

Ralf Kornhuber

Rainer Roitzsch

Adaptive Finite-Element-Methoden für
konvektionsdominierte Randwertprobleme
bei partiellen Differentialgleichungen

Vortrag im Rahmen des 4. TECFLAM-Seminars,
Stuttgart, 3. November 1988

1 Einleitung

Die Simulation von Verbrennungsprozessen führt im stationären Fall auf elliptische Systeme von partiellen Differentialgleichungen mit dominierenden konvektiven Termen. Die numerische Behandlung solcher Systeme mit herkömmlichen Methoden macht große Schwierigkeiten, einerseits was die Komplexität der Algorithmen, andererseits was die Verlässlichkeit der Ergebnisse betrifft. Vor diesem Hintergrund ist es zur Entwicklung leistungsfähiger Codes erforderlich, über das bloße Variieren bekannter Ansätze hinaus nach grundsätzlich neuen Lösungen zu suchen.

Erstes Ergebnis und Basis für weitere Entwicklungen ist dabei die Arbeit von Deuffhard, Leinen und Yserontant [3] über die Behandlung skalarer, rein elliptischer Probleme mit adaptiven Finite-Element-Methoden. Die Grundidee, eine gegebene Ausgangsnäherung schrittweise bis zur gewünschten Genauigkeit zu verbessern, findet mathematisch ihren Niederschlag in der Verwendung hierarchischer finiter Elemente, aus der sich dann in natürlicher Weise die Wahl des iterativen Gleichungslösers, Fehlerschätzers und der Verfeinerungsstrategie ergibt.

Die Erweiterung der in [3] dargestellten Vorgehensweise auf konvektiv dominierte Probleme erfordert gegenüber dem rein elliptischen Fall eine Neu- oder Weiterentwicklung all dieser einzelnen Bausteine. Dazu soll in der vorliegenden Arbeit ein Beitrag geleistet werden.

Wir beschäftigen uns dabei mit dem Modellproblem

$$\begin{aligned} -\varepsilon \Delta u + \beta \cdot \nabla u &= f, & \text{auf } \Omega \subset \mathbb{R}^2 \\ u|_{\Gamma_0} &= u_0, \quad \frac{\partial u}{\partial n} \Big|_{\Gamma_n} = 0, & \text{auf } \partial\Omega = \Gamma_0 \cup \Gamma_n \end{aligned} \quad (M)$$

mit $\beta = \beta(x, y) \in \mathbb{R}^2$, $(x, y) \in \bar{\Omega}$. Den Einflußrand bezeichnen wir mit

$$\Gamma_{\text{in}} = \{(x, y) \in \partial\Omega \mid n(x, y) \cdot \beta(x, y) < 0\},$$

wobei $n(x, y)$ die nach außen gerichtete Normale im Punkt $(x, y) \in \partial\Omega$ bedeutet.

Der folgende Abschnitt enthält eine kurze Beschreibung des sogenannten Stromlinien-Diffusions-Verfahrens, einer stabilen Modifikation des Standard-Galerkin-Verfahrens. Es zeigt sich, daß dieses Verfahren seinem Namen nur gerecht wird, wenn die Richtung der zugrunde liegenden Triangulierung der Flußrichtung β angepaßt ist. Im Abschnitt 3 stellen wir eine neue Verfeinerungsstrategie zur schnellen Auflösung von Grenzschichten vor, welche insbesondere bei Anwendung auf innere Grenzschichten ein vorheriges Ausrichten der Kanten nötig macht. Durch die vorangegangenen Überlegungen motiviert, wird im letzten Abschnitt erstmals ein Algorithmus zur adaptiven

Ausrichtung der Triangulierung vorgeschlagen, dessen Wirksamkeit anhand eines Beispiels demonstriert wird.

2 Das Stromlinien-Diffusions-Verfahren

Der Grad, in dem die Diskretisierung eine der Differentialgleichung zugrunde liegende physikalische Realität widerspiegelt, ist ein Maß für die Stabilität dieses Verfahrens [2]. Wir wollen dieses Grundprinzip zunächst an dem einfachen Beispiel

$$-\varepsilon u'' + u' = 0, \quad u(0) = 0, \quad u(1) = 1, \quad (1)$$

erläutern. Das Standard-Galerkin-Verfahren (siehe [7]) zu einem äquidistanten Gitter zur Schrittweite $h = 1/N$ führt auf das zentrale Differenzschema

$$\frac{\varepsilon}{h^2}(-U_{i-1} + 2U_i - U_{i+1}) + \frac{1}{2h}(U_{i+1} - U_{i-1}) = 0 \quad (2)$$

mit $i = 1, \dots, N-1$ und $U_0 = 0$, $U_N = 1$. Hier hängt U_i in gleicher Weise von U_{i+1} und U_{i-1} ab, was der physikalischen Realität, nämlich einer stärkeren Abhängigkeit vom stromaufwärts genommenen Wert U_{i-1} offenbar widerspricht. Entsprechend ist das Verfahren bei dominierender Konvektion, das heißt $\varepsilon < h/2$ instabil. Ersetzt man die zentrale Approximation der Konvektion durch eine Mittelung von vor- und rückwärts genommenen Differenzenquotienten, so erhält man das upwind-Schema:

$$\frac{\varepsilon}{h^2}(-U_{i-1} + 2U_i - U_{i+1}) + \frac{1+2\delta}{2h}(U_i - U_{i-1}) + \frac{1-2\delta}{2h}(U_{i+1} - U_i) = 0. \quad (3)$$

Die Wahl des upwind-Parameters $\delta \geq 0$ ermöglicht eine physikalisch sinnvolle Gewichtung von U_{i-1} und U_{i+1} : Das Verfahren (3) ist für $\delta \geq (h - 2\varepsilon)/2h$ stabil. Diese Gewichtung kann man als Addition von künstlicher Diffusion interpretieren, denn (3) läßt sich umschreiben in

$$\frac{\varepsilon + \delta h}{h^2}(-U_{i-1} + 2U_i - U_{i+1}) + \frac{1}{2h}(U_{i+1} - U_{i-1}) = 0. \quad (4)$$

Schließlich führt diese rein aus Stabilitätsgründen angebrachte künstliche Diffusion nicht zu einem Verlust an Approximationsgenauigkeit. Im Gegenteil kann man im vorliegenden Fall δ sogar so wählen, daß (4) die ersetzte Lösung in den Knotenpunkten reproduziert (siehe [4]).

Zur Diskretisierung des zweidimensionalen Modellproblems (M) aus dem ersten Abschnitt wählen wir zunächst eine uniforme Triangulierung \mathcal{T} von Ω mit der Gitterweite h und bezeichnen mit $V_{\mathcal{T}}$ den Raum aller stetigen Funktionen, die auf jedem Dreieck aus \mathcal{T} linear sind. Das Standard-Galerkin-Verfahren zu (M) lautet dann gemäß [7]:

Finde ein $u^T \in V_T$ mit $u^T|_{\Gamma_0} = u_0$, so daß gilt

$$\varepsilon(\nabla u^T \cdot \nabla v) + (u_\beta^T, v) = (f, v), \quad (5)$$

für alle $v \in V_T$ mit $v|_{\Gamma_0} = 0$,

wobei wir von den Abkürzungen

$$(v, w) = \int_{\Omega} v(x)w(x)dx, \quad w_\beta := \beta \cdot \nabla w$$

Gebrauch gemacht haben. Wie im 1-D-Fall wird (5) für große Reynolds-Zahlen $Re = \|\beta\|/\varepsilon$ instabil. Überträgt man nun die obigen Überlegungen (zunächst etwa anhand des Beispiels $\beta = (1, 0)$) auf die Diskretisierung (5), so wird deutlich, daß die Addition stabilisierender, künstlicher Diffusion nur in Flußrichtung β physikalisch sinnvoll ist. Zusätzlich eingeschleppte Diffusion senkrecht zu β (crosswind) ist dagegen nicht zu rechtfertigen und verfälscht die Ergebnisse.

Der Diffusion in Richtung β entspricht formal der Term (u_β, v_β) . Um diesen Term in die Diskretisierung einzuführen, wurde von Hughes und Brooks [5] vorgeschlagen, in (5) die Testfunktionen $v \in V_T$ durch $v + \delta h v_\beta$, $v \in V_T$, zu ersetzen. Man erhält so das Stromlinien-Diffusions-Verfahren.

Finde ein $u^T \in V_T$ mit $u^T|_{\Gamma_0} = u_0$, so daß gilt

$$\varepsilon(\nabla u^T \cdot \nabla v) + (u_\beta^T, v + \delta h v_\beta) = (f, v + \delta h v_\beta) \quad (6)$$

für alle $v \in V_T$ mit $v|_{\Gamma_0} = 0$.

Dabei ist

$$\frac{\partial}{\partial x} v_\beta|_t = \frac{\partial}{\partial y} v_\beta|_t = 0, \quad t \in \mathcal{T}, \quad \text{für alle } v \in V_T$$

berücksichtigt worden. Man beachte das Auftreten des Stromlinien-Diffusionsterms $\delta h(u_\beta^T, v_\beta)$ mit dem upwind-Parameter $\delta \geq 0$.

Analog zum 1-D-Fall zeigen theoretische Untersuchungen wie auch numerische Experimente, daß das Stromlinien-Diffusions-Verfahren (6) nicht nur stabiler, sondern auch von höherer Genauigkeit ist als das Standard-Galerkin-Verfahren (5). Für Einzelheiten verweisen wir auf [7] und [8].

Allerdings hängt das Verhalten des Diffusionsterms (u_β^T, v_β) und damit die Güte des Verfahrens (6) wesentlich von der Richtung der Kanten der Dreiecke aus \mathcal{T} ab, wie folgendes einfache Beispiel zeigt.

Beispiel 1: Wir setzen in (M):

$\varepsilon = 10^{-4}$, $\beta = (1, 1)$, $f = 0$, $\Omega = [0, 1] \times [0, 1]$ sowie

$$u_0(x, y) = \begin{cases} 0, & \text{falls } y > 0 \\ 1, & \text{sonst} \end{cases} \quad (x, y) \in \Gamma_0$$

am Einflußrand $\Gamma_0 = \Gamma_{\text{in}}$ und $\Gamma_n = \partial\Omega \setminus \Gamma_0$. Die Bilder 1 und 2 zeigen die Höhenlinien der von (6) mit $\delta = 1$ gelieferten Lösungen zusammen mit den zugrundeliegenden Triangulierungen \mathcal{T}_1 und \mathcal{T}_2 , welche sich offenbar nur in der Richtung der Diagonalkanten unterscheiden. Während auf \mathcal{T}_1 die von der Randunstetigkeit ausgehende Grenzschicht optimal aufgelöst wird, ist sie auf \mathcal{T}_2 über mehrere Gitterlinien verschmiert.

Dieser Effekt wird verständlich, wenn man (6) als Differenzenverfahren schreibt. Dem Stromlinien-Diffusionsterm (u_β^τ, v_β) entsprechen im obigen Beispiel dann nämlich die Differenzensterne

$$\frac{1}{2h^2} \begin{pmatrix} \cdot & \cdot & -1 \\ \cdot & 2 & \cdot \\ -1 & \cdot & \cdot \end{pmatrix} \quad \frac{1}{h^2} \begin{pmatrix} \frac{1}{2} & -1 & \cdot \\ -1 & 3 & -1 \\ \cdot & -1 & \frac{1}{2} \end{pmatrix}$$

auf \mathcal{T}_1 auf \mathcal{T}_2 .

Während also \mathcal{T}_1 eine optimale Modellierung der gewünschten zweiten Ableitung in Richtung $\beta = (1, 1)$ zuläßt, degeneriert auf \mathcal{T}_2 der entsprechende Term zu einer Approximation des Laplace-Operators. Aus dem senkrecht zu β wirkenden Anteil dieser isotropen Diffusion resultiert offenbar die in Bild 2 zu beobachtende Verschmutzung der Näherungslösung. Damit liegt es nahe, wenigstens in der Umgebung von Grenzschichten die Gitterlinien der Flußrichtung anzupassen. Wir kommen auf diesen Punkt in Abschnitt 4 zurück.

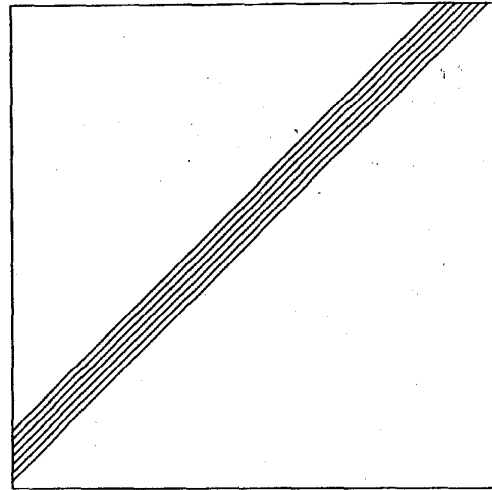
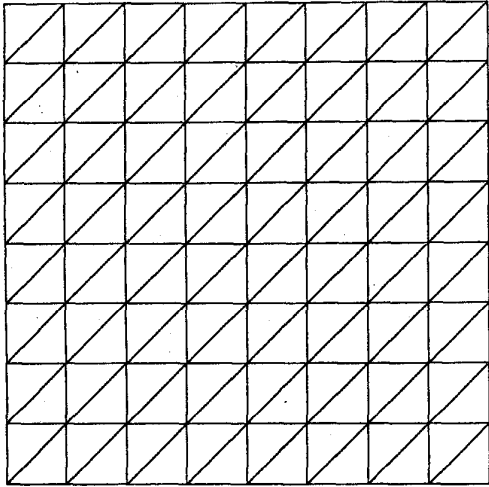


Bild 1 Stromlinien-Diffusionsverfahren auf \mathcal{T}_1 (optimale Kantenrichtung)

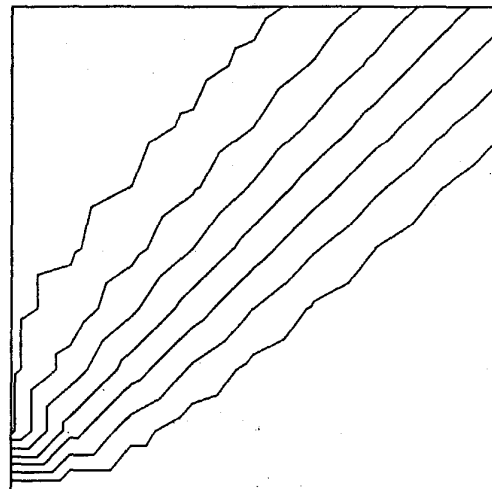
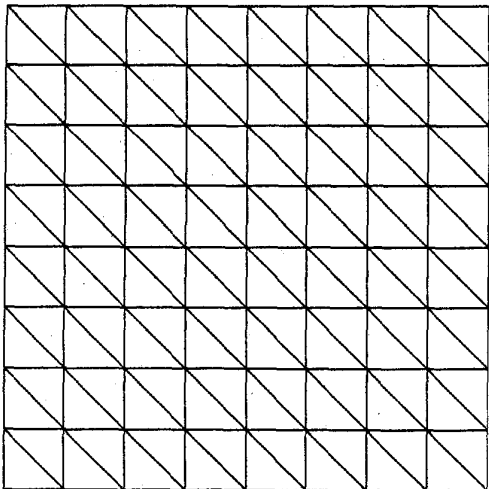


Bild 2 Stromlinien-Diffusionsverfahren auf \mathcal{T}_2 (ungünstige Kantenrichtung)

3 Richtungsorientierte Verfeinerung

Bei der Diskretisierung eines isotropen, rein elliptischen Problems erfolgt der Fehlertransport von einem festgehaltenen Gitterpunkt aus in alle Richtungen gleichmäßig. Dem entspricht die in alle Richtungen gleich wirkende reguläre (rote) Verfeinerung von Bank, Sherman und Weiser [1] (vergleiche auch [3]), bei der ein zu verfeinerndes Dreieck durch Halbierung der Kanten in vier ähnliche Dreiecke unterteilt wird (Bild 3).

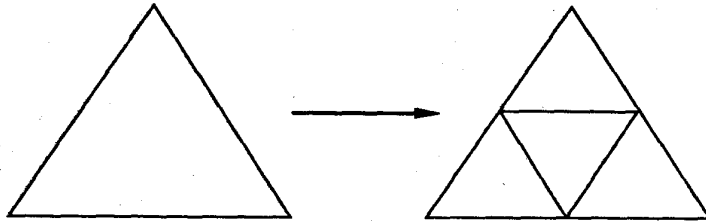


Bild 3 Reguläre (rote) Verfeinerung

Im konvektiv dominierten Fall sind dagegen durch Stromlinien oder eventuell auftretende Grenzschichten gewisse Richtungen ausgezeichnet, was sich auch in der Verfeinerungsstrategie widerspiegeln muß. So macht es beispielsweise keinen Sinn, zur Auflösung einer Grenzschicht die parallel zur Grenzschicht liegenden Kanten ebensooft zu verfeinern wie solche, die die Grenzschicht schneiden. Daher werden für Differenzenverfahren auch schon lange Gitter mit stark unterschiedlichen Schrittweiten in unterschiedliche Richtungen verwendet. Um entsprechende Triangulierungen adaptiv erzeugen zu können, bedarf es neben der regulären (roten) auch einer gerichteten (blauen) Verfeinerung (Bild 4).

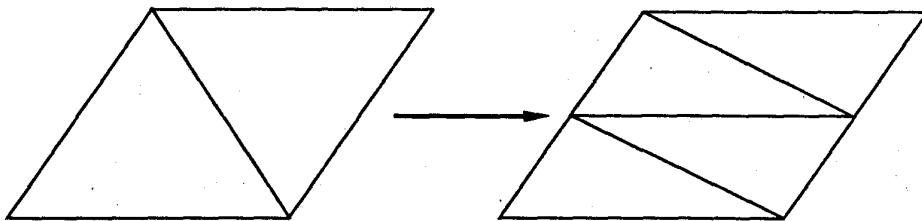


Bild 4 Gerichtete (blaue) Verfeinerung in y -Richtung

Läßt sich die verwendete Triangulierung – wie in Beispiel 2 – auf ein Rechteckgitter zurückführen, so bewirkt eine uniforme blaue Verfeinerung, etwa in

y -Richtung, gerade die Halbierung der Schrittweite in y -Richtung, während einer uniformen roten Verfeinerung die Halbierung beider Schrittweiten entspricht.

Die blaue Verfeinerung beliebiger Dreiecke mit gemeinsamer Kante ist weder möglich noch sinnvoll. Im einzelnen werden zwei Dreiecke t, t' (Bild 5) zur gemeinsamen blauen Verfeinerung senkrecht zu a zugelassen, wenn folgende drei Bedingungen erfüllt sind.

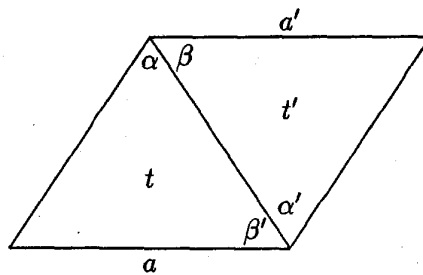


Bild 5 Verfeinerungskriterien

Die Kanten a und a' sind parallel. (1)

Die Kanten a und a' sollen nicht verfeinert werden. (2)

Sind α, α' die größten Winkel in t, t' , so müssen β, β' die kleinsten Winkel in t, t' sein. (3)

Die Bedingungen (1) und (2) definieren die Verfeinerungsrichtung, während die etwas technische Bedingung (3) dafür sorgt, daß die Innenwinkel der entstehenden Dreiecke nicht zu groß werden (vergleiche [6]). Anders als im rein elliptischen Fall schaden kleine Innenwinkel dagegen nicht. Im Gegenteil zeigen theoretische Untersuchungen an Modellproblemen, wie auch numerische Experimente, daß blau erzeugte Triangulierungen auf besser konditionierte Gleichungssysteme führen als man sie durch rote Verfeinerung erhält.

Beispiel 2: Wir setzen in (M):

$\epsilon = 10^{-4}$, $\beta = (1, 1)$, $f = 1$, $\Omega = [0, 1] \times [0, 1]$ und $u_0 \equiv 0$ auf $\Gamma_0 = \partial\Omega$. Ausgehend von der Ausgangstriangulierung (Bild 6) soll die sich am Ausflußrand bildende Grenzschicht adaptiv aufgelöst werden.

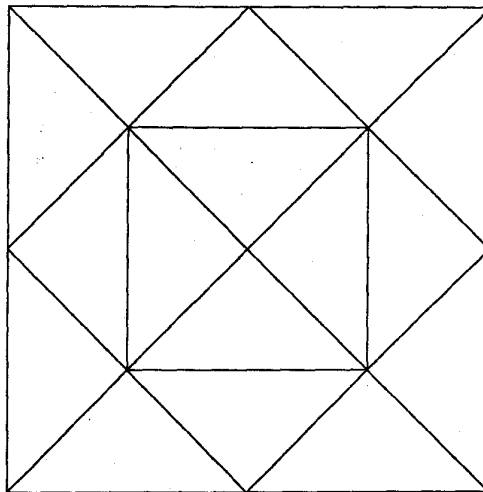


Bild 6 Ausgangstriangulierung für Beispiel 2

Dabei verwenden wir als provisorischen Fehlerschätzer schlicht die Steigung längs der jeweiligen Kante. Die auftretenden linearen Gleichungssysteme werden mit ORTHOMIN, einer Verallgemeinerung des CG-Verfahrens auf nichtsymmetrische Matrizen (siehe zum Beispiel [9]), iterativ gelöst. Als upwind-Parameter wählen wir $\delta = 1/2$ und in (2.6) wird die für nichtuniforme Triangulierungen sinnlose globale Schrittweite h durch die lokale Schrittweite in Flußrichtung ersetzt. Die Bilder 7 und 8 zeigen nun die durch lokale blaue und durchweg rote Verfeinerung erzeugten Triangulierungen \mathcal{T}_1 und \mathcal{T}_2 nebst den Höhenlinien der zugehörigen Lösungen. Offenbar hat man bei erheblich reduzierter Anzahl von Unbekannten keinerlei Einbußen an Genauigkeit. Die Anzahl der benötigten Iterationsschritte zur Lösung der Gleichungssysteme war für die rote Triangulierung \mathcal{T}_2 , bei etwa neunfachem Aufwand pro Iterationsschritt, fast doppelt so hoch wie im lokal blauen Fall. Insgesamt ergibt sich also nur durch Verwendung der blauen Verfeinerungsstrategie gegenüber der bisher üblichen Vorgehensweise eine Beschleunigung etwa um das achtzehnfache.

Im obigen Beispiel war die Bedingung (1) im Bereich der Grenzschicht trivialerweise erfüllt. Bei inneren Grenzschichten ist (1) im allgemeinen jedoch nur durch lokale Ausrichtung der Kanten zu gewährleisten. Darauf gehen wir im folgenden Abschnitt näher ein.

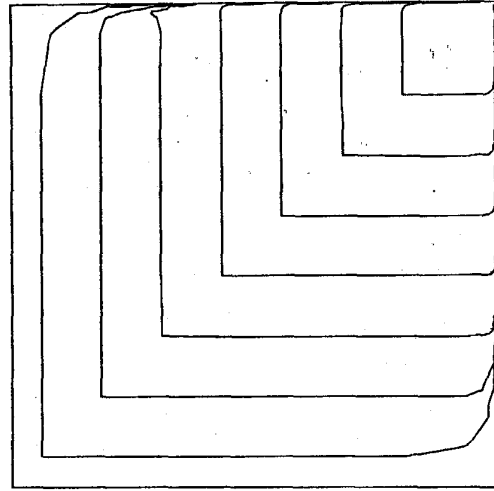
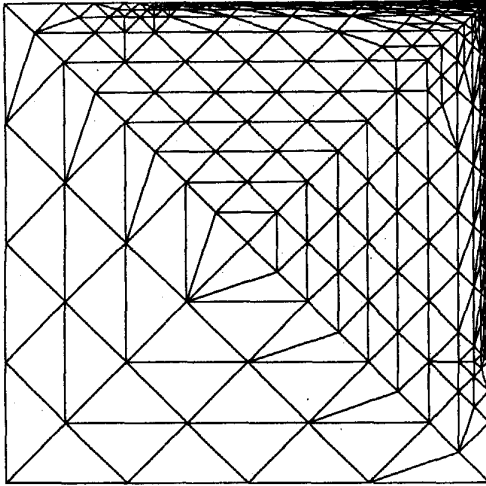


Bild 7 Beispiel 2 mit blauer Verfeinerung, 484 Knoten

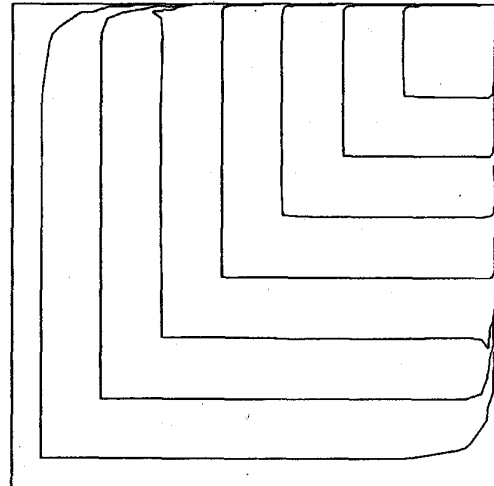
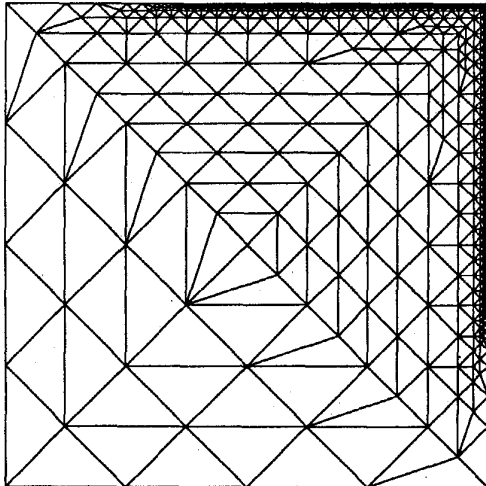


Bild 8 Beispiel 2 mit roter Verfeinerung, 1412 Knoten

4 Lokale Kantenausrichtung

Wie bereits gezeigt wurde, hat die Richtung der Kanten der verwendeten Triangulierung einen großen Einfluß auf das Verhalten der zugehörigen Näherungslösungen. In diesem Abschnitt soll daher ein Algorithmus zur adaptiven Ausrichtung der Kanten im Bereich von inneren Grenzschichten vorgestellt werden.

Die Grundidee besteht dabei darin, jeweils im Anschluß an einen Verfeinerungsschritt die neu entstandenen (aktuellen) Kanten im Bereich einer inneren Grenzschicht dem Verlauf dieser Grenzschicht anzupassen (Bild 9).

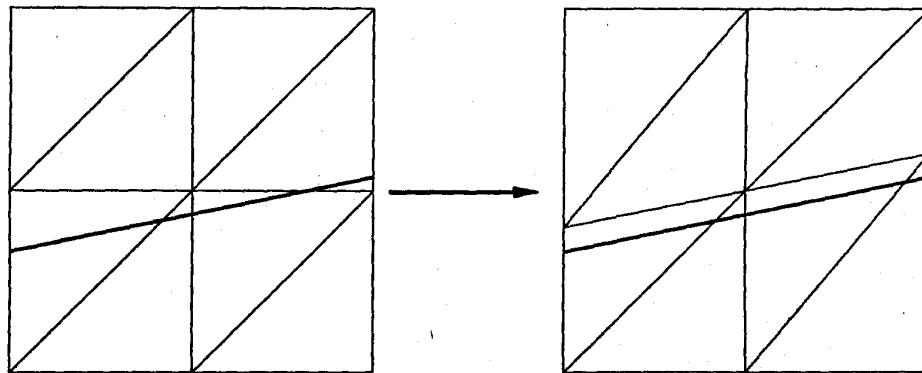


Bild 9 Ausrichten der Kanten

Die algorithmische Umsetzung dieser Idee zerfällt in zwei Teile: Grenzschichtschätzer und Kantenausrichter. Der Grenzschichtschätzer liefert dabei einen Polygonzug aus aktuellen Kanten, welcher anschließend vom Kantenausrichter dem Verlauf der Grenzschicht optimal angepaßt wird. Kenntnis über diesen Verlauf hat man dabei aus der jeweils aktuellen Näherungslösung. Der Aufwand für diese Prozedur ist im Verhältnis zum Gesamtaufwand aus zwei Gründen gering. Zunächst arbeitet der Algorithmus nur auf Punkten in der Nähe der Grenzschicht, deren Anzahl um eine Größenordnung kleiner ist als die Gesamtpunktzahl. Geht man zweitens von relativ glatten Grenzschichten aus, wie sie im allgemeinen in der Praxis vorkommen, so braucht das Ausrichten der Kanten nur auf entsprechend niedrigen Verfeinerungsstufen vorstatten zu gehen, da ja die bloße Richtung der Grenzschicht schon auf relativ groben Triangulierungen gut genug approximiert wird.

Anschließend sollen Arbeitsweise und Wirkung der adaptiven, lokalen Kantenausrichtung anhand eines einfachen Beispiels demonstriert werden.

Beispiel 3: Wir setzen in (M):

$\varepsilon = 10^{-4}$, $\beta = (y, -x)$, $f = 0$, $\Omega = [0, 1] \times [0, 1]$, sowie

$$u_0(x, y) = \begin{cases} 0, & \text{falls } y > \frac{2}{3} \\ 1, & \text{sonst} \end{cases} \quad (x, y) \in \Gamma_0$$

am Einflußrand $\Gamma_0 = \Gamma_{\text{in}}$ und $\Gamma_n = \partial \Omega \setminus \Gamma_0$. Die exakte Lösung bildet eine von der Unstetigkeit in u_0 ausgehende Grenzschicht in Form eines Viertelkreises aus, welche sich durch den Diffusionsanteil zum Ausflußrand hin verbreitert. Zur Diskretisierung verwenden wir das Stromlinien-Diffusions-Verfahren (6) mit dem upwind-Parameter $\delta = 1$. Fehlerschätzer und Gleichungslöser werden wie in Beispiel 2 gewählt. Ausgehend von derselben Ausgangstriangulierung (Bild 10) wurden adaptiv jeweils unter Verwendung lokaler Kantenausrichtung und in üblicher Weise die in den Bildern 11 und 12 dargestellten Triangulierungen \mathcal{T}_1 und \mathcal{T}_2 erzeugt.

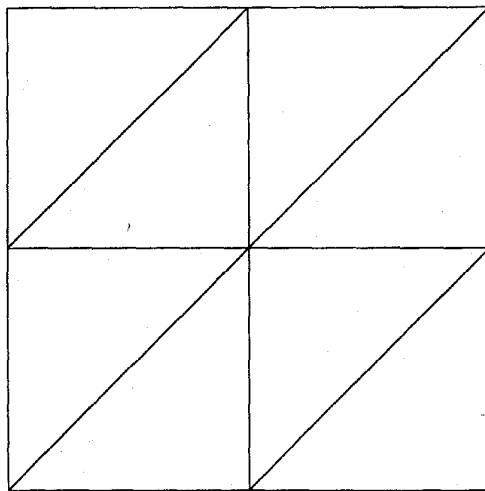


Bild 10 Ausgangstriangulierung für Beispiel 3

Die Höhenlinien der zu \mathcal{T}_1 und \mathcal{T}_2 gehörigen Näherungslösungen zeigt jeweils die rechte Seite der Bilder 11 und 12. Deutlich ist in Bild 12 der störende Einfluß von crosswind zu erkennen, insbesondere im mittleren Bereich, wo die Kanten ähnlich wie in Bild 2 senkrecht zur Flußrichtung liegen.

Schließlich geht die bessere Auflösung und höhere Stabilität der Näherung auf dem lokal ausgerichteten Gitter analog zum Beispiel 2 mit einer erheblichen Einsparung an Gitterpunkten und entsprechender Reduktion des Rechenaufwandes einher.

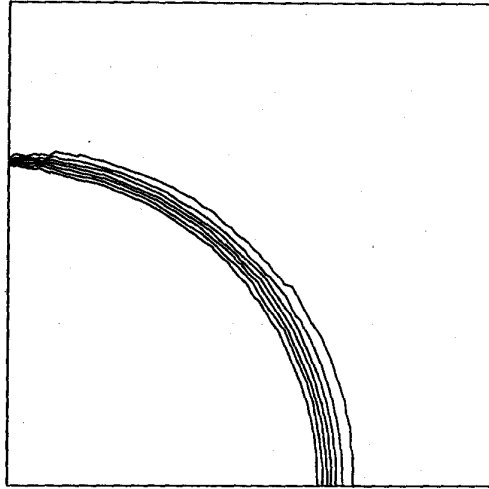
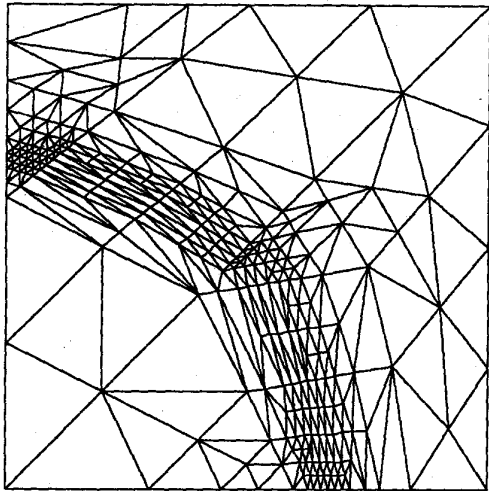


Bild 11 Beispiel 3 mit Drehung der Kanten, 243 Knoten

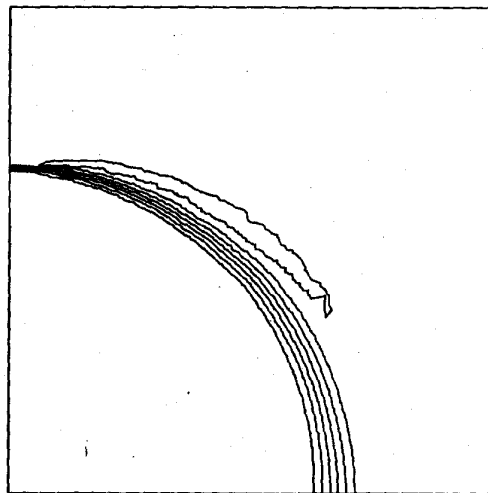
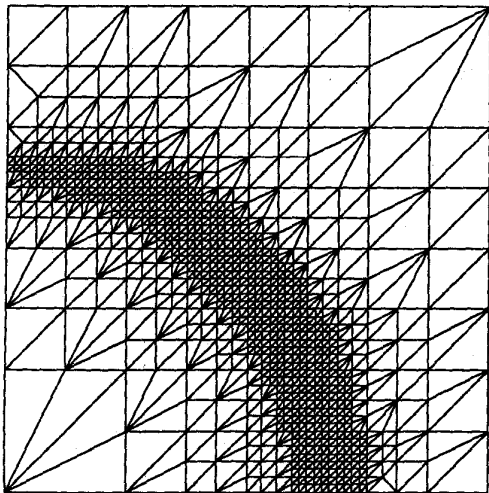


Bild 12 Beispiel 3 ohne Drehung der Kanten, 877 Knoten

Bibliographie

- [1] R.E. Bank, A.H. Sherman, A. Weiser: *Refinement Algorithms and Data Structures for Regular Local Mesh Refinement*. In: Scientific Computing, R. Stepleman et al. (eds.), Amsterdam: IMACS/North Holland, p. 3-17 (1983).
- [2] R. Courant, K.O. Friedrichs, H. Lewy: *Über die partiellen Differentialgleichungen der Physik*. Math. Ann. 100, p. 32-74 (1928).
- [3] P. Deuffhard, P. Leinen, H. Yserentant: *Concepts of an Adaptive Hierarchical Finite Element Code*. Konrad-Zuse-Zentrum Berlin, Preprint SC 88-5 (1988).
- [4] C.A.J. Fletcher: *Computational Galerkin Methods*. Springer-Verlag New York (1984).
- [5] T.J.R. Hughes, A. Brooks: *A Multidimensional Upwind Scheme with no Crosswind Diffusion*. In: Finite Element Methods for Convection Dominated Flows. T.J.R. Hughes (ed.)- AMD Vol. 34. (ASME, New York) p. 19-35 (1979).
- [6] P. Jamet: *Estimation d'erreur pour des éléments finis droits presque dégénérés*. R.A.I.R.O., Série Analyse Numérique, 10, no 3, p. 43-61 (1976).
- [7] C. Johnson: *Numerical Solutions of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, Cambridge (1987).
- [8] U. Nävert: *A Finite Element Method for Convection-Diffusion Problems, Thesis*. Department of Computer Science, Chalmers University of Technology, Göteborg, Sweden (1982).
- [9] J. Stoer: *Solution of Large Systems of Linear Equations by Conjugate Gradient Type Methods*. In: Mathematical Programming the State of the Art, A. Bachem et al. (eds.). Springer-Verlag Berlin, p. 540-565 (1983).

Veröffentlichungen des Konrad-Zuse-Zentrums für Informationstechnik Berlin
Preprints

Dezember 1988

SC 86-1. P. Deuffhard; U. Nowak. *Efficient Numerical Simulation and Identification of Large Chemical Reaction Systems.*

SC 86-2. H. Melenk; W. Neun. *Portable Standard LISP for CRAY X-MP Computers.*

SC 87-1. J. Anderson; W. Galway; R. Kessler; H. Melenk; W. Neun. *The Implementation and Optimization of Portable Standard LISP for the CRAY.*

SC 87-2. Randolph E. Bank; Todd F. Dupont; Harry Yserentant. *The Hierarchical Basis Multigrid Method.* (vergriffen)

SC 87-3. Peter Deuffhard. *Uniqueness Theorems for Stiff ODE Initial Value Problems.*

SC 87-4. Rainer Buhtz. *CGM-Concepts and their Realization.*

SC 87-5. P. Deuffhard. *A Note on Extrapolation Methods for Second Order ODE Systems.*

SC 87-6. Harry Yserentant. *Preconditioning Indefinite Discretization Matrices.*

SC 88-1. Winfried Neun; Herbert Melenk. *Implementation of the LISP-Arbitrary Precision Arithmetic for a Vector Processor.*

SC 88-2. H. Melenk; H.M. Möller; W. Neun. *On Gröbner Bases Computation on a Supercomputer Using REDUCE.*

SC 88-3. J. C. Alexander; B. Fiedler. *Global Decoupling of Coupled Symmetric Oscillators.*

SC 88-4. Herbert Melenk; Winfried Neun. *Parallel Polynomial Operations in the Buchberger Algorithm.*

SC 88-5. P. Deuffhard; P. Leinen; H. Yserentant. *Concepts of an Adaptive Hierarchical Finite Element Code.*

SC 88-6. P. Deuffhard; M. Wulkow. *Computational Treatment of Polyreaction Kinetics by Orthogonal Polynomials of a Discrete Variable.*

SC 88-7. H. Melenk; H. M. Möller; W. Neun. *Symbolic Solution of Large Stationary Chemical Kinetics Problems.*

SC 88-8. Ronald H. W. Hoppe; Ralf Kornhuber. *Multi-Grid Solution of Two Coupled Stefan Equations Arising in Induction Heating of Large Steel Slabs.*

Veröffentlichungen des Konrad-Zuse-Zentrums für Informationstechnik Berlin
Technical Reports Dezember 1988

- TR 86-1. H.J. Schuster. *Tätigkeitsbericht 1985. (vergriffen)*
- TR 87-1. Hubert Busch; Uwe Pöhle; Wolfgang Stech. *CRAY-Handbuch. - Einführung in die Benutzung der CRAY..*
- TR 87-2. Herbert Melenk; Winfried Neun. *Portable Standard LISP Implementation for CRAY X-MP Computers. Release of PSL 3.4 for COS.*
- TR 87-3. Herbert Melenk; Winfried Neun. *Portable Common LISP Subset Implementation for CRAY X-MP Computers.*
- TR 87-4. Herbert Melenk; Winfried Neun. *REDUCE Installation Guide for CRAY 1 / X-MP Systems Running COS Version 3.2.*
- TR 87-5. Herbert Melenk; Winfried Neun. *REDUCE Users Guide for the CRAY 1 / X-MP Series Running COS. Version 3.2.*
- TR 87-6. Rainer Buhtz; Jens Langendorf; Olaf Paetsch; Danuta Anna Buhtz. *ZUGRIFF - Eine vereinheitlichte Datenspezifikation für graphische Darstellungen und ihre graphische Aufbereitung.*
- TR 87-7. J. Langendorf; O. Paetsch. *GRAZIL (Graphical ZIB Language).*
- TR 88-1. Rainer Buhtz; Danuta Anna Buhtz. *TDLG 3.1 - Ein interaktives Programm zur Darstellung dreidimensionaler Modelle auf Rastergraphikgeräten.*
- TR 88-2. Herbert Melenk; Winfried Neun. *REDUCE User's Guide for the CRAY 1 / CRAY X-MP Series Running UNICOS. Version 3.3.*
- TR 88-3. Herbert Melenk; Winfried Neun. *REDUCE Installation Guide for CRAY 1 / X-MP Systems Running UNICOS. Version 3.3.*
- TR 88-4. Danuta Anna Buhtz; Jens Langendorf; Olaf Paetsch. *GRAZIL-3D. Ein graphisches Anwendungsprogramm zur Darstellung von Kurven- und Funktionsverläufen im räumlichen Koordinatensystem.*
- TR 88-5. Gerhard Maierhöfer; Georg Skorobohatyj. *Ein paralleler, adaptiver Algorithmus zur numerischen Integration ; seine Implementierung für SUPRENUM-artige Architekturen mit SUSI.*