

Diplomarbeit

von

Farzaneh Mehdizadeh-Latreh

**Adaptive spektrale Least-Squares Verfahren
für die Konvektions-Diffusions-Gleichung
mit Triangulierung**

Prof. Dr. Wilhelm Heinrichs

Fakultät der Mathematik, Universität Duisburg-Essen

Danksagung

Zu Beginn möchte ich mich bei allen Menschen bedanken, die mir die Möglichkeit erschafft haben diese Arbeit zu erstellen:

Herrn Prof. Dr. Heinrichs für die wissenschaftliche Betreuung und Beratung während der ganzen Arbeit.

Peter Norek, Christoph Colberg, Pascal Marquardt und Rajaramesh Rajendiram für die wertvollen Diskussionen und das Korrekturlesen.

Zu guter Letzt bedanke ich mich bei meinem Ehemann und meinen Eltern für ihre Geduld, Unterstützung und Ermunterung.

Erklärung

Hiermit versichere ich, diese Diplomarbeit selbständig und nur unter Verwendung der im Literaturverzeichnis angegebenen Hilfsmittel angefertigt zu haben.

Essen, im Juli 2013

Farzaneh Mehdizadeh-Latreh

Inhaltsverzeichnis

1	Grundlage	10
1.1	Spektrale Kollokation	11
1.2	Least-Squares Verfahren	11
1.2.1	Normalengleichungen	12
1.2.2	Die Pseudoinverse einer Matrix	12
1.2.3	QR-Zerlegung	14
2	Dreiecksbasierte Spektrale-Elemente-Methode	15
2.1	Dubiner-Basis	15
2.2	Fekete-Punkte	16
2.2.1	Quadraturformel	18
3	Singulär-gestörte Probleme	19
3.1	Singulär-gestörte Differentialgleichungen	19
3.2	Konvektions-Diffusions-Gleichungen	20
4	Diskretisierung	22
4.1	Triangulierung	22
4.2	Formulierung als System	24
4.2.1	Die Randoperatoren	27
4.2.2	Die Interface-Operatoren	29
4.2.3	Kollokations-System	31
4.3	Die skalare Formulierung	35
4.3.1	Kollokations-System	36
5	Adaptive Verfeinerungsstrategien	38
5.1	Markierungsstrategie	38
5.1.1	Gradienten-Kriterium	38
5.1.2	Abfallende Entwicklungskoeffizienten	39
5.1.3	Spektrums-Ende	41
5.2	Verfeinerungsregel	42
6	Numerische Ergebnisse	43
6.1	Beispiel 1: moderater Boundary-Layer	44
6.1.1	Ergebnisse für die Formulierung als System	45
6.1.2	Ergebnisse für die skalare Formulierung	47
6.1.3	Skalare vs. System	49
6.1.4	Adaptive Verfeinerung	52
6.2	Beispiel 2: Verschärfter Boundary-Layer	57
6.2.1	Ergebnisse für die Formulierung als System	58

6.2.2	Ergebnisse für die skalare Formulierung	60
6.2.3	Skalare vs. System	62
6.2.4	Adaptive Verfeinerung	63
6.3	Beispiel 3: Hutlayer	68
6.3.1	Ergebnisse für die Formulierung als System	69
6.3.2	Ergebnisse für die skalare Formulierung	71
6.3.3	Skalare vs. System	74
6.3.4	Adaptive Verfeinerung	76
6.4	Beispiel 4: Eckenlayer	81
6.4.1	Ergebnisse für die Formulierung als System	82
6.4.2	Ergebnisse für die skalare Formulierung	84
6.4.3	Skalare vs. System	86
6.4.4	Adaptive Verfeinerung	87
7	Ausblick	92
	Abbildungsverzeichnis	94
	Tabellenverzeichnis	97
	Anhang A	98
	Anhang B	103
	Literaturverzeichnis	104
	Index	107

Einleitung

Eine der anspruchsvollsten und omnipräsentesten Aufgaben der numerischen Approximation partieller Differentialgleichungen ist die exakte Modellierung der Interaktion zwischen diffusiven und konvektiven Prozessen. Ein Grund dafür liegt in der breiten Vielfalt der Konvektions-Diffusions-Probleme. Ferner sind diese mit singular-gestörten Problemen und den dazugehörigen Boundary-Layer-Theorien eng verbunden. Oft ist der dimensionslose Parameter ε , welcher mit der zweiten Ableitung multipliziert wird, sehr klein. Der Term bestehend aus der zweiten Ableitung modelliert die Diffusion. Der Konvektions-Term wird mit der ersten Ableitung nachgebildet. Bei solchen Problemen mit $\varepsilon \rightarrow 0^+$ entstehen dünne Boundary- oder Interiorlayer, welche dünne Schichten bezeichnen, wo die Singularität erhöht wird bzw. sich die Lösung stark variiert. Es gibt eine große Anzahl von Veröffentlichungen, die in den letzten Jahrzehnten verschiedene Techniken für Analyse und Überwindung solcher Probleme beschrieben haben [17].

In [15] werden mit Hilfe einer Viereckszerlegung des Grundgebietes zwei Formulierungsmethoden zum Lösen der Konvektions-Diffusions-Probleme untersucht: die skalare Formulierung, welche direkten Bezug auf die zweite Ableitung nimmt, und die Formulierung als System, welche eine Reduzierung der Ordnung $p = 2$ des Problems auf $p = 1$ vornimmt.

In der vorliegenden Arbeit versuchen wir die gleichen Probleme aus [15] mit Hilfe einer Dreieckszerlegung des Grundgebietes zu lösen. Dabei verwenden wir die Fekete-Punkte der Ordnung $N \in \{3, 6, 9, 12\}$ als Kollokationspunkte.

Zuerst möchten wir wissen, wie sich die beiden in [15] vorgestellten Formulierungsmethoden in einem triangulierten Gebiet verhalten. In [15] zeigen die Schlussfolgerungen, dass die skalare Formulierung effizienter ist, obwohl man bei der Formulierung als System Genauigkeitsvorteile beobachtet. Die Unterschiede der Ergebnisse beider Methoden sind stets gering, sodass man angesichts der numerischen Kosten die skalare Formulierung vorzieht.

Wir testen diese beiden Methoden in einem nicht-adaptiven Gitter, welches durch die Friedrichs-Keller-Triangulierung in Dreieckselemente zerlegt wird.

Das Problem der singular-gestörten Aufgaben ist das Vorhandensein von Boundary- oder Interiorlayern. Diese beeinträchtigen die approximierten Lösung. Durch eine adaptive Verfeinerung des Grundgebietes versuchen wir diese Probleme abzumildern und damit den Approximationsfehler zu verringern.

Durch die Triangulierung erhalten wir eine bessere Auflösung des Gebietes, vor allem gezielt in den Bereichen, wo ein Bedarf nach Verfeinerung besteht. Die Entscheidung, in welchem Element eine Verfeinerung vorgenommen werden soll, treffen wir mit Hilfe des sogenannten a-posteriori Fehlerschätzers. Ein a-posteriori Fehlerschätzer nutzt die Informationen der approximierten Lösung, um eine berechenbare Schranke für den Approximationsfehler zu liefern. Diese Schranke ist entscheidend dafür, ob das Gitter bzw. ein Element des Gitters adaptiv verfeinert werden soll.

In der vorliegenden Arbeit testen wir drei a-posteriori Fehlerschätzer, welche unter Adaptivitätskriterien vorgestellt werden.

Heinrichs hat in [7] für die Ausführung einer adaptiven Verfeinerung mittels Triangulierung das Gradienten-Kriterium angewandt. Dieses kommt zum Einsatz, wenn keine anderen a-posteriori Fehlerschätzer zur Verfügung stehen. Wir bestimmen die Gradienten in jedem Element mit Hilfe der approximierten Lösungen. Angesichts einer vorgegebenen Schranke für die Gradienten entscheiden wir, ob ein Element zur Verfeinerung markiert werden soll.

In [18] werden zwei weitere Adaptivitätskriterien präsentiert. Das zweite Kriterium nutzt den direkten Vorteil von hochgradigen Polynomial-Basen. Die beste polynomiale Approximation für eine gegebene Funktion u ist die Projektion von u in einen Polynomraum. Bei dem zweiten Kriterium nutzen wir die Entwicklungskoeffizienten bzgl. der Projektion von u . Wir bilden aus dem zweidimensionalen Spektrum ein äquivalentes eindimensionales diskretes Spektrum, welches wir durch die Approximation einer absteigenden Exponentialfunktion ersetzen. Diese ist eine Least-Squares-Beste-Anpassung der letzten vier Terme des diskreten Spektrums und wird mit Hilfe einer linearen Regression bestimmt. Damit sind wir in der Lage durch eine Extrapolation die fehlenden Spektrumswerte für höhere Exponenten zu bestimmen. Schließlich erhalten wir eine Abschätzung für den Approximationsfehler $\|u - u_h\|$, wobei u_h die approximierte Lösung ist. Das bedeutet, wir erzeugen eine Abschätzung des Approximationsfehlers, indem wir den Rest des Spektrums untersuchen. Der Nachteil des zweiten Kriteriums ist, dass dieses aufgrund der linearen Regression hauptsächlich für Polynomgrade $N \geq 6$ vertrauenswürdige Ergebnisse liefert und daher vielmehr für höhere Polynomgrade geeignet ist. Für weitere Details verweisen wir auf [19].

Die Einschränkung des zweiten Kriteriums auf höhere Polynomgrade wird beim dritten Kriterium beseitigt. Das dritte Kriterium, genannt *Spektrums-Ende*, ist simpler als das zweite Kriterium, enthält nur die Summe des Spektrums-Ende und liefert fast ähnliche Ergebnisse wie das zweite Kriterium.

Diese drei Adaptivitätskriterien untersuchen wir bei der adaptiven Verfeinerung für vier verschiedene Konvektions-Diffusions-Probleme, die wir aus [15] entnommen haben. Dabei approximieren wir die Lösung durch die Formulierung als System.

Aufbau

Die Diplomarbeit ist wie folgt strukturiert:

- In **Kapitel 1** beschreiben wir zuerst die spektrale Kollokationsmethode und zum Lösen des daraus resultierenden überbestimmten linearen Gleichungssystems einige direkte Löser von Least-Squares-Problemen.
- In **Kapitel 2** erläutern wir die Notwendigkeit für die dreiecksbasierte spektrale-Element-Methode. Die Begriffe wie Dubiner-Basis und Fekete-Punkte werden in diesem Kapitel erläutert.
- In **Kapitel 3** geben wir eine kurze Einführung in singular-gestörte Probleme und die Konvektions-Diffusions-Gleichung.
- In **Kapitel 4** präsentieren wir die Diskretisierungsmethoden jeweils für die Formulierung als System und die skalare Gleichung mit sämtlichen notwendigen Differenzial-, Rand- und Interface-Operatoren, die wir in dieser Arbeit durchführen.
- In **Kapitel 5** werden wir adaptive Verfeinerungsstrategien aufzeigen. Hierbei stellen wir drei Adaptivitätskriterien als Markierungsstrategien vor, die wir in der vorliegenden Arbeit untersuchen werden. Der Umgang mit den sogenannten hängenden Knoten wird ebenso in diesem Kapitel beschrieben.
- In **Kapitel 6** werden die numerischen Ergebnisse dargestellt, welche durch die Ausführung der Formulierung als System und der skalaren Gleichung erzielt werden. Wir vergleichen die beiden Verfahren mit denen der Viereckszerlegung aus [15]. Danach wenden wir die adaptive Verfeinerung an und analysieren die drei Adaptivitätskriterien.
- In **Kapitel 7** ziehen wir Schlussfolgerungen zu den in der Arbeit angewandten Methoden und geben einen kurzen Ausblick.

1 Grundlage

Partielle Differentialgleichungen treten in fast allen Bereichen der Wissenschaft auf. Zwei flexible und stark universale Verfahren sind die Finite-Differenzen- und die Finite-Elemente-Methode, die eine lange Geschichte in diesem Bereich haben. In den letzten Jahrzehnten sind spektrale Verfahren, insbesondere spektrale Kollokations-Methode, als alternative Verfahren in vielen Situationen in den Fokus der Wissenschaft gerückt. Eine besondere Eigenschaft der spektralen Verfahren ist ihre hohe spektrale Genauigkeit, wenn die Lösungsfunktion analytisch ist. Während die Finite-Differenzen- und die Finite-Elemente-Methode auch für irreguläre Geometrien gut geeignet sind, zeigen spektrale Verfahren hier ihre Schwäche. Zudem wird man bei spektralen Verfahren meist mit vollbesetzten und unsymmetrischen Matrizen konfrontiert, welche wiederum ein Nachteil gegenüber den ersten beiden Methoden darstellen.

Wir betrachten die folgende allgemeine partielle Differentialgleichung:

$$\begin{aligned}\mathcal{L}u &= f & \text{in } \Omega &= (-1, 1)^2 \\ \mathcal{B}u &= g & \text{auf } \partial\Omega\end{aligned}\tag{1.1}$$

wobei \mathcal{L} einen beliebigen Differentialoperator und \mathcal{B} den Randoperator symbolisiert.

Bei spektralen Methoden versucht man das Residuum $R(x) := \mathcal{L}u_N(x) - f(x)$ möglichst klein zu halten, wobei u_N hier eine approximierte Lösung von (1.1) bezeichnet.

Dazu wählt man eine endliche Menge von sogenannten Entwicklungsfunktionen Φ_j für $j = 0, \dots, N - 1$ und entwickelt u_N in diese Funktionen:

$$u_N(x) = \sum_{j=0}^{N-1} \hat{u}_j \Phi_j(x).\tag{1.2}$$

Im nächsten Schritt sucht man eine Menge von sogenannten Testfunktionen χ_n mit $n = 0, \dots, N - 1$ aus und fordert, dass das Skalarprodukt von Testfunktionen χ_n und Residuum R gleich Null ist. Das heißt, $\langle \chi_n, R \rangle = 0$ für $n = 0, \dots, N - 1$ gilt.

Es gibt verschiedene spektrale Methoden, die sich durch die Wahl der Testfunktionen χ_n unterscheiden. Eines dieser Verfahren heißt *Kollokation* oder *pseudo-spektrale Methode*. Diese werden wir im folgenden Abschnitt kurz beschreiben.

1.1 Spektrale Kollokation

Die Kollokation oder pseudo-spektrale Methode besitzt die Dirac-Funktion als Testfunktionen in der Form $\chi_n = \delta(x - x_n)$, wobei x_n die sogenannten *Kollokationspunkte* sind.

Somit haben wir:

$$\begin{aligned} 0 &= \langle \chi_n, R \rangle = R(x_n) = Lu_N(x_n) - f(x_n) \\ &\implies \sum_{j=0}^{N-1} \hat{u}_j L\Phi_j(x_n) = f(x_n), \quad \text{für } n = 0, \dots, N-1 \end{aligned}$$

wobei L die zugehörige pseudo-spektrale Ableitungsmatrix bezeichnet.

Dadurch erhält man ein Gleichungssystem mit Entwicklungskoeffizienten \hat{u}_j als Unbekannte. Zu dem resultierenden Gleichungssystem werden die Randbedingungen hinzugefügt. Schließlich haben wir ein überbestimmtes Gleichungssystem zu lösen.

Hierbei gibt es verschiedene direkte Löser für überbestimmte Gleichungssysteme. Im Folgenden gehen wir auf diese Verfahren näher ein.

1.2 Least-Squares Verfahren

Ein grundlegendes Problem der Wissenschaft ist, ein passendes Modell zu einer Menge von Messwerten in Abhängigkeit von Fehlern zu bestimmen. Dabei ist es klar, je höher die Anzahl der Messwerte sind, desto genauer können die Modellparameter bestimmt werden. Dies führt uns zu dem Problem **Lösen eines überbestimmten linearen oder nicht-linearen Gleichungssystems**.

Es sei $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$ mit $m > n$, dann ist $Ax = b$ mit Unbekanntenvektor x ein überbestimmtes Gleichungssystem. Eine optimale Lösung ist dann diejenige, die $\|b - Ax\|_2$ minimiert, wobei hier $\|\cdot\|_2$ die euklidische Norm ist. Dieses Minimierungsproblem kann also auch als eine Art Optimierungsproblem betrachtet werden.

Die Frage der Existenz einer optimalen Lösung wird durch das folgende Theorem beantwortet:

Theorem 1.1 *A, b, m und n seien wie oben vorausgesetzt. Das Least-Squares Problem $\min_{x \in \mathbb{R}^n} \|b - Ax\|_2$ besitzt mindestens eine optimale Lösung x_0 .*

Beweis: Siehe [2]

□

1.2.1 Normalgleichungen

Aus der Charakterisierung von Lösungen des Least-Squares-Problems (LSQP) geht hervor, dass ein Vektor x genau dann eine optimale Lösung des LSQP ist, wenn x die Orthogonalitätsbedingung $A^\top (b - Ax) = 0$ erfüllt, das heißt, x eine Lösung der sogenannten Normalgleichungen $A^\top Ax = A^\top b$ ist.

Theorem 1.2 *Es sei $A \in \mathbb{R}^{m \times n}$ mit $m > n$. Wenn $\text{rang}(A) = n$ ist, dann ist die Lösung x des LSQP eindeutig bestimmt und ist gegeben durch*

$$x = (A^\top A)^{-1} A^\top b$$

Beweis: Siehe [1]

□

Das Problem der Normalgleichung liegt in $\text{rang}(A) < n$. So hat A einen nicht trivialen Kern und somit hat LSQP keine eindeutige Lösung. In diesem Fall, wenn x_0 eine spezielle Lösung des LSQP ist, dann kann die Lösungsmenge des LSQP durch $\mathbb{L} = \{x_0 + x : x \in \text{kern}(A)\}$ angegeben werden.

Die klassische Methode zum Lösen der Normalgleichung basiert auf der Cholesky-Zerlegung.

Für dünnbesetzte Matrizen gibt es diverse Programmpakete wie **YSMP** (Yale sparse matrix package), **SuperLU**, **CHOLMOD** und **UMFPACK**. Das Programmpaket UMFPACK ist für die LU-Zerlegung dünnbesetzter unsymmetrischer Koeffizientenmatrix A entwickelt und kann auch in MATLAB ausgeführt werden (vgl. [2, 21]).

1.2.2 Die Pseudoinverse einer Matrix

Die Pseudoinverse einer Matrix beruht sich auf der *Singulärwertzerlegung* der Matrix, die für die Behandlung des LSQP von großer Bedeutung ist.

Jede Matrix $A \in \mathbb{R}^{m \times n}$ mit $\text{rang}(A) = r$ besitzt eine Singulärwertzerlegung der Form:

$$A = U \Sigma V^\top \tag{1.3}$$

mit

$$\Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix},$$

$U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ die orthogonalen Matrizen, und $\Sigma \in \mathbb{R}^{m \times n}$,

$\Sigma_1 = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$ mit $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ die sog. Singulärwerte von A .

Definition 1.1 Die Pseudoinverse A^\dagger einer Matrix A mit

$$A = U \begin{pmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{pmatrix} V^\top$$

ist definiert durch

$$A^\dagger = V \begin{pmatrix} \Sigma_1^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^\top.$$

Für eine reguläre Matrix A gilt $A^{-1} = A^\dagger$.

Die Singulärwertzerlegung ist ein wichtiges Mittel zum Lösen des LSQP. Vor allem weil die euklidische Norm der Vektoren durch die orthogonalen Matrizen unverändert bleibt.

Theorem 1.3 Es sei $A \in \mathbb{R}^{m \times n}$ mit $\text{rang}(A) = r \leq \min(m, n)$. Dann besitzt das LSQP

$$\min_{x \in \mathbb{L}} \|x\|_2, \quad \mathbb{L} = \{x \in \mathbb{R}^n \mid \|b - Ax\|_2 = \min\}$$

eine eindeutige Lösung $x = A^\dagger b$.

Beweis: Siehe Theorem 1.2.10 in [1]. □

$x = A^\dagger b$ heißt eine optimale Lösung des LSQP, wenn sie unter allen Lösungen die kleinste euklidische Norm besitzt.

Die Pseudoinverse Matrix A^\dagger ist auch als *Moore-Penrose-Inverse* bekannt und ist durch die folgenden Eigenschaften eindeutig charakterisiert:

Theorem 1.4 (Penrose Bedingungen). Die Pseudoinverse $X = A^\dagger$ von $A \in \mathbb{R}^{m \times n}$ ist durch die folgenden Bedingungen eindeutig bestimmt:

- 1) $AXA = A$
- 2) $XAX = X$
- 3) $(AX)^\top = AX$
- 4) $(XA)^\top = XA$

Beweis: Siehe [2]. □

Insbesondere folgt daraus, dass A^\dagger von der Wahl von U und V unabhängig ist.

Nachteil der Singulärwertzerlegung ist ihr hoher Aufwand in vielen Anwendungen. Deshalb greift man auf einfacheren Zerlegungsmethoden zurück. Unter diesen Methoden ist die QR-Zerlegung eine der wichtigsten.

1.2.3 QR-Zerlegung

Wir betrachten die Matrix $A \in \mathbb{R}^{m \times n}$, $m \geq n$ und $b \in \mathbb{R}^m$. Ferner sei $Q \in \mathbb{R}^{m \times m}$ eine orthogonale Matrix.

Eine der wichtigsten Eigenschaften der orthogonalen Matrizen ist, dass die euklidische Norm einer Matrix oder eines Vektors durch Multiplikation mit einer orthogonalen Matrix unverändert bleibt. Dies führt uns zur folgenden Äquivalenz:

$$\min_x \|Q^\top (b - Ax)\|_2 = \min_x \|b - Ax\|_2$$

Zudem gibt es zu jeder Matrix $A \in \mathbb{R}^{m \times n}$, $m \geq n$, eine orthogonale Matrix $Q \in \mathbb{R}^{m \times m}$ und somit eine QR-Zerlegung der folgenden Form:

$$A = Q \begin{pmatrix} R \\ 0 \end{pmatrix}$$

wobei R eine obere Dreiecksmatrix mit nicht negativen Diagonalelementen ist und R-Faktor genannt wird.

Im Falle von $\text{rang}(A) < n$ kann A mehrere QR-Zerlegungen besitzen. Dagegen ist bei $\text{rang}(A) = n$ die Zerlegung und somit die Lösung des LSQP eindeutig bestimmt:

$$\min_x \|b - Ax\|_2 = \min_x \left\| Q^\top b - \begin{pmatrix} R \\ 0 \end{pmatrix} x \right\|_2$$

mit $R \in \mathbb{R}^{n \times n}$, $Q^\top b =: \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}$, $d_1 \in \mathbb{R}^n$ und $d_2 \in \mathbb{R}^{m-n}$.

Die Lösung x des LSQP ist durch $Rx = d_1$ eindeutig festgelegt.

2 Dreiecksbasierte Spektrale-Elemente-Methode

Bei dreiecksbasierten Spektrale-Elemente-Methoden benötigt man eine Menge von Basis-Funktionen, welche in den Stützstellen jedes Dreieckselements definiert sind.

In einem Viereckselement kann man die Basis-Funktionen durch das Tensor-Produkt der eindimensionalen Basis-Funktionen bilden. Die dadurch entstehenden Funktionen sind auf dem Gebiet, welches ein kartesisches Produkt von Intervallen ist, definiert und vererben die Eigenschaften der zugehörigen eindimensionalen Basis-Funktionen.

Für ein Nicht-Tensor-Produkt-Gebiet, also insbesondere für ein Dreieckselement, ist dieser Ansatz nicht mehr möglich. In den späten 90er Jahren wurden die sog. *Dubiner-Basis* als orthogonale Basis von Polynomen und die *Fekete-Punkte* als Stützpunkte eines Dreiecks vorgeschlagen.

2.1 Dubiner-Basis

Wir betrachten das folgende Standard-Dreieck:

$$\mathbb{T} = \{(x, y) : -1 \leq x, y; x + y \leq 0\}. \quad (2.1)$$

Üblicherweise werden die Monome $\{x^m y^n : 0 \leq m + n \leq N\}$ zur Interpolation auf \mathbb{T} verwendet. In der Praxis wird aber eine orthogonale Basis wie Dubiner-Polynomen bevorzugt, weil die Monome uns zu einer numerisch schlecht-konditionierten Vandermonde-Matrix führen.

Dubiner-Polynome sind ein verzerrtes Produkt von Jacobi-Polynomen. Sie sind in $L^2(\mathbb{T})$ orthogonal und wie folgt definiert :

$$\phi_{mn}(x, y) = P_m^{0,0} \left(2 \frac{1+x}{1-y} - 1 \right) \cdot (1-y)^m \cdot P_n^{2m+1,0}(y), \quad m+n \leq N \quad (2.2)$$

wobei $P_m^{\alpha,\beta}$ die Jacobi-Polynome der Ordnung m mit den Gewichten α und β sind.

Das Skalarprodukt über \mathbb{T} (2.1) kann wie folgt bestimmt werden:

$$\langle \phi_{mn}, \phi_{pq} \rangle_{\mathbb{T}} = \int_{\mathbb{T}} \phi_{mn} \phi_{pq} dx dy = \gamma_{mn} \delta_{mp} \delta_{nq} \quad (2.3)$$

δ_{mp} und δ_{nq} bezeichnen das Kronecker-Delta und γ_{mn} den Normierungsfaktor mit

$$\gamma_{mn} = \frac{2^{2m+1}}{(2m+1)(m+n+1)}. \quad (2.4)$$

2.2 Fekete-Punkte

Fekete-Punkte sind eine Menge von Punkten, welche unabhängig von den ausgewählten Basis-Funktionen den Betrag der Determinante einer Vandermonde-Matrix maximieren. Sie sind für jede Art von Geometrien definierbar.

Häufig benutzte Kollokationspunkte sind die Gauß-Lobatto Punkte, die aber ausschließlich in Tensor-Produkt-Gebieten Anwendung finden.

In [3] wurde gezeigt, dass die Fekete-Punkte ein n-dimensionales Tensor-Produkt der Gauß-Lobatto-Punkte sind.

Ferner sind die Fekete-Punkte auf Dreiecken oder Tetraedern bekannt und somit ist es möglich, die spektrale Kollokation, die auf Gauß-Lobatto-Punkten basiert, für solche Nicht-Tensor-Produkt-Gebiete zu erweitern. Das bedeutet, die Fekete-Punkte sind eine Art Verallgemeinerung der Gauß-Lobatto-Punkte.

In Abbildung 2.1 sehen wir die Darstellung der Fekete-Punkte vom Grad $N = 3, 6, 9$ und 12 im Standard-Dreieck \mathbb{T} (2.1).

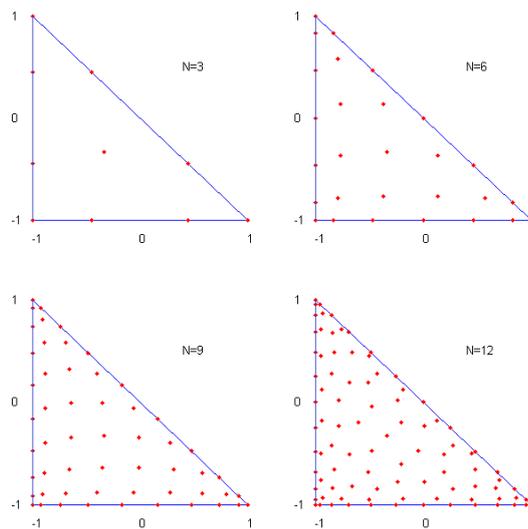


Abbildung 2.1: Fekete-Punkte vom Grad $N = 3, 6, 9$ und 12 im Standard-Dreieck \mathbb{T}

Für $N > 10$ haben die Fekete-Punkte die kleinste *Lebesgue Konstante* und somit bezeichnet man sie als optimale Interpolationspunkte in Dreiecken.

Wir betrachten eine beliebige Menge von Basis-Funktionen $\{\psi_j\}_{j=1}^{N_t}$ in dem Standard-Dreieck \mathbb{T} (2.1) mit

$$N_t = \frac{(N+1)(N+2)}{2}. \quad (2.5)$$

Die Fekete-Punkte sind dann diejenigen Punkte, welche die Determinante der Vandermonde-Matrix $V = (v_{ij})$ maximieren, wobei $v_{ij} = \psi_j(y_i)$ mit $y_i \in \mathbb{T}$, $i = 1, \dots, N_t$ ist.

Bei der Wahl der Dubiner-Basis (2.2) erhält man für die Fekete-Punkte eine gutkonditionierte Vandermonde-Matrix.

Es gibt eine eindeutige Bijektion $j \longleftrightarrow (m, n)$ zwischen der Zeilennummer j der Vandermonde-Matrix V und den Parametern m und n der Dubiner-Basis (2.2), sodass man $V = (v_{ij})$ wie folgt definieren kann [9]:

$$v_{ij} = \phi_j(x_i, y_i), \quad i, j = 1, \dots, N_t \quad (2.6)$$

und (x_i, y_i) Fekete-Punkte aus \mathbb{T} .

In der Praxis kann man verschiedene Lösungen finden, welche mit unterschiedlichen Normen ausgewertet sind und die Determinante der Vandermonde-Matrix maximieren. Daher weiß man nicht, ob die berechneten Lösungen ein globales Maximum angeben und dadurch die wahren Fekete-Punkte sind. Man kann die Fekete-Punkte also nur numerisch bestimmen und somit bezeichnen wir die lokalen Maxima als die approximierten Fekete-Punkte. In [10] wurden die Fekete-Punkte bis $N=19$ mit höherer Genauigkeit approximiert. Diese weisen eine kleinere Lebesgue-Konstante auf als die zuvor approximierten Fekete-Punkte.

2.2.1 Quadraturformel

Für die Quadraturformel in den Fekete-Punkten (x_i, y_i) über \mathbb{T} bzgl. der Dubiner-Basis benötigen wir die Quadratur-Gewichte w_i , die wie folgt zu bestimmen sind. Hierfür nutzen wir die Eigenschaft des Dubiner-Polynoms $\phi_{00} = 1$ aus :

$$\sum_{i=1}^{N_t} w_i \phi_{mn}(x_i, y_i) = \int_{\mathbb{T}} \phi_{mn} dx dy = \gamma_{00} \delta_{0m} \delta_{0n} = 2 \delta_{0m} \delta_{0n}, \quad 0 \leq m + n \leq N \quad (2.7)$$

$$\implies (w_1, \dots, w_{N_t}) \cdot V = 2 \cdot (1, 0, \dots, 0)$$

$$\implies w_i = 2 \cdot (V^{-1})_{1i}, \quad i = 1, \dots, N_t. \quad (2.8)$$

So können wir dann das Integral von u über \mathbb{T} wie folgt approximieren:

$$\int_{\mathbb{T}} u dx dy = \sum_{i=1}^{N_t} w_i u(x_i, y_i). \quad (2.9)$$

3 Singulär-gestörte Probleme

Singulär-gestörte Probleme sind Probleme mit einem kleinen Störungsparameter ε , welcher mit der höchsten Ableitung multipliziert ist. Sie bilden eine umfassenden Klasse von den Boundary-Layer-Problemen.

Die Analyse von singulär-gestörten Problemen geht zurück auf die frühen zwanziger Jahre. Die systematische Studie über numerische Methoden für singulär-gestörte Probleme begann aber erst in den 70er Jahren [11].

3.1 Singulär-gestörte Differentialgleichungen

Eine partielle Differentialgleichung (PDGL) wird als singulär-gestört bezeichnet, falls sich die PDGL bei Tendenz eines (oder mehrerer) Koeffizienten (Störungsparameter) gegen einen kritischen Wert signifikant unterschiedlich verhält. Die Lösung von singulär-gestörten Problemen haben, im Gegenteil zu regulären Problemen, Boundary-oder/und Interior-Layer. Diese sind Teilgebiete in denen sich die Ableitung der Lösung sehr stark ändert.

Die numerischen Methoden, die für reguläre Probleme entwickelt worden sind, führen bei singulär-gestörten Problemen zu einem Fehler in der approximierten Lösung, welcher von dem Wert des Störungsparameters abhängt. Der Fehler wird klein, wenn die effektive Gitter-Größe in Layerregionen viel kleiner ist als der Wert des Störungsparameters. Es gibt spezielle numerische Verfahren (ε -gleichmäßig konvergente Methode), bei denen der Fehler unabhängig vom Störungsparameter ist (siehe [23, 24, 25]). Bei solchen Verfahren hängt der Fehler nur vom Durchmesser des Gitters ab.

Ein Verfahren ist in L_∞ -Norm ε -gleichmäßig konvergent der Ordnung p , falls gilt

$$\|u - u_h\|_\infty \leq Ch^p \tag{3.1}$$

wobei C eine Konstante unabhängig vom Störungsparameter ε , u die exakte und u_h die approximierte Lösung ist. h bezeichnet den Durchmesser des zugehörigen Gitters.

Ein bekanntes Beispiel für singulär-gestörte Differentialgleichungen sind die Konvektions-Diffusions-Gleichungen.

3.2 Konvektions-Diffusions-Gleichungen

Bei den Konvektions-Diffusions-Gleichungen handelt es sich um Prozesse, bei denen Diffusion und Konvektion zusammen auftreten. Aus der physikalischen Sicht betrachtet kann man sich einen Fluss mit starker und gleichmäßiger Strömung vorstellen. Wenn eine flüssige Verschmutzung in den Fluss gelangt, so wirken zwei physikalische Prozesse mit: Die Verschmutzung beginnt sich langsam im Fluss auszubreiten (Diffusion), während sie wegen der starken Strömung des Flusses schnell in Richtung der Strömung transferiert wird (Konvektion). Hierbei ist die schnelle Strömung dominant.

Diese beiden Prozesse können zusammen mit Hilfe einer Konvektions-Diffusions-Gleichung modelliert werden:

$$\begin{aligned} -\varepsilon \Delta u + b \cdot \nabla u + cu &= f \quad \text{in } \Omega \in \mathbb{R}^n \quad \text{offen,} \\ u &= g \quad \text{auf } \partial\Omega \end{aligned} \tag{3.2}$$

dabei ist $-\varepsilon \Delta u$ der Diffusionsanteil, $b \cdot \nabla u$ der Konvektionsanteil und ε der Störungsparameter mit $0 < \varepsilon \ll 1$. Für kleine ε ist der Konvektionsterm der dominierende Anteil.

Im Folgenden betrachten wir ein Beispiel von einer linearen Konvektions-Diffusions-Gleichung, um zu sehen wie sich die Lösung verhält, wenn $\varepsilon \rightarrow 0$ geht.

Beispiel 3.1 *Die Randwertaufgabe*

$$-\varepsilon u'' + u' = 1 \quad \text{in } \Omega = (0, 1), \quad \text{mit } u(0) = u(1) = 0$$

hat die folgende exakte Lösung

$$u_\varepsilon(x) = x - \frac{e^{-\frac{1-x}{\varepsilon}} - e^{-\frac{1}{\varepsilon}}}{1 - e^{-\frac{1}{\varepsilon}}}.$$

Wir können leicht herausfinden, dass

$$\lim_{\varepsilon \rightarrow 0} u_\varepsilon(x) = x.$$

Ferner gilt für ein $t \in [0, 1)$:

$$\lim_{\varepsilon \rightarrow 0} \lim_{x \rightarrow t} u_\varepsilon(x) = t = \lim_{x \rightarrow t} \lim_{\varepsilon \rightarrow 0} u_\varepsilon(x).$$

Aber aufgrund der Randbedingung $u(1) = 0$ haben wir am rechten Rand eine starke Änderung der Kurve, wo die Ableitung von u_ε drastisch sinkt:

$$0 = \lim_{\varepsilon \rightarrow 0} \lim_{x \rightarrow 1} u_\varepsilon(x) \neq \lim_{x \rightarrow 1} \lim_{\varepsilon \rightarrow 0} u_\varepsilon(x) = 1.$$

Das heißt, es gibt einen *Boundary-Layer* (*Randschicht*) bei $x = 1$. Somit sind wir hier mit einer *singulär-gestörten Randwertaufgabe* konfrontiert. Die *Breite des Boundary-Layers* verhält sich wie $O(\varepsilon)$. Die *Umgebungen, in denen die Ableitung die Größe $O(\frac{1}{\varepsilon})$ erreicht, nennt man Boundary-Layer*.

Abbildung 3.1 veranschaulicht den *Boundary-Layer* für das Beispiel 3.1.

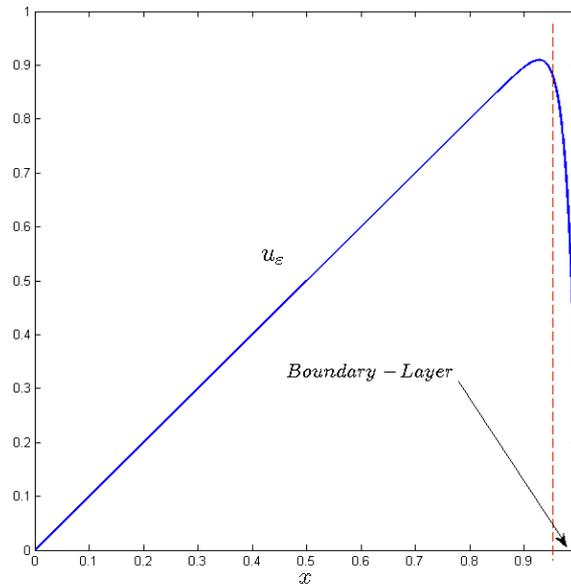


Abbildung 3.1: *Boundary-Layer*, Verhalten der exakten Lösung des Beispiels 3.1.

Singulär-gestörte PDGL sind numerisch schwer zu lösen. Das Problem des *Boundary-Layers* benötigt eine geeignete Auflösung in diesen Bereichen. Dazu kommen Instabilitäten bei der Diskretisierungen und schlechte Kondition der Matrizen bei der iterativen Lösung des zugehörigen Gleichungssystems. Stabilisierungen und Adaptive Methoden sind zur Verminderung von auftretenden Oszillationen und somit zum Gewinn einer akzeptablen Lösung nötig.

4 Diskretisierung

In diesem Kapitel stellen wir die geeignete Diskretisierung dar, welche später in den nächsten Kapiteln zur Lösung von verschiedenen Beispielen der Konvektions-Diffusions-Gleichung verwendet wird.

Die folgende allgemeine Form der Konvektions-Diffusions-Gleichung wird in den nächsten Kapitel behandelt.

$$\begin{aligned} -\varepsilon\Delta u + b \cdot \nabla u + cu &= f & \text{in } \Omega = (0, 1)^2 \\ u &= g & \text{auf } \partial\Omega \end{aligned} \tag{4.1}$$

dabei sind $b = (b_1, b_2)$, c und f im allg. Funktionen in x und y .

4.1 Triangulierung

Das Gebiet Ω wird zunächst äquidistant in k^2 Quadraten zerlegt. Danach werden die Quadrate jeweils in zwei Dreiecke geteilt, sodass wir am Ende $2k^2$ Dreieckselemente zur Verfügung haben.

Diese Art der Dreieckszerlegung ist als *Friedrichs-Keller-Triangulierung* bekannt.

Abbildung 4.1 zeigt uns die Zerlegung von Ω in 18 Dreieckselemente Ω_{ij} mit $i = 1, \dots, k = 3$ und $j = 1, \dots, 2k = 6$.

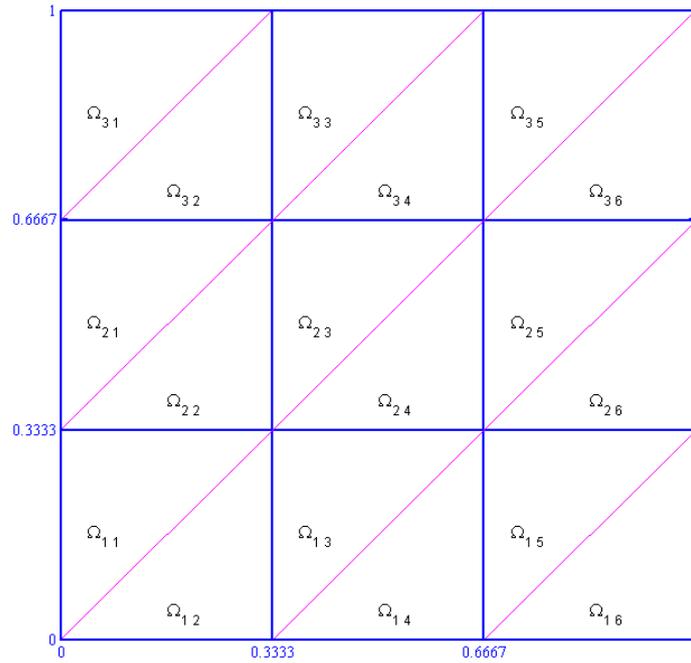


Abbildung 4.1: Dreieckszerlegung von $\Omega = (0, 1)^2$ für $k=3$, Nummerierung der Dreiecke mit zwei Indizes

Die Lösung u der Konvektions-Diffusions-Gleichung (4.1) wird in jedem Element in entsprechenden Kollokationspunkten approximiert. Für die Kollokationspunkte verwenden wir die Fekete-Punkte (x, y) , welche allerdings in jedes Dreieckselement Ω_{ij} zu transferieren sind. Dazu benutzen wir die folgende affine Transformation.

$$\begin{aligned}\hat{x} &= r_1 + r_2x + r_3y \\ \hat{y} &= s_1 + s_2x + s_3y\end{aligned}\quad (4.2)$$

Die Fekete-Punkte (x, y) für die Polynomgrade $N = 3, 6, 9$ und 12 entnehmen wir dem Anhang A.

Da die Eckpunkte stets auf die Eckpunkte abgebildet werden, erhalten wir für ein Dreieck mit gegebenen Eckpunkten (\hat{x}_1, \hat{y}_1) , (\hat{x}_2, \hat{y}_2) und (\hat{x}_3, \hat{y}_3) das folgende lineare Gleichungssystem:

$$\hat{X} = Ar, \quad \hat{Y} = As \quad (4.3)$$

$$\text{mit } \hat{X} = \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \end{pmatrix}, \hat{Y} = \begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \end{pmatrix}, r = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix} \text{ und } s = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix}.$$

Die Transformationsmatrix A sieht wie folgt aus:

$$A = \begin{pmatrix} 1 & -1 & -1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix}. \quad (4.4)$$

Zur Bestimmung der pseudo-spektralen Ableitungsmatrix benötigen wir die in Kapitel 2 definierte Vandermonde-matrix V . Nach der Berechnung der partiellen Ableitungen U_x und U_y der Dubiner-Basis (2.2) können wir die spektralen partiellen Ableitungen wie folgt bestimmen:

$$D_x = U_x V^{-1}, \quad D_y = U_y V^{-1}. \quad (4.5)$$

Die transformierten spektralen partiellen Ableitungsmatrizen ergeben sich wie folgt:

$$D_{\hat{x}} = \frac{1}{|J|} (s_3 D_x - s_2 D_y), \quad D_{\hat{y}} = \frac{1}{|J|} (-r_3 D_x + r_2 D_y) \quad (4.6)$$

mit $|J|$ die Jacobi-Determinante:

$$|J| = \begin{vmatrix} \frac{\partial \hat{x}}{\partial x} & \frac{\partial \hat{x}}{\partial y} \\ \frac{\partial \hat{y}}{\partial x} & \frac{\partial \hat{y}}{\partial y} \end{vmatrix} = r_2 s_3 - s_2 r_3. \quad (4.7)$$

Die Konvektions-Diffusions-Gleichung (4.1) teilen wir in zwei verschiedene äquivalente Formen auf: die Formulierung als System und die skalare Formulierung. Diese werden wir in den nächsten Abschnitten beschreiben.

4.2 Formulierung als System

Wir setzen $v_1 := u_x$ und $v_2 := u_y$ in der Konvektions-Diffusions-Gleichung (4.1) ein. Damit erhalten wir das folgende System erster Ordnung, welches äquivalent zu der Gleichung (4.1) ist.

$$\begin{aligned} -\varepsilon (v_{1,x} + v_{2,y}) + b_1 v_1 + b_2 v_2 + cu &= f && \text{in } \Omega \\ u_x - v_1 &= 0 && \text{in } \Omega \\ u_y - v_2 &= 0 && \text{in } \Omega \\ u &= g && \text{auf } \partial\Omega. \end{aligned} \quad (4.8)$$

Zusätzlich ist die Stetigkeit der Funktionen u und v als Interface-Bedingung an den Übergängen der Dreieckselemente punktweise gefordert.

Die Interface-Bedingungen in den Dreiecken sind auf der Sekante des Dreiecks, auf der waagerechten Kante des Dreiecks und auf der senkrechten Kante des Dreiecks zu betrachten.

Somit erhalten wir das folgende System erster Ordnung, welches die sämtlichen Interface- und Rand-Bedingungen für alle Dreieckselemente Ω_{ij} des Gebietes $\Omega = (0, 1)^2$ beherbergt.

Auf $\bar{\Omega}_{ij}$ mit $i = 1, \dots, k$ und $j = 1, \dots, 2k$:

$$-\varepsilon (v_{1,x}^{ij} + v_{2,y}^{ij}) + b_1^{ij} v_1^{ij} + b_2^{ij} v_2^{ij} + c^{ij} u^{ij} = f^{ij} \quad (4.9)$$

$$\begin{aligned} u_x^{ij} - v_1^{ij} &= 0 \\ u_y^{ij} - v_2^{ij} &= 0 \end{aligned} \quad (4.10)$$

Auf $\bar{\Omega}_{i,2j-1} \cap \bar{\Omega}_{i,2j}$ mit $i, j = 1, \dots, k$:

$$\begin{aligned} v_1^{i,2j} - v_1^{i,2j-1} &= 0 \\ v_2^{i,2j} - v_2^{i,2j-1} &= 0 \\ u^{i,2j} - u^{i,2j-1} &= 0 \end{aligned} \quad (4.11)$$

Auf $\bar{\Omega}_{i,2j-1} \cap \bar{\Omega}_{i+1,2j}$ mit $i = 1, \dots, k-1$ und $j = 1, \dots, 2k$:

$$\begin{aligned} v_1^{i+1,2j} - v_1^{i,2j-1} &= 0 \\ v_2^{i+1,2j} - v_2^{i,2j-1} &= 0 \\ u^{i+1,2j} - u^{i,2j-1} &= 0 \end{aligned} \quad (4.12)$$

Auf $\bar{\Omega}_{i,2j} \cap \bar{\Omega}_{i,2j+1}$ mit $i = 1, \dots, k$ und $j = 1, \dots, k-1$:

$$\begin{aligned} v_1^{i,2j+1} - v_1^{i,2j} &= 0 \\ v_2^{i,2j+1} - v_2^{i,2j} &= 0 \\ u^{i,2j+1} - u^{i,2j} &= 0 \end{aligned} \quad (4.13)$$

Mit den Randbedingungen auf $\partial\Omega_{ij} \cap \partial\Omega$, $i = 1, \dots, k$ und $j = 1, \dots, 2k$:

$$u^{ij} = g^{ij} \quad (4.14)$$

Die Gleichungen (4.11), (4.12) und (4.13) verwirklichen sämtliche Interface-Bedingungen.

Zur Vereinfachung der Arbeit ändern wir die Nummerierung der Elemente Ω_{ij} durch $l := (i - 1)2k + j$ in Ω_l um, wobei $l = 1, \dots, 2k^2$.

Abbildung 4.2 zeigt uns die Darstellung der Dreieckselemente Ω_l mit einfachem Index.

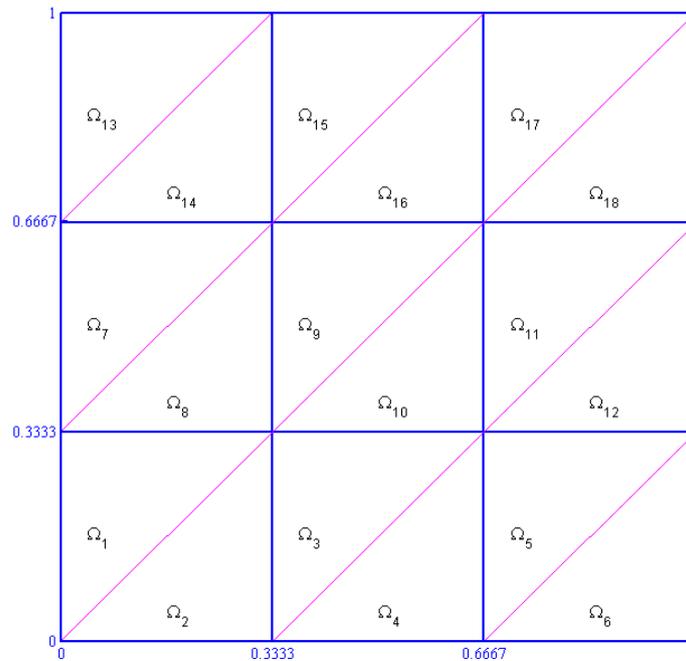


Abbildung 4.2: Dreieckszerlegung von $\Omega = (0, 1)^2$ für $k=3$ mit einfachem Index.

Wie es Abbildung 4.2 zu entnehmen ist, gibt es zwei Dreiecksformen: die mit ungeradem Index und die mit geradem Index. Für jede Form gibt es jeweils eine geeignete Nummerierungsart der Kollokationspunkte.

Die Nummerierungen, welche wir im Folgenden beschreiben, stehen mit der Reihenfolge der Fekete-Punkte (x, y) , die im Anhang A vorkommen, in Übereinstimmung.

Die Nummerierung der Kollokationspunkte für Dreieckselemente mit ungeradem Index läuft entlang der Sekante des Dreiecks, von unten nach oben, während diese für Dreieckselemente mit geradem Index lexikografisch von links nach rechts zu durchlaufen sind. Die Kollokationspunkte über die ganzen Elemente des Gebietes Ω haben eine fortlaufende Nummerierung.

In Abbildung 4.3 ist die Nummerierung der Punkte von der Ordnung $N=3$ für zwei Nachbarelemente dargestellt.

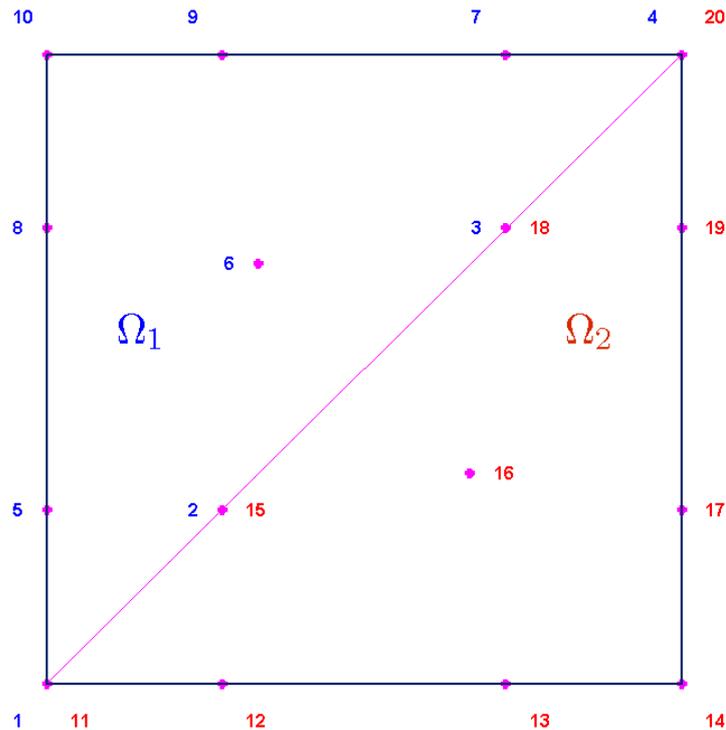


Abbildung 4.3: Nummerierung der Kollokationspunkte für $N = 3$.

In den folgenden Abschnitten beschreiben wir die Struktur der Interface- und Randoperatoren.

4.2.1 Die Randoperatoren

Zuerst nummerieren wir die Randpunkte elementweise in Richtung der Kollokationspunkte. Für Dreieckselemente mit ungeradem Index erfolgt dies entlang der Sekante, und für die mit geradem Index lexikografisch von links nach rechts.

Für Fekete-Punkte der Ordnung N wissen wir, dass es insgesamt $4kN$ Randpunkte auf Ω existieren, wobei $2k^2$ die Anzahl der gesamten Dreieckselemente des Gebietes ist.

Der Randoperator wird durch die Matrix $B = (b_{ij}) \in \mathbb{R}^{4kN \times 2k^2N_t}$ repräsentiert. Hierbei durchläuft der Index i die Stützpunkte auf $\partial\Omega$ entsprechend deren Nummerierungen, und der Index j bezeichnet die globale Nummer der Kollokationspunkte über das ganze Gebiet Ω .

Zu beachten ist, dass ein Randpunkt gleichzeitig zu mehreren Elementen gehören kann. Dadurch dass mehrfach auftretenden Randpunkte nur beim ersten Auftreten in der Randmatrix B berücksichtigt werden, spart man sich redundante Randbedingungen.

Die Einträge der Randmatrix B werden wie folgt definiert:

$$b_{ij} = \begin{cases} 1, & \text{wenn } (i, j) \text{ auf } \partial\Omega \\ 0, & \text{sonst} \end{cases}$$

In Abbildung 4.4 ist das Gebiet Ω in zwei Dreieckselemente zerlegt worden. Hierbei sehen wir wie die Randpunkte in Ω zu durchlaufen sind.

Die inneren Nummer (hier blau bzw. rot) sind die globalen Nummer der Kollokationspunkte, welche dem Index j entsprechen. Die Nummer am Rand (hier schwarz) beschreiben die Reihenfolge der vorkommenden Randpunkte und somit den Werteverlauf des Index i .

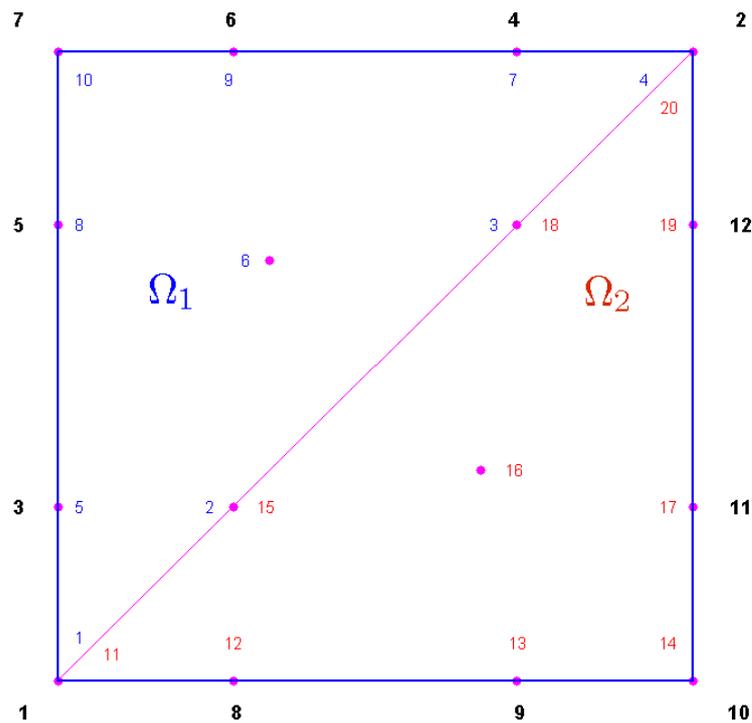


Abbildung 4.4: Nummerierung der Randpunkte von Ω mit zwei Dreieckselementen ($k = 1$), $N = 3$, $i = 1, 2, \dots, 4kN = 12$ und $j = 1, 2, \dots, 2k^2N_t = 20$.

Der Randmatrix B entsprechend der Abbildung 4.4 sieht wie folgt aus:

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Im Anhang B finden wir die Randmatrix B für $N = 2$ und 8 Dreieckselemente.

Zusätzlich definieren wir die Matrix $\text{Rand} \in \mathbb{R}^{4kN \times 2}$, welche die x und y-Koordinate aller Randpunkte beinhaltet. Die Reihenfolge, in der die Randpunkte in dieser Matrix vorkommen, entspricht genau unserer Nummerierungsart der Randpunkte.

$$\text{Rand} := \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_{11} & y_{11} \\ x_{12} & y_{12} \end{bmatrix} \quad (4.15)$$

4.2.2 Die Interface-Operatoren

Die Interface-Operatoren unterscheiden sich zwischen Elementen mit ungeradem und geradem Index. Wir bezeichnen die Interface-Operatoren auf der Sekante des Dreiecks mit $B_s^{(i)}$, auf der waagerechten Kante des Dreiecks mit $B_y^{(i)}$ und auf der senkrechten Kante mit $B_x^{(i)}$, $i = 1, 2$. Hier drückt der Index $i = 1$ die Zugehörigkeit des Interface-Operators zu einem Element mit ungeradem Index aus. Entsprechend ist $i = 2$ für Dreieckselemente mit geradem Index.

Es gibt jeweils $N_t = \frac{(N+1)(N+2)}{2}$ Kollokationspunkte auf jedem Dreieckselement. Davon liegen jeweils $N + 1$ Punkte auf jeder Kante.

Wir präsentieren zunächst $B_s^{(1)}$, $B_y^{(1)}$, $B_x^{(1)} \in \mathbb{R}^{(N+1) \times N_t}$ wie folgt für Elemente mit ungeradem Index. Dabei durchläuft der Index j stets die Kollokationspunkte in der entsprechenden lokalen Nummerierungsform:

$$b_{s, ij}^{(1)} = \begin{cases} 1, & \text{wenn } (i, j) \text{ auf der Sekante des Dreiecks} \\ 0, & \text{sonst} \end{cases}$$

wobei der erste Index $i = 1, \dots, N+1$ nur die Stützpunkten auf der Sekante durchläuft und zwar in der Richtung, wie diese lokal in dem Element nummeriert sind.

$$b_{y, ij}^{(1)} = \begin{cases} 1, & \text{wenn } (i, j) \text{ auf der waagerechten Kante des Dreiecks} \\ 0, & \text{sonst} \end{cases}$$

und

$$b_{x, ij}^{(1)} = \begin{cases} 1, & \text{wenn } (i, j) \text{ auf der senkrechten Kante des Dreiecks} \\ 0, & \text{sonst} \end{cases}$$

In Abbildung 4.5 sehen wir für $N = 3$, wie die Interface-Punkte auf der Sekante eines Dreiecks mit ungeradem Index zu durchlaufen sind.

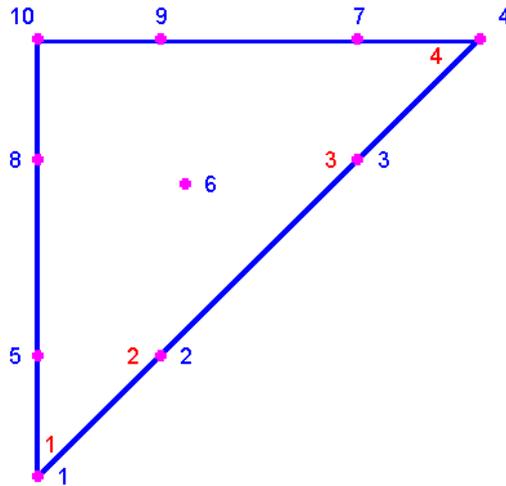


Abbildung 4.5: Nummerierte Interface-Punkte auf der Sekante eines Elementes mit ungeradem Index.

Der folgende Interface-Operator stimmt mit der Abbildung 4.5 überein:

$$B_s^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

sowie

$$\mathcal{D}_u := \begin{bmatrix} c^{11} & & & & \\ & c^{12} & & & \\ & & \ddots & & \\ & & & c^{23} & \\ & & & & c^{24} \end{bmatrix} \in \mathbb{R}^{2k^2 N_t \times 2k^2 N_t}$$

wobei für $n = 1, 2$

$$b_n^{ij} := \begin{bmatrix} b_n^{ij}(x_1, y_1) & & & \\ & \ddots & & \\ & & & b_n^{ij}(x_{N_t}, y_{N_t}) \end{bmatrix} \in \mathbb{R}^{N_t \times N_t}, \quad (4.16)$$

$$c^{ij} := \begin{bmatrix} c^{ij}(x_1, y_1) & & & \\ & \ddots & & \\ & & & c^{ij}(x_{N_t}, y_{N_t}) \end{bmatrix} \in \mathbb{R}^{N_t \times N_t} \quad (4.17)$$

mit $N_t = \frac{(N+1)(N+2)}{2}$.

Zur Bestimmung der ersten pseudo-spektralen Ableitungsmatrizen nutzen wir den Trick der negativen Summe (Negative sum trick) aus [27].

Nun können wir den Gesamt-Differentialoperator \mathcal{D} wie folgt einrichten:

$$\mathcal{D} := \begin{bmatrix} \mathcal{D}_{v_1} & \mathcal{D}_{v_2} & \mathcal{D}_u \\ \mathbf{E} & \mathbf{0} & -\mathcal{D}_x \\ \mathbf{0} & \mathbf{E} & -\mathcal{D}_y \end{bmatrix} \in \mathbb{R}^{6k^2 N_t \times 6k^2 N_t}$$

\mathbf{E} bezeichnet hier die Einheitsmatrix.

Die Struktur des Gesamt-Interfaceoperators \mathcal{I} ist die folgende:

$$\mathcal{I} := \begin{bmatrix} \mathbf{I} & & \\ & \mathbf{I} & \\ & & \mathbf{I} \end{bmatrix} \in \mathbb{R}^{3(k^2(N+1)+2k(k-1)(N+1)) \times 6k^2N_t} \quad \text{mit} \quad \mathbf{I} := \begin{bmatrix} \mathbf{I}_s \\ \mathbf{I}_y \\ \mathbf{I}_x \end{bmatrix}$$

wobei für $k = 2$ die Matrizen $\mathbf{I}_s, \mathbf{I}_y$ und \mathbf{I}_x die folgenden Darstellungen haben :

$$\mathbf{I}_s := \begin{bmatrix} -\mathbf{B}_s^{(1)} & \mathbf{B}_s^{(2)} & & & & & & & \\ & & -\mathbf{B}_s^{(1)} & \mathbf{B}_s^{(2)} & & & & & \\ & & & & -\mathbf{B}_s^{(1)} & \mathbf{B}_s^{(2)} & & & \\ & & & & & & -\mathbf{B}_s^{(1)} & \mathbf{B}_s^{(2)} & \\ & & & & & & & & -\mathbf{B}_s^{(1)} & \mathbf{B}_s^{(2)} \end{bmatrix}$$

$$\mathbf{I}_y := \begin{bmatrix} -\mathbf{B}_y^{(1)} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P} \cdot \mathbf{B}_y^{(2)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{B}_y^{(1)} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{P} \cdot \mathbf{B}_y^{(2)} \end{bmatrix}$$

$$\mathbf{I}_x := \begin{bmatrix} \mathbf{0} & -\mathbf{B}_x^{(1)} & \mathbf{B}_x^{(2)} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{B}_x^{(1)} & \mathbf{B}_x^{(2)} & \mathbf{0} \end{bmatrix}$$

wobei $\mathbf{P} \in \mathbb{R}^{(N+1) \times (N+1)}$ eine Permutationsmatrix der folgenden Form ist:

$$\mathbf{P} := \begin{bmatrix} & & & & & & & 1 \\ & & & & & & & & 1 \\ & & & & & & & & & 1 \\ & & & & & & & & & & 1 \\ & & & & & & & & & & & 1 \\ & & & & & & & & & & & & 1 \\ & & & & & & & & & & & & & 1 \\ & & & & & & & & & & & & & & 1 \\ & & & & & & & & & & & & & & & 1 \\ & & & & & & & & & & & & & & & & 1 \\ & & & & & & & & & & & & & & & & & 1 \\ & & & & & & & & & & & & & & & & & & 1 \\ & & & & & & & & & & & & & & & & & & & 1 \\ & 1 \\ & 1 \\ & 1 \end{bmatrix}.$$

Die Permutation der Matrix $\mathbf{B}_y^{(2)}$ ist notwendig, da die Interface-Punkte auf der waagerechten Kante eines Dreieckselementes mit geradem Index von links nach rechts nummeriert werden, während diese beim Nachbarelement mit ungeradem Index von rechts nach links nummeriert sind. Damit die Interface-Bedingungen auf der waagerechten Kante stimmen, permutieren wir die Matrix $\mathbf{B}_y^{(2)}$ mit Hilfe der Matrix \mathbf{P} .

Durch den Gesamt-Randoperator $\mathcal{B} := [\mathbf{0} \ \mathbf{0} \ \mathbf{B}]$ haben wir nun alle nötigen Mittel zum Aufbau des Super-Gleichungssystems $Ax = r$ für $k = 2$:

$$A := \begin{bmatrix} \mathcal{D} \\ \mathcal{I} \\ \mathcal{B} \end{bmatrix} \in \mathbb{R}^{(6k^2N_t + M + 4kN) \times 6k^2N_t}, \quad x := \begin{bmatrix} v_1^{11} \\ v_1^{12} \\ \vdots \\ v_1^{23} \\ v_1^{24} \\ \\ v_2^{11} \\ v_2^{12} \\ \vdots \\ v_2^{23} \\ v_2^{24} \\ \\ u_{11} \\ u_{12} \\ \vdots \\ u_{23} \\ u_{24} \end{bmatrix}, \quad r := \begin{bmatrix} f^{11} \\ f^{12} \\ \vdots \\ f^{23} \\ f^{24} \\ \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \\ g_1 \\ g_2 \\ \vdots \\ g_{4kN} \end{bmatrix} \quad (4.18)$$

mit $M := 3(k^2(N+1) + 2k(k-1)(N+1))$.

Wie erwartet, haben wir nun ein überbestimmtes Gleichungssystem, welches durch einen der geeigneten Löser, die wir in Kapitel 1 beschrieben haben, gelöst werden kann.

Unter den drei vorgestellten Lösern stellt sich die Methode der Normalgleichung als die schnellste Variante dar, während die QR-Zerlegung eine höhere Genauigkeit aufweist.

4.3 Die skalare Formulierung

In der folgenden skalaren Formulierung wird die zweite Ableitung zum Einsatz kommen. Wir suchen eine Lösung $u \in H^2(\Omega)$, welche die Konvektions-Diffusions-Gleichung (4.1) erfüllt. Aus diesem Grund sollte man unter den Interface-Bedingungen zusätzlich die Stetigkeit der Normalableitungen verlangen.

Äquivalent zu (4.1) haben wir das folgende System zu lösen:

Für $(x, y) \in \Omega_{ij}$, $i = 1, \dots, k$ und $j = 1, \dots, 2k$:

$$-\varepsilon (u_{xx}^{ij} + u_{yy}^{ij}) + b_1^{ij} u_x^{ij} + b_2^{ij} u_y^{ij} + c^{ij} u^{ij} = f^{ij} \quad (4.19)$$

Für $(x, y) \in \bar{\Omega}_{i,2j-1} \cap \bar{\Omega}_{i,2j}$ mit $i, j = 1, \dots, k$:

$$\begin{aligned} u^{i,2j} - u^{i,2j-1} &= 0 \\ \frac{\partial}{\partial x} u^{i,2j} - \frac{\partial}{\partial x} u^{i,2j-1} &= 0. \end{aligned} \quad (4.20)$$

Für $(x, y) \in \bar{\Omega}_{i,2j-1} \cap \bar{\Omega}_{i+1,2j}$ mit $i = 1, \dots, k-1$ und $j = 1, \dots, 2k$:

$$\begin{aligned} u^{i+1,2j} - u^{i,2j-1} &= 0 \\ \frac{\partial}{\partial y} u^{i+1,2j} - \frac{\partial}{\partial y} u^{i,2j-1} &= 0. \end{aligned} \quad (4.21)$$

Für $(x, y) \in \bar{\Omega}_{i,2j} \cap \bar{\Omega}_{i,2j+1}$ mit $i = 1, \dots, k$ und $j = 1, \dots, k-1$:

$$\begin{aligned} u^{i,2j+1} - u^{i,2j} &= 0 \\ \frac{\partial}{\partial x} u^{i,2j+1} - \frac{\partial}{\partial x} u^{i,2j} &= 0. \end{aligned} \quad (4.22)$$

Mit den Randbedingungen auf $\partial\Omega_{ij} \cap \partial\Omega$, $i = 1, \dots, k$ und $j = 1, \dots, 2k$:

$$u^{ij} = g^{ij}. \quad (4.23)$$

Die Gleichungen (4.20), (4.21) und (4.22) sind die entsprechenden Interface-Bedingungen.

Die Nummerierungsarten bleiben hier genauso wie in den vorherigen Abschnitten. Ebenso ändert sich der Randoperator hier nicht.

Zum Schluss können wir das Super-Gleichungssystem $Ax = r$ für $k = 2$ wie folgt aufbauen:

$$A := \begin{bmatrix} \mathcal{D} \\ \mathcal{I} \\ \mathcal{B} \end{bmatrix} \in \mathbb{R}^{2k^2 N_t + (2k^2 + 4k(k-1))(N+1) + 4kN \times 2k^2 N_t}, \quad x := \begin{bmatrix} u^{11} \\ u^{12} \\ \vdots \\ u^{23} \\ u^{24} \end{bmatrix}, \quad r := \begin{bmatrix} f^{11} \\ f^{12} \\ \vdots \\ f^{23} \\ f^{24} \\ 0 \\ \vdots \\ 0 \\ g_1 \\ g_2 \\ \vdots \\ g_{4kN} \end{bmatrix} \quad (4.25)$$

Auch hier haben wir ein überbestimmtes Gleichungssystem zu lösen.

5 Adaptive Verfeinerungsstrategien

Die approximierten Lösungen in verschiedenen Elementen des Gebietes Ω können sich unterschiedlich verhalten. In Elementen des Boundary-Layer-Bereiches können diese mit größerem Fehler behaftet sein als in den anderen Elementen.

Damit wir eine akzeptable Genauigkeit der approximierten Lösung erzielen, markieren wir Elemente mit Hilfe eines a-posteriori Fehlerschätzers zur Verfeinerung. Dies nennen wir *Markierungsstrategie*. Zudem benötigen wir Regeln zur Verfeinerung der markierten Elemente. Durch Anwendung dieser Regeln, genannt *Verfeinerungsregeln*, erhalten wir in jedem Verfeinerungsschritt eine zulässige Zerlegung.

5.1 Markierungsstrategie

In den folgenden Abschnitten befassen wir uns mit drei Kriterien zur adaptiven Verfeinerung, welche wir später als Markierungsstrategie verwenden.

5.1.1 Gradienten-Kriterium

Hierbei werden die Elemente zur Verfeinerung gekennzeichnet, welche einen hohen Gradienten besitzen. Die Elemente l , welche die folgende Ungleichung nicht erfüllen, sind in der Liste der zu verfeinernden Elemente.

$$\|\nabla u^{(l)}\| \leq \epsilon \|u_h\|_1 \quad (5.1)$$

wobei $\|\cdot\|$ die L_2 -Norm, $\|\cdot\|_1$ die H_1 -Norm und ϵ eine vorgegebene Toleranz bezeichnen. Mit u_h meinen wir hier die approximierte Lösung.

Unter den Markierungsstrategien ist dieses ein einfaches Kriterium, welches häufig benutzt wird, wenn es keine andere Alternative gibt.

5.1.2 Abfallende Entwicklungskoeffizienten

Das zweite verwendete Kriterium benutzt die Entwicklungskoeffizienten der Funktion u bzgl. einer gegebenen Polynom-Basis. In unserem Fall heißt dies, die Entwicklung von u bzgl. der Dubiner-Basis (2.2) über dem Standard-Dreieck \mathbb{T} (2.1) wie folgt zu betrachten:

$$u(x, y) = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} a_{mn} \phi_{mn}(x, y) \quad (5.2)$$

mit a_{mn} die Entwicklungskoeffizienten:

$$a_{mn} = \frac{1}{\langle \phi_{mn}, \phi_{mn} \rangle} \int_{\mathbb{T}} u \phi_{mn} dx dy. \quad (5.3)$$

Hier kommt die Quadraturformel (2.9) zum Einsatz.

Um a_{mn} in jedem Element l berechnen zu können, müssen wir entsprechend die Jacobi-Determinante $|J|$ (4.7) berücksichtigen:

$$a_{mn}^{(l)} = \frac{1}{\langle \phi_{mn}, \phi_{mn} \rangle} \int_{\mathbb{T}} u \phi_{mn} |J| dx dy. \quad (5.4)$$

Für die nächsten Schritte betrachten wir die Projektion von u auf \mathbb{P}_N , den Raum der Polynome vom Grad $\leq N$:

$$P_N(u) = \sum_{n=0}^N \sum_{m=0}^{N-n} a_{mn} \phi_{mn}. \quad (5.5)$$

Abbildung 5.1 zeigt uns die Belegungsstruktur der Matrix $\mathbf{A} = (a_{mn})_{0 \leq m+n \leq N}$ für $N = 6$.

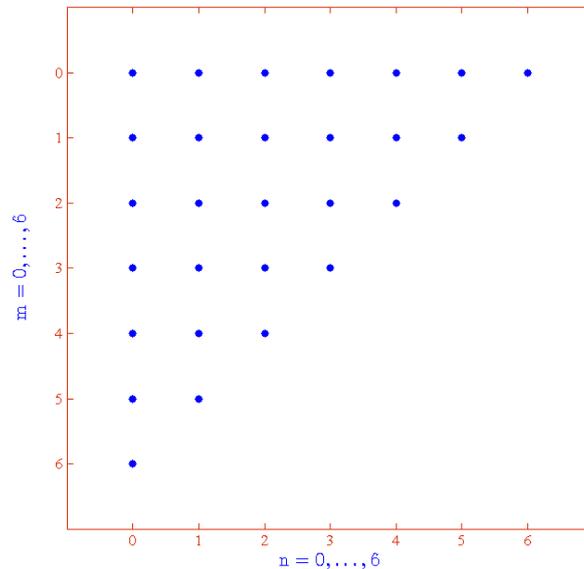


Abbildung 5.1: Die Belegungsstruktur der Koeffizienten a_{mn} für $N = 6$.

Zur Formulierung des zweiten Kriteriums nutzen wir das diskrete Spektrum der Entwicklungskoeffizienten und stellen dieses durch \bar{a}_ℓ wie folgt dar:

$$\bar{a}_\ell = \sum_{\substack{m=0 \\ m \leq \ell}}^{N-\ell} |a_{m\ell}| + \sum_{\substack{m=0 \\ m < \ell}}^{N-\ell} |a_{\ell m}|, \quad \ell = 0, \dots, N. \quad (5.6)$$

Nun approximieren wir \bar{a}_ℓ durch eine abfallende Funktion \tilde{a} , welche die beste Least-Squares-Anpassung zu $\bar{a}_{(N-3)}, \bar{a}_{(N-2)}, \bar{a}_{(N-1)}$ und \bar{a}_N ist:

$$\tilde{a}(\ell) = C \cdot \exp(-\alpha\ell) \quad (5.7)$$

wobei C und α Konstanten sind.

Damit wir diese Konstanten bestimmen können, wenden wir den Logarithmus auf \tilde{a} an und bestimmen die lineare Regression von $\log(\tilde{a}(\ell))$ in $\ell = (N-3), \dots, N$:

$$\log(\tilde{a}(\ell)) = \log(C) - \alpha \cdot \ell, \quad \ell = (N-3), \dots, N. \quad (5.8)$$

Nach der Bestimmung der linearen Regression bzw. der Konstanten C und α sind wir in der Lage, den Wert von \tilde{a} in $\ell = N+1$ zu extrapolieren.

Die zweite Markierungsstrategie lautet wie folgt:

$$\left(\tilde{a}(N)^2 + \int_{N+1}^{\infty} \tilde{a}(\ell)^2 d\ell \right)^{1/2} \leq \epsilon \|u_h\|. \quad (5.9)$$

Durch Einsetzen der Funktion \tilde{a} in das Integral der Ungleichung (5.9) können wir das zweite Kriterium vereinfachen:

$$\left(\tilde{a}(N)^2 + \frac{\tilde{a}(N+1)^2}{2\alpha} \right)^{1/2} \leq \epsilon \|u_h\|. \quad (5.10)$$

Aus statistischen Gründen sind die Ergebnisse der Regression für $N = 3, \dots, 5$ nicht vertrauenswürdig und vielmehr für $N \geq 6$ maßgebend. Die Zuverlässigkeit des Fehler-Schätzers nimmt mit zunehmendem Polynomgrad N zu.

In Abbildung 5.2 ist das exakte diskrete Spektrum \bar{a} für eine glatte Funktion $u = \exp(0.5x + y)$ bis zum Grad $N = 6$ zu sehen. Wir erkennen wie das Spektrum bei zunehmendem Grad abnimmt.

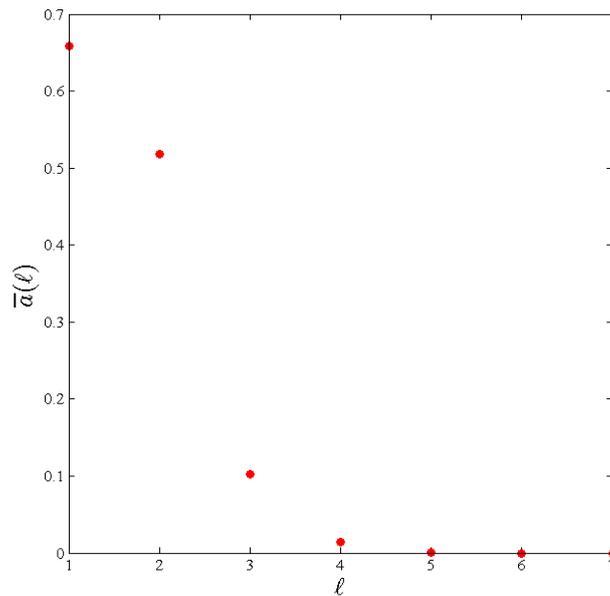


Abbildung 5.2: $\bar{\alpha}$ für die glatte Funktion $u = \exp(0.5x + y)$.

5.1.3 Spektrums-Ende

Das dritte Kriterium nimmt ebenso wie das zweite Kriterium direkten Bezug auf das Spektrum und zwar durch die Berücksichtigung der Entwicklungskoeffizienten der höchsten Ordnung. Somit müssen wir nur $\bar{a}_N = |a_{0N}| + |a_{N0}|$ bestimmen.

Es werden die Elemente zur Verfeinerung markiert, welche die folgende Ungleichung nicht erfüllen:

$$|a_{0N}| + |a_{N0}| \leq \epsilon \|u_h\| \quad (5.11)$$

Die Entscheidung über die Toleranz ϵ ist problemabhängig. Bei Wahl einer zu kleinen Toleranz werden zu viele Elemente zur Verfeinerung markiert. Das fordert mehr Speicherplatz und kann mehr Zeit in Anspruch nehmen. Hingegen könnte eine zu große Toleranz zwar die Verfeinerung weniger Elemente bewirken, jedoch ist eventuell eine größere Anzahl von Verfeinerungsschritten erforderlich.

Die adaptiven Verfeinerungen, welche aufgrund der Markierungsstrategien erfolgen, werden als *reguläre Verfeinerung* bezeichnet. Danach müssen wir für die Verfeinerung auf gewisse Regeln achten, welche zur Stabilisierung des Verfeinerungs-Algorithmus notwendig sind. Diese Regeln werden im nächsten Abschnitt präsentiert.

5.2 Verfeinerungsregel

Für die reguläre Verfeinerung unterteilen wir jedes markierte Dreieck in vier kleinere Dreiecke, indem wir die Mittelpunkte der Kanten verbinden. Dabei entstehen sogenannte *hängende Knoten*, welche die Eckpunkte der verfeinerten Dreiecke, aber keine Eckpunkte der nicht verfeinerten Nachbardreiecke sind. Die Zulässigkeit einer Triangulierung fordert die Eliminierung hängender Knoten. Die Verfeinerung dieser zusätzlichen Dreiecke wird manchmal *irreguläre Verfeinerung* benannt.

Durch reguläre Verfeinerung entstandene Dreiecke stempelt man als *rot* ab. Wenn ein Element drei hängende Knoten auf den Kanten hat, wird es ebenso als rotes Element bezeichnet und somit regulär in vier Teildreiecke aufgeteilt.

Ein *grünes Element* liegt vor, wenn es genau einen hängenden Knoten in dem Dreieckselement vorhanden ist. Solche hängenden Knoten werden als *halbierende Knoten* bezeichnet.

Bei zwei hängenden Knoten wird das Element als *blau* charakterisiert.

Die Elemente mit hängenden Knoten werden wie folgt behandelt:

- Dreiecke mit drei hängenden Knoten werden regulär verfeinert.
- Grüne Dreiecke verfeinern wir durch eine halbierende Kante in zwei Teildreiecke.
- Blaue Dreiecke werden durch Halbierung von genau zwei Kanten in drei Teildreiecke unterteilt.

Somit entsteht eine stabile zulässige Verfeinerung, welche rekursiv durchführbar ist. Aus der Abbildung 5.3 kann man sich einen Überblick schaffen, wie die hängenden Knoten zu behandeln sind.

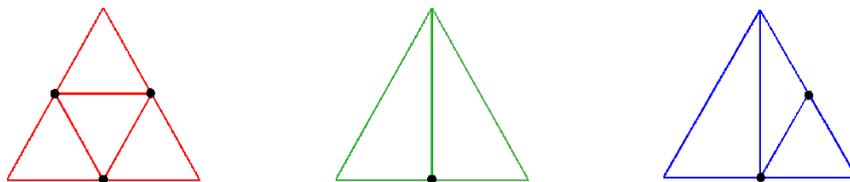


Abbildung 5.3: Verfeinerung von Dreiecken: (links) rotes, (Mitte) grünes und (rechts) blaues Element.

6 Numerische Ergebnisse

Wir haben in den vorherigen Kapiteln verschiedene Verfahren kennengelernt und sind nun in der Lage diese für die Konvektions-Diffusions-Gleichung (4.1) auszuführen.

Die Test-Beispiele sind der Diplomarbeit von Kippar [15] entnommen worden. Die adaptiven Verfeinerungskriterien testen wir nur für die Formulierung als System. Für das Gradienten-Kriterium führen wir das Programm solange aus, bis kein freier Speicherplatz mehr vorhanden ist.

Zur Lösung des Least-Squares-Problems verwenden wir den Matlab-Befehl **Backslash** "`\`". Das heißt, ein Gleichungssystem der Form $Ax = B$ wird durch $x = A \setminus B$ gelöst.

Der Backslash-Befehl enthält einen Algorithmus basierend auf der Form der Matrix A . Er untersucht zuerst, welche Struktur die Matrix A hat. Dieser Prozess verursacht natürlich einen Overhead. Wenn A keine quadratische Matrix ist, dann wird das QR-Verfahren verwendet. Für mehr Details siehe [16].

Zur Bestimmung des Approximationsfehlers verwenden wir die folgende gewichtete L_2 -Norm. Damit bekommen wir eine Art durchschnittlichen Approximationsfehler über alle Elemente des Gebietes.

$$\|x\|_{0,w} := \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2}, \quad x \in \mathbb{R}^d. \quad (6.1)$$

Die Konditionen werden bei allen Beispielen durch den Matlab-Befehl `cond` bestimmt.

6.1 Beispiel 1: moderater Boundary-Layer

Hierbei handelt es sich um die folgende Funktion:

$$u = \exp\left(-\frac{(x-1)(y-1)}{\varepsilon}\right), \quad (x, y) \in \Omega = (0, 1)^2 \quad (6.2)$$

welche die exakte Lösung einer Konvektions-Diffusions-Gleichung (4.1) mit

$$\mathbf{b} = (1, 1)^t, \quad c = 1, \quad \varepsilon = 0.1$$

und der entsprechenden rechten Seite

$$f(x, y) = -\frac{1}{\varepsilon}(x(x-1) - \varepsilon + y(y-1))u$$

ist.

In Abbildung 6.1 kann man beobachten, wie sich die Funktion u verhält, wenn man die Ränder $x = 1$ bzw. $y = 1$ nähert.

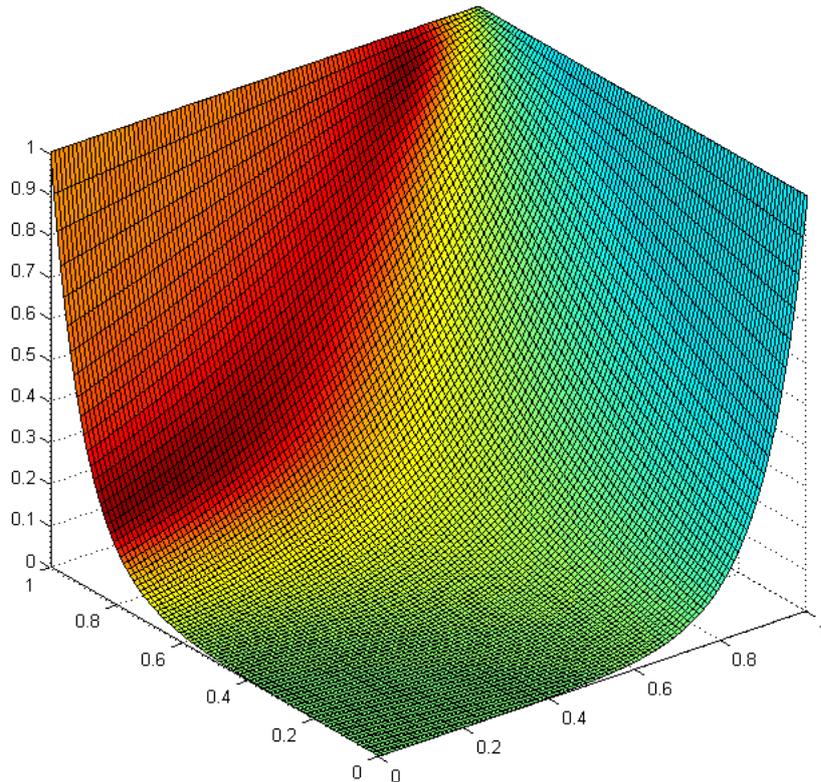


Abbildung 6.1: Plot der exakten Lösung von Beispiel 1.

6.1.1 Ergebnisse für die Formulierung als System

Hier wenden wir die im Unterkapitel 4.2 vorgestellte Methode an. Zu lösen ist das Gleichungssystem $Ax = r$, welches nach (4.18) aufgebaut ist. Die Anzahl der Kollokationspunkte in den einzelnen Dreieckselementen ist nach (2.5) zu bestimmen.

Tabelle 6.1: Konditionen der Matrix A, Formulierung als System, Beispiel 1.

# Elemente	N = 3	N = 6	N = 9	N = 12
8	5.8513e+02	5.7860e+03	3.4206e+04	8.2041e+04
18	1.0931e+03	1.2251e+04	7.3864e+04	1.7747e+05
32	1.7843e+03	2.0946e+04	1.2739e+05	3.0704e+05

Wie erwartet, sehen wir in Tabelle 6.1 große Konditionszahlen vor allem bei höherer Elementanzahl und größeren Polynomgraden N . Dadurch ist der Einfluss der kumulierten Rundungsfehler bei steigender Elementanzahl und ebenso wachsendem Polynomgrad nicht übersehbar.

In Tabelle 6.2 beobachten wir die Approximationsfehler entsprechend der Formulierung als System. Trotz der Singularität an beiden Rändern $x = 1$ und $y = 1$ erreichen wir hohe spektrale Genauigkeit.

Durch einen nur ungefähren Vergleich dieser Ergebnisse mit den Ergebnissen der Viereckszerlegung aus [15] beobachten wir hier eine leicht verschlechterte Approximationsgenauigkeit. Dafür sind wir hier aber aufgrund der hohen Anzahl von Kollokationspunkten mit deutlich höheren numerischen Kosten konfrontiert.

Tabelle 6.2: $\|u - u_h\|_{0,w}$, Formulierung als System, Beispiel 1.

# Elemente	N = 3	N = 6	N = 9	N = 12
8	6.4905e-02	8.8689e-04	5.9165e-06	1.1367e-08
18	4.6178e-02	9.3751e-05	1.8126e-07	1.6215e-10
32	3.0477e-02	1.5426e-05	1.5008e-08	5.5098e-12

Der Verlauf der Fehler ist in Abbildung 6.2 dargestellt.

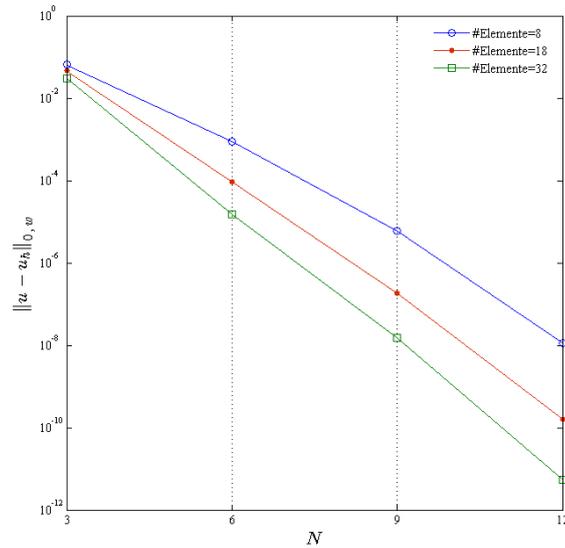


Abbildung 6.2: Approximationsfehler, $N=3, 6, 9, 12$, #Elemente= 8, 18, 32, Formulierung als System, Beispiel 1.

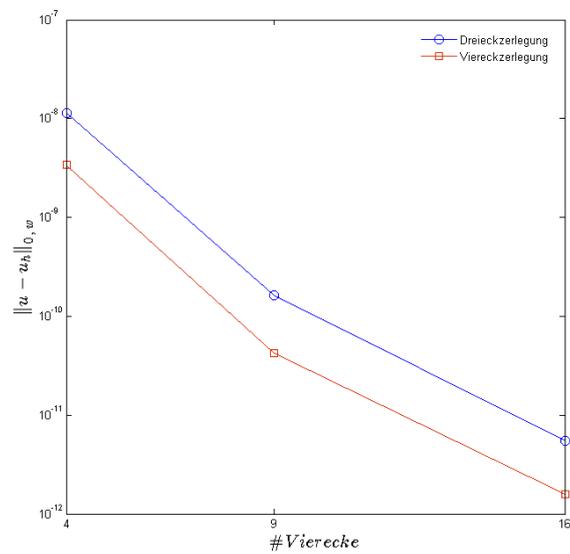


Abbildung 6.3: Vergleich des Fehlerverlaufs, $N=12$, Drei- und Viereckszerlegung, Formulierung als System, Beispiel 1.

Aus Abbildung 6.3 erhalten wir eine Vorstellung der Differenz der Fehler bei Drei- und Viereckszerlegung zumindest für $N = 12$.

6.1.2 Ergebnisse für die skalare Formulierung

Im Unterkapitel 4.3 haben wir Verfahren kennen gelernt, welche einen direkten Bezug auf die zweiten Ableitungsmatrizen nehmen. Der Aufbau des Gleichungssystems $Ax = r$ wurde in (4.25) präsentiert.

Tabelle 6.3 zeigt die Konditionszahlen der skalaren Formulierung für Beispiel 1.

Tabelle 6.3: Konditionen der Matrix A, skalare Formulierung, Beispiel 1.

# Elemente	N = 3	N = 6	N = 9	N = 12
8	2.2062e+02	2.9785e+03	2.1272e+04	5.6230e+04
18	5.5905e+02	9.1271e+03	6.7129e+04	1.7818e+05
32	1.1615e+03	2.0548e+04	1.5433e+05	4.1005e+05
50	2.1226e+03	3.8901e+04	2.9618e+05	7.8706e+05
72	3.5415e+03	6.5861e+04	5.0587e+05	1.3441e+06

Tabelle 6.4 enthält die Approximationsfehler für die skalare Formulierung. Eine exponentielle Konvergenz kann hier wie bei der Formulierung als System festgestellt werden. Die Anwendung der zweiten pseudo-spektralen Ableitungsmatrizen hat zur Folge, dass die Ergebnisse hier etwas schlechter ausfallen als bei der Formulierung als System. Der gleiche Effekt ist auch bei der Viereckszerlegung in [15] zu sehen.

Tabelle 6.4: $\|u - u_h\|_{0,w}$, skalare Formulierung, Beispiel 1.

# Elemente	N = 3	N = 6	N = 9	N = 12
8	8.7814e-02	3.8526e-03	2.0283e-05	1.5029e-07
18	6.8785e-02	6.4885e-04	8.4008e-07	1.7760e-09
32	5.7594e-02	1.5284e-04	9.7739e-08	7.5428e-11
50	4.8936e-02	4.9581e-05	1.6467e-08	7.4289e-12
72	4.1725e-02	2.0466e-05	3.7140e-09	4.2681e-12

In Abbildung 6.4 ist der Fehlerverlauf schematisch dargestellt. Ab 50 Elementen impliziert eine gleichmäßige Erhöhung der Elementanzahl nur eine mäßige Verbesserung der Approximationsfehler. Für $N = 12$ gibt es sogar nach 72 Elementen eine Tendenz zur Verschlechterung der Approximationsgenauigkeit.

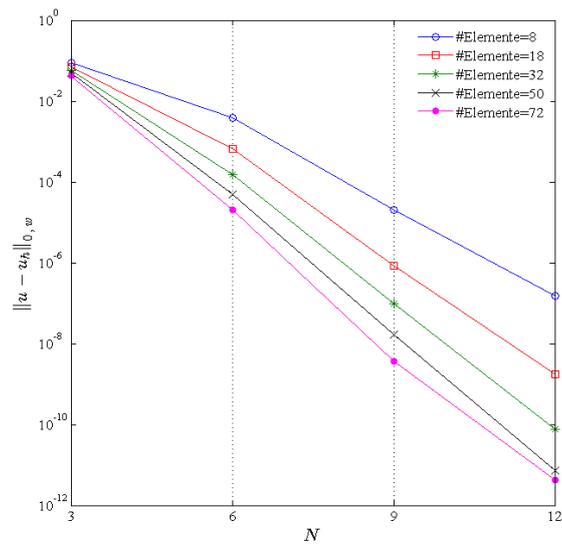


Abbildung 6.4: Approximationsfehler, $N=3, 6, 9, 12$, $\#\text{Elemente}=8, 18, 32, 50, 72$, skalare Formulierung, Beispiel 1.

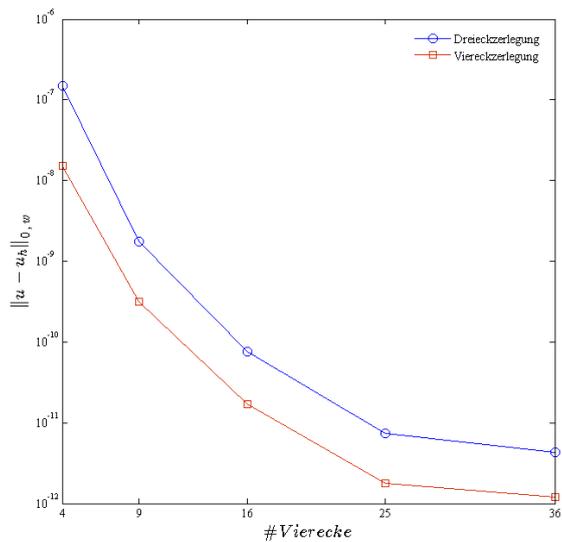


Abbildung 6.5: Vergleich des Fehlerverlaufs, $N=12$, Drei- und Viereckszerlegung, skalare Formulierung, Beispiel 1.

Abbildung 6.5 zeigt, wie sich die Fehler bei Drei- und Viereckszerlegung für $N = 12$ im Vergleich zueinander verhalten. Die Konvergenzgeschwindigkeit bezüglich der Anzahl der Vierecke bleibt bei beiden Zerlegungsarten ähnlich.

6.1.3 Skalare vs. System

Durch die Abbildungen 6.6 und 6.7 können wir den Fehlerverlauf und die Konditionsgrößen der beiden Formulierungsmethoden für eine Elementanzahl von 32 miteinander vergleichen. Die Formulierung als System liefert zwar bessere Approximationsgenauigkeiten, die Unterschiede sind aber gering.

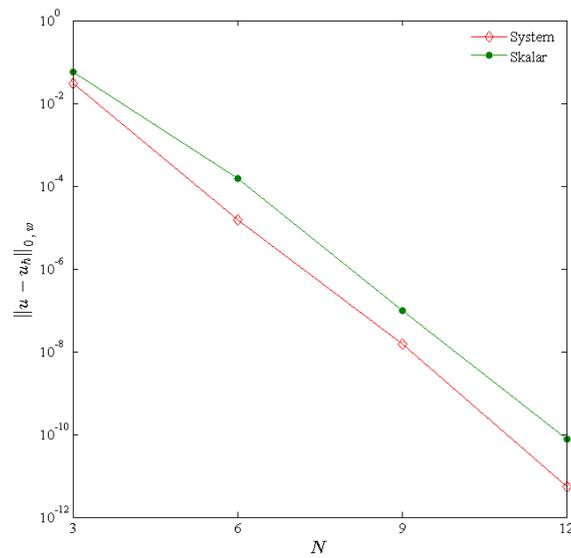


Abbildung 6.6: Approximationsfehler, Skalar vs. System für 32 Elemente, Beispiel 1.

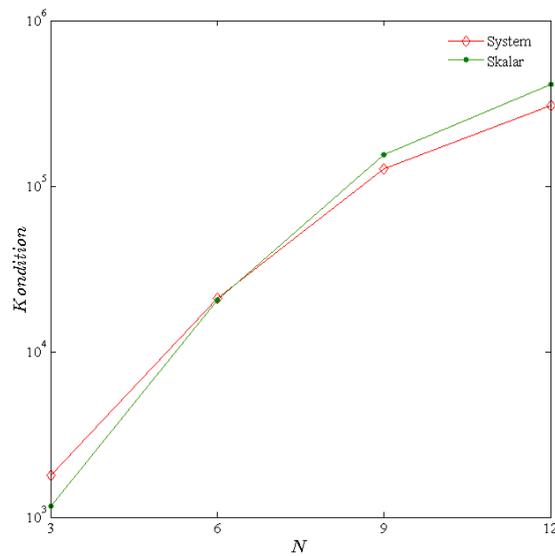


Abbildung 6.7: Konditionen der Matrix A , Skalar vs. System für 32 Elemente, Beispiel 1.

Die Abbildungen 6.8 und 6.9 verschaffen uns ein Bild über die Struktur der Matrix A für $N = 6$ und eine Elementanzahl von 32 jeweils für die skalare und die Systemformulierung.

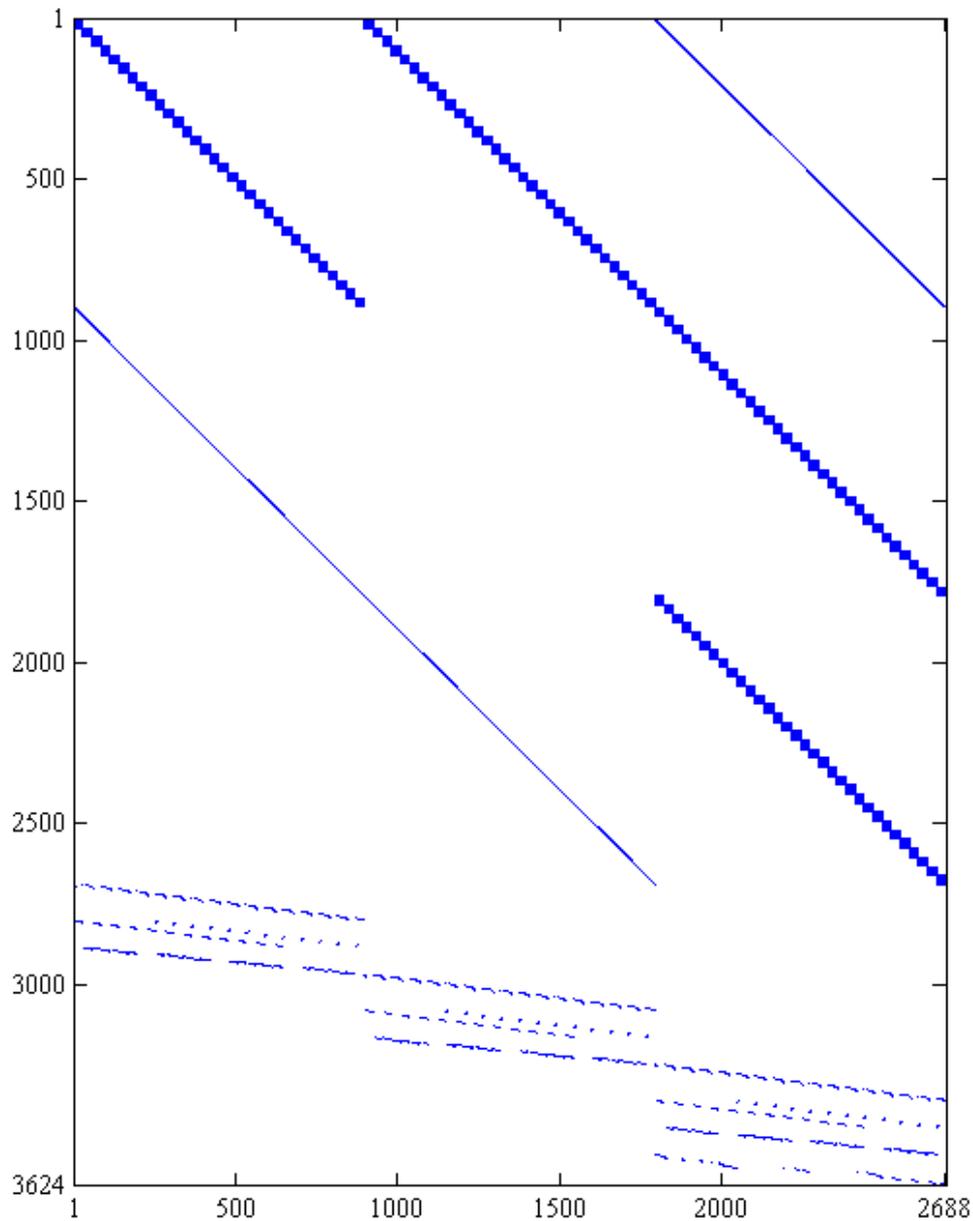


Abbildung 6.8: Belegungsstruktur der Matrix A , $N = 6$, #Elemente= 32, Formulierung als System, Beispiel 1, Anzahl der von 0 verschiedenen Einträge = 104016, A eine 3624×2688 -Matrix.

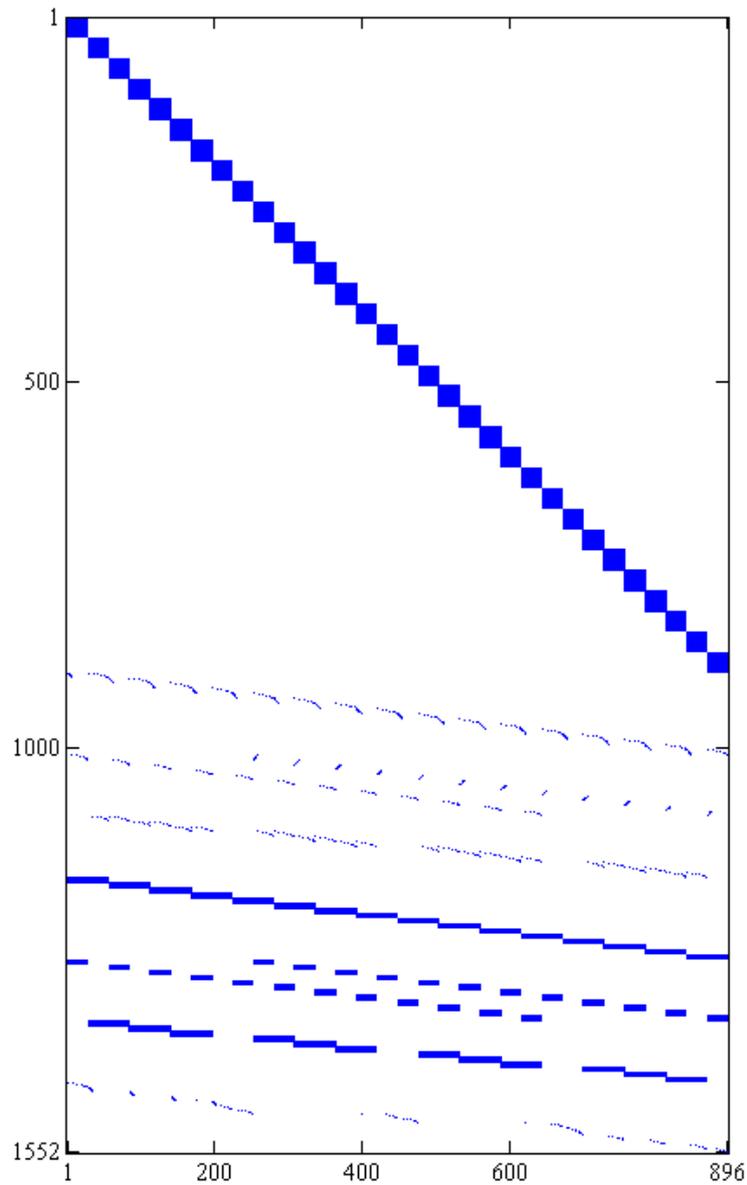


Abbildung 6.9: Belegungsstruktur der Matrix A für $N = 6$, $\#Elemente = 32$, skalare Formulierung, Beispiel 1, Anzahl der von 0 verschiedenen Einträge = 41336, A eine 1552×896 -Matrix.

In Abbildung 6.8 sind etwa 1.07% der Matrix A besetzt, während diese in Abbildung 6.9, also bei der skalaren Formulierung um etwa 2% mehr besetzt sind, nämlich etwa 2.97% von A .

6.1.4 Adaptive Verfeinerung

Wie bereits erwähnt, wenden wir die adaptiven Verfeinerungen nur für die Formulierung als System an. Dazu benutzen wir die drei Kriterien, die wir in Kapitel 5 präsentiert haben. Das Anfangsgitter umfasst jeweils 32 Elemente.

6.1.4.1 Gradienten-Kriterium

Wir nehmen $\epsilon = 0.1$ an und verfeinern die Elemente, welche das Kriterium (5.1) nicht erfüllen.

In Abbildung 6.10 ist gut ersichtlich, dass hauptsächlich die Elemente verfeinert wurden, welche im Bereich des Boundary-Layers liegen, in dem die Steigung drastisch zunimmt. Abbildung 6.11 gibt uns einen Überblick über die Konturlinien der exakten Lösung.

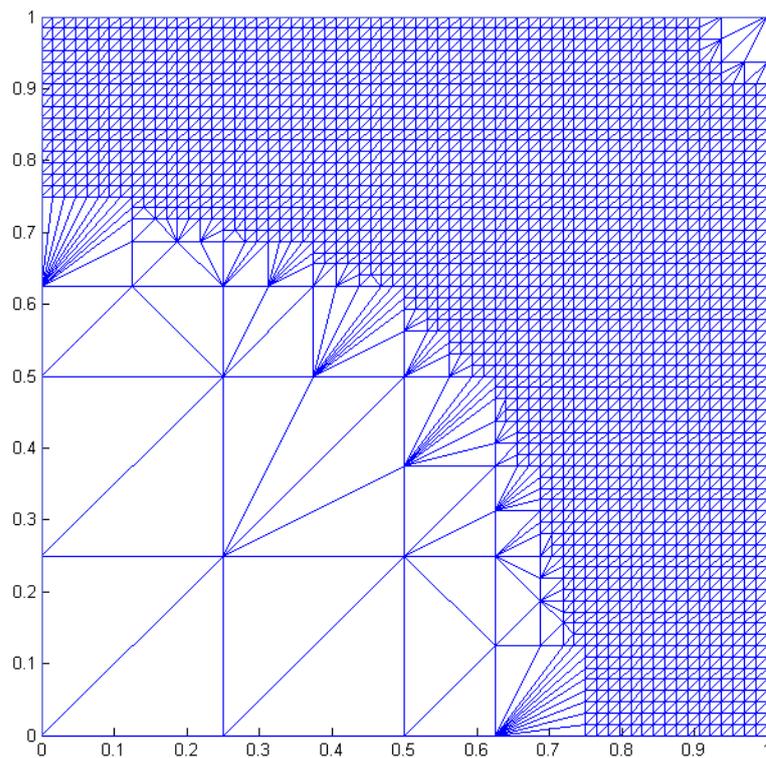


Abbildung 6.10: Adaptive Verfeinerung, Gradienten-Kriterium, #Elemente=4556, $N = 3$, Beispiel 1.

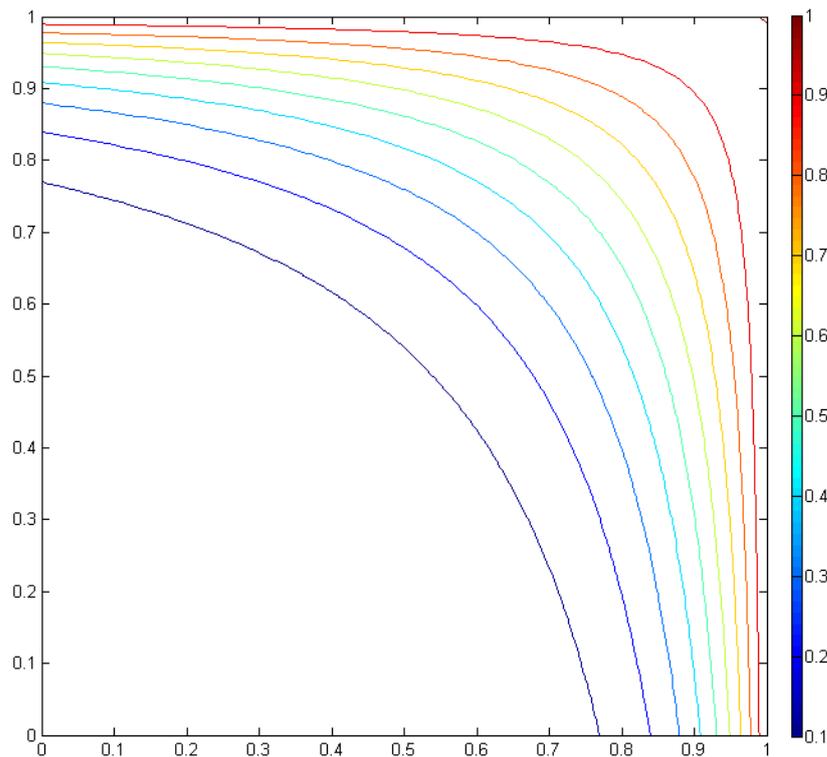
Abbildung 6.11: Konturlinien der exakten Lösung u , Beispiel 1.

Tabelle 6.5 enthält die Ergebnisse der adaptiven Verfeinerungen. Je höher der Polynomgrad ist, desto weniger Verfeinerungsbedarf ist vorhanden.

Tabelle 6.5: Numerische Ergebnisse nach dem Gradienten-Kriterium, Beispiel 1.

N	$\ u - u_h\ _{0,w}$	#Elemente	#Verfeinerung
3	1.3534e-04	4556	4
6	1.6986e-07	1214	3
9	1.2567e-10	340	2
12	2.7952e-13	340	2

Abbildung 6.12 veranschaulicht den Verfeinerungseffekt abhängig vom Polynomgrad. Hier wird der Verlauf der Fehler vor und nach der Verfeinerung gegenübergestellt. Entsprechend der Anzahl der Verfeinerungen beobachten wir bei $N = 3$ und 6 im Vergleich zu den anderen Polynomgraden einen größeren Verbesserungseffekt durch die Verfeinerung. Jedoch sind wir bei $N=3$ mit deutlich höherem Freiheitsgrad konfrontiert.

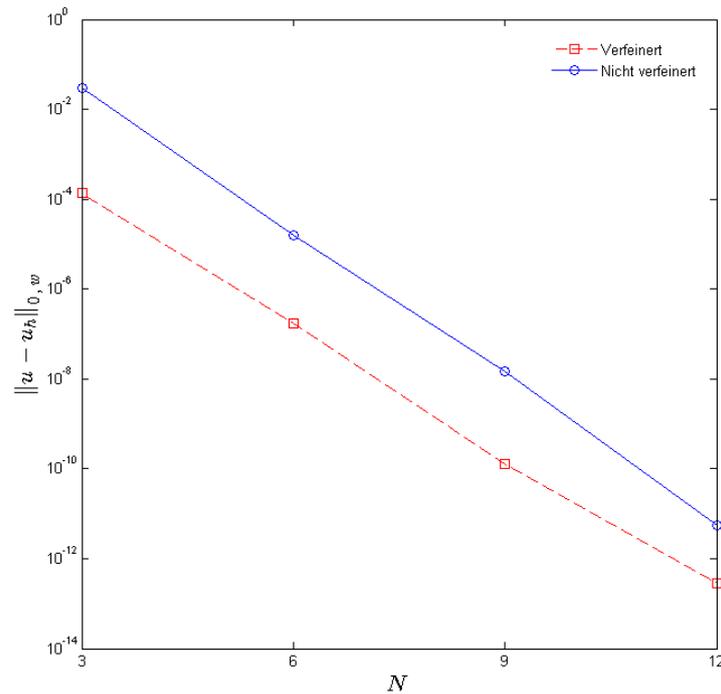


Abbildung 6.12: Vergleich der Approximationsfehler, Beispiel 1, vorher vs. nachher.

6.1.4.2 Abfallende Entwicklungskoeffizienten

Das zweite Kriterium können wir, wie in Abschnitt 5.1.2 angesprochen wurde, nur für $N = 6, 9$ und 12 testen. Wir nehmen $\epsilon = 4.65e - 6$ an.

Tabelle 6.6: Numerische Ergebnisse nach dem zweiten Kriterium, Beispiel 1.

N	$\ u - u_h\ _{0,w}$	#Elemente	#Verfeinerung
6	8.9057e-07	120	3
9	1.3820e-08	38	1
12	1.5687e-12	55	1

Wie man Tabelle 6.6 und Abbildung 6.13 entnehmen kann, werden für $N = 12$ mehr Verfeinerungen benötigt als für $N = 9$. Für den Polynomgrad $N=6$ erzielen wir eine größere Verbesserung durch die adaptive Verfeinerung. Abbildung 6.13 gibt einen Überblick über die sich durch die Verfeinerung ergebende Verbesserung.

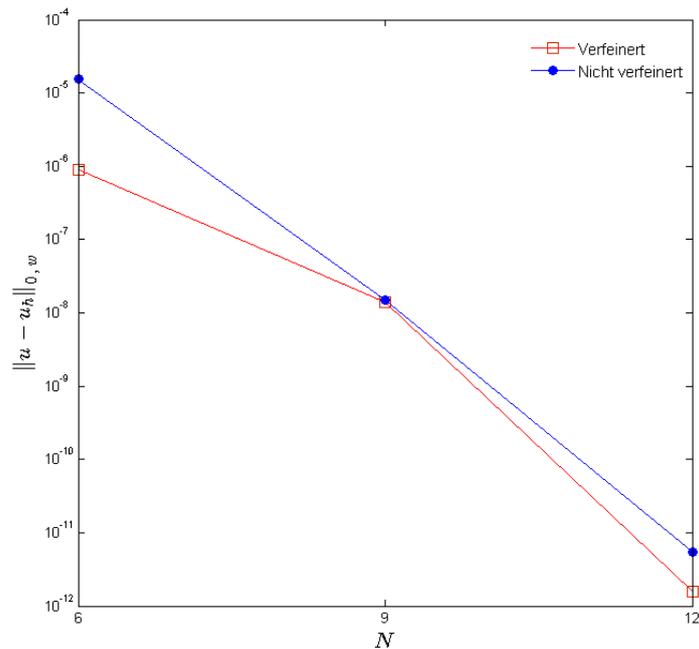


Abbildung 6.13: Vergleich der Approximationsfehler, Kriterium 2, Beispiel 1, vorher vs. nachher.

6.1.4.3 Spektrums-Ende

Hier entscheiden wir uns für die Toleranz $\epsilon = 0.00005$. Das Kriterium 5.1.3 angewandt auf Beispiel 1 führt uns zu den Ergebnissen in Tabelle 6.7.

Tabelle 6.7: Numerische Ergebnisse nach dem dritten Kriterium, Beispiel 1.

N	$\ u - u_h\ _{0,w}$	#Elemente	#Verfeinerung
3	4.8783e-04	341	2
6	1.4274e-06	93	1
9	9.4332e-10	87	1
12	4.1195e-12	38	1

Bei den Polynomgraden $N = 9$ und 12 sind die Verbesserungen nur dürftig. Hierbei haben wir beim Polynomgrad $N=9$ den größten Freiheitsgrad.

Wenn wir die Verhältnisse zwischen dem Speicherplatzbedarf und der Verbesserung für alle vier Polynomgrade vergleichen, können wir zu dem Schluss kommen, dass sich die Verfeinerung nach dem dritten Kriterium, zumindest für Beispiel 1, für $N = 3$ und 6 effektiver erweist, als für höhere Polynomgrade.

Abbildung 6.14 stellt die Ergebnisse vor und nach der Verfeinerung grafisch dar.

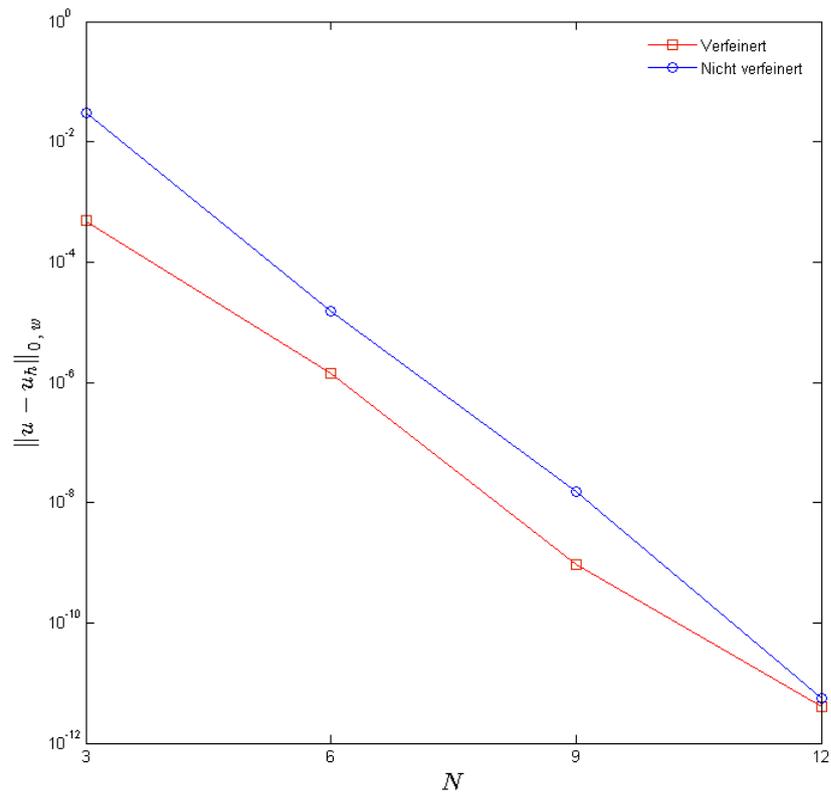


Abbildung 6.14: Vergleich der Approximationsfehler, Kriterium 3, Beispiel 1, vorher vs. nachher.

6.2 Beispiel 2: Verschärfter Boundary-Layer

In Beispiel 2 verwenden wir dieselbe Funktion u wie in Beispiel 1 mit den entsprechenden Angaben für die Konvektions-Diffusions-Gleichung (4.1). Allerdings nehmen wir $\varepsilon = 10^{-3}$, welches eine verstärkte singuläre Störung hervorruft. Dabei handelt es sich um eine Funktion mit immensom Anstieg sehr nahe an den Rändern $x = 1$ bzw. $y = 1$, also im Bereich des Boundary-Layers. Die Singularität ist hierbei deutlich erkennbar.

In Abbildung 6.15 können wir den Verlauf der Funktion u beobachten. u ist fast überall gleich 0 und nur sehr nahe an den Rändern $x = 1$ und $y = 1$ steigt sie drastisch an. Vor allem in solchen Bereichen ist eine höhere Auflösung des Gebietes notwendig, damit man eine insgesamt bessere Approximation erzielen kann.

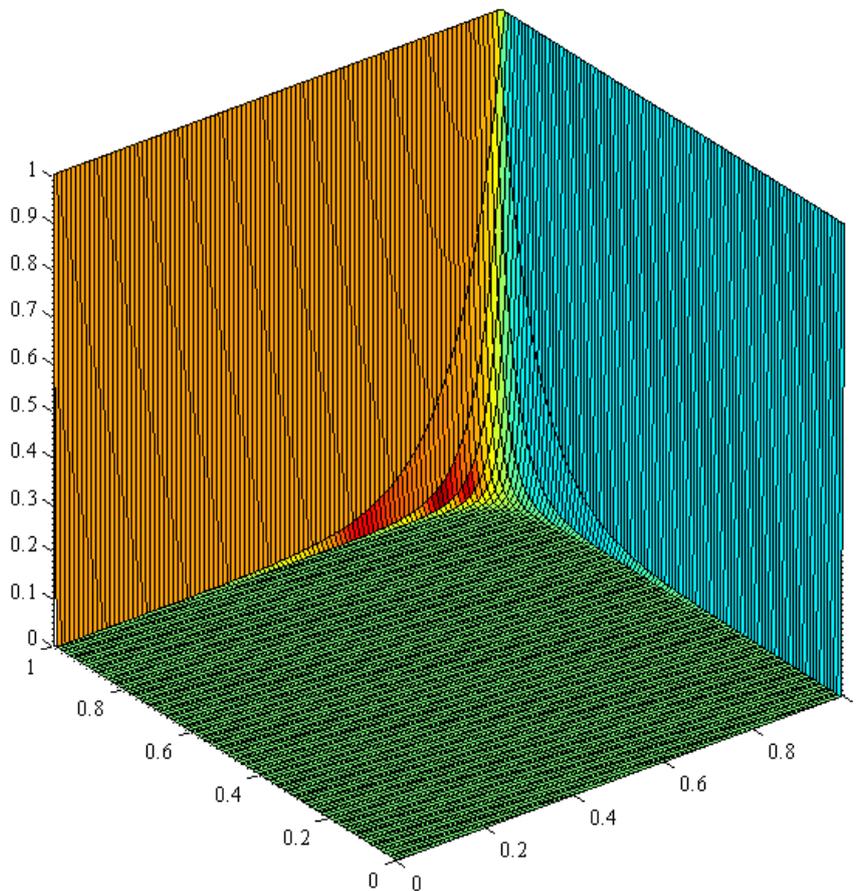


Abbildung 6.15: Plot der exakten Lösung von Beispiel 2.

6.2.1 Ergebnisse für die Formulierung als System

In Tabelle 6.8 sehen wir die Konditionszahlen, welche im Vergleich zum Beispiel 1 und vor allem bei einer Elementanzahl von 8 um eine Zehnerpotenz größer sind.

Tabelle 6.8: Konditionen der Matrix A, Formulierung als System, Beispiel 2.

# Elemente	N = 3	N = 6	N = 9	N = 12
8	1.6413e+03	1.1293e+04	6.8989e+04	1.7077e+05
18	3.2817e+03	2.4286e+04	1.5063e+05	3.7361e+05
32	5.9812e+03	4.2115e+04	2.6036e+05	6.4592e+05

Tabelle 6.9 enthält die Werte der Approximationsfehler für Beispiel 2. Die im Vergleich zu Beispiel 1 verstärkte Singularität macht sich in den Werten der Approximationsfehler deutlich bemerkbar. Somit ist hier der Bedarf an einer besseren Auflösung im Bereich des Boundary-Layers entsprechend größer.

Unsere Ergebnisse weisen im Vergleich zu den Werten von Kippar [15] eine leichte Verbesserung auf. Hierbei ist jedoch der erhöhte Aufwand durch die Dreieckszerlegung nicht außer Acht zu lassen.

Tabelle 6.9: $\|u - u_h\|_{0,w}$, Formulierung als System, Beispiel 2.

# Elemente	N = 3	N = 6	N = 9	N = 12
8	6.6940e+00	1.2266e+00	4.5622e-01	1.7518e-01
18	3.9901e+00	6.7050e-01	2.0635e-01	1.0615e-01
32	2.5764e+00	4.4153e-01	1.1713e-01	1.0135e-01

In Abbildung 6.16 betrachten wir den Fehlerverlauf für 8, 18 und 32 Elemente. Für $N = 12$ ist ab einer Elementanzahl von 32 die Verbesserung des Approximationsfehlers nur dürftig. Hierbei sinkt auch die Konvergenzrate bei einem Sprung von $N=9$ auf $N=12$ ab.

Wir versuchen einen ungefähren Vergleich zwischen unseren Ergebnissen und den Ergebnissen der Viereckszerlegung von Kippar [15] durchzuführen. Daher zeigen wir in Abbildung 6.17 wie der Fehlerverlauf für $N = 12$ bei Drei- und Viereckszerlegung aussieht.

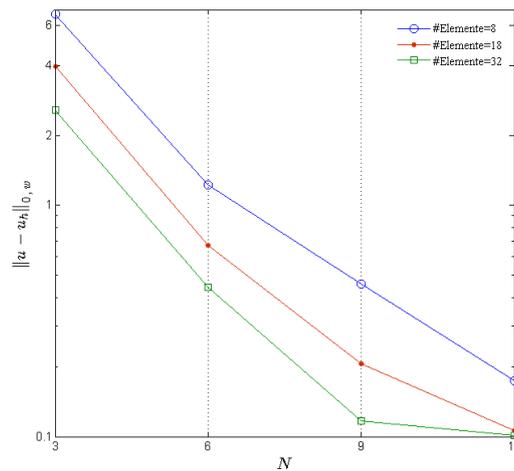


Abbildung 6.16: Approximationsfehler, $N=3, 6, 9, 12$, $\#\text{Elemente}= 8, 18, 32$, Formulierung als System, Beispiel 2.

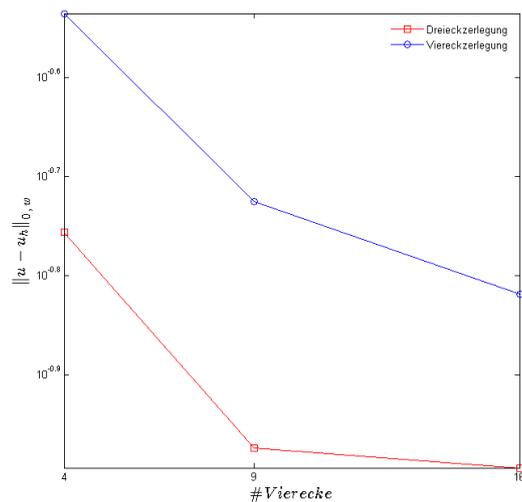


Abbildung 6.17: Vergleich des Fehlerverlaufs, $N=12$, für Drei- und Viereckszerlegung, Formulierung als System, Beispiel 2.

In Abbildung 6.17 ist die Verbesserung der Ergebnisse durch die Dreieckszerlegung deutlich zu erkennen. Wir beobachten auch bei der Viereckszerlegung, dass sich der Fehler ab 9 Vierecken, das entspricht 18 Dreiecken, deutlich langsamer verbessert. Im Vergleich zu Beispiel 1 merken wir, dass die Dreieckszerlegung bei verstärkter Singularität bessere Ergebnisse liefern kann als die Viereckszerlegung. Das liegt daran, dass wir bei der Dreieckszerlegung eine etwas höhere Auflösung des Gebietes erzielen können.

6.2.2 Ergebnisse für die skalare Formulierung

In Tabelle 6.10 sind die Konditionszahlen bezüglich der skalaren Formulierung für Beispiel 2 angegeben.

Hier erhalten wir im Vergleich zu der skalaren Formulierung von Beispiel 1 eine kleinere Konditionsgröße, welche an der höheren Singularität des Beispiels 2 liegt.

Tabelle 6.10: Konditionen der Matrix A, skalare Formulierung, Beispiel 2.

# Elemente	N = 3	N = 6	N = 9	N = 12
8	1.7270e+02	7.3931e+02	2.2886e+03	3.7951e+03
18	3.6543e+02	1.5565e+03	4.8912e+03	8.2400e+03
32	6.2625e+02	2.6836e+03	8.4882e+03	1.4462e+04
50	9.5345e+02	4.1172e+03	1.3129e+04	2.2617e+04
72	1.3476e+03	5.8697e+03	1.8875e+04	3.2906e+04

Die Approximationsfehler sind in Tabelle 6.11 zu sehen. Die Approximationsgenauigkeit verschlechtert sich im Vergleich zu Beispiel 1 drastisch. Der Einfluss einer höheren Singularität ist hier auffallend.

Tabelle 6.11: $\|u - u_h\|_{0,w}$, skalare Formulierung, Beispiel 2.

# Elemente	N = 3	N = 6	N = 9	N = 12
8	9.3863e+00	1.4015e+00	6.2126e-01	2.6883e-01
18	5.2620e+00	7.9153e-01	2.5647e-01	1.5801e-01
32	3.6574e+00	5.8308e-01	1.6130e-01	1.2851e-01
50	2.8032e+00	4.4127e-01	1.1899e-01	1.1559e-01
72	2.3134e+00	3.2683e-01	9.9334e-02	1.0728e-01

Die Auswirkung einer gleichmäßigen Verfeinerung des Gitters ist in Abbildung 6.18 zu sehen. Bei $N = 9$ und 12 sind die Verbesserungseffekte ab 32 Elementen nur moderat. Ab einer Elementanzahl von 50 Dreiecken führt die Erhöhung des Polynomgrades von $N = 9$ auf $N = 12$ zu einer Verschlechterung der Ergebnisse.

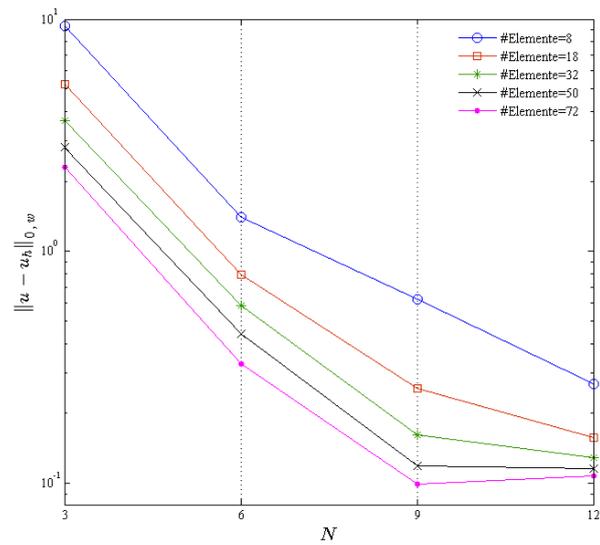


Abbildung 6.18: Approximationsfehler, $N=3, 6, 9, 12$, #Elemente=8, 18, 32, 50, 72, skalare Formulierung, Beispiel 2.

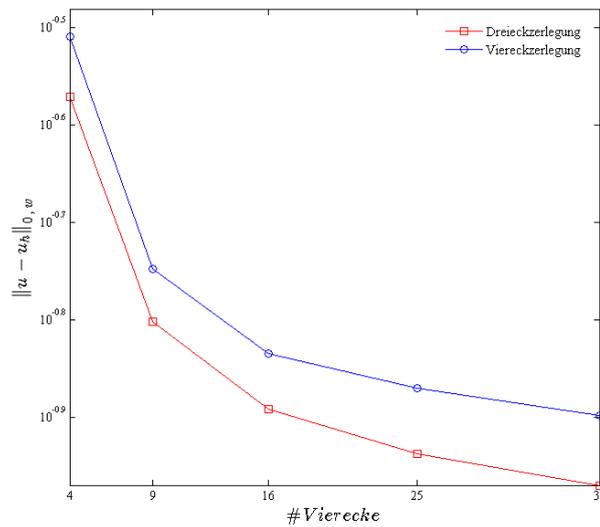


Abbildung 6.19: Vergleich des Fehlerverlaufs, $N=12$ für Drei- und Viereckszerlegung, skalare Formulierung, Beispiel 2.

In Abbildung 6.19 sind die Approximationsfehler für Drei- und Viereckszerlegung jeweils für $N = 12$ grafisch dargestellt. Die beiden Zerlegungsarten zeigen hier bezogen auf die Elementanzahl eine ähnliche Konvergenzgeschwindigkeit.

6.2.3 Skalare vs. System

Für das Beispiel 1 mit $\varepsilon = 0.1$ hat sich die Formulierung als System als bessere Methode bezüglich der Approximationsgenauigkeit durchgesetzt. Hingegen erscheint die skalare Formulierung für eine verstärkte Singularität mit $\varepsilon = 0.001$ die bessere Methode zu sein. Dies können wir in Abbildung 6.20 beobachten. Einen Vergleich der Konditionsgrößen für die beiden Formulierungsarten zeigt Abbildung 6.21.

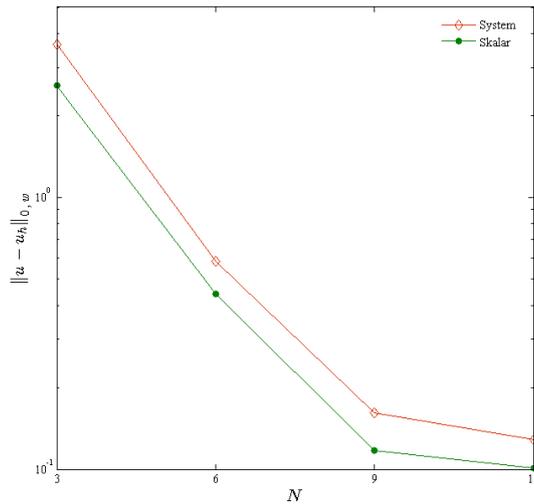


Abbildung 6.20: Approximationsfehler, Skalar vs. System, #Elemente=32, Beispiel 2.

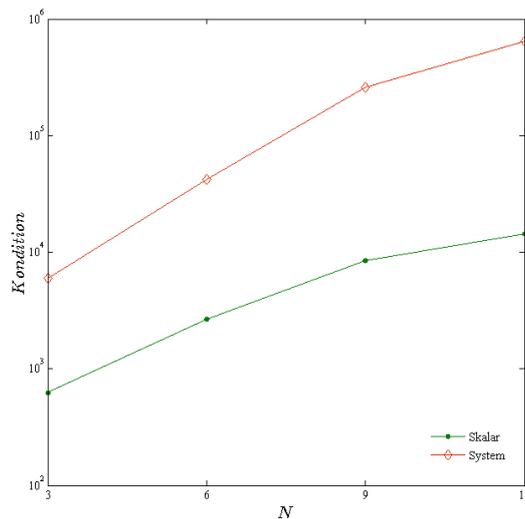


Abbildung 6.21: Konditionen der Matrix A, Skalar vs. System, #Elemente=32, Beispiel 2.

6.2.4 Adaptive Verfeinerung

In den folgenden Abschnitten führen wir eine adaptive Verfeinerung, geregelt durch die drei Kriterien für Beispiel 2, mit einem Anfangsgitter von 32 Elementen durch.

6.2.4.1 Gradienten-Kriterium

Wir entschieden uns ebenso wie bei Beispiel 1 für die Toleranz $\epsilon = 0.1$ und verfeinern die Dreieckselemente, welche dem Kriterium (5.1) nicht genügen.

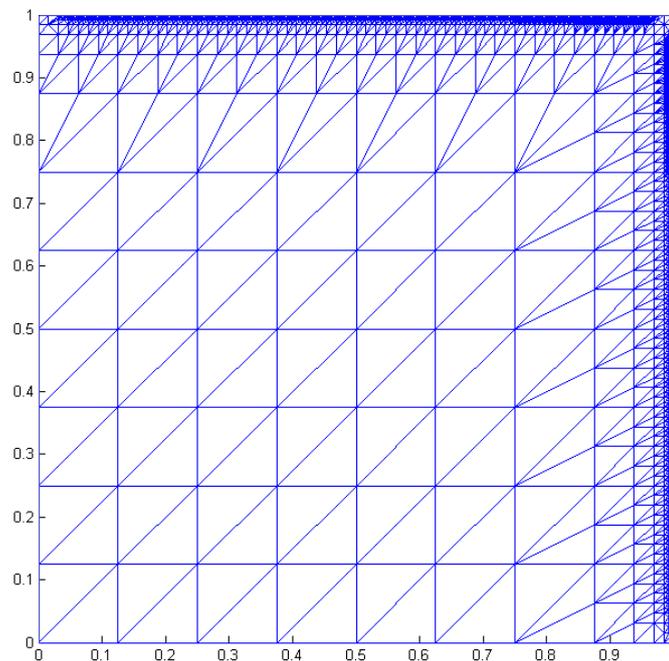


Abbildung 6.22: Adaptive Verfeinerung, Gradienten-Kriterium, #Elemente=3706, $N=3$, Beispiel 2.

Abbildung 6.22 zeigt das adaptiv verfeinerte Gitter. Durch einen Vergleich mit den Konturlinien (siehe Abbildung 6.23) wird klar, warum die meisten Verfeinerungen an den beiden Rändern $x = 1$ und $y = 1$ entstanden sind.

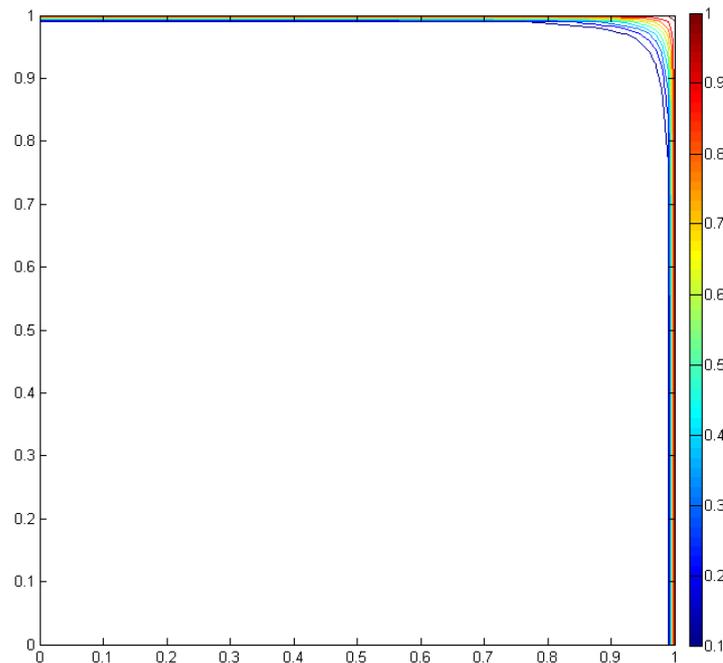


Abbildung 6.23: Konturlinien für Beispiel 2.

Die numerischen Ergebnisse sind in Tabelle 6.12 gegeben. Für $N = 3$ erhält man die meisten Verfeinerungen und entsprechend ist auch hier der Verbesserungseffekt am größten. Ebenso führt die Verfeinerung für $N=6$ zu einer Verbesserung. Bezogen auf die Anzahl der Freiheitsgrade ist hier $N = 3$ unter den vier vorhandenen Polynomgraden am effektivsten. Ein Vergleich mit den entsprechenden Ergebnissen von Beispiel 1 weist darauf hin, dass bei einer verstärkten Singularität die Verwendung niedrigerer Polynomgrade zu einer Verbesserung der Ergebnisse führt. Zudem ist der Verbesserungseffekt für Beispiel 2 geringer als für Beispiel 1 mit $\varepsilon = 0.1$.

Tabelle 6.12: Numerische Ergebnisse nach dem Gradienten-Kriterium , Beispiel 2.

N	$\ u - u_h\ _{0,w}$	#Elemente	#Verfeinerung
3	2.3557e-01	3706	6
6	2.1391e-01	1694	5
9	1.7150e-01	820	4
12	1.3436e-01	396	3

Wie man in Abbildung 6.24 erkennen kann, führt die Verfeinerung für die Polynomgrade $N = 9$ und 12 zu einer leichten Verschlechterung der Ergebnisse.

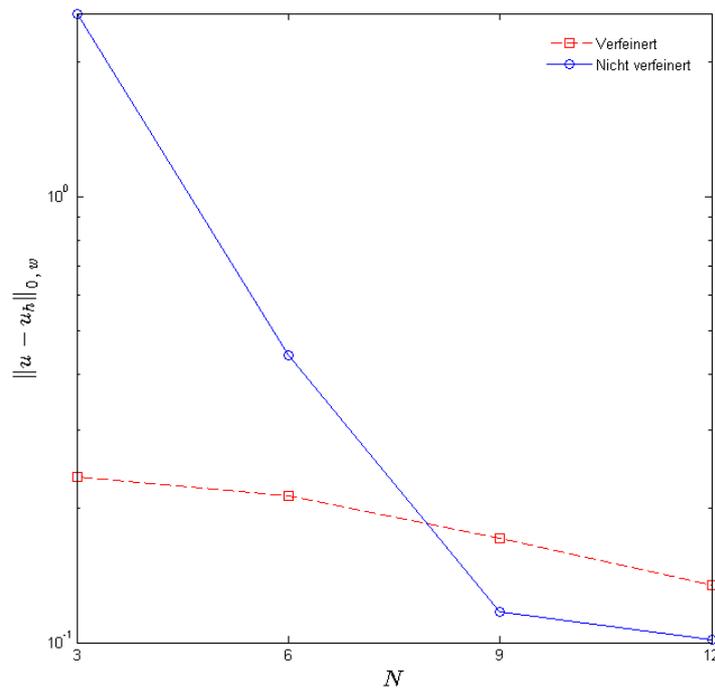


Abbildung 6.24: Vergleich der Approximationsfehler, Beispiel 2, vorher vs. nachher.

6.2.4.2 Abfallende Entwicklungskoeffizienten

Für das zweite Kriterium wählen wir $\epsilon = 1.7250e-04$ aus. Wir führen die adaptive Verfeinerung für $N = 6, 9$ und 12 durch.

Tabelle 6.13: Numerische Ergebnisse nach dem zweiten Kriterium, Beispiel 2.

N	$\ u - u_h\ _{0,w}$	#Elemente	#Verfeinerung
6	3.5819e-01	51	2
9	1.2104e-01	56	1
12	1.1199e-01	38	1

Die numerischen Ergebnisse sind in Tabelle 6.13 enthalten. Die meisten Elemente sind bei dem Polynomgrad $N = 9$ verfeinert worden. Für den Polynomgrad $N=6$ erreichen wir im Vergleich eine größere Verbesserung mit kleinerem Freiheitsgrad. Der Aufwand ist im Vergleich zu den erreichten Ergebnissen unter Verwendung des zweiten Kriteriums geringer als beim Gradienten-Kriterium.

Abbildung 6.25 zeigt die Ergebnisse vor und nach der Verfeinerung für 32 Elemente.

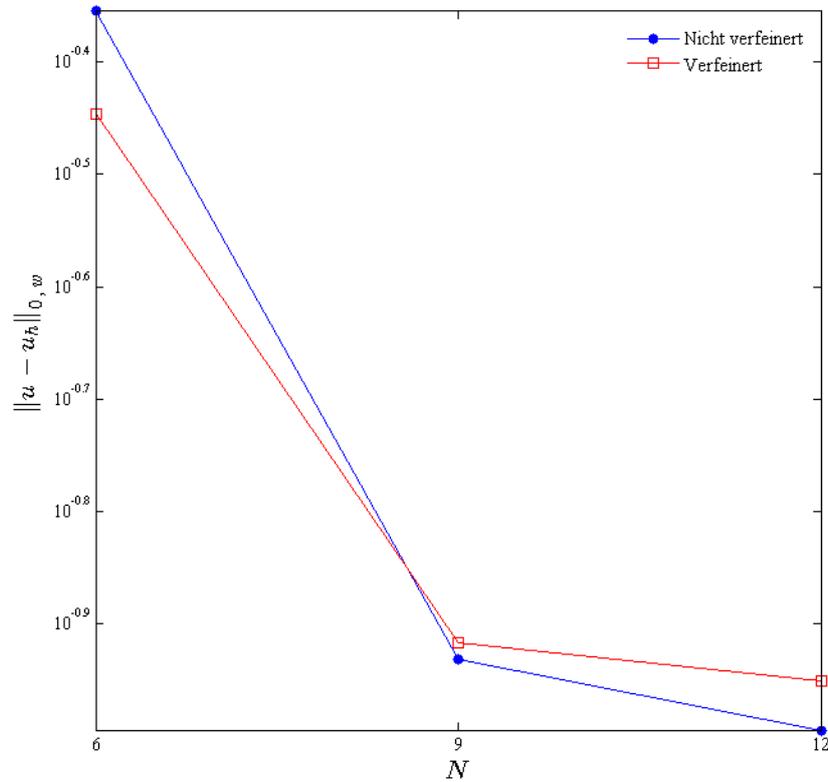


Abbildung 6.25: Vergleich der Approximationsfehler, Kriterium 2, Beispiel 2, vorher vs. nachher.

6.2.4.3 Spektrums-Ende

Die von uns ausgewählte Toleranz beträgt $\epsilon = 5.0000e-05$. Tabelle 6.14 enthält die numerischen Ergebnisse. Hier erhalten wir nur für $N=12$ eine sehr geringe Verschlechterung der Ergebnisse. Für den Polynomgrad $N=3$ erzielen wir mit kleinstem Freiheitsgrad die meisten Verbesserungen.

Tabelle 6.14: Numerische Ergebnisse nach dem dritten Kriterium, Beispiel 2.

N	$\ u - u_h\ _{0,w}$	#Elemente	#Verfeinerung
3	8.8856e-01	167	3
6	3.1738e-01	87	2
9	9.9887e-02	67	1
12	1.0927e-01	43	2

In Abbildung 6.26 haben wir die Approximationsfehler vor und nach der Verfeinerung jeweils für 32 Elemente dargestellt.

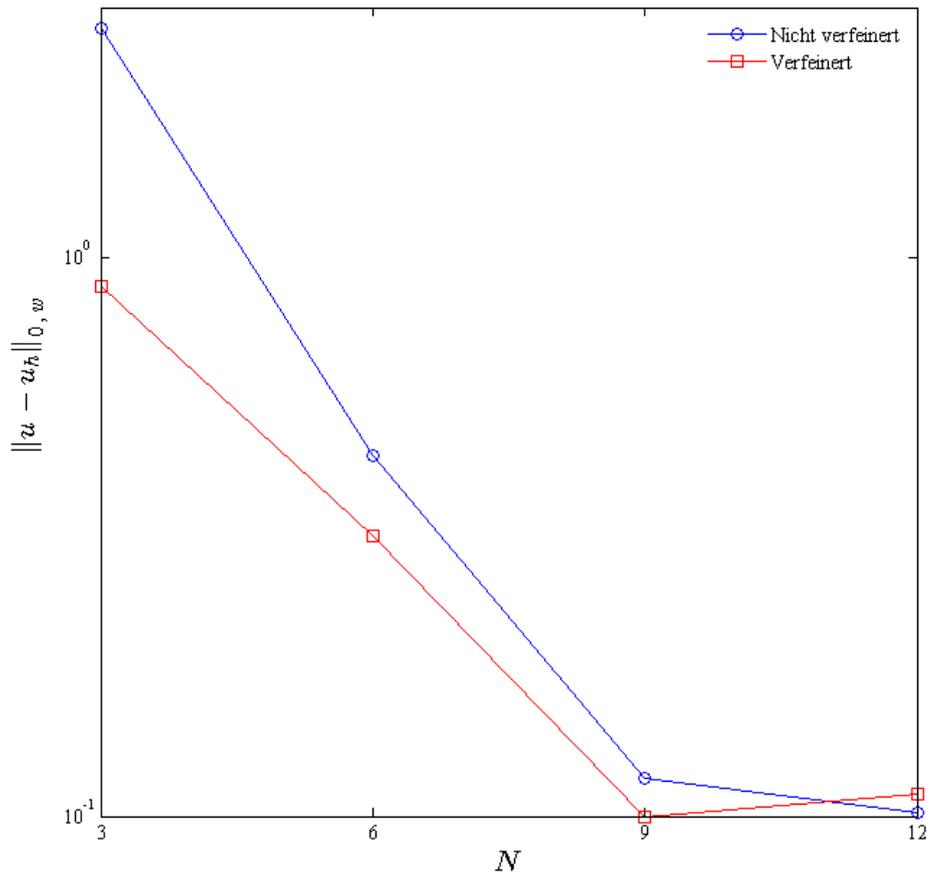


Abbildung 6.26: Vergleich der Approximationsfehler, Kriterium 3, Beispiel 2, vorher vs. nachher.

6.3 Beispiel 3: Hutlayer

Bei Beispiel 3 handelt es sich um eine sogenannte Hut-Funktion. Der Hut ist hier seitlich begrenzt durch den Kreis $(x - 1/2)^2 + (y - 1/2)^2 = 1/16$. Die exakte Lösung u der Konvektions-Diffusions-Gleichung (4.1) ist wie folgt definiert:

$$u = \tanh \left[\frac{1}{\varepsilon} \left(\frac{1}{16} - \left(x - \frac{1}{2} \right)^2 - \left(y - \frac{1}{2} \right)^2 \right) \right], \quad (x, y) \in \Omega = (0, 1)^2 \quad (6.3)$$

wobei $\varepsilon = 0.05$ ist. Die notwendigen Parameter sind wie folgt gegeben:

$$\mathbf{b} = ((4x - 2)u, (4y - 2)u)^t, \quad c = 4$$

mit der entsprechenden rechten Seite

$$f(x, y) = 4 + 4u - 4u^2.$$

Abbildung 6.27 stellt die exakte Lösung u (6.3) grafisch dar.

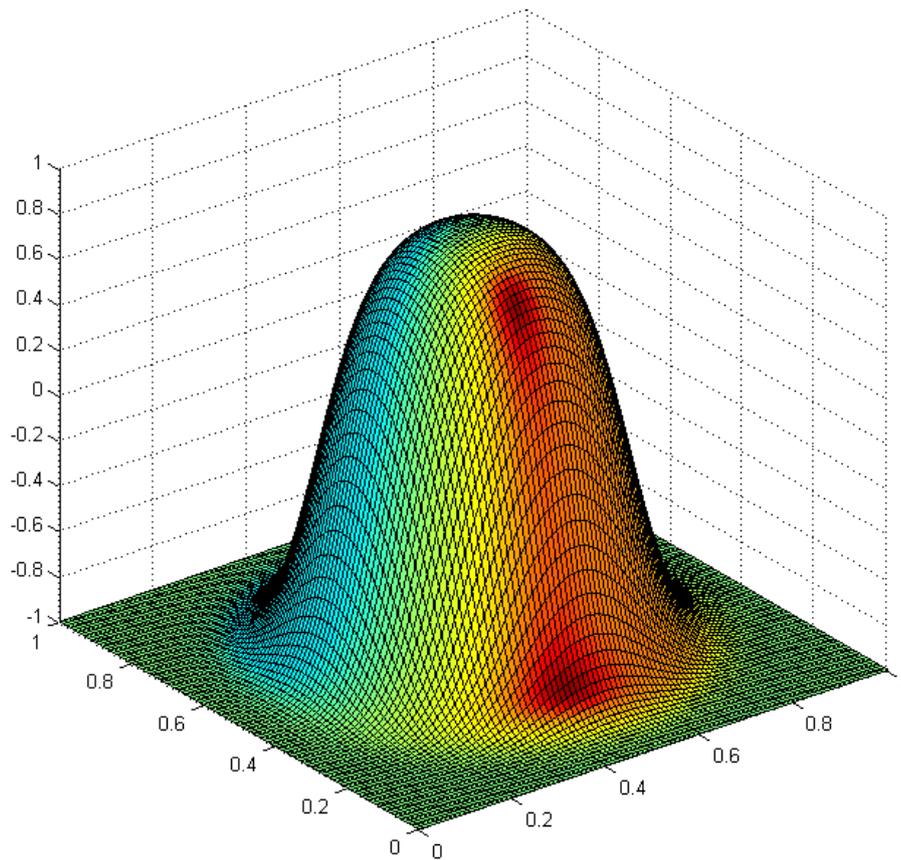


Abbildung 6.27: Plot der exakten Lösung von Beispiel 3.

6.3.1 Ergebnisse für die Formulierung als System

Die Konditionen für die Formulierung als System haben wir in Tabelle 6.15 angegeben.

Tabelle 6.15: Konditionen der Matrix A, Formulierung als System, Beispiel 3.

# Elemente	N = 3	N = 6	N = 9	N = 12
8	4.2176e+02	5.0512e+03	2.8300e+04	7.2177e+04
18	7.2540e+02	1.1074e+04	6.6362e+04	1.6673e+05
32	1.2630e+03	1.9251e+04	1.1978e+05	2.9428e+05

In Tabelle 6.16 beobachten wir die numerischen Ergebnisse, welche wegen der vorhandenen singulären Störung eine geringe Approximationsgenauigkeit aufweisen. Der kleinste Fehler liegt im Bereich 10^{-5} und ist beim Polynomgrad $N=12$ erzielt worden.

Tabelle 6.16: $\|u - u_h\|_{0,w}$, Formulierung als System, Beispiel 3.

# Elemente	N = 3	N = 6	N = 9	N = 12
8	1.3783e+00	6.4151e-01	1.4761e-01	5.3968e-02
18	7.7849e-01	1.3505e-01	7.2309e-03	7.2659e-04
32	7.2694e-01	4.7089e-02	1.2856e-03	8.3748e-05

Die Ergebnisse aus Tabelle 6.16 sind in Abbildung 6.28 grafisch veranschaulicht.

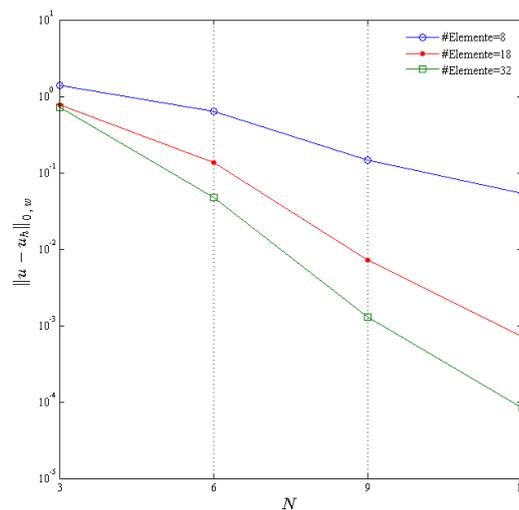


Abbildung 6.28: Approximationsfehler, $N=3, 6, 9, 12$, #Elemente=8, 18, 32, Formulierung als System, Beispiel 3.

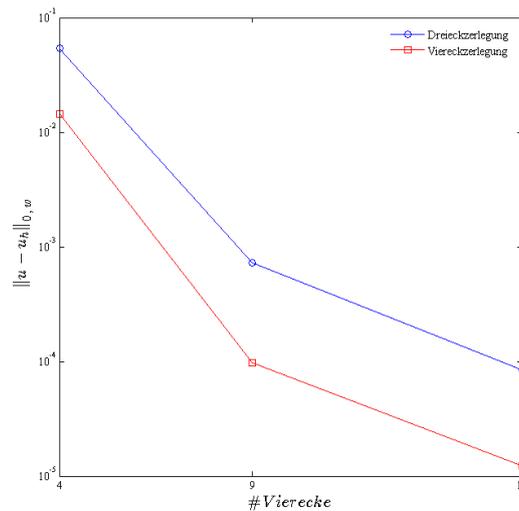


Abbildung 6.29: Vergleich des Fehlerverlaufs, $N=12$, für Drei- und Viereckszerlegung, Formulierung als System, Beispiel 3.

Abbildung 6.29 stellt für uns einen Vergleich der Ergebnisse der Drei- und Viereckszerlegung grafisch dar. Die Werte der Viereckszerlegung wurden der Diplomarbeit von Kippar [15] entnommen.

Wir versuchen nun herauszufinden, wie sich eine verschärfte Singularität auf die Ergebnisse des Beispiels 3 auswirkt. Dies untersuchen wir, indem wir $\varepsilon = 5.00e-3$ wählen. Dadurch verstärken wir die Singularität des Problems.

Die Ergebnisse für die Formulierung als System für den verschärften Hutlayer befinden sich in Tabelle 6.17. Es ist ganz offensichtlich, dass sich die Approximationsgenauigkeit im Vergleich zu der moderaten Singularität deutlich verschlechtert hat. Ebenso ist die Konvergenzgeschwindigkeit verlangsamt worden. Die Fehler liegen in diesem Fall im Bereich 10^0 bis 10^{-1} .

Tabelle 6.17: $\|u - u_h\|_{0,w}$, Beispiel 3 mit verschärfter Singularität, $\varepsilon = 5.00e-3$.

# Elemente	N = 3	N = 6	N = 9	N = 12
8	1.6946e+00	1.8416e+00	1.8821e+00	1.8594e+00
18	1.1214e+00	9.8389e-01	9.7327e-01	9.0475e-01
32	1.3619e+00	1.1175e+00	7.9452e-01	7.4716e-01

In Abbildung 6.30 vergleichen wir die Ergebnisse des verschärften Hutlayers für Drei- und Viereckszerlegung jeweils für $N = 12$.

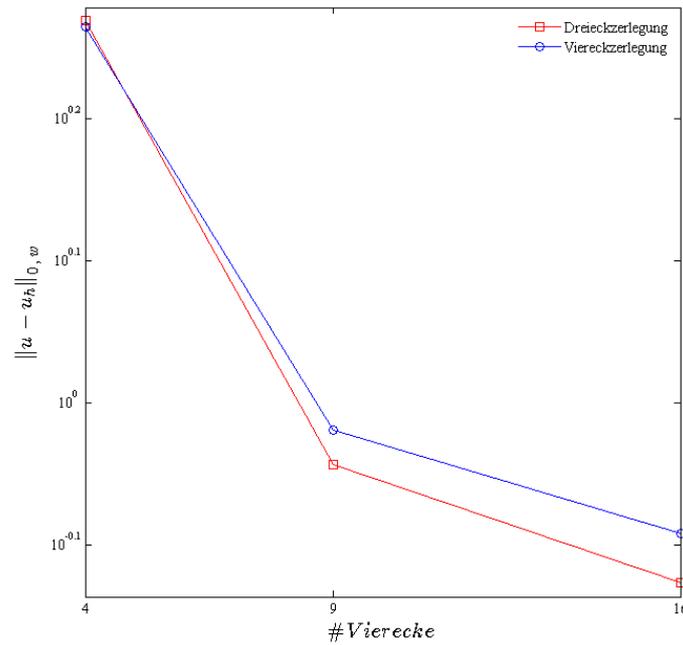


Abbildung 6.30: Vergleich des Fehlerverlaufs, $N=12$, für Drei- und Viereckszerlegung, Formulierung als System, Beispiel 3 mit verschärfter Singularität, $\varepsilon = 5.00e-3$.

Die Dreieckszerlegung scheint somit bessere Ergebnisse zu liefern, wenn die Singularität verstärkt wird.

6.3.2 Ergebnisse für die skalare Formulierung

Tabelle 6.18 beinhaltet die Konditionswerte der Matrix A für Beispiel 3 entsprechend der skalaren Formulierung.

Tabelle 6.18: Konditionen der Matrix A, skalare Formulierung, Beispiel 3.

# Elemente	N = 3	N = 6	N = 9	N = 12
8	1.2522e+02	1.0666e+03	7.5417e+03	1.9224e+04
18	2.5159e+02	3.2308e+03	2.1757e+04	5.4923e+04
32	5.1268e+02	7.2662e+03	4.9539e+04	1.2596e+05
50	8.5793e+02	1.3296e+04	9.2093e+04	2.3582e+05
72	1.3980e+03	2.1762e+04	1.5303e+05	3.9398e+05

Die numerischen Ergebnisse der skalaren Formulierung stehen in Tabelle 6.19 zur Verfügung. Der kleinste Fehler befindet sich im Bereich 10^{-5} und wird für den Polynomgrad $N=12$ erst bei 72 Elementen erreicht. Bei der Formulierung als System erzielen wir bereits bei 32 Elementen den Fehlerbereich 10^{-5} .

Tabelle 6.19: $\|u - u_h\|_{0,w}$, skalare Formulierung, Beispiel 3.

# Elemente	N = 3	N = 6	N = 9	N = 12
8	1.6995e+00	1.0663e+00	3.7060e-01	1.2059e-01
18	1.5116e+00	2.4349e-01	3.7014e-02	5.0581e-03
32	1.0370e+00	1.4791e-01	1.2841e-02	8.3616e-04
50	8.5543e-01	1.0671e-01	3.6955e-03	1.6928e-04
72	7.6591e-01	4.4287e-02	1.6316e-03	2.4491e-05

Die grafische Darstellung der Ergebnisse in Abbildung 6.31 macht uns klar, dass die Konvergenzgeschwindigkeit bei ansteigender Elementanzahl zunimmt.

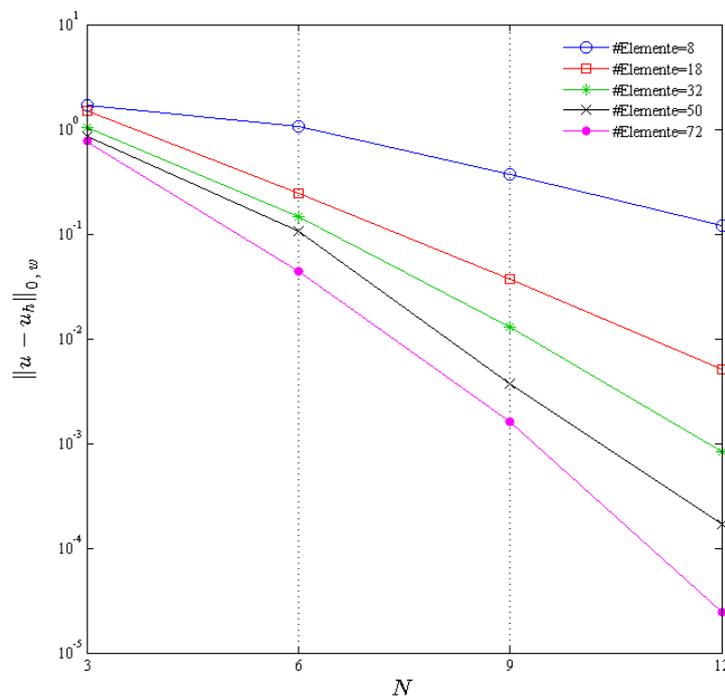


Abbildung 6.31: Approximationsfehler, $N=3, 6, 9, 12$, #Elemente=8, 18, 32, 50, 72, Skalare Formulierung, Beispiel 3.

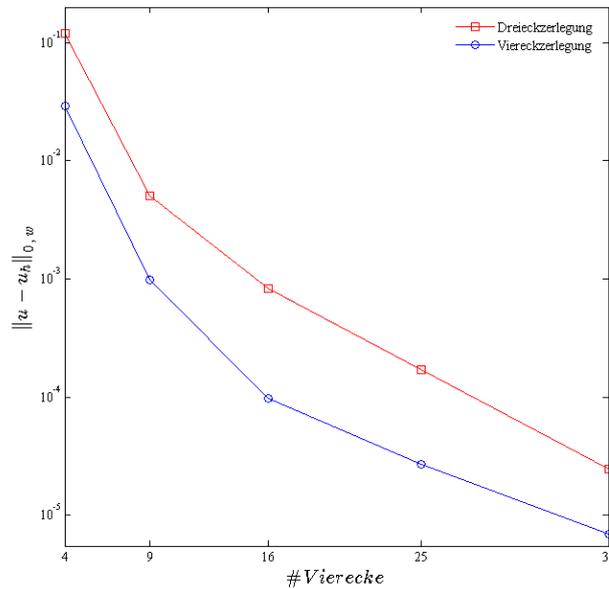


Abbildung 6.32: Vergleich des Fehlerverlaufs, $N=12$, für Drei- und Viereckszerlegung, skalare Formulierung, Beispiel 3.

In Abbildung 6.32 führen wir einen einfachen Vergleich zwischen den Ergebnissen des Beispiels 3 für Drei- und Viereckszerlegung durch und zwar für die skalare Formulierung jeweils nur für $N = 12$.

Für den Polynomgrad $N=12$ liegt der kleinste Fehler bei der Viereckszerlegung im Bereich 10^{-6} . Die Approximationsfehler sind hierbei um eine Zehnerpotenz besser als bei der Dreieckszerlegung.

Wir möchten nun den Einfluss des Störungsparameters ε auf die Ergebnisse der skalaren Formulierung untersuchen. Dafür wählen wir $\varepsilon = 5.00e-3$ aus. Die Ergebnisse finden sich in Tabelle 6.20. Genauso wie bei der Formulierung als System stellt sich eine Verschlechterung der Ergebnisse bzw. eine mildere Konvergenzgeschwindigkeit heraus, welche eindeutig durch die verschärfte Singularität hervorgerufen werden. Die Fehler liegen im Bereich 10^0 bis 10^{-1} .

Tabelle 6.20: $\|u - u_h\|_{0,w}$, skalare Formulierung, Beispiel 3 mit verschärfte Singularität, $\varepsilon = 5.00e-3$

# Elemente	N = 3	N = 6	N = 9	N = 12
8	1.9375e+00	2.1197e+00	2.0562e+00	2.0560e+00
18	1.2999e+00	9.8885e-01	9.2738e-01	7.5954e-01
32	1.3144e+00	1.0930e+00	8.0602e-01	7.4087e-01
50	9.7614e-01	9.1425e-01	7.8919e-01	8.4666e-01
72	9.3152e-01	9.9138e-01	8.8154e-01	8.4589e-01

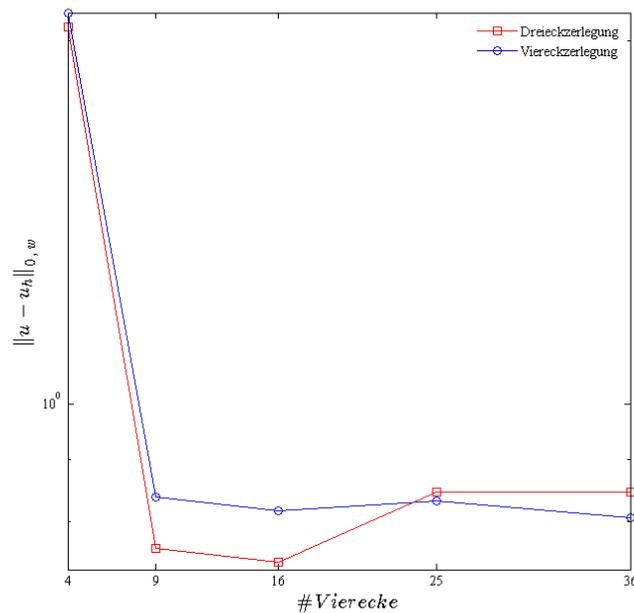


Abbildung 6.33: Vergleich des Fehlerverlaufs, $N=12$, für Drei- und Viereckszerlegung, skalare Formulierung, Beispiel 3 mit verschärfter Singularität, $\varepsilon = 5.00e-3$.

Die Ergebnisse des verschärften Hutlayers für $N = 12$ bezüglich der Drei- und Viereckszerlegung sind in Abbildung 6.33 dargestellt.

Es scheint also die Dreieckszerlegung für verschärfte Singularitäten angesichts der Approximationsgenauigkeit zumindest bis zu einer Elementanzahl von 16 Vierecken die Viereckszerlegung zu übertreffen. Ferner führt eine gleichmäßige Verfeinerung des Gebietes auf 25 Vierecke, das entspricht 50 Dreiecken, bezogen auf die skalare Formulierung für die beiden Zerlegungsarten zu einer Verschlechterung der Approximationsgenauigkeit.

6.3.3 Skalare vs. System

In den Abbildungen 6.34 und 6.35 vergleichen wir die Ergebnisse der beiden Formulierungsarten jeweils für 32 Elementanzahl. Hier sehen wir, dass die Approximationsgenauigkeit der Formulierung als System bei wachsendem Polynomgrad um eine Zehnerpotenz besser als die skalare Formulierung ist. Die Formulierung als System scheint für eine moderate Singularität bezogen auf die Approximationsgenauigkeit bessere Ergebnisse zu liefern. Dafür wird aber deutlich mehr Speicherplatz und Zeit beansprucht als bei der skalaren Formulierung.

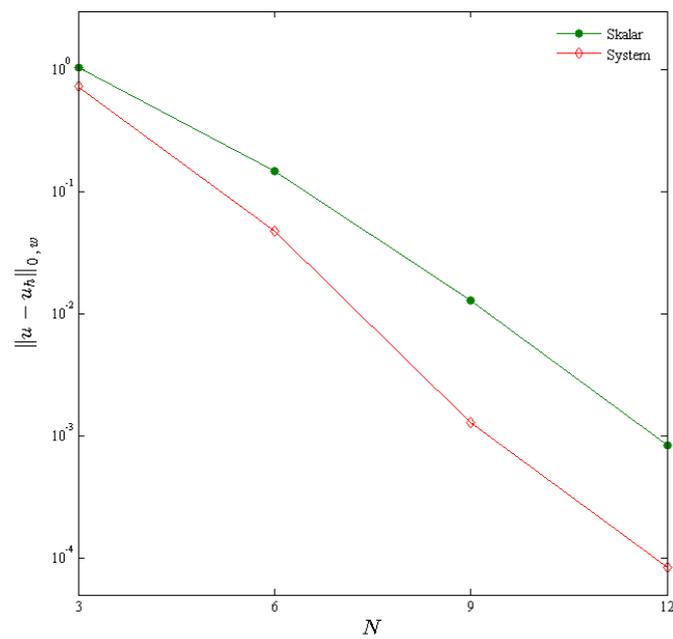
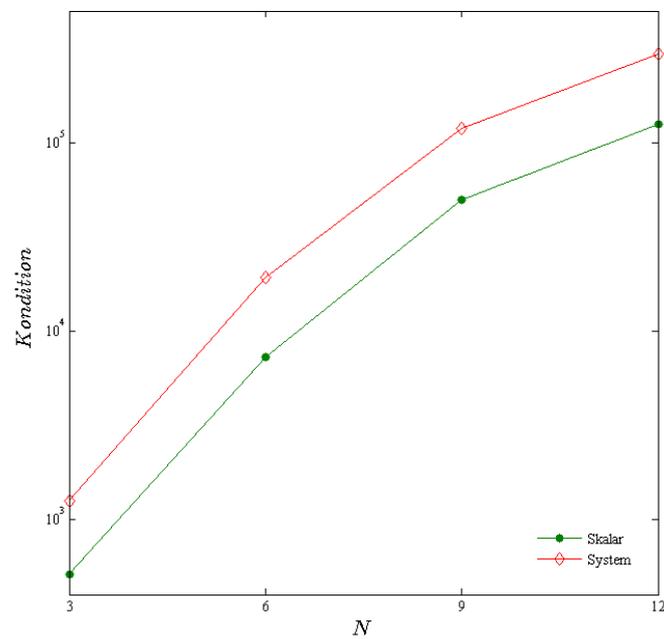


Abbildung 6.34: Approximationsfehler, Skalar vs. System, #Elemente=32, Beispiel 3.

Abbildung 6.35: Konditionen der Matrix A , Skalar vs. System, #Elemente=32, Beispiel 3.

6.3.4 Adaptive Verfeinerung

Im Folgenden führen wir die adaptive Verfeinerung für Beispiel 3 mit $\varepsilon = 0.05$ durch.

6.3.4.1 Gradienten-Kriterium

Das Anfangsgitter besteht aus 32 Dreiecken. Die Toleranz beträgt $\varepsilon = 0.23$. Abbildung 6.36 zeigt uns, wie die adaptive Verfeinerung des Gebietes nach dem Gradienten-Kriterium aussieht.

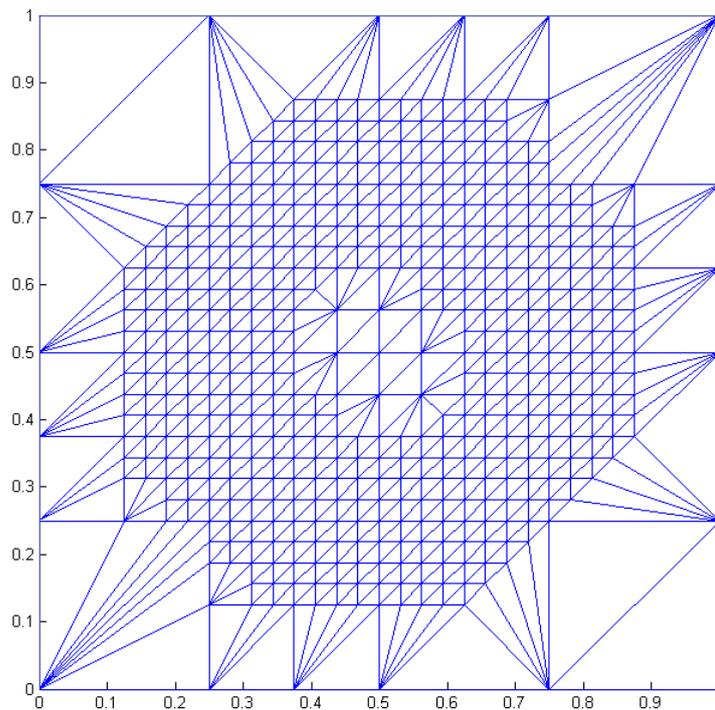


Abbildung 6.36: Adaptive Verfeinerung, Gradienten-Kriterium, $N=6$, #Elemente=972, Beispiel 3.

Mit einem Blick auf die Konturlinien in Abbildung 6.37 stellen wir fest, warum es in Abbildung 6.36 die Verfeinerung in Form eines Kreises entstanden ist.

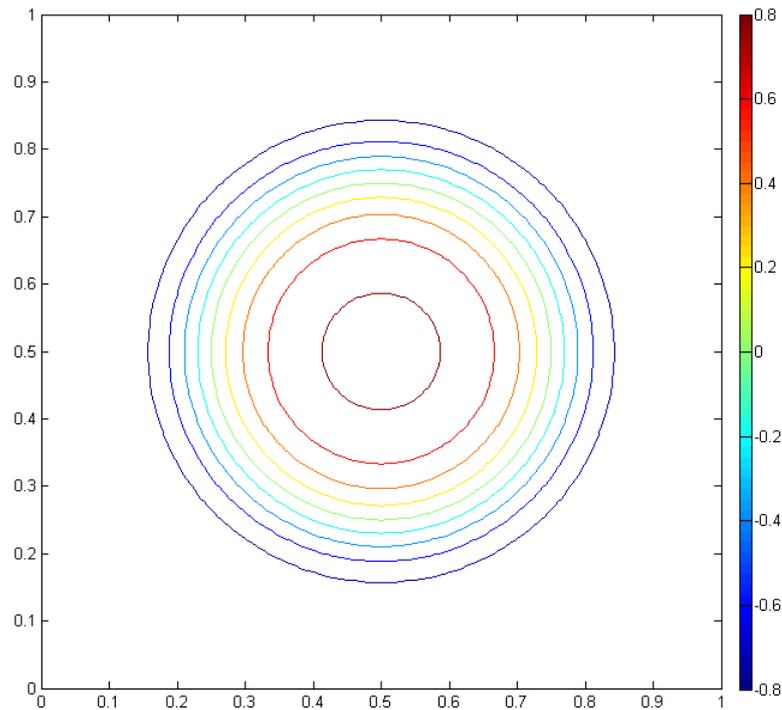


Abbildung 6.37: Konturlinien für Beispiel 3.

Die Ergebnisse der Verfeinerung sind in Tabelle 6.21 dargestellt. Wie erwartet, werden bei $N=3$ mehr Elemente verfeinert als bei höheren Polynomgraden.

Tabelle 6.21: Numerische Ergebnisse nach dem Gradienten-Kriterium, Beispiel 3.

N	$\ u - u_h\ _{0,w}$	#Elemente	#Verfeinerung
3	1.7399e-02	3566	4
6	1.2927e-04	972	3
9	1.6217e-05	300	2
12	6.5923e-07	300	2

Wie in Abbildung 6.38 erkennbar, scheint für $N=3$ und 6 die Verfeinerung mehr zur Verbesserung der Ergebnisse beigetragen zu haben. Der Polynomgrad $N=3$ besitzt zudem den größten Freiheitsgrad, während $N=6$ den zweitkleinsten Freiheitsgrad hat.

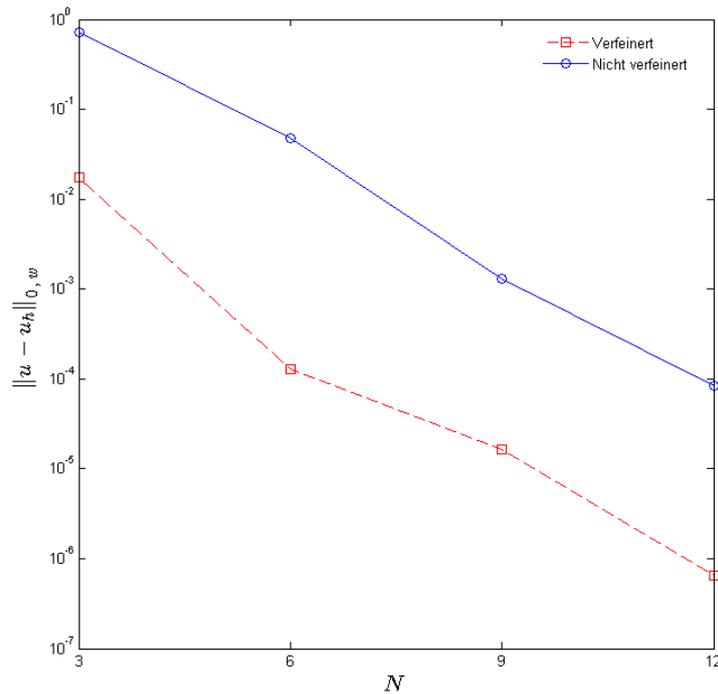


Abbildung 6.38: Vergleich der Approximationsfehler, Beispiel 3, vorher vs. nachher.

6.3.4.2 Abfallende Entwicklungskoeffizienten

Für das zweite Kriterium nehmen wir die Toleranz $\epsilon = 2.2300e-06$ an. Das Anfangsgitter enthält 32 Dreiecke.

Tabelle 6.22: Numerische Ergebnisse nach dem zweiten Kriterium, Beispiel 3

N	$\ u - u_h\ _{0,w}$	#Elemente	#Verfeinerung
6	1.3974e-03	213	3
9	1.2925e-03	38	1
12	1.3619e-05	93	2

Wenn wir die Ergebnisse in Tabelle 6.22 analysieren bzw. Abbildung 6.39 betrachten, fällt auf, dass die Verfeinerung in diesem Fall vor allem für den Polynomgrad $N=6$ am effektivsten ist. Für $N=9$ haben wir zwar weniger Verfeinerungen und den kleinsten Freiheitsgrad, das Ergebnis hat sich aber nach der Verfeinerung leicht verschlechtert. Die Approximationsgenauigkeit ist bei $N=6$ im Bereich 10^{-3} , während diese bei $N=12$ mit dem größten Freiheitsgrad im Bereich 10^{-5} liegt.

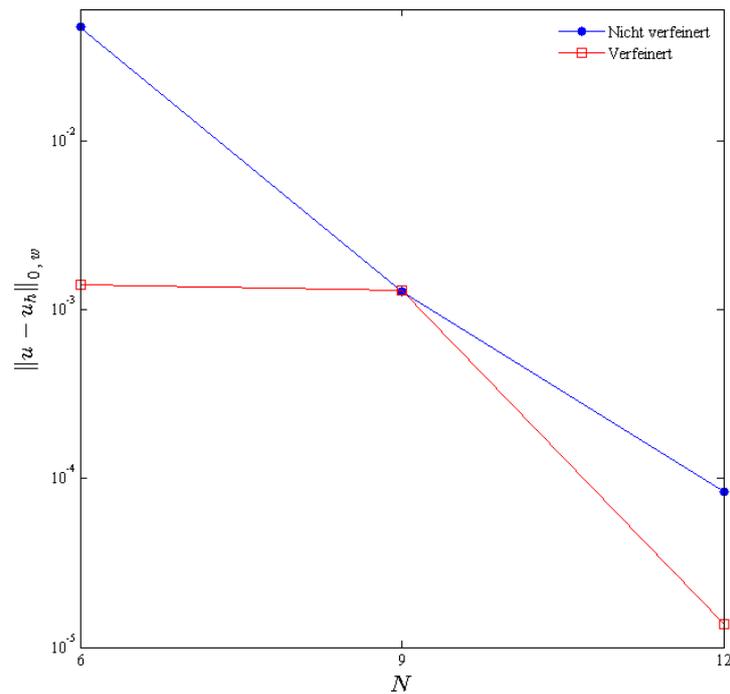


Abbildung 6.39: Vergleich der Approximationsfehler, Kriterium 2, Beispiel 3, vorher vs. nachher.

6.3.4.3 Spektrums-Ende

Das dritte Kriterium testen wir mit $\epsilon = 9.0000e-06$ und einem Anfangsgitter von 32 Dreiecken.

Tabelle 6.23: Numerische Ergebnisse nach dem dritten Kriterium, Beispiel 3.

N	$\ u - u_h\ _{0,w}$	#Elemente	#Verfeinerung
3	8.9157e-02	323	4
6	1.4063e-03	100	2
9	3.7564e-04	87	1
12	6.8597e-05	37	2

Tabelle 6.23 und Abbildung 6.40 entnehmen wir, dass die höchsten Verbesserungseffekte bei $N=3$ und 6 erreicht werden. $N=6$ verfügt über den kleinsten Freiheitsgrad unter den vier Polynomgraden.

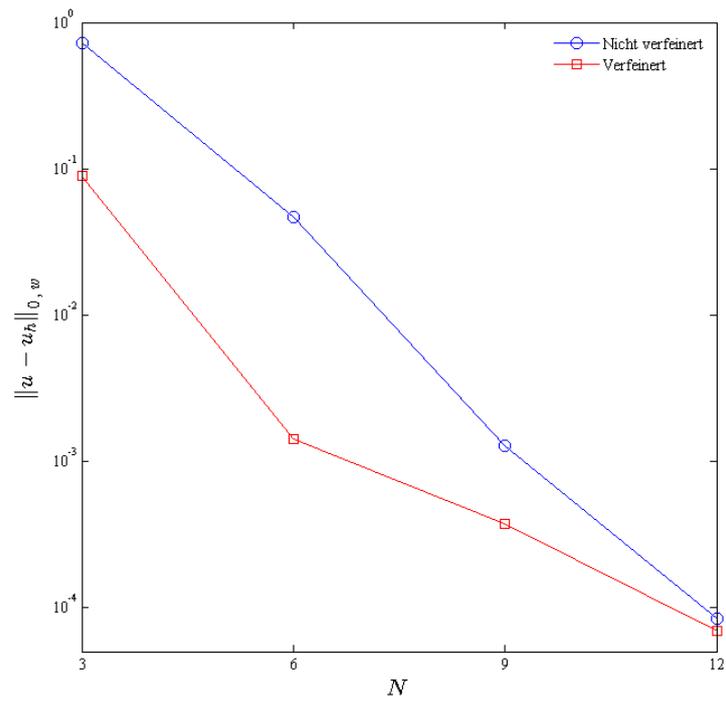


Abbildung 6.40: Vergleich der Approximationsfehler, Kriterium 3, Beispiel 3, vorher vs. nachher.

6.4 Beispiel 4: Eckenlayer

In Beispiel 4 betrachten wir eine Funktion, welche überall in dem Gebiet den Wert 0 annimmt und nur nahe des Punktes $(0, 0)$ sehr steil ansteigt. Dadurch entsteht in der Ecke $(0, 0)$ einen Boundary-Layer, also eine Singularität, welche die Approximationsgenauigkeit negativ beeinflusst.

Die exakte Lösung u der Konvektions-Diffusions-Gleichung (4.1) ist gegeben durch

$$u = \exp\left(-\frac{x^6 + y^6}{\varepsilon}\right), \quad (x, y) \in \Omega = (0, 1)^2 \quad (6.4)$$

wobei $\varepsilon = 1.0e-4$ ist. Die notwendigen Parameter sind wie folgt gegeben:

$$\mathbf{b} = (1/4 - x, 1/4 - y)^t, \quad c = 1$$

mit der Funktion der rechten Seite

$$f(x, y) = [-36(x^{10} + y^{10}) - 1.5(x^5 + y^5) + 6(x^6 + y^6) + (30(x^4 + y^4) + 1)\varepsilon] \frac{u}{\varepsilon}.$$

Die grafische Darstellung von u ist in Abbildung 6.41 illustriert.

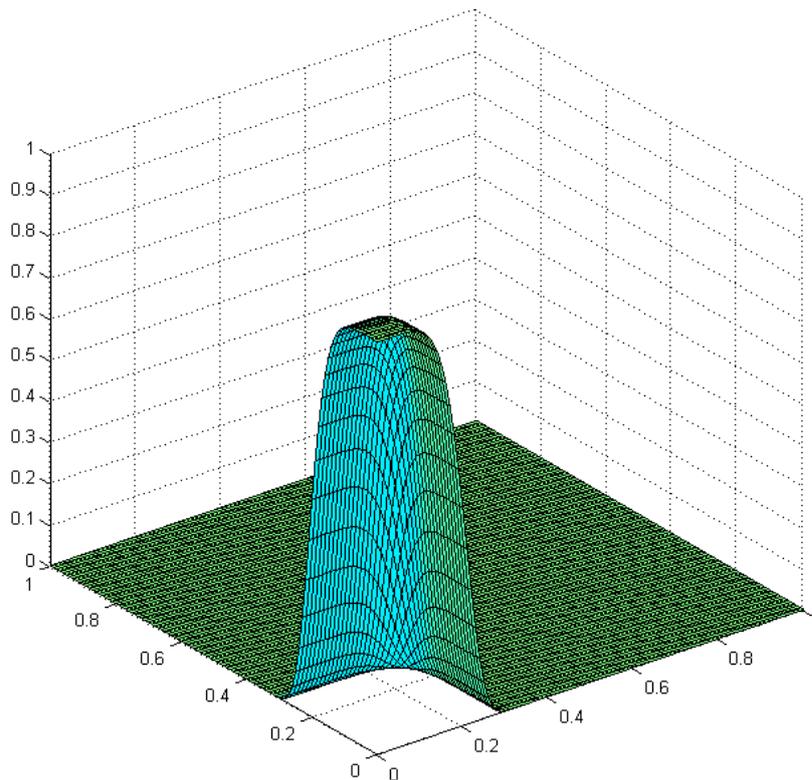


Abbildung 6.41: Plot der exakten Lösung von Beispiel 4.

6.4.1 Ergebnisse für die Formulierung als System

Die Konditionszahlen und die numerischen Ergebnisse sind in den Tabellen 6.24 bzw. 6.25 angegeben.

Die Approximationsfehler liegen im Bereich 10^{-1} bis 10^{-5} . Der kleinste Fehler erreichen wir beim Polynomgrad $N=12$.

Tabelle 6.24: Konditionen der Matrix A, Formulierung als System, Beispiel 4.

# Elemente	N = 3	N = 6	N = 9	N = 12
8	7.3185e+02	1.2017e+04	5.8264e+04	1.0095e+05
18	1.7823e+03	2.8929e+04	1.0448e+05	2.4631e+05
32	3.4109e+03	4.5335e+04	1.8594e+05	4.5314e+05

Tabelle 6.25: $\|u - u_h\|_{0,w}$, Formulierung als System, Beispiel 4.

# Elemente	N = 3	N = 6	N = 9	N = 12
8	1.0783e-01	3.8892e-02	1.8068e-02	7.3148e-03
18	8.5610e-02	1.7449e-02	3.9319e-03	8.3939e-04
32	6.6483e-02	4.3256e-03	5.1395e-04	4.8105e-05

Die Daten aus Tabelle 6.25 haben wir in Abbildung 6.42 grafisch dargestellt. Die Konvergenzrate steigt mit wachsender Elementanzahl an.

In Abbildung 6.43 haben wir, wie bei den anderen Beispielen, die Ergebnisse der Dreieckszerlegung mit den Ergebnissen der Viereckszerlegung aus [15] verglichen. Im Gegensatz zur Dreieckszerlegung tritt für die Viereckszerlegung ab einer Elementanzahl von 9 Vierecken eine Verschlechterung der Approximationsgenauigkeit ein. Hier scheint, dass die Dreieckszerlegung für höhere Polynomgrade bessere Ergebnisse liefert, je mehr man das Gebiet gleichmäßig verfeinert.

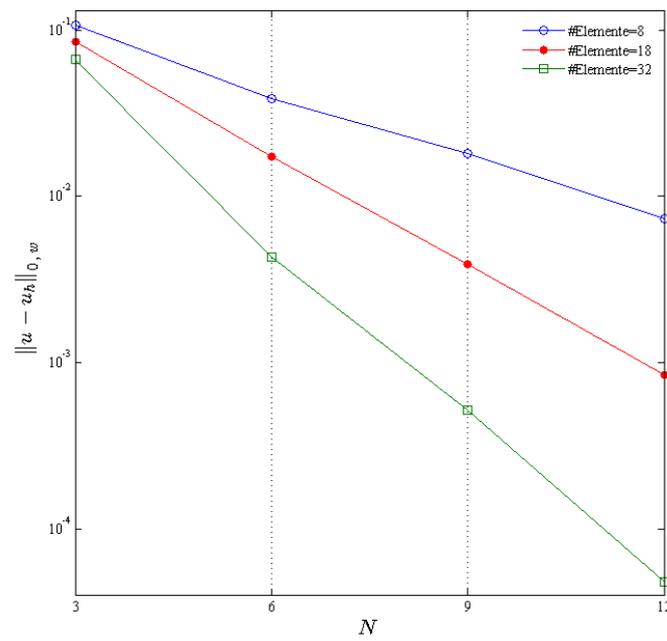


Abbildung 6.42: Approximationsfehler, $N=3, 6, 9, 12$, #Elemente=8, 18, 32, Formulierung als System, Beispiel 4.

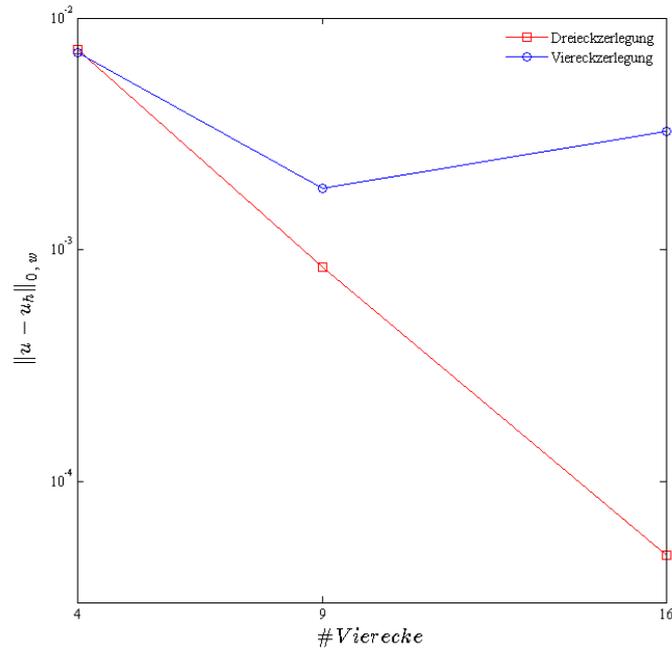


Abbildung 6.43: Vergleich des Fehlerverlaufs, $N=12$, für Drei- und Viereckszerlegung, Beispiel 4, Formulierung als System.

6.4.2 Ergebnisse für die skalare Formulierung

Die Konditionsgrößen und die Approximationsfehler sind in den Tabellen 6.26 und 6.27 enthalten. Die Approximationsfehler der beiden Formulierungsarten liegen für 8 bis 32 Elemente nahe beieinander. Den kleinsten Fehler erreichen wir für eine Elementanzahl von 72 Dreiecken für den Polynomgrad $N=12$.

Tabelle 6.26: Konditionen der Matrix A, skalare Formulierung, Beispiel 4.

# Elemente	N = 3	N = 6	N = 9	N = 12
8	1.3754e+02	5.9798e+02	1.8800e+03	3.0833e+03
18	2.7053e+02	1.1912e+03	3.7215e+03	6.1396e+03
32	4.4130e+02	2.0938e+03	6.6695e+03	1.1052e+04
50	6.5514e+02	3.1379e+03	1.0465e+04	1.7658e+04
72	9.1243e+02	4.3063e+03	1.4297e+04	2.4176e+04

Tabelle 6.27: $\|u - u_h\|_{0,w}$, skalare Formulierung, Beispiel 4.

# Elemente	N = 3	N = 6	N = 9	N = 12
8	1.1885e-01	3.9615e-02	1.8228e-02	7.3703e-03
18	8.9497e-02	1.7122e-02	4.1138e-03	8.6602e-04
32	6.2329e-02	3.7283e-03	4.8061e-04	5.5518e-05
50	1.7713e-02	1.8082e-03	1.9184e-04	3.5805e-05
72	2.7664e-02	1.3620e-03	1.5463e-04	7.9948e-06

In Abbildung 6.44 können wir die Ergebnisse aus Tabelle 6.27 grafisch betrachten. Ab 50 Elementen tragen die weiteren Erhöhungen der Elementanzahl nicht unbedingt zu einer bedeutsamen Verbesserung der Approximationsgenauigkeit bei. Hier scheint die Elementanzahl von 32 eine optimale Gittergröße zu sein.

Kippar hat in [15] das gleiche Beispiel nur für bis zu 25 Viereckselementen getestet. Durch Abbildung 6.45 können wir die Ergebnisse der Drei- und Viereckszerlegung miteinander vergleichen, und zwar für $N=12$ und 4, 9, 16 und 25 Vierecke. Ebenso wie bei der Formulierung als System erzielt die Dreieckszerlegung hier die besseren Ergebnisse. Auch ist wie bei der Formulierung als System eine Verschlechterung der Ergebnisse der Viereckszerlegung für eine Vierecksanzahl von 16 erkennbar, während wir bei der Dreieckszerlegung vor allem für eine Vierecksanzahl von 4 bis 16 eine Verbesserung um zwei Zehnerpotenzen beobachten.

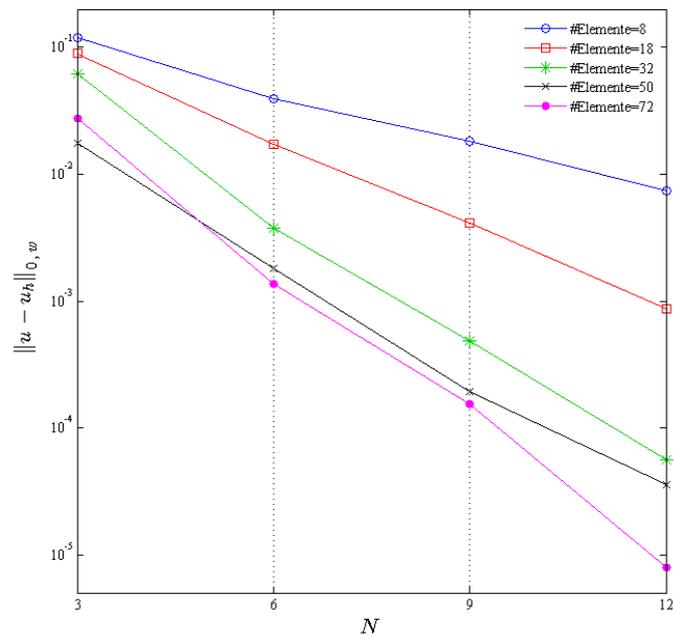


Abbildung 6.44: Approximationsfehler, $N=3, 6, 9, 12$, #Elemente=8, 18, 32, 50, Skalare Formulierung, Beispiel 4.

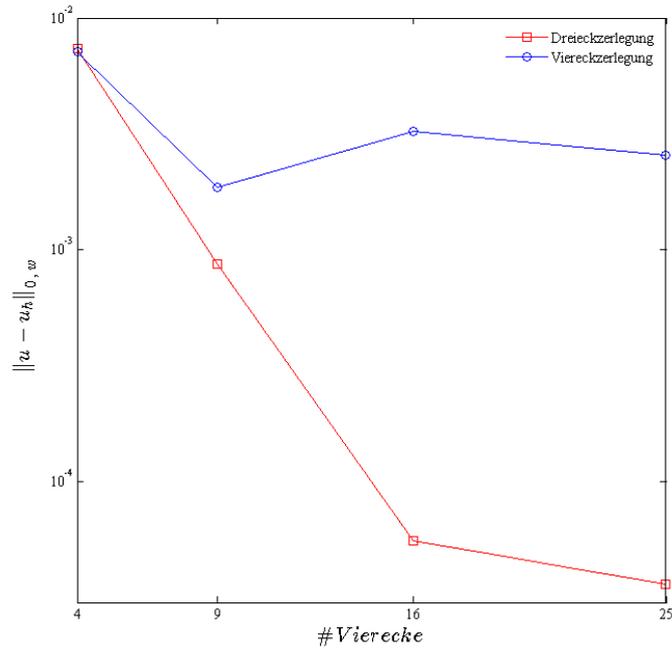


Abbildung 6.45: Vergleich des Fehlerverlaufs, $N=12$, für Drei- und Viereckszerlegung, Beispiel 4, skalare Formulierung.

6.4.3 Skalare vs. System

Die Ergebnisse der beiden Formulierungsarten sind in Abbildung 6.46 skizziert. Dabei können keine imposanten Unterschiede zwischen den Approximationsfehlern der beiden Methoden erkannt werden. Dies trifft für die Viereckszerlegung in [15] auch zu.

Abbildung 6.47 visualisiert die Konditionszahlen der beiden Methoden. Bei der Formulierung als System läuft das Wachstum der Konditionsgröße etwas schneller als bei der skalaren Formulierung.

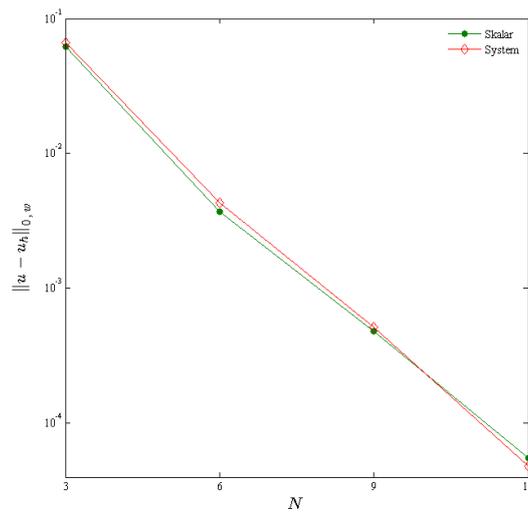


Abbildung 6.46: Approximationsfehler, Skalar vs. System, Beispiel 4, #Elemente=32.

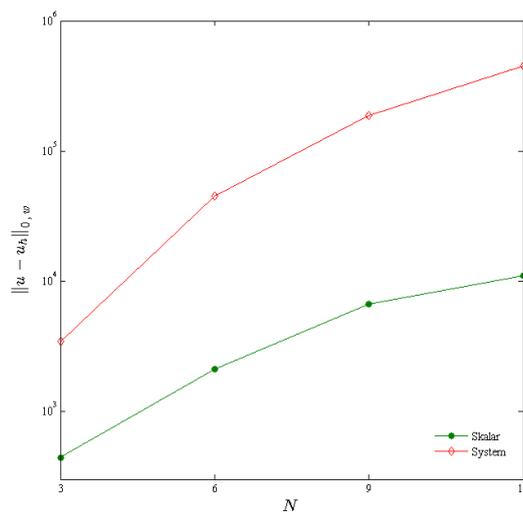


Abbildung 6.47: Konditionen der Matrix A, Skalar vs. System, Beispiel 4, #Elemente=32.

6.4.4 Adaptive Verfeinerung

Mit einem Ausgangsgitter von 32 Dreieckselementen testen wir im Folgenden die drei Adaptivitätskriterien.

6.4.4.1 Gradienten Kriterium

Wir nehmen die Toleranz $\epsilon = 6.7500e - 02$ an. Abbildung 6.48 zeigt uns, dass die adaptive Verfeinerung erwartungsgemäß im Bereich des Boundary-Layers erfolgt ist.

Die Konturlinien sind in Abbildung 6.49 dargestellt.

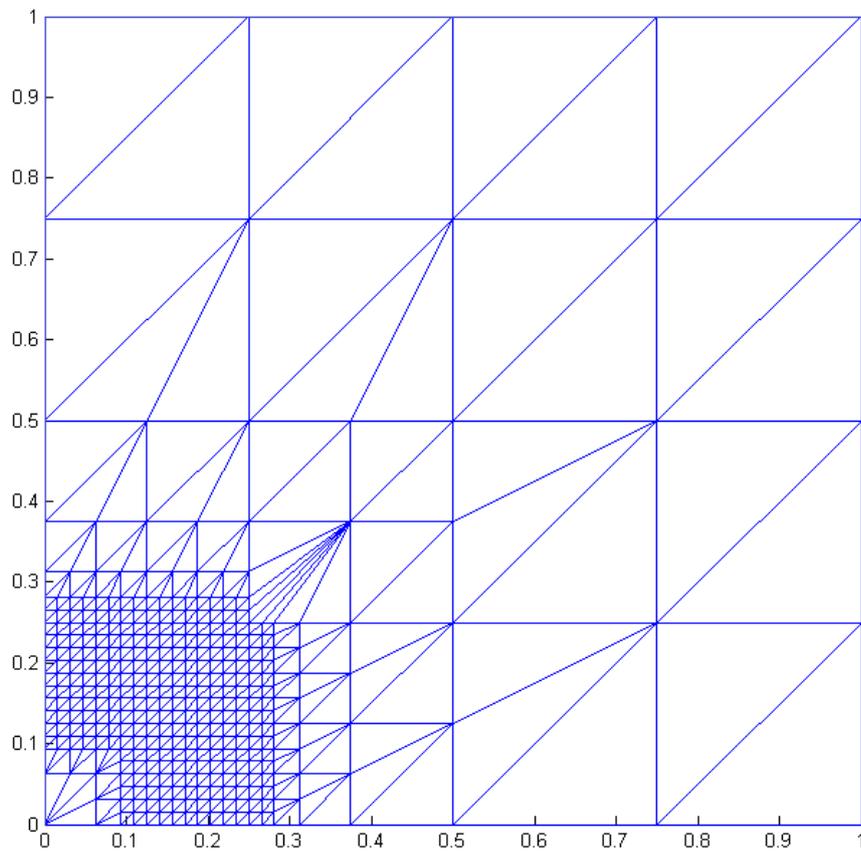


Abbildung 6.48: Adaptive Verfeinerung, Gradienten-Kriterium, $N=6$, #Elemente=714, Beispiel 4.

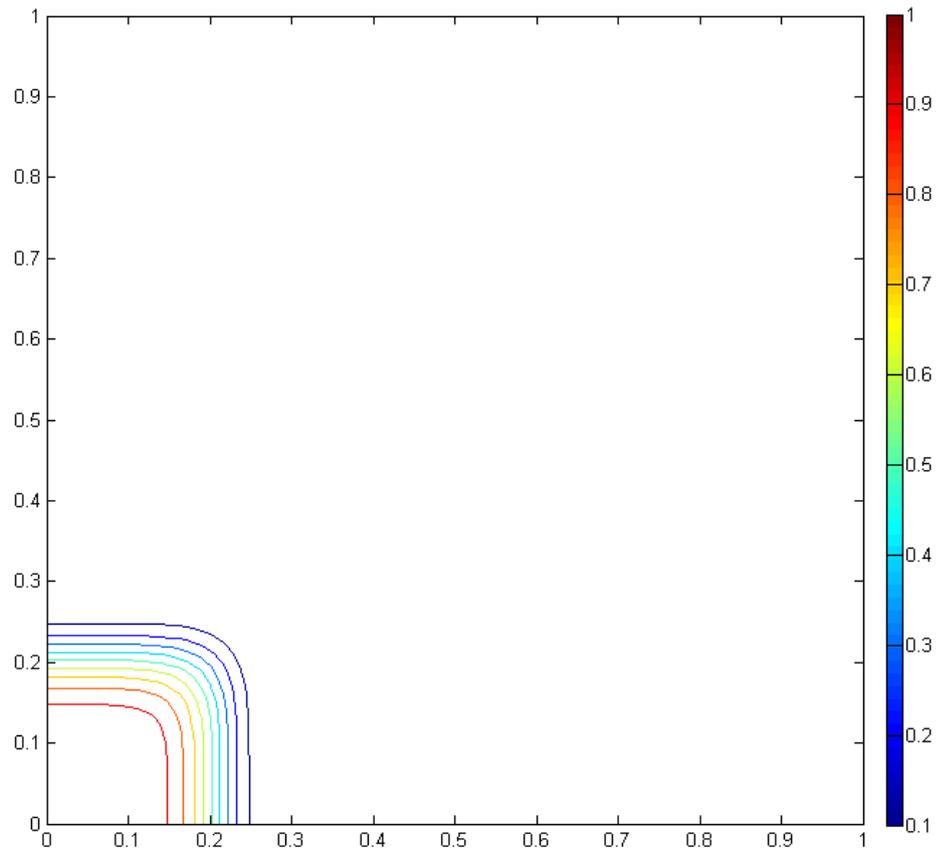


Abbildung 6.49: Konturlinien für Beispiel 4.

In Tabelle 6.28 befinden sich die numerischen Ergebnisse, welche nach den adaptiven Verfeinerungen entstanden sind. Für $N=3$ und 6 haben wir eine Verbesserung der Approximationsfehler um zwei Zehnerpotenzen, während diese für $N=9$ und 12 um nur eine Zehnerpotenz verbessert wurden. Der Polynomgrad $N=9$ verfügt über den kleinsten Freiheitsgrad.

Tabelle 6.28: Numerische Ergebnisse nach dem Gradienten-Kriterium, Beispiel 4.

N	$\ u - u_h\ _{0,w}$	#Elemente	#Verfeinerung
3	4.4010e-04	2466	5
6	2.3486e-05	714	4
9	1.0364e-05	254	3
12	1.4471e-06	254	3

Abbildung 6.50 vergleicht die Approximationsgenauigkeit vor und nach der Verfeinerung bezogen auf das Gradienten-Kriterium. Der Verbesserungseffekt ist nach der Verfeinerung bei einer Erhöhung des Polynomgrades von $N=6$ auf $N=9$ nur dürftig.

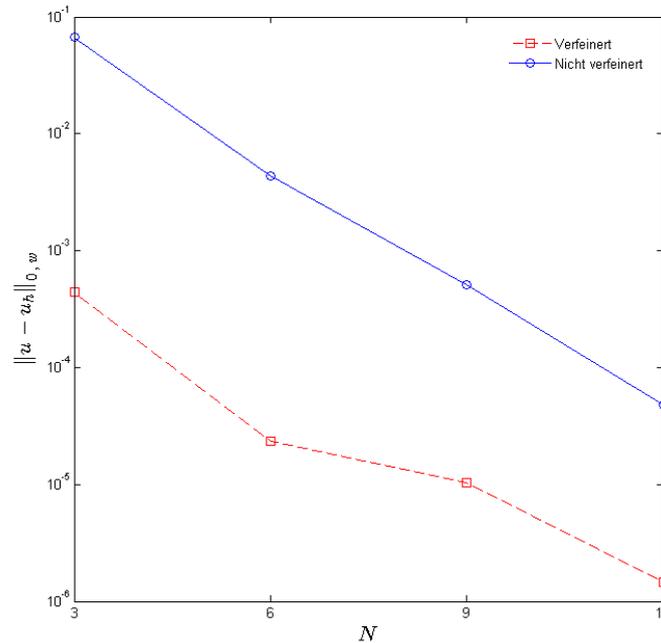


Abbildung 6.50: Vergleich der Approximationsfehler, Beispiel 4, vorher vs. nachher.

6.4.4.2 Abfallende Entwicklungskoeffizienten

Wir setzen $\epsilon = 1.0000e-06$ ein und verfeinern das Gitter nach dem zweiten Kriterium. Die Ergebnisse sind in Tabelle 6.29 enthalten.

Tabelle 6.29: Numerische Ergebnisse nach dem zweiten Kriterium, Beispiel 4.

N	$\ u - u_h\ _{0,w}$	#Elemente	#Verfeinerung
6	5.4768e-04	174	4
9	3.3661e-04	48	1
12	3.7228e-05	43	2

Abbildung 6.51 zeigt die Ergebnisse für vor und nach der adaptiven Verfeinerung entsprechend dem zweiten Kriterium. Für $N=6$ haben wir eine stärkere Verbesserung durch die adaptive Verfeinerung erreicht, wobei diese einen größeren Freiheitsgrad hat als $N=9$ und 12.

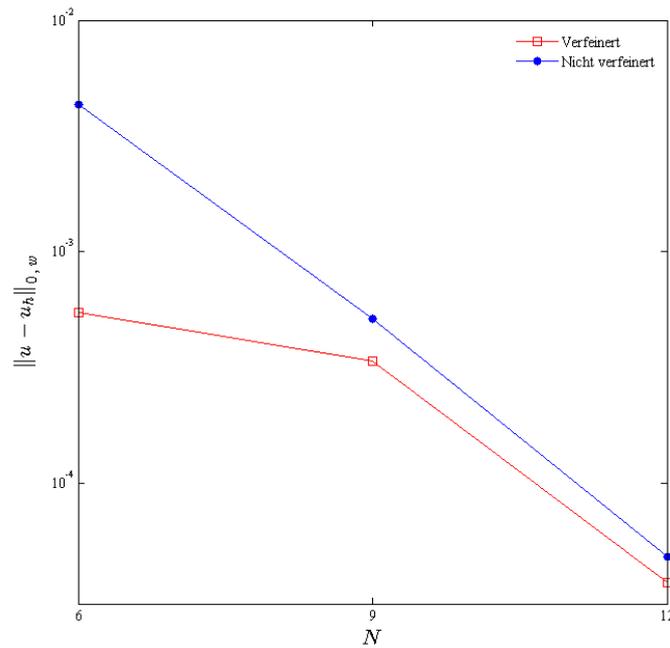


Abbildung 6.51: Vergleich der Approximationsfehler, Kriterium 2, Beispiel 4, vorher vs. nachher.

6.4.4.3 Spektrums-Ende

Für die Toleranz ϵ entscheiden wir uns für $4.2500e - 05$. Tabelle 6.30 enthält die Ergebnisse der Verfeinerung.

Für $N=6$ erreichen wir eine Verbesserung um etwa eine Zehnerpotenz durch die adaptive Verfeinerung. Zudem besitzt der Polynomgrad $N=6$ den kleinsten Freiheitsgrad.

Tabelle 6.30: Numerische Ergebnisse nach dem dritten Kriterium, Beispiel 4.

N	$\ u - u_h\ _{0,w}$	#Elemente	#Verfeinerung
3	1.1774e-02	149	2
6	8.4338e-04	44	1
9	3.7584e-04	40	1
12	4.2674e-05	37	1

Die Ergebnisse der Verfeinerung nach dem dritten Kriterium sind in Abbildung 6.52 grafisch dargestellt. Bei einem Sprung vom Polynomgrad $N=6$ auf $N=9$ erreichen wir hierbei eine geringe Verbesserung der Ergebnisse, während der Freiheitsgrad deutlich steigt.

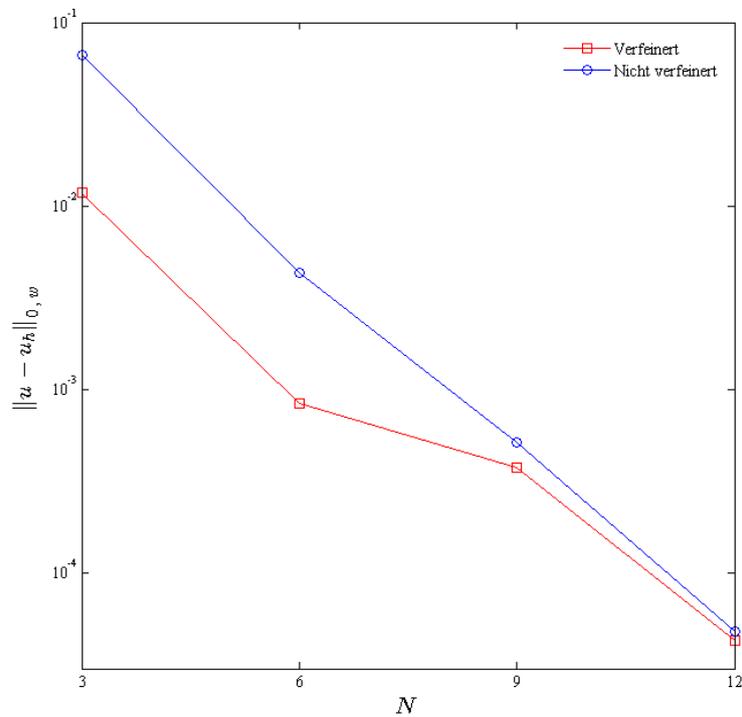


Abbildung 6.52: Vergleich der Approximationsfehler, Kriterium 3, Beispiel 4, vorher vs. nachher.

7 Ausblick

Wir haben für die vier Beispiele jeweils zwei Formulierungsmethoden eingesetzt, nämlich die skalare Formulierung und die Formulierung als System. Die Analyse der Ergebnisse hat gezeigt, dass die Formulierung als System vor allem bei den Problemen eine bessere Approximationsgenauigkeit liefert, in denen die Singularität vergleichsweise moderater ist. Bei einer Verkleinerung der Störungsparameter, das bedeutet bei einem verschärften Boundary-Layer, ermittelt die skalare Formulierung generell eine bessere Approximationsgenauigkeit. Da die Unterschiede zwischen den Ergebnissen nicht zu groß sind, kann man angesichts der numerischen Kosten der Formulierung als System die skalare Formulierung als effizientere Methode präsentieren.

Wir haben versucht einen ungefähren Vergleich zwischen der Drei- und Viereckszerlegung durchzuführen. Diesen konnten wir aufgrund der vorhandenen Daten in [15] nur für den Polynomgrad $N=12$ auszuführen. Somit basiert unsere Schlussfolgerung nur auf die Ergebnisse für $N=12$.

Eine Gegenüberstellung der beiden Zerlegungsarten weist darauf hin, dass die Dreieckszerlegung bei verschärften Singularitäten generell bessere Ergebnisse liefern kann. Allerdings sind dabei die Unterschiede der Ergebnisse der beiden Zerlegungsarten so gering, dass man wegen der hohen numerischen Kosten die Viereckszerlegung empfehlen kann.

Die vier Probleme, die wir in dem letzten Kapitel bearbeitet haben, sind die singulargestörten Probleme. Durch adaptive Verfeinerung haben wir versucht die Güte der approximierten Lösungen zu verbessern. Dabei verwendeten wir drei adaptiven Kriterien:

- Gradienten-Kriterium
- Abfallende Entwicklungskoeffizienten
- Spektrums-Ende

Durch das Gradienten-Kriterium bekommen wir zwar eine gute Auflösung im Bereich des Boundary-Layers, dafür ist aber der Speicherplatzbedarf groß. Dagegen erhalten wir durch das zweite und dritte Kriterium mit geringeren numerischen Kosten gute Approximationsgenauigkeiten. Der Nachteil des zweiten Kriteriums ist die Anwendung der linearen Regression. Dadurch kann man dieses Kriterium nur für Polynomgrade $N \geq 6$ verwenden.

Das Problem haben wir aber bei dem dritten Kriterium nicht und daher kann das dritte Kriterium hierbei eher empfohlen werden.

Unter den vier Polynomgraden ($N \in \{3, 6, 9, 12\}$) hat sich $N=6$, aber auch $N=3$, als optimale Polynomgrade erwiesen, welche im Allgemeinen mit einem kleinerem Freiheitsgrad die größten Verbesserungen durch die adaptiven Verfeinerungen erreicht haben.

Die hier durchgeführte adaptive Verfeinerung (h -Verfeinerung) kann man in Zukunft weiterentwickeln, indem man auch Polynomgrade in einzelnen Dreieckselementen ändert (hp -Verfeinerung), vorausgesetzt dass mehr Polynomgrade von Fekete-Punkten vorhanden sind.

Abbildungsverzeichnis

2.1	Fekete-Punkte vom Grad $N = 3, 6, 9$ und 12 im Standard-Dreieck \mathbb{T} . . .	16
3.1	Boundary-Layer, Verhalten der exakten Lösung des Beispiels 3.1.	21
4.1	Dreieckszerlegung von $\Omega = (0, 1)^2$ für $k=3$, Nummerierung der Dreiecke mit zwei Indizes	23
4.2	Dreieckszerlegung von $\Omega = (0, 1)^2$ für $k=3$ mit einfachem Index.	26
4.3	Nummerierung der Kollokationspunkte für $N = 3$	27
4.4	Nummerierung der Randpunkte von Ω mit zwei Dreieckselementen ($k = 1$), $N = 3$, $i = 1, 2, \dots, 4kN = 12$ und $j = 1, 2, \dots, 2k^2N_t = 20$	28
4.5	Nummerierte Interface-Punkte auf der Sekante eines Elementes mit ungeradem Index.	30
5.1	Die Belegungsstruktur der Koeffizienten a_{mn} für $N = 6$	39
5.2	\bar{a} für die glatte Funktion $u = \exp(0.5x + y)$	41
5.3	Verfeinerung von Dreiecken: (links) rotes, (Mitte) grünes und (rechts) blaues Element.	42
6.1	Plot der exakten Lösung von Beispiel 1.	44
6.2	Approximationsfehler, $N=3, 6, 9, 12$, #Elemente= 8, 18, 32, Formulierung als System, Beispiel 1.	46
6.3	Vergleich des Fehlerverlaufs, $N=12$, Drei- und Viereckszerlegung, Formulierung als System, Beispiel 1.	46
6.4	Approximationsfehler, $N=3, 6, 9, 12$, #Elemente= 8, 18, 32, 50, 72, skalare Formulierung, Beispiel 1.	48
6.5	Vergleich des Fehlerverlaufs, $N=12$, Drei- und Viereckszerlegung, skalare Formulierung, Beispiel 1.	48
6.6	Approximationsfehler, Skalar vs. System für 32 Elemente, Beispiel 1.	49
6.7	Konditionen der Matrix A , Skalar vs. System für 32 Elemente, Beispiel 1.	49
6.8	Belegungsstruktur der Matrix A , $N = 6$, #Elemente= 32, Formulierung als System, Beispiel 1, Anzahl der von 0 verschiedenen Einträge = 104016, A eine 3624×2688 -Matrix.	50
6.9	Belegungsstruktur der Matrix A für $N = 6$, #Elemente= 32, skalare Formulierung, Beispiel 1, Anzahl der von 0 verschiedenen Einträge = 41336, A eine 1552×896 -Matrix.	51
6.10	Adaptive Verfeinerung, Gradienten-Kriterium, #Elemente=4556, $N = 3$, Beispiel 1.	52
6.11	Konturlinien der exakten Lösung u , Beispiel 1.	53

6.12	Vergleich der Approximationsfehler, Beispiel 1, vorher vs. nachher. . . .	54
6.13	Vergleich der Approximationsfehler, Kriterium 2, Beispiel 1, vorher vs. nachher.	55
6.14	Vergleich der Approximationsfehler, Kriterium 3, Beispiel 1, vorher vs. nachher.	56
6.15	Plot der exakten Lösung von Beispiel 2.	57
6.16	Approximationsfehler, $N=3, 6, 9, 12$, #Elemente= 8, 18, 32, Formulierung als System, Beispiel 2.	59
6.17	Vergleich des Fehlerverlaufs, $N=12$, für Drei- und Viereckszerlegung, Formulierung als System, Beispiel 2.	59
6.18	Approximationsfehler, $N=3, 6, 9, 12$, #Elemente=8, 18, 32, 50, 72, skalare Formulierung, Beispiel 2.	61
6.19	Vergleich des Fehlerverlaufs, $N=12$ für Drei- und Viereckszerlegung, skalare Formulierung, Beispiel 2.	61
6.20	Approximationsfehler, Skalar vs. System, #Elemente=32, Beispiel 2. . .	62
6.21	Konditionen der Matrix A , Skalar vs. System, #Elemente=32, Beispiel 2.	62
6.22	Adaptive Verfeinerung, Gradienten-Kriterium, #Elemente=3706, $N=3$, Beispiel 2.	63
6.23	Konturlinien für Beispiel 2.	64
6.24	Vergleich der Approximationsfehler, Beispiel 2, vorher vs. nachher. . . .	65
6.25	Vergleich der Approximationsfehler, Kriterium 2, Beispiel 2, vorher vs. nachher.	66
6.26	Vergleich der Approximationsfehler, Kriterium 3, Beispiel 2, vorher vs. nachher.	67
6.27	Plot der exakten Lösung von Beispiel 3.	68
6.28	Approximationsfehler, $N=3, 6, 9, 12$, #Elemente=8, 18, 32, Formulierung als System, Beispiel 3.	69
6.29	Vergleich des Fehlerverlaufs, $N=12$, für Drei- und Viereckszerlegung, Formulierung als System, Beispiel 3.	70
6.30	Vergleich des Fehlerverlaufs, $N=12$, für Drei- und Viereckszerlegung, Formulierung als System, Beispiel 3 mit verschärfter Singularität, $\varepsilon = 5.00e-3$	71
6.31	Approximationsfehler, $N=3, 6, 9, 12$, #Elemente=8, 18, 32, 50, 72, Skalare Formulierung, Beispiel 3.	72
6.32	Vergleich des Fehlerverlaufs, $N=12$, für Drei- und Viereckszerlegung, skalare Formulierung, Beispiel 3.	73
6.33	Vergleich des Fehlerverlaufs, $N=12$, für Drei- und Viereckszerlegung, skalare Formulierung, Beispiel 3 mit verschärfter Singularität, $\varepsilon = 5.00e-3$	74
6.34	Approximationsfehler, Skalar vs. System, #Elemente=32, Beispiel 3. . .	75
6.35	Konditionen der Matrix A , Skalar vs. System, #Elemente=32, Beispiel 3.	75

6.36 Adaptive Verfeinerung, Gradienten-Kriterium, $N=6$, #Elemente=972, Beispiel 3.	76
6.37 Konturlinien für Beispiel 3.	77
6.38 Vergleich der Approximationsfehler, Beispiel 3, vorher vs. nachher. . . .	78
6.39 Vergleich der Approximationsfehler, Kriterium 2, Beispiel 3, vorher vs. nachher.	79
6.40 Vergleich der Approximationsfehler, Kriterium 3, Beispiel 3, vorher vs. nachher.	80
6.41 Plot der exakten Lösung von Beispiel 4.	81
6.42 Approximationsfehler, $N=3, 6, 9, 12$, #Elemente=8, 18, 32, Formulierung als System, Beispiel 4.	83
6.43 Vergleich des Fehlerverlaufs, $N=12$, für Drei- und Viereckszerlegung, Beispiel 4, Formulierung als System.	83
6.44 Approximationsfehler, $N=3, 6, 9, 12$, #Elemente=8, 18, 32, 50, Skalare Formulierung, Beispiel 4.	85
6.45 Vergleich des Fehlerverlaufs, $N=12$, für Drei- und Viereckszerlegung, Beispiel 4, skalare Formulierung.	85
6.46 Approximationsfehler, Skalar vs. System, Beispiel 4, #Elemente=32. . .	86
6.47 Konditionen der Matrix A, Skalar vs. System, Beispiel 4, #Elemente=32.	86
6.48 Adaptive Verfeinerung, Gradienten-Kriterium, $N=6$, #Elemente=714, Beispiel 4.	87
6.49 Konturlinien für Beispiel 4.	88
6.50 Vergleich der Approximationsfehler, Beispiel 4, vorher vs. nachher. . . .	89
6.51 Vergleich der Approximationsfehler, Kriterium 2, Beispiel 4, vorher vs. nachher.	90
6.52 Vergleich der Approximationsfehler, Kriterium 3, Beispiel 4, vorher vs. nachher.	91

Tabellenverzeichnis

6.1	Konditionen der Matrix A, Formulierung als System, Beispiel 1.	45
6.2	$\ u - u_h\ _{0,w}$, Formulierung als System, Beispiel 1.	45
6.3	Konditionen der Matrix A, skalare Formulierung, Beispiel 1.	47
6.4	$\ u - u_h\ _{0,w}$, skalare Formulierung, Beispiel 1.	47
6.5	Numerische Ergebnisse nach dem Gradienten-Kriterium, Beispiel 1.	53
6.6	Numerische Ergebnisse nach dem zweiten Kriterium, Beispiel 1.	54
6.7	Numerische Ergebnisse nach dem dritten Kriterium, Beispiel 1.	55
6.8	Konditionen der Matrix A, Formulierung als System, Beispiel 2.	58
6.9	$\ u - u_h\ _{0,w}$, Formulierung als System, Beispiel 2.	58
6.10	Konditionen der Matrix A, skalare Formulierung, Beispiel 2.	60
6.11	$\ u - u_h\ _{0,w}$, skalare Formulierung, Beispiel 2.	60
6.12	Numerische Ergebnisse nach dem Gradienten-Kriterium, Beispiel 2.	64
6.13	Numerische Ergebnisse nach dem zweiten Kriterium, Beispiel 2.	65
6.14	Numerische Ergebnisse nach dem dritten Kriterium, Beispiel 2.	66
6.15	Konditionen der Matrix A, Formulierung als System, Beispiel 3.	69
6.16	$\ u - u_h\ _{0,w}$, Formulierung als System, Beispiel 3.	69
6.17	$\ u - u_h\ _{0,w}$, Beispiel 3 mit verschärfter Singularität, $\varepsilon = 5.00e-3$	70
6.18	Konditionen der Matrix A, skalare Formulierung, Beispiel 3.	71
6.19	$\ u - u_h\ _{0,w}$, skalare Formulierung, Beispiel 3.	72
6.20	$\ u - u_h\ _{0,w}$, skalare Formulierung, Beispiel 3 mit verschärfter Singu- larität, $\varepsilon = 5.00e-3$	73
6.21	Numerische Ergebnisse nach dem Gradienten-Kriterium, Beispiel 3.	77
6.22	Numerische Ergebnisse nach dem zweiten Kriterium, Beispiel 3	78
6.23	Numerische Ergebnisse nach dem dritten Kriterium, Beispiel 3.	79
6.24	Konditionen der Matrix A, Formulierung als System, Beispiel 4.	82
6.25	$\ u - u_h\ _{0,w}$, Formulierung als System, Beispiel 4.	82
6.26	Konditionen der Matrix A, skalare Formulierung, Beispiel 4.	84
6.27	$\ u - u_h\ _{0,w}$, skalare Formulierung, Beispiel 4.	84
6.28	Numerische Ergebnisse nach dem Gradienten-Kriterium, Beispiel 4.	88
6.29	Numerische Ergebnisse nach dem zweiten Kriterium, Beispiel 4.	89
6.30	Numerische Ergebnisse nach dem dritten Kriterium, Beispiel 4.	90

Anhang A

Tabelle der approximierten Fekete-Punkte

Die x und y -Koordinate bezogen auf Standard-Dreieck \mathbb{T} (2.1):

N=2 :

-1.0000000000	-1.0000000000
0.0000000000	-1.0000000000
1.0000000000	-1.0000000000
-1.0000000000	0.0000000000
0.0000000000	0.0000000000
-1.0000000000	1.0000000000

N=3:

-1.0000000000	-1.0000000000
-0.4472135954	-1.0000000000
0.4472135954	-1.0000000000
1.0000000000	-1.0000000000
-1.0000000000	-0.4472135954
-0.3333333333	-0.3333333333
0.4472135954	-0.4472135954
-1.0000000000	0.4472135954
-0.4472135954	0.4472135954
-1.0000000000	1.0000000000

N=6:

-1.0000000000	-1.0000000000
-0.8302291554	-1.0000000000
-0.4688697196	-1.0000000000
0.0000000000	-1.0000000000
0.4688697196	-1.0000000000
0.8302291554	-1.0000000000
1.0000000000	-1.0000000000
-1.0000000000	-0.8302291554
-0.7873290632	-0.7873290632
-0.3674604082	-0.7656381658
0.1330985740	-0.7656381658

0.5746581264	-0.7873290632
0.8302291554	-0.8302291554
-1.0000000000	-0.4688697196
-0.7656381658	-0.3674604082
-0.3333333333	-0.3333333333
0.1330985740	-0.3674604082
0.4688697196	-0.4688697196
-1.0000000000	0.0000000000
-0.7656381658	0.1330985740
-0.3674604082	0.1330985740
0.0000000000	0.0000000000
-1.0000000000	0.4688697196
-0.7873290632	0.5746581264
-0.4688697196	0.4688697196
-1.0000000000	0.8302291554
-0.8302291554	0.8302291554
-1.0000000000	1.0000000000

N=9:

-1.0000000000	-1.0000000000
-0.9195339860	-1.0000000000
-0.7387741816	-1.0000000000
-0.4779256080	-1.0000000000
-0.1652794130	-1.0000000000
0.1652794130	-1.0000000000
0.4779256080	-1.0000000000
0.7387741816	-1.0000000000
0.9195339860	-1.0000000000
1.0000000000	-1.0000000000
-1.0000000000	-0.9195339860
-0.9021308608	-0.9021308608
-0.6912196112	-0.8896483842
-0.3979515780	-0.8822870242
-0.0600824712	-0.8798350576
0.2802386022	-0.8822870242
0.5808679954	-0.8896483842
0.8042617216	-0.9021308608
0.9195339860	-0.9195339860
-1.0000000000	-0.7387741816
-0.8896483842	-0.6912196112
-0.6591363598	-0.6591363598
-0.3495130200	-0.6431324824
-0.0073544976	-0.6431324824
0.3182727196	-0.6591363598

0.5808679954	-0.6912196112
0.7387741816	-0.7387741816
-1.0000000000	-0.4779256080
-0.8822870242	-0.3979515780
-0.6431324824	-0.3495130200
-0.3333333333	-0.3333333333
-0.0073544976	-0.3495130200
0.2802386022	-0.3979515780
0.4779256080	-0.4779256080
-1.0000000000	-0.1652794130
-0.8798350576	-0.0600824712
-0.6431324824	-0.0073544976
-0.3495130200	-0.0073544976
-0.0600824712	-0.0600824712
0.1652794130	-0.1652794130
-1.0000000000	0.1652794130
-0.8822870242	0.2802386022
-0.6591363598	0.3182727196
-0.3979515780	0.2802386022
-0.1652794130	0.1652794130
-1.0000000000	0.4779256080
-0.8896483842	0.5808679954
-0.6912196112	0.5808679954
-0.4779256080	0.4779256080
-1.0000000000	0.7387741816
-0.9021308608	0.8042617216
-0.7387741816	0.7387741816
-1.0000000000	0.9195339860
-0.9195339860	0.9195339860
-1.0000000000	1.0000000000

N=12:

-1.0000000000	-1.0000000000
-0.9533098466	-1.0000000000
-0.8463475646	-1.0000000000
-0.6861884690	-1.0000000000
-0.4829098210	-1.0000000000
-0.2492869302	-1.0000000000
0.0000000000	-1.0000000000
0.2492869302	-1.0000000000
0.4829098210	-1.0000000000
0.6861884690	-1.0000000000
0.8463475646	-1.0000000000
0.9533098466	-1.0000000000

1.0000000000	-1.0000000000
-1.0000000000	-0.9533098466
-0.9326048528	-0.9326048528
-0.7586347292	-0.9474709564
-0.5021440620	-0.9499754770
-0.4250356576	-0.8502497696
-0.1856301448	-0.9395364342
0.1251665790	-0.9395364342
0.2752854272	-0.8502497696
0.4521195390	-0.9499754770
0.7061056856	-0.9474709564
0.8652097056	-0.9326048528
0.9533098466	-0.9533098466
-1.0000000000	-0.8463475646
-0.9474709564	-0.7586347292
-0.8385226450	-0.8385226450
-0.6605731084	-0.8225925648
-0.2324962484	-0.7820061420
-0.1988883478	-0.6022233046
0.0145023904	-0.7820061420
0.4831656732	-0.8225925648
0.6770452900	-0.8385226450
0.7061056856	-0.7586347292
0.8463475646	-0.8463475646
-1.0000000000	-0.6861884690
-0.9499754770	-0.5021440620
-0.8225925648	-0.6605731084
-0.6818331042	-0.5091364040
-0.5091364040	-0.6818331042
0.1909695082	-0.6818331042
0.1909695082	-0.5091364040
0.4521195390	-0.5021440620
0.4831656732	-0.6605731084
0.6861884690	-0.6861884690
-1.0000000000	-0.4829098210
-0.8502497696	-0.4250356576
-0.7820061420	-0.2324962484
-0.4763189598	-0.4763189598
-0.3333333333	-0.3333333333
-0.0473620804	-0.4763189598
0.0145023904	-0.2324962484
0.2752854272	-0.4250356576
0.4829098210	-0.4829098210
-1.0000000000	-0.2492869302

-0.9395364342	-0.1856301448
-0.6818331042	0.1909695082
-0.6022233046	-0.1988883478
-0.5091364040	0.1909695082
-0.1988883478	-0.1988883476
0.1251665790	-0.1856301448
0.2492869302	-0.2492869302
-1.0000000000	0.0000000000
-0.9395364342	0.1251665790
-0.7820061420	0.0145023904
-0.4763189598	-0.0473620804
-0.2324962484	0.0145023904
-0.1856301448	0.1251665790
0.0000000000	0.0000000000
-1.0000000000	0.2492869302
-0.9499754770	0.4521195390
-0.8502497696	0.2752854272
-0.5021440620	0.4521195390
-0.4250356576	0.2752854272
-0.2492869302	0.2492869302
-1.0000000000	0.4829098210
-0.8385226450	0.6770452900
-0.8225925648	0.4831656732
-0.6605731084	0.4831656732
-0.4829098210	0.4829098210
-1.0000000000	0.6861884690
-0.9474709564	0.7061056856
-0.7586347292	0.7061056856
-0.6861884690	0.6861884690
-1.0000000000	0.8463475646
-0.9326048528	0.8652097056
-0.8463475646	0.8463475646
-1.0000000000	0.9533098466
-0.9533098466	0.9533098466
-1.0000000000	1.0000000000

Literaturverzeichnis

- [1] Åke Björck: *Numerical Methods For Least Squares Problems*. Philadelphia: SIAM, 1996. -ISBN 978-0-898-71360-2.
- [2] Roland W. Freund; Ronald W. Hoppe: *Stoer/Bulirsch: Numerische Mathematik 1*. 10. neu bearb. Aufl. 2007. Berlin: Springer DE, 2007. -ISBN 978-3-540-45389-5.
- [3] L. Bos, M.A. Taylor ; B.A. Wingate: *Tensor product Gauss-Lobatto points are Fekete points for the cube*. Math. Comp. 2000. V.70, Number 236, Pages 1543-1547.
- [4] M.A. Taylor; B. A. Wingate: *The Fekete collocation points for triangular spectral elements*. SIAM J. on Num. Anal., 1998.
- [5] Richard Pasquetti; Francesca Rapetti: *Spectral element methods on triangles and quadrilaterals: comparisons and applications*. J. Comput. Phys. ,V.198, Nr.1 (July 2004), Pages 349-362.
- [6] M.A. Taylor; B. A. Wingate: *The Natural Function Space For Triangular And Tetrahedral Spectral Elements*. SIAM J. Num. Anal., 1998.
- [7] Wilhelm Heinrichs: *An adaptive least-squares spectral collocation method with triangular elements for the incompressible Navier-Stokes equations*. J. Eng. Math., 2006, Volume 56, Issue 3, 337-350.
- [8] E.D. Mercerat; J.P. Vilotte; F. J. Sánchez-Sesma: *Triangular Spectral Element simulation of two-dimensional elastic wave propagation using unstructured triangular grids*. Geophys. J. Int., 2006, V.166, Issue 2, 679-698 .
- [9] J.P. Pontaza: *A spectral element least-squares formulation for incompressible Navier-Stokes flows using triangular nodal elements*. J. of Computational Physics, V.221,2007, 649-665.
- [10] M.A. Taylor; B. A. Wingate; R. E. Vincent: *An algorithm for computing fekete points in the triangle* . SIAM J. Numer. Anal. ,V.38, 2000,1707-1720.
- [11] Hans-Görg Roos; Martin Stynes; Lutz Tobiska: *Robust Numerical Methods for Singularly Perturbed Differential Equations - Convection-diffusion-reaction and*

- Flow Problems*. Berlin, Heidelberg: Springer, 2008. ISBN 978-3-540-34467-4.
- [12] Gerhard Starke: *Numerik partieller Differenzialgleichungen*. Skript zur Vorlesung, Leibniz Universität Hannover, Wintersemester 2012/13.
- [13] Wilhelm Heinrichs: *Numerik partieller Differenzialgleichungen*. Skript zur Vorlesung, Universität Duisburg-Essen, Campus Essen, Wintersemester 2010/11.
- [14] R. Verfürth: *Adaptive Finite Element Methods*. Lecture Notes, Fakultät für Mathematik, Ruhr-Universität Bochum, Winter Term 2011/12.
- [15] Jan Kippar: *Spektrale Least-Squares Verfahren für die Konvektions-Diffusions-Gleichung*. Universität Duisburg-Essen, Diplomarbeit, 2009.
- [16] Cleve Moler: *Numerical Computing with MATLAB*, Chapter 5. MathWorks, <http://www.mathworks.de/moler/chapters.html>, 2004.
- [17] K. W. Morton: *Numerical Solution of Convection-Diffusion Problems*. 1995. Aufl.. London: Chapman and Hall, 1996.
- [18] Ronald D. Henderson: *Adaptive spectral element methods for turbulence and transition*. In: Barth, Deconinck (Hg.) 1999-High-Order Methods for Computational Physics. 225-324
- [19] C. Mavriplis: *Nonconforming Discretizations and a Posteriori Error Estimators for Adaptive Spectral Element Techniques*. Ph.D. Thesis, MIT, 1989.
- [20] Bengt Fornberg: *A Practical Guide to Pseudospectral Methods*. Cambridge: Cambridge University Press, 1998.
- [21] Jörg Frochte: *Ein Splitting-Algorithmus höherer Ordnung für die Navier-Stokes-Gleichung auf der Basis der Finite-Element-Methode*. Dissertation, Universität Duisburg-Essen, 2005.
- [22] Claudio G. Canuto; M. Yousuff Hussaini; Alfio Quarteroni; Thomas A. Zang: *Spectral Methods : Evolution to Complex Geometries and Applications to Fluid Dynamics*. Berlin, Heidelberg: Springer, 2007.
- [23] Grigory I. Shishkin; Lidia P. Shishkina: *Difference Methods for Singular Perturbation Problems*. 1. Aufl.. Boca Raton, Fla: CRC Press, 2008.
- [24] Hans-Görg Roos; Martin Stynes; Lutz Tobiska: *Numerical Methods for Singularly Perturbed Differential Equations : Convection-Diffusion and Flow Problems*. 1.

Aufl.. Berlin, Heidelberg: Springer, 1996.

- [25] Christian Grossmann; Hans-Görg Roos: *Numerical Treatment of Partial Differential Equations*. Berlin, Heidelberg: Springer, 2007.
- [26] Bruno D. Welfert: *Generation Of Pseudospectral Differentiation Matrices I*. SIAM J. NUMER.ANAL., Vol. 34, No. 4, pp. 1640-1657, August 1997.
- [27] Richard Baltensperger; Manfred R. Trummer: *Spectral Differencing with a Twist*. SIAM J. Sci. Comput. 24 (2003), pp. 1465-1487.

Index

- Least-Squares Verfahren, 11
- a-posteriori Fehlerschätzer, 8, 38
 - Ableitungsmatrix, 24
- Adaptive Verfeinerungsstrategien , 38
 - adaptiven Verfeinerung, 38
 - affine Transformation, 23
 - Anhang, 98
 - Ausblick, 92
 - Backslash-Befehl, 43
 - blau, 42
 - Boundary-Layer, 7, 21
 - CHOLMOD, 12
 - Die skalare Formulierung, 35
 - Differentialoperator, 31, 32, 36
 - Dirac-Funktion, 11
 - Diskretisierung, 22
 - Dreieckselement, 22
 - Dubiner, 15, 17, 24, 39
 - Eckenlayer, 81
 - Extrapolation, 8
 - Fekete-Punkte, 15–17, 23, 98
 - Formulierung als System, 7
 - Friedrichs-Keller-Triangulierung, 7, 22
 - Gauß-Lobatto, 16
 - gewichtete Norm, 43
 - grün, 42
 - Gradienten, 8, 38, 52, 63, 76, 87, 92
 - Grundgebiet, 7
 - hängende Knoten, 42
 - Hutlayer, 68
 - Interface, 25, 29, 30, 33, 36
 - Jacobi-Determinante, 24, 39
 - Jacobi-Polynom, 15
 - Kippar, 43, 84
 - Kollokation, 10, 11
 - Kollokationspunkte, 28
 - Konturlinie, 52
 - Konvektions-Diffusions-Gleichung, 19, 20
 - Kronecker-Delta, 16
 - Least-Squares-Problem, 12
 - Lebesgue-Konstante, 17
 - Markierungsstrategie, 38
 - moderater Boundary-Layer, 44
 - Monome, 15
 - Moore-Penrose-Inverse, 13
 - Nicht-Tensor-Produkt-Gebiet, 15
 - Normalableitung, 35, 36
 - Normalengleichung, 12
 - Numerische Ergebnisse, 43
 - Nummerierung, 35
 - partielle Differentialgleichung, 7
 - Penrose Bedingungen, 13
 - Projektion, 39
 - Pseudo-spektrale Methode, 10, 11
 - Pseudoinverse, 12, 13
 - Quadratur-Gewichte, 18
 - Quadraturformel, 18, 39
 - Randmatrix, 29, 36
 - Randoperator, 35, 103
 - Randschicht, 21
 - Regression, 40
 - rot, 42
 - singulär-gestört, 7, 19, 21
 - singuläre Störung, 57
 - Singulärwertzerlegung, 12

- skalare Formulierung, 7
- Skalare vs. System, 49, 62, 74, 86
 - Skalarprodukt, 16
- Spektrale Kollokation, 11
- Spektrum, 8, 40, 41, 55, 66, 79, 90, 92
- Störungsparameter, 19, 73, 92
- Standard-Dreieck, 15, 16, 39
 - SuperLU, 12

- Testfunktion, 10, 11
 - Toleranz, 38, 41
- Transformationsmatrix, 24

- UMFPACK, 12

- Vandermonde, 15, 17, 24
- Verfeinerungsregel, 38, 42
- Verschärfter Boundary-Layer, 57
 - verzerrtes Produkt, 15
- Viereckszerlegung, 7, 47, 58, 82

- Yale sparse matrix Package, 12
 - YSMP, 12