C_R + C_C) C_H^{l-k} n_l \right] + \gamma^{l-1} C_0 \quad (3.19)$$

da das Gitter der Ebene k , $C_H^{l-k} n_l$ Punkte hat und γ^{l-k} mal bearbeitet wird. Der zweite Summand schätzt den Aufwand auf Ebene 0 ab.

Gl. 3.19 vereinfacht sich zu

$$C_l(\nu, \gamma) \leq (\nu C_S + C_D + C_R + C_C) n_l \sum_{k=1}^l (\gamma C_H)^{l-k} + \gamma^{l-1} C_0 \quad (3.20)$$

Für $\gamma C_H < 1$ konvergiert die Reihe für $l \rightarrow \infty$ mit dem Grenzwert $\frac{1}{1-\gamma C_H}$:

$$C_l(\nu, \gamma) \leq n_l \left[\frac{\nu C_S + C_D + C_R + C_C}{1 - \gamma C_H} + \frac{(\gamma C_H)^{l-1} C_0}{n_1} \right] = C_l n_l \quad (3.21)$$

Wenn die Konvergenzrate des Verfahrens $\rho(\nu, \gamma) \leq \rho < 1$ kleiner 1 beschränkt ist, dann löst das Verfahren *das lineare Gleichungssystem* mit n_l Unbekannten mit dem Aufwand $O(n_l)$ in 2D, wenn $\gamma \leq 3$ und in 3D, wenn $\gamma \leq 7$.

Zur Analyse der Frequenzzzerlegungsmethode ist in Gleichung 3.19 noch die Anzahl der Grobgittergleichungen zu berücksichtigen. Diese hängt davon ab wieviele Korrekturen $q = |I_0|$ eingeschaltet sind.

$$C_l(\nu, \gamma, q) \leq (\nu C_S + C_D + qC_R + qC_C)n_l \sum_{k=1}^l (\gamma q C_H)^{l-k} + (\gamma q)^{l-1} C_0 \quad (3.22)$$

Für $\gamma q C_H < 1$ konvergiert die geometrische Reihe und damit ist der Aufwand $O(n_l)$ in

- 2D für $\gamma q < 4$, d. h. bei V-Zyklen ($\gamma = 1$) mit maximal 3 Korrekturen und bei W-Zyklen ($\gamma = 2$) mit einer Korrektur.
- 3D für $\gamma q < 8$, d. h. bei V-Zyklen mit maximal 7 Korrekturen oder W-Zyklen mit höchstens 3 Korrekturen.

Sei $\gamma q C_H = 1$. Dann ist $\sum_{k=1}^l (\gamma q C_H)^{l-k} = l$. Der Aufwand

$$C_l(\nu, \gamma, q) \leq n_l \left[(\nu C_S + C_D + qC_R + qC_C)l + \frac{C_0}{n_1} \right] = n_l \left[C_l(q) \left(\frac{\log n_l}{\log 2^d} - 1 \right) + \frac{C_0}{n_1} \right] \quad (3.23)$$

ist damit $O(n_l \log n_l)$ in

- 2D für V-Zyklen mit 4 Korrekturen oder W-Zyklen mit 2 Korrekturen.
- 3D für V-Zyklen mit 8 Korrekturen oder W-Zyklen mit 4 Korrekturen.

Eine prinzipielle Verminderung des Aufwandes läßt sich dadurch erreichen, daß man nur die notwendigen Grobgitterkorrekturen berechnet. Nach Gleichung 3.15 gibt es auf Ebene k in 2D, $3 \cdot 2^{l-k} - 2$ Grobgittergleichungen. Daraus ergibt sich für den Aufwand bei $I_0 = \{00, 10, 01, 11\}$:

$$C_l(\nu, \gamma) \leq n_l(\nu C_S + C_D + 4C_R + 4C_C) \sum_{k=1}^l (\gamma C_H)^{l-k} (3 \cdot 2^{l-k} - 2) + \gamma^{l-1} (3 \cdot 2^{l-1} - 2) C_0 \quad (3.24)$$

Damit ist für $\gamma = 1$ der Algorithmus von optimaler Komplexität $O(n_l)$ und der wichtige Fall des W-Zyklus mit allen Grobgitterkorrekturen (siehe Kapitel 4) ist fast optimal, d. h. $O(n_l \log n_l)$. Hackbusch zeigt in [8], daß im dreidimensionalen mit notwendigen Grobgitterkorrekturen die Ordnung $O(n_l)$ für $\frac{3}{8}\gamma < 1$, also sogar für den W-Zyklus, gilt.

3.4 Fourieranalyse

Die Methoden aus Abschnitt 2.4 können auf die Frequenzzerlegungsmethode übertragen werden. Gleichung 2.15 muß dazu um die neuen, zusätzlichen Grobgitterkorrekturen erweitert werden:

$$M_l^{(m)} = C_l(I_0)S_l^{(m)} = \left(I - \sum_{\iota \in I_0} p_\iota (L_{l-1}^\iota)^{-1} r_\iota L_l \right) S_l^m \quad (3.25)$$

$I_0 \subseteq \{00, 10, 01, 11\}$ bezeichnet die Menge der eingeschalteten Grobgitterkorrekturen. $M_l^{(m)}$ kann wieder durch Transformation mit der Matrix Q_l auf Blockdiagonalgestalt gebracht werden. Für die 4×4 Blöcke $M^{\nu\mu}$ gilt entsprechend Gleichung 2.38:

$$M^{\nu\mu} = \left(I_{4 \times 4} - \sum_{\iota \in I_0} p_l^{\nu\mu} (L_{l-1,\iota}^{\nu\mu})^{-1} r_l^{\nu\mu} L_l^{\nu\mu} \right) (S_l^{\nu\mu})^m \quad (3.26)$$

$S_l^{\nu\mu}$ und $L_l^{\nu\mu}$ sind direkt aus Abschnitt 2.4 zu übernehmen. $p_{00}^{\nu\mu}$ und $r_{00}^{\nu\mu}$ entsprechen genau $p^{\nu\mu}$ und $r^{\nu\mu}$. Für die anderen Indizes rechnet man nach:

$$\begin{aligned} r_{00}^{\nu\mu} &= \frac{1}{2} [c_\nu^2 c_\mu^2 & s_\nu^2 c_\mu^2 & c_\nu^2 s_\mu^2 & s_\nu^2 s_\mu^2] \\ r_{10}^{\nu\mu} &= \frac{1}{2} [s_\nu^2 c_\mu^2 & -c_\nu^2 c_\mu^2 & s_\nu^2 s_\mu^2 & -c_\nu^2 s_\mu^2] \\ r_{01}^{\nu\mu} &= \frac{1}{2} [c_\nu^2 s_\mu^2 & s_\nu^2 s_\mu^2 & -c_\nu^2 c_\mu^2 & -s_\nu^2 c_\mu^2] \\ r_{11}^{\nu\mu} &= \frac{1}{2} [s_\nu^2 s_\mu^2 & -c_\nu^2 s_\mu^2 & -s_\nu^2 c_\mu^2 & c_\nu^2 c_\mu^2] \end{aligned} \quad (3.27)$$

Die $p_l^{\nu\mu}$ erhält man durch $p_l^{\nu\mu} = 4(r_l^{\nu\mu})^T$. Mit diesen Beziehungen kann man die Wirkungsweise der neuen Restriktionen im Frequenzbereich erläutern: Jeder der Vektoren $e \in \{e_l^{\nu\mu}, e_l^{\nu'\mu}, e_l^{\nu\mu'}, e_l^{\nu'\mu'}\}$ wird durch eine Restriktion r_l auf ein und den selben Vektor $e_{l-1}^{\nu\mu}$ auf dem groben Gitter abgebildet, jedoch mit unterschiedlichen Faktoren gewichtet. So wird $e_l^{\nu\mu}$ von r_{00} auf $r_{00}e_l^{\nu\mu} = \frac{1}{2}c_\nu^2 c_\mu^2 e_{l-1}^{\nu\mu}$, oder $e_l^{\nu'\mu}$ auf $r_{00}e_l^{\nu'\mu} = \frac{1}{2}s_\nu^2 c_\mu^2 e_{l-1}^{\nu\mu}$ abgebildet. Wegen $\frac{1}{2} \leq c_\nu^2, c_\mu^2 \leq 1$ und $s_\nu^2, s_\mu^2 \leq \frac{1}{2}$ werden von r_{00} vorzugsweise die niedrigen Frequenzen auf das grobe Gitter abgebildet. r_{10} bildet wegen $r_{10}e_l^{\nu'\mu} = -\frac{1}{2}c_\nu^2 c_\mu^2 e_{l-1}^{\nu\mu}$ vorzugsweise die in x hochfrequenten Fehler ab, usw. Die Minuszeichen in 3.27 ergeben sich im Übrigen durch Verschieben der Gitter.

Die Skalare $L_{l-1,\iota}^{\nu\mu}$ lassen sich mit der Beziehung $L_{l-1,\iota}^{\nu\mu} = r_l^{\nu\mu} L_l^{\nu\mu} p_l^{\nu\mu}$ bestimmen. Auf eine Darstellung sei hier verzichtet (siehe Abschnitt 4.5). Der Spektralradius der Zweigitteriteration kann dann mit Gleichung 2.22 bestimmt werden. Tabelle 3.1 zeigt die ermittelten Werte für die anisotrope Gleichung, bei verschiedenen Werten von α und β sowie unterschiedlichen Grobgitterkorrekturen (aus Hackbusch [8]). Die entsprechenden Mehrgitterraten werden im nächsten Kapitel vorgestellt.

α	β	$I_0 = \{00\}$	$I_0 = \{00, 10\}$	$I_0 = \{00, 10, 01\}$	$I_0 = \{00, 10, 01, 11\}$
1	1	0.562	0.562	0.252	0.252
$\frac{1}{2}$	2	0.809	0.359	0.228	0.228
10^{-5}	10^5	0.999	0.250	0.250	0.135

Tabelle 3.1: Zweigitterraten für die Frequenzerlegungsmethode (ged. Jacobi Glätter mit $\omega = \frac{1}{2}$, $h = \frac{1}{16}$, $\nu_1 = 2$, $\nu_2 = 0$) angewandt auf die anisotrope Gleichung

3.5 Numerische Ergebnisse

Dieser Abschnitt beinhaltet die Konvergenzraten der Frequenzerlegungsmethode *mit notwendigen Grobgitterkorrekturen* nach Algorithmus 3.2, angewandt auf die anisotrope Gleichung mit periodischen Randbedingungen. Die exakte Lösung war $u(x, y) = 0$, die u -Werte wurden mit einem Zufallszahlengenerator vorbelegt. Die festen Parameter waren $\gamma = 1$ (V-Zyklen), $\nu_1 = 2$ und $\nu_2 = 0$. Variiert wurden: der Glätter (ged. Jacobi mit $\omega = \frac{1}{2}$ oder red-black Gauß-Seidel), I_0 , h , α , β sowie das Gitter $h_l = \frac{1}{2^{l+1}}$. Zur Konvergenzmessung wurde die, auf die Anzahl der Punkte bezogene, euklidische Norm des echten Fehlers

$$v = \sqrt{\frac{1}{4N_l^2} \sum_{i,j=1-N_l}^{N_l} (u_{i,j} - u(ih_l, jh_l))^2} \quad (3.28)$$

berechnet und über die Zyklen 20 bis 25 gemittelt ($\rho = \sqrt[5]{v_{25}/v_{20}}$) um Schwankungen in den ersten Zyklen nicht zu berücksichtigen.

Die Konvergenzraten aus den Tabellen 3.2 und 3.3 bestätigen die Robustheit des Verfahrens für festes l im Sinne von Definition 2.1. Betrachtet man die Raten für festes α, β aber variables l , so bemerkt man für $\alpha = \frac{1}{10}, \beta = 10$ eine stetige Verschlechterung der Konvergenzrate für steigendes l . Hier zeigt das Verfahren *nicht* die für Mehrgitterverfahren typische Eigenschaft, daß die Konvergenzrate für beliebig großes l kleiner als eine Konstante echt kleiner eins ist. Normalerweise erreicht die Konvergenzrate bei Mehrgitterverfahren ihr Maximum schon bei relativ groben Gittern ($l \approx 4$).

Um das Verhalten des Verfahrens weiter zu studieren wurden nicht nur V- Zyklen untersucht. Eine Verallgemeinerung der Zyklenform erreicht man durch Einführung eines frequenzabhängigen Parameters $\gamma(l)$, der es erlaubt die Zweige des Baumes der notwendigen Grobgitterkorrekturen verschieden oft, je nach Frequenz, zu durchlaufen. Die Motivation dafür wird im Kapitel über die Parallelisierung nachgereicht. Wir erhalten damit Algorithmus 3.3, der auch für die Dirichlet Randbedingungen verwendet wurde.

Im folgenden wollen wir die Schreibweise $\gamma = \langle n_0, n_1, n_2, n_3 \rangle$ als Abkürzung für $\gamma(00) =$

$n_0, \gamma(10) = n_1, \gamma(01) = n_2, \gamma(11) = n_3$ verwenden. Durch Vorgabe von $\gamma = \langle 1, 0, 0, 0 \rangle$ erhält man den Standardmehrgitter V- Zyklus, $\gamma = \langle 2, 2, 2, 2 \rangle$ steht für die Frequenzzerlegungsmethode mit allen notwendigen Korrekturen und W-Zyklus. Ein Zyklus von Algorithmus 3.2 mit $I_0 = \{00, 01\}$ und $\gamma = 1$ läßt sich mit dem neuen Algorithmus durch $\gamma = \langle 1, 0, 1, 0 \rangle$ realisieren. Wie später erläutert wird, ist der Zyklus $\gamma = \langle 1, 2, 2, 2 \rangle$ für die Parallelisierung besonders interessant. Die Tabelle 3.4 enthält Konvergenzraten für die Zyklenformen $\gamma = \langle 1, 2, 2, 2 \rangle, \gamma = \langle 2, 2, 2, 2 \rangle$ und gedämpften Jacobi Glätter. Es stellte sich heraus, daß das typische Mehrgitterverhalten erst mit dem vollständigen W-Zyklus erreicht wird.

Algorithmus 3.3 *Frequenzzerlegungsverfahren, Version 3 mit notwendigen Grobgitterkorrekturen und frequenzabhängigem $\gamma(\iota)$. Beim Aufruf des Algorithmus auf Ebene l ist $\iota = 00$ anzugeben.*

```

PROCEDURE fdm3 (L : matrix; u, f : grid; l : INTEGER;  $\iota$  : freq,  $I_0$  : set);
BEGIN
  IF l = 0 THEN u :=  $L^{-1}f$ 
  ELSE
    u :=  $\mathcal{S}_l^{\nu_1}(L, u, f)$ ;
    d := f - Lu; v := 0;
    IF  $\iota = 00$  THEN
      FOR  $\kappa \in I_0$  DO
        IF  $\gamma(\kappa) > 0$  THEN
           $v_{l-1} := 0$ 
          FOR j := 1 TO  $\gamma(\kappa)$  DO fdm3( $r_\kappa L p_\kappa, v_{l-1}, r_\kappa d, l - 1, \kappa, I_0$ ); END;
          v := v +  $p_\kappa v_{l-1}$ ;
        END
      END
    ELSE
       $v_{l-1} := 0$ 
      FOR j := 1 TO  $\gamma(\iota)$  DO fdm3( $r_{00} L p_{00}, v_{l-1}, r_{00} d, l - 1, \iota, I_0$ ); END;
      v := v +  $p_{00} v_{l-1}$ ;
       $v_{l-1} := 0$ 
      FOR j := 1 TO  $\gamma(\iota)$  DO fdm3( $r_\iota L p_\iota, v_{l-1}, r_\iota d, l - 1, \iota, I_0$ ); END;
      v := v +  $p_\iota v_{l-1}$ ;
    END;
    u := u + v;
    u :=  $\mathcal{S}_l^{\nu_2}(L, u, f)$ ;
  END;
END;

```

α	β	l	$I_0 = \{00\}$	$I_0 = \{00, 10\}$	$I_0 = \{00, 01\}$	$I_0 = \{00, 10, 01\}$	$I_0 = \{00, 10, 01, 11\}$
1	1	3	0.540	0.533	0.545	0.301	0.297
		4	0.555	0.555	0.554	0.290	0.283
		5	0.550	0.549	0.550	0.291	0.284
$\frac{1}{2}$	2	3	0.795	0.369	0.795	0.369	0.366
		4	0.795	0.396	0.795	0.397	0.397
		5	0.793	0.413	0.793	0.413	0.413
2	$\frac{1}{2}$	3	0.776	0.776	0.362	0.363	0.358
		4	0.794	0.794	0.396	0.397	0.397
$\frac{1}{10}$	10	3	0.979	0.484	0.979	0.484	0.484
		4	0.979	0.644	0.979	0.644	0.644
		5	0.976	0.766	0.979	0.766	0.766
10	$\frac{1}{10}$	3	0.983	0.983	0.487	0.487	0.487
		4	0.979	0.979	0.644	0.644	0.644
10^{-5}	10^5	3	0.997	0.423	0.997	0.423	0.423
		4	0.993	0.434	0.993	0.433	0.427
		5	0.991	0.490	0.991	0.490	0.490
10^9	10^{-9}	3	0.997	0.997	0.413	0.412	0.398
		4	0.992	0.992	0.454	0.454	0.454

Tabelle 3.2: Mehrgitterraten für die Frequenzzerlegungsmethode (ged. Jacobi Glätter mit $\omega = \frac{1}{2}$, $h = \frac{1}{2^{l+1}}$, $\nu_1 = 2$, $\nu_2 = 0$, $\gamma = 1$) angewandt auf die anisotrope Gleichung mit periodischen Randbedingungen

α	β	l	$I_0 = \{00\}$	$I_0 = \{00, 10\}$	$I_0 = \{00, 01\}$	$I_0 = \{00, 10, 01\}$	$I_0 = \{00, 10, 01, 11\}$
1	1	3	0.088	0.088	0.084	0.082	0.096
		4	0.102	0.103	0.094	0.098	0.098
		5	0.121	0.121	0.108	0.111	0.110
$\frac{1}{2}$	2	3	0.396	0.172	0.396	0.176	0.201
		4	0.403	0.207	0.402	0.211	0.230
		5	0.397	0.211	0.397	0.214	0.230
$\frac{1}{10}$	10	3	0.954	0.463	0.956	0.457	0.456
		4	0.953	0.607	0.954	0.603	0.603
		5	0.944	0.712	0.946	0.709	0.709
10^{-5}	10^5	3	0.999	0.084	0.999	0.089	0.091
		4	0.999	0.169	0.998	0.169	0.169
		5	0.990	0.261	0.990	0.261	0.261

Tabelle 3.3: Mehrgitterraten für die Frequenzzerlegungsmethode (red-black Gauß-Seidel Glätter, $h = \frac{1}{2^{l+1}}$, $\nu_1 = 2$, $\nu_2 = 0$, $\gamma = 1$) angewandt auf die anisotrope Gleichung mit periodischen Randbedingungen

α	β	l	$\gamma = \langle 1, 2, 2, 2 \rangle$	$\gamma = \langle 2, 2, 2, 2 \rangle$
1	1	3	0.242	0.205
		4	0.272	0.227
		5	0.270	0.226
		6	0.273	0.225
$\frac{1}{2}$	2	3	0.330	0.272
		4	0.420	0.302
		5	0.456	0.300
		6	0.483	0.296
$\frac{1}{10}$	10	3	0.477	0.331
		4	0.652	0.357
		5	0.778	0.358
		6	0.883	0.357
10^{-5}	10^5	3	0.215	0.226
		4	0.255	0.225
		5	0.299	0.224
		6	0.316	0.220

Tabelle 3.4: Mehrgitterraten für die Frequenzerlegungsmethode (ged. Jacobi Glätter $\omega = \frac{1}{2}$, $h = \frac{1}{2^{l+1}}$, $\nu_1 = 2$, $\nu_2 = 0$) mit verschiedenen Zyklusformen, angewandt auf die anisotrope Gleichung mit periodischen Randbedingungen

æ

Kapitel 4

Die Frequenzzzerlegungsmethode für Dirichlet und von Neumann Randbedingungen

Wir beschreiben zunächst die genaue Konstruktion der groben Gitter und die effiziente Berechnung des Galerkin Produktes für den Fall variabler Koeffizienten. Nach der Vorstellung der Konvergenzraten für einige Testprobleme werden noch offene Fragen zur Frequenzzzerlegungsmethode diskutiert. Den Abschluß dieses Kapitels bildet ein neuer Ansatz zur Lösung einiger der offenen Fragen.

4.1 Die groben Gitter

Die in Gleichung 3.11 gegebene rekursive Konstruktion der groben Gitter muß für Dirichlet Randbedingungen modifiziert werden. Als Definitionsbereich betrachten wir das Einheitsrechteck $(0, 1) \times (0, 1)$, wie etwa in Problem 2.1. Sei das feinste Gitter Ω_l gegeben durch

$$\Omega_l = \{(h_l + (k - 1)h_l, h_l + (j - 1)h_l) \mid 0 \leq k, j \leq N_l, N_l = 2^{l+1}, h_l = \frac{1}{N_l}\} \quad (4.1)$$

Über das Verschieben wird wie in Gl. 3.11 mit Hilfe der Position des Gitterpunktes $(1, 1)$, (x_1, y_1) , Buch geführt. Ist dann ein Gitter

$$\begin{aligned} \Omega_r &= \{(x, y) \mid x \in \{0, x_1, \dots, x_1 + (k - 1)h_r, \dots, 1\}, \\ &\quad y \in \{0, y_1, \dots, y_1 + (j - 1)h_r, \dots, 1\}, \\ &\quad 0 \leq k \leq sx, 0 \leq j \leq sy\} \end{aligned} \quad (4.2)$$

auf Ebene r gegeben, so sind die daraus konstruierten vier gröberen Gitter:

$$\begin{aligned}\Omega_{r-1} = & \{(x, y) \mid x \in \{0, x_1(\iota), \dots, x_1(\iota) + (k-1)h_{r-1}, \dots, 1\}, \\ & y \in \{0, y_1(\iota), \dots, y_1(\iota) + (j-1)h_{r-1}, \dots, 1\}, \\ & 0 \leq k \leq sx(\iota), 0 \leq j \leq sy(\iota)\}\end{aligned}\quad (4.3)$$

mit

$$\begin{aligned}(x_1(\iota), y_1(\iota)) &= \begin{cases} (x_1 + h_r, y_1 + h_r) & \iota = 00 \\ (x_1, y_1 + h_r) & \iota = 10 \\ (x_1 + h_r, y_1) & \iota = 01 \\ (x_1, y_1) & \iota = 11 \end{cases} \\ (sx(\iota), sy(\iota)) &= \begin{cases} \left(\left\lceil \frac{sx}{2} \right\rceil, \left\lceil \frac{sy}{2} \right\rceil\right) & \iota = 00 \\ \left(\left\lceil \frac{sx+1}{2} \right\rceil, \left\lceil \frac{sy}{2} \right\rceil\right) & \iota = 10 \\ \left(\left\lceil \frac{sx}{2} \right\rceil, \left\lceil \frac{sy+1}{2} \right\rceil\right) & \iota = 01 \\ \left(\left\lceil \frac{sx+1}{2} \right\rceil, \left\lceil \frac{sy+1}{2} \right\rceil\right) & \iota = 11 \end{cases}\end{aligned}\quad (4.4)$$

Abbildung 4.1 zeigt alle Gitter auf Ebene 0 und 1 ausgehend von Ω_2 als feinstem Gitter. Wegen der fehlenden Periodizität sind die Gitterpunkte in der Nähe des Randes i.A. nicht mehr äquidistant. Die Koeffizienten dieser Punkte werden sich deshalb von denen im Inneren des Definitionsbereiches, auch bei konstanten Koeffizienten auf dem feinsten Gitter, unterscheiden. Das Galerkin Produkt berücksichtigt die unregelmäßigen Abstände der Gitterpunkte zum Rand hin automatisch.

Aus programmiertechnischen Gründen sollten die Randpunkte mit gespeichert werden. Die Randbedingungen selbst werden in die rechte Seite integriert und die entsprechenden Koeffizienten zu Randpunkten hin zu 0 gesetzt, damit das Galerkin Produkt die Ränder richtig erfaßt.

Für die nach Gleichung 4.3 konstruierten Gitter gilt außerdem, daß

$$\bigcup_{w=\iota_1\iota_2\dots\iota_k} \Omega_{l-k}^{\iota_1\iota_2\dots\iota_k} = \Omega_l \quad (4.5)$$

und

$$\left(\Omega_{l-k}^{\iota_1\iota_2\dots\iota_k} \setminus Rand\right) \cap \left(\Omega_{l-k}^{\lambda_1\lambda_2\dots\lambda_k} \setminus Rand\right) = \emptyset \quad \forall \iota_1\iota_2\dots\iota_k \neq \lambda_1\lambda_2\dots\lambda_k \quad (4.6)$$

Die Positionen der Unbekannten in den Gittern der Ebene $l-k$ sind alle verschieden und zusammen ergeben sie alle Positionen der Unbekannten auf dem feinsten Gitter.

Abbildung 4.1: grobe Gitter bei der Frequenzerlegungsmethode mit Dirichlet Randbedingungen

M. a. W. gibt es auf jeder Ebene so viele Unbekannte wie auf dem feinsten Gitter. Erst indem man nur die notwendigen Grobgitterkorrekturen berechnet, nimmt die Anzahl der Unbekannten auf den groben Gittern ab.

4.2 Berechnung des Galerkin Produktes

Wir wollen das Matrizenprodukt $L_{l-1} = rL_l p$ im Fall *variabler* Koeffizienten berechnen. Die Berechnung erfordert spezielle Techniken, da

- die Matrizen sehr dünn besetzt sind und somit der Aufwand gegenüber dem Produkt von voll besetzten Matrizen wesentlich niedriger ist
- alle drei Matrizen nicht als Matrix, sondern nur die Nichtnullelemente in gitterartiger Form gespeichert sind.

Ordnet man die Gitterpunkte lexikographisch erst in x, dann in y an, so sind L_l und L_{l-1} Neundiagonalmatrizen:

$$L_l = \begin{bmatrix} L_l^{NW}(k, j) & L_l^N(k, j) & L_l^{NE}(k, j) \\ L_l^W(k, j) & L_l^C(k, j) & L_l^E(k, j) \\ L_l^{SW}(k, j) & L_l^S(k, j) & L_l^{SE}(k, j) \end{bmatrix} \quad (4.7)$$

Sei $e_{k,j} = (0, \dots, 1, \dots, 0)^T$ der Einheitsvektor, der an der Stelle eins ist, die dem Punkt (k, j) entspricht (nicht zu verwechseln mit den Eigenvektoren $e_l^{\nu\mu}$). Das Produkt $L_{l-1}e_{k,j}$ ergibt genau die *Spalte* der Matrix L_{l-1} , die alle Koeffizienten enthält mit denen der Wert an der Stelle (k, j) verknüpft wird:

$$L_{l-1}e_{k,j} = \begin{pmatrix} \vdots \\ L_{l-1}^{NE}(k-1, j-1) \\ L_{l-1}^N(k, j-1) \\ L_{l-1}^{NW}(k+1, j-1) \\ \vdots \\ L_{l-1}^E(k-1, j) \\ L_{l-1}^C(k, j) \\ L_{l-1}^W(k+1, j) \\ \vdots \\ L_{l-1}^{SE}(k-1, j+1) \\ L_{l-1}^S(k, j+1) \\ L_{l-1}^{SW}(k+1, j+1) \\ \vdots \end{pmatrix} \quad (4.8)$$

Abbildung 4.2: Gitterausschnitt zur Erläuterung des Galerkin Produktes

Wir erhalten alle Einträge von L_{l-1} durch schrittweises Aufbauen der Produkte $rL_l pe_{k,j}$ von rechts her. Dazu betrachten wir den in Abbildung 4.2 links dargestellten Ausschnitt aus dem feinen Gitter. Der Punkt (k', j') entspricht dem Punkt (k, j) auf dem groben Gitter. Wegen dem Verschieben der Gitter ist dies nicht wie bei Standardmehrgitter der Punkt $(2k, 2j)$ sondern je nach Frequenz ι der Punkt:

$$(k', j') = \begin{cases} (2k, 2j) & \iota = 00 \\ (2k - 1, 2j) & \iota = 10 \\ (2k, 2j - 1) & \iota = 01 \\ (2k - 1, 2j - 1) & \iota = 11 \end{cases} \quad (4.9)$$

Der Vektor $pe_{k,j}$ hat dann neun Einträge wie in Bild 4.2 rechts dargestellt. Der Vektor $L_l pe_{k,j}$ ist an genau 25 Stellen ungleich Null, die den in Abbildung 4.2 gezeigten Gitterpunkten entsprechen. $rL_l pe_{k,j}$ schließlich hat die Form von Gleichung 4.8. Führt man bei den Matrix mal Vektor Produkten nur die Produkte mit Faktoren ungleich Null durch, so zählt man 134 Multiplikationen pro grobem Gitterpunkt. Eine *zeilenweise* Berechnung der Matrix L_{l-1} ist auch möglich, aber ineffizienter, da Zwischenergebnisse mehrfach berechnet werden.

4.3 Numerische Ergebnisse

Wir behandeln zunächst ausführlich die anisotrope Gleichung, da hierfür bereits die Zweigitterraten aus der Fourieranalyse und praktische Ergebnisse mit periodischen Randbedingungen vorgestellt wurden. Da die vorliegende Implementierung für Dirichlet Randbedingungen mit variablen Koeffizienten nach unseren Informationen die erste ihrer Art ist, wurden noch weitere Probleme berechnet. Die Konvektions-Diffusionsgleichung und die

Diffusionsgleichung mit stark diskontinuierlichen Koeffizienten können mit dem bisher beschriebenen Verfahren noch nicht gelöst werden. Deshalb sind die meisten gerechneten Testprobleme Gleichungen, die durch Koordinatentransformationen aus der anisotropen Gleichung entstehen. Die Testprobleme wurden in Übereinstimmung mit Horton [12] gewählt.

Alle Probleme wurden mit Algorithmus 3.3, also frequenzabhängigem γ und nur den notwendigen Grobgitterkorrekturen gerechnet. Falls nicht angegeben, wurde Gauß-Seidel red-black als Glätter verwendet mit $\nu_1 = 2, \nu_2 = 0$ — die genaue Zyklusform wird jeweils angegeben. Alle Probleme hatten als Definitionsbereich $(0, 1) \times (0, 1)$ und die exakte Lösung $u(x, y) = 0$. Startwert für die $u_{k,j}$ war 10^5 . Zur Konvergenzmessung wurden, wenn möglich, 20 Zyklen gerechnet und die auf die Anzahl der Punkte bezogene, euklidische Norm des echten Fehlers über die letzten fünf Zyklen gemittelt. Die Gitterweite war für alle Probleme $h = \frac{1}{32}$.

Problem 4.1 Anisotrope Gleichung mit Dirichlet Randbedingungen wie in Problem 2.1 definiert.

Als Diskretisierung wurde die übliche 5-Punkt Formel verwendet:

$$L_l = h_l^{-2} \begin{bmatrix} & -\beta & \\ -\alpha & 2(\alpha + \beta) & -\alpha \\ & -\beta & \end{bmatrix} \quad (4.10)$$

Die Tabellen 4.1 und 4.2 enthalten die Konvergenzraten für gedämpften Jacobi und Gauß-Seidel red-black Glätter bei verschiedenen Zyklusformen.

Die Tabellen zeigen, daß in jedem Fall der vollständige W-Zyklus $\gamma = \langle 2, 2, 2, 2 \rangle$ verwendet werden muß um für alle Werte von α und β eine Konvergenzrate unabhängig von h zu bekommen. Bei $\alpha = \frac{1}{10}$ und $\beta = 10$ ist selbst der W-Zyklus anscheinend nicht ausreichend (Dieses Verhalten ist ähnlich dem der unmodifizierten ILU Iteration, siehe [15]).

Der Aufwand für einen Zyklus der Frequenzerlegungsmethode ist sehr hoch. Um zu zeigen, daß sich dieser Aufwand dennoch lohnt betrachten wir die auf die Konvergenzrate bezogene, *normierte Rechenzeit* T_n :

$$T_n = \frac{T_{Cycle}}{-\ln \rho} \quad (4.11)$$

T_{Cycle} ist die Zeit die für eine Iteration benötigt wird und ρ die mittlere Konvergenzrate. T_n ist dann die Rechenzeit, die für eine Reduktion des Fehlers um den Faktor $1/e$ benötigt wird. In Tabelle 4.8 ist T_n für verschiedene Zyklusformen und Werte von α bzw. β

angegeben. Der Zyklus $\gamma = \langle 1, 0, 0, 0 \rangle$ entspricht dem Standardverfahren mit red-black Glätter. Trotz des hohen Aufwandes ist die Frequenzerlegungsmethode im Falle stark anisotroper Koeffizienten ca. 15 mal schneller.

Problem 4.2 Anisotrope Gleichung mit Dirichlet Randbedingungen und variablen Koeffizienten:

$$-\alpha(x, y) \frac{\partial^2}{\partial x^2} u(x, y) - \beta(x, y) \frac{\partial^2}{\partial y^2} u(x, y) = f(x, y) \quad (4.12)$$

Wobei für die Koeffizienten gilt:

$$\alpha(x, y) = 10^{\phi 2(x - \frac{1}{2})}, \quad \beta(x, y) = 10^{-\phi 2(y - \frac{1}{2})} \quad (4.13)$$

Damit variiert α über den Bereich $\alpha(0, y) = 10^{-\phi}, \dots, \alpha(1, y) = 10^{\phi}$ und β über den Bereich $\beta(x, 0) = 10^{\phi}, \dots, \beta(x, 1) = 10^{-\phi}$. In der linken, unteren Ecke des Definitionsbereiches ist $\alpha \ll \beta$, in der rechten, oberen Ecke ist $\alpha \gg \beta$. Dazwischen findet ein stetiger Übergang statt. Tabelle 4.3 zeigt Konvergenzraten und normierte Rechenzeiten für verschiedene Werte von ϕ .

Zur Diskretisierung wurde wieder die 5-Punkt Formel verwendet (siehe letztes Problem nur mit ortsabhängigem α, β).

Problem 4.3 Anisotrope Gleichung mit gemischten Randbedingungen.

Am oberen und unteren Rand wurden Dirichlet Randbedingungen angesetzt:

$$u(x, 0) = 0, \quad u(x, 1) = 0, \quad x \in (0, 1) \quad (4.14)$$

Am linken und rechten Rand wurden von Neumann Randbedingungen angesetzt:

$$\frac{\partial}{\partial x} u(0, y) = 0, \quad \frac{\partial}{\partial x} u(1, y) = 0, \quad y \in (0, 1) \quad (4.15)$$

Im Inneren des Definitionsbereiches wurde die 5-Punkt Formel verwendet, an den beiden Neumann Rändern wurden die Randbedingungen durch folgende Differenzensterne berücksichtigt:

$$L_l(h_l, y) = h_l^{-2} \begin{bmatrix} & -\beta & \\ 0 & \alpha + 2\beta & -\alpha \\ & -\beta & \end{bmatrix}, \quad L_l(1 - h_l, y) = h_l^{-2} \begin{bmatrix} & -\beta & \\ -\alpha & \alpha + 2\beta & 0 \\ & -\beta & \end{bmatrix} \quad (4.16)$$

Für $\alpha \rightarrow 0$ sind keine anderen Ergebnisse wie für die anisotrope Gleichung mit Dirichlet Randbedingungen zu erwarten. Für $\beta \rightarrow 0$ wird das lineare Gleichungssystem singulär, da die Lösung für jede Zeile des Gitters nur bis auf eine Konstante bestimmt ist. Dies äußert sich in einer Verschlechterung der Konvergenzrate. Tabelle 4.4 zeigt die Konvergenzraten für verschiedene Werte von β .

Die beiden folgenden Probleme dienen als Modellfälle für körperangepaßte Gitter.

Problem 4.4 Gedrehte, anisotrope Gleichung mit Dirichlet Randbedingungen.

Die Gleichung

$$-(\alpha c^2 + \beta s^2) \frac{\partial^2}{\partial x^2} u(x, y) - 2(\alpha - \beta)sc \frac{\partial^2}{\partial xy} u(x, y) - (\alpha s^2 + \beta c^2) \frac{\partial^2}{\partial y^2} u(x, y) = f(x, y) \quad (4.17)$$

mit $s = \sin \theta, c = \cos \theta$ beschreibt die Lösung der anisotropen Gleichung im um den Winkel θ gedrehten Einheitsquadrat.

Eine Diskretisierung 2. Ordnung für das Mischglied ist:

$$\frac{\partial^2}{\partial xy} u(x, y) = \frac{h_l^{-2}}{4} \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix} + O(h_l^2) \quad (4.18)$$

Zusammen mit der üblichen Diskretisierung der 2. Ableitung ergibt sich folgender Differenzenstern:

$$L_l = h_l^{-2} \begin{bmatrix} \frac{sc(\alpha-\beta)}{2} & -(\alpha s^2 + \beta c^2) & -\frac{sc(\alpha-\beta)}{2} \\ -(\alpha c^2 + \beta s^2) & 2(\alpha + \beta) & -(\alpha c^2 + \beta s^2) \\ -\frac{sc(\alpha-\beta)}{2} & -(\alpha s^2 + \beta c^2) & \frac{sc(\alpha-\beta)}{2} \end{bmatrix} \quad (4.19)$$

Für $\alpha = \beta$ reduziert sich der Differenzenstern auf den für die Poisson Gleichung. Erst für $\alpha \neq \beta$ und $\theta \neq 0$ ist das Mischglied verschieden von Null. Die Konvergenzraten für verschiedene Werte von α, β, θ zeigt Tabelle 4.5.

Problem 4.5 Gescherte, anisotrope Gleichung mit Dirichlet Randbedingungen.

Die Gleichung

$$-(\alpha + \beta t^2) \frac{\partial^2}{\partial x^2} u(x, y) - 2\beta t \frac{\partial^2}{\partial xy} u(x, y) - \beta \frac{\partial^2}{\partial y^2} u(x, y) = f(x, y) \quad (4.20)$$

Abbildung 4.3: Gitter zum gesicherten Problem

mit $t = \tan \theta$ entsteht aus der anisotropen Gleichung durch die lineare Koordinatentransformation

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & \tan \theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \xi \\ \psi \end{bmatrix} \quad (4.21)$$

und entspricht einer Lösung der anisotropen Gleichung auf dem in Abbildung 4.3 dargestellten Gitter. Dieses Gitter ist *nicht orthogonal*. Mit der Diskretisierung des Mischgliedes wie im letzten Problem ergibt sich der Differenzenstern:

$$L_l = h_l^{-2} \begin{bmatrix} \frac{\beta t}{2} & -\beta & -\frac{\beta t}{2} \\ -(\alpha + \beta t^2) & 2\alpha + 2\beta(1 + t^2) & -(\alpha + \beta t^2) \\ -\frac{\beta t}{2} & -\beta & \frac{\beta t}{2} \end{bmatrix} \quad (4.22)$$

Tabelle 4.7 zeigt die Konvergenzraten.

Problem 4.6 Diagonalanisotrope Gleichung mit Dirichlet Randbedingungen.

Der Differenzenstern

$$L_l = h_l^{-2} \frac{1}{2} \begin{bmatrix} -\epsilon & 0 & -1 \\ 0 & 2(1 + \epsilon) & 0 \\ -1 & 0 & -\epsilon \end{bmatrix} \quad (4.23)$$

ist für $\epsilon = 1$ eine alternative Diskretisierung 2. Ordnung der Poisson Gleichung. Viele Verfahren (ILU, alternating x,y-line Gauß- Seidel, Punktrelaxationsverfahren, ...) versagen bei $\epsilon = 1$ als Glätter für dieses Problem. In diesem Fall werden die in x und y hochfrequenten Fehler nicht gedämpft. Die Frequenzerlegungsmethode beherrscht diesen Fall sehr gut (siehe Tabelle 4.6), versagt allerdings im Fall $\epsilon \rightarrow 0$, den wiederum das ILU-Verfahren gut beherrscht, da das zu lösende Gleichungssystem tridiagonal ist.

α	β	l	$\langle 1, 0, 0, 0 \rangle$	$\langle 1, 1, 0, 0 \rangle$	$\langle 1, 1, 1, 0 \rangle$	$\langle 1, 1, 1, 1 \rangle$	$\langle 1, 2, 2, 2 \rangle$	$\langle 2, 2, 2, 2 \rangle$
1	1	3	0.539	0.539	0.251	0.257	0.249	0.222
		4	0.544	0.544	0.274	0.272	0.263	0.222
		5	0.545	0.545	0.283	0.283	0.277	0.227
		6	0.545	0.545	0.285	0.285	0.278	0.227
$\frac{1}{2}$	2	3	0.779	0.338	0.318	0.317	0.289	0.267
		4	0.785	0.367	0.368	0.368	0.348	0.286
		5	0.783	0.398	0.398	0.398	0.379	0.297
		6	0.779	0.414	0.414	0.414	0.413	0.298
$\frac{1}{10}$	10	3	0.959	0.350	0.350	0.350	0.227	0.181
		4	0.968	0.468	0.468	0.468	0.246	0.225
		5	0.970	0.557	0.557	0.557	0.347	0.309
		6	0.971	0.607	0.607	0.607	0.533	0.348
10^{-2}	10^2	3	0.969	0.358	0.358	0.358	0.246	0.183
		4	0.978	0.483	0.484	0.484	0.282	0.212
		5	0.982	0.599	0.599	0.599	0.308	0.217
		6	0.983	0.692	0.692	0.692	0.324	0.217
10^{-5}	10^5	3	0.969	0.358	0.358	0.358	0.246	0.183
		4	0.978	0.485	0.485	0.485	0.283	0.212
		5	0.982	0.602	0.602	0.602	0.311	0.217
		6	0.983	0.699	0.699	0.699	0.330	0.217

Tabelle 4.1: Mehrgitterraten für die Frequenzzerlegungsmethode (ged. Jacobi Glätter $\omega = \frac{1}{2}$, $h = \frac{1}{2^{l+1}}$, $\nu_1 = 2$, $\nu_2 = 0$) angewandt auf die anisotrope Gleichung mit Dirichlet Randbedingungen

α	β	l	$\langle 1, 0, 0, 0 \rangle$	$\langle 1, 1, 0, 0 \rangle$	$\langle 1, 1, 1, 0 \rangle$	$\langle 1, 1, 1, 1 \rangle$	$\langle 1, 2, 2, 2 \rangle$	$\langle 2, 2, 2, 2 \rangle$
1	1	3	0.058	0.058	0.052	0.083	0.083	0.082
		4	0.088	0.088	0.077	0.090	0.090	0.085
		5	0.108	0.108	0.093	0.099	0.099	0.086
		6	0.119	0.119	0.102	0.106	0.106	0.087
$\frac{1}{2}$	2	3	0.369	0.141	0.144	0.174	0.173	0.170
		4	0.390	0.181	0.183	0.208	0.207	0.187
		5	0.393	0.209	0.211	0.229	0.228	0.196
		6	0.393	0.216	0.219	0.233	0.233	0.198
$\frac{1}{10}$	10	3	0.882	0.169	0.171	0.182	0.123	0.101
		4	0.929	0.311	0.312	0.317	0.207	0.208
		5	0.938	0.421	0.422	0.425	0.348	0.295
		6	0.941	0.488	0.487	0.489	0.522	0.334
10^{-2}	10^2	3	0.925	0.159	0.164	0.168	0.107	0.048
		4	0.967	0.298	0.299	0.300	0.164	0.051
		5	0.977	0.444	0.445	0.445	0.211	0.050
		6	0.981	0.573	0.573	0.573	0.242	0.102
10^{-5}	10^5	3	0.925	0.160	0.164	0.169	0.109	0.048
		4	0.967	0.298	0.300	0.301	0.167	0.051
		5	0.977	0.445	0.445	0.445	0.218	0.051
		6	0.982	0.575	0.575	0.575	0.257	0.051

Tabelle 4.2: Mehrgitterraten für die Frequenzzerlegungsmethode (red-black Gauß-Seidel Glätter, $h = \frac{1}{2^{l+1}}$, $\nu_1 = 2$, $\nu_2 = 0$) angewandt auf die anisotrope Gleichung mit Dirichlet Randbedingungen

ϕ	$\gamma = \langle 1, 0, 0, 0 \rangle$	$\gamma = \langle 2, 2, 2, 2 \rangle$
0	0.088	0.085
0.5	0.459	0.178
1	0.789	0.229
1.2	0.848	0.236
1.4	0.885	0.239
1.6	0.908	0.240
1.8	0.923	0.240
2.0	0.933	0.240
2.4	0.909	0.236
2.6	1.117	0.233
2.8	1.526	0.984

Tabelle 4.3: Mehrgitterraten für die Frequenzzerlegungsmethode (red-black Gauß-Seidel Glätter, $h = \frac{1}{32}$, $\nu_1 = 2$, $\nu_2 = 0$) angewandt auf die anisotrope Gleichung mit variablen Koeffizienten

α	β	$\gamma = \langle 1, 0, 0, 0 \rangle$	$\gamma = \langle 2, 2, 2, 2 \rangle$
1	1	0.824	0.791
2	$\frac{1}{2}$	0.912	0.896
10	$\frac{1}{10}$	0.993	0.993
10^2	10^{-2}	0.999	0.999
10^5	10^{-5}	1.000	1.000

Tabelle 4.4: Mehrgitterraten für die Frequenzzerlegungsmethode (red-black Gauß-Seidel Glätter, $h = \frac{1}{32}$, $\nu_1 = 2$, $\nu_2 = 0$) angewandt auf die anisotrope Gleichung mit gemischten Randbedingungen

γ	α	β	Winkel θ						
			0	$\frac{\pi}{16}$	$\frac{\pi}{8}$	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$\frac{3\pi}{4}$	π
$\langle 1, 0, 0, 0 \rangle$	1	1	0.088	0.088	0.088	0.088	0.088	0.088	0.088
	2	$\frac{1}{2}$	0.390	0.347	0.251	0.236	0.390	0.236	0.389
	10	$\frac{1}{10}$	0.929	0.817	0.733	0.709	0.928	0.708	0.927
	10^2	10^{-2}	0.967	0.855	0.778	0.750	0.967	0.750	0.966
	10^5	10^{-5}	0.967	0.855	0.778	0.751	0.968	0.751	0.966
$\langle 2, 2, 2, 2 \rangle$	1	1	0.085	0.085	0.085	0.085	0.085	0.085	0.085
	2	$\frac{1}{2}$	0.187	0.186	0.183	0.178	0.187	0.178	0.187
	10	$\frac{1}{10}$	0.208	0.562	0.583	0.549	0.208	0.549	0.209
	10^2	10^{-2}	0.051	0.641	0.647	0.602	0.051	0.602	0.051
	10^5	10^{-5}	0.051	0.642	0.647	0.602	0.051	0.602	0.051

Tabelle 4.5: Mehrgitterraten für die Frequenzzerlegungsmethode (red-black Gauß-Seidel Glätter, $h = \frac{1}{32}$, $\nu_1 = 2$, $\nu_2 = 0$) angewandt auf die gedrehte, anisotrope Gleichung mit Dirichlet Randbedingungen

ϵ	$\gamma = \langle 1, 0, 0, 0 \rangle$	$\gamma = \langle 2, 2, 2, 2 \rangle$
1	0.956	0.092
$\frac{1}{2}$	0.955	0.097
$\frac{1}{10}$	0.950	0.443
10^{-2}	0.915	0.876
10^{-5}	0.953	0.947

Tabelle 4.6: Mehrgitterraten für die Frequenzzerlegungsmethode (red-black Gauß-Seidel Glätter, $h = \frac{1}{32}$, $\nu_1 = 2$, $\nu_2 = 0$) mit diagonalanisotroper Diskretisierung

γ	α	β	Winkel θ					
			0	$\frac{\pi}{16}$	$\frac{\pi}{8}$	$\frac{\pi}{4}$	$\frac{3\pi}{8}$	$\frac{49\pi}{100}$
$\langle 1, 0, 0, 0 \rangle$	1	1	0.088	0.097	0.135	0.351	0.718	0.963
	$\frac{1}{2}$	2	0.390	0.346	0.299	0.554	0.761	0.963
	$\frac{1}{10}$	10	0.929	0.819	0.718	0.740	0.778	0.963
	10^{-2}	10^2	0.967	0.856	0.752	0.751	0.778	0.963
	10^{-5}	10^5	0.967	0.857	0.752	0.751	0.778	0.963
$\langle 2, 2, 2, 2 \rangle$	1	1	0.085	0.088	0.104	0.264	0.543	0.277
	$\frac{1}{2}$	2	0.187	0.196	0.248	0.371	0.613	0.277
	$\frac{1}{10}$	10	0.208	0.569	0.597	0.588	0.646	0.277
	10^{-2}	10^2	0.051	0.641	0.645	0.602	0.647	0.277
	10^{-5}	10^5	0.051	0.642	0.645	0.602	0.647	0.277

Tabelle 4.7: Mehrgitterraten für die Frequenzerlegungsmethode (red-black Gauß-Seidel Glätter, $h = \frac{1}{32}$, $\nu_1 = 2$, $\nu_2 = 0$) angewandt auf die gescherte, anisotrope Gleichung mit Dirichlet Randbedingungen

α	β	$\gamma = \langle 1, 0, 0, 0 \rangle$	$\gamma = \langle 2, 2, 0, 0 \rangle$	$\gamma = \langle 2, 2, 2, 2 \rangle$
1	1	4.75	12.7	27.9
$\frac{1}{2}$	2	12.3	18.7	41.0
$\frac{1}{10}$	10	156.8	19.9	43.7
10^{-2}	100	344.2	10.5	23.1
10^{-5}	10^5	344.2	10.5	23.1

Tabelle 4.8: normierte Rechenzeiten für verschiedene Zyklusformen (red-black Gauß-Seidel Glätter, $h = \frac{1}{32}$, $\nu_1 = 2$, $\nu_2 = 0$) für die anisotrope Gleichung mit Dirichlet Randbedingungen

4.4 Probleme mit der Frequenzzzerlegungsmethode

Die Frequenzzzerlegungsmethode ist ein aussichtsreicher Kandidat für ein universelles, robustes Verfahren, insbesondere in 3D, und zudem sehr gut parallelisierbar (siehe nächstes Kapitel). Trotzdem gibt es noch einige Probleme mit dem Verfahren:

- Das Galerkin Produkt ist bei variablen Koeffizienten aufwendig zu berechnen. Es kostet ca. 134 Multiplikationen in 2D und 434 Multiplikationen in 3D pro grobem Gitterpunkt oder ca 1/3 eines kompletten W-Zyklus. Bei einem linearen Problem kann dies in einer Initialisierungsphase geschehen. Bei nichtlinearen Problemen sind die Grobgittergleichungen in jedem Schritt neu zu berechnen.
- Die Konvektions-Diffusionsgleichung (Problem 2.3) kann für $\epsilon \rightarrow 0$ mit dem bisher beschriebenen Verfahren *nicht* gelöst werden. Der Grund ist *kein* prinzipielles Versagen der Frequenzzzerlegungsmethode, sondern eine Eigenschaft des Galerkin Produktes, das (stabile) Upwind-Differenzen auf dem feinsten Gitter schrittweise in (instabile) Zentralfdifferenzen auf den gröbereren Gittern überführt. Dies wurde auch bei Standardmehrgitterverfahren mit Galerkin Produkten beobachtet (siehe [6, S. 221 unten]). Einen Ausweg bieten sog. matrixabhängige Prolongationen (siehe [6, 10.3], [1], [16]), deren Kombination mit der Frequenzzzerlegungsmethode allerdings nicht offensichtlich ist, da auch die Frequenzzzerlegungsmethode Forderungen an die Prolongationen stellt. Ein Versuch in diese Richtung wurde bereits von Hagemann in [9] unternommen. Ein weiterer Nachteil der matrixabhängigen Prolongationen ist, daß der ohnehin hohe Speicher- und Rechenzeitaufwand noch weiter erhöht wird.
- Bei der Lösung der Navier-Stokes'schen Gleichungen arbeitet man häufig mit finiten Volumen Diskretisierungen (siehe z.B. [3]). Es ist zu prüfen, wie sich dies mit der Frequenzzzerlegungsmethode kombinieren läßt.

Zum letzten Punkt kann man sich das lineare Gleichungssystem auf der obersten Ebene aus einer finitem Volumen Diskretisierung entstanden denken. Die eindeutige Lösung des Gleichungssystemes ist dann in jedem Fall konservativ, egal mit welchem Verfahren sie berechnet wurde.

Eine Idee für die ersten beiden Punkte ist die Folgende: Im Standardmehrgitterverfahren kann man die Grobgittergleichungen $L_k, k < l$ statt mit dem Galerkin Produkt auch durch eine Neudiskretisierung der Differentialgleichung auf Ebene k gewinnen. Wir konnten eine robuste Zweigitteriteration ohne Galerkin Produkt für die anisotrope Gleichung konstruieren, die im nächsten Abschnitt vorgestellt wird. Eine rekursive Erweiterung auf Mehrgitter funktionierte nicht und konnte im Rahmen dieser Arbeit nicht weiter untersucht werden.

4.5 Eine Modifikation

Ersetzt man in der Frequenzerlegungsmethode die Galerkin-Grobgridgleichungen durch eine Diskretisierung der ursprünglichen Differentialgleichung (im folgenden Standardansatz genannt), so stellt man zunächst Divergenz des Verfahrens fest. Kann man vielleicht durch eine einfache Modifikation des Standardansatzes diesen Defekt beheben? Wir betrachten dazu den Unterschied zwischen Standardansatz und Galerkinansatz mit der Fourieranalyse.

In der Gleichung 3.26 unterscheiden sich Standard- und Galerkinansatz nur durch unterschiedliche Skalare $L_{l-1,\iota}^{\nu\mu}$. Für den Standardansatz gilt unabhängig von der Frequenz ι mit den Definitionen für $s_\nu, s_\mu, c_\nu, c_\mu$ und \mathcal{L} aus Kapitel 2.4 :

$$L_{l-1}^{\nu\mu} = 4h_l^{-2}(\alpha s_\nu^2 c_\nu^2 + \beta s_\mu^2 c_\mu^2) + \frac{1}{4}, \quad \nu, \mu \in \mathcal{L} \quad (4.24)$$

Für den Galerkinansatz gilt hingegen:

$$L_{l-1,00}^{\nu\mu} = c_\nu^4 c_\mu^4 \lambda_l^{\nu\mu} + s_\nu^4 c_\mu^4 \lambda_l^{\nu'\mu} + c_\nu^4 s_\mu^4 \lambda_l^{\nu\mu'} + s_\nu^4 s_\mu^4 \lambda_l^{\nu'\mu'} \quad (4.25)$$

$$L_{l-1,10}^{\nu\mu} = s_\nu^4 c_\mu^4 \lambda_l^{\nu\mu} + c_\nu^4 c_\mu^4 \lambda_l^{\nu'\mu} + s_\nu^4 s_\mu^4 \lambda_l^{\nu\mu'} + c_\nu^4 s_\mu^4 \lambda_l^{\nu'\mu'} \quad (4.26)$$

$$L_{l-1,01}^{\nu\mu} = c_\nu^4 s_\mu^4 \lambda_l^{\nu\mu} + s_\nu^4 s_\mu^4 \lambda_l^{\nu'\mu} + c_\nu^4 c_\mu^4 \lambda_l^{\nu\mu'} + s_\nu^4 c_\mu^4 \lambda_l^{\nu'\mu'} \quad (4.27)$$

$$L_{l-1,11}^{\nu\mu} = s_\nu^4 s_\mu^4 \lambda_l^{\nu\mu} + c_\nu^4 s_\mu^4 \lambda_l^{\nu'\mu} + s_\nu^4 c_\mu^4 \lambda_l^{\nu\mu'} + c_\nu^4 c_\mu^4 \lambda_l^{\nu'\mu'} \quad (4.28)$$

mit den Eigenwerten von L_l :

$$\lambda_l^{\nu\mu} = 4h_l^{-2}(\alpha s_\nu^2 + \beta s_\mu^2) + \frac{1}{4}$$

$$\lambda_l^{\nu'\mu} = 4h_l^{-2}(\alpha c_\nu^2 + \beta s_\mu^2) + \frac{1}{4}$$

$$\lambda_l^{\nu\mu'} = 4h_l^{-2}(\alpha s_\nu^2 + \beta c_\mu^2) + \frac{1}{4}$$

$$\lambda_l^{\nu'\mu'} = 4h_l^{-2}(\alpha c_\nu^2 + \beta c_\mu^2) + \frac{1}{4}$$

Betrachten wir speziell die Grobgridkorrektur

$$C_l^{\nu\mu}(I_0) = I - \sum_{\iota \in I_0} p_\iota^{\nu\mu} \left(L_{l-1,\iota}^{\nu\mu} \right)^{-1} r_\iota^{\nu\mu} L_l^{\nu\mu} = I - \sum_{\iota \in I_0} \left(L_{l-1,\iota}^{\nu\mu} \right)^{-1} E_{l-1,\iota}^{\nu\mu} \quad (4.29)$$

für $\nu = \mu = 0 \Rightarrow s_\nu^2 = s_\mu^2 = 0, c_\nu^2 = c_\mu^2 = 1$ und $\iota = 00$. Damit gilt für $E_{l-1,00}^{0,0}$:

$$\begin{aligned}
E_{l-1,00}^{0,0} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{4} & 0 & 0 & 0 \\ 0 & 4h_l^{-2}\alpha + \frac{1}{4} & 0 & 0 \\ 0 & 0 & 4h_l^{-2}\beta + \frac{1}{4} & 0 \\ 0 & 0 & 0 & 4h_l^{-2}(\alpha + \beta) + \frac{1}{4} \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.30}
\end{aligned}$$

Da sowohl für den Standard- als auch für den Galerkinansatz $L_{l-1,00}^{0,0} = \frac{1}{4}$ gilt, ist

$$C_l^{0,0}(\{00\}) = I - \left(\frac{1}{4}\right)^{-1} E_{l-1,00}^{0,0} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.31}$$

D. h. ein konstanter Fehler $v \in \text{span}\{e_l^{0,0}\}$ wird von der Grobgitterkorrektur exakt korrigiert. Betrachten wir jetzt aber $\iota = 10$. Hier unterscheiden sich die Eigenwerte des Standard- und des Galerkinansatzes:

$$L_{l-1}^{0,0} = \frac{1}{4}, \text{ aber } L_{l-1,10}^{0,0} = 4h_l^{-2}\alpha + \frac{1}{4} \tag{4.32}$$

Für $E_{l-1,10}^{0,0}$ gilt dann

$$E_{l-1,10}^{0,0} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 4h_l^{-2}\alpha + \frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.33}$$

Insgesamt errechnet man für die Grobgitterkorrektur des Standardansatzes mit 4.29:

$$I - \left(\frac{1}{4}\right)^{-1} E_{l-1,10}^{0,0} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 16h_l^{-2}\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.34}$$

hingegen für den Galerkinansatz

$$I - \left(4h_l^{-2}\alpha + \frac{1}{4}\right)^{-1} E_{l-1,10}^{0,0} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.35)$$

Rechnet man zu Gleichung 4.34 noch den Glätter hinzu, zeigt sich daß $\rho(M_l^{0,0}) = 4h_l^{-2}\alpha > 1$ (bei zwei Glättungsschritten, und $\alpha > \frac{1}{4}h_l^2$) und damit die Divergenz der Zweigitteriteration. Mit dem Galerkinansatz (Gl. 4.35) hingegen werden alle Fehler $v \in \text{span}\{e_l^{N_{l-1},0}\}$ exakt korrigiert, unabhängig von α .

Ebenso analysiert man $E_{l-1,01}^{0,0}$ und $E_{l-1,11}^{0,0}$:

$$E_{l-1,01}^{0,0} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 4h_l^{-2}\beta + \frac{1}{4} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad E_{l-1,11}^{0,0} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4h_l^{-2}(\alpha + \beta) + \frac{1}{4} \end{bmatrix} \quad (4.36)$$

In Gleichung 4.34 fehlt dem Standardansatz ein Summand $4h_l^{-2}\alpha$ im Eigenwert $(0,0)$. Wir werden durch eine geeignete Modifikation den korrekten Eigenwert erzwingen, d. h. wir betrachten für $L_{l-1,10}$ den *modifizierten Standardansatz*

$$L_{l-1,10} = h_{l-1}^{-2} \begin{bmatrix} & -\beta & & \\ -\alpha & 2(\alpha + \beta) + \frac{1}{4}h_{l-1}^2 & -\alpha & \\ & -\beta & & \end{bmatrix} + h_{l-1}^{-2} \begin{bmatrix} 0 & 0 & 0 \\ 4\alpha & 8\alpha & 4\alpha \\ 0 & 0 & 0 \end{bmatrix} \quad (4.37)$$

Man errechnet für die Eigenwerte von L_{l-1} wegen $Q_{l-1}^{-1}L_{l-1}Q_{l-1} = \text{diag}\{\dots, \xi_{l-1}^{\nu\mu}, \dots\}$:

$$\xi_{l-1}^{\nu\mu} = 4h_l^{-2}(\alpha s_\nu^2 c_\nu^2 + \beta s_\mu^2 c_\mu^2) + \frac{1}{4} + 4h_l^{-2}\alpha(1 - 4s_\nu^2 c_\nu^2), \quad 1 - N_{l-1} \leq \nu, \mu \leq N_{l-1} \quad (4.38)$$

Denn die Eigenwerte von $h_{l-1}^{-2}\alpha[4 \ 8 \ 4]$ sind:

$$\begin{aligned} 4h_{l-1}^{-2}\alpha(e^{-i\pi h_{l-1}\nu} + 2 + e^{i\pi h_{l-1}\nu}) &= 8h_{l-1}^{-2}\alpha(1 + \cos \nu\pi h_{l-1}) \\ &= 16h_{l-1}^{-2}\alpha \cos^2\left(\frac{\nu\pi h_{l-1}}{2}\right) \\ &= 4h_l^{-2}\alpha \cos^2(\nu\pi h_l) = 4h_l^{-2}\alpha \left(\cos^2\left(\frac{\nu\pi h_l}{2}\right) - \sin^2\left(\frac{\nu\pi h_l}{2}\right)\right)^2 \\ &= 4h_l^{-2}\alpha(c_\nu^4 - 2c_\nu^2 s_\nu^2 + s_\nu^4) = 4h_l^{-2}\alpha(1 - 4s_\nu^2 c_\nu^2) \end{aligned} \quad (4.39)$$

α	β	$I_0 = \{00\}$	$I_0 = \{00, 10\}$	$I_0 = \{00, 01\}$	$I_0 = \{00, 10, 01\}$	$I_0 = \{00, 10, 01, 11\}$
1	1	0.562	0.562	0.562	0.311	0.307
$\frac{1}{10}$	10	0.911	0.371	0.911	0.371	0.371
10	$\frac{1}{10}$	0.911	0.911	0.371	0.371	0.371
10^{-5}	1	0.999	0.495	0.999	0.495	0.495

Tabelle 4.9: Zweigitterraten für den modifizierten Standardansatz (ged. Jacobi Glätter mit $\omega = \frac{1}{2}$, $h = \frac{1}{16}$, $\nu_1 = 2$, $\nu_2 = 0$) angewandt auf die anisotrope Gleichung

Für $\nu = \mu = 0$ gilt nun wie gefordert $\xi_{l-1}^{0,0} = 4h_l^{-2}\alpha + \frac{1}{4}$. Für $\nu \rightarrow N_{l-1}$ wird die Modifikation immer kleiner bis bei $\nu = N_{l-1}$ gilt:

$$4h_l^{-2}\alpha(1 - 4s_{N_{l-1}}^2 c_{N_{l-1}}^2) = 4h_l^{-2}\alpha \left(1 - 4 \left(\frac{\sqrt{2}}{2} \right)^2 \left(\frac{\sqrt{2}}{2} \right)^2 \right) = 0$$

Aus Symmetriegründen erhält man für die modifizierten 01 und 11 Grobgittergleichungen:

$$\mathbb{L}_{l-1,01} = h_{l-1}^{-2} \begin{bmatrix} & -\beta & & \\ -\alpha & 2(\alpha + \beta) + \frac{1}{4}h_{l-1}^2 & -\alpha & \\ & -\beta & & \end{bmatrix} + h_{l-1}^{-2} \begin{bmatrix} 0 & 4\beta & 0 \\ 0 & 8\beta & 0 \\ 0 & 4\beta & 0 \end{bmatrix} \quad (4.40)$$

$$\mathbb{L}_{l-1,11} = h_{l-1}^{-2} \begin{bmatrix} & -\beta & & \\ -\alpha & 2(\alpha + \beta) + \frac{1}{4}h_{l-1}^2 & -\alpha & \\ & -\beta & & \end{bmatrix} + h_{l-1}^{-2} \begin{bmatrix} 0 & 4\beta & 0 \\ 4\alpha & 8(\alpha + \beta) & 4\alpha \\ 0 & 4\beta & 0 \end{bmatrix} \quad (4.41)$$

Tabelle 4.9 enthält die Zweigitterraten für diesen modifizierten Standardansatz aus der Fourieranalyse. Als nächster Schritt wäre eine rekursive Modifikation zu entwickeln um die Erweiterung auf Mehrgitter vornehmen zu können. Danach könnte man mit prinzipiell den selben Mitteln die Upwind Diskretisierung der Konvektions-Diffusionsgleichung untersuchen.

Bei Verwendung des modifizierten Standardansatzes mit Dirichlet Randbedingungen werden die unregelmäßigen Abstände der Gitterpunkte zum Rand hin nicht mehr automatisch berücksichtigt. Sie müssen jetzt bei der Diskretisierung speziell behandelt werden, etwa mit dem Shortley-Weller scheme nach [6, S. 45]. Sei in 1D der Abstand des Punktes i zu seinem rechten Nachbarn h_e und zu seinem linken Nachbarn h_w , so ist der Standardstern $h_l^{-2}[-1 \ 2 \ -1]$ für die zweite Ableitung in $h_l^{-2}[-1 \frac{2}{s(1+s)} \ 2\frac{1}{s} \ -1 \frac{2}{1+s}]$, $s = \frac{h_w}{h_e}$ abzuändern. Ist $h_e \neq h_w$, so ist dies nur noch eine Approximation erster Ordnung für die zweite Ableitung. æ

Kapitel 5

Parallelisierung

Nach einer Einführung in die Parallelisierung von Mehrgitterverfahren wird eine Parallelisierungsmöglichkeit der Frequenzerlegungsmethode besprochen, die zu einem übersichtlichen Algorithmus für Hypercube-Strukturen führt. Der nächste Abschnitt erläutert einige implementierungsspezifische Details des DIRMU Programmes. Den Abschluß bilden die Speedup-Tabellen, ein Rechenzeitvergleich zwischen der Frequenzerlegungsmethode und dem Standardmehrgitterverfahren mit red-black Glätter bzw. ILU Glätter sowie eine Wertung der Ergebnisse.

5.1 Zur Parallelisierung von Mehrgitterverfahren

Zunächst führen wir zwei Begriffe ein. Der Speedup $S(p)$ eines parallelen Algorithmus ist definiert als

$$S(p) = \frac{T_{Mono}}{T_{Multi}(p)} \quad (5.1)$$

mit der Zeit T_{Mono} , die ein einzelner Rechner zur Ausführung des Algorithmus benötigt und der Zeit $T_{Multi}(p)$, die p Rechner vom gleichen Typ zur Ausführung der parallelen Form des Algorithmus benötigen. Zum Vergleich iterativer Verfahren auf dem Parallelrechner ist die Definition des Speedup allein nicht ausreichend. Wir definieren deshalb die normierte, parallele Rechenzeit:

Definition 5.1 Sei T_{Cycle} die Zeit, die ein Rechner zur Ausführung eines Zyklus eines iterativen Verfahrens benötigt, $S(p)$ der Speedup für die parallele Berechnung auf p Rechnern und ρ die Konvergenzrate des Verfahrens, so ist die normierte, parallele Rechenzeit T_{np} :

$$T_{np} = \frac{T_{Cycle}}{-\ln \rho \ S(p)} \quad (5.2)$$

Die normierte, parallele Rechenzeit ist die Zeit, die der Algorithmus auf dem Parallelrechner benötigt um den Fehler um den Faktor $1/e$ zu reduzieren. Gleichung 5.2 bewirkt, daß ein etwas weniger gut parallelisierbares aber robustes Mehrgitterverfahren für die oben genannten Modellprobleme effizienter sein kann als ein nicht robustes Mehrgitterverfahren mit einem optimal parallelisierbaren Glätter. Um dies deutlich zu machen ein Zahlenbeispiel: Für $h = 1/16$ ist nach Tabelle 2.1 die Konvergenzrate eines Standardmehrgitterverfahrens mit gedämpften Jacobi Glätter $\rho = 0.974$, bei Anwendung auf die anisotrope Gleichung 2.2 mit $\alpha = 1, \beta = 0.01$. Um die gleiche normierte, parallele Rechenzeit zu erzielen wie ein robustes Mehrgitterverfahren (mit gleichem T_{Cycle}) und Konvergenzrate $\rho = 1/2$ müßte das Standardverfahren mit dem 26 fachen Speedup parallelisiert werden wie das robuste Verfahren!

Mehrgitterverfahren bestehen aus den fünf Komponenten

1. Glätter
2. Defektbildung
3. Restriktion
4. Interpolation (Prolongation)
5. Korrektur

Bei den bisher betrachteten Fünf- bzw. Neundiagonalmatrizen sind die pro Punkt erforderlichen Rechenoperationen für alle Komponenten *streng lokal*, d. h. es werden nur Werte aus der unmittelbaren Umgebung eines Punktes benötigt. Dies gilt auch für kompliziertere Glätter wie ILU oder line Gauß-Seidel.

Die Berechnungen, die für die Komponenten 2. bis 5. pro Punkt auszuführen sind, können außerdem für alle Punkte *gleichzeitig* durchgeführt werden.

Bei den verschiedenen Glättern können nicht immer alle Punkte gleichzeitig bearbeitet werden. Tabelle 5.1 gibt einen Überblick, welche Punkte bei welchem Verfahren gleichzeitig bearbeitet werden können.

Der Parallelisierungsgrad ist für das Jacobi Verfahren am größten, aber auch das red-black Gauß-Seidel Verfahren wird auf Parallelrechnern häufig verwendet, da die bessere Konvergenzrate (für die Poisson Gleichung) den schlechteren Parallelisierungsgrad wieder ausgleicht.

Um die parallele Bearbeitung der einzelnen Gitterpunkte zu ermöglichen, wird jedem Prozessor eine Menge von Punkten zugewiesen, die er zu bearbeiten hat. Bei dieser *Datenaufteilung* sind verschiedene Aspekte zu beachten:

Verfahren	gleichzeitig zu bearbeitende Punkte
Jacobi	alle Punkte
red-black Gauß-Seidel	erst alle Punkte (k, j) mit $(k + j) \bmod 2 = 0$, dann alle mit $(k + j) \bmod 2 = 1$
ILU, Gauß-Seidel lex. (keine period. RB)	alle Punkte (k, j) mit $k + j = \text{const}$
line Gauß-Seidel	wird vom Tridiagonallöser bestimmt
zebra line Gauß-Seidel	jeweils die Hälfte aller Zeilen kann gleichzeitig gelöst werden

Tabelle 5.1: Parallelisierungsgrad von verschiedenen Glättungsalgorithmen

- **Verbindungsstruktur:** Da zur Berechnung jedes Punktes die Werte in den Nachbarpunkten erforderlich sind, werden Daten aus anderen Prozessoren benötigt. Zu diesen Prozessoren sollte, wenn möglich, eine schnelle Verbindung bestehen.
- **Lastverteilung:** Alle Prozessoren sollten möglichst für die gleiche Anzahl von Punkten zuständig sein.

Die Verbindungsstruktur des Multiprozessors ist so zu wählen, daß einerseits ein Prozessor alle Information die er von anderen Prozessoren benötigt schnell bekommen kann und andererseits auch alle Prozessoren gleichzeitig beschäftigt werden können (beachte die Bearbeitungsreihenfolge im Glätter, siehe auch [12]).

Wir betrachten, wie in den Testproblemen, den Definitionsbereich $(0, 1) \times (0, 1)$. Die gesuchten Werte entsprechen der Lösung an den Gitterpunkten (kh_l, jh_l) , $1 \leq k, j \leq N_l$, $N_l = 2^{l+1}$, $h_l = \frac{1}{N_l}$. Es sollen periodische Randbedingungen gegeben sein. Ein natürlicher Ansatz ist die Verwendung einer zweidimensionalen Prozessorstruktur mit $n' \times m'$ Prozessoren. Jeder Prozessor (x, y) soll eine Verbindung zu den Prozessoren $((x+1) \bmod n', y)$, $((x-1) \bmod n', y)$, $(x, (y+1) \bmod m')$ und $(x, (y-1) \bmod m')$ besitzen. Die Verbindungen des rechten zum linken und des oberen zum unteren Rand sind wegen der periodischen Randbedingungen nötig. Diese Konfiguration nennen wir einen *zweidimensionalen Torus*. Jeder Prozessor erhält dann $(N_l \text{ DIV } n') \times (N_l \text{ DIV } m')$ Unbekannte. Ist N_l kein Vielfaches von n' und m' , so bekommen einige Prozessoren noch zusätzliche Unbekannte und die Aufteilung wird ungleichmäßig. In Abbildung 5.1 ist die Aufteilung eines Gitters mit 16×16 Unbekannten auf einen 4×4 Torus dargestellt.

Im weiteren beschränken wir uns auf den Jacobi bzw. den red-black Gauß-Seidel Glätter, da diese für die Frequenzerlegungsmethode ausreichend sind.

Das Mehrgitterverfahren verwendet die zu Beginn aufgelisteten Komponenten auf unterschiedlich feinen Gittern. In zwei Dimensionen hat, bei Halbierung der Gitterweite, jedes Gitter höchstens ein viertel der Punkte des nächst feineren Gitters. Die erreichbaren Speedups auf jeder Ebene sind stark von der Anzahl der Unbekannten pro Prozessor

Abbildung 5.1: Aufteilung der Gitterpunkte auf die Prozessorkonfiguration

abhängig. Wir können drei verschiedene Phasen unterscheiden:

- $N_l^2 \gg n' \cdot m'$: Speedup sehr hoch, Verluste ergeben sich aus ungleicher Lastverteilung und Kommunikation mit Nachbarprozessoren (gering).
- $N_l^2 \approx n' \cdot m'$: Speedup stark von Rechnerarchitektur abhängig, da nahezu für jeden Punkt Information mit einem Nachbarprozessor ausgetauscht werden muß (Verhältnis von Synchronisations- und Kommunikationszeiten zu Rechenzeiten). Ungleiche Lastverteilung wirkt sich noch stärker aus.
- $N_l^2 < n' \cdot m'$: Wie im letzten Punkt, jedoch gibt es sogar Prozessoren, die gar keine Punkte mehr berechnen müssen.

Bei der Verwendung von Multiprozessoren mit *lokaler Nachbarschaft*, etwa obiger Torussstruktur, sind im Fall $N_l^2 < n' \cdot m'$ zusätzliche Transporte erforderlich, da nicht mehr alle notwendigen Verbindungen vorhanden sind. Trotzdem läßt sich das Standardmehrgitterverfahren sehr gut parallelisieren, wenn auf dem feinsten Gitter $N_l^2 \gg n' \cdot m'$ gilt, da die Arbeit auf den gröberen Gittern rapide (mit dem Faktor $\frac{1}{4}$) abnimmt. Insgesamt wird bei einem V-Zyklus bei sehr großen Gittern $\frac{3}{4}$ der Rechenzeit auf dem feinsten Gitter verbraucht.

Anders ist die Situation bei Multiprozessoren mit einer großen Prozessorzahl, wie etwa der Connection Machine. Dort gilt bereits auf der feinsten Ebene $N_l^2 \approx n' \cdot m'$, da $n' \cdot m' = 2^{16}$ (siehe hierzu [4]).

Einige Glätter, wie das ILU-Verfahren, sind bis jetzt nur auf einem Ring von Prozessoren effizient parallelisiert worden. Dort läßt der Speedup (pro Ebene) bereits bei $N_l \approx n' \cdot m'$

stark nach und schon relativ große Gitter können nicht mehr von allen Prozessoren bearbeitet werden (siehe [12]).

5.2 Parallele Frequenzzerlegungsmethode

Die Frequenzzerlegungsmethode kann sehr effizient parallelisiert werden, da

- Jacobi oder red-black Gauß-Seidel Glätter verwendet werden, die im Fall $N_l^2 \gg n' \cdot m'$ hohe Speedups erlauben
- im Bereich $N_l^2 < n' \cdot m'$ die sonst untätigen Prozessoren zur Lösung der zusätzlichen Gleichungen verwendet werden können

Das Grobkonzept des parallelen Algorithmus lautet also: Die Gitter der höchsten Ebene ml bis einschließlich einer Ebene dl werden auf alle Prozessoren möglichst gleichmäßig verteilt und *seriell nacheinander* berechnet. Ab Ebene $dl - 1$ werden jeweils Teile der Konfiguration für verschiedene Grobgittergleichungen verwendet, die *gleichzeitig* bearbeitet werden. Der Parameter dl soll so gewählt werden, daß spätestens auf Ebene 0 eine Grobgittergleichung nur noch von genau einem Prozessor berechnet wird.

Zur genauen Zuordnung der Grobgittergleichungen auf die Teilkonfigurationen betrachten wir den Baum der notwendigen Grobgitterkorrekturen in Bild 3.3. In diesem Baum gibt es zwei verschiedene Arten von Verzweigungen: Typ I mit vier Söhnen im am weitesten links verlaufenden Pfad und Typ II mit nur zwei Söhnen sonst. Die Implementierung wird wesentlich vereinfacht wenn wir die vierfache Verzweigung als zwei zweifache Verzweigungen betrachten und nur jeweils Paare von Grobgittergleichungen gleichzeitig berechnen. Die Verteilung der Unteraufgaben reduziert sich damit auf folgendes Problem: Ist das Gitter der Ebene k auf einem $2n' \times 2m'$ Torus verteilt, so sind die zwei Probleme der Ebene $k-1$ auf zwei $2n' \times m'$ (bzw. $n' \times 2m'$, falls $n' > m'$) Konfigurationen zu verteilen. Damit die Konfigurationen bis zur Größe 1×1 hinunter immer rekursiv in zwei gleichgroße Hälften zerlegt werden können muß der Torus von der Form $n' = 2^n, m' = 2^m$ sein. Eine solche Konfiguration kann genau $n + m$ mal halbiert werden. Wenn, wie oben gefordert, spätestens die Gleichungen auf Ebene 0 von einem Prozessor bearbeitet werden sollen, muß der Parameter $dl \geq n + m$ gewählt werden. Der Bereich $(n + m, ml)$ für dl kann zur Optimierung des Speedups benutzt werden.

Bei der Verteilung der Unteraufgaben auf je eine Hälfte entstehen Transportanforderungen, da der Defekt auf der ganzen Konfiguration gespeichert ist und der jeweilige restringierte Defekt nur in einer Hälfte benötigt wird. Nach Bearbeitung der Grobgittergleichungen müssen die beiden Korrekturen wieder auf die ganze Konfiguration verteilt werden. Bild 5.2 zeigt die notwendigen Transporte bei Halbierung der Konfiguration in x Richtung (y Richtung ist analog). Es ist nur eine Zeile von Prozessoren dargestellt, da

in jeder Zeile die selben Anforderungen bestehen. D_i und D'_i sind die Teile der beiden Gitter, die in Prozessor i gespeichert sind. Dies können entweder die beiden restriktierten Defekte $r_i d$ und $r_\kappa d$ sein oder die prolongierten Korrekturen $p_i v$ und $p_\kappa v$. Die Transportanforderungen können so reduziert werden, daß höchstens Daten aus einem im Hypercube benachbarten Prozessor geholt werden müssen. Bild 5.3 zeigt wieder die Zeile von acht Prozessoren aus Bild 5.2 nur wurden die Prozessornummern in binär eingetragen. Ordnet man jetzt die Prozessoren in Form einer Ziehharmonika an, so bilden die oberen vier und die unteren vier Prozessoren wieder einen Ring, wie aus den Prozessornummern ersichtlich ist. Außerdem bilden beide Hälften wieder einen Hypercube, da in der oberen Hälfte Bit 0 gleich 0 und in der unteren Hälfte Bit 1 gleich 1 gilt. Das Verfahren läßt sich somit rekursiv anwenden. Teilt man die Zeile so in zwei Hälften und betrachtet wieder die Transportanforderungen, so stellt man fest, daß sich alle benötigten Daten in benachbarten Prozessoren befinden (siehe Bild 5.4).

Neben den unvermeidlichen Transporten gibt es noch eine weitere Quelle für Effizienzverluste bei der Verteilung der Grobgittergleichungen $L_{l-1,\iota}$ und $L_{l-1,\kappa}$ auf Teilkonfigurationen. Jede Teilkonfiguration bearbeitet ja nicht nur eine Grobgittergleichung, sondern auch alle weiteren, die von dieser erzeugt werden. M. a. W. jede Teilkonfiguration bearbeitet einen *Teilbaum* des Baumes der notwendigen Grobgitterkorrekturen mit der Wurzel $L_{l-1,\iota}$ bzw. $L_{l-1,\kappa}$. Soll der optimale Speedup erreicht werden, so muß der Rechenaufwand für beide Teilbäume gleich groß sein (optimales *load balancing*). Ist die zu bearbeitende Verzweigung vom Typ II, so haben die beiden Teilbäume gleiche Struktur und der Rechenaufwand ist bei periodischen Randbedingungen gleich groß. Bei Dirichlet Randbedingungen ergeben sich geringe Unterschiede. Probleme bereiten die Typ I Verzweigungen. Hier werden im ersten Schritt die beiden Teilbäume der 00 und der 10 Frequenz parallel bearbeitet (siehe gestrichelter bzw. punktierter Teilbaum in Bild 5.5). Man kann in diesem Fall durch den in Algorithmus 3.3 eingeführten, frequenzabhängigen Parameter $\gamma(\iota)$ die Lastverteilung optimieren. Betrachten wir einen Zyklus der Form $\gamma = \langle 1, 2, 2, 2 \rangle$: Zählt man, wie oft Algorithmus 3.3 die Gitter jeder Ebene in den beiden Teilbäumen besucht, so ergibt sich für Ebene $l-1$ das Verhältnis 1:2, für Ebene $l-2$ das Verhältnis 7:8, für $l-3$ das Verhältnis 31:32 usw. Wie die Tabellen belegen werden ergeben sich für den $\langle 1, 2, 2, 2 \rangle$ Zyklus die besten Speedups. Allerdings wissen wir, daß man den vollen W-Zyklus verwenden muß um eine Konvergenzrate unabhängig von h zu erhalten. Der $\langle 1, 2, 2, 2 \rangle$ -Zyklus ist somit nur für die Parallelisierung interessant.

Abbildung 5.2: Transportanforderung bei acht Prozessoren pro Zeile

Abbildung 5.3: Ziehharmonikakonstruktion für acht Prozessoren

Abbildung 5.4: Transportanforderungen bei der Ziehharmonikakonstruktion

Abbildung 5.5: Aufteilung der Teilbäume einer Typ I Verzweigung

Bis jetzt waren wir von einer $2^n \times 2^m$ Torusstruktur ausgegangen. Es ist aber zu beachten, daß bei Halbierung der Konfiguration auch jede Hälfte wegen der periodischen Randbedingungen wieder einen Torus realisieren muß. Alle notwendigen Verbindungen, auch bei fortgesetzter Halbierung der Teilkonfigurationen, erhält man mit der *Hypercube* Verbindungsstruktur. Die Definition und einige Eigenschaften dieser Struktur sollen nun aufgelistet werden:

1. Eine Hypercube Struktur der Dimension d besteht aus 2^d Prozessoren. Sei $(b_{d-1}b_{d-2} \dots b_1b_0)_2 = i$ die binäre Darstellung der Zahl i . Der Prozessor mit der Nummer i ist mit allen Prozessoren $(\bar{b}_{d-1}b_{d-2} \dots b_1b_0)_2 \dots (b_{d-1}b_{d-2} \dots b_1\bar{b}_0)_2$ verbunden, die sich in genau einer Bitposition von ihm unterscheiden.
2. Aus jedem Hypercube der Dimension d kann man 2^{d_1} Hypercubes der Dimension d_2 bilden wobei $d = d_1 + d_2$ ist. Man braucht dazu nur die Binärdarstellung der Prozessornummern zu betrachten:

$$i = \underbrace{(b_{d-1}b_{d-2} \dots b_{d_2})}_{d_1} \underbrace{b_{d_2-1}b_{d_2-2} \dots b_1b_0}_{d_2}$$

Alle Prozessoren die die oberen d_1 Bits identisch haben, bilden zusammen einen d_2 dimensionalen Hypercube. Für d_1 Bits gibt es 2^{d_1} verschiedene Möglichkeiten. Die Bitstellen müssen nicht unbedingt zusammenhängend gewählt werden, es genügt eine beliebige Aufteilung der Bitstellen in Teilmengen mit d_1 bzw. d_2 Elementen.

3. In jeden Hypercube der Dimension d kann man einen $2^n \times 2^m$ Torus einlegen, wobei $d = n + m$ gilt. Dazu benötigt man sog. *gray codes*. Ein gray code der Länge l ist eine Abbildung der Menge $\{0, 1, 2, \dots, 2^l - 1\}$ in sich : $gc_l : \{0, 1, 2, \dots, 2^l - 1\} \rightarrow \{0, 1, 2, \dots, 2^l - 1\}$ mit der Eigenschaft, daß sich $gc_l(i)$ und $gc_l((i + 1) \bmod 2^l)$ in genau einer Bitstelle unterscheiden. gray codes beliebiger Länge erhält man durch folgende Rekursion:

$$gc_l(i) = \begin{cases} i & l \leq 1 \\ gc_{l-1}(i) & l > 1 \wedge i < 2^{l-1} \\ 2^{l-1} + gc_{l-1}(2^l - 1 - i) & l > 1 \wedge i \geq 2^{l-1} \end{cases} \quad (5.3)$$

Nun ordnet man der Torusposition (x, y) den Prozessor mit der Nummer $gc_n(x) + 2^n gc_m(y)$ zu. Wegen der gray code Eigenschaft und der Definition der Hypercube Verbindungsstruktur sind dann die Prozessoren $((x + 1) \bmod 2^n, y)$, $((x - 1) \bmod 2^n, y)$, $(x, (y + 1) \bmod 2^m)$ und $(x, (y - 1) \bmod 2^m)$ miteinander verbunden. Dieses Prinzip läßt sich auch auf 3D Tori übertragen.

Aus 2. und 3. folgt, daß man einen $2^n \times 2^m$ Torus in einen $n + m$ dimensionalen Hypercube einbetten kann und daß jede Hälfte für sich wieder ein Torus ist. Bevor wir den parallelisierten Frequenzzерlegungsalgorithmus formulieren, wollen wir die Hauptpunkte der Parallelisierung zusammenfassen:

- Verwendet wird eine Hypercube Struktur der Dimension d mit 2^d Prozessoren. Das feinste Gitter wird auf einen $2^n \times 2^m$ Torus aufgeteilt, wobei sich n und m um höchstens eins unterscheiden und $n + m = d$ gilt.
- Die oberste Ebene sei ml . Bis *einschließlich* der Ebene dl (Gitterweite $h = 1/2^{dl+1}$) werden die Grobgittergleichungen auf den $2^n \times 2^m$ Torus verteilt. Ab Ebene $dl - 1$ wird die Konfiguration sukzessive in 2, 4, 8, ... Teile zerteilt, so daß kleinere Gitter auf kleineren Tori berechnet werden. Gilt $dl \geq n + m$, so wird ein Problem der Ebene 0 auf jeden Fall von einem Rechner bearbeitet.
- Im Baum der notwendigen Grobgitterkorrekturen gibt es Verzweigungen vom Grad zwei und vier. Vierfache Verzweigungen werden als zwei zweifache Verzweigungen betrachtet und *nacheinander* bearbeitet. Die Lastverteilung kann durch frequenzabhängiges $\gamma(\ell)$ optimiert werden.
- Bei der Verteilung der Grobgittergleichungen sind Daten zu transportieren. Wird ein $2^n \times 2^m$ Torus in x Richtung halbiert, so entstehen zwei $2^{n-1} \times 2^m$ Tori. Die maximale Transportentfernung für eine Datenportion ist 2^{n-1} . Benutzt man eine kompliziertere Abbildung der Tori auf den Hypercube, so läßt sich die maximale Transportentfernung auf 1 reduzieren!

Ausgehend von Algorithmus 3.3 formulieren wir nun die parallele Version.

Algorithmus 5.1 *Paralleles Frequenzerlegungsverfahren. Es werden nur die notwendigen Grobgitterkorrekturen bearbeitet. Die Prozeduren `xsplit` und `ysplit` halbieren die aktuelle Konfiguration in x bzw. y Richtung. Die jeweils andere Hälfte ist danach bis zur Ausführung eines entsprechenden `xmerge` oder `ymerge` unsichtbar. Die Variable `leftordown` in jedem Prozessor wird bei Halbierung der Konfiguration auf TRUE gesetzt, falls sich der Prozessor bei Halbierung in x Richtung in der linken oder bei Halbierung in y Richtung in der unteren Hälfte befand. Die `transport` Funktion, die hier nicht näher spezifiziert werden soll, erledigt den Transport der Datenportionen in die richtigen Prozessoren.*

```

PROCEDURE parfdm ( $L$  : matrix;  $u, f$  : grid;  $l, n, m$  : INTEGER;  $\iota$  : freq);

  PROCEDURE computeinparallel ( $\kappa, \lambda$  : freq);
  BEGIN
    transport( $r_\kappa d \rightarrow d_{l-1, \kappa}, r_\lambda d \rightarrow d_{l-1, \lambda}$ );
     $n' := n; m' := m$ ; IF  $n > m$  THEN xsplit;  $n' := n - 1$  ELSE ysplit;  $m' := m - 1$  END;
    IF leftorddown THEN (* berechne  $\kappa$  *)
       $v_{l-1, \kappa} := 0$ ; IF  $\iota = 00$  THEN  $\eta := \kappa$  ELSE  $\eta := \iota$  END;
      FOR  $j := 1$  TO  $\min(\gamma(\eta), l)$  DO parfdm( $r_\kappa Lp_\kappa, v_{l-1, \kappa}, d_{l-1, \kappa}, l - 1, n', m', \eta$ ); END
    ELSE (* berechne  $\lambda$  *)
       $v_{l-1, \lambda} := 0$ ; IF  $\iota = 00$  THEN  $\eta := \lambda$  ELSE  $\eta := \iota$  END;
      FOR  $j := 1$  TO  $\min(\gamma(\eta), l)$  DO parfdm( $r_\lambda Lp_\lambda, v_{l-1, \lambda}, d_{l-1, \lambda}, l - 1, n', m', \eta$ ); END
    END;
    IF  $n > m$  THEN xmerge ELSE ymerge END;
    transportandadd( $p_\kappa v_{l-1, \kappa} \rightarrow v, p_\lambda v_{l-1, \lambda} \rightarrow v$ );
  END;

BEGIN
  IF  $l = 0$  THEN  $u := L^{-1} f$ 
  ELSE
     $u := S_l^{\nu 1}(L, u, f)$ ;
     $d := f - Lu; v := 0$ ;
    IF  $\iota = 00$  THEN (* Typ I Verzweigung *)
      IF  $(l > dl)$  OR  $(n + m = 0)$  THEN (* kein Halbieren *)
        FOR  $\kappa := 00$  TO 11 DO
          IF  $\gamma(\kappa) > 0$  THEN
             $v_{l-1} := 0$ 
            FOR  $j := 1$  TO  $\min(\gamma(\kappa), l)$  DO parfdm( $r_\kappa Lp_\kappa, v_{l-1}, r_\kappa d, l - 1, n, m, \kappa$ ); END;
             $v := v + p_\kappa v_{l-1}$ ;
          END
        END
      ELSE (* halbieren *)
        computeinparallel(00,10);
        computeinparallel(01,11);
      END
    ELSE (* Typ II Verzweigung *)
      IF  $(l > dl)$  OR  $(n + m = 0)$  THEN (* kein Halbieren *)
        FOR  $\kappa \in \{00, \iota\}$  DO
          IF  $\gamma(\kappa) > 0$  THEN
             $v_{l-1} := 0$ 
            FOR  $j := 1$  TO  $\min(\gamma(\iota), l)$  DO parfdm( $r_\kappa Lp_\kappa, v_{l-1}, r_\kappa d, l - 1, n, m, \iota$ ); END;
             $v := v + p_\kappa v_{l-1}$ ;
          END
        END
      ELSE (* halbieren *)
        computeinparallel(00,  $\iota$ );
      END;
    END;
     $u := u + v$ ;
     $u := S_l^{\nu 2}(L, u, f)$ ;
  END;
END;

```

5.3 DIRMU Implementierung

Die Parallelisierung der Frequenzzerlegungsmethode wurde bis jetzt ohne Berücksichtigung eines speziellen Rechners formuliert (bis auf die Verbindungsstruktur). Im Rahmen

Abbildung 5.6: Abspeicherung eines Gitters im Privat- und Multiportspeicher

der Arbeit wurde eine Implementierung von Algorithmus 5.1 auf dem DIRMU Multiprozessor des IMMD III erstellt.

DIRMU (siehe [11]) ist ein Multiprozessor bestehend aus 25 eigenständigen Mikrorechnern mit 8086/87 Prozessor. Die Kommunikation geschieht über *verteilten, gemeinsamen Speicher*, d. h. jedem Prozessor ist ein sog. Multiportspeicher zugeordnet, auf den sowohl von ihm selbst, als auch von seinen Nachbarn (je nach gewählter Verbindungsstruktur) zugegriffen werden kann. Der Zugriff auf den Speicher eines Nachbarn ist völlig transparent und erscheint wie ein normaler Hauptspeicherzugriff.

Mit Hilfe des parallelen Betriebssystems DIRMOS können Programme für unterschiedliche Verbindungsstrukturen entworfen werden. Da der Algorithmus die Hypercube Struktur verwendet, konnten nur maximal 16 Prozessoren eingesetzt werden.

Da der Multiportspeicher nur 64 Kbyte groß ist, wurden nur die nötigsten Daten dort gespeichert. Insbesondere wurden alle Gitterpunkte überlappend im Privatspeicher gehalten und nur Platz für die Ränder im Multiport reserviert (siehe Bild 5.6 hierzu). Vor einem Glättungsschritt müssen dann erst die überlappenden Bereiche über den Multiportspeicher ausgetauscht werden. Ein Vorteil dieser Vorgehensweise ist die einfache Formulierung der Operationen Glättung, Defektbildung, Restriktion und Prolongation, die sich vom Monoprogramm nur noch in anderen Schleifengrenzen unterscheiden, da die Datenbereiche zusammenhängend abgespeichert sind.

Das Programm besteht insgesamt aus den fünf Modulen MAIN, FDM, GM, VCUBE und MISC. Alle numerischen Prozeduren sind in FDM enthalten. GM enthält Prozeduren zur dynamischen Speicherverwaltung der Gitter und VCUBE realisiert die Verwaltung der Hypercube Struktur einschließlich der Einbettung der Tori, eines Baumes zur schnellen Synchronisation und der Prozeduren `xsplit` und `ysplit` zur (rekursiven) Halbierung der Konfigurationen.

5.4 Speedups

Neben dem bereits definierten Speedup $S(p)$ wird in den folgenden Tabellen auch die Effizienz $E(p)$ angegeben, die definiert ist als

$$E(p) = \frac{S(p)}{p} \quad (5.4)$$

$E(p)$ ist eine Zahl zwischen 0 und 1.

Es wurden die Speedups für 2,4,8 und 16 Rechner bei verschiedenen Gittergrößen für die Zyklenformen $\langle 1, 1, 1, 1 \rangle$, $\langle 1, 2, 2, 2 \rangle$ und $\langle 2, 2, 2, 2 \rangle$ gemessen. Die Monorechenzeiten konnten nur bis $h = \frac{1}{32}$ auf einem einzelnen DIRMU Modul bestimmt werden. Die Zeiten für $h = \frac{1}{64}$ und $h = \frac{1}{128}$ wurden durch Hochrechnen der entsprechenden Re-

α	β	GS - rb			FZM			ILU		
		ρ	$T_{np}(8)$	$T_{np}(16)$	ρ	$T_{np}(8)$	$T_{np}(16)$	ρ	$T_{np}(8)$	$T_{np}(16)$
1	1	0.108	0.9	0.5	0.086	5.9	3.4	0.121	0.86	0.73
$\frac{1}{2}$	2	0.393	2.23	1.1	0.196	8.8	5.2	0.150	0.96	0.81
$\frac{1}{10}$	10	0.938	32.5	16.2	0.295	11.8	6.9	0.135	0.91	0.77
10^{-2}	10^2	0.977	89.4	44.7	0.050	4.8	2.8	$8 \cdot 10^{-4}$	0.26	0.22
10^{-5}	10^5	0.977	89.4	44.7	0.051	4.8	2.8	$4 \cdot 10^{-15}$	0.05	0.05

Tabelle 5.2: Normierte, parallele Rechenzeiten für verschiedene Verfahren angewandt auf die anisotrope Gleichung, $h=1/64$, $p = 8,16$

chenzeiten einer SUN Workstation mit einem bei kleineren Gittern ermittelten Faktor, bestimmt. In der Parallelversion wurden jeweils alle Möglichkeiten für dl ausprobiert und der schnellste Lauf zur Berechnung des Speedups herangezogen. Dabei waren erwartungsgemäß bei $\gamma = \langle 1, 1, 1, 1 \rangle$ und $\gamma = \langle 2, 2, 2, 2 \rangle$ möglichst niedrige Werte für dl optimal (meißt $n + m$ oder $n + m + 1$), da die Arbeit in den Teilbäumen sehr ungleich ist. Beim $\langle 1, 2, 2, 2 \rangle$ -Zyklus waren immer möglichst große Werte von dl optimal.

Die Tabellen zeigen, daß die parallele Frequenzzerlegungsmethode auch schon bei sehr wenigen Unbekannten pro Prozessor sehr effizient arbeitet. Die entstehenden Verluste entstehen nicht durch ungleichmäßige Lastverteilung der Grobgittergleichungen, wie durch eine Modellierung des Speedups herausgefunden wurde, sondern müssen durch den relativ hohen Kommunikations- und Synchronisationsaufwand bei groben Gittern entstehen.

Tabelle 5.2 zeigt die normierten, parallelen Rechenzeiten für die Frequenzzerlegungsmethode und das Standardmehrgitterverfahren mit red-black Gauß-Seidel Glätter bzw. ILU Glätter. Das Standardmehrgitterverfahren ist der Frequenzzerlegungsmethode im anisotropen Fall sowohl in der Konvergenzrate als auch im Speedup unterlegen. Für die Poisson Gleichung ($\alpha = \beta = 1$) ist das Standard Verfahren wegen dem viel geringeren Aufwand klar besser. Gegen das ILU Verfahren hat die parallele Frequenzzerlegungsmethode bei der 2D anisotropen Gleichung in keinem Fall eine Chance, trotz besserem Speedup. Die Daten für das ILU Verfahren wurden von G. Horton freundlicherweise zur Verfügung gestellt.

ml	2 Rechner		4 Rechner		8 Rechner		16 Rechner	
	$S(2)$	$E(2)$	$S(4)$	$E(4)$	$S(8)$	$E(8)$	$S(16)$	$E(16)$
1	1.58	0.79						
2	1.79	0.90	2.93	0.73				
3	1.82	0.91	3.26	0.81	5.43	0.68		
4	1.90	0.95	3.58	0.89	6.28	0.79	11.00	0.69
5			3.59	0.90	6.80	0.85	12.48	0.78
6							13.95	0.87

Tabelle 5.3: Speedup für den $\gamma = \langle 1, 1, 1, 1 \rangle$ Zyklus

ml	2 Rechner		4 Rechner		8 Rechner		16 Rechner	
	$S(2)$	$E(2)$	$S(4)$	$E(4)$	$S(8)$	$E(8)$	$S(16)$	$E(16)$
1	1.60	0.80						
2	1.83	0.92	2.92	0.73				
3	1.93	0.97	3.59	0.90	5.82	0.73		
4	1.98	0.99	3.80	0.95	7.04	0.88	11.89	0.74
5			3.92	0.98	7.63	0.95	14.39	0.90
6							15.47	0.97

Tabelle 5.4: Speedup für den $\gamma = \langle 1, 2, 2, 2 \rangle$ Zyklus

æ

ml	2 Rechner		4 Rechner		8 Rechner		16 Rechner	
	$S(2)$	$E(2)$	$S(4)$	$E(4)$	$S(8)$	$E(8)$	$S(16)$	$E(16)$
1	1.58	0.79						
2	1.81	0.90	2.91	0.73				
3	1.86	0.93	3.21	0.80	5.24	0.66		
4	1.88	0.94	3.50	0.88	5.78	0.72	9.47	0.59
5			3.56	0.89	6.49	0.81	11.04	0.69
6							12.29	0.77

Tabelle 5.5: Speedup für den $\gamma = \langle 2, 2, 2, 2 \rangle$ Zyklus

Kapitel 6

Zusammenfassung und Ausblick

Es konnte nachgewiesen werden, daß sich die Frequenzzzerlegungsmethode , auch bei Beschränkung auf die notwendigen Grobitterkorrekturen, sehr gut parallelisieren läßt.

Alle weiteren Arbeiten müssen sich vielmehr mit den numerischen Aspekten des Verfahrens befassen. In Bezug auf die anisotrope Gleichung kann gesagt werden, daß die Frequenzzzerlegungsmethode zwar dem Standardverfahren mit Jacobi oder red-black Gauß-Seidel Glätter klar überlegen ist, aber in 2D mit dem ILU Verfahren und seinen Varianten nicht konkurrieren kann. Allerdings ist das ILU Verfahren in 3D *nicht* mehr robust, da dort im anisotropen Fall i. A. keine Tridiagonalsysteme mehr entstehen. Die Frequenzzzerlegungsmethode ist für diesen Fall ein aussichtsreiches Verfahren, da zudem die Komplexitätsabschätzungen noch günstiger sind (siehe Abschnitt 3.3).

Ein weiteres, wichtiges Ergebnis dieser Arbeit ist, daß der volle W-Zyklus notwendig ist, um für alle Werte von α und β eine Konvergenzrate unabhängig von h zu erreichen. Dies gilt uneingeschränkt für periodische Randbedingungen und für Dirichlet Randbedingungen bei Ausnahme von $\frac{\alpha}{\beta} \approx 10^{-2}$ sowie $\frac{\alpha}{\beta} \approx 10^2$.

Von den in Abschnitt 4.4 besprochenen Problemen scheint die Erweiterung auf die Konvektions-Diffusionsgleichung am wichtigsten. Bevor dieses Problem nicht gelöst ist, ist keine Anwendung auf die Navier-Stokes Gleichungen möglich. Prinzipiell gibt es zwei Wege dieses Problem zu lösen:

1. Konstruktion von matrixabhängigen Prolongationen
2. Konstruktion einer Modifikation im Sinne von Abschnitt 4.5

Falls die Frequenzzzerlegungsmethode auf die Konvektions-Diffusionsgleichung erweitert werden kann, steht damit ein robustes und effizientes Verfahren für dreidimensionale Probleme auf Parallelrechnern zur Verfügung. æ

Literaturverzeichnis

- [1] ALCOUFFE, R. E., BRANDT, A., DENDY, J. E., PAINTER, J. W.: The Multi-Grid Method for the Diffusion Equation with Strongly Discontinuous Coefficients. *SIAM J. Sci. Stat. Comput.* 2 (1981) 430-454.
- [2] BASTIAN, P., FERZIGER, J. H., HORTON, G., VOLKERT, J.: Adaptive Multi-grid Solution of the Convection-Diffusion Equation on the DIRMU Multiprocessor. in *Robust Multi-Grid Methods, Proceedings of the Fourth GAMM Seminar, Notes on Numerical Fluid Mechanics, Volume 23*, Vieweg Verlag, Braunschweig, 1989.
- [3] BECKER, C., FERZIGER, J. H., PERIC, M., SCHEUERER, G.: Finite Volume Multi-Grid Solution of the Two-dimensional Incompressible Navier-Stokes Equations. in *Robust Multi-Grid Methods, Proceedings of the Fourth GAMM Seminar, Notes on Numerical Fluid Mechanics, Volume 23*, Vieweg Verlag, Braunschweig, 1989.
- [4] FREDERICKSON, P. O., MCBRYAN O. A.: *Parallel Superconvergent Multigrid*. Theory Center Technical Report 7/87, Cornell University, 1987.
- [5] GOLUB, G. H., VAN LOAN, C. F.: *Matrix Computations*. North Oxford Academic Publishing Company, Oxford, England, 1986.
- [6] HACKBUSCH, W.: *Multi-Grid Methods and Applications*. Springer, Berlin, Heidelberg 1985.
- [7] HACKBUSCH, W.: *Theorie und Numerik elliptischer Differentialgleichungen*. Teubner Verlag, Stuttgart, 1986.
- [8] HACKBUSCH, W.: The Frequency Decomposition Multi-Grid Method. in *Robust Multi-Grid Methods, Proceedings of the Fourth GAMM Seminar, Notes on Numerical Fluid Mechanics, Volume 23*, Vieweg Verlag, Braunschweig, 1989.
- [9] HACKBUSCH, W., HAGEMANN, S.: *Frequency Decomposition Multi-Grid Methods for Hyperbolic Problems*. Institut für Informatik und Praktische Mathematik, CAU, Kiel.

- [10] HACKBUSCH, W., TROTTENBERG, U., (editors): Multi-Grid Methods. Proceedings, Köln-Porz, Nov. 1981. Lecture Notes in Mathematics, 960, Springer, Berlin, 1982.
- [11] HÄNDLER, W.: Projekt DIRMU – Abschlußbericht. Ha 417/18-3 an die DFG, Juni 1983.
- [12] HORTON, G.: Parallelisierung eines Mehrgitterverfahrens mit ILU - Glättung. Diplomarbeit am IMMD III, Universität Erlangen-Nürnberg, 1989.
- [13] VARGA, R. S.: Matrix Iterative Analysis. Prentice Hall, 1962.
- [14] WESSELING, P.: Theoretical and Practical Aspects of a Multigrid Method. SIAM J. Sci. Stat. Comput. 3 (1982) 387-407.
- [15] WITTUM, G.: On the Robustness of ILU Smoothing. Preprint Nr. 451, February 1988, SFB 123, Universität Heidelberg. auch in Robust Multi-Grid Methods, Proceedings of the Fourth GAMM Seminar, Notes on Numerical Fluid Mechanics, Volume 23, Vieweg Verlag, Braunschweig, 1989.
- [16] DE ZEEUW, P. M.: Matrixdependent Prolongations and Restrictions in a Black Box Multigrid Solver. Centre for Mathematics and Computer Science, Report NM-R8801, February 1988.