8 Was man noch über MGV wissen sollte

Ziel dieses Abschnittes ist einen Überblick über bestimmte fortgeschrittene Aspekte von MGV zu geben, ohne auf Details einzugehen, so dass sich der Leser dieser Aspekte bewusst ist, und die entsprechenden Schlagworte sowie einige Referenzen kennt, um einen schnellen Einstieg in die spezielle Literatur zu finden.

8.1 MGV als Vorkonditionierer

Es ist möglich Mehrgitterverfahren zur Vorkonditionierung eines anderen Iterativen Verfahrens zu verwendet, z.B. das vorkonditionierte konjugierte Gradienten Verfahren (PCG). Dabei wird im allgemeinen die Anwendung des Vorkonditionierers C^{-1} auf einen Vektor v benötigt, welche durch Anwendung eines Mehrgitter-Zykluses auf den rechte Seite Vektor v realisiert wird.

Diese Kombination von Verfahren verursacht i.A. höhere Kosten pro Iterationsschritt, da zusätzlich zu den Kosten des MG-Zyklus noch mindesten ein Matrix-Vektor Produkt benötigt wird. Für das Poisson Problem ergibt die Vorkonditionierung mit einem einfachen V-Zyklus bereits einen optimalen Vorkonditionierer, d.h. Konditionszahl $\mathcal{O}(1)$. Wegen der höheren Kosten gegenüber einem reinen MGV ist jedoch in diesem Fall meist keine Einsparung an Rechenzeit zu verzeichnen.

Die Bedeutung des kombinierten Verfahrens besteht vielmehr darin, dass es wesentlich robuster ist [TOS01, Section 7.8]. Bei schwierigeren PDE-Problemen, wie z.B. Konvektion-Diffusion, kommt es häufig vor, dass ein einfaches MGV (z.B. mit wiederholten W-Zyklus) divergiert, jedoch ein MGV vorkonditioniertes Krylov-Unterraumverfahren (z.B. GMRES) konvergiert, und das sogar h-unabhängig.

Vorkonditionierer für CG Verfahren müssen symmetrisch positiv definit sein. Der verbreitetste Zyklus zum Vorkonditionieren von Poisson Problemen besteht darin auf jedem Level

1. ein Schritt Gauß-Seidel vorwärts,
2. Grobgitter-Korrektur,
3. ein Schritt Gauß-Seidel rückwärts,

durchzuführen. Die Symmetrie wird dabei durch die alternierenden Richtungen im Glätter sichergestellt, vgl. Kommentar in Abschnitt 3.4.

8.2 Verwandte Verfahren

8.2.1 Algebraisches Multigrid

Die hier in der Vorlesung vorgestellten Verfahren sind alle davon ausgegangen, dass ein gegebenes grobes Gitter verfeinert wurde um das feine Gitter zu erhalten (Hierarchische Gitter) und die PDE darauf zu diskretisieren. Die Grobgitter-Operatoren wurden stets basierend auf den Geometrieinformationen

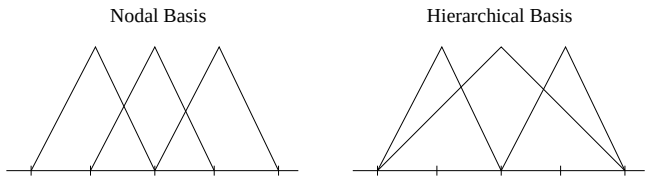

Abbildung 1: Hierarchische Basis

der groben Gitter erstellt. Diese Vorgehensweise wird auch *Geometrisches Multigrid* (*GMG*) genannt.

Im Gegensatz dazu gibt es auch Verfahren welche als Input nur die schwachbesetzte Systemmatrix und rechte Seite zu einem beliebigen (feinen) Netz verwenden. Grobgitter-Operatoren werden hierbei durch Vergröberungen basierend auf der Besetzungsstruktur der Matrix konstruiert, indem z.B. Freiheitsgrade ausgewählt werden, welche der jeweils gröberen Repräsentation des Gleichungssystems angehören sollen, und welche nicht. Die Galerkin Eigenschaft des Grobgitterproblems

$$A^{2h} = (I_{2h}^h)^T A^h I_{2h}^h$$

(vgl. Abschnitt 4.2.) gibt ein Beispiel dafür, dass man Grobgitteroperatoren auch ohne Verwendung eines Grobgitters erzeugen kann. Das vergröberte Problem muss dabei nicht notwendigerweise als Diskretisierung der PDE interpretierbar sein. Es ist vielmehr eine gröbere Darstellung des Gleichungssystems.

Da hierbei nur die algebraische Struktur des Gleichungsystems verwendet wird, spricht man von *Algebraischem Multigrid* (*AMG*).

Die geringen Informationsanforderungen von AMG bedeuten das man diese Verfahren in einer Softwarebibliothek implementieren kann, welche leicht in andere Programme eingebunden werden kann (Modularität). Die Annahme des Vorhandenseins eines hierarchischen Gitters ist in industriellen Anwendungen oft problematisch, da vorhandene Netzgeneratoren eingesetzt werden, die nur ein Gitter vorgegebener Feinheit als Output erzeugen. In diesem Sinne ist also GMG oft gar nicht möglich.

Ein weiterer Vorteil ist, das man die Ansätze von AMG auch recht einfach auf andere Klassen von Gleichungssystemen anwenden kann, z.B. auch wenn diese nicht aus der Diskretisierung einer PDE entstehen. Ob dies jedoch funktioniert ist offenbar stark von dem Problem abhängig.

8.2.2 Mutlilevel Methoden

Im Fall der FEM hat die Wahl der Basisfunktionen einen entscheidenden Einfluss auf die Kondition der entstehenden Matrizen. Verwendet man anstelle der üblichen Knoten-Basis die so genannte hierarchische Basis, siehe Abbildung 1, so hat die FE Steifigkeitsmatrix Konditionszahl $\mathcal{O}(|\ln(h)|^2)$ für das 2d Poisson Problem (statt $\mathcal{O}(h^{-2})$ in der Knotenbasis). Jedoch ist diese Basis unpraktisch, da die entstehenden Matrizen nicht schwachbesetzt sind.

Ein Ausweg ist die Matrizen in der Knotenbasis zu speichern, jedoch in der Hierarchischen Basis zu rechnen. Das Umrechnen zwischen den Basen erfolgt einfach durch Interpolation. Dieses erzeugt den so genannten *Yserentant* oder *Hierarchische Basis Vorkonditionierer* [Yse86], bei dem die Anwendung von C^{-1} daraus besteht die Interpolationshierarchie einmal von fein nach grob und einmal von grob nach fein zu durchlaufen. Kosten hierfür sind $\mathcal{O}(N)$.

Leider ist dieser Vorkonditionierer nur in 2d fast optimal, da sich in 3d die Konditionzahl $\mathcal{O}(h^{-1})$ ergibt. Eine Verbesserung ist der Bramble-Pasciak-Xu (*BPX*) *Vorkonditionierer* [BPX90], welcher im Wesentlichen die selben Rechenoperationen erfordert, zusätzlich jedoch pro Level der Hierarchie Speicherplatz für einen Vektor benötigt. Mit dem BPX Vorkonditionierer wird eine Konditionszahl $\mathcal{O}(1)$ erreicht, er ist also optimal, und dies sowohl in 2d als auch in 3d.

Die Anzahl Iterationen um eine vorgegebene Reduzierung des Residuums zu erreichen ist bei den beiden angesprochenen Vorkonditionierern für CG i.A. höher als bei MGV, jedoch die Kosten pro Iteration sind wesentlich geringer, so dass die Verfahren bzgl. gesamt Aufwand in etwa gleichwertig sind. Ein wesentlicher Vorteil besteht jedoch darin das keine Grobgitter-Steifigkeitsmatrizen benötigt werden.

8.2.3 $\mathcal{H}$-Matrix Methoden

Hierarchische Matrizen oder kurz *$\mathcal{H}$-Matrizen* basieren auf einer speziellen Darstellung und Speicherung von Matrizen, und zugehöriger Umsetzung arithmetischer Operationen. Eine Besonderheit dabei ist das auch voll besetzte Matrizen in einer data-sparse Darstellung mit beliebiger, fester Genauigkeit gespeichert werden können, wobei der Speicherbedarf von der Genauigkeit abhängt. Insbesondere kann eine Näherung der Inversen einer PDE-Diskretisierung (welche voll besetzt ist), auf diese Weise gespeichert und für Arithmetik benutzt werden. Die Arithmetischen Operationen die durchgeführt werden können sind u.A. Matrix-Vektor Produkte, Addition, Multiplikation, Inversion, LU-Zerlegung und Cholesky Zerlegung, alle mit fast optimalem Rechen- und Speicheraufwandt.

Ursprünglich wurden diese Techniken für Randelement Diskretisierungen (Boundary Element Method, BEM) entwickelt. Diese Diskretisierungen benötigen nur eine Vernetzung des Randes des Gebiets einer PDE (daher wesentlich weniger Freiheitsgrade) jedoch sind die entstehenden Matrizen voll besetzt. Dieses Problem kann mit $\mathcal{H}$-Matrizen überkommen werden. Später hat man festgestellt, das der Lösungsoperator des Poisson Problems, wenn er mit der Greenschen Funktion dargestellt wird, ebenfalls in diese Darstellung passt, und somit die Technik auf die Inverse der Steifigkeitsmatrix anwenden kann. Wir verweisen auf [BGH03] für eine detaillierte Einführung in dieses aktuelle Forschungsgebiet.

8.3 MGV für andere PDEs

Ein Vorteil der MGV (insbesondere bei GMG) ist, das die zugrundeliegende Idee einfach ist und auf viele PDE Probleme angewendet werden kann. Auch

wenn häufig eine rigorose Konvergenzanalysis nicht verfügbar ist, bieten MGV bewährte und robuste Verfahren zum schnellen Lösen der Gleichungssysteme die bei der Diskretisierung von PDEs entstehen. Einer der kritischsten Punkte bei der Konstruktion eines MGV ist der Glätter.

Konvention: Im folgenden werden bei der Diskussion von PDEs die Randbedingungen weggelassen, da es nur um die Differentialoperatoren der PDEs geht. Es sei also angenommen die Probleme werden durch geeignete Randbedingungen vervollständigt.

8.3.1 Allgemeinere lineare elliptische Probleme 2. Ordnung

Eine allgemeine skalare elliptische PDE zweiter Ordnung kann man als

$$-\nabla \cdot (A\nabla u) + b^T \cdot \nabla u + cu = f \tag{8.1}$$

schreiben. Dabei ist A (u.U. ortsabhängige) symmetrisch positiv definite Matrix [1], b (ortsabhängiger) Vektor und c eine skalare Funktion. Die Terme stehen im einzelnen für folgende physikalischen Prozesse:

A**:** Diffusion (Ausgleichsvorgänge),

b**:** Konvektion (Transport, z.B. durch Strömung),

c**:** Reaktion (z.B. chemische Reaktionen).

Meist wird die Einschränkung $c \geq 0$ getroffen um die Analysis zu vereinfachen, und da damit viele bedeutende Fälle abgedeckt werden. Falls $c < 0$ vorkommt, so ist im allgemeinen damit zu Rechnen das die Lösung u in diesem Bereich oszilliert mit einer von c abhängigen Wellenlänge. Die standard Analysis zu Existenz und Eindeutigkeit deckt diesen Fall nicht ab. Weiterhin ist es erforderlich das die Diskretisierung fein genug ist um die Oszillationen auflösen zu können, was sinnvolle Grobgitterkorrektur u.U. unmöglich macht.

Im folgenden betrachten wir kurz zwei bedeutende Spezialfälle, die auch häufig auftretende Probleme aufzeigen.

Reaktion-Diffusion Hierbei wird die Konvektion $b = 0$ gesetzt. Der Einfachheit halber betrachten wir konstante, isotrope Diffusion $A = \varepsilon I$ mit $\varepsilon > 0$ und konstante Reaktionskoeffizienten $c > 0$,

$$-\varepsilon\Delta u + cu = f. \tag{8.2}$$

Probleme dieses Typs lassen sich i.A. gut mit Multigrid lösen, und die Analysis dieser Vorlesung lässt sich auf diesen Fall verallgemeneinern. Auch BPX und Yserentant Vorkonditionierer decken diesen Fall gut ab.

Probleme treten jedoch auf wenn $0 < \varepsilon \ll c$, im so genannten *singulär gestörten Fall*. In diesem Fall ist es wichtig eine in ε uniforme Konvergenz der

[1] Falls A ortsabhängig ist fordern wir gleichmäßige Elliptizität: $\exists\gamma > 0 : \; y^T A(x) y \geq \gamma ||y||^2 \; \forall x \in \Omega, \, y \in \mathbb{R}^d$.

diskreten Lösung u_h zur Lösung u zu erreichen, d.h. die Konstanten in der Konvergenzanalysis müssen unabhängig von ε sein. Eine weit verbreitete Strategie um dies zu erreichen sind anisotrop verfeinerte Gitter, sieh z.B. [AL98]. Diese führen jedoch i.A. zu einer schlechten Konditionszahl für die Steifigkeitsmatrix und verschlechtern die Konvergenzgeschwindigkeit der MGV. Als Ausweg werden oft angepasste Glätter (Line-Smoother) und Vergröberunsstrategien (Semicoarsening) verwendet, siehe z.B. [AS02].

Konvektion-Diffusion Hierbei wird die Reaktion $c = 0$ gesetzt. Der Einfachheit halber betrachten wir wieder konstante, isotrope Diffusion $A = \varepsilon I$ mit $\varepsilon > 0$ und ein konstantes Konvektionsfeld b,

$$-\varepsilon \Delta u + b^T \cdot \nabla u = f. \tag{8.3}$$

Im Vergleich zu (8.2) präsentiert dies eine gravierende neue Schwierigkeit: die entstehenden Steifigkeitsmatrizen sind nichtsymmetrisch. Dies bedeutet u.A. das CG als Löser nicht mehr in Frage kommt, und ggf. teurere Verfahren wie GMRES oder BiCG-Stab verwendet werden müssen.

Hinzu kommt das die diskreten Lösungen bei linearen bzw. bilineare Finiten Elementen nicht-physikalische Oszillationen aufweisen können falls die so genannte Gitter-Peclet Zahl (e.g. [Ram99]),

$$Pe := \frac{h\ ||b||_\infty}{2\varepsilon},$$

$Pe > 1$ wird. Um dies zu verhindern muss h klein genug gewählt werden, und falls dies nicht möglich ist (z.B. $0 < \varepsilon \ll ||b||$) sollten der Diskretisierung zusätzliche stabilisierende Terme hinzugefügt werden um die Oszillationen gering zu halten bzw. zu vermeiden, siehe z.B. [Joh87, Section 9.5] oder [AL96, Kno06]. Der singulär gestörte Fall $0 < \varepsilon \ll ||b||$ spielt auch hier eine besondere Rolle. Analog zu (8.2) sollten ε-uniforme Diskretisierungsverfahren verwendet werden, welche oft angepasste anisotrope Gitter verwenden, z.B. [AL96].

Im Diffusions-dominierten Fall ($\varepsilon \geq ||b||$)) können die gleichen MGV zum Lösen verwendet werden wie für die Poisson-Gleichung, womit man i.A. vergleichbare Konvergenzraten erhält.

Jedoch im Konvektions-dominierten Fall $\varepsilon < ||b||$ treten besondere Effekte auf. Im reinen Konvektions-Fall $\varepsilon = 0$ wird Information nur in einer Richtung, entlang der Charakteristiken, transportiert, so dass (8.3) zu einer ODE entlang der Charakteristik degeneriert. In diesem Fall ist Gauß-Seidel Glättung nur sinnvoll wenn diese in der durch b vorgegebenen Richtung erfolgt. Also erst alle Knoten am Einströmrand, dann alle Knoten stromabwärts von diesen, usw. bis zum Ausströmrand. Auch im Fall $\varepsilon > 0$ ist dieser Effekt noch zu beobachten, und natürlich besonders stark ausgeprägt im singulär gestörten Fall $\varepsilon \ll ||b||$. Wenn die Stromrichtung beachtet wird ist das Gauß-Seidel Verfahren ein hervorragender Glätter, und es sind sogar nichttriviale Fälle bekannt wo ein Schritt des Verfahrens das Problem exakt löst [Joh87]. Leider ist es schwierig dieses Verhalten bei nicht-konstanten Konvektionsfeldern auszunutzen, da

es unter anderem zu Rezirkulationen kommen kann, so das eine entsprechende Knotennummerierung nicht erfolgen kann.

Weitere Schwierigkeiten MGV für solche Probleme zu verwenden begründen sich in der schlechten Approximation der Lösung auf groben Netzen. Diese Resultiert aus der höheren Gitter Peclet Zahl und der erforderlichen Stabilisierung. Die Qualität der Diskretisierung auf groben Netzen und die Geschwindigkeit von MGV sind also bei diesem Problemen eng gekoppelt. Die meisten Verfahren versuchen die schlechte Grobgitter-Korrektur durch besonders gute Glätter zu kompensieren. Dies ist jedoch nur in bestimmten Fällen möglich. [Joh06] gibt einen aktuellen Einblick in die Thematik und zeigt insbesondere auch diese Probleme auf.

Es sei bemerkt, dass auch wenn MGV in diesem Fall keinen optimalen Löser darstellen, so stellen sie doch einen der besten, wenn nicht sogar den besten, bekannten Lösungsansatz für großdimensionierte Probleme dieser Art dar.

Gekoppelte PDE-Systeme, Bsp. Oseen Gleichungen Wenn wir den Bereich der skalaren PDEs verlassen und PDE-Systeme mit vektorwertigen Lösungen betrachten, ergeben sich unter Umständen weitere Schwierigkeiten. Als Beispiel betrachten wir die Oseen-Gleichungen welche bei der Linearisierung der inkompressiblen, stationären Navier-Stokes Gleichungen auftreten,

$$-\Delta \mathbf{u} + \mathbf{b}^T \cdot \nabla \mathbf{u} + \nabla p = f \tag{8.4a}$$

$$\nabla \cdot \mathbf{u} = 0. \tag{8.4b}$$

Diskretisierung dieser Gleichungen erfordert gemischte Finite Elemente (unterschiedliche Ansatzräume für $\mathbf{u}$ und p) welche eine Kompatibilitätsbedingung erfüllen müssen (LBB-Bedingung, aka. inf-sup Bedingung), siehe z.B. [Gun89]. Diese Diskretisierungen führen dann auf Gleichungsprobleme mit Sattelpunktsstruktur,

$$\begin{bmatrix} F & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}} \\ \underline{p} \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}. \tag{8.5}$$

Hierbei stellt insbesondere der Null-Block in der rechten unteren Ecke der Matrix ein Problem dar, so dass z.B. Gauß-Seidel oder Jacobi Verfahren nicht als Glätter verwendet werden können.

Einen Möglichen Glätter erhält man wenn man ähnlich zum Gauß-Seidel Verfahren alle Gitter-Zellen durchläuft und lokale Gleichungsysteme vom Typ (8.5) mit den zur Gitter-Zelle gehörenden $(\mathbf{u}, p)$ Freiheitsgraden löst. Dieser Glätter wird Vanka-Smoother [Van86] oder Box-Smoother [TOS01, pp. 320–322] genannt. Auch wenn die lokalen Gleichungssysteme relative klein sind ist der Rechenaufwand für diese Art von Glätter deutlich höher als für Gauß-Seidel im skalaren Fall. Hinzu kommt das der Erfolg des erzeugten MGV stark von der Diskretisierung der PDE abhängt [Sch06].

Weitere Möglichkeiten zum effizienten Lösen dieser Systeme bestehen in Schur-Komplement Vorkonditionierern für GMRES [ESW05], welche auch als Glätter für MGV eingesetzt werden können [Tur99].

Es sei bemerkt das dies nur ein Beispiel für Systeme von PDEs darstellt, welches zudem besondere Schwierigkeiten mitbringt. Ein anderes bedeutendes Beispiel sind die Lamé Gleichungen für lineare Elastizität, für die das Gauß-Seidel Verfahren bereits einen geeigneten Glätter darstellt.

Nichtlineare PDEs Nichtlineare PDEs führen beim Diskretisieren i.A. auf nichtlineare algebraische Gleichungen. Zum Lösen dieser können das Newton Verfahren oder geeignete Fixpunktiterationen verwendet werden. Dabei muss in der Regel in jedem Schritt ein lineares Gleichungssystem gelöst werden mit der Jacobi Matrix der nichtlinearen Gleichungen als Systemmatrix.

Zum Lösen der nichtlinearen Algebraischen Gleichungen benötigt man den Jacobian dieser. Meist ist es jedoch so, das man die gleiche Matrix erhält wenn man erst die PDE linearisiert, und dann diese diskretisiert. Also kann man die diskretisierte, linearisierte PDE wieder mit MGV lösen. Wird dies wieder im Rahmen eines Full Multigrid Verfahrens realisiert, so dient die interpolierte Grobgitterlösung jeweils wieder als gute Startnäherung für das Newton Verfahren auf dem aktuellen Gitter, so dass häufig bereits ein einziger Newton Schritt ausreicht um den Fehler auf die Größenordnung des Diskretisierungsfehlers zu reduzieren. Lediglich auf den gröbsten Netzen der Hierarchie müssen mehrere Newton Schritte durchgeführt werden, bzw. eine Fixpunkt Iteration durchgeführt werden um in den Einzugsbereich des Newton Verfahrens zu gelangen.

Neben dieser Möglichkeit MGV nur für die linearisierten Gleichungen anzuwenden, gibt es auch MGV für nichtlineare PDEs, z.B. das *Full Approximation Scheme* (*FAS*) (z.B. [BHM00, TOS01]). Dabei wird eine Linearisierung nur im Glätter vorgenommen, z.B. indem ein nichtlineares Gauß-Seidel Verfahren benutzt wird, wobei für jede Zeile das Residuum der nichtlinearen Gleichung bestimmt wird, durch das Diagonalelement der lokalen Jacobi Matrix geteilt und zum Update verwendet wird. Es wird also auf allen Ebenen der Hierarchie mit nichtlinearen Problemen gearbeitet.

Beide Zugänge erlauben es viele nichtlineare PDE Diskretisierungen mit optimalem Aufwand zu lösen. Insbesondere auch der Mehraufwand durch die Nichtlinearität wird durch die MGV weitestgehend eliminiert.

Zeitabhängige PDEs Zur Wiederholung: Bei Zeitabhängigen PDEs werden in der Regel die Diskretisierungen bzgl. Ort und Zeit separat durchgeführt. Zum Beispiel kann man die PDE

$$\frac{\partial u}{\partial t} = \Delta u$$

zuerst in der Zeit diskretisieren, mit dem Vorwärts-Euler Verfahren

$$\frac{u^{(k+1)} - u^{(k)}}{\tau} = \Delta u^{(k)},$$

oder dem Rückwärts-Euler Verfahren

$$\frac{u^{(k+1)} - u^{(k)}}{\tau} = \Delta u^{(k+1)},$$

wobei $\tau > 0$ jeweils die Schrittweite der Zeitdiskretisierung ist. Anschließend wird die Ortsdiskretisierung z.B. mit FEM oder FDM durchgeführt. Die Stabilitätsbedingungen für beide Ansätze sind aus der Numerik Partieller Differentialgleichungen bekannt.

Das Vorwärts-Euler Verfahren ist ein direktes Verfahren, d.h. es müssen keine Gleichungsysteme gelöst werden, also ist die Anwendung von MGV nicht notwendig.

Das Rückwärts-Euler Verfahren führt auf eine PDE

$$-\tau\Delta u^{(k+1)} + u^{(k+1)} = u^{(k)}$$

in $u^{(k+1)}$, d.h. der Zeit Term erzeugt einen zusätzlichen Term nullter Ordnung, einen Reaktions Term. Diese PDEs können natürlich mit MGV behandelt werden, siehe oben. Zu bemerken ist noch das für $\tau \to +0$ eine Singulär gestörte PDE erzeugt wird.

Lokale Mode Analysis Wir haben gesehen das die Konvergenzanalyse für Mehrgitterverfahren ein aufwendiger Prozess ist, und das selbst die beste Abschätzung in dieser Vorlesung noch sehr pessimistisch ist, wenn man die Abschätzungen mit numerischen Experimenten vergleicht. Es gibt jedoch Heuristiken mit denen man realistischere Schätzungen der Konvergenzrate erhalten kann. Sehr verbreitet ist die lokale Mode Analysis (*local mode analysis*, aka. normal mode analysis, aka. Fourier analysis). Diese setzt im Wesentlichen auf die Idee von (5.2) auf, wobei der Glättungsfaktor geschätzt wird indem rein lokale Betrachtungen angestellt werden (ein Knoten im Netz + umliegende). Mit diesem Ansatz wurden für viele Probleme realistische Schätzungen für die Konvergenzrate erhalten. Eine Einführung kann man z.B. in [BHM00, TOS01] finden.

Literatur

[AL96] T. Apel and G. Lube. Anisotropic mesh refinement in stabelized Galerkin methods. *Numerische Mathematik*, 74:261–282, 1996.

[AL98] T. Apel and G. Lube. Anisotropic mesh refinement for a singularly perturbed reaction diffusion model problem. *Applied Numerical Mathematics*, 26:415–433, 1998.

[AO00] M. Ainsworth and J.T. Oden. *A Posteriori Error Estimation in Finite Element Analysis*. Wiley, 2000.

[AS02] T. Apel and J. Schöberl. Multigrid methods for anisotropic edge refinement. *SIAM Journal on Numerical Analysis*, 40(5):1993–2006, 2002.

[Bä91] E. Bänsch. Local mesh refinement in 2 and 3 dimensions. *Impact of Computing in Science and Engineering*, 3:181–191, 1991.

[BGH03] S. Börm, L. Grasedyck, and W. Hackbusch. Introduction to hierarchical matrices with applications. *Engineering Analysis with Boundary Elements*, 27(5):405–422, 2003.

[BHM00] W.L. Briggs, V.E. Henson, and S.F. McCormick. *A Multigrid Tutorial*. SIAM, 2000.

[BPX90] J.H. Bramble, J.E. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Mathematics of Computation*, 55(191):1–22, July 1990.

[BR03] W. Bangerth and R. Rannacher. *Adaptive Finite Element Methods for Differential Equations*. Birkhäuser Verlag, 2003.

[Bra03] D. Braess. *Finite Elemente – Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*. Springer Lehrbuch, Berlin, 3rd edition, 2003.

[ESW05] H. Elman, D. Silvester, and A. Wathen. *Finite Elements and Fast Iterative Solvers*. Oxford University Press, 2005.

[Gun89] M.D. Gunzburger. *Finite Element Methods for Viscous Incompressible Flows*. Academic Press, 1989.

[Hac93] W. Hackbusch. *Iterative Lösung großer schwachbesetzter Gleichungssysteme*. Teubner, 1993.

[Hac94] W. Hackbusch. *Iterative Solution of Large Sparse Systems of Equations*. Springer, 1994.

[HL04] G. Haase and U. Langer. Mutligrid - methoden. http://www.numa.uni-linz.ac.at/Teaching/Notes/, 2004. Lecture Notes.

[Joh87] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, 1987.

[Joh06] K. Johannsen. *Robuste Mehrgitterverfahren für die Konvektions-Diffusions-Gleichung mit wirbelbehafteter Konvektion*. Teubner, 2006.

[Kno06] P. Knobloch. Improvements of the muzikami-hughes method for convection-diffusion equations. *Computer methods in applied machanics and engineering*, 196:579–594, 2006.

[Mey02] A. Meyer. Projection techniques embedded in the PCGM for handling hanging nodes and boundary restrictions. In B.H.V. Topping and Z. Bittnar, editors, *Engineering Computational Technology*, pages 147–165. Saxe-Cobourg Publications, 2002.

[Ram99] A. Ramage. A multigrid preconditioner for stabilised discretisations of advection-diffusion problems. *Journal of Computational and Applied Mathematics*, 110:187–203, 1999.

[Sch06] R. Schneider. *Applications of the Discrete Adjoint Method in Computational Fluid Dynamics*. PhD thesis, University of Leeds, 2006. http://www.comp.leeds.ac.uk/cgi-bin/sis/ext/rs_pub.cgi?cmd=listtheses#2006.

[TOS01] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, 2001.

[Tur99] S. Turek. *Efficient Solvers for Incompressible Flow Problems, An Algorithmic and Computational Approach*. Lecture Notes in Computational Science and Engineering. Springer, 1999.

[Van86] S.P. Vanka. Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. *Journal of Computational Physics*, 65:138–158, 1986.

[Yse86] H. Yserentant. On the multi-level splitting of finite element spaces for indefinite elliptic boundary value problems. *SIAM Journal on Numerical Analysis*, 23:581–595, 1986.