

A posteriori Finite-Elemente-Analyse
zur adaptiven Ortsdiskretisierung
in der Elastoviskoplastizität

Dissertation

zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften
(Dr. rer. nat.)
der Technischen Fakultät
der Christian-Albrechts-Universität zu Kiel

Roland Klose

Kiel
2003

1. Gutachter : Prof. Dr. Carsten Carstensen

2. Gutachter : Prof. Dr. Dr. h.c. Wolfgang Hackbusch

Datum der mündlichen Prüfung: 12.5.2003

Zusammenfassung

Ein Modell zur Beschreibung von elastoviskoplastischem Materialverhalten ist die Perzyna-Viskoplastizität. In der betrachteten Problemstellung wird das Prandtl-Reuß-Fließgesetz mit von-Mises-Fließfunktionen verwendet, die perfekte Plastizität, isotrope Verfestigung und lineare kinematische Verfestigung beinhalten. Die mathematische Modellierung führt auf eine zeitabhängige Variationsungleichung. Eine analytische Lösung dieses Problems ist in den meisten Fällen nicht möglich, so daß man auf eine numerische Approximation angewiesen ist. In dieser Arbeit wird hierzu ein numerisches Verfahren entwickelt, das auf einem Zeitdiskretisierungsverfahren und der Finite-Elemente-Methode beruht. Die Genauigkeit der Finite-Elemente-Approximation läßt sich durch eine Verfeinerung der Elementgröße sowie durch eine Erhöhung des Polynomgrades verbessern.

Zur Kontrolle des Ortsdiskretisierungsfehlers in einem Zeitschritt werden a posteriori Fehlerabschätzungen bewiesen, die eine garantierte obere Schranke für den Spannungsfehler liefern. Die in der a posteriori Fehleranalyse auftretenden Konstanten werden bestimmt, so daß sich auf der Basis von residualen Schätzern berechenbare Fehlerschranken ergeben, die eine Aussage über die in der Finite-Elemente-Rechnung erzielte Genauigkeit erlauben.

Im Rahmen dieser Arbeit wurden eine Finite-Elemente-Methode für die Lösung des elastoviskoplastischen Problems sowie verschiedene residuenbasierte a posteriori Fehlerschätzer implementiert. Zur Verbesserung des Approximationsverhaltens der Finite-Elemente-Methode werden automatisierte adaptive Netzverfeinerungen auf der Basis der residualen a posteriori Fehlerschätzer realisiert. Anhand von numerische Beispielen werden die Eigenschaften der Fehlerschätzer in adaptiven Algorithmen untersucht und Aussagen über die Genauigkeit der Abschätzungen getroffen.

Danksagung

Ich danke Herrn Prof. Dr. Carsten Carstensen für den Vorschlag zu dieser Arbeit, für seine persönliche Betreuung und für viele wertvolle Diskussionen.

Diese Arbeit wurde im Rahmen des Graduiertenkollegs „Effiziente Algorithmen und Mehrskalenmethoden“ durch die DFG gefördert. Ich bedanke mich für die Unterstützung durch das Graduiertenkolleg bei Prof. Dr. Dr. hc. W. Hackbusch und Prof. Dr. A. Srivastav.

Herrn Dipl. Phys. Axel Hecht und Herrn Dr. habil Stefan Funken danke ich für hilfreiche Diskussionen und Problembewältigungen zum Thema Matlab.

Ich bedanke mich bei Herrn Dr. Jochen Albrety, Herrn Dr. Sören Bartels, Herrn Dipl.

Math. Dirk Praetorius, Herrn Dr. habil Andreas Prohl, Herrn Dr. Jan Valdmann und Frau Dr. Kerstin Weinberg, mit denen ich hilfreiche Diskussionen führen konnte.

Herrn Dipl. Math. Carsten Boecker danke ich, daß er mir Software zu Finite-Elemente-Methoden höherer Ordnung in der Elastizität zur Verfügung stellte.

Bei Frau Gudrun Thiel bedanke ich mich für ihre freundliche Hilfe bei Organisationsproblemen.

Teile dieser Arbeit sind während meines sechsmonatigen Forschungsaufenthalts an der Technischen Universität Wien entstanden. Ich bedanke mich für die Möglichkeit, die dortigen Computerressourcen zu nutzen.

Kiel, Mai 2003
Roland Klose

Inhaltsverzeichnis

Zusammenfassung und Danksagung	iii
1 Einleitung	1
2 Das elastoviskoplastische Modellproblem	7
2.1 Kontinuumsmechanik und lineare Elastizität	7
2.2 Regularisierte Viskoplastizität	9
2.3 Elastoviskoplastisches Modellproblem	13
2.4 Zeitdiskretisierung und Existenz von schwachen Lösungen	15
3 Numerische Lösung des Modellproblems	19
3.1 Analytische Darstellung der verallgemeinerten Spannung	19
3.1.1 Perfekte Plastizität	21
3.1.2 Isotrope Verfestigung	22
3.1.3 Kinematische Verfestigung	24
3.2 Finite-Elemente-Lösung der Gleichgewichtsbedingung	26
3.2.1 Triangulierung und Finite-Elemente-Raum	27
3.2.2 Ortsdiskretes Modellproblem	30
3.2.3 Verallgemeinerte Randbedingungen	31
3.2.4 Iterative Lösung mit dem Newton-Verfahren	33
3.3 Adaptiver Algorithmus	37
3.3.1 Verfeinerung der Finite-Elemente-Diskretisierung	39
3.3.2 Interpolation von Anfangswerten	43
4 A posteriori Fehlerabschätzung in einem Zeitschritt	47
4.1 Garantierte obere Schranken für den Spannungsfehler	48
4.1.1 Überblick	48
4.1.2 Grundlegende Ergebnisse der a posteriori Fehleranalyse	51
4.2 Der Residuenschätzer	66
4.2.1 Zuverlässigkeit der Fehlerabschätzung	66
4.2.2 Effizienz der Fehlerabschätzung	70
4.3 Die Methode der äquilibrierten Residuen	71
4.3.1 Zuverlässigkeit der Fehlerabschätzung	71
4.3.2 Lösung der lokalen Probleme	72
4.3.3 Konstruktion der Momente der äquilibrierten Randkräfte	77
4.3.4 Effizienz der Fehlerabschätzung	83

4.4	Die Methode der Patch-Residuen	89
4.4.1	Zuverlässigkeit der Fehlerabschätzung	89
4.4.2	Lösung der lokalen Probleme	91
4.4.3	Effizienz der Fehlerabschätzung	92
5	Numerische Ergebnisse	95
5.1	Definition von Testproblemen	95
5.2	Der Effektivitätsindex in Abhängigkeit von Materialparametern	99
5.3	Experimentelle Konvergenzraten	105
5.4	Bewertung der Resultate	125
6	Zusammenfassung der Ergebnisse	127
A	Implementierung	129
A.1	Hauptprogramm	129
A.2	Finite-Elemente-Methode	134
A.3	Interpolation von Anfangswerten	144
A.4	Spannungsberechnung	155
A.5	Fehlerschätzer η_R : Residuenschätzer	159
A.6	Fehlerschätzer η_{EQ} : Methode der äquilibrierten Residuen	177
A.7	Fehlerschätzer η_L : Patch-Residuen-Schätzer	185
	Literaturverzeichnis	191
	Veröffentlichungsliste	197

Kapitel 1

Einleitung

In der Werkstoffoptimierung und -charakterisierung werden heute vielfach numerische Simulationen angewendet. Dazu beigetragen hat der rasche Anstieg der Computerleistung in den letzten Jahren. Numerische Simulationen unterstützen werkstoffmechanische Experimente, wodurch sich Entwicklungszeiten und Kosten sparen lassen.

Die Finite-Elemente-Methode (FEM) ist das dominierende numerische Verfahren in der Spannungs- und Verformungsberechnung elastoplastischer Probleme geworden. Basierend auf der Variationsformulierung der zugrundeliegenden partiellen Differentialgleichung wird eine Lösung in einem endlichdimensionalen Raum gesucht. Das System der Basisfunktionen des Finite-Elemente-Raumes besteht aus global stetigen Funktionen, die auf jedem Element der Finite-Elemente-Triangulierung Polynome niedriger Ordnung sind.

Bei der numerischen Lösung partieller Differentialgleichungen treten Approximationsfehler auf. Fehlerquellen sind

- der Zeitdiskretisierungsfehler,
- der Ortsdiskretisierungsfehler,
- Abbruchfehler bei iterativen Verfahren,
- Interpolationsfehler bei Netzverfeinerungen.

Gegenstand dieser Arbeit ist die Kontrolle des Ortsdiskretisierungsfehlers aufgrund der Finite-Elemente-Approximation.

Eine Verbesserung der Approximationsqualität läßt sich durch eine Erhöhung der Anzahl der Freiheitsgrade erreichen. Hierfür gibt es verschiedene Möglichkeiten.

- h-Version der FEM: Die Anzahl der Freiheitsgrade wird durch eine Verringerung der Elementgröße h erhöht.
- p-Version der FEM: Die Erhöhung der Anzahl der Freiheitsgrade wird durch eine Erhöhung der Polynomgrade der FE-Basisfunktionen erreicht.
- hp-Version der FEM: Die Anzahl der Freiheitsgrade wird erhöht, indem eine geeignete Kombination aus einer Verringerung der Größe von Elementen und der Erhöhung des Polynomgrades von Ansatzfunktionen durchgeführt wird.

Um die Güte der numerischen Approximation abzuschätzen und um das Konvergenzverhalten zu steuern, wurden intensive Forschungen auf dem Gebiet der a posteriori Fehlerschätzer unternommen. Ein a posteriori Fehlerschätzer η hängt von der berechneten Finite-Elemente-Lösung \mathbf{u}_h sowie weiteren gegebenen Größen ab. Der Fehler $\mathbf{e} = \mathbf{u} - \mathbf{u}_h$ ist die Differenz aus der exakten Lösung \mathbf{u} und der Finite-Elemente-Lösung \mathbf{u}_h . Der a posteriori Fehlerschätzer η sollte die Abschätzungen

$$|||\mathbf{e}||| \leq C_1 \eta \quad \text{und} \quad \eta \leq C_2 |||\mathbf{e}||| \quad (1.1)$$

erfüllen, das heißt die Größe η soll bis auf Konstanten C_1 bzw. C_2 eine obere bzw. untere Grenze für den Fehler in einer geeigneten Norm $|||\cdot|||$ darstellen. A posteriori Fehlerschätzern können Beiträge auf einem Element T einer Triangulierung \mathcal{T} zugewiesen werden, d.h. es gilt

$$\eta = \left(\sum_{T \in \mathcal{T}} \eta_T^2 \right)^{1/2}. \quad (1.2)$$

Die Qualität eines Fehlerschätzers wird durch den globalen bzw. lokalen Effektivitätsindex

$$C_{Eff} = \frac{C_1 \eta}{|||\mathbf{e}|||} \quad \text{bzw.} \quad C_{Eff,T} = \frac{C_1 \eta_T}{|||\mathbf{e}|||_T} \quad (1.3)$$

beurteilt. Optimal wäre ein globaler Effektivitätsindex möglichst nahe eins, in der Praxis werden aber auch höhere Werte als akzeptabel betrachtet.

Um die Zuverlässigkeit der numerischen Lösung zu kennen, ist eine garantierte obere Schranke des Ortsdiskretisierungsfehlers nötig. Dies erfordert Fehlerschätzer, die eine berechenbare obere Schranke für den Gesamtfehler bilden. Das bedeutet, der Wert der Konstante C_1 in (1.1) sollte berechnet werden können. Ist diese Voraussetzung erfüllt, können a posteriori Fehlerschätzer als Abbruchkriterium in Finite-Elemente-Rechnungen eingesetzt werden. Einen Fehlerschätzer der eine garantierte obere Schranke für den Gesamtfehler bildet, nennt man zuverlässig.

Um eine Lösung mit vorgegebener Genauigkeit mit möglichst geringem Rechenaufwand zu erzielen, ist eine optimale Verfeinerungsstrategie nötig. Dies führt auf die Konstruktion adaptiver Verfahren, bei denen eine Verfeinerung der Finite-Elemente-Triangulierung nur lokal durchgeführt wird. A posteriori Fehlerschätzer werden dazu eingesetzt, diese adaptiven Verfahren zu steuern. Der Schätzer dient hierbei als Verfeinerungsindikator. Der Verfeinerungsindikator ist der Elementbeitrag η_T in (1.2). Um zu garantieren, daß der Fehlerschätzer mit derselben Rate konvergiert wie der Fehler, ist es erforderlich, daß der Fehlerschätzer eine untere Schranke des Fehlers ist, daß also die rechte Seite von (1.1) erfüllt ist. Diese Eigenschaft nennt man Effizienz.

Man kann a posteriori Fehlerschätzer in zwei Klassen einteilen, explizite und implizite Fehlerschätzer. Explizite Fehlerschätzer erlauben eine direkte Berechnung der Fehlerschranke aus vorhandenen Daten. Dies sind die numerische Lösung sowie Randdaten. Hier wird ein residuenbasierter Schätzer nach I. Babuška und W. Rheinboldt behandelt [12, 13].

Implizite Fehlerschätzer erfordern die Lösung von kleinen lokalen Problemen. Ein Fehlerschätzer basierend auf lokalen Problemen auf Elementen findet sich erstmals bei D. Ladeveze und D. Leguillon [43] sowie R. Bank und A. Weiser [17] und ist

bekannt als die Methode der äquilibrierten Residuen. Diese Methode wurde weiterentwickelt von M. Ainsworth und J.T. Oden [1, 2, 3]. In dem Buch „A Posteriori Error Estimation in Finite Element Analysis“ [4] findet sich eine Version des äquilibrierten Residuenschätzers, auf die hier aufgebaut wird. Eine zweite Methode zur a posteriori Fehlerschätzung durch Lösen lokaler Probleme stammt von I. Babuška und W. Rheinboldt und wurde verfeinert von C. Carstensen und S.A. Funken [26]. Im Gegensatz zur Methode der äquilibrierten Residuen werden hier für jeden Knoten auf dem zugehörigen Patch lokale Probleme gelöst.

Arbeiten zu a posteriori Fehlerschätzern in der Plastizität existieren von verschiedenen Autoren. Residuale Fehlerschätzer wurden von J. Albery, C. Carstensen und D. Zarrabi in den Arbeiten [9, 24, 23] veröffentlicht. In [49] wird von S. Ohnuma die Methode der äquilibrierten Residuen für die Fehlerabschätzung bei der Berechnung des elastischen Prädiktors veröffentlicht. Weitere Arbeiten zur a posteriori Fehlerabschätzung in der Plastizität stammen von C. Johnson und P. Hansbo [42], sowie zur Fehlerabschätzung über Dualitätstechniken von R. Rannacher und F.-T. Suttmeier [53, 55, 54] sowie S.I. Repin und L.S. Xanthis [56].

In dieser Arbeit wird das Modell der regularisierten Viskoplastizität nach Perzyna betrachtet. Die mathematische Modellierung führt auf eine zeitabhängige Variationsungleichung, die in dem Zeitintervall $[0, t_G]$ gestellt ist. Gegeben sind Funktionenräume X, X_M für die Verschiebung, S für die Spannung und M für interne Verfestigungsvariablen. Der Raum der verallgemeinerten Spannungen ist definiert als $\mathbb{T} = S \times M$. Weiterhin sind Bilinearformen $b : X \times S \rightarrow \mathbb{R}$ und $A : \mathbb{T} \times \mathbb{T} \rightarrow \mathbb{R}$ und ein Funktional $\psi_\varrho : S \rightarrow \mathbb{R}$ gegeben. Das Funktional ψ_ϱ ist abhängig von einem Regularisierungsparameter ϱ , der die Bedeutung einer Viskosität hat. Für den Grenzfall $\varrho \rightarrow 0$ erhält man ein Modell der Plastizität. Geben ist außerdem ein von Oberflächenkräften und einer Volumenkraft abhängiges Lastfunktional $l \in H^1(0, t_G; X')$ mit $l(0) = 0$.

Das zu lösende Problem besteht darin, die Verschiebung $\mathbf{u}(t)$ und die verallgemeinerte Spannung $\Sigma(t) = (\boldsymbol{\sigma}(t), \boldsymbol{\alpha}(t))$ zu finden, wobei $(\mathbf{u}, \Sigma) = (\mathbf{u}, \boldsymbol{\sigma}, \boldsymbol{\alpha}) : [0, t_G] \rightarrow X_M \times \mathbb{T}$ mit $\mathbf{u}(0) = 0, \Sigma(0) = 0$, so daß für fast alle $t \in [0, t_G]$,

$$b(\mathbf{v}, \boldsymbol{\sigma}(t)) = \langle \mathbf{l}(t), \mathbf{v} \rangle \quad \forall \mathbf{v} \in X_0, \quad (1.4)$$

$$\psi_\varrho(\mathbf{T}) - \psi_\varrho(\Sigma(t)) + A(\dot{\Sigma}(t), \mathbf{T} - \Sigma(t)) + b(\dot{\mathbf{u}}(t), \boldsymbol{\tau} - \boldsymbol{\sigma}(t)) \geq 0 \quad \forall \mathbf{T} \in \mathbb{T}. \quad (1.5)$$

Gleichung (1.4) beschreibt das quasistatische Kräftegleichgewicht und Gleichung (1.5) modelliert das Fließgesetz plastischer Materialien. Das betrachtete Problem verwendet das Prandtl-Reuß-Fließgesetz mit von-Mises-Fließfunktionen, die perfekte Plastizität, isotrope Verfestigung und lineare kinematische Verfestigung beinhalten. In Kapitel 2 wird das elastoviskoplastische Problem ausführlich dargestellt.

Die numerische Lösung des elastoviskoplastischen Problems erfolgt in dieser Arbeit auf der Grundlage einer Zeitdiskretisierung nach der verallgemeinerten Mittelpunktsregel und einer Finite-Elemente-Diskretisierung.

In jedem Zeitschritt des numerischen Lösungsverfahrens ist eine nichtlineare Variationsgleichung mit einer Finite-Elemente-Methode zu lösen. Eine Kontrolle des Ortsdiskretisierungsfehlers erfolgt durch einen a posteriori Fehlerschätzer. In jedem

Zeitschritt ist ein Abbruchkriterium für die erzielte Genauigkeit zu erfüllen. Andernfalls ist die Finite-Elemente-Rechnung in diesem Zeitschritt mit erhöhter Genauigkeit zu wiederholen. Der a posteriori Fehlerschätzer wird zur Steuerung von adaptiver Algorithmen eingesetzt werden, mit dem Ziel, eine vorgegebene Genauigkeit mit möglichst wenig Freiheitsgraden zu erzielen. Gerade in zeitabhängigen Problemen ist es von Bedeutung, daß eine Finite-Elemente-Methode mit optimaler Verfeinerungsstrategie verwendet wird. Durch die Vielzahl an Zeitschritten vervielfacht sich der mögliche Zeitgewinn. Wie eine optimale Verfeinerungsstrategie zu erfolgen hat, ist zunächst nicht klar. In dieser Arbeit werden h -, p - und hp -Verfeinerungen untersucht. Die genannten Möglichkeiten zur Erhöhung der Freiheitsgrade unterscheiden sich in der Konvergenzrate des Ortsdiskretisierungsfehlers. Eine Beschreibung der Konvergenztheorie für finite Elemente wird in [20, 35, 52, 57] gegeben. Die Konvergenz hängt von der Regularität des Problems ab, von der Elementgröße h , von dem verwendeten Polynomgrad p für die Ansatzfunktionen des Finite-Elemente-Raumes sowie der verwendeten Fehlernorm. Das Konvergenzverhalten ist bestimmt durch die Interpolationseigenschaften der gewählten finiten Elemente. Für h -Verfeinerung ist die Konvergenzrate algebraisch. Für p -Verfeinerung kann die Konvergenz exponentiell sein, bei Auftreten von Singularitäten nimmt der Fehler allerdings auch hier im asymptotischen Bereich nur algebraisch ab [14, 16]. Die hp -Methode, eine Kombination aus h - und p -Methode weist bei geeignet konstruierten Netzen asymptotische exponentielle Konvergenz auf [10, 11, 16, 15].

In der Kontinuumsmechanik wurden p - und hp -Methoden bisher hauptsächlich in linearen Problemen angewendet. Die p -Version der Finiten-Elemente in der Elastoplastizität ist in den Arbeiten [32, 31, 40, 41, 61] untersucht worden.

In der bisherigen Forschung fehlen a posteriori Fehlerabschätzungen, die für die Finite-Elemente-Approximation des elastoviskoplastischen Problems mit finiten Elementen höherer Ordnung garantierte obere Fehlerschranken für residuenbasierte Fehlerschätzer liefern.

In dieser Arbeit werden a posteriori Fehlerabschätzungen für den Ortsdiskretisierungsfehler in einem Zeitschritt bewiesen. Dabei wird vorausgesetzt, daß der exakten und der numerischen Lösung die selben Anfangswerte zugrundeliegen. Folgende Anforderungen werden bei der a posteriori Fehleranalyse berücksichtigt.

- Es werden garantierte obere Fehlerschranken bewiesen. Da die Konstante C_1 in der Abschätzung (1.1) bestimmt wird, erhält man eine berechenbare obere Schranke für den Spannungsfehler in einem Zeitschritt.
- Die a posteriori Fehleranalyse läßt finite Elemente höherer Ordnung zu. Damit ist eine zuverlässige a posteriori Fehlerabschätzung bewiesen, die in allen eingangs genannten Verfeinerungsstrategien Gültigkeit besitzt.

Die Arbeit ist wie folgt aufgebaut. Kapitel 2 definiert das elastoviskoplastische Problem.

Kapitel 3 beschreibt die numerische Lösung des Problems. Teile dieses Kapitels sind in [28] veröffentlicht.

Gegenstand von Kapitel 4 ist die a posteriori Fehlerabschätzung des Spannungsfehlers. Kapitel 4 zeigt als ein wesentliches Ergebnis dieser Arbeit, daß in der Elasto-

viskoplastizität eine a posteriori Fehlerabschätzung auf der Grundlage von residuenbasierten Schätzern möglich ist, die explizit berechenbare Fehlerschranken liefert und in Finite-Elemente-Methoden höherer Ordnung eingesetzt werden kann. Die Abschätzung des Spannungsfehlers setzt voraus, daß eine obere Abschätzung des Verzerrungsfehlers durch den Spannungsfehler existiert. In Abschnitt 4.1 werden diese Abschätzungen für Elastoviskoplastizität mit Verfestigung bewiesen, und auftretende Konstanten werden berechnet. In den Abschnitten 4.2, 4.3 und 4.4 wird mit den Ergebnissen aus Abschnitt 4.1 die Zuverlässigkeit des Residuenschätzers, der Methode der äquilibrierten Residuen und des Patch-Residuen-Schätzers bewiesen.

Anhand von praktischen Beispielen werden in Kapitel 5 die numerischen Eigenschaften der Fehlerschätzer untersucht. Zu diesem Zweck wurde im Rahmen dieser Arbeit eine Finite-Elemente-Methode mit der Möglichkeit einer hp-Verfeinerung, sowie die genannten Fehlerschätzer implementiert.

Kapitel 2

Das elastoviskoplastische Modellproblem

Dieses Kapitel faßt kontinuumsmechanische Grundlagen zusammen und definiert ein elastoviskoplastisches Modellproblem. Die Ausführungen zur Viskoplastizität folgen [36, 9], als weitere Literatur wird auf [58, 46, 44, 37, 27, 22] verwiesen.

Der erste Abschnitt ist eine Einführung in fundamentale Begriffe der Kontinuumsmechanik. Es wird die lokale Form des quasistatischen Kräftegleichgewichts beschrieben und das Verschiebungsproblem der linearen, isotropen Elastizität angegeben. Anschließend werden der ebene Spannungszustand und der ebene Verzerrungszustand definiert, auf den sich die Modellrechnungen im weiteren Verlauf beschränken.

Der zweite Abschnitt führt das physikalische Modell der regularisierten Viskoplastizität nach Perzyna ein, und es werden das Prandtl-Reuß-Fließgesetz sowie die von-Mises-Fließfunktionen definiert.

Der dritte Abschnitt beschreibt das im weiteren Verlauf der Arbeit verwendete Modellproblem der regularisierten Viskoplastizität und gibt die Formulierung als Variationsungleichung an.

Im vierten Abschnitt ist die Zeitdiskretisierung nach der verallgemeinerten Mittelpunktsregel beschrieben. Es wird die eindeutige Existenz einer schwachen Lösung in einem Zeitschritt gezeigt.

2.1 Kontinuumsmechanik und lineare Elastizität

In der Kontinuumsmechanik werden Veränderungen in der Gestalt eines festen Körpers aufgrund von äußeren Kräften berechnet. Dabei wird angenommen, daß der Körper aus einem Kontinuum von Partikeln, den materiellen Punkten besteht, die, abhängig von der Zeit t , den Raum Ω_t einnehmen. Die Region, die der Körper zur Zeit $t = 0$ einnimmt, wird als Referenzkonfiguration Ω bezeichnet. Die Positionen der materiellen Punkte werden durch den Ortsvektor \boldsymbol{x} in der Referenzkonfiguration identifiziert.

Die Bewegung $\boldsymbol{y} = \boldsymbol{y}(\boldsymbol{x}, t)$ ist auf Ω eine injektive Abbildung der Referenzkonfiguration auf die Momentankonfiguration $\boldsymbol{y} : \overline{\Omega} \rightarrow \overline{\Omega}_t \subset \mathbb{R}^3$, mit $\det \boldsymbol{D}\boldsymbol{y} > 0$ auf Ω . Hier

ist mit $D\mathbf{y}$ die Jakobi-Matrix mit den Komponenten $(D\mathbf{y})_{ij} = \frac{\partial y_i}{\partial x_j}$ bezeichnet. Die Verschiebung $\mathbf{u}(\mathbf{x}, t)$ ist definiert durch

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{y}(\mathbf{x}, t) - \mathbf{x}.$$

Veränderungen in der Länge, der Fläche und dem Volumen durch eine Deformation werden mit Hilfe des Deformationsgradienten $\mathbf{F} = D\mathbf{y}$ ausgedrückt. Das Modellproblem beschränkt sich auf kleine Verschiebungen und kleine Verzerrungen. Die Beschränkung auf kleine Verschiebungen erlaubt es, einen linearisierten Greenschen Verzerrungstensor $\boldsymbol{\epsilon}$ zu verwenden,

$$\boldsymbol{\epsilon} = \frac{1}{2}(D\mathbf{u} + (D\mathbf{u})^T).$$

Eine infinitesimale Starrkörperverschiebung erfüllt die Bedingung $\boldsymbol{\epsilon}(\mathbf{u}) = \mathbf{0}$. Die zugehörigen Bewegungen und Deformationen haben die allgemeine Darstellung

$$\mathbf{y}(\mathbf{x}) = \mathbf{y}_0 + \boldsymbol{\omega}(\mathbf{x} - \mathbf{x}_0) \quad \text{und} \quad \mathbf{u}(\mathbf{x}) = \mathbf{u}_0 + \boldsymbol{\omega}(\mathbf{x} - \mathbf{x}_0).$$

Hierbei ist $\boldsymbol{\omega}$ ein schiefsymmetrischer Tensor.

Quasistatisches Kräftegleichgewicht

Die auf den Körper in der Momentankonfiguration Ω_t einwirkenden Kräfte sind die Volumenkraft $\mathbf{f}(\mathbf{y})$ und Oberflächenkräfte $\mathbf{g}(\mathbf{y})$, die auf den Neumannrand $\Gamma_N \subset \Gamma$ wirken, sowie Kontaktkräfte $\mathbf{t}(\mathbf{y}) \cdot \mathbf{n}(\mathbf{y})$, die von einem Teil des Körpers auf den anderen ausgeübt werden. Das Cauchy-Theorem besagt, daß sich die Kontaktkräfte als Produkt aus dem Cauchy-Spannungstensor und der Normalen ausdrücken lassen, $\mathbf{t} = \mathbf{T}\mathbf{n}$. Über die Abbildung $\mathbf{y} = \mathbf{y}(\mathbf{x}, t)$ lassen sich Größen in der Momentankonfiguration auf die Referenzkonfiguration transformieren. In der Referenzkonfiguration lautet die lokale Form des 2. Newtonschen Axioms,

$$\operatorname{div} \boldsymbol{\sigma} + \mathbf{f} = \rho \ddot{\mathbf{y}} \quad \text{in } \Omega. \quad (2.1)$$

Hier ist $\boldsymbol{\sigma} = \det \mathbf{F} \mathbf{T} \mathbf{F}^{-T}$ der 1. Piola-Kirchhoff-Spannungstensor. Im folgenden wird der quasistatische Fall betrachtet, es wird eine vernachlässigbare Beschleunigung vorausgesetzt, $\ddot{\mathbf{y}} \approx 0$. Gleichung (2.1) wird dann quasistatisches Kräftegleichgewicht genannt. In der Kontinuumsmechanik ist die Volumenkraft \mathbf{f} in Inertialsystemen gleichzusetzen mit der Gewichtskraft pro Volumen des Körpers. Die Gewichtskraft ist im Vergleich zu den Kontaktkräften vernachlässigbar, weshalb in der Kontinuumsmechanik in guter Näherung mit $\mathbf{f} = 0$ gerechnet werden kann.

Definition 2.1 (Vektorraum der symmetrischen Tensoren). *Symmetrische Tensoren zweiter Ordnung sind lineare Transformationen in \mathbb{S}_d , definiert als*

$$\mathbb{S}_d := \{ \mathbf{A} : \mathbb{R}^d \rightarrow \mathbb{R}^d \mid \mathbf{A} \text{ ist linear und } \mathbf{A} = \mathbf{A}^T \}. \quad (2.2)$$

Das ist ein Vektorraum mit Skalarprodukt $\mathbf{A} : \mathbf{A} = \operatorname{tr}(\mathbf{A}^T \mathbf{A}) = \sum_{i,j=1}^d A_{ij} A_{ij}$.

Elastische konstitutive Gleichung

Im Fall kleiner Verzerrungen ist die Beziehung zwischen dem Spannungstensor $\boldsymbol{\sigma}$ und dem Verzerrungstensor $\boldsymbol{\epsilon}$ linear,

$$\boldsymbol{\sigma} = \mathbb{C}\boldsymbol{\epsilon}.$$

Hierbei ist \mathbb{C} der Elastizitätstensor, der vom Ort aber nicht von der Zeit abhängig ist. Falls die Dichte ρ und der Elastizitätstensor \mathbb{C} des Körpers unabhängig von der Position sind, ist das Material homogen. Ein richtungsunabhängiges Material nennt man isotrop. Für homogene und isotrope Materialien sind die Komponenten des Elastizitätstensors gegeben durch

$$C_{ijkl} = \lambda\delta_{ij}\delta_{kl} + \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}),$$

wobei δ_{jk} das Kronecker-Delta ist. Mit λ und μ sind die Lamé-Konstanten bezeichnet. Die Spannungs-Dehnungs-Beziehung ist in diesem Fall gegeben durch

$$\boldsymbol{\sigma} = \lambda(\text{tr } \boldsymbol{\epsilon})\mathbf{I} + 2\mu\boldsymbol{\epsilon}.$$

In gewissen Situationen lassen sich dreidimensionale Probleme der Kontinuumsmechanik auf ein zweidimensionales Problem zurückführen. Man unterscheidet zwischen dem ebenen Verzerrungszustand und dem ebenen Spannungszustand.

Ebener Verzerrungszustand

Ein ebener Verzerrungszustand (EVZ) liegt (in der xy-Ebene) für einen Verzerrungstensor $\boldsymbol{\epsilon}$ vor, wenn für $\boldsymbol{\epsilon} : \Omega \rightarrow \mathbb{R}_{sym}^{3 \times 3}$ gilt $\epsilon_{xz} = \epsilon_{yz} = \epsilon_{zz} = 0$. Der ebene Verzerrungszustand liegt näherungsweise vor, wenn ein sehr langer Körper mit Längsachse in z-Richtung einen von z unabhängigen Querschnitt besitzt und einer von z unabhängigen Belastung ausgesetzt ist. Für diese Geometrie tritt keine Verschiebung in z-Richtung auf ($u_z \equiv 0$).

Das in Abschnitt 2.3 beschriebene elastoplastische Modellproblem beschränkt sich auf den ebenen Verzerrungszustand.

Ebener Spannungszustand

Ein ebener Spannungszustand (ESZ) liegt (in der xy-Ebene) für einen Spannungstensor $\boldsymbol{\sigma}$ vor, wenn für $\boldsymbol{\sigma} : \Omega \rightarrow \mathbb{R}_{sym}^{3 \times 3}$ gilt $\sigma_{xz} = \sigma_{yz} = \sigma_{zz} = 0$. Der ebene Spannungszustand liegt näherungsweise vor, wenn eine dünne Scheibe nur in ihrer Ebene belastet wird.

2.2 Regularisierte Viskoplastizität

Die freie Energie $F = F(\boldsymbol{\epsilon})$ und die Spannung $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{\epsilon})$ elastischer Materialien hängen nur von der Verzerrung $\boldsymbol{\epsilon}$ ab. Die freie Energie $F = F(\boldsymbol{\epsilon}, \boldsymbol{\xi})$ und die Spannung $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{\epsilon}, \boldsymbol{\xi})$ elastoplastischer Materialien sind von einer Reihe interner Variablen $\boldsymbol{\xi} = (\xi_i)_{i=1}^m$ abhängig, die Skalare oder Tensoren sind. Das Verhalten elastoplastischer

Materialien hängt von der Materialgeschichte ab. Daher ist eine Evolutionsgleichung der Form

$$\dot{\boldsymbol{\xi}}_i = \boldsymbol{\beta}_i(\boldsymbol{\epsilon}, \boldsymbol{\xi}) \quad 1 \leq i \leq m \quad (2.3)$$

erforderlich.

Während der plastischen Deformation treten interne Spannungen $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_i)_{i=1}^m$ auf, die zu den internen Variablen konjugierte Größen sind, in dem Sinne, daß

$$\boldsymbol{\alpha}_i = -\frac{\partial F}{\partial \boldsymbol{\xi}_i}, \quad i = 1, \dots, m.$$

Die Größe $\boldsymbol{\alpha}_i : \boldsymbol{\xi}_i$ stellt eine interne Dissipation dar. Aus thermodynamischen Überlegungen folgt die fundamentale Annahme der geometrisch linearisierten Elastoplastizität, die additive Aufspaltung des Verzerrungstensors $\boldsymbol{\epsilon}$ in einen elastischen Anteil $\boldsymbol{e} = \mathbb{C}^{-1}\boldsymbol{\sigma}$ und einen irreversiblen plastischen Anteil \boldsymbol{p} ,

$$\boldsymbol{\epsilon} = \boldsymbol{e}(\boldsymbol{\sigma}) + \boldsymbol{p}(\boldsymbol{\xi}). \quad (2.4)$$

Die elastische Verzerrung hängt von der Spannung $\boldsymbol{\sigma}$ ab, die plastische Verzerrung hängt von weiteren internen Variablen $\boldsymbol{\xi}$ ab. Die kinematischen Variablen \boldsymbol{p} und $\boldsymbol{\xi}$ bilden die verallgemeinerte Verzerrung $\boldsymbol{P} = (\boldsymbol{p}, \boldsymbol{\xi})$. Die entsprechende verallgemeinerte Spannung ist $\boldsymbol{\Sigma} = (\boldsymbol{\sigma}, \boldsymbol{\alpha})$.

Im folgenden soll die Evolutionsgleichung (2.3) für verschiedene elastoplastische Modelle angegeben werden. Diese Evolutionsgleichung wird Fließgesetz genannt. Die Fließfunktion Φ definiert eine abgeschlossene, konvexe Menge K durch

$$K = \{\boldsymbol{\Sigma} \in \mathbb{S}_d \times \mathbb{S}_m : \Phi(\boldsymbol{\Sigma}) \leq 0\}.$$

Mit D^* ist die Indikatorfunktion der Menge K bezeichnet, die definiert ist durch

$$D^*(\boldsymbol{\Sigma}) = \begin{cases} 0 & \text{falls } \Phi(\boldsymbol{\Sigma}) \leq 0, \\ \infty & \text{falls } \Phi(\boldsymbol{\Sigma}) > 0. \end{cases} \quad (2.5)$$

In der ratenunabhängigen Plastizität ist die zeitliche Ableitung der verallgemeinerten Verzerrung durch das Fließgesetz gegeben,

$$\begin{aligned} \dot{\boldsymbol{P}} \in \partial D^*(\boldsymbol{\Sigma}) &= \{(\boldsymbol{\sigma}^*, \boldsymbol{\alpha}^*) \in K : \forall (\boldsymbol{\tau}, \boldsymbol{\beta}) \in K, \\ &D^*(\boldsymbol{\sigma}, \boldsymbol{\alpha}) + \boldsymbol{\sigma}^* : (\boldsymbol{\tau} - \boldsymbol{\sigma}) + \boldsymbol{\alpha}^* : (\boldsymbol{\beta} - \boldsymbol{\alpha}) \leq D^*(\boldsymbol{\tau}, \boldsymbol{\beta})\}. \end{aligned} \quad (2.6)$$

Durch eine Yosida-Regularisierung der Indikatorfunktion D^* der Menge K ergibt sich ein viskoplastisches Modell, wie in [9] gezeigt. Die Regularisierung der Funktion D^* lautet

$$\varphi_\rho^*(\boldsymbol{\Sigma}) = \frac{1}{2\rho} |\boldsymbol{\Sigma} - \Pi_K \boldsymbol{\Sigma}|^2 \quad \text{für } \boldsymbol{\Sigma} \in \mathbb{S}_d \times \mathbb{S}_m. \quad (2.7)$$

Hierbei ist $\rho > 0$ ein kleiner Regularisierungsparameter und Π_K stellt den Projektionsoperator auf die Menge K dar. Die Projektion $\boldsymbol{T} = \Pi_K \boldsymbol{\Sigma}$ ist definiert durch

$$|\boldsymbol{\Sigma} - \boldsymbol{T}| = \inf\{|\boldsymbol{\Sigma} - \boldsymbol{T}'| : \boldsymbol{T}' \in \mathbb{S}_d \times \mathbb{S}_m \text{ mit } \Phi(\boldsymbol{T}') \leq 0\}. \quad (2.8)$$

Die Regularisierung φ_ρ^* konvergiert für $\rho \rightarrow 0$ gegen die Indikatorfunktion D^* [60]. Man erhält für diesen Grenzfall das Modell der Plastizität.

Satz 2.2. [36] Das Funktional φ_ρ^* ist konvex und Gateaux differenzierbar mit der Ableitung

$$(\varphi_\rho^*(\boldsymbol{\Sigma}))' = \frac{1}{\rho}(\boldsymbol{\Sigma} - \Pi_K \boldsymbol{\Sigma}). \quad (2.9)$$

Beweis. Der Beweis folgt [36]. Die Gateaux-Ableitung ist definiert durch

$$(\varphi_\rho^*(\boldsymbol{\Sigma})', \mathbf{T}) = \lim_{d \rightarrow 0} \frac{1}{d} [\varphi_\rho^*(\boldsymbol{\Sigma} + d\mathbf{T}) - \varphi_\rho^*(\boldsymbol{\Sigma})].$$

$$\begin{aligned} \varphi_\rho^*(\boldsymbol{\Sigma} + d\mathbf{T}) &= \frac{1}{2\rho} |\boldsymbol{\Sigma} + d\mathbf{T} - \Pi(\boldsymbol{\Sigma} + d\mathbf{T})|^2 \\ &= \frac{1}{2\rho} |(\boldsymbol{\Sigma} - \Pi\boldsymbol{\Sigma}) + d\mathbf{T} + \Pi\boldsymbol{\Sigma} - \Pi(\boldsymbol{\Sigma} + d\mathbf{T})|^2 \\ &= \frac{1}{2\rho} (|\boldsymbol{\Sigma} - \Pi\boldsymbol{\Sigma}|^2 + 2d(\boldsymbol{\Sigma} - \Pi\boldsymbol{\Sigma}, \mathbf{T}) + 2(\boldsymbol{\Sigma} - \Pi\boldsymbol{\Sigma}, \Pi\boldsymbol{\Sigma} - \Pi(\boldsymbol{\Sigma} + d\mathbf{T})) \\ &\quad + |d\mathbf{T} + \Pi\boldsymbol{\Sigma} - \Pi\boldsymbol{\Sigma} - \Pi(\boldsymbol{\Sigma} + d\mathbf{T})|^2). \end{aligned} \quad (2.10)$$

Der letzte Term in (2.10) ist von der Ordnung $O(|d|^2)$.

$$\begin{aligned} |d\mathbf{T} + \Pi\boldsymbol{\Sigma} - \Pi(\boldsymbol{\Sigma} + d\mathbf{T})| &\leq |d\mathbf{T}| + |\Pi\boldsymbol{\Sigma} - \Pi(\boldsymbol{\Sigma} + d\mathbf{T})| \leq |d||\mathbf{T}| + |\boldsymbol{\Sigma} - (\boldsymbol{\Sigma} + d\mathbf{T})| \\ &= |d||\mathbf{T}| + |d||\mathbf{T}| = 2|d||\mathbf{T}|. \end{aligned}$$

Im zweiten Schritt wurde ausgenutzt, daß die Projektion nichtexpansiv ist. Der vorletzte Summand in (2.10) ist ebenfalls von der Ordnung $O(|d|^2)$. Aufgrund der Eigenschaften der Projektion gilt

$$(\boldsymbol{\Sigma} - \Pi\boldsymbol{\Sigma}, \Pi\boldsymbol{\Sigma} - \Pi(\boldsymbol{\Sigma} + d\mathbf{T})) \geq 0. \quad (2.11)$$

Außerdem gilt für den vorletzten Summanden in (2.10)

$$\begin{aligned} &(\boldsymbol{\Sigma} - \Pi\boldsymbol{\Sigma}, \Pi\boldsymbol{\Sigma} - \Pi(\boldsymbol{\Sigma} + d\mathbf{T})) \\ &= (\boldsymbol{\Sigma} - \Pi(\boldsymbol{\Sigma} + d\mathbf{T}), \Pi\boldsymbol{\Sigma} - \Pi(\boldsymbol{\Sigma} + d\mathbf{T})) - (\Pi\boldsymbol{\Sigma} - \Pi(\boldsymbol{\Sigma} + d\mathbf{T}), \Pi\boldsymbol{\Sigma} - \Pi(\boldsymbol{\Sigma} + d\mathbf{T})) \\ &\leq (\boldsymbol{\Sigma} - \Pi(\boldsymbol{\Sigma} + d\mathbf{T}), \Pi\boldsymbol{\Sigma} - \Pi(\boldsymbol{\Sigma} + d\mathbf{T})) \\ &= \underbrace{(\boldsymbol{\Sigma} + d\mathbf{T} - \Pi(\boldsymbol{\Sigma} + d\mathbf{T}), \Pi\boldsymbol{\Sigma} - \Pi(\boldsymbol{\Sigma} + d\mathbf{T}))}_{\leq 0} - (d\mathbf{T}, \Pi\boldsymbol{\Sigma} - \Pi(\boldsymbol{\Sigma} + d\mathbf{T})) \\ &\leq (d\mathbf{T}, \Pi d\mathbf{T}). \end{aligned} \quad (2.12)$$

Die letzte Ungleichung gilt, da aufgrund der Projektionseigenschaften der erste Summand in der vorletzten Zeile von (2.12) kleiner gleich null ist. Da $\mathbf{0}$ in der konvexen Menge K liegt, folgt

$$\begin{aligned} (d\mathbf{T}, \Pi d\mathbf{T}) &= (-d\mathbf{T}, -\Pi d\mathbf{T}) \\ &= \underbrace{(-d\mathbf{T} - (\Pi(-d\mathbf{T})), \mathbf{0} - \Pi d\mathbf{T})}_{\leq 0} + (-\Pi d\mathbf{T}, -\Pi d\mathbf{T}) \leq (\Pi d\mathbf{T}, \Pi d\mathbf{T}). \end{aligned}$$

Wegen der Nichtexpansivität der Projektion folgt

$$(\Pi d\mathbf{T}, \Pi d\mathbf{T}) \leq |\Pi d\mathbf{T}|^2 = |\Pi d\mathbf{T} - \Pi \mathbf{0}|^2 \leq |d\mathbf{T} - \mathbf{0}|^2 = |d|^2 |\mathbf{T}|^2 = O(|d|^2).$$

Damit erhält man

$$\varphi_\varrho^*(\boldsymbol{\Sigma} + d\mathbf{T}) = \frac{1}{2\varrho} (|\boldsymbol{\Sigma} - \Pi\boldsymbol{\Sigma}|^2 + 2d(\boldsymbol{\Sigma} - \Pi\boldsymbol{\Sigma}, \mathbf{T}) + O(|d|^2))$$

und

$$\varphi_\varrho^*(\boldsymbol{\Sigma} + d\mathbf{T}) - \varphi_\varrho^*(\boldsymbol{\Sigma}) = \frac{1}{2\varrho} (2d(\boldsymbol{\Sigma} - \Pi\boldsymbol{\Sigma}, \mathbf{T}) + O(|d|^2)).$$

Für die Gateauxableitung erhält man

$$\lim_{d \rightarrow 0} \frac{1}{d} [\varphi_\varrho^*(\boldsymbol{\Sigma} + d\mathbf{T}) - \varphi_\varrho^*(\boldsymbol{\Sigma})] = \frac{1}{\varrho} (\boldsymbol{\Sigma} - \Pi\boldsymbol{\Sigma}, \mathbf{T}).$$

Die Gateaux-Ableitung $(\varphi_\varrho(\boldsymbol{\Sigma}))'$ ist monoton, sodaß $\varphi_\varrho(\boldsymbol{\Sigma})$ konvex ist. ■

Das Fließgesetz der regularisierten Viskoplastizität lautet mit Satz 2.2

$$\dot{\mathbf{P}} \in \partial\varphi_\varrho^*(\boldsymbol{\Sigma}) = \left\{ \frac{1}{\varrho} (\boldsymbol{\Sigma} - \Pi_K \boldsymbol{\Sigma}) \right\} \quad \text{für } \boldsymbol{\Sigma} \in \mathbb{S}_d \times \mathbb{S}_m. \quad (2.13)$$

Mit den Definitionen der Fließfunktion aus [9] lassen sich die Fließgesetze für die perfekte Viskoplastizität sowie Viskoplastizität mit Verfestigung angeben, indem die Projektion der verallgemeinerten Spannung auf die Menge K berechnet wird und in das Fließgesetz (2.13) eingesetzt wird.

Perfekte Viskoplastizität

Im Fall der perfekten Viskoplastizität, gibt es keine Verfestigung und keine internen Variablen $\boldsymbol{\xi}$. Die Fließfunktion ist gegeben durch

$$\Phi(\boldsymbol{\sigma}) = |\text{dev}(\boldsymbol{\sigma})| - \sigma_y. \quad (2.14)$$

Durch Minimierung des Ausdrucks $f(\boldsymbol{\tau}) = |\boldsymbol{\sigma} - \boldsymbol{\tau}|^2$ unter der Nebenbedingung $\Phi(\boldsymbol{\tau}) = |\text{dev} \boldsymbol{\tau}| - \sigma_y = 0$ ergibt sich mit $(s)_+ := \max(0, s)$, $s \in \mathbb{R}$, das Fließgesetz

$$\dot{\mathbf{p}} = 1/\varrho (1 - \sigma_y/|\text{dev}(\boldsymbol{\sigma})|)_+ \text{dev}(\boldsymbol{\sigma}). \quad (2.15)$$

Viskoplastizität mit isotroper Verfestigung

Viskoplastizität mit isotroper Verfestigung ist charakterisiert durch einen skalaren Verfestigungsparameter $\alpha \geq 0$ und das Verfestigungsmodul $H > 0$. Die Fließfunktion ist

$$\Phi(\boldsymbol{\sigma}, \alpha) = |\text{dev}(\boldsymbol{\sigma})| - \sigma_y(1 + H\alpha). \quad (2.16)$$

Durch Minimierung des Ausdrucks $f(\boldsymbol{\tau}, \beta) = |(\boldsymbol{\sigma} - \boldsymbol{\tau}, \alpha - \beta)|^2$ unter der Nebenbedingung $\Phi(\boldsymbol{\tau}, \beta) = |\operatorname{dev} \boldsymbol{\tau}| - \sigma_y(1 + H\beta) = 0$ ergibt sich für das Fließgesetz

$$\begin{pmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{\xi}} \end{pmatrix} = \frac{1}{\varrho} \frac{1}{1 + H^2 \sigma_y^2} \left(1 - \frac{(1 + \alpha H) \sigma_y}{|\operatorname{dev}(\boldsymbol{\sigma})|} \right)_+ \begin{pmatrix} \operatorname{dev}(\boldsymbol{\sigma}) \\ -H \sigma_y |\operatorname{dev}(\boldsymbol{\sigma})| \end{pmatrix}. \quad (2.17)$$

In diesem Modell ist der plastische Anteil der freien Energie gegeben durch $F^p = \frac{1}{2} H_1 \xi^2$. Die interne Spannung ist definiert durch $\alpha = -\frac{\partial F^p}{\partial \xi}$. Dies bedeutet $\alpha = -H_1 \xi$, wobei H_1 ein positiver Verfestigungsparameter ist.

Viskoplastizität mit kinematischer Verfestigung

Bei Viskoplastizität mit kinematischer Verfestigung ist die Fließfunktion gegeben durch

$$\Phi(\boldsymbol{\sigma}, \boldsymbol{\alpha}) = |\operatorname{dev}(\boldsymbol{\sigma}) - \operatorname{dev}(\boldsymbol{\alpha})| - \sigma_y. \quad (2.18)$$

Durch Minimierung des Ausdrucks $f(\boldsymbol{\tau}, \boldsymbol{\beta}) = |(\boldsymbol{\sigma} - \boldsymbol{\tau}, \boldsymbol{\alpha} - \boldsymbol{\beta})|^2$ unter der Nebenbedingung $\Phi(\boldsymbol{\tau}, \boldsymbol{\beta}) = |\operatorname{dev} \boldsymbol{\tau} - \operatorname{dev} \boldsymbol{\beta}| - \sigma_y = 0$ ergibt sich für das Fließgesetz

$$\begin{pmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{\xi}} \end{pmatrix} = \frac{1}{2\varrho} \left(1 - \frac{\sigma_y}{|\operatorname{dev}(\boldsymbol{\sigma} - \boldsymbol{\alpha})|} \right)_+ \begin{pmatrix} \operatorname{dev}(\boldsymbol{\sigma} - \boldsymbol{\alpha}) \\ -\operatorname{dev}(\boldsymbol{\sigma} - \boldsymbol{\alpha}) \end{pmatrix}. \quad (2.19)$$

Im Fall der linearen kinematischen Verfestigung hat der plastische Anteil der freien Energie die Form $F^p = \frac{1}{2} k_1 |\boldsymbol{\xi}|^2$. Die interne Spannung ist definiert durch $\boldsymbol{\alpha} = -\frac{\partial F^p}{\partial \boldsymbol{\xi}}$, wodurch $\boldsymbol{\alpha} = -k_1 \boldsymbol{\xi}$, mit einem positiven Parameter k_1 .

2.3 Elastoviskoplastisches Modellproblem

Ein materieller Körper nimmt ursprünglich das Gebiet $\Omega \subset \mathbb{R}^d$ ein. Der Rand Γ besteht aus den disjunkten Anteilen Γ_N und Γ_D , $\Gamma = \Gamma_N \cup \Gamma_D$. Auf dem Dirichlet-Rand sind zusätzlich zu den Dirichlet-Randbedingungen tangential Gleitrandbedingungen zugelassen. Beide Arten von Randbedingungen lassen sich in der Forderung

$$\boldsymbol{M} \cdot \boldsymbol{u} = \boldsymbol{w} \quad \text{auf } \Gamma_D \quad (2.20)$$

zusammenfassen. Dabei gilt für $\boldsymbol{M} \in \mathbb{R}^{d \times d} : \Gamma_D \rightarrow \mathbb{R}^d$,

$$\boldsymbol{M}(\boldsymbol{x}) = \begin{pmatrix} \boldsymbol{n}(\boldsymbol{x}) \\ \mathbf{0} \end{pmatrix} \quad (\text{Gleitrand}), \quad \boldsymbol{M}(\boldsymbol{x}) = \boldsymbol{I} = \begin{pmatrix} \boldsymbol{e}_1^T \\ \vdots \\ \boldsymbol{e}_d^T \end{pmatrix} \quad \text{Dirichlet-Rand}. \quad (2.21)$$

Für den Vektor $\boldsymbol{w} \in \mathbb{R}^d : \Gamma_D \rightarrow \mathbb{R}^d$ gilt

$$\boldsymbol{w}(\boldsymbol{x}) = \mathbf{0} \quad (\text{Gleitrand}), \quad \boldsymbol{w}(\boldsymbol{x}) = \boldsymbol{u}_D(\boldsymbol{x}) \quad (\text{Dirichlet-Rand}). \quad (2.22)$$

Mit diesen Definitionen lassen sich folgende Räume für die Verschiebungen definieren,

$$X_M := \{ \boldsymbol{v} \in H^1(\Omega)^d : \boldsymbol{M} \cdot \boldsymbol{v} = \boldsymbol{w} \text{ auf } \Gamma_D \}, \quad (2.23)$$

$$X_0 := \{ \boldsymbol{v} \in H^1(\Omega)^d : \boldsymbol{M} \cdot \boldsymbol{v} = \mathbf{0} \text{ auf } \Gamma_D \}. \quad (2.24)$$

Für die Spannungen wird der Raum S , für die internen Spannungen der Raum M und für die verallgemeinerten Spannungen der Raum \mathbb{T} definiert durch

$$S := \{\boldsymbol{\tau} \in \mathbb{S}_d, \tau_{ij} \in L^2(\Omega)\}, \quad (2.25)$$

$$M := \{\boldsymbol{\mu} = (\mu_j) : \mu_j \in L^2(\Omega), j = 1, \dots, m\}, \quad (2.26)$$

$$\mathbb{T} := S \times M. \quad (2.27)$$

Für eine gegebene Volumenkraft $\mathbf{f}(\mathbf{x}, t) \in H^1(0, t_G; L^2(\Omega)^d)$ und eine gegebene Oberflächenkraft $\mathbf{g}(\mathbf{x}, t) \in H^1(0, t_G; L^2(\Gamma_N)^d)$ lautet das Anfangsrandwertproblem der regularisierten Viskoplastizität in der schwachen Formulierung:

Problem 1 (*Duales Problem der Viskoplastizität*).

Man finde die Verschiebung $\mathbf{u}(\mathbf{x}, t) \in H^1(0, t_G; X_M)$, die Spannung $\boldsymbol{\sigma} \in H^1(0, t_G; S)$ und die plastische Variable $\boldsymbol{\alpha} \in H^1(0, t_G; M)$, so daß für fast alle $t \in [0, t_G]$

$$\int_{\Omega} \boldsymbol{\sigma}(\mathbf{x}, t) : \boldsymbol{\epsilon}(\mathbf{v}) \, d\mathbf{x} = \int_{\Omega} \mathbf{f}(\mathbf{x}, t) \cdot \mathbf{v} \, d\mathbf{x} + \int_{\Gamma_N} \mathbf{g}(\mathbf{x}, t) \cdot \mathbf{v} \, ds, \quad (2.28)$$

für alle $\mathbf{v} \in X_0$ gilt und das Fließgesetz fast überall in Ω erfüllt ist:

$$\begin{bmatrix} \boldsymbol{\epsilon}(\mathbf{u}(\mathbf{x}, t)) - \mathbb{C}^{-1} \boldsymbol{\sigma}(\mathbf{x}, t) \\ \boldsymbol{\xi}(\boldsymbol{\alpha}(\mathbf{x}, t)) \end{bmatrix} = \frac{1}{\varrho} \begin{bmatrix} \boldsymbol{\sigma}(\mathbf{x}, t) - \Pi_K \boldsymbol{\sigma}(\mathbf{x}, t) \\ \boldsymbol{\alpha}(\mathbf{x}, t) - \Pi_K \boldsymbol{\alpha}(\mathbf{x}, t) \end{bmatrix}. \quad (2.29)$$

Dabei liegen Anfangsbedingungen vor

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}), \quad \boldsymbol{\sigma}(\mathbf{x}, 0) = \boldsymbol{\sigma}_0(\mathbf{x}), \quad \boldsymbol{\alpha}(\mathbf{x}, 0) = \boldsymbol{\alpha}_0(\mathbf{x}), \quad (2.30)$$

und die Spannung ergibt sich aus der konstitutiven Gleichung durch

$$\boldsymbol{\sigma} = \frac{\partial F(\boldsymbol{\epsilon}, \boldsymbol{\xi})}{\partial \boldsymbol{\epsilon}}. \quad (2.31)$$

Im folgenden wird eine äquivalente Formulierung für das Modellproblem nach [36] angegeben.

Definition 2.3. *Es werden Bilinearformen definiert durch*

$$\bar{a} : S \times S \rightarrow \mathbb{R}, \quad \bar{a}(\boldsymbol{\sigma}, \boldsymbol{\tau}) = \int_{\Omega} \mathbb{C}^{-1} \boldsymbol{\sigma} : \boldsymbol{\tau} \, d\mathbf{x}, \quad (2.32)$$

$$b : X \times S \rightarrow \mathbb{R}, \quad b(\mathbf{v}, \boldsymbol{\tau}) = - \int_{\Omega} \boldsymbol{\epsilon}(\mathbf{v}) : \boldsymbol{\tau} \, d\mathbf{x}, \quad (2.33)$$

$$c : M \times M \rightarrow \mathbb{R}, \quad c(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \int_{\Omega} \mathbb{H}^{-1} \boldsymbol{\alpha} : \boldsymbol{\beta} \, d\mathbf{x}, \quad (2.34)$$

wobei $\mathbb{H} = H_1$ bei isotroper Verfestigung und $\mathbb{H} = k_1$ bei kinematischer Verfestigung. Mit $\boldsymbol{\Sigma} = (\boldsymbol{\sigma}, \boldsymbol{\alpha})$ und $\mathbf{T} = (\boldsymbol{\tau}, \boldsymbol{\beta})$ ist die Bilinearform A definiert durch

$$A : \mathbb{T} \times \mathbb{T} \rightarrow \mathbb{R}, \quad A(\boldsymbol{\Sigma}, \mathbf{T}) = \bar{a}(\boldsymbol{\sigma}, \boldsymbol{\tau}) + c(\boldsymbol{\alpha}, \boldsymbol{\beta}). \quad (2.35)$$

und die Funktionale $\psi_\rho : S \rightarrow \mathbb{R}$, $l(t) : X_0 \rightarrow \mathbb{R}$ sind definiert durch

$$\psi_\rho(\boldsymbol{\tau}, \boldsymbol{\beta}) = \int_{\Omega} \varphi_\rho(\boldsymbol{\tau}, \boldsymbol{\beta}) \, d\mathbf{x}, \quad (2.36)$$

$$\langle l(t), \mathbf{v} \rangle = - \int_{\Omega} \mathbf{f}(t) \cdot \mathbf{v} \, d\mathbf{x} - \int_{\Gamma_N} \mathbf{g}(t) \cdot \mathbf{v} \, ds. \quad (2.37)$$

Problem 2 (*Duales Problem der Viskoplastizität*).

Man finde für ein gegebenes $l \in H^1(0, t_G; X')$ mit $l(0) = 0$ Abbildungen $(\mathbf{u}, \boldsymbol{\Sigma}) = (\mathbf{u}, \boldsymbol{\sigma}, \boldsymbol{\alpha}) : [0, t_G] \rightarrow X_M \times \mathbb{T}$ mit $\mathbf{u}(0) = 0$, $\boldsymbol{\Sigma}(0) = 0$, so daß für fast alle $t \in [0, t_G]$,

$$b(\mathbf{v}, \boldsymbol{\sigma}(t)) = \langle l(t), \mathbf{v} \rangle \quad \forall \mathbf{v} \in X_0, \quad (2.38)$$

$$\psi_\rho(\mathbf{T}) - \psi_\rho(\boldsymbol{\Sigma}(t)) + A(\dot{\boldsymbol{\Sigma}}(t), \mathbf{T} - \boldsymbol{\Sigma}(t)) + b(\dot{\mathbf{u}}(t), \boldsymbol{\tau} - \boldsymbol{\sigma}(t)) \geq 0 \quad \forall \mathbf{T} \in \mathbb{T}. \quad (2.39)$$

2.4 Zeitdiskretisierung und Existenz von schwachen Lösungen

Das Modellproblem wird in der Zeit diskretisiert und ein Existenzsatz für eine schwache Lösung in einem Zeitschritt angegeben. Weiter wird eine a priori Fehlerabschätzung für den Zeitdiskretisierungsfehler bewiesen. Die Ausführungen folgen [36].

Das Zeitdiskretisierungsschema ist die verallgemeinerte Mittelpunktsregel. Es werden $n + 1$ diskrete Zeitpunkte $t_k \in [0, t_G]$ gewählt mit $t_0 = 0$, $t_n = t_G$ und $t_k < t_{k+1}$. Die Zeitintervalle sind nicht notwendigerweise äquidistant. Es gilt

$$t_k = t_{k-1} + \Delta t_k.$$

Das Zeitdiskretisierungsschema der verallgemeinerten Mittelpunktsregel definiert für einen Parameter $1/2 \leq \theta \leq 1$ Zeitpunkte $t_{k-1+\theta}$ durch

$$t_{k-1+\theta} = \theta t_k + (1 - \theta)t_{k-1}.$$

Die Werte der Verschiebung, der Spannung und der plastischen Variablen zum diskreten Zeitpunkt t_k sind bezeichnet durch \mathbf{u}_k , $\boldsymbol{\sigma}_k$, $\boldsymbol{\alpha}_k$ und $\boldsymbol{\xi}_k$. Die Werte zum Zeitpunkt $t_{k-1+\theta}$ werden durch lineare Interpolation bestimmt,

$$\begin{aligned} \mathbf{u}_{k-1+\theta} &= \theta \mathbf{u}_k + (1 - \theta) \mathbf{u}_{k-1}, & \boldsymbol{\sigma}_{k-1+\theta} &= \theta \boldsymbol{\sigma}_k + (1 - \theta) \boldsymbol{\sigma}_{k-1}, \\ \boldsymbol{\alpha}_{k-1+\theta} &= \theta \boldsymbol{\alpha}_k + (1 - \theta) \boldsymbol{\alpha}_{k-1}, & \boldsymbol{\xi}_{k-1+\theta} &= \theta \boldsymbol{\xi}_k + (1 - \theta) \boldsymbol{\xi}_{k-1}. \end{aligned} \quad (2.40)$$

Das Modellproblem aus Abschnitt 2.3 wird nach der verallgemeinerten Mittelpunktsregel in der Zeit diskretisiert. Für Viskoplastizität mit Verfestigung wird das zeitdiskretisierte Modellproblem angegeben.

Problem 3 (*Zeitdiskretes Modellproblem 1*). Man finde das Verschiebungsfeld $\mathbf{u}_k \in X_M$, die Spannung $\boldsymbol{\sigma}_k \in S$ und die plastische Variable $\boldsymbol{\alpha}_k \in M$ zu den diskreten Zeitpunkten t_k für $k = 1, \dots, n$, so daß für alle $\mathbf{v} \in X_0$

$$\int_{\Omega} \boldsymbol{\sigma}_{k-1+\theta} : \boldsymbol{\epsilon}(\mathbf{v}) \, d\mathbf{x} = \int_{\Omega} \mathbf{f}(t_{k-1+\theta}) \cdot \mathbf{v} \, d\mathbf{x} + \int_{\Gamma_N} \mathbf{g}(t_{k-1+\theta}) \cdot \mathbf{v} \, ds \quad (2.41)$$

und das zeitdiskrete Fließgesetz

$$\frac{1}{\theta \Delta t_k} \begin{bmatrix} \boldsymbol{\epsilon}(\mathbf{u}_{k-1+\theta} - \mathbf{u}_{k-1}) - \mathbb{C}^{-1}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{k-1}) \\ \boldsymbol{\xi}_{k-1+\theta} - \boldsymbol{\xi}_{k-1} \end{bmatrix} = \frac{1}{\varrho} \begin{bmatrix} \boldsymbol{\sigma}_{k-1+\theta} - \Pi_K \boldsymbol{\sigma}_{k-1+\theta} \\ \boldsymbol{\alpha}_{k-1+\theta} - \Pi_K \boldsymbol{\alpha}_{k-1+\theta} \end{bmatrix} \quad (2.42)$$

in Ω erfüllt sind. Die Anfangsbedingungen lauten

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}), \boldsymbol{\sigma}(\mathbf{x}, 0) = \boldsymbol{\sigma}_0(\mathbf{x}), \boldsymbol{\alpha}(\mathbf{x}, 0) = \boldsymbol{\alpha}_0(\mathbf{x}). \quad (2.43)$$

Bemerkung 1. Das Zeitdiskretisierungsschema nach der verallgemeinerten Mittelpunktsregel ist für $\theta = 1/2$ das Crank-Nicolson-Verfahren und für $\theta = 1$ das implizite Euler-Verfahren.

Eine äquivalente Formulierung für das zeitdiskrete Problem 3 lautet folgendermaßen.

Problem 4 (*Zeitdiskretes duales Problem*).

Man finde für ein gegebenes $l \in H^1(0, t_G; X')$ mit $l(0) = 0$ eine Folge $\{\mathbf{u}_k, \boldsymbol{\Sigma}_k\}_{k=0}^n \in X_M \times \mathbb{T}$ mit $\mathbf{u}_0 = 0, \boldsymbol{\Sigma}_0 = 0$, so daß für $k = 1, 2, \dots, n$

$$b(\mathbf{v}, \boldsymbol{\sigma}_{k-1+\theta}) = \langle l(t_{k-1+\theta}), \mathbf{v} \rangle \quad \forall \mathbf{v} \in X_0 \quad (2.44)$$

$$\begin{aligned} & \psi_{\varrho}(\mathbf{T}) - \psi_{\varrho}(\boldsymbol{\Sigma}_{k-1+\theta}) + A\left(\frac{\boldsymbol{\Sigma}_{k-1+\theta} - \boldsymbol{\Sigma}_{k-1}}{\theta \Delta t_k}, \mathbf{T} - \boldsymbol{\Sigma}_{k-1+\theta}\right) \\ & + b\left(\frac{\mathbf{u}_{k-1+\theta} - \mathbf{u}_{k-1}}{\theta \Delta t_k}, \boldsymbol{\tau} - \boldsymbol{\sigma}_{k-1+\theta}\right) \geq 0 \quad \forall \mathbf{T} \in \mathbb{T}. \end{aligned} \quad (2.45)$$

Der folgende Satz aus [36] wird benötigt, um die eindeutige Lösung des zeitdiskreten dualen Problems zu zeigen.

Satz 2.4. [36] *Mit V ist ein reeller Hilbertraum bezeichnet, $a : V \times V \rightarrow \mathbb{R}$ eine stetige, V -elliptische Bilinearform, $l : V \rightarrow \mathbb{R}$ ein beschränktes lineares Funktional und $j : V \rightarrow \mathbb{R}$ ein eigentliches, konvexes und unterhalbstetiges Funktional auf V . Dann hat die elliptische Variationsungleichung zweiter Art,*

$$a(u, v - u) + j(v) - j(u) \geq \langle l, v - u \rangle \quad \forall v \in V \quad (2.46)$$

eine eindeutige Lösung $u \in V$.

Satz 2.5. *Das zeitdiskrete duale Problem 4 hat eine eindeutige Lösung.*

Beweis. Das zeitdiskrete duale Problem ist mit $x := (\mathbf{u}, \boldsymbol{\sigma}, \boldsymbol{\alpha})$ und $y := (\mathbf{v}, \boldsymbol{\tau}, \boldsymbol{\beta})$, sowie

$$a(\mathbf{x}, \mathbf{y}) := \int_{\Omega} (\mathbb{C}^{-1} \boldsymbol{\sigma} - \boldsymbol{\epsilon}(\mathbf{u})) : \boldsymbol{\tau} \, d\mathbf{x} + \int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\epsilon}(\mathbf{v}) \, d\mathbf{x} + \int_{\Omega} \boldsymbol{\alpha} : \mathbb{H}^{-1} \boldsymbol{\beta} \, d\mathbf{x}$$

äquivalent zu der Variationsungleichung

$$\begin{aligned} & \psi_{\varrho}(\mathbf{y}) - \psi_{\varrho}(\mathbf{x}_{k-1+\theta}) + a\left(\frac{\mathbf{x}_{k-1+\theta} - \mathbf{x}_{k-1}}{\theta \Delta t_k}, \mathbf{y} - \mathbf{x}_{k-1+\theta}\right) \\ & \geq \left\langle \frac{l(t_{k-1+\theta}) - l(t_{k-1})}{\theta \Delta t_k}, \mathbf{y} - \mathbf{x}_{k-1+\theta} \right\rangle \quad \forall \mathbf{y} \in X_0 \times \mathbb{T}. \end{aligned}$$

Dies ist eine elliptische Variationsungleichung zweiter Art. Bei positiver Verfestigung existiert eine eindeutige Lösung $(\mathbf{u}_{k-1+\theta}, \boldsymbol{\Sigma}_{k-1+\theta}) \in X_M \times \mathbb{T}$, da die Voraussetzungen von Satz 2.4 erfüllt sind. Die Bilinearform $a(\mathbf{x}, \mathbf{y})$ ist symmetrisch, stetig und $X \times \mathbb{T}$ -elliptisch. Das Funktional $\psi_{\varrho} : S \rightarrow \mathbb{R}$ ist stetig, eigentlich und mit Satz 2.2 konvex. ■

Für das duale Problem kann eine a priori Fehlerabschätzung in der Zeit abgeleitet werden. Der Beweis für die Konvergenzordnungen der verallgemeinerten Mittelpunktmethode im Fall des viskoplastischen Variationsproblems kann auf den Fall der Elastoplastizität zurückgeführt werden, so daß im weiteren auf [36] zurückgegriffen werden kann, um folgenden Satz zu zeigen.

Satz 2.6. *Sei $\boldsymbol{\Sigma} \in W^{2,1}(0, t_G; \mathbb{T})$ falls $\theta \neq 1/2$ und $\boldsymbol{\Sigma} \in W^{3,1}(0, t_G; \mathbb{T})$ falls $\theta = 1/2$. Das zeitdiskrete duale Problem wird für eine äquidistante Unterteilung des Zeitintervalls $[0, t_G]$ in n Teile der Länge $\Delta t = t_G/n$ betrachtet. Die Unterteilungspunkte sind $t_k = k\Delta t$, $k = 0, \dots, n$. Dann gilt für die Lösung $\boldsymbol{\Sigma}_k$ des zeitdiskreten dualen Problems die a priori Abschätzung*

$$\max_{0 \leq k \leq n} \|\boldsymbol{\Sigma}(t_k) - \boldsymbol{\Sigma}_k\|_{L^2(\Omega)} \leq c \Delta t \|\boldsymbol{\Sigma}\|_{W^{2,1}(0, T; \mathbb{T})} \quad \text{falls } \theta \neq 1/2, \quad (2.47)$$

$$\max_{0 \leq k \leq n} \|\boldsymbol{\Sigma}(t_k) - \boldsymbol{\Sigma}_k\|_{L^2(\Omega)} \leq c (\Delta t)^2 \|\boldsymbol{\Sigma}\|_{W^{3,1}(0, T; \mathbb{T})} \quad \text{falls } \theta = 1/2. \quad (2.48)$$

Beweis. Aus dem zeitkontinuierlichen Problem erhält man mit $\mathbf{T} = \boldsymbol{\Sigma}_{k-1+\theta} \in \mathcal{P}_{k-1+\theta}$ und $\mathcal{P}_{k-1+\theta} := \{\mathbf{T} = (\boldsymbol{\tau}, \boldsymbol{\beta}) \in \mathbb{T} : \forall \mathbf{v} \in X_0 \quad b(\mathbf{v}, \boldsymbol{\tau}) = \langle l(t_{k-1+\theta}), \mathbf{v} \rangle\}$ die Abschätzung

$$\psi_{\varrho}(\boldsymbol{\Sigma}_{k-1+\theta}) - \psi_{\varrho}(\boldsymbol{\Sigma}(t_{k-1+\theta}) + A(\dot{\boldsymbol{\sigma}}(t_{k-1+\theta}), \boldsymbol{\Sigma}_{k-1+\theta} - \boldsymbol{\Sigma}(t_{k-1+\theta}))) \geq 0. \quad (2.49)$$

Aus dem zeitdiskreten Problem erhält man mit $\mathbf{T} = \boldsymbol{\Sigma}(t_{k-1+\theta}) \in \mathcal{P}_{k-1+\theta}$,

$$\psi_{\varrho}(\boldsymbol{\Sigma}(t_{k-1+\theta})) - \psi_{\varrho}(\boldsymbol{\Sigma}_{k-1+\theta}) + A\left(\frac{\boldsymbol{\Sigma}_{k-1+\theta} - \boldsymbol{\Sigma}_{k-1}}{\theta \Delta t}, \boldsymbol{\Sigma}(t_{k-1+\theta}) - \boldsymbol{\Sigma}_{k-1+\theta}\right) \geq 0. \quad (2.50)$$

Da A eine Bilinearform ist, ist (2.49) äquivalent zu

$$\psi_{\varrho}(\boldsymbol{\Sigma}_{k-1+\theta}) - \psi_{\varrho}(\boldsymbol{\Sigma}(t_{k-1+\theta})) + A(-\dot{\boldsymbol{\sigma}}(t_{k-1+\theta}), \boldsymbol{\Sigma}(t_{k-1+\theta}) - \boldsymbol{\Sigma}_{k-1+\theta}) \geq 0, \quad (2.51)$$

und mit

$$\frac{\Sigma_{k-1+\theta} - \Sigma_{k-1}}{\theta \Delta t} = \frac{\Sigma_k - \Sigma_{k-1}}{\Delta t} \quad (2.52)$$

ist (2.50) äquivalent zu

$$\Delta t (\psi_\varrho(\Sigma(t_{k-1+\theta})) - \psi_\varrho(\Sigma_{k-1+\theta})) + A(\Sigma_k - \Sigma_{k-1}, \Sigma(t_{k-1+\theta}) - \Sigma_{k-1+\theta}) \geq 0. \quad (2.53)$$

Multipliziert man (2.52) mit Δt und addiert das Ergebnis zu (2.53), dann folgt

$$A(\Sigma_k - \Sigma_{k-1} - \dot{\Sigma}(t_{k-1+\theta}), \Sigma(t_{k-1+\theta}) - \Sigma_{k-1+\theta}) \geq 0. \quad (2.54)$$

Das ist (13.7) aus [36]. Der weitere Beweis ist analog zu [36].

■

Kapitel 3

Numerische Lösung des Modellproblems

Dieses Kapitel beschreibt die Lösung des viskoplastischen Modellproblems mit der Methode der finiten Elemente. Der Lösungsalgorithmus beruht auf einer expliziten Darstellung des Spannungstensors $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\mathbf{u})$ in Abhängigkeit der unbekanntenen Verschiebung \mathbf{u} innerhalb eines Zeitschrittes der verallgemeinerten Mittelpunktsregel. Die Ableitung dieser Beziehung aus dem zeitdiskreten Fließgesetz zeigt Abschnitt 3.1. Neben der Verschiebung hängt der Spannungstensor $\boldsymbol{\sigma}_k$ zum diskreten Zeitpunkt t_k von Anfangswerten der Spannung $\boldsymbol{\sigma}_{k-1}$, der internen Spannung $\boldsymbol{\alpha}_{k-1}$ und der Verschiebung \mathbf{u}_{k-1} zum diskreten Zeitpunkt t_{k-1} ab. Diese Anfangswerte sind aus dem letzten Zeitschritt zu übernehmen in der Art, daß das Fließgesetz in allen Quadraturpunkten erfüllt ist. Die Lösung des nichtlinearen Variationsproblems in einem Zeitschritt beschreibt Abschnitt 3.2. Abschnitt 3.3 zeigt, wie ein adaptiver Algorithmus für das Modellproblem aufgebaut ist.

3.1 Analytische Darstellung der verallgemeinerten Spannung

Numerische Lösungen elastoplastischer Probleme können durch Prädiktor-Korrektor-Verfahren bestimmt werden, die auch Radial-Return-Algorithmen genannt werden und in dem Buch von J.C. Simo und T.J.R. Hughes [58] sowie in Arbeiten von C. Wiens beschrieben werden [66, 67]. Diese Methoden erfordern in jedem Zeitschritt die Lösung eines lokalen nichtlinearen Problems in jedem Quadraturpunkt, vorgegeben durch die Evolutionsgleichung des Materials, sowie die Lösung eines globalen linearen Variationsproblems.

In dieser Arbeit wird eine Lösungsstrategie entwickelt, die ausnutzt, daß für die Spannung und die interne Spannung in jedem Zeitschritt analytische Darstellungen in Abhängigkeit der Verschiebung sowie Werten aus dem vorherigen Zeitschritt existieren. Diese Methode wurde in [28] veröffentlicht und bildet in Kapitel 4 die Grundlage für eine a posteriori Fehlerabschätzung mit residuenbasierten Fehlerschätzern. Die Sätze 3.1, 3.2 und 3.3 zeigen, daß Abbildungen

$$\mathbf{S} : X_M \times X_M \times L^2(\Omega)^{2 \times 2} \times L^2(\Omega)^{m \times m} \rightarrow L^2(\Omega)^{2 \times 2}$$

und

$$\mathbf{R} : X_M \times X_M \times L^2(\Omega)^{2 \times 2} \times L^2(\Omega)^{m \times m} \rightarrow L^2(\Omega)^{m \times m},$$

existieren, so daß

$$\boldsymbol{\sigma}_{k-1+\theta} = \mathbf{S}(\mathbf{u}_{k-1+\theta}, \mathbf{u}_{k-1}, \boldsymbol{\sigma}_{k-1}, \boldsymbol{\alpha}_{k-1}) \quad k \in \{1, \dots, n\}, \quad (3.1)$$

$$\boldsymbol{\alpha}_{k-1+\theta} = \mathbf{R}(\mathbf{u}_{k-1+\theta}, \mathbf{u}_{k-1}, \boldsymbol{\sigma}_{k-1}, \boldsymbol{\alpha}_{k-1}) \quad k \in \{1, \dots, n\}. \quad (3.2)$$

Die Spannung $\boldsymbol{\sigma}_{k-1+\theta}$ und die plastische Variable $\boldsymbol{\alpha}_{k-1+\theta}$ zum Zeitpunkt $t_{k-1+\theta}$ lassen sich als Funktion der unbekanntenen Verschiebung $\mathbf{u}_{k-1+\theta}$ sowie Variablen zum Zeitpunkt t_{k-1} darstellen. Es gilt $m = d$ ($d = 2, 3$) im Fall der kinematischen Verfestigung, $m = 1$ im Fall der isotropen Verfestigung. Mit d ist die Dimension des Problems bezeichnet. Für perfekte Plastizität treten in (3.1) und (3.2) keine plastischen Variablen auf. Durch Einsetzen der Ausdrücke (3.1) und (3.2) in das quasistatische Kräftegleichgewicht (Problem 3) ist die Fließbedingung eliminiert. Damit ist in jedem Zeitschritt ein nichtlineares Variationsproblem mit der Verschiebung $\mathbf{u}_{k-1+\theta}$ als unbekannter Größe zu lösen.

Problem 5 (*Zeitdiskretes Modellproblem 2*). Man finde das Verschiebungsfeld $\mathbf{u}_k \in X_M$ zu den diskreten Zeitpunkten t_k für $k = 1, \dots, n$, so daß für alle $\mathbf{v} \in X_0$

$$\int_{\Omega} \mathbf{S}(\mathbf{u}_{k-1+\theta}, \mathbf{u}_{k-1}, \boldsymbol{\sigma}_{k-1}, \boldsymbol{\alpha}_{k-1}) : \boldsymbol{\epsilon}(\mathbf{v}) \, d\mathbf{x} = \int_{\Omega} \mathbf{f}(t_{k-1+\theta}) \cdot \mathbf{v} \, d\mathbf{x} + \int_{\Gamma_N} \mathbf{g}(t_{k-1+\theta}) \cdot \mathbf{v} \, ds \quad (3.3)$$

mit den Anfangsbedingungen

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}), \boldsymbol{\sigma}(\mathbf{x}, 0) = \boldsymbol{\sigma}_0(\mathbf{x}), \boldsymbol{\alpha}(\mathbf{x}, 0) = \boldsymbol{\alpha}_0(\mathbf{x}). \quad (3.4)$$

Die Ableitung der Beziehungen (3.1) und (3.2) für die Materialgesetze aus Abschnitt 2.2 (Fließgesetze (2.15), (2.17), (2.19)) ist Gegenstand des folgenden Teils in diesem Abschnitt. Zunächst wird die elastische Verzerrung in Abhängigkeit der Spannung ausgedrückt.

Bemerkung 2. Die elastische Verzerrung hat die Darstellung

$$\mathbb{C}^{-1} \boldsymbol{\sigma} = \frac{1}{d^2 \lambda + d 2 \mu} \operatorname{tr} \boldsymbol{\sigma} \mathbf{I} + \frac{1}{2 \mu} \operatorname{dev} \boldsymbol{\sigma}. \quad (3.5)$$

Dies folgt aus der Tatsache, daß Konstanten α , β , γ und δ existieren, so daß

$$\boldsymbol{\sigma} = \gamma \operatorname{tr} \mathbf{e} \mathbf{I} + \delta \operatorname{dev} \mathbf{e} \quad \text{und} \quad \mathbf{e} = \alpha \operatorname{tr} \boldsymbol{\sigma} \mathbf{I} + \beta \operatorname{dev} \boldsymbol{\sigma}. \quad (3.6)$$

Setzt man den Ausdruck für die Spannung $\boldsymbol{\sigma}$ aus (3.6) in den Ausdruck für die elastische Verzerrung \mathbf{e} aus (3.6) ein, ergibt sich

$$\mathbf{e} = d \alpha \gamma \operatorname{tr} \mathbf{e} \mathbf{I} + \beta \delta \operatorname{dev} \mathbf{e} \quad \text{mit} \quad \alpha = 1/(d^2 \gamma) \quad \text{und} \quad \beta = 1/\delta. \quad (3.7)$$

Wegen (3.6) gilt für den Spannungstensor $\boldsymbol{\sigma}$

$$\boldsymbol{\sigma} = \gamma \operatorname{tr} \mathbf{e} \mathbf{I} + \delta (\mathbf{e} - 1/d \operatorname{tr} \mathbf{e} \mathbf{I}) = (\gamma - \frac{\delta}{d}) \operatorname{tr} \mathbf{e} \mathbf{I} + \delta \mathbf{e} = \lambda \operatorname{tr} \mathbf{e} \mathbf{I} + 2 \mu \mathbf{e}. \quad (3.8)$$

Dies zeigt $\delta = 2 \mu$ und $\gamma = \lambda + (2 \mu)/d$. Eingesetzt in den Ausdruck für die elastische Verzerrung \mathbf{e} aus (3.6) erhält man mit α und β aus (3.7) die Darstellung (3.5).

3.1.1 Perfekte Plastizität

Dieser Abschnitt zeigt, daß für elastoviskoplastische Materialien ohne Verfestigung der Spannungstensor in Abhängigkeit von der Verzerrung und von Variablen aus dem letzten Zeitschritt dargestellt werden kann.

Satz 3.1. *Im Modell der perfekten Viskoplastizität (Abschnitt 2.2) gilt für die Abbildung \mathcal{S} aus (3.1)*

$$\begin{aligned} \mathcal{S}(\mathbf{u}_{k-1+\theta}, \mathbf{u}_{k-1}, \boldsymbol{\sigma}_{k-1}) &:= \boldsymbol{\sigma}_{k-1+\theta} \\ &= (\lambda + 2\mu/d) \operatorname{tr}(\theta \Delta t_k \mathbf{A}_{k-1+\theta}) \mathbf{I} + 2\mu h(|\operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta}|) \operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta} \end{aligned} \quad (3.9)$$

mit

$$h(x) = 1 - \max\left\{0, \frac{1}{1 + \Gamma_1} \left(1 - \frac{\Gamma_2}{2\mu x}\right)\right\}, \quad (3.10)$$

$$\mathbf{A}_{k-1+\theta} := \boldsymbol{\epsilon} \left(\frac{\mathbf{u}_{k-1+\theta} - \mathbf{u}_{k-1}}{\theta \Delta t_k} \right) + \mathbb{C}^{-1} \frac{\boldsymbol{\sigma}_{k-1}}{\theta \Delta t_k} \quad (3.11)$$

und den Konstanten

$$\Gamma_1 = \frac{\varrho}{2\mu\theta\Delta t_k} \quad \text{und} \quad \Gamma_2 = \sigma_y. \quad (3.12)$$

Beweis. Das Fließgesetz der perfekten Viskoplastizität (2.15) wird zeitdiskretisiert. Es ergibt sich

$$\mathbf{A}_{k-1+\theta} - \mathbb{C}^{-1} \frac{\boldsymbol{\sigma}_{k-1+\theta}}{\theta \Delta t_k} = \frac{1}{\varrho} \left(1 - \frac{\sigma_y}{|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}|}\right)_+ \operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}. \quad (3.13)$$

Daraus erhält man mit $\mathbb{C}^{-1} \boldsymbol{\sigma}_{k-1+\theta} = \alpha \operatorname{tr} \boldsymbol{\sigma}_{k-1+\theta} \mathbf{I} + \beta \operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}$

$$\operatorname{tr} \boldsymbol{\sigma}_{k-1+\theta} = \frac{1}{d\alpha} \operatorname{tr} \theta \Delta t_k \mathbf{A}_{k-1+\theta} = d \left(\lambda + \frac{2\mu}{d}\right) \operatorname{tr} \theta \Delta t_k \mathbf{A}_{k-1+\theta}.$$

Für rein elastisches Materialverhalten ist der Deviatoranteil des Spannungstensors in linearer Weise abhängig von der Verzerrung,

$$\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta} = \frac{1}{\beta} \operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta} = 2\mu \operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta}.$$

Für elastoviskoplastisches Materialverhalten, das auftritt, falls $(1 - \sigma_y/|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}|) > 0$, erhält man mit $\mathbb{C}^{-1} \boldsymbol{\sigma}_{k-1+\theta} = \alpha \operatorname{tr} \boldsymbol{\sigma}_{k-1+\theta} \mathbf{I} + \beta \operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}$ aus dem zeitdiskretisierten Fließgesetz (3.13)

$$\mathbf{A}_{k-1+\theta} - \frac{1}{\theta \Delta t_k} (\alpha \operatorname{tr} \boldsymbol{\sigma}_{k-1+\theta} \mathbf{I} + \beta \operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}) = \frac{1}{\varrho} \left(1 - \frac{\sigma_y}{|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}|}\right) \operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta} \quad (3.14)$$

und nach Deviatorbildung

$$\operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta} = \left(\beta + \frac{\theta \Delta t_k}{\varrho} \left(1 - \frac{\sigma_y}{|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}|}\right)\right) \operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}. \quad (3.15)$$

Das Ziel ist es im folgenden, aus dieser Gleichung den Deviatoranteil des Spannungstensors in Abhängigkeit der Verzerrung darzustellen. Zunächst kann ein Ausdruck für $|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}|$ abgeleitet werden. Aus (3.15) folgt

$$|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}| = \frac{\varrho |\operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta}| + \theta \Delta t_k \sigma_y}{\theta \Delta t_k + \beta \varrho}$$

und

$$\operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta} = \left(\beta + \frac{\theta \Delta t_k}{\varrho} \left(1 - \frac{\sigma_y (\theta \Delta t_k + \beta \varrho)}{\varrho |\operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta}| + \theta \Delta t_k \sigma_y} \right) \right) \operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}.$$

Damit ergibt sich eine Darstellung für den Deviatoranteil des Spannungstensors in Abhängigkeit der Verzerrung,

$$\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta} = \frac{\varrho |\operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta}| + \theta \Delta t_k \sigma_y}{(\beta \varrho + \theta \Delta t_k) |\operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta}|} \operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta}. \quad (3.16)$$

Mit der Bedingung $1 - \sigma_y / (2\mu |\operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta}|) > 0$ für das Auftreten von plastischem Materialverhalten, folgt aus (3.16), daß der Spannungstensor die Darstellung

$$\begin{aligned} \boldsymbol{\sigma}_{k-1+\theta} = & \left(\lambda + \frac{2\mu}{d} \right) \operatorname{tr} \theta \Delta t_{k+1} \mathbf{A}_{k-1+\theta} \\ & + 2\mu \left(1 - \max \left\{ 0, \frac{1}{1 + \Gamma_1} \left(1 - \frac{\sigma_y}{2\mu |\operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta}|} \right) \right\} \right) \operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta} \end{aligned}$$

besitzt mit $\Gamma_1 = \varrho / (2\mu \theta \Delta t_k)$. ■

3.1.2 Isotrope Verfestigung

Dieser Abschnitt zeigt, daß für elastoviskoplastische Materialien mit isotroper Verfestigung der Spannungstensor und die interne Spannung in Abhängigkeit von der Verzerrung und von Variablen aus dem letzten Zeitschritt dargestellt werden können.

Satz 3.2. *Im Modell der Elastoviskoplastizität mit isotroper Verfestigung (Abschnitt 2.2) gilt für die Abbildungen \mathbf{S} und \mathbf{R} aus (3.1) und (3.2)*

$$\begin{aligned} \mathbf{S}(\mathbf{u}_{k-1+\theta}, \mathbf{u}_{k-1}, \boldsymbol{\sigma}_{k-1}, \alpha_{k-1}) & := \boldsymbol{\sigma}_{k-1+\theta} \\ & = \left(\lambda + \frac{2\mu}{d} \right) \operatorname{tr} \theta \Delta t_k \mathbf{A}_{k-1+\theta} + 2\mu h(|\operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta}|) \operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta} \end{aligned} \quad (3.17)$$

und

$$\begin{aligned} \mathbf{R}(\mathbf{u}_{k-1+\theta}, \mathbf{u}_{k-1}, \boldsymbol{\sigma}_{k-1}, \alpha_{k-1}) & := \alpha_{k-1+\theta} \\ & = \frac{(\beta \varrho (1 + H^2 \sigma_y^2) + \theta \Delta t_k) \alpha_{k-1} + \theta \Delta t_k H_1 H \sigma_y (|\operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta}| - \beta \sigma_y)}{\theta \Delta t_k + \beta (\varrho (1 + H^2 \sigma_y^2) + \theta \Delta t_k H_1 H^2 \sigma_y^2)} \end{aligned} \quad (3.18)$$

mit

$$h(x) = 1 - \max \left\{ 0, \frac{1}{1 + \Gamma_1} \left(1 - \frac{\Gamma_2(\alpha_{k-1})}{2\mu x} \right) \right\}, \quad (3.19)$$

$$\mathbf{A}_{k-1+\theta} := \epsilon \left(\frac{\mathbf{u}_{k-1+\theta} - \mathbf{u}_{k-1}}{\theta \Delta t_k} \right) + \mathbb{C}^{-1} \frac{\boldsymbol{\sigma}_{k-1}}{\theta \Delta t_k} \quad (3.20)$$

und

$$\Gamma_1 = \frac{\varrho(1 + H^2 \sigma_y^2) + \theta \Delta t_{k-1} H_1 H^2 \sigma_y^2}{2\mu \theta \Delta t_{k-1}} \quad \text{und} \quad \Gamma_2(\alpha_{k-1}) = \sigma_y(1 + \alpha_{k-1} H). \quad (3.21)$$

Beweis. Das Fließgesetz der Viskoplastizität mit isotroper Verfestigung (2.17) wird zeitdiskretisiert. Es ergibt sich

$$\mathbf{A}_{k-1+\theta} - \mathbb{C}^{-1} \frac{\boldsymbol{\sigma}_{k-1+\theta}}{\theta \Delta t_k} = \frac{1}{\varrho} \frac{1}{1 + H^2 \sigma_y^2} \left(1 - \frac{(1 + \alpha_{k-1+\theta} H) \sigma_y}{|\text{dev } \boldsymbol{\sigma}_{k-1+\theta}|} \right)_+ \text{dev } \boldsymbol{\sigma}_{k-1+\theta}, \quad (3.22)$$

$$-H_1^{-1} \frac{\alpha_{k-1+\theta} - \alpha_{k-1}}{\theta \Delta t_k} = -\frac{1}{\varrho} \frac{1}{1 + H^2 \sigma_y^2} \left(1 - \frac{(1 + \alpha_{k-1+\theta} H) \sigma_y}{|\text{dev } \boldsymbol{\sigma}_{k-1+\theta}|} \right)_+ H \sigma_y |\text{dev } \boldsymbol{\sigma}_{k-1+\theta}|. \quad (3.23)$$

Mit $\mathbb{C}^{-1} \boldsymbol{\sigma}_{k-1+\theta} = \alpha \text{tr } \boldsymbol{\sigma}_{k-1+\theta} \mathbf{I} + \beta \text{dev } \boldsymbol{\sigma}_{k-1+\theta}$ erhält man aus (3.22)

$$\text{tr } \boldsymbol{\sigma}_{k-1+\theta} = \frac{1}{d\alpha} \text{tr } \theta \Delta t_k \mathbf{A}_{k-1+\theta} = d \left(\lambda + \frac{2\mu}{d} \right) \text{tr } \theta \Delta t_k \mathbf{A}_{k-1+\theta}. \quad (3.24)$$

Für rein elastisches Materialverhalten ist der Deviatoranteil des Spannungstensors in linearer Weise von der Verzerrung abhängig,

$$\text{dev } \boldsymbol{\sigma}_{k-1+\theta} = \frac{1}{\beta} \text{dev } \theta \Delta t_k \mathbf{A}_{k-1+\theta} = 2\mu \text{dev } \theta \Delta t_k \mathbf{A}_{k-1+\theta}. \quad (3.25)$$

Für $1 - (1 + \alpha_{k-1+\theta} H) \sigma_y / |\text{dev } \boldsymbol{\sigma}_{k-1+\theta}| > 0$ tritt elastoviskoplastisches Materialverhalten auf. Mit $\mathbb{C}^{-1} \boldsymbol{\sigma}_{k-1+\theta} = \alpha \text{tr } \boldsymbol{\sigma}_{k-1+\theta} \mathbf{I} + \beta \text{dev } \boldsymbol{\sigma}_{k-1+\theta}$ in (3.22) gilt dann

$$\text{dev } \theta \Delta t_k \mathbf{A}_{k-1+\theta} = \left(\beta + \frac{\theta \Delta t_k}{\varrho} \frac{1}{1 + H^2 \sigma_y^2} \left(1 - \frac{(1 + \alpha_{k-1+\theta} H) \sigma_y}{|\text{dev } \boldsymbol{\sigma}_{k-1+\theta}|} \right) \right) \text{dev } \boldsymbol{\sigma}_{k-1+\theta}. \quad (3.26)$$

Das Ziel ist es im folgenden, aus dieser Gleichung einen Ausdruck für den Deviatoranteil des Spannungstensors in Abhängigkeit der Verzerrung zu bestimmen. Mit der zweiten Komponente des zeitdiskretisierten Fließgesetzes kann die unbekannt interne Spannung eliminiert werden, indem (3.23) zunächst nach der internen Spannung aufgelöst wird:

$$\alpha_{k-1+\theta} = \frac{\theta \Delta t_k H_1 H \sigma_y (|\text{dev } \boldsymbol{\sigma}_{k-1+\theta}| - \sigma_y) + \varrho(1 + H^2 \sigma_y^2) \alpha_{k-1}}{(1 + H^2 \sigma_y^2) \varrho + \theta \Delta t_k H_1 H^2 \sigma_y^2}, \quad (3.27)$$

und das Ergebnis in (3.26) eingesetzt wird:

$$\begin{aligned} \text{dev } \theta \Delta t_k \mathbf{A}_{k-1+\theta} &= \text{dev } \boldsymbol{\sigma}_{k-1+\theta} \times \\ \frac{\theta \Delta t_k (|\text{dev } \boldsymbol{\sigma}_{k-1+\theta}| - (1 + \alpha_{k-1} H) \sigma_y) + \beta(\varrho(1 + H^2 \sigma_y^2) + \theta \Delta t_k H_1 H^2 \sigma_y^2) |\text{dev } \boldsymbol{\sigma}_{k-1+\theta}|}{(\varrho(1 + H^2 \sigma_y^2) + \theta \Delta t_k H_1 H^2 \sigma_y^2) |\text{dev } \boldsymbol{\sigma}_{k-1+\theta}|} & \end{aligned} \quad (3.28)$$

Auflösen nach $|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}|$ ergibt

$$|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}| = \frac{\theta \Delta t_k \sigma_y (1 + \alpha_{k-1} H) + (\varrho(1 + H^2 \sigma_y^2) + \theta \Delta t_k H_1 H^2 \sigma_y^2) |\operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta}|}{\varrho \Delta t_k + \beta(\varrho(1 + H^2 \sigma_y^2) + \theta \Delta t_k H_1 H^2 \sigma_y^2)}. \quad (3.29)$$

Einsetzen von (3.29) in (3.28) ergibt die Darstellung des Deviatoranteils der Spannung in Abhängigkeit der Verzerrung,

$$\begin{aligned} \operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta} &= \\ &= \frac{\theta \Delta t_k \sigma_y (1 + \alpha_{k-1} H) + (\varrho(1 + H^2 \sigma_y^2) + \theta \Delta t_k H_1 H^2 \sigma_y^2) |\operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta}|}{(\beta \varrho(1 + H^2 \sigma_y^2) + \theta \Delta t_k (1 + \beta H_1 H^2 \sigma_y^2)) |\operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta}|} \\ &\times \operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta}. \end{aligned} \quad (3.30)$$

Aus (3.27) und (3.29) erhält man einen Ausdruck für die interne Spannung in Abhängigkeit der Verzerrung,

$$\alpha_{k-1+\theta} = \frac{(\beta \varrho(1 + H^2 \sigma_y^2) + \theta \Delta t_k) \alpha_{k-1} + \theta \Delta t_k H_1 H \sigma_y (|\operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta}| - \beta \sigma_y)}{\theta \Delta t_k + \beta(\varrho(1 + H^2 \sigma_y^2) + \theta \Delta t_k H_1 H^2 \sigma_y^2)}. \quad (3.31)$$

Mit (3.29) und (3.31) ergibt sich aus $1 - (1 + \alpha_{k-1+\theta} H) \sigma_y / |\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}| > 0$ die Bedingung $1 - \beta(1 + \alpha_{k-1} H) \sigma_y / |\operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta}| > 0$ für das Auftreten von plastischem Materialverhalten. Mit (3.30) folgt die Darstellung

$$\begin{aligned} \boldsymbol{\sigma}_{k-1+\theta} &= \left(\lambda + \frac{2\mu}{d} \right) \operatorname{tr} \theta \Delta t_{k+1} \mathbf{A}_{k-1+\theta} \\ &+ 2\mu \left(1 - \max \left\{ 0, \frac{1}{1 + \Gamma_1} \left(1 - \frac{(1 + \alpha_{k-1} H) \sigma_y}{2\mu |\operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta}|} \right) \right\} \right) \operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta} \end{aligned} \quad (3.32)$$

für den Spannungstensor, wobei $\Gamma_1 = (\varrho(1 + H^2 \sigma_y^2) + \theta \Delta t_{k-1} H_1 H^2 \sigma_y^2) / (2\mu \theta \Delta t_{k-1})$. ■

3.1.3 Kinematische Verfestigung

Dieser Abschnitt zeigt, daß für elastoviskoplastische Materialien mit kinematischer Verfestigung der Spannungstensor und die interne Spannung in Abhängigkeit von der Verzerrung und von Variablen aus dem letzten Zeitschritt dargestellt werden können.

Satz 3.3. *Im Modell der Elastoviskoplastizität mit kinematischer Verfestigung (Abschnitt 2.2) gilt für die Abbildungen \mathbf{S} und \mathbf{R} aus (3.1) und (3.2)*

$$\begin{aligned} \mathbf{S}(\mathbf{u}_{k-1+\theta}, \mathbf{u}_{k-1}, \boldsymbol{\sigma}_{k-1}, \boldsymbol{\alpha}_{k-1}) &:= \boldsymbol{\sigma}_{k-1+\theta} \\ &= \left(\lambda + \frac{2\mu}{d} \right) \operatorname{tr}(\theta \Delta t_k \mathbf{A}_{k-1+\theta}) \mathbf{I} \\ &+ 2\mu h(|\operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta} - \beta \boldsymbol{\alpha}_{k-1}|) \operatorname{dev}(\theta \Delta t_k \mathbf{A}_{k-1+\theta} - \beta \boldsymbol{\alpha}_{k-1}) + \operatorname{dev} \boldsymbol{\alpha}_{k-1} \end{aligned} \quad (3.33)$$

und

$$\mathbf{R}(\mathbf{u}_{k-1+\theta}, \mathbf{u}_{k-1}, \boldsymbol{\sigma}_{k-1}, \boldsymbol{\alpha}_{k-1}) = \boldsymbol{\alpha}_{k-1+\theta} = \boldsymbol{\alpha}_{k-1} + k_1(\operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta} - \beta \operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}) \quad (3.34)$$

mit

$$h(x) = 1 - \max\left\{0, \frac{1}{1 + \Gamma_1} \left(1 - \frac{\Gamma_2}{2\mu x}\right)\right\}, \quad (3.35)$$

$$\mathbf{A}_{k-1+\theta} := \boldsymbol{\epsilon} \left(\frac{\mathbf{u}_{k-1+\theta} - \mathbf{u}_{k-1}}{\theta \Delta t_k} \right) + \mathbb{C}^{-1} \frac{\boldsymbol{\sigma}_{k-1}}{\theta \Delta t_k} \quad (3.36)$$

und den Konstanten

$$\Gamma_1 = \frac{2\varrho + \theta \Delta t_{k-1} k_1}{2\mu \theta \Delta t_{k-1}} \quad \text{und} \quad \Gamma_2 = \sigma_y. \quad (3.37)$$

Beweis. Das Fließgesetz der Viskoplastizität mit kinematischer Verfestigung (2.19) wird zeitdiskretisiert. Es ergibt sich

$$\begin{aligned} \mathbf{A}_{k-1+\theta} - \mathbb{C}^{-1} \frac{\boldsymbol{\sigma}_{k-1+\theta}}{\theta \Delta t_k} &= \frac{1}{2\varrho} \left(1 - \frac{\sigma_y}{|\operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1+\theta})|}\right)_+ \operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1+\theta}) \\ &= \frac{1}{k_1} \frac{\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1}}{\theta \Delta t_k}. \end{aligned} \quad (3.38)$$

Mit $\mathbb{C}^{-1} \boldsymbol{\sigma}_{k-1+\theta} = \alpha \operatorname{tr} \boldsymbol{\sigma}_{k-1+\theta} \mathbf{I} + \beta \operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}$ erhält man aus (3.38)

$$\operatorname{tr} \boldsymbol{\sigma}_{k-1+\theta} = \frac{1}{d\alpha} \operatorname{tr} \theta \Delta t_k \mathbf{A}_{k-1+\theta} = d \left(\lambda + \frac{2\mu}{d} \right) \operatorname{tr} \theta \Delta t_k \mathbf{A}_{k-1+\theta}. \quad (3.39)$$

Für rein elastisches Materialverhalten ist der Deviatoranteil des Spannungstensors in linearer Weise von der Verzerrung abhängig,

$$\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta} = \frac{1}{\beta} \operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta} = 2\mu \operatorname{dev} \theta \Delta t_k \mathbf{A}_{k-1+\theta}. \quad (3.40)$$

Für $1 - \sigma_y / |\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1+\theta}| > 0$ ist das Materialverhalten plastisch. Mit $\mathbb{C}^{-1} \boldsymbol{\sigma}_{k-1+\theta} = \alpha \operatorname{tr} \boldsymbol{\sigma}_{k-1+\theta} \mathbf{I} + \beta \operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}$ im ersten Teil von (3.38) wird die interne Spannung in Abhängigkeit der Spannung und der Verzerrung ausgedrückt,

$$\boldsymbol{\alpha}_{k-1+\theta} = \boldsymbol{\alpha}_{k-1} + k_1(\theta \Delta t_k \mathbf{A}_{k-1+\theta} - \beta \operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}). \quad (3.41)$$

Aus der ersten Komponente des Fließgesetzes ergibt sich damit der Ausdruck

$$\begin{aligned} &\operatorname{dev}(\theta \Delta t_k \mathbf{A}_{k-1+\theta} - \beta \boldsymbol{\sigma}_{k-1+\theta}) \\ &= \frac{\theta \Delta t_k}{2\varrho} \left(1 - \frac{\sigma_y}{|\operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1} - k_1 \theta \Delta t_k \mathbf{A}_{k-1+\theta} + k_1 \beta \boldsymbol{\sigma}_{k-1+\theta})|}\right) \\ &\quad \times \operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1} - k_1 \theta \Delta t_k \mathbf{A}_{k-1+\theta} + k_1 \beta \boldsymbol{\sigma}_{k-1+\theta}), \end{aligned} \quad (3.42)$$

aus dem sich ein Ausdruck für den Deviatoranteil des Spannungstensors in Abhängigkeit der Verzerrung bestimmen läßt. Zunächst erhält man daraus einen Ausdruck

für $|\text{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1} - k_1\theta\Delta t_k \mathbf{A}_{k-1+\theta} + k_1\beta\boldsymbol{\sigma}_{k-1+\theta})|$ und setzt diesen in (3.42) ein, wodurch

$$\begin{aligned} & \left(\left(\frac{2\varrho}{\theta\Delta t_k} + k_1 \right) |\text{dev}(\theta\Delta t_k \mathbf{A}_{k-1+\theta} - \beta\boldsymbol{\sigma}_{k-1+\theta})| + \sigma_y \right) \text{dev}(\theta\Delta t_k \mathbf{A}_{k-1+\theta} - \beta\boldsymbol{\sigma}_{k-1+\theta}) \\ & = |\text{dev}(\theta\Delta t_k \mathbf{A}_{k-1+\theta} - \beta\boldsymbol{\sigma}_{k-1+\theta})| \text{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1}). \end{aligned} \quad (3.43)$$

Daraus erhält einen Ausdruck für $|\text{dev}(\theta\Delta t_k \mathbf{A}_{k-1+\theta} - \beta\boldsymbol{\sigma}_{k-1+\theta})|$, der in (3.43) eingesetzt wird, so daß

$$\begin{aligned} & |\text{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1})| \text{dev}(\theta\Delta t_k \mathbf{A}_{k-1+\theta} - \beta\boldsymbol{\alpha}_{k-1}) \\ & = (\beta |\text{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1})| + \frac{|\text{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1})| - \sigma_y}{k_1 + 2\varrho/(\theta\Delta t_k)}) \text{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1}) \end{aligned} \quad (3.44)$$

und

$$|\text{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1})| = \frac{(\theta\Delta t_k k_1 + 2\varrho) |\text{dev}(\theta\Delta t_k \mathbf{A}_{k-1+\theta} - \beta\boldsymbol{\alpha}_{k-1})| + \theta\Delta t_k \sigma_y}{\theta\Delta t_k + \beta\theta\Delta t_k k_1 + 2\beta\varrho}. \quad (3.45)$$

Einsetzen von (3.45) in (3.44) ergibt die Darstellung des Deviators der Spannung in Abhängigkeit der Verzerung,

$$\begin{aligned} \text{dev } \boldsymbol{\sigma}_{k-1+\theta} = & \left(\frac{\theta\Delta t_k k_1 + 2\varrho}{\theta\Delta t_k + \theta\Delta t_k k_1/(2\mu) + \varrho/\mu} \right. \\ & \left. + \frac{\theta\Delta t_k \sigma_y}{(\theta\Delta t_k + \theta\Delta t_k k_1/(2\mu) + \varrho/\mu) |\text{dev}(\theta\Delta t_k \mathbf{A}_{k-1+\theta} - \beta\boldsymbol{\alpha}_{k-1})|} \right) \\ & \times \text{dev}(\theta\Delta t_k \mathbf{A}_{k-1+\theta} - \beta\boldsymbol{\alpha}_{k-1}) + \text{dev } \boldsymbol{\alpha}_{k-1}. \end{aligned} \quad (3.46)$$

Für $1 - \sigma_y/|\text{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1+\theta})| > 0$ ist das Materialverhalten plastisch, dies entspricht der Bedingung $1 - \beta\sigma_y/|\text{dev}(\theta\Delta t_k \mathbf{A}_{k-1+\theta} - \beta\boldsymbol{\alpha}_{k-1})| > 0$. Mit (3.46) hat der Spannungstensor die Darstellung

$$\begin{aligned} \boldsymbol{\sigma}_{k-1+\theta} = & \left(\lambda + \frac{2\mu}{d} \right) \text{tr } \theta\Delta t_k \mathbf{A}_{k-1+\theta} \\ & + 2\mu \left(1 - \max\left\{ 0, \frac{1}{1 + \Gamma_1} \left(1 - \frac{\sigma_y}{2\mu |\text{dev}(\theta\Delta t_k \mathbf{A}_{k-1+\theta} - \beta\boldsymbol{\alpha}_{k-1})|} \right) \right\} \right) \\ & \times \text{dev}(\theta\Delta t_k \mathbf{A}_{k-1+\theta} - \beta\boldsymbol{\alpha}_{k-1}) + \text{dev } \boldsymbol{\alpha}_{k-1}, \end{aligned}$$

wobei $\Gamma_1 = (2\varrho + \theta\Delta t_{k-1} k_1)/(2\mu\theta\Delta t_{k-1})$. ■

3.2 Finite-Elemente-Lösung der Gleichgewichtsbedingung

Dieser Abschnitt beschreibt die Ortsdiskretisierung des Modellproblems (Problem 5) und die numerische Lösung der Gleichgewichtsbedingung.

3.2.1 Triangulierung und Finite-Elemente-Raum

Gegeben sei ein offenes, beschränktes und zusammenhängendes Gebiet $\Omega \subset \mathbb{R}^2$ mit polygonalem Rand und eine Triangulierung \mathcal{T} des Gebietes Ω in abgeschlossene Dreiecke T . Die Menge der Elementkanten ist mit \mathcal{E} bezeichnet, die Menge der Knoten mit \mathcal{N} . Es werden nur reguläre Triangulierungen zugelassen. Die Zerlegung

$$\bar{\Omega} = \cup_{T \in \mathcal{T}} T$$

von $\bar{\Omega}$ ist regulär, wenn für zwei verschiedene Elemente T_1 und T_2 gilt, daß $\overset{\circ}{T}_1 \cap \overset{\circ}{T}_2 = \emptyset$ und wenn $F = T_1 \cap T_2 \neq \emptyset$ entweder eine gemeinsame Kante oder ein gemeinsamer Knoten ist. Für jedes Element $T \in \mathcal{T}$ ist

$$h_T = \text{diam}(T), \quad (3.47)$$

$$\rho_T = \sup\{\text{diam}(B) : B \text{ ist ein Kreis, der in } T \text{ enthalten ist}\}. \quad (3.48)$$

Die Netzweite h von \mathcal{T} ist $h(\mathcal{T}) = \max_{T \in \mathcal{T}} \{h_T\}$. Für die Familie $\{\mathcal{T}_i\}_{i=1}^{\infty}$ wird κ -Regularität vorausgesetzt, in dem Sinne, daß eine Konstante κ unabhängig von i existiert, so daß

$$\sup_{T \in \mathcal{T}_i} \frac{h_T}{\rho_T} \leq \kappa \leq \infty, \quad i = 1, 2, \dots \quad (3.49)$$

Im weiteren Verlauf dieser Arbeit wird die Definition

$$\Omega_z = \cup\{T' \in \mathcal{T} : z \subset T'\} \quad (3.50)$$

benötigt. Mit Ω_z ist der Patch zum Knoten $z \in \mathcal{N}$ bezeichnet. Das Gebiet \tilde{T} besteht aus dem Element T und den angrenzenden Elementen, die einen gemeinsamen Knoten mit T haben:

$$\tilde{T} = \{\cup T', T' \in \mathcal{T} : T' \cap T \neq \emptyset\}. \quad (3.51)$$

Dann gilt mit [4], daß eine Konstante C existiert, die nur von κ abhängt, so daß

$$h_{\tilde{T}} \leq C h_T \quad \forall T' \subset \tilde{T}. \quad (3.52)$$

Im folgenden soll der Finite-Elemente-Raum $\mathcal{S}^p(\Omega) \subset H^1(\Omega)^2$ konstruiert werden. Dazu wird das Referenzelement

$$\hat{T} = \text{conv}\{(-1, 0), (1, 0), (0, \sqrt{3})\}$$

eingeführt. Für jedes Element $T \in \mathcal{T}$ ist eine affine Abbildung

$$F_T : \hat{T} \rightarrow T$$

definiert. Gilt $(x, y) \in T = \text{conv}\{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}$ und $\hat{x}, \hat{y} \in \hat{T}$, dann ist

$$\begin{bmatrix} x \\ y \end{bmatrix} = F_T \left(\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} \right) = \begin{bmatrix} \frac{x_2 - x_1}{2} & \left(\frac{-x_1 - x_2}{2\sqrt{3}} + \frac{x_3}{\sqrt{3}} \right) \\ \frac{y_2 - y_1}{2} & \left(\frac{-y_1 - y_2}{2\sqrt{3}} + \frac{y_3}{\sqrt{3}} \right) \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} + \begin{bmatrix} \frac{x_1 + x_2}{2} \\ \frac{y_1 + y_2}{2} \end{bmatrix}.$$

Der Raum der Polynome vom totalen Grad p auf dem Referenzelement \hat{T} ist

$$\mathcal{S}^p(\hat{T}) = \left\{ \mathbf{g}(x, y) = \sum_{i=1}^2 \sum_{\substack{j,k=0 \\ j+k \leq p}}^p a_{jk} x^j y^k \mathbf{e}_i : a_{jk} \in \mathbb{R}, (x, y) \in \hat{T} \right\}. \quad (3.53)$$

Mit $\mathbf{e}_1, \mathbf{e}_2$ sind die Einheitsvektoren in \mathbb{R}^2 bezeichnet. Es gilt $\dim(\mathcal{S}^p(\hat{T})) = (p+1)(p+2)$. Für den Raum $\mathcal{S}^p(\hat{T})$ werden nun hierarchische Basisfunktionen definiert. Diese Basisfunktionen sind abgeleitet von Legendre-Polynomen P_n , $n \in \mathbb{N}_0$, und einer Funktion $\psi_j(x)$, $j = 2, \dots, p$. Legendre-Polynome sind rekursiv definiert durch

$$\begin{aligned} P_0(x) &= 1, \\ P_1(x) &= x, \\ P_{n+1}(x) &= \frac{2n+1}{n+1} x P_n(x) - \frac{n}{n+1} P_{n-1}(x) \quad \text{für } n \in \mathbb{N}. \end{aligned} \quad (3.54)$$

Die Funktion $\psi_j(x)$ ist für $j = 2, \dots, p$ definiert durch

$$\psi_j(x) := \frac{4}{\sqrt{2(2j-1)}} \frac{P_j(x) - P_{j-2}(x)}{1-x^2}. \quad (3.55)$$

Basisfunktionen auf dem Referenzelement \hat{T} werden in drei Gruppen eingeteilt. Dies sind Knotenbasisfunktionen, Kantenbasisfunktionen und innere Basisfunktionen. Die Kantenbasisfunktionen treten auf, falls für den Polynomgrad des Referenzelementes $p \geq 2$ gilt, und die inneren Basisfunktionen treten auf, falls $p \geq 3$.

Zu den Elementknoten auf dem Referenzelement \hat{T} gehören die Knotenbasisfunktionen

$$\hat{N}_1(\hat{x}, \hat{y}) := \frac{1}{2} \left(1 - \hat{x} - \frac{\hat{y}}{\sqrt{3}} \right), \quad \hat{N}_2(\hat{x}, \hat{y}) := \frac{1}{2} \left(1 + \hat{x} - \frac{\hat{y}}{\sqrt{3}} \right), \quad \hat{N}_3(\hat{x}, \hat{y}) := \frac{\hat{y}}{\sqrt{3}}. \quad (3.56)$$

Zu den Elementkanten des Referenzelementes \hat{T} sind für $j = 2, \dots, p$ die folgenden Kantenbasisfunktionen definiert,

$$\begin{aligned} \hat{N}_j^1(\hat{x}, \hat{y}) &:= \hat{N}_1(\hat{x}, \hat{y}) \hat{N}_2(\hat{x}, \hat{y}) \psi_j(\hat{N}_2(\hat{x}, \hat{y}) - \hat{N}_1(\hat{x}, \hat{y})), \\ \hat{N}_j^2(\hat{x}, \hat{y}) &:= \hat{N}_2(\hat{x}, \hat{y}) \hat{N}_3(\hat{x}, \hat{y}) \psi_j(\hat{N}_3(\hat{x}, \hat{y}) - \hat{N}_2(\hat{x}, \hat{y})), \\ \hat{N}_j^3(\hat{x}, \hat{y}) &:= \hat{N}_3(\hat{x}, \hat{y}) \hat{N}_1(\hat{x}, \hat{y}) \psi_j(\hat{N}_1(\hat{x}, \hat{y}) - \hat{N}_3(\hat{x}, \hat{y})). \end{aligned} \quad (3.57)$$

Für $0 \leq i+j \leq p-3$ sind $(p-1)(p-2)/2$ innere Basisfunktionen auf dem Referenzelement \hat{T} definiert durch

$$\hat{N}_{i,j}(\hat{x}, \hat{y}) = \hat{N}_1(\hat{x}, \hat{y}) \hat{N}_2(\hat{x}, \hat{y}) \hat{N}_3(\hat{x}, \hat{y}) P_i(\hat{N}_2(\hat{x}, \hat{y}) - \hat{N}_1(\hat{x}, \hat{y})) P_j(2\hat{N}_3(\hat{x}, \hat{y}) - 1). \quad (3.58)$$

Eine Basis des Raumes $\mathcal{S}^p(\hat{T})$ ist gegeben durch $\{\hat{\varphi}_1 \mathbf{e}_1, \hat{\varphi}_1 \mathbf{e}_2, \dots, \hat{\varphi}_k \mathbf{e}_1, \hat{\varphi}_k \mathbf{e}_2\}$ mit

$k = (p + 1)(p + 2)/2$ und

$$\begin{aligned}
\hat{\varphi}_1(\hat{x}, \hat{y}) &= \hat{N}_1(\hat{x}, \hat{y}), & \hat{\varphi}_2(\hat{x}, \hat{y}) &= \hat{N}_2(\hat{x}, \hat{y}), & \hat{\varphi}_3(\hat{x}, \hat{y}) &= \hat{N}_3(\hat{x}, \hat{y}), \\
\hat{\varphi}_4(\hat{x}, \hat{y}), \dots, \hat{\varphi}_{2+p}(\hat{x}, \hat{y}) &= \hat{N}_2^1(\hat{x}, \hat{y}), \dots, \hat{N}_p^1(\hat{x}, \hat{y}), \\
\hat{\varphi}_{3+p}(\hat{x}, \hat{y}), \dots, \hat{\varphi}_{1+2p}(\hat{x}, \hat{y}) &= \hat{N}_2^2(\hat{x}, \hat{y}) \dots \hat{N}_p^2(\hat{x}, \hat{y}), \\
\hat{\varphi}_{2+2p}(\hat{x}, \hat{y}), \dots, \hat{\varphi}_{3p}(\hat{x}, \hat{y}) &= \hat{N}_2^3(\hat{x}, \hat{y}) \dots \hat{N}_p^3(\hat{x}, \hat{y}), \\
\hat{\varphi}_{3p+1}(\hat{x}, \hat{y}), \dots, \hat{\varphi}_{(2+3p+p^2)/2}(\hat{x}, \hat{y}) &= \hat{N}_{0,0}(\hat{x}, \hat{y}) \dots \hat{N}_{i,j}(\hat{x}, \hat{y}) \quad 0 \leq i + j \leq p - 3.
\end{aligned} \tag{3.59}$$

Im folgenden wird der globale Finite-Elemente-Raum $\mathcal{S}^{\mathbf{p}}(\Omega)$ definiert. Die Bestimmung der Anzahl der Freiheitsgrade erfordert die Definition eines Verteilungsvektors für Polynomgrade. Jedem Element $T \in \mathcal{T} = \{T_j\}_{j=1}^{M(\mathcal{T})}$ ist ein Polynomgrad p_T zugeordnet. Diese Polynomgrade werden in dem Vektor $\mathbf{p} = (p_1, \dots, p_{M(\mathcal{T})})$ zusammengefaßt. Mit I wird im weiteren Verlauf dieser Arbeit die Indexmenge der globalen Freiheitsgrade bezeichnet. Die Freiheitsgrade von $\mathcal{S}^{\mathbf{p}}(\Omega)$ setzen sich aus drei verschiedenen Anteilen zusammen. In der folgenden Aufzählung ist bereits die Dimension des Problems berücksichtigt.

1. Jedem Knoten sind zwei Freiheitsgrade zugeordnet. Es gibt $2\#\mathcal{N}$ globale Knotenfreiheitsgrade. Die Indexmenge der Knotenbasisfunktionen wird im folgenden mit I_1 bezeichnet.
2. Jeder Kante $E \in \mathcal{E}$ wird ein Kantenpolynomgrad p_E zugeordnet. Falls $E = T_1 \cap T_2$, wird $p_E = \max\{p_{T_1}, p_{T_2}\}$ gesetzt. Die Anzahl der Freiheitsgrade zu der Kante $E \in \mathcal{E}$ ist durch p_E vorgegeben. Für $p_E > 1$ sind der Kante E $2(p_E - 1)$ Freiheitsgrade zugeordnet. Die globalen Kantenfreiheitsgrade setzen sich aus den Freiheitsgraden aller Kanten $E \in \mathcal{E}$ zusammen. Es gibt demnach $\sum_{E \in \mathcal{E}} 2(p_E - 1)$ globale Kantenfreiheitsgrade. Die Indexmenge der globalen Kantenfreiheitsgrade wird mit I_2 bezeichnet.
3. Der Vektor \mathbf{p} bestimmt für jedes Element $T \in \mathcal{T}$ die Anzahl interner Basisfunktionen. Für $p_T > 2$ gehören zu dem Element T $(p_T - 1)(p_T - 2)$ interne Freiheitsgrade. Es gibt $\sum_{T \in \mathcal{T}} (p_T - 1)(p_T - 2)$ globale interne Freiheitsgrade.

Definition 3.4 ($\mathcal{S}^{\mathbf{p}}(T)$). *Auf dem Element T ist der Polynomgrad p_T gegeben. Die Kantenpolynomgrade dieses Elements sind mit p_{E_1} , p_{E_2} und p_{E_3} bezeichnet. Vorausgesetzt wird, daß die Basisfunktionen auf dem Referenzelement \hat{T} entsprechend (3.59) bis zum Polynomgrad $p = \max\{p_T : T \in \mathcal{T}\}$ berechnet sind. Eine Basis des Raumes $\mathcal{S}^{\mathbf{p}}(T)$ ist gegeben durch*

$$\{\boldsymbol{\phi}_{\pi(T,1)}, \dots, \boldsymbol{\phi}_{\pi(2N(T))}\} := \{\varphi_{T,1}\mathbf{e}_1, \varphi_{T,1}\mathbf{e}_2, \dots, \varphi_{T,N(T)}\mathbf{e}_1, \varphi_{T,N(T)}\mathbf{e}_2\} \tag{3.60}$$

mit $N(T) = p_{E_1} + p_{E_2} + p_{E_3} + (p_T - 1)(p_T - 2)/2$. Die Funktion $\pi : \{1, \dots, 2N(T)\} \rightarrow I$ bildet die lokale Nummerierung der Freiheitsgrade in die Indexmenge I ab. Es gilt

$$\varphi_{T,1}(x, y) = \hat{\varphi}_1(F_T^{-1}(x, y)), \quad \varphi_{T,2}(x, y) = \hat{\varphi}_2(F_T^{-1}(x, y)), \quad \varphi_{T,3}(x, y) = \hat{\varphi}_3(F_T^{-1}(x, y)),$$

$$\begin{aligned}\varphi_{T,3+i}(x, y) &= \hat{\varphi}_{3+i}(F_T^{-1}(x, y)) && \text{für } i = 1, \dots, p_{E_1} - 1, \\ \varphi_{T,2+p_{E_1}+i}(x, y) &= \hat{\varphi}_{2+p+i}(F_T^{-1}(x, y)) && \text{für } i = 1, \dots, p_{E_2} - 1, \\ \varphi_{T,1+p_{E_1}+p_{E_2}+i}(x, y) &= \hat{\varphi}_{1+2p+i}(F_T^{-1}(x, y)) && \text{für } i = 1, \dots, p_{E_3} - 1,\end{aligned}$$

$$\varphi_{T,p_{E_1}+p_{E_2}+p_{E_3}+i}(x, y) = \hat{\varphi}_{3p+i}(F_T^{-1}(x, y)) \quad \text{für } i = 1, \dots, (p_T - 1)(p_T - 2)/2.$$

Um für den aus den Räumen $\mathcal{S}^p(T)$ konstruierten globalen Finite-Elemente-Raum Stetigkeit zu garantieren, werden Kantenbasisfunktionen ungerader Ordnung mit dem Faktor -1 multipliziert, falls die entsprechende Elementkante von einem Knoten höherer Nummerierung zu einem Knoten niedrigerer Nummerierung verläuft [18].

Definition 3.5 (Finite-Elemente-Raum $\mathcal{S}^p(\Omega)$). Der globale Finite-Elemente-Raum ist definiert durch

$$\mathcal{S}^p(\Omega) = \{\mathbf{u} \in H^1(\Omega)^2 : \forall T \in \mathcal{T} \mathbf{u}|_T \in \mathcal{S}^p(T)\}. \quad (3.61)$$

Für Funktionen mit Gleitrandbedingungen werden folgende Notationen eingeführt,

$$\mathcal{S}_M^p(\Omega) = \{\mathbf{u} \in H^1(\Omega)^2 : \forall T \in \mathcal{T} \mathbf{u}|_T \in \mathcal{S}^p(T), \mathbf{M} \cdot \mathbf{u} = \mathbf{w} \text{ auf } \Gamma_D\}, \quad (3.62)$$

$$\mathcal{S}_0^p(\Omega) = \{\mathbf{u} \in H^1(\Omega)^2 : \forall T \in \mathcal{T} \mathbf{u}|_T \in \mathcal{S}^p(T), \mathbf{M} \cdot \mathbf{u} = \mathbf{0} \text{ auf } \Gamma_D\}. \quad (3.63)$$

Dabei sind \mathbf{M} und \mathbf{w} durch (2.21) und (2.22) definiert. Ist der Polynomgrad auf allen Elementen gleich, wird auch $\mathcal{S}^p(\Omega)$ geschrieben.

Die Dimension des globalen Finite-Elemente-Raumes ergibt sich aus den oben genannten Punkten zu

$$\dim \mathcal{S}^p(\Omega) = 2N = 2\#\mathcal{N} + \sum_{E \in \mathcal{E}} 2(p_E - 1) + \sum_{T \in \mathcal{T}} (p_T - 1)(p_T - 2), \quad (3.64)$$

falls $p_T \geq 2$ für alle $T \in \mathcal{T}$. Die globalen Basisfunktionen $\{\phi_1, \dots, \phi_{2N}\}$ definieren die Funktionen $\{\varphi_1, \dots, \varphi_N\}$ durch

$$\{\varphi_1 \mathbf{e}_1, \varphi_1 \mathbf{e}_2, \dots, \varphi_N \mathbf{e}_1, \varphi_N \mathbf{e}_2\} := \{\phi_1, \phi_2, \dots, \phi_{2N-1}, \phi_{2N}\}.$$

Dabei sind \mathbf{e}_1 und \mathbf{e}_2 die Einheitsvektoren in \mathbb{R}^2 .

3.2.2 Ortsdiskretes Modellproblem

Die Ortsdiskretisierung erfolgt durch die Finite-Elemente-Methode. In Problem 5 werden die Räume X und X_M in der Variationsformulierung (3.3) durch die endlich-dimensionalen Räume \mathcal{S}^p und \mathcal{S}_M^p , wie in Abschnitt 3.2.1 beschrieben, ersetzt. Die diskreten Verschiebungen $\mathbf{u}_{h,k-1+\theta}$ zu den diskreten Zeitpunkten $t_{k-1+\theta}$, $k = 0, \dots, n$, haben die Basisdarstellung

$$\mathbf{u}_{h,k-1+\theta} = \sum_{\ell \in I} U_{k-1+\theta}^\ell \phi_\ell, \quad (3.65)$$

wobei I die in Abschnitt 3.2.1 beschriebene Indexmenge der globalen Freiheitsgrade ist. Die Komponenten der Verschiebung werden in dem Vektor $\mathbf{U}_{k-1+\theta} =$

$(U_{k-1+\theta}^1, \dots, U_{k-1+\theta}^{2N})$ zusammengefaßt. Das Fließgesetz (2.42) ist in jedem Quadraturpunkt zu erfüllen. Zusätzlich müssen die Anfangswerte in jedem Zeitschritt das quasistatische Kräftegleichgewicht in der schwachen Formulierung erfüllen. Daher sind die Spannung $\boldsymbol{\sigma}_{h,k}$ und die plastische Variable $\boldsymbol{\alpha}_{h,k}$ für jedes Element $T \in \mathcal{T}$ in den Quadraturpunkten der Flächenintegration zu speichern. Zusätzlich sind die Spannung und die interne Spannung für jedes Element in den Quadraturpunkten der Kanten sowie für jedes Element in den Knoten zu speichern. Die elementweise Speicherung der Kanten- und Knotenwerte ist erforderlich, da Spannung und interne Spannung nicht stetig sind. Die in Kapitel 4 verwendeten Fehlerschätzer benötigen Anfangswerte auf den Kanten. Die in Abschnitt 3.3.2 dargestellte Interpolation der Anfangswerte des Finite-Elemente-Netzes benötigt Werte der Spannung und internen Spannung sowohl in den Knoten als auch in den Quadraturpunkten der Kanten. Neben der Spannung und der internen Spannung sind weitere Größen punktweise zu speichern. Diese Größen werden in den Fehlerschätzern des Kapitels 4 eingeführt. Es handelt sich um Ableitungen der Spannung und der internen Spannung.

Problem 6 (*Diskretes Modellproblem 3*). Finde das Verschiebungsfeld $\mathbf{u}_{h,k-1+\theta} \in \mathcal{S}_M^p(\Omega)$ zu den diskreten Zeitpunkten $t_{k-1+\theta}$ für $k = 1, \dots, n$, so daß für alle $\mathbf{v}_h \in \mathcal{S}^p(\Omega)$

$$\begin{aligned} \int_{\Omega} \mathbf{S}(\mathbf{u}_{h,k-1+\theta}, \mathbf{u}_{h,k-1}, \boldsymbol{\sigma}_{h,k-1}, \boldsymbol{\alpha}_{h,k-1}) : \boldsymbol{\epsilon}(\mathbf{v}_h) \, d\mathbf{x} = \\ \int_{\Omega} \mathbf{f}(t_{k-1+\theta}) \cdot \mathbf{v}_h \, d\mathbf{x} + \int_{\Gamma_N} \mathbf{g}(t_{k-1+\theta}) \cdot \mathbf{v}_h \, d\mathbf{s} \end{aligned} \quad (3.66)$$

mit den Anfangsbedingungen

$$\mathbf{u}_h(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}), \boldsymbol{\sigma}_h(\mathbf{x}, 0) = \boldsymbol{\sigma}_0(\mathbf{x}), \boldsymbol{\alpha}_h(\mathbf{x}, 0) = \boldsymbol{\alpha}_0(\mathbf{x}). \quad (3.67)$$

Aus der Verschiebung $\mathbf{u}_{h,k-1+\theta}$ und den Variablen $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$, $\boldsymbol{\alpha}_{h,k-1}$ zum Zeitpunkt t_{k-1} berechnen sich Spannung und interne Spannung nach den Sätzen 3.1, 3.2 bzw. 3.3. Im Beispiel der kinematischen Verfestigung gilt

$$\begin{aligned} \boldsymbol{\sigma}_{h,k-1+\theta} &= \mathbf{S}(\mathbf{u}_{h,k-1+\theta}, \mathbf{u}_{h,k-1}, \boldsymbol{\sigma}_{h,k-1}, \boldsymbol{\alpha}_{h,k-1}), \\ \boldsymbol{\alpha}_{h,k-1+\theta} &= \mathbf{R}(\mathbf{u}_{h,k-1+\theta}, \mathbf{u}_{h,k-1}, \boldsymbol{\sigma}_{h,k-1}, \boldsymbol{\alpha}_{h,k-1}). \end{aligned}$$

Nach (2.40) sind die Werte zum Zeitpunkt $t_{k-1+\theta}$ als lineare Interpolation der Werte zu den Zeitpunkten t_{k-1} und t_k definiert. Das heißt, die Variablen zum Zeitpunkt t_k sind

$$\begin{aligned} \mathbf{u}_{h,k} &= \frac{1}{\theta}(\mathbf{u}_{h,k-1+\theta} - (1-\theta)\mathbf{u}_{h,k-1}), & \boldsymbol{\sigma}_{h,k} &= \frac{1}{\theta}(\boldsymbol{\sigma}_{h,k-1+\theta} - (1-\theta)\boldsymbol{\sigma}_{h,k-1}), \\ \boldsymbol{\alpha}_{h,k} &= \frac{1}{\theta}(\boldsymbol{\alpha}_{h,k-1+\theta} - (1-\theta)\boldsymbol{\alpha}_{h,k-1}). \end{aligned}$$

3.2.3 Verallgemeinerte Randbedingungen

Dieser Abschnitt beschreibt die numerische Realisierung von Dirichlet-Randbedingungen und Gleitrandbedingungen für Ansatzfunktionen höherer Ordnung. Für

eine Elementkante $\gamma \subset \Gamma_D$ mit Endpunkten (x_1, y_1) und (x_2, y_2) liegt eine Dirichlet-Randbedingung vor, wenn auf γ die Verschiebung vorgegeben ist, $\mathbf{u}_h|_\gamma = \mathbf{u}_D$. Eine Verschiebungsrandbedingung liegt vor, wenn auf γ die Normalenkomponente der Verschiebung null ist, $\mathbf{u}_h|_\gamma \cdot \mathbf{n} = 0$. Es werden Gleitrandbedingungen parallel zu kartesischen Koordinatenachsen zugelassen. Im folgenden wird die Situation eines zweidimensionalen Problems dargestellt, $d = 2$. Die globalen Nummerierungen der Basisfunktionen auf der Kante γ gehören zur Indexmenge $I(\gamma)$. Unterschieden werden lineare Basisfunktionen (Indexmenge $I_1(\gamma)$) und Basisfunktionen höherer Ordnung mit Träger auf der Kante γ (Indexmenge $I_2(\gamma)$). Es gilt $I(\gamma) = I_1(\gamma) \cup I_2(\gamma)$. Es wird eine Matrix $M(x, y)$ definiert durch

$$M(x, y) = \begin{cases} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} & \text{falls } (x, y) \text{ auf dem Dirichlet-Rand,} \\ \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} & \text{falls } (x, y) \text{ auf einem Gleitrand parallel zur x-Achse,} \\ \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & \text{falls } (x, y) \text{ auf einem Gleitrand parallel zur y-Achse,} \end{cases} \quad (3.68)$$

und ein Vektor $w(x, y)$ durch

$$\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{cases} \begin{pmatrix} u_{D,1}(x, y) \\ u_{D,2}(x, y) \end{pmatrix} & \text{falls } (x, y) \text{ auf dem Dirichlet-Rand,} \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \text{falls } (x, y) \text{ auf dem Gleitrand.} \end{cases} \quad (3.69)$$

Zunächst werden die Freiheitsgrade zu den linearen Basisfunktionen betrachtet. In den Endpunkten (x_1, y_1) und (x_2, y_2) der Elementkante γ gilt für die zugehörigen Freiheitsgrade der Verschiebung,

$$M \begin{pmatrix} U^{2j_k-1} \\ U^{2j_k} \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \quad k = 1, 2. \quad (3.70)$$

Hier sind j_k für $k = 1, 2$ die globalen Nummern der beiden Knoten, die zur Kante γ gehören. Nun werden Bedingungen für die Freiheitsgrade der Verschiebung abgeleitet, die zu Kantenbasisfunktionen auf der Kante γ gehören. Der Verschiebungsvektor mit Komponenten zu Kantenbasisfunktionen ist $\bar{\mathbf{U}} = \{U^i : i \in I_2(\gamma)\}$. Die Minimierung

$$\min_{\bar{\mathbf{U}}} \left\| \sum_{i \in I_2(\gamma)} U^i M \phi_i(x, y) - \bar{w}(x, y) \right\|_{2,\gamma}^2 \quad (3.71)$$

mit

$$\bar{w}(x, y) = \begin{cases} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} - \sum_{j \in I_1(\gamma)} U^j \phi_j(x, y) & \text{falls } (x, y) \text{ auf dem Dirichlet-Rand,} \\ \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} & \text{falls } (x, y) \text{ auf dem Gleitrand} \end{cases} \quad (3.72)$$

ergibt das lineare Gleichungssystem

$$\sum_{i \in I_2(\gamma)} \left(\int_{\gamma} (M \phi_i(x, y)) \cdot (M \phi_k(x, y)) \, ds \right) U^i = \int_{\gamma} (M \phi_k(x, y)) \cdot \bar{w}(x, y) \, ds \quad k \in I_2(\gamma). \quad (3.73)$$

Die Matrizen in den Gleichungssystemen (3.70) und (3.73) werden zu einer globalen Matrix \mathbf{B} assembliert, die zugehörigen rechten Seiten zu einer globalen rechten Seite \mathbf{w} . Die verallgemeinerte Randbedingung für den Verschiebungsvektor \mathbf{U} ist damit

$$\mathbf{B}\mathbf{U} = \mathbf{w}. \quad (3.74)$$

Die Menge der Freiheitsgrade, die nicht durch Randbedingungen festgelegt sind, wird mit \tilde{I} bezeichnet.

3.2.4 Iterative Lösung mit dem Newton-Verfahren

Das quasistatische Kräftegleichgewicht (3.66) ist ein nichtlineares Variationsproblem, das mit einem Newton-Verfahren in Kombination mit der Finite-Elemente-Methode gelöst wird. Im Zeitschritt k sind die Komponenten $(U_{h,k-1+\theta}^1, \dots, U_{h,k-1+\theta}^{2N})$ des Verschiebungsvektors $\mathbf{U}_{h,k-1+\theta}$ zu bestimmen, so daß für $p = 1, \dots, dN$ das Kräftegleichgewicht

$$\begin{aligned} F_p := & \int_{\Omega} \mathbf{S}(\mathbf{U}_{h,k-1+\theta}, \mathbf{U}_{h,k-1}, \boldsymbol{\sigma}_{h,k-1}, \boldsymbol{\alpha}_{h,k-1}) : \boldsymbol{\epsilon}(\phi_p) \, d\mathbf{x} - \int_{\Omega} \mathbf{f}(t_{k-1+\theta}) \cdot \phi_p \, d\mathbf{x} \\ & - \int_{\Gamma_N} \mathbf{g}(t_{k-1+\theta}) \cdot \phi_p \, ds = 0 \end{aligned} \quad (3.75)$$

erfüllt ist und die Verschiebung einer verallgemeinerten Randbedingung genügt,

$$\mathbf{B}\mathbf{U}_{h,k-1+\theta} = \mathbf{w}. \quad (3.76)$$

Die Komponenten F_p bestehen aus einem Anteil Q_p , der von der unbekanntem Verschiebung abhängt, und einem Anteil P_p , der in einem Iterationsverfahren nur einmal berechnet werden muß,

$$Q_p = \int_{\Omega} \mathbf{S}(\mathbf{U}_{h,k-1+\theta}, \mathbf{U}_{h,k-1}, \boldsymbol{\sigma}_{h,k-1}, \boldsymbol{\alpha}_{h,k-1}) : \boldsymbol{\epsilon}(\phi_p) \, d\mathbf{x}, \quad (3.77)$$

$$P_p = \int_{\Omega} \mathbf{f}(t_{k-1+\theta}) \cdot \phi_p \, d\mathbf{x} + \int_{\Gamma} \mathbf{g}(t_{k-1+\theta}) \cdot \phi_p \, ds. \quad (3.78)$$

Die Komponenten werden in den Vektoren $\mathbf{F} \in \mathbb{R}^{2N}$, $\mathbf{Q} \in \mathbb{R}^{2N}$ und $\mathbf{P} \in \mathbb{R}^{2N}$ zusammengefaßt.

$$\mathbf{F} = (F_1, \dots, F_{2N}), \quad \mathbf{Q} = (Q_1, \dots, Q_{2N}), \quad \mathbf{P} = (P_1, \dots, P_{2N}). \quad (3.79)$$

Der Vektor Q_p ist stückweise differenzierbar. Es ist daher möglich, eine globale Steifigkeitsmatrix $DF \in \mathbb{R}^{2N \times 2N}$ folgendermaßen zu definieren,

$$(DF(U_{h,k-1+\theta}^1, \dots, U_{h,k-1+\theta}^{2N}))_{pq} := \frac{\partial Q_p(U_{h,k-1+\theta}^1, \dots, U_{h,k-1+\theta}^{2N})}{\partial U_{h,k-1+\theta}^q}. \quad (3.80)$$

Die Finite-Elemente-Methode (FEM) assembliert die globale Steifigkeitsmatrix DF (3.80) und den Vektor der rechten Seite F (3.79) aus Elementbeiträgen. Der Elementbeitrag zur globalen Steifigkeitsmatrix nennt sich lokale Steifigkeitsmatrix M . Die Berechnung der lokalen Steifigkeitsmatrix und der lokalen rechten Seite für das Element T im Iterationsschritt i des Newton-Verfahrens hängt von den Komponenten

$$(U_{h,k-1+\theta}^{i-1,\pi(T,1)}, \dots, U_{h,k-1+\theta}^{i-1,\pi(T,2N(T))}) \quad (3.81)$$

der Lösung im letzten Iterationsschritt ab. Es gelten die Definitionen:

Definition 3.6 (Lokale rechte Seite). Für $p \in \{1, \dots, 2N(T)\}$ ist die lokale rechte Seite im Iterationsschritt i definiert durch

$$F_{T,p}^i = Q_{T,p}^i - P_{T,p}^i \quad (3.82)$$

mit

$$Q_{T,p}^i = \int_T \mathbf{S}((U_{h,k-1+\theta}^{i-1,\pi(T,1)}, \dots, U_{h,k-1+\theta}^{i-1,\pi(T,2N(T))}), (U_{h,k-1}^{\pi(T,1)}, \dots, U_{h,k-1}^{\pi(T,2N(T))}), \boldsymbol{\sigma}_{h,k-1}, \boldsymbol{\alpha}_{h,k-1}) : \boldsymbol{\epsilon}(\boldsymbol{\phi}_{\pi(T,p)}) \mathbf{d}\mathbf{x}, \quad (3.83)$$

$$P_{T,p}^i = \int_T \mathbf{f}(t_{k-1+\theta}) \cdot \boldsymbol{\phi}_{\pi(T,p)} \mathbf{d}\mathbf{x} + \int_{\Gamma} \mathbf{g}(t_{k-1+\theta}) \cdot \boldsymbol{\phi}_{\pi(T,p)} \mathbf{d}\mathbf{s}. \quad (3.84)$$

Definition 3.7 (Lokale Steifigkeitsmatrix). Die Komponente der lokalen Steifigkeitsmatrix zum Element T , M_{pq}^i ($p, q \in \{1, \dots, 2N(T)\}$), ist im Iterationsschritt i definiert durch die Ableitung von (3.83) nach den Verschiebungskomponenten,

$$M_{pq}((U_{h,k-1+\theta}^{i-1,\pi(T,1)}, \dots, U_{h,k-1+\theta}^{i-1,\pi(T,2N(T))})) = \frac{Q_{T,p}((U_{h,k-1+\theta}^{i-1,\pi(T,1)}, \dots, U_{h,k-1+\theta}^{i-1,\pi(T,2N(T))}))}{\partial U_{h,k-1+\theta}^{\pi(T,p)}}. \quad (3.85)$$

Bei der Berechnung der lokalen Steifigkeitsmatrix wird berücksichtigt, daß der Integrand des lokalen Vektors der rechten Seite (3.83) in der elastischen und der plastischen Phase nach den Verschiebungskomponenten differenzierbar ist. Verläuft eine elastisch-plastische Grenze durch das Element, existiert die Ableitung dort nicht, da diese Grenze eine Nullmenge ist, es existiert aber das Lebesgue-Integral. Die Unterscheidung nach elastischen und plastischen Bereichen wird abhängig vom Vektor (3.81) getroffen.

Mit der Definition

$$\mathbf{A}_{h,k-1+\theta}^i = \boldsymbol{\epsilon} \left(\frac{(U_{h,k-1+\theta}^{i-1,\pi(T,1)}, \dots, U_{h,k-1+\theta}^{i-1,\pi(T,2N(T))}) - (U_{h,k-1}^{\pi(T,1)}, \dots, U_{h,k-1}^{\pi(T,2N(T))})}{\theta \Delta t_k} \right) + \mathbb{C}^{-1} \frac{\boldsymbol{\sigma}_{h,k-1}}{\theta \Delta t_k}$$

ergeben sich für die lokale Steifigkeitsmatrix und die lokale rechte Seite abhängig vom betrachteten Modell die Ausdrücke der folgenden Unterabschnitte.

Perfekte Viskoplastizität

Mit Satz 3.1 und Definition 3.6 erhält man im Fall der perfekten Viskoplastizität für den lokalen Vektor der rechten Seite

$$\begin{aligned} Q_p^i &= \\ &\int_T ((\lambda + \mu) \operatorname{tr}(\theta \Delta t \mathbf{A}_{h,k-1+\theta}^i) \mathbf{I} \\ &+ 2\mu(1 - \max\{0, \frac{1}{1 + \Gamma_1} (1 - \frac{\Gamma_2}{2\mu |\operatorname{dev} \theta \Delta t \mathbf{A}_{h,k-1+\theta}^i|})\}) \operatorname{dev} \theta \Delta t \mathbf{A}_{h,k-1+\theta}^i) : \boldsymbol{\epsilon}(\boldsymbol{\phi}_p) d\mathbf{x}. \end{aligned}$$

Die lokale Steifigkeitsmatrix im Iterationsschritt i ist mit Definition 3.7

$$\begin{aligned} M_{pq}^i &= \\ &\int_T \left((\lambda + \mu) \operatorname{tr} \boldsymbol{\epsilon}(\boldsymbol{\phi}_p) \operatorname{tr} \boldsymbol{\epsilon}(\boldsymbol{\phi}_q) + 2\mu(1 - \max\{0, \frac{1}{1 + \Gamma_1} (1 - \frac{\Gamma_2}{2\mu |\operatorname{dev} \theta \Delta t_k \mathbf{A}_{h,k-1+\theta}^i|})\}) \right. \\ &\times \left. \left(1 + \frac{\Gamma_2}{(2\mu |\operatorname{dev} \theta \Delta t \mathbf{A}_{h,k-1+\theta}^i| - \Gamma_2) |\operatorname{dev} \theta \Delta t_k \mathbf{A}_{h,k-1+\theta}^i|^2} \right) \right) \\ &\times \operatorname{dev} \boldsymbol{\epsilon}(\boldsymbol{\phi}_p) : \operatorname{dev} \boldsymbol{\epsilon}(\boldsymbol{\phi}_q) d\mathbf{x}. \end{aligned}$$

Viskoplastizität mit isotroper Verfestigung

Mit Satz 3.2 und Definition 3.6 erhält man im Fall der isotropen Viskoplastizität für den lokalen Vektor der rechten Seite

$$\begin{aligned} Q_p^i &= \\ &\int_T ((\lambda + \mu) \operatorname{tr}(\theta \Delta t \mathbf{A}_{h,k-1+\theta}^i) \mathbf{I} \\ &+ 2\mu(1 - \max\{0, \frac{1}{1 + \Gamma_1} (1 - \frac{\Gamma_2(\alpha_{h,k})}{2\mu |\operatorname{dev} \theta \Delta t \mathbf{A}_{h,k-1+\theta}^i|})\}) \operatorname{dev} \theta \Delta t \mathbf{A}_{h,k-1+\theta}^i) : \boldsymbol{\epsilon}(\boldsymbol{\phi}_p) d\mathbf{x}. \end{aligned}$$

Die lokale Steifigkeitsmatrix im Iterationsschritt i ist mit Definition 3.7

$$\begin{aligned} M_{pq}^i &= \\ &\int_T \left((\lambda + \mu) \operatorname{tr} \boldsymbol{\epsilon}(\boldsymbol{\phi}_p) \operatorname{tr} \boldsymbol{\epsilon}(\boldsymbol{\phi}_q) + 2\mu(1 - \max\{0, \frac{1}{1 + \Gamma_1} (1 - \frac{\Gamma_2(\alpha_{h,k-1})}{2\mu |\operatorname{dev} \theta \Delta t_k \mathbf{A}_{h,k-1+\theta}^i|})\}) \right. \\ &\times \left. \left(1 + \frac{\Gamma_2(\alpha_{h,k-1})}{(2\mu |\operatorname{dev} \theta \Delta t \mathbf{A}_{h,k-1+\theta}^i| - \Gamma_2(\alpha_{h,k-1})) |\operatorname{dev} \theta \Delta t_k \mathbf{A}_{h,k-1+\theta}^i|^2} \right) \right) \\ &\times \operatorname{dev} \boldsymbol{\epsilon}(\boldsymbol{\phi}_p) : \operatorname{dev} \boldsymbol{\epsilon}(\boldsymbol{\phi}_q) d\mathbf{x}. \end{aligned}$$

Viskoplastizität mit kinematischer Verfestigung

Mit Satz 3.3 und Definition 3.6 erhält man im Fall der Viskoplastizität mit kinematischer Verfestigung für den lokalen Vektor der rechten Seite

$$\begin{aligned} Q_p^i &= \int_T ((\lambda + \mu) \operatorname{tr}(\theta \Delta t \mathbf{A}_{h,k-1+\theta}^i) \mathbf{I} \\ &\quad + 2\mu(1 - \max\{0, \frac{1}{1 + \Gamma_1} (1 - \frac{\Gamma_2}{2\mu |\operatorname{dev} \theta \Delta t_k \mathbf{A}_{h,k-1+\theta}^i - \beta \boldsymbol{\alpha}_{h,k-1}|})\})) \\ &\quad \times \operatorname{dev}(\theta \Delta t_k \mathbf{A}_{h,k-1+\theta}^i - \beta \boldsymbol{\alpha}_{h,k-1})) : \boldsymbol{\epsilon}(\boldsymbol{\phi}_p) d\mathbf{x}. \end{aligned}$$

Die lokale Steifigkeitsmatrix im Iterationsschritt i ist mit Definition 3.7

$$\begin{aligned} M_{pq}^i &= \\ &\int_T \left((\lambda + \mu) \operatorname{tr} \boldsymbol{\epsilon}(\boldsymbol{\phi}_p) \operatorname{tr} \boldsymbol{\epsilon}(\boldsymbol{\phi}_q) \right. \\ &\quad \left. + 2\mu(1 - \max\{0, \frac{1}{1 + \Gamma_1} (1 - \frac{\Gamma_2}{2\mu |\operatorname{dev}(\theta \Delta t_k \mathbf{A}_{h,k-1+\theta}^i - \beta \boldsymbol{\alpha}_{h,k-1})|})\}) \right) \\ &\quad \times \left(1 + \frac{\Gamma_2}{(2\mu |\operatorname{dev}(\theta \Delta t_k \mathbf{A}_{h,k-1+\theta}^i - \beta \boldsymbol{\alpha}_{h,k-1})| - \Gamma_2) |\operatorname{dev}(\theta \Delta t_k \mathbf{A}_{h,k-1+\theta}^i - \beta \boldsymbol{\alpha}_{h,k-1})|^2} \right) \\ &\quad \times \operatorname{dev} \boldsymbol{\epsilon}(\boldsymbol{\phi}_p) : \operatorname{dev} \boldsymbol{\epsilon}(\boldsymbol{\phi}_q) \Big) d\mathbf{x}. \end{aligned}$$

Der folgende Algorithmus definiert das Newton-Verfahren. Vorausgesetzt wird, daß der Vektor P nach (3.78) berechnet ist, und die Menge $\tilde{I} = \{\ell_1, \dots, \ell_Z\}$ der globalen Nummern der Freiheitsgrade bestimmt ist, die nicht durch Randbedingungen festgelegt sind. Außerdem werden vor dem Start des Newton-Verfahrens die Matrix \mathbf{B} und der Vektor \mathbf{w} aus (3.74) berechnet.

Algorithmus 3.8 (Newton-Verfahren).

1. Bestimme den Startvektor $\mathbf{U}_{h,k-1+\theta}^0 = (U_{h,k-1+\theta}^{0,1}, \dots, U_{h,k-1+\theta}^{0,2N}) := \mathbf{U}_{h,k-1}$.
2. Setze $i := 0$ (Anzahl der Iterationen).
3. Berechne $Q(\mathbf{U}_{h,k-1+\theta}^i)$ nach (3.77). Setze $F(\mathbf{U}_{h,k-1+\theta}^i) = Q(\mathbf{U}_{h,k-1+\theta}^i) - P$.
4. Setze $\delta := |F((\mathbf{U}_{h,k-1+\theta}^{0,\ell_1}, \dots, \mathbf{U}_{h,k-1+\theta}^{0,\ell_Z}))|$, $\delta_0 := \delta$ und $\delta_s := \delta$.
5. Durchlaufe die Schritte 6-12 solange $(\delta/\delta_s > 10^{-4})$ oder $\delta < \delta_0$.
6. Berechne $DF(\mathbf{U}_{h,k-1+\theta}^{i,1}, \dots, \mathbf{U}_{h,k-1+\theta}^{i,2N})$ nach (3.80) durch Assemblierung aus Elementbeiträgen.
7. Löse das lineare Gleichungssystem

$$\begin{pmatrix} DF(\mathbf{U}_{h,k-1+\theta}^i) & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{U}_{h,k-1+\theta}^{i+1} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{w} \end{pmatrix} \quad (3.86)$$

$$\mathbf{b} := DF(\mathbf{U}_{h,k-1+\theta}^i) \mathbf{U}_{h,k-1+\theta}^i - F(\mathbf{U}_{h,k-1+\theta}^i). \quad (3.87)$$

8. Berechne $Q(\mathbf{U}_{h,k-1+\theta}^{i+1})$ nach (3.77). Setze $F(\mathbf{U}_{h,k-1+\theta}^{i+1}) = Q(\mathbf{U}_{h,k-1+\theta}^{i+1}) - P$.
9. Setze $\delta_0 = \delta$ und berechne die Norm des Residuums, $\delta = |(F^{\ell_1}, \dots, F^{\ell_z})|$.
10. Setze $i := i + 1$.
11. Falls ($i = 20$ und $\delta < \delta_0$) breche man die Iteration ab.
12. Gehe zu Schritt 5.

In jedem Iterationsschritt des Newton-Verfahrens ist ein lineares Gleichungssystem zu lösen. Lineare Gleichungssysteme können mit direkten Methoden [50, 51] oder iterativen Methoden [34] gelöst werden. In dieser Arbeit wird der direkte Matlab-Löser verwendet [45], der ausnutzt, daß die globale Steifigkeitsmatrix schwachbesetzt ist. Die Implementierung der Finite-Elemente-Methode erfolgte in Matlab 6.5, in Verbindung mit C-Routinen, die als mex-files eingebunden werden. Die Hauptrechenzeit wird im Bereich der getesteten Genauigkeit (bis zu 100000 Freiheitsgrade) für das Berechnen der lokalen Steifigkeitsmatrizen benötigt. Iterative Löser mit linearem Aufwand wären bei noch mehr Freiheitsgraden eine Alternative, da dann der Anteil der Zeit zum Lösen des Gleichungssystems (3.86) an der Gesamtrechenzeit nicht mehr klein ist.

Das Abbruchkriterium für das Newton-Verfahren soll gewährleisten, daß die Iteration unabhängig von der Maschinengenauigkeit des Rechners die Iteration beendet. Die erste Bedingung in Schritt 5 sorgt dafür, daß die Iteration zumindest einmal durchlaufen wird. Die Iteration wird beendet, wenn die Norm des Residuums aufgrund von Rundungsfehlern wieder anwächst. Als Lösung wird die letzte berechnete Verschiebung gewählt, obwohl für diese die Norm des Residuums minimal größer ist als für die vorletzte berechnete Verschiebung. Diese Wahl spielt eine Rolle bei der numerischen Berechnung der Fehlerschätzer nach der Methode der äquilibrierten Residuen und der Patch-Residuen nach Kapitel 4. In diesen Fehlerschätzern wird die zur Verschiebung gehörige Spannung benötigt. In der Spannungsberechnung ist eine Fallunterscheidung durchzuführen, ob das Material in der elastischen oder plastischen Phase ist. Diese Fallunterscheidung ist nach (3.2) und (3.3) von der Verschiebung $\mathbf{U}_{k-1+\theta}$ abhängig. In den impliziten Fehlerschätzern sind Lösbarkeitsbedingungen von lokalen Problemen zu erfüllen. Diese Lösbarkeitsbedingungen können nur dann erfüllt werden, wenn die Fallunterscheidung für die elastische oder plastische Phase in der Lösungsberechnung (Newton-Verfahren) und in den Fehlerschätzern bis auf Rundungsfehler aufgrund derselben Verschiebung getroffen wird. Das ist durch die Wahl der Abbruchbedingung in dem Newton-Verfahren gegeben.

3.3 Adaptiver Algorithmus

Die effiziente numerische Lösung des zeitabhängigen Problems 6 erfordert die Einbeziehung von a posteriori Fehlerschätzern. Die a posteriori Fehleranalyse in dieser Arbeit beschränkt sich auf die Abschätzung des Ortsdiskretisierungsfehlers in jedem Zeitschritt.

Ein a posteriori Fehlerschätzer erfüllt zwei Aufgaben. Die erste ist die Funktion als Abbruchkriterium. Bildet ein a posteriori Fehlerschätzer eine garantierte obere Schranke für einen Fehler, wie z.B. den Spannungsfehler, so kann die Entscheidung getroffen werden, ob die Ortsdiskretisierung in einem Zeitschritt genau genug ist. Falls ja, kann zum nächsten Zeitschritt übergegangen werden, andernfalls ist eine Verfeinerung der Diskretisierung nötig.

Die zweite Aufgabe eines a posteriori Fehlerschätzers besteht darin, adaptive Netzverfeinerungen zu steuern. Das Ziel jeder adaptiven Verfeinerungsstrategie besteht darin, Finite-Elemente-Räume zu generieren, die es ermöglichen, die vom Benutzer vorgegebene Fehlertoleranz mit einer möglichst geringen Anzahl an Freiheitsgraden einzuhalten. Der a posteriori Fehlerschätzer liefert zu diesem Zweck Informationen über die Verteilung des Fehlers auf die Elemente.

Der in dieser Arbeit verwendete Lösungsalgorithmus setzt sich aus folgenden Schritten zusammen.

1. Initialisierung. Dem Algorithmus müssen Materialparameter, Netzdaten, Basisfunktionen und Anfangswerte übergeben werden. Das Zeitdiskretisierungsverfahren und die Zeitschrittweite Δt sind zu wählen. Eine Fehlertoleranz $\delta = \delta(\Delta t)$ für den Ortsdiskretisierungsfehler ist in Abhängigkeit von der Zeitschrittweite zu bestimmen. Bei einer Reduzierung der Zeitschrittweite ist die Toleranz δ zu verringern.
2. In jedem Zeitschritt ist der Algorithmus 3.8 für das Newton-Verfahren aufzurufen. Als Ergebnis wird die Verschiebung geliefert. Zusätzlich kann ein Vektor ausgegeben werden, der Informationen darüber enthält, welche Elemente in der elastischen, plastischen oder gemischt plastisch-elastischen Phase sind. So kann beim Übergang von elastischem zu plastischem Materialverhalten eine Zeitschrittweitenadaption durchgeführt werden.
3. Als Abbruchkriterium und als Verfeinerungsindikator wird ein a posteriori Fehlerschätzer η berechnet. In jedem Zeitschritt $[t_{k-1}, t_k]$ ($1 \leq k \leq n$) ist die Bedingung

$$\frac{C_1 \eta}{\|\boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)}} \leq \delta(\Delta t) \quad (3.88)$$

zu erfüllen. In (3.88) ist die berechnete Konstante in der a posteriori Fehlerabschätzung mit C_1 bezeichnet. Auf der linken Seite der Abschätzung steht eine relative Größe, um ein vergleichbares Abbruchkriterium für alle Zeitschritte zu erhalten (s. Kapitel 5). Ist das Abbruchkriterium (3.88) erfüllt, kann der nächste Zeitschritt berechnet werden oder falls der letzte Zeitschritt erreicht ist, kann das Verfahren beendet werden.

4. Ist das Abbruchkriterium (3.88) nicht erfüllt, ist die Finite-Elemente-Triangulierung zu verfeinern. Hierfür wird eine durch einen a posteriori Fehlerschätzer gesteuerte adaptive Strategie verwendet. In Abschnitt 3.3.1 sind verschiedene Verfeinerungsstrategien dargestellt. Da das Verfahren von Anfangswerten abhängig ist, die punktweise in den Quadraturpunkten gespeichert sind, ist nach einer Verfeinerung der Ortsdiskretisierung eine Interpolation erforderlich,

falls $t_{k-1} > 0$, um die Anfangswerte bezüglich der verfeinerten Triangulierung zu erhalten (Abschnitt 3.3.1). Anschließend ist für den aktuellen Zeitschritt der Algorithmus 3.8 für das Newton-Verfahren erneut aufzurufen.

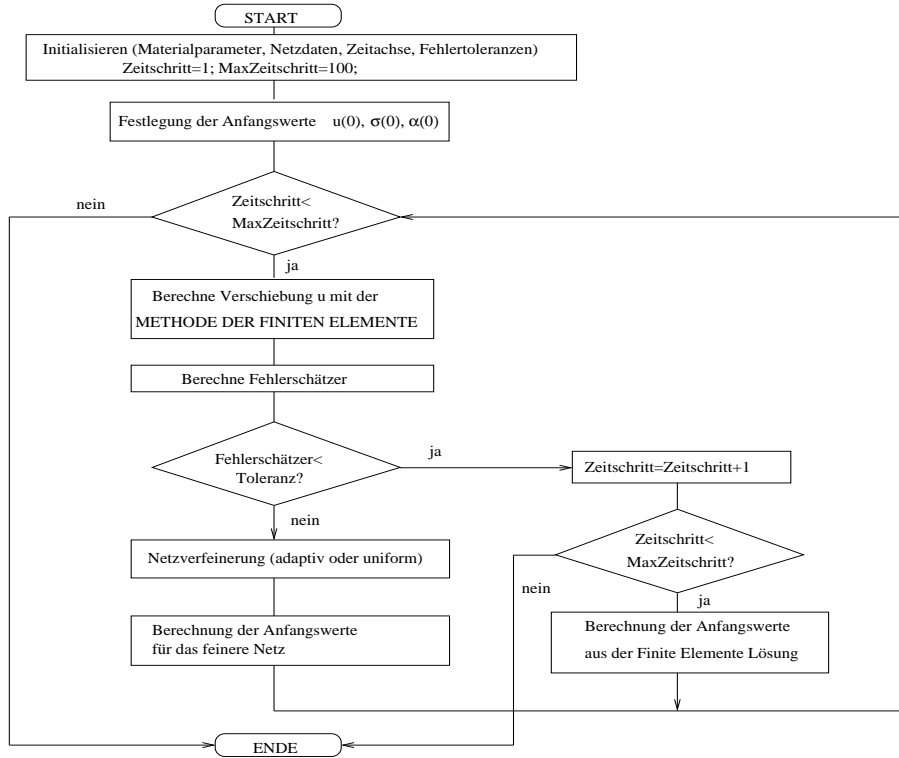


Abbildung 3.1: Adaptiver Algorithmus: Schematischer Ablauf

3.3.1 Verfeinerung der Finite-Elemente-Diskretisierung

Die Fehlerabschätzung $C_1\eta$ ist eine obere Schranke für den Ortsdiskretisierungsfehler in einem Zeitschritt. Einzelheiten hierzu werden in Kapitel 4 ausführlich dargestellt. Wenn das Abbruchkriterium (3.88) nicht erfüllt ist, dann ist die Lösung in diesem Zeitschritt bezüglich einer verfeinerten Triangulierung erneut zu berechnen. Der folgende Satz zeigt, daß der Ortsdiskretisierungsfehler in einem Zeitschritt von der Approximationseigenschaft des Finite-Elemente-Raumes abhängt.

Satz 3.9. *Ist $\sigma_{k-1+\theta}$ die Lösung von Problem 5 im Zeitschritt $[t_{k-1}, t_k]$ und $\sigma_{h,k-1+\theta}$ die Lösung von Problem 6 im Zeitschritt $[t_{k-1}, t_k]$, und werden für beide Probleme die Anfangswerte $\sigma_{h,k-1}$, $\alpha_{h,k-1}$, $\mathbf{u}_{h,k-1}$ vorausgesetzt, dann gilt*

$$\|\sigma_{k-1+\theta} - \sigma_{h,k-1+\theta}\|_{L^2(\Omega)} \leq (2\lambda + 2\mu) \inf_{\mathbf{v}_h \in \mathcal{S}^p(\Omega)} \|\mathbf{u}_{k-1+\theta} - \mathbf{v}_h\|_{H^1(\Omega)}. \quad (3.89)$$

Beweis. Satz 4.4 in Kapitel 4 zeigt mit $c_1 := (2\lambda + 2\mu)$

$$\begin{aligned} \|\sigma_{k-1+\theta} - \sigma_{h,k-1+\theta}\|_{L^2(\Omega)}^2 &\leq c_1 \int_{\Omega} (\sigma_{k-1+\theta} - \sigma_{h,k-1+\theta}) : \boldsymbol{\epsilon}(\mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta}) \, dx \\ &= c_1 \int_{\Omega} (\sigma_{k-1+\theta} - \sigma_{h,k-1+\theta}) : \boldsymbol{\epsilon}(\mathbf{u}_{k-1+\theta} - \mathbf{v}_h + \mathbf{v}_h - \mathbf{u}_{h,k-1+\theta}) \, dx. \end{aligned}$$

Aufgrund der Galerkin-Orthogonalität und mit der Hölder-Ungleichung erhält man

$$\|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} \leq c_1 \|\boldsymbol{\epsilon}(\mathbf{u}_{k-1+\theta} - \mathbf{v}_h)\|_{L^2(\Omega)} \leq c_1 \|\mathbf{u}_{k-1+\theta} - \mathbf{v}_h\|_{H^1(\Omega)}.$$

■

Nach Theorem 4.1 aus [16] ist für eine uniforme Polynomgradverteilung und für quasi-uniforme Netze der Interpolationsfehler des Finite-Elemente-Raumes $\mathcal{S}^p(\Omega)$ gegeben durch

$$\inf_{\mathbf{v}_h \in \mathcal{S}^p(\Omega)} \|\mathbf{v} - \mathbf{v}_h\|_{H^1(\Omega)} \leq c_2 h^\mu p^{-(m-1)} \|\mathbf{v}\|_{H^m(\Omega)} \quad (3.90)$$

wobei $\mu = \min(p, m-1)$ und c_2 eine von der Elementgröße h und dem Polynomgrad p unabhängige Konstante ist, und $\mathbf{u} \in H^m(\Omega)$ gilt. Mit Satz 3.9 folgt

$$\|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} \leq (2\lambda + 2\mu) c_2 h^\mu p^{-(m-1)} \|\mathbf{u}_{k-1+\theta}\|_{H^m(\Omega)}. \quad (3.91)$$

Konvergenz des Spannungsfehlers kann demnach durch Reduzierung der Elementgröße h , durch eine Erhöhung des Polynomgrades p sowie durch eine Kombination beider Verfahren erreicht werden. Diese Methoden werden in folgenden Unterabschnitten definiert und sind als h-, p- und hp-Methode bekannt.

Die Verfeinerung der Ortsdiskretisierung wird durch Verfeinerungsindikatoren gesteuert, die adaptive Netze aus der Kenntnis der numerischen Lösung und des Lastvektors generieren.

Die in dieser Arbeit betrachteten Fehlerschätzer setzen sich aus Elementbeiträgen zusammen. Der Schätzer nach der Methode der äquilibrierten Residuen berechnet für jedes Element $T \in \mathcal{T}$ die Verfeinerungsindikatoren $\eta_{EQ,T}$. Der Patch-Residuen-Schätzer berechnet für jeden Knoten z den Wert $\eta_{L,z}$. Diesem Fehlerschätzer können Elementbeiträge zugewiesen werden durch $\eta_{L,T} = \sum_{z \in \mathcal{N}(T)} \frac{1}{3} \eta_{L,z}$. Der Residuenschätzer bestimmt elementweise Verfeinerungsindikatoren $\eta_{R,T}$.

Die Elementbeiträge η_T der Fehlerschätzer definieren die folgende Verfeinerungsregel.

Definition 3.10 (Verfeinerungsregel). Für einen Parameter $0 \leq \Theta_1 < 1$ wird ein Element $T \in \mathcal{T}$ zur Verfeinerung markiert, falls

$$\Theta_1 \max_{T' \in \mathcal{T}} \eta_{T'} \leq \eta_T. \quad (3.92)$$

h-Verfeinerung

In der h-Version der Finite-Elemente-Methode wird mit einem festen Polynomgrad der Ansatzfunktionen gerechnet. Eine Verfeinerung der Triangulierung erfolgt durch Verringerung der Elementgröße. Die Netzverfeinerung erfolgt nach der Rot-, Blau-, Grün-Verfeinerung zur Vermeidung hängender Knoten. Dabei werden zu halbierende Kanten festgelegt. Ist ein Element aufgrund des Fehlerindikators zur Verfeinerung vorgesehen, werden alle drei Kanten dieses Elements zur Verfeinerung vorgesehen, das Element wird rotverfeinert, d.h in vier Teildreiecke zerlegt. Es werden nun weitere Kanten umliegender Elemente markiert, so daß in jedem Element entweder keine

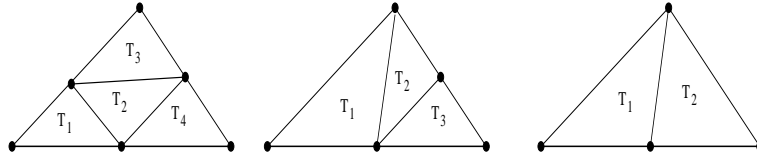


Abbildung 3.2: Rot-, Blau-, und Grün-Verfeinerung eines Elements.

oder mindestens auch die längste Kante halbiert wird. Dabei können blauverfeinerte Elemente auftreten, wobei ein Element in drei Teildreiecke zerlegt wird, oder grünverfeinerte Elemente, bei denen ein Dreieck in zwei Teildreiecke zerlegt wird.

Um Konvergenz zu erreichen, wird das Netz uniform oder adaptiv nach folgendem Algorithmus verfeinert.

Algorithmus 3.11 (h-Verfeinerung). Gegeben ist: Triangulierung \mathcal{T}_0 , $j=0$.

1. Berechne eine Lösung von Problem 6 im Zeitschritt $[t_{k-1}, t_k]$ mit Algorithmus 3.8 zur Triangulierung \mathcal{T}_j .
2. Berechne η_T für alle $T \in \mathcal{T}_j$.
3. Falls

$$\frac{C_1 \eta}{\|\sigma_{h,k-1+\theta}\|_{L^2(\Omega)}} \leq \delta(\Delta t),$$

gehe zum nächsten Zeitschritt oder beende die Rechnung, falls der letzte Zeitschritt erreicht ist. Andernfalls gehe zu Schritt 4.

4. Markiere Elemente nach Verfeinerungsregel 3.10 mit $0 \leq \Theta_1 < 1$.
5. Generiere eine neue reguläre Triangulierung \mathcal{T}_{j+1} nach der Rot-, -Blau-, -Grünverfeinerung, erhöhe j und gehe zu Schritt 1.

Abhängig von dem genauen Wert für Θ_1 ist die Verfeinerung uniform oder adaptiv. Falls $\Theta_1 = 0$, wird jedes Element zur Verfeinerung markiert, die Verfeinerung ist uniform. Für Werte $0 < \Theta_1 < 1$ werden nur bestimmte Elemente zur Verfeinerung markiert, die Verfeinerung ist adaptiv. In numerischen Experimenten des Kapitels 5 wird in diesem Fall der Wert $\Theta = 0.5$ gewählt.

p-Verfeinerung

In der p-Version der Finite-Elemente-Methode bleiben die Elementgrößen unverändert. Eine Erhöhung der Anzahl der Freiheitsgrade erfolgt durch Erhöhung des Polynomgrades der Finite-Elemente-Basisfunktionen.

Algorithmus 3.12 (p-Verfeinerung). Gegeben ist: Die Triangulierung \mathcal{T}_0 , eine Polynomgradverteilung \mathbf{p}_0 , $j = 0$.

1. Berechne eine Lösung von Problem 6 im Zeitschritt $[t_{k-1}, t_k]$ mit Algorithmus 3.8 zur Triangulierung \mathcal{T}_0 mit der Polynomgradverteilung \mathbf{p}_j .

2. Berechne η_T für alle $T \in \mathcal{T}_0$.

3. Falls

$$\frac{C_1 \eta}{\|\boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)}} \leq \delta(\Delta t),$$

gehe zum nächsten Zeitschritt oder beende die Rechnung, falls der letzte Zeitschritt erreicht ist. Andernfalls gehe zu Schritt 4.

4. Markiere Elemente nach Verfeinerungsregel 3.10 mit $\Theta_1 = 0$.

5. Generiere eine neue Polynomgradverteilung, indem auf jedem markierten Element $p_T := p_T + 1$ gesetzt wird. Erhöhe j und gehe zu Schritt 1.

hp-Verfeinerung

Die hp-Version der Finite-Elemente-Methode ist eine geeignete Kombination aus h-Verfeinerung und p-Verfeinerung.

Abschätzung 3.91 zeigt, daß die Konvergenz der p-Methode bei begrenzter Regularität algebraisch ist. Bei hoher Regularität ist die Konvergenzrate der p-Version jedoch sehr groß. Das Ziel der hp-Methode ist, die vorteilhafte Konvergenzrate der p-Methode auch in Situationen zu erhalten, in denen lokale Singularitäten auftreten, indem diese Methode geeignet mit der h-Methode kombiniert wird.

In Arbeiten von M. Ainsworth und Mitarbeitern werden hierzu Verfahren entwickelt, die die lokale Regularität der Lösung abschätzen und in Abhängigkeit davon Elemente, die von einem Verfeinerungsindikator zur Verfeinerung vorgesehen sind, entweder h- oder p-verfeinern [5, 6].

Das Prinzip der in dieser Arbeit verwendeten hp-Methode besteht darin, den Einfluß von lokalen Singularitäten durch eine adaptive h-Methode zu verringern, und diese Methode mit einer uniformen p-Methode zu kombinieren.

Algorithmus 3.13 (hp-Verfeinerung). Gegeben ist: Die Triangulierung \mathcal{T}_0 , eine Polynomgradverteilung \mathbf{p}_0 , $j = 0$.

1. Berechne eine Lösung von Problem 6 im Zeitschritt $[t_{k-1}, t_k]$ mit Algorithmus 3.8 zur Triangulierung \mathcal{T}_j mit der Polynomgradverteilung \mathbf{p}_j .

2. Berechne η_T für alle $T \in \mathcal{T}_k$.

3. Falls

$$\frac{C_1 \eta}{\|\boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)}} \leq \delta(\Delta t),$$

gehe zum nächsten Zeitschritt oder beende die Rechnung, falls der letzte Zeitschritt erreicht ist. Andernfalls gehe zu Schritt 4.

4. Markiere Elemente nach Verfeinerungsregel 3.10 mit $0 \leq \Theta_1 < 1$.

5. Generiere eine neue reguläre Triangulierung \mathcal{T}_{k+1} nach der Rot-, -Blau-, -Grünverfeinerung, setze $j := j + 1$. Bestimme die Polynomgradverteilung \mathbf{p}_j , indem auf verfeinerten Elementen die Polynomgrade übernommen werden.

6. Markiere Elemente nach Verfeinerungsregel 3.10 mit $\Theta_1 = 0$.
7. Generiere die Polynomgradverteilung \mathbf{p}_j , indem auf jedem markierten Element $p_T := p_T + 1$ gesetzt wird. Gehe zu Schritt 1.

3.3.2 Interpolation von Anfangswerten

Eine h -Verfeinerung für $t_{k-1} > 0$ ist mit einem Transfer der Anfangswerte von der Triangulierung \mathcal{T} auf die feinere Triangulierung $\hat{\mathcal{T}}$ verbunden. Im Zeitschritt $[t_{k-1}, t_k]$ sind die Anfangswerte $\hat{\mathbf{u}}_{h,k-1}$, $\hat{\boldsymbol{\sigma}}_{h,k-1}$ und $\hat{\boldsymbol{\alpha}}_{h,k-1}$ für die h -verfeinerte Triangulierung $\hat{\mathcal{T}}$ aus den Anfangswerten $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$ und $\boldsymbol{\alpha}_{h,k-1}$ der Triangulierung \mathcal{T} zu bestimmen. Es wird ein Verfahren verwendet, das den Zeitschritt $[t_{k-2}, t_{k-1}]$ mitberücksichtigt. Die Verschiebungen $\hat{\mathbf{u}}_{h,k-2}$, $\hat{\mathbf{u}}_{h,k-1}$ in der Basisdarstellung der Triangulierung $\hat{\mathcal{T}}$ werden durch L^2 -Projektion aus der Basisdarstellung der Triangulierung \mathcal{T} bestimmt. Spannungen und interne Spannungen, werden hier als Quadraturpunktdaten bezeichnet. Die Werte der Quadraturpunktdaten zum Zeitpunkt t_{k-2} bezüglich der verfeinerten Triangulierung können auf zwei verschiedene Arten berechnet werden.

In einem ersten Verfahren werden diese Werte durch L^2 -Projektion bestimmt. Dafür wird der Polynomraum $\mathcal{L}^q(T)^{2 \times 2}$ für alle $T \in \mathcal{T}$ definiert durch

$$\mathcal{L}^q(T)^{2 \times 2} = \{v \in L^2(T)^{2 \times 2} : v_{ij} \in \mathcal{P}_q(T) \text{ für } i, j = 1, 2\}. \quad (3.93)$$

Die projizierten Spannungen und internen Spannungen sind gegeben durch

$$\hat{\boldsymbol{\sigma}}_{\hat{h},k-2}|_T = \min_{\mathbf{p} \in \mathcal{L}^{p-1}(T)^{2 \times 2}} \|\mathbf{p} - \boldsymbol{\sigma}_{h,k-2}\|_{L^2(T)} \quad \text{für alle } T \in \mathcal{T}, \quad (3.94)$$

$$\hat{\boldsymbol{\alpha}}_{\hat{h},k-2}|_T = \min_{\mathbf{p} \in \mathcal{L}^{p-1}(T)^{2 \times 2}} \|\mathbf{p} - \boldsymbol{\alpha}_{h,k-2}\|_{L^2(T)} \quad \text{für alle } T \in \mathcal{T}. \quad (3.95)$$

In einem zweiten Verfahren werden die Quadraturpunktdaten zum Zeitpunkt t_{k-2} bezüglich der verfeinerten Triangulierung durch stückweise lineare Interpolation bestimmt. Dazu wird für jedes Element $T \in \mathcal{T}$ eine lokale Triangulierung $\mathcal{T}_{lok}(T)$ definiert. Die Elemente der lokalen Triangulierung sind mit T_{lok} bezeichnet. Die Knoten dieser lokalen Triangulierung setzen sich zusammen aus den Quadraturpunkten der zweidimensionalen Quadratur, aus den Quadraturpunkten der eindimensionalen Quadratur auf den Kanten, sowie den Eckpunkten des Dreiecks T . Abb. 3.3 zeigt für ein Element T die lokale Triangulierung und für den Fall der Rotverfeinerung die Positionen der zu interpolierenden Größen auf einem Element der verfeinerten Triangulierung $\hat{\mathcal{T}}$.

In den Knoten der lokalen Triangulierung $\mathcal{T}_{lok}(T)$ sind die Größen $\boldsymbol{\sigma}_{h,k-2}$, $\boldsymbol{\alpha}_{h,k-2}$ zum Zeitpunkt t_{k-2} gegeben. Ein zu verfeinerndes Element T wird in n Teildreiecke \hat{T} zerlegt, wobei $n = 4$ für Rotverfeinerung, $n = 3$ für Blauverfeinerung und $n = 2$ für Grünverfeinerung. Auf jedem Teildreieck \hat{T} sind Daten in den Punkten (x_ℓ, y_ℓ) zu ermitteln. Das sind die Quadraturpunkte der Flächenintegration des Elements \hat{T} , die Quadraturpunkte auf den Kanten von \hat{T} und die Eckpunkte des Elements

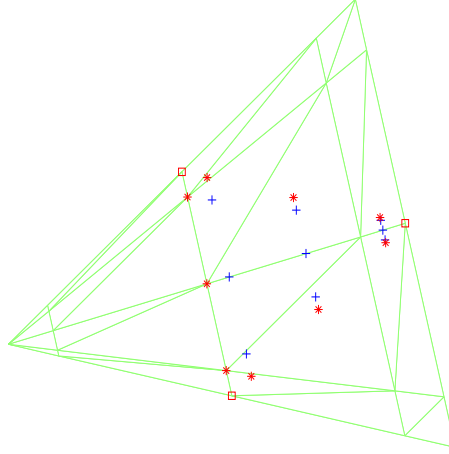


Abbildung 3.3: Dargestellt ist ein Element T mit einer lokalen Triangulierung. Die Knoten dieser Triangulierung sind die Quadraturpunkte der Flächenintegration, die Quadraturpunkte auf den Kanten, sowie die Eckpunkte von T . Kreuze und Quadrate kennzeichnen Interpolationspunkte auf einem Element einer rot-verfeinerten Triangulierung.

\hat{T} . Hat ein Element T_{lok} der lokalen Triangulierung die Eckpunkte P_1, P_2, P_3 , dann liegt der Punkt $P = (x_\ell, y_\ell)$ in dem Element T_{lok} , falls

$$F_1 := \frac{1}{2} \det \begin{bmatrix} 1 & 1 & 1 \\ P_1^T & P_2^T & P^T \end{bmatrix} \geq 0 \quad \wedge \quad F_2 := \frac{1}{2} \det \begin{bmatrix} 1 & 1 & 1 \\ P_2^T & P_3^T & P^T \end{bmatrix} \geq 0 \quad \wedge \\ F_3 := \frac{1}{2} \det \begin{bmatrix} 1 & 1 & 1 \\ P_3^T & P_1^T & P^T \end{bmatrix} \geq 0.$$

Die baryzentrischen Koordinaten des Punktes $P = (x_\ell, y_\ell)$ sind $\alpha = |F_1|/|T_{lok}|$, $\beta = |F_2|/|T_{lok}|$ und $\gamma = |F_3|/|T_{lok}|$. Die lineare Interpolation der Spannung im Punkt P ist durch

$$\begin{bmatrix} \hat{\sigma}_{\hat{h},k-2}^{11}(P) & \hat{\sigma}_{\hat{h},k-2}^{12}(P) & \hat{\sigma}_{\hat{h},k-2}^{21}(P) & \hat{\sigma}_{\hat{h},k-2}^{22}(P) \end{bmatrix} = \\ [\alpha \quad \beta \quad \gamma] \begin{bmatrix} \sigma_{h,k-2}^{11}(P_1) & \sigma_{h,k-2}^{12}(P_1) & \sigma_{h,k-2}^{21}(P_1) & \sigma_{h,k-2}^{22}(P_1) \\ \sigma_{h,k-2}^{11}(P_2) & \sigma_{h,k-2}^{12}(P_2) & \sigma_{h,k-2}^{21}(P_2) & \sigma_{h,k-2}^{22}(P_2) \\ \sigma_{h,k-2}^{11}(P_3) & \sigma_{h,k-2}^{12}(P_3) & \sigma_{h,k-2}^{21}(P_3) & \sigma_{h,k-2}^{22}(P_3) \end{bmatrix} \quad (3.96)$$

gegeben. Entsprechend wird die interne Spannung interpoliert. Durch eine Erhöhung der Anzahl der Quadraturpunkte läßt sich eine Verbesserung der Interpolation erreichen.

Die Startwerte für die Spannung $\hat{\sigma}_{\hat{h},k-1}$ und die interne Spannung $\hat{\alpha}_{\hat{h},k-1}$ bestimmen sich in Abhängigkeit von $\hat{\mathbf{u}}_{\hat{h},k-1}$, $\hat{\mathbf{u}}_{\hat{h},k-2}$, sowie den berechneten Werten $\hat{\sigma}_{\hat{h},k-2}$ und $\hat{\alpha}_{\hat{h},k-2}$ für die Spannung und die interne Spannung nach Satz 3.3 im Fall der kinematischen Verfestigung durch

$$\hat{\sigma}_{\hat{h},k-1} = \mathbf{S}(\hat{\mathbf{u}}_{\hat{h},k-2}, \hat{\mathbf{u}}_{\hat{h},k-2}, \hat{\sigma}_{\hat{h},k-2}, \bar{\alpha}_{\hat{h},k-2}), \quad (3.97)$$

$$\hat{\alpha}_{\hat{h},k-1} = \mathbf{R}(\hat{\mathbf{u}}_{\hat{h},k-1}, \hat{\mathbf{u}}_{\hat{h},k-2}, \hat{\sigma}_{\hat{h},k-2}, \bar{\alpha}_{\hat{h},k-2}). \quad (3.98)$$

Durch diesen Spannungstransfer ist gewährleistet, das zum Zeitpunkt t_{k-1} in den Quadraturpunkten das Fließgesetz erfüllt ist. Entsprechend erhält man nach Satz 3.2 die Spannung $\hat{\sigma}_{\hat{h},k-1}$ und die interne Spannung $\hat{\alpha}_{\hat{h},k-1}$ im Fall der isotropen Verfestigung und mit Satz 3.1 die Spannung $\hat{\sigma}_{\hat{h},k-1}$ bei perfekter Plastizität.

Neben der Spannung und der internen Spannung werden nach demselben Verfahren weitere Größen auf die feinere Triangulierung transferiert. Diese Größen werden in den Fehlerschätzern des Kapitels 4 eingeführt. Es handelt sich um Ableitungen der Spannung und der internen Spannung.

Falls die Daten auf einem bestimmten Element T zum Zeitpunkt t_{k-2} aus einem elastischen Zeitschritt stammen, gilt $\sigma_{h,k-2}|_T \in \mathcal{L}^{p-1}(T)^{2 \times 2}$. Daher werden in diesem Fall die Anfangswerte der feineren Triangulierung durch L^2 -Projektion bestimmt. Die gleiche Überlegung gilt in der Plastizität ($\varrho \rightarrow 0$), falls das Element T zum Zeitpunkt t_{k-2} komplett plastifiziert ist. Falls die Daten zum Zeitpunkt t_{k-2} auf einem bestimmten Element aus einem elastisch-plastischen Zeitschritt stammen, ist eine elementweise L^2 -Projektion der Spannungen und internen Spannungen in einen Polynomraum nicht von Vorteil, um Werte zwischen Quadraturpunkten zu erhalten, da diese Größen eine nichtstetige Ableitung besitzen und schlecht durch elementweise Polynome approximiert werden können. Stattdessen werden diese Daten nach dem beschriebenen Verfahren auf jedem Element stückweise linear interpoliert.

In der p-Version der Finite-Elemente-Methode bleibt das Netz unverändert und der Polynomgrad der Ansatzfunktionen wird lokal oder global erhöht. Da der Aufwand der Berechnungen wesentlich von der Anzahl der Quadraturpunkte abhängt, wird in jeder Verfeinerungsstufe die geringst nötige Anzahl an Quadraturpunkten verwendet. Bei einer Erhöhung des Polynomgrades muß eine Quadratur mit mehr Quadraturpunkten verwendet werden. In den neuen Quadraturpunkten sind die Anfangswerte im Fall $t_{k-1} > 0$ nicht mehr bekannt. Ein Transfer der Anfangswerte erfolgt mit den für die h-Verfeinerung beschriebenen Methoden.

Kapitel 4

A posteriori Fehlerabschätzung in einem Zeitschritt

Gegenstand dieses Kapitels ist die a posteriori Fehlerabschätzung für das elasto-viskoplastische Modellproblem mit residuenbasierten Schätzern.

Die in dieser Arbeit behandelte a posteriori Fehlerabschätzung für die Viskoplastizität ist eine Abschätzung des Ortsdiskretisierungsfehlers in einem Zeitschritt. Das bedeutet, es werden dem ortsdiskreten und dem ortskontinuierlichen Problem in einem Zeitschritt dieselben Anfangswerte für die Verschiebung, die Spannung und die interne Spannung zugrundegelegt. Bezeichnet werden diese Anfangswerte mit $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$ und $\boldsymbol{\alpha}_{h,k-1}$ bzw. $\alpha_{h,k-1}$. Zeitdiskrete aber exakte Lösungen im Ort für die Spannung, die interne Spannung, die internen Variablen und die Verschiebung zum Zeitpunkt $t_{k-1+\theta}$ werden mit $\boldsymbol{\sigma}_{k-1+\theta}$, $\boldsymbol{\alpha}_{k-1+\theta}$ bzw. $\alpha_{k-1+\theta}$, $\boldsymbol{\xi}_{k-1+\theta}$ bzw. $\xi_{k-1+\theta}$ und $\mathbf{u}_{k-1+\theta}$ bezeichnet.

Es wird eine garantierte obere Schranke für den Spannungsfehler hergeleitet, das ist eine Fehlerschranke, bei der in der Abschätzung auftretende Konstanten berechnet werden. Dabei muß Viskoplastizität mit Verfestigung vorausgesetzt werden. Im einzelnen werden folgende Fehlerschätzer betrachtet: Ein residuenbasierter Schätzer nach I. Babuška und W. Rheinboldt [12, 13] in Abschnitt 4.2, der Schätzer nach der Methode der äquilibrierten Residuen nach M. Ainsworth und J.T. Oden [4] in Abschnitt 4.3 und der Patch-Residuen-Schätzer nach C. Carstensen und S.A. Funken [26] in Abschnitt 4.4.

Abschnitt 4.1 beweist Abschätzungen, die eine Anwendung der residuenbasierten Fehlerschätzer auf das elastoviskoplastische Modellproblem ermöglichen und dann garantierte obere Fehlerschranken liefern. Es wird bewiesen, daß sich der Betrag des Verzerrungsfehlers durch den Betrag des Spannungsfehlers abschätzen läßt, falls entweder isotrope oder kinematische Verfestigung gegeben ist. Außerdem wird für diese Fälle gezeigt, daß der Betrag des Fehlers der internen Spannungen durch den Betrag des Spannungsfehlers beschränkt ist. In allen Fällen werden die in den Abschätzungen beteiligten Konstanten angegeben.

In Kapitel 3 wurde gezeigt, daß in jedem Zeitschritt eine analytische Darstellung des Spannungstensors in Abhängigkeit der Verschiebung sowie Daten aus dem letzten Zeitschritt existiert. Aufgrund dieser Tatsache kommen explizite und implizite residuenbasierte Schätzer zur a posteriori Fehlerabschätzung des Modellproblems in

Betracht. Aufbauend auf Arbeiten von R. Verfürth [64], M. Ainsworth und J.T. Oden [4] sowie C. Carstensen und Mitarbeitern [24, 26] zu residuenbasierten Schätzern, wird in den Abschnitten 4.2, 4.3 und 4.4 Zuverlässigkeit und unter Einschränkungen Effizienz der genannten Fehlerschätzer bewiesen.

4.1 Garantierte obere Schranken für den Spannungsfehler

4.1.1 Überblick

Dieser Unterabschnitt gibt einen kurzen Überblick über den Aufbau dieses Kapitels. Die drei zentralen Ergebnisse zur Spannungsfehlerabschätzung mit dem Residuenschätzer, der Methode der äquilibrierten Residuen und der Patch-Residuen-Methode werden angegeben und in den Beweisen auf die Probleme hingewiesen, die in den späteren Abschnitten 4.1-4.4 behandelt werden.

Residuenschätzer

Der Residuenschätzer nach Babuška und Rheinboldt [12, 13] ist ein expliziter Fehlerschätzer, der von der Konstanten der 2. Kornschen Ungleichung (Satz 4.26) und Konstanten c_6, c_7 aus Interpolationsabschätzungen, (4.49), (4.50) bzw. (4.59), (4.60), abhängt. In [26, 25] wird von C. Carstensen und S.A. Funken gezeigt, wie die Konstanten c_6, c_7 abgeschätzt werden können. Die Konstante aus der Kornschen Ungleichung kann einmalig gebietsabhängig durch Lösen eines Eigenwertproblems numerisch bestimmt werden. Mit den in diesem Abschnitt bewiesenen Sätzen 4.15 und 4.16, stellt damit im Fall der h-Verfeinerung der Residuenschätzer eine zuverlässige, berechenbare obere Abschätzung für den Spannungsfehler dar.

Der Beweis für Satz 4.1 besteht aus drei Teilen. Die erste Abschätzung (4.3) ist die Beschränkung des Spannungsfehlers durch das Residuum und wird in [29] gezeigt. Der zweite Teil (4.4) ist Technik des Residuenschätzers. Auf diesen Teil wird in Abschnitt 4.2 ausführlich eingegangen. Der dritte Teil (4.5) ist das Ergebnis der Sätze 4.15 und 4.16.

Satz 4.1. *Für gegebene Anfangswerte $\mathbf{u}_{h,k-1}, \boldsymbol{\sigma}_{h,k-1}, \boldsymbol{\alpha}_{k-1}$ gilt im Zeitschritt $[t_{k-1}, t_k]$ bei Viskoplastizität mit isotroper Verfestigung die Spannungsfehlerabschätzung*

$$\begin{aligned} \|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} &\leq (2\lambda + 2\mu) \left(\frac{1}{2\mu} + \frac{\sqrt{3}\theta\Delta t_k}{\varrho(1 + H^2\sigma_y^2) + \theta\Delta t_k H_1 H^2\sigma_y^2} \right) \\ &\quad \times \frac{1}{\sqrt{c_K}} \left(\sqrt{c_6} \left(\sum_{T \in \mathcal{T}} \eta_T^2 \right)^{1/2} + \sqrt{c_7} \left(\sum_{E \in \mathcal{E}} \eta_E^2 \right)^{1/2} \right). \end{aligned} \quad (4.1)$$

Für gegebene Anfangswerte $\mathbf{u}_{h,k-1}, \boldsymbol{\sigma}_{h,k-1}, \boldsymbol{\alpha}_{k-1}$ gilt im Zeitschritt $[t_{k-1}, t_k]$ bei

Viskoplastizität mit kinematischer Verfestigung die Spannungsfehlerabschätzung

$$\begin{aligned} \|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} &\leq (2\lambda + 2\mu) \left(\frac{1}{2\mu} + \frac{1}{k_1} \sqrt{2 + \frac{1}{2} \left(\frac{4\rho + 2\theta \Delta t_k k_1}{\theta \Delta t_k k_1} \right)^2} \right) \\ &\quad \times \frac{1}{\sqrt{c_K}} \left(\sqrt{c_6} \left(\sum_{T \in \mathcal{T}} \eta_T^2 \right)^{1/2} + \sqrt{c_7} \left(\sum_{E \in \mathcal{E}} \eta_E^2 \right)^{1/2} \right). \end{aligned} \quad (4.2)$$

Dabei ist c_K die Konstante aus der 2. Kornschen Ungleichung (Satz 4.26).

Beweis. Es gilt

$$\frac{1}{2\lambda + 2\mu} \|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)}^2 \leq \langle R(\mathbf{u}_{h,k-1+\theta}), \mathbf{e}_{k-1+\theta} \rangle \quad (4.3)$$

$$\leq c_K \sqrt{c_6} \left(\sum_{T \in \mathcal{T}} \eta_T^2 \right)^{1/2} \|\boldsymbol{\epsilon}(\mathbf{e}_{k-1+\theta})\|_{L^2(\Omega)} + c_K \sqrt{c_7} \left(\sum_{E \in \mathcal{E}} \eta_E^2 \right)^{1/2} \|\boldsymbol{\epsilon}(\mathbf{e}_{k-1+\theta})\|_{L^2(\Omega)} \quad (4.4)$$

$$\begin{aligned} &\leq c_K \left(\sqrt{c_6} \left(\sum_{T \in \mathcal{T}} \eta_T^2 \right)^{1/2} + \sqrt{c_7} \left(\sum_{E \in \mathcal{E}} \eta_E^2 \right)^{1/2} \right) \\ &\quad \times \begin{cases} \left(\frac{1}{2\mu} + \frac{\sqrt{3}\theta \Delta t_k}{\varrho(1+H^2\sigma_y^2) + \theta \Delta t_k H_1 H^2 \sigma_y^2} \right) \|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} & \text{(a)} \\ \left(\frac{1}{2\mu} + \frac{1}{k_1} \sqrt{2 + \frac{1}{2} \left(\frac{4\rho + 2\theta \Delta t_k k_1}{\theta \Delta t_k k_1} \right)^2} \right) \|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} & \text{(b)}. \end{cases} \end{aligned} \quad (4.5)$$

Fall (a) bezieht sich auf isotrope Verfestigung und Fall (b) auf kinematische Verfestigung. \blacksquare

Methode der äquilibrierten Residuen

Die Methode der äquilibrierten Residuen in der in dieser Arbeit dargestellten Form folgt Ainsworth und Oden [4]. Satz 4.2 zeigt, daß diese Methode eine berechenbare obere Schranke für den Spannungsfehler in der Viskoplastizität mit Verfestigung liefert. Der Beweis gliedert sich in drei Teile. Zunächst ist eine Abschätzung für den Spannungsfehler durch das Residuum nötig (4.8), die in [29] gezeigt wurde. Der zweite Teil des Beweises beruht auf den Techniken in [4] und wird in Abschnitt 4.3 ausführlich dargestellt. Der dritte Teil des Beweises ist das Ergebnis dieses Abschnitts, die Beschränkung des Verzerrungsfehlers durch den Spannungsfehler, dargestellt in den Sätzen 4.15 und 4.16.

Satz 4.2. *Für gegebene Anfangswerte $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$, $\boldsymbol{\alpha}_{k-1}$ gilt im Zeitschritt $[t_{k-1}, t_k]$ bei Viskoplastizität mit isotroper Verfestigung die Spannungsfehlerabschätzung*

$$\|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} \leq (2\lambda + 2\mu) \left(\frac{1}{2\mu} + \frac{\sqrt{3}\theta \Delta t_k}{\varrho(1 + H^2\sigma_y^2) + \theta \Delta t_k H_1 H^2 \sigma_y^2} \right) \eta_{EQ}. \quad (4.6)$$

Für gegebene Anfangswerte $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$, $\boldsymbol{\alpha}_{k-1}$ gilt im Zeitschritt $[t_{k-1}, t_k]$ bei Viskoplastizität mit kinematischer Verfestigung die Spannungsfehlerabschätzung

$$\|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} \leq (2\lambda + 2\mu) \left(\frac{1}{2\mu} + \frac{1}{k_1} \sqrt{2 + \frac{1}{2} \left(\frac{4\rho + 2\theta \Delta t_k k_1}{\theta \Delta t_k k_1} \right)^2} \right) \eta_{EQ}. \quad (4.7)$$

Beweis. Es gilt

$$\frac{1}{2\lambda + 2\mu} \|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)}^2 \leq \langle R(\mathbf{u}_{h,k-1+\theta}), \mathbf{e}_{k-1+\theta} \rangle \quad (4.8)$$

$$\leq \eta_{EQ} \|\boldsymbol{\epsilon}(\mathbf{e}_{k-1+\theta})\|_{L^2(\Omega)} \quad (4.9)$$

$$\leq \eta_{EQ} \begin{cases} \left(\frac{1}{2\mu} + \frac{\sqrt{3}\theta\Delta t_k}{\varrho(1+H^2\sigma_y^2)+\theta\Delta t_k H_1 H^2\sigma_y^2} \right) \|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} & \text{(a)} \\ \left(\frac{1}{2\mu} + \frac{1}{k_1} \sqrt{2 + \frac{1}{2} \left(\frac{4\varrho+2\theta\Delta t_k k_1}{\theta\Delta t_k k_1} \right)^2} \right) \|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} & \text{(b)}. \end{cases} \quad (4.10)$$

Fall (a) bezieht sich auf isotrope Verfestigung und Fall (b) auf kinematische Verfestigung. ■

Patch-Residuen-Methode

Die Patch-Residuen-Methode ist in der Arbeit [26] von C. Carstensen und S.A. Funken zur a posteriori Fehlerabschätzung der Poisson-Gleichung veröffentlicht. Satz 4.3 zeigt, daß diese Methode in der Viskoplastizität mit Verfestigung eine berechenbare obere Schranke für den Spannungsfehler liefert. Der erste Teil des Beweises entspricht den vorherigen Sätzen. Auf den zweiten Teil wird in Abschnitt 4.4 ausführlich eingegangen. Dabei wird sich zeigen, daß dieser Fehlerschätzer eine Finite-Elemente-Methode mindestens zweiter Ordnung erfordert, um eine notwendige Bedingung zur Lösbarkeit lokaler Probleme zu erfüllen. Der dritte Teil des Beweises ist identisch zu den vorherigen Sätzen.

Satz 4.3. Für gegebene Anfangswerte $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$, $\boldsymbol{\alpha}_{k-1}$ gilt im Zeitschritt $[t_{k-1}, t_k]$ bei Viskoplastizität mit isotroper Verfestigung die Spannungsfehlerabschätzung

$$\|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} \leq (2\lambda + 2\mu) \left(\frac{1}{2\mu} + \frac{\sqrt{3}\theta\Delta t_k}{\varrho(1 + H^2\sigma_y^2) + \theta\Delta t_k H_1 H^2\sigma_y^2} \right) \eta_L. \quad (4.11)$$

Für gegebene Anfangswerte $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$, $\boldsymbol{\alpha}_{k-1}$ gilt im Zeitschritt $[t_{k-1}, t_k]$ bei Viskoplastizität mit kinematischer Verfestigung die Spannungsfehlerabschätzung

$$\|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} \leq (2\lambda + 2\mu) \left(\frac{1}{2\mu} + \frac{1}{k_1} \sqrt{2 + \frac{1}{2} \left(\frac{4\varrho + 2\theta\Delta t_k k_1}{\theta\Delta t_k k_1} \right)^2} \right) \eta_L. \quad (4.12)$$

Beweis. Es gilt

$$\frac{1}{2\lambda + 2\mu} \|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_2^2 \leq \langle R(\mathbf{u}_{h,k-1+\theta}), \mathbf{e}_{k-1+\theta} \rangle \quad (4.13)$$

$$\leq \eta_L \|\boldsymbol{\epsilon}(\mathbf{e}_{k-1+\theta})\|_{L^2(\Omega)} \quad (4.14)$$

$$\leq \eta_L \begin{cases} \left(\frac{1}{2\mu} + \frac{\sqrt{3}\theta\Delta t_k}{\varrho(1+H^2\sigma_y^2)+\theta\Delta t_k H_1 H^2\sigma_y^2} \right) \|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} & \text{(a)} \\ \left(\frac{1}{2\mu} + \frac{1}{k_1} \sqrt{2 + \frac{1}{2} \left(\frac{4\varrho+2\theta\Delta t_k k_1}{\theta\Delta t_k k_1} \right)^2} \right) \|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} & \text{(b)}. \end{cases} \quad (4.15)$$

Fall (a) bezieht sich auf isotrope Verfestigung und Fall (b) auf kinematische Verfestigung. ■

4.1.2 Grundlegende Ergebnisse der a posteriori Fehleranalyse

Dieser Unterabschnitt bildet die Grundlage für die Anwendung von residuenbasierten a posteriori Fehlerschätzern zur zuverlässigen Spannungsfehlerabschätzung in der Elastoviskoplastizität mit Verfestigung. Die zentralen Ergebnisse sind die Sätze 4.15 und 4.16 zur Abschätzung des Verzerrungsfehlers durch den Spannungsfehler.

Abschätzungen in diesem Abschnitt gelten punktweise. Zur Vereinfachung der Notation wird auf die Darstellung der Abhängigkeit vom Ort verzichtet.

Ein erstes fundamentales Ergebnis für die a posteriori Fehlerabschätzung in der Elastoviskoplastizität ist in dem folgenden Satz von C. Carstensen und S. Müller [29] dargestellt.

Satz 4.4 ([29]). *Für Spannungen der Form*

$$\boldsymbol{\sigma}(\mathbf{E}) = (\lambda + 2\mu/d)(\text{tr } \mathbf{E})\mathbf{I} + 2\mu h(|\text{dev } \mathbf{E}|) \text{dev } \mathbf{E} \quad (4.16)$$

mit $h : [0, \infty) \rightarrow (0, 1]$,

$$h(x) = 1 - \max\{0, 1/(1 + \Gamma_1)(1 - \Gamma_2/(2\mu x))\},$$

gilt die Monotonieabschätzung

$$\frac{1}{d\lambda + 2\mu} |\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})|^2 \leq |\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})|_{\mathbb{C}^{-1}}^2 \leq (\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})) : (\mathbf{A} - \mathbf{B}), \quad (4.17)$$

wobei $|\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})|_{\mathbb{C}^{-1}}^2 := (\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})) : \mathbb{C}^{-1}(\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B}))$ und d die Dimension ist.

Beweis. Zunächst werden die in der Abschätzung beteiligten Ausdrücke berechnet. Für die linke Seite der 2. Abschätzung gilt

$$\begin{aligned} |\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})|_{\mathbb{C}^{-1}}^2 &= (\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})) : \mathbb{C}^{-1}(\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})) \\ &= (\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})) : \left(\frac{1}{d^2\lambda + 4\mu} \text{tr}(\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B}))\mathbf{I} + \frac{1}{2\mu} \text{dev}(\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})) \right) \\ &= \left(\left(\lambda + \frac{2\mu}{d} \right) \text{tr}(\mathbf{A} - \mathbf{B})\mathbf{I} + 2\mu(h(|\text{dev } \mathbf{A}|) \text{dev } \mathbf{A} - h(|\text{dev } \mathbf{B}|) \text{dev } \mathbf{B}) \right) : \\ &\quad \left(\frac{d\lambda + 2\mu}{d^2\lambda + 4\mu} \text{tr}(\mathbf{A} - \mathbf{B})\mathbf{I} + (h(|\text{dev } \mathbf{A}|) \text{dev } \mathbf{A} - h(|\text{dev } \mathbf{B}|) \text{dev } \mathbf{B}) \right) \\ &= \frac{(d\lambda + 2\mu)^2}{(d^2\lambda + 4\mu)} \text{tr}(\mathbf{A} - \mathbf{B})^2 + 2\mu|h(|\text{dev } \mathbf{A}|) \text{dev } \mathbf{A} - h(|\text{dev } \mathbf{B}|) \text{dev } \mathbf{B}|^2. \end{aligned} \quad (4.18)$$

Für die rechte Seite der Abschätzung (4.17) gilt

$$\begin{aligned} (\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})) : (\mathbf{A} - \mathbf{B}) &= \left(\lambda + \frac{2\mu}{d} \right) \text{tr}(\mathbf{A} - \mathbf{B})\mathbf{I} : (\mathbf{A} - \mathbf{B}) \\ &\quad + 2\mu(h(|\text{dev } \mathbf{A}|) \text{dev } \mathbf{A} - h(|\text{dev } \mathbf{B}|) \text{dev } \mathbf{B}) : (\mathbf{A} - \mathbf{B}). \end{aligned} \quad (4.19)$$

Zu zeigen bleibt damit $\Delta \leq 0$, mit

$$\begin{aligned} \Delta := & |h(|\operatorname{dev} \mathbf{A}|) \operatorname{dev} \mathbf{A} - h(|\operatorname{dev} \mathbf{B}|) \operatorname{dev} \mathbf{B}|^2 \\ & - (h(|\operatorname{dev} \mathbf{A}|) \operatorname{dev} \mathbf{A} - h(|\operatorname{dev} \mathbf{B}|) \operatorname{dev} \mathbf{B}) : (\mathbf{A} - \mathbf{B}). \end{aligned}$$

Die Funktion $h : [0, \infty) \rightarrow (0, 1]$, $h(x) = 1 - \max\{0, 1/(1 + \Gamma_1)(1 - \Gamma_2/(2\mu x))\}$ ist stetig und monoton fallend,

$$h(0) = 1 \geq h(x) > \frac{\Gamma_1}{1 + \Gamma_1} \geq 0.$$

Ohne Beschränkung der Allgemeinheit kann $a := |\operatorname{dev} \mathbf{A}| \leq b := |\operatorname{dev} \mathbf{B}|$ vorausgesetzt werden. Mit $\alpha := h(a)$ und $\beta := h(b)$ gilt

$$\begin{aligned} \Delta = & \alpha^2 a^2 - 2\alpha\beta \operatorname{dev} \mathbf{A} : \operatorname{dev} \mathbf{B} + \beta^2 b^2 \\ & - (\alpha a^2 + \beta b^2 - \alpha \operatorname{dev} \mathbf{A} : \operatorname{dev} \mathbf{B} - \beta \operatorname{dev} \mathbf{A} : \operatorname{dev} \mathbf{B}) \\ = & \alpha^2 a^2 + \beta^2 b^2 - \alpha a^2 - \beta b^2 + \operatorname{dev} \mathbf{A} : \operatorname{dev} \mathbf{B} (\alpha + \beta - 2\alpha\beta) \\ = & (\alpha a - \beta b)^2 - (\alpha a - \beta b)(a - b) + (\operatorname{dev} \mathbf{A} - \operatorname{dev} \mathbf{B} - ab)(\alpha + \beta - 2\alpha\beta) \\ = & (\alpha a - \beta b)^2 - (\alpha a - \beta b)(a - b)(\operatorname{dev} \mathbf{A} : \operatorname{dev} \mathbf{B} - ab)((1 - \alpha)\beta + \alpha(1 - \beta)) \\ \leq & (\alpha a - \beta b)^2 - (\alpha a - \beta b)(a - b) \\ = & (h(a)a - h(b)b)((h(a) - 1)a - (h(b) - 1)b). \end{aligned}$$

Der Ausdruck $xh(x) \geq 0$ ist monoton steigend in $0 \leq x < \infty$, während $x(h(x) - 1) \leq 0$ monoton fallend ist. Daher folgt aus $a \leq b$, daß $h(a)a \leq h(b)b$ und $(h(a) - 1)a \geq (h(b) - 1)b$. Deshalb gilt $\Delta \leq 0$.

Der Beweis der ersten Abschätzung in (4.17) folgt durch

$$\begin{aligned} & \frac{1}{d\lambda + 2\mu} |\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})|^2 \\ = & \frac{1}{d\lambda + 2\mu} \left(\frac{1}{d^2} (\operatorname{tr}(\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})))^2 \mathbf{I} : \mathbf{I} + \operatorname{dev}(\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})) : \operatorname{dev}(\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})) \right) \\ \leq & \frac{1}{d^2\lambda + d2\mu} \frac{1}{d} (\operatorname{tr}(\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})))^2 \mathbf{I} : \mathbf{I} \\ & + \frac{1}{2\mu} \operatorname{dev}(\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})) : \operatorname{dev}(\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})) \\ = & (\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})) : \mathbb{C}^{-1}(\boldsymbol{\sigma}(\mathbf{A}) - \boldsymbol{\sigma}(\mathbf{B})). \end{aligned}$$

■

Im weiteren Verlauf dieses Abschnitts werden Abschätzungen bewiesen, die eine Kontrolle des Verzerrungsfehlers durch den Spannungsfehler ermöglichen. Die numerische Lösung des Problems 6 und die exakte Lösung des Problems 5 können in einem Gebiet $\omega \subset \Omega$ für das Materialverhalten elastische oder plastische Phasen ergeben. Es kann aber nicht vorausgesetzt werden, daß beide Lösungen notwendigerweise einen Zustand derselben Phase darstellen. Daher sind drei verschiedene Fälle zu untersuchen. Die numerische Lösung und die exakte Lösung können in $\omega \subset \Omega$

einen Zustand derselben Phase ergeben, der elastisch oder plastisch sein kann, oder es können in $\omega \subset \Omega$ die numerische und die exakten Lösung Zustände unterschiedlicher Phase darstellen. Die drei Fälle werden in den folgenden Unterabschnitten behandelt und als elastisch-elastisch, plastisch-plastisch bzw. elastisch-plastisch bezeichnet.

Bei Auftreten von plastischem Materialverhalten wird der folgende Satz benötigt.

Satz 4.5. *Der Verzerrungsfehler kann abgeschätzt werden durch die Summe aus dem Spannungsfehler und dem Fehler der plastischen Verzerrung. Es gilt fast überall in dem Gebiet Ω*

$$|\epsilon(\mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta})| \leq \frac{1}{2\mu} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}| + |\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1+\theta}|. \quad (4.20)$$

Beweis. Wegen $\epsilon(\mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta}) = \mathbb{C}^{-1}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}) + (\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1+\theta})$ erhält man

$$|\epsilon(\mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta})| \leq |\mathbb{C}^{-1}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta})| + |\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1+\theta}|. \quad (4.21)$$

Mit

$$\begin{aligned} & \mathbb{C}^{-1}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}) \\ &= \frac{1}{d(d\lambda + 2\mu)} \operatorname{tr}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}) \mathbf{I} + \frac{1}{2\mu} \operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}) \end{aligned}$$

folgt

$$\begin{aligned} |\mathbb{C}^{-1}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta})|^2 &= \frac{1}{d(d\lambda + 2\mu)^2} (\operatorname{tr}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}))^2 \\ &\quad + \frac{1}{4\mu^2} \operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}) : \operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}). \end{aligned}$$

Wegen $\lambda, \mu > 0$ gilt $1/(4\mu^2) \geq 1/(d\lambda + 2\mu)^2$. Dies zeigt

$$|\mathbb{C}^{-1}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta})|^2 \leq \frac{1}{4\mu^2} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|^2.$$

Eingesetzt in (4.21) ergibt sich die Behauptung. ■

1. Fall: elastisch-elastisch

In diesem Unterabschnitt wird unter den Voraussetzungen der folgenden Definition eine Abschätzung des Verzerrungsfehlers durch den Spannungsfehler bewiesen.

Definition 4.6 (Voraussetzungen 1). *Es wird vorausgesetzt, daß für das orts- und zeitdiskrete Problem 6 sowie für das zeitdiskrete Problem 5 in dem Zeitschritt $[t_{k-1}, t_k]$ dieselben Anfangswerte $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$ gegeben sind. Es wird der Fall betrachtet, daß die Lösungen von Problem 5 und Problem 6 zum Zeitpunkt $t_{k-1+\theta}$ in einem Gebiet $\omega \subset \Omega$ elastisches Materialverhalten aufweisen.*

Satz 4.7. *Unter den Voraussetzungen aus Definition 4.6 gilt fast überall in ω*

$$|\boldsymbol{\epsilon}(\mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta})| \leq \frac{1}{2\mu} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|. \quad (4.22)$$

Beweis. Es gilt

$$\begin{aligned} & |\boldsymbol{\epsilon}(\mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta})|^2 \\ &= \frac{1}{(d\lambda + 2\mu)^2} \frac{1}{d^2} (\text{tr}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta})^2 \mathbf{I} : \mathbf{I} + \frac{1}{(2\mu)^2} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|^2). \end{aligned}$$

■

2. Fall: plastisch-plastisch

In diesem Unterabschnitt wird unter den Voraussetzungen der nachfolgenden Definition eine Abschätzung des Verzerrungsfehlers sowie des Fehlers der internen Spannungen durch den Spannungsfehler bewiesen. Der Unterabschnitt teilt sich nochmals in zwei Teile. Die benötigte Abschätzung des Verzerrungsfehlers wird für den Spezialfall der Plastizität ($\varrho \rightarrow 0$) bewiesen, sowie auf eine zweite unabhängige Art für die Viskoplastizität ($\varrho > 0$). Die Abschätzungen der Viskoplastizität stellen damit auch eine Kontrolle für die Abschätzungen der Plastizität dar, die sich im Grenzfall $\varrho \rightarrow 0$ wieder ergeben.

Definition 4.8 (Voraussetzungen 2). *Es wird vorausgesetzt, daß für das orts- und zeitdiskrete Problem 6 sowie für das zeitdiskrete Problem 5 in dem Zeitschritt $[t_{k-1}, t_k]$ dieselben Anfangswerte $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$ und $\boldsymbol{\alpha}_{h,k-1}$ (kinematische Verfestigung) bzw. $\alpha_{h,k-1}$ (isotrope Verfestigung) gegeben sind. Es wird der Fall betrachtet, daß die Lösungen von Problem 5 und Problem 6 zum Zeitpunkt $t_{k-1+\theta}$ in einem Gebiet $\omega \subset \Omega$ plastisches Materialverhalten aufweisen.*

a. Abschätzungen für den Verzerrungsfehler in der Plastizität

Dieser Unterabschnitt beschreibt eine Methode, den Verzerrungsfehler durch den Spannungsfehler abzuschätzen für den Fall der ratenunabhängigen Plastizität im Grenzfall $\varrho \rightarrow 0$.

Der folgende Satz findet sich in [23] und wird hier mit expliziten Konstanten angegeben.

Satz 4.9. *Im Fall von isotroper und kinematischer Verfestigung kann der Fehler der plastischen Verzerrung durch den Spannungsfehler abgeschätzt werden. Unter den Voraussetzungen aus Definition 4.8 gilt fast überall in $\omega \subset \Omega$ für Plastizität ($\varrho \rightarrow 0$) mit isotroper Verfestigung*

$$|\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1+\theta}|^2 \leq \frac{1}{H_1^2 H^2 \sigma_y^2} |\alpha_{k-1+\theta} - \alpha_{h,k-1+\theta}|^2 + \frac{4}{H_1^2 H^4 \sigma_y^4} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|^2 \quad (4.23)$$

und für Plastizität mit kinematischer Verfestigung

$$|\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1+\theta}|^2 = \frac{1}{k_1^2} |\alpha_{k-1+\theta} - \alpha_{h,k-1+\theta}|^2. \quad (4.24)$$

Beweis. Der Beweis folgt [23]. Zunächst wird der Fall der isotropen Verfestigung behandelt. Es gilt

$$\begin{aligned} & |\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1+\theta}|^2 \\ = & (|\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1}| - |\mathbf{p}_{h,k-1+\theta} - \mathbf{p}_{h,k-1}|)^2 \\ & + 2(|\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1}| |\mathbf{p}_{h,k-1+\theta} - \mathbf{p}_{h,k-1}| - (\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1}) : (\mathbf{p}_{h,k-1+\theta} - \mathbf{p}_{h,k-1})). \end{aligned} \quad (4.25)$$

Aus der ersten Komponente des Fließgesetzes erhält man

$$\frac{\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1}}{|\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1}|} = \frac{\text{dev } \boldsymbol{\sigma}_{k-1+\theta}}{|\text{dev } \boldsymbol{\sigma}_{k-1+\theta}|}.$$

Mit der zweiten Komponente des Fließgesetzes,

$$-H_1^{-1} \frac{\alpha_{k-1+\theta} - \alpha_{h,k-1}}{\theta \Delta t_k} = -\frac{1}{\varrho} \frac{1}{1 + H^2 \sigma_y^2} \left(1 - \frac{(1 + \alpha_{k-1+\theta} H) \sigma_y}{|\text{dev } \boldsymbol{\sigma}_{h,k-1+\theta}|}\right) H \sigma_y |\text{dev } \boldsymbol{\sigma}_{k-1+\theta}|,$$

erhält man

$$\begin{aligned} |\text{dev } \boldsymbol{\sigma}_{k-1+\theta}| &= \frac{1}{HH_1 \sigma_y \theta \Delta t_k} \\ &\times (-\alpha_{h,k-1} (\varrho + \varrho H^2 \sigma_y^2) + HH_1 \sigma_y^2 \theta \Delta t_k + \alpha_{k-1+\theta} (\varrho + \varrho H^2 \sigma_y^2 + H^2 H_1 \sigma_y^2 \theta \Delta t_k)). \end{aligned}$$

Für $\varrho \rightarrow 0$ gilt $|\text{dev } \boldsymbol{\sigma}_{k-1+\theta}| = \sigma_y (1 + \alpha_{k-1+\theta} H)$, so daß

$$\frac{\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1}}{|\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1}|} = \frac{\text{dev } \boldsymbol{\sigma}_{k-1+\theta}}{\sigma_y (1 + \alpha_{k-1+\theta} H)}. \quad (4.26)$$

Zwischen plastischer Verzerrung und interner Variable gilt die Beziehung

$$|\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1}| = -\frac{1}{H \sigma_y} (\xi_{k-1+\theta} - \xi_{h,k-1}).$$

Dies zeigt mit (4.26)

$$\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1} = -\frac{\xi_{k-1+\theta} - \xi_{h,k-1}}{H \sigma_y^2 (1 + \alpha_{k-1+\theta} H)} \text{dev } \boldsymbol{\sigma}_{k-1+\theta}.$$

Daher gilt

$$\begin{aligned} |\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1}| - |\mathbf{p}_{h,k-1+\theta} - \mathbf{p}_{h,k-1}| &= \left| \frac{\xi_{k-1+\theta} - \xi_{h,k-1}}{H \sigma_y} - \frac{\xi_{h,k-1+\theta} - \xi_{h,k-1}}{H \sigma_y} \right| \\ &= \frac{1}{H \sigma_y} |\xi_{k-1+\theta} - \xi_{h,k-1+\theta}| \end{aligned} \quad (4.27)$$

und

$$\begin{aligned} & 2(|\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1}| |\mathbf{p}_{h,k-1+\theta} - \mathbf{p}_{h,k-1}| - (\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1}) : (\mathbf{p}_{h,k-1+\theta} - \mathbf{p}_{h,k-1})) \\ = & 2 \frac{\xi_{k-1+\theta} - \xi_{h,k-1}}{H \sigma_y^2 (1 + \alpha_{k-1+\theta} H)} \frac{\xi_{h,k-1+\theta} - \xi_{h,k-1}}{H \sigma_y^2 (1 + \alpha_{h,k-1+\theta} H)} \\ & \times (|\text{dev } \boldsymbol{\sigma}_{k-1+\theta}| |\text{dev } \boldsymbol{\sigma}_{h,k-1+\theta}| - \text{dev } \boldsymbol{\sigma}_{k-1+\theta} : \text{dev } \boldsymbol{\sigma}_{h,k-1+\theta}). \end{aligned} \quad (4.28)$$

Es wird gezeigt, daß der skalare Faktor in (4.28) beschränkt ist. Mit $\alpha_{k-1+\theta} \geq 0$ und $\alpha_{k-1+\theta} = -H_1 \xi_{k-1+\theta}$ gilt

$$\begin{aligned} \frac{\xi_{k-1+\theta} - \xi_{h,k-1}}{H\sigma_y^2(1 + \alpha_{k-1+\theta}H)} &\leq \frac{|\xi_{k-1+\theta} - \xi_{h,k-1}|}{|H\sigma_y^2(1 + \alpha_{k-1+\theta}H)|} \leq \frac{|\xi_{k-1+\theta} - \xi_{h,k-1}|}{|H^2\sigma_y^2\alpha_{k-1+\theta}|} \\ &= \frac{1}{H_1H^2\sigma_y^2} \frac{|\alpha_{k-1+\theta} - \alpha_{h,k-1}|}{|\alpha_{k-1+\theta}|} \leq \frac{1}{H_1H^2\sigma_y^2} \left(1 + \frac{|\alpha_{h,k-1}|}{|\alpha_{k-1+\theta}|}\right) \leq \frac{2}{H_1H^2\sigma_y^2}, \end{aligned} \quad (4.29)$$

wobei der letzte Schritt wegen der zweiten Komponente des Fließgesetzes gültig ist. Damit gilt

$$-\frac{1}{H_1} \frac{\alpha_{k-1+\theta} - \alpha_{h,k-1}}{\theta \Delta t_k} \leq 0.$$

Daher ist $\alpha_{k-1+\theta} \geq \alpha_{h,k-1}$, so daß $|\alpha_{k-1+\theta}| \geq |\alpha_{h,k-1}|$ wegen $\alpha \geq 0$. Mit (4.25) und

$$(\xi_{k-1+\theta} - \xi_{h,k-1+\theta}) = -1/H_1(\alpha_{k-1+\theta} - \alpha_{h,k-1+\theta})$$

erhält man

$$\begin{aligned} &|\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1+\theta}|^2 \\ &\leq \frac{1}{H_1^2H^2\sigma_y^2} |\alpha_{k-1+\theta} - \alpha_{h,k-1+\theta}|^2 \\ &\quad + \frac{8}{H_1^2H^4\sigma_y^4} (|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}| |\operatorname{dev} \boldsymbol{\sigma}_{h,k-1+\theta}| - \operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta} : \operatorname{dev} \boldsymbol{\sigma}_{h,k-1+\theta}) \\ &\leq \frac{1}{H_1^2H^2\sigma_y^2} |\alpha_{k-1+\theta} - \alpha_{h,k-1+\theta}|^2 + \frac{4}{H_1^2H^4\sigma_y^4} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|^2. \end{aligned}$$

Es wird der Fall der kinematischen Verfestigung betrachtet. Aus $\mathbf{p} = -\boldsymbol{\xi}$ folgert man

$$|\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1+\theta}| = |\boldsymbol{\xi}_{k-1+\theta} - \boldsymbol{\xi}_{h,k-1+\theta}|,$$

und mit $\boldsymbol{\alpha} = -k_1 \boldsymbol{\xi}$ erhält man $|\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1+\theta}| = \frac{1}{k_1} |\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1+\theta}|$. ■

Um den Verzerrungsfehler durch den Spannungsfehler zu kontrollieren, ist zusätzlich zu den Sätzen 4.5 und 4.9 eine Abschätzung erforderlich, die den Fehler der internen Spannungen durch den Spannungsfehler abschätzt.

Satz 4.10. *Unter den Voraussetzungen aus Definition 4.8 gilt fast überall in $\omega \subset \Omega$ im Fall der isotropen Verfestigung*

$$|\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1+\theta}| \leq \frac{1}{H\sigma_y} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}| \quad (4.30)$$

und im Fall der kinematischen Verfestigung

$$|\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{\theta,h}| \leq 2 |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|. \quad (4.31)$$

Beweis. Zunächst wird der Fall der isotropen Verfestigung betrachtet. Die zweite Komponente vom Fließgesetz,

$$-H_1^{-1} \frac{\alpha_{k-1+\theta} - \alpha_{h,k-1}}{\theta \Delta t_k} = -\frac{1}{\varrho} \frac{1}{1 + H^2 \sigma_y^2} \left(1 - \frac{(1 + \alpha_{k-1+\theta} H) \sigma_y}{|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}|}\right) + H \sigma_y |\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}|$$

läßt sich nach der internen Spannung $\alpha_{k-1+\theta}$ auflösen. Im plastischen Fall gilt

$$\alpha_{k-1+\theta} = \frac{\varrho(1 + H^2 \sigma_y^2) \alpha_{h,k-1} - \theta \Delta t_k H_1 H \sigma_y^2}{\varrho(1 + H^2 \sigma_y^2) + \theta \Delta t_k H_1 H^2 \sigma_y^2} + \frac{\theta \Delta t_k H_1 H \sigma_y}{\varrho(1 + H^2 \sigma_y^2) + \theta \Delta t_k H_1 H^2 \sigma_y^2} |\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}|.$$

Im Fall $\varrho \rightarrow 0$ erhält man

$$\alpha_{k-1+\theta} = -\frac{1}{H} + \frac{1}{H \sigma_y} |\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}|.$$

Daraus folgt

$$\begin{aligned} & |\alpha_{k-1+\theta} - \alpha_{h,k-1+\theta}|^2 \\ &= \frac{1}{H^2 \sigma_y^2} (|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}| - |\operatorname{dev} \boldsymbol{\sigma}_{h,k-1+\theta}|)^2 \leq \frac{1}{H^2 \sigma_y^2} |\operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta})|^2 \\ &\leq \frac{1}{H^2 \sigma_y^2} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|^2. \end{aligned}$$

Im zweiten Fall wird kinematische Verfestigung betrachtet. Aus der zweiten Komponente vom Fließgesetz,

$$\frac{1}{k_1} \frac{\alpha_{k-1+\theta} - \alpha_{h,k-1}}{\theta \Delta t_k} = \frac{1}{2\varrho} \left(1 - \frac{\sigma_y}{|\operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1+\theta})|}\right) \operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1+\theta}), \quad (4.32)$$

erhält man

$$|\operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1+\theta})| = \frac{2\varrho}{k_1 \theta \Delta t_k} |\alpha_{k-1+\theta} - \alpha_{h,k-1}| + \sigma_y.$$

Damit folgt aus (4.32)

$$\frac{\alpha_{k-1+\theta} - \alpha_{h,k-1}}{k_1 \theta \Delta t_k} = \frac{1}{2\varrho} \frac{2\varrho |\alpha_{k-1+\theta} - \alpha_{h,k-1}|}{2\varrho |\alpha_{k-1+\theta} - \alpha_{h,k-1}| + k_1 \theta \Delta t_k \sigma_y} \operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1+\theta})$$

und für $\varrho \rightarrow 0$

$$\alpha_{k-1+\theta} - \alpha_{h,k-1} = \frac{1}{\sigma_y} |\alpha_{k-1+\theta} - \alpha_{h,k-1}| \operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1+\theta}).$$

Umformen ergibt

$$\operatorname{dev}(\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}) = \frac{|\alpha_{k-1+\theta} - \alpha_{h,k-1}|}{|\alpha_{k-1+\theta} - \alpha_{h,k-1}| + \sigma_y} \operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1})$$

und

$$\operatorname{dev}(\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}) = \frac{|\operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1})| - \sigma_y}{|\operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1})|} \operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}).$$

Damit gilt

$$\begin{aligned} \boldsymbol{\tau}_1 - \boldsymbol{\tau}_2 &:= \operatorname{dev}(\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1+\theta}) \\ &= \frac{|\operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1})| - \sigma_y}{|\operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1})|} \operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}) \\ &\quad - \frac{|\operatorname{dev}(\boldsymbol{\sigma}_{h,k-1+\theta} - \boldsymbol{\alpha}_{h,k-1})| - \sigma_y}{|\operatorname{dev}(\boldsymbol{\sigma}_{h,k-1+\theta} - \boldsymbol{\alpha}_{h,k-1})|} \operatorname{dev}(\boldsymbol{\sigma}_{h,k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}) \\ &=: a\mathbf{A} - b\mathbf{B}. \end{aligned}$$

Mit Hilfe der Dreiecksungleichung schätzt man $|\boldsymbol{\tau}_1 - \boldsymbol{\tau}_2|^2$ ab,

$$\begin{aligned} |\boldsymbol{\tau}_1 - \boldsymbol{\tau}_2|^2 &= \left| \frac{a+b}{2}(\mathbf{A} - \mathbf{B}) + \frac{a-b}{2}(\mathbf{A} + \mathbf{B}) \right|^2 \\ &\leq \frac{1}{4}(a+b)^2 |\mathbf{A} - \mathbf{B}|^2 + \frac{1}{2}|a+b||a-b| |\mathbf{A} - \mathbf{B}| |\mathbf{A} + \mathbf{B}| + \frac{1}{4}(a-b)^2 |\mathbf{A} + \mathbf{B}|^2 \\ &\leq \frac{1}{2}(a+b)^2 |\mathbf{A} - \mathbf{B}|^2 + \frac{1}{2}(a-b)^2 |\mathbf{A} + \mathbf{B}|^2. \end{aligned}$$

Der letzte Summand läßt sich weiter abschätzen durch

$$\begin{aligned} \frac{1}{2}(a-b)^2 |\mathbf{A} + \mathbf{B}|^2 &\leq \frac{1}{2}(a-b)^2 (|\mathbf{A}| + |\mathbf{B}|)^2 = \frac{\sigma_y^2 (|\mathbf{A}| + |\mathbf{B}|)^2}{2 |\mathbf{A}|^2 |\mathbf{B}|^2} (|\mathbf{A}| - |\mathbf{B}|)^2 \\ &= \frac{1}{2} \left(\frac{|\mathbf{A}| - \sigma_y}{|\mathbf{A}|} + \frac{|\mathbf{B}| - \sigma_y}{|\mathbf{B}|} - 2 \right)^2 (|\mathbf{A}| - |\mathbf{B}|)^2 \\ &\leq 2(|\mathbf{A}| - |\mathbf{B}|)^2 \leq 2|\mathbf{A} - \mathbf{B}|^2. \end{aligned}$$

Der Fehler der internen Spannung, $|\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1+\theta}|$, läßt sich also durch die Spannung wie folgt abschätzen,

$$\begin{aligned} |\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1+\theta}|^2 &= |\operatorname{dev}(\boldsymbol{\alpha}_{k-1+\theta}) - \operatorname{dev}(\boldsymbol{\alpha}_{h,k-1+\theta})|^2 = |\boldsymbol{\tau}_1 - \boldsymbol{\tau}_2|^2 \\ &\leq \frac{1}{2}(a+b)^2 |\mathbf{A} - \mathbf{B}|^2 + 2|\mathbf{A} - \mathbf{B}|^2 \leq 4|\mathbf{A} - \mathbf{B}|^2 = 4|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta} - \operatorname{dev} \boldsymbol{\sigma}_{h,k-1+\theta}|^2 \\ &\leq 4|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|^2. \end{aligned}$$

■

Satz 4.11. *Unter den Voraussetzungen aus Definition 4.8 gilt fast überall in $\omega \subset \Omega$, daß sich der Verzerrungsfehler durch den Spannungsfehler abschätzen läßt. Im Fall der isotropen Verfestigung gilt*

$$|\boldsymbol{\epsilon}(\mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta})| \leq \left(\frac{1}{2\mu} + \frac{\sqrt{5}}{H_1 H^2 \sigma_y^2} \right) |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|, \quad (4.33)$$

und im Fall der kinematischen Verfestigung gilt

$$|\boldsymbol{\epsilon}(\mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta})| \leq \left(\frac{1}{2\mu} + \frac{2}{k_1} \right) |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|. \quad (4.34)$$

Beweis. Zunächst wird der Fall der isotropen Verfestigung betrachtet. Mit (4.30) (Satz 4.10) in (4.23) (Satz 4.9) ergibt sich

$$|\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1+\theta}| \leq \frac{\sqrt{5}}{H_1 H^2 \sigma_y^2} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|.$$

Mit (4.20) folgt die Behauptung.

Im zweiten Fall wird kinematische Verfestigung betrachtet. Mit (4.31) (Satz 4.10) in (4.24) (Satz 4.9) ergibt sich

$$|\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1+\theta}| \leq \frac{2}{k_1} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|.$$

Mit (4.20) folgt die Behauptung. ■

b. Abschätzungen für den Verzerrungsfehler in der Viskoplastizität

Dieser Unterabschnitt beschreibt eine Methode, den Verzerrungsfehler durch den Spannungsfehler abzuschätzen. Der Regularisierungsparameter $\varrho \geq 0$ ist beliebig.

Der Fehler der internen Spannungen läßt sich durch den Spannungsfehler abschätzen.

Satz 4.12. *Unter den Voraussetzungen aus Definition 4.8 gilt fast überall in $\omega \subset \Omega$ im Fall der isotropen Verfestigung*

$$|\alpha_{k-1+\theta} - \alpha_{h,k-1+\theta}| \leq \left(\frac{\theta \Delta t_k H_1 H \sigma_y}{\varrho(1 + H^2 \sigma_y^2) + \theta \Delta t_k H_1 H^2 \sigma_y^2} \right) |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}| \quad (4.35)$$

und im Fall der kinematischen Verfestigung

$$|\alpha_{k-1+\theta} - \alpha_{h,k-1+\theta}| \leq \sqrt{2 + \frac{1}{2} \left(\frac{4\varrho + 2\theta \Delta t_k k_1}{\theta \Delta t_k k_1} \right)^2} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|. \quad (4.36)$$

Beweis. Zunächst wird der Fall der isotropen Verfestigung betrachtet. Für die interne Spannung gilt

$$\alpha_{k-1+\theta} = \frac{\varrho(1 + H^2 \sigma_y^2) \alpha_{h,k-1} - \theta \Delta t_k H_1 H \sigma_y^2}{\varrho(1 + H^2 \sigma_y^2) + \theta \Delta t_k H_1 H^2 \sigma_y^2} + \frac{\theta \Delta t_k H_1 H \sigma_y}{\varrho(1 + H^2 \sigma_y^2) + \theta \Delta t_k H_1 H^2 \sigma_y^2} |\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}|.$$

Damit läßt sich der Fehler der internen Spannung abschätzen,

$$\begin{aligned} |\alpha_{k-1+\theta} - \alpha_{h,k-1+\theta}| &= \frac{\theta \Delta t_k H_1 H \sigma_y}{\varrho(1 + H^2 \sigma_y^2) + \theta \Delta t_k H_1 H^2 \sigma_y^2} (|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}| - |\operatorname{dev} \boldsymbol{\sigma}_{h,k-1+\theta}|) \\ &\leq \frac{\theta \Delta t_k H_1 H \sigma_y}{\varrho(1 + H^2 \sigma_y^2) + \theta \Delta t_k H_1 H^2 \sigma_y^2} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|. \end{aligned}$$

Im zweiten Fall wird kinematische Verfestigung betrachtet. Aus der zweiten Komponente vom Fließgesetz,

$$\frac{1}{k_1} \frac{\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}}{\theta \Delta t_k} = \frac{1}{2\varrho} \left(1 - \frac{\sigma_y}{|\operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1+\theta})|}\right) \operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1+\theta})$$

erhält man

$$\begin{aligned} & \frac{1}{k_1 \theta \Delta t_k} (\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}) \\ &= \frac{1}{2\varrho} \frac{2\varrho |\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}|}{2\varrho |\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}| + k_1 \theta \Delta t_k \sigma_y} \operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1+\theta}) \\ \Rightarrow \boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1} &= \frac{k_1 \theta \Delta t_k |\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}|}{2\varrho |\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}| + k_1 \theta \Delta t_k \sigma_y} \operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{k-1+\theta}) \\ \Rightarrow \left(1 + \frac{k_1 \theta \Delta t_k |\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}|}{2\varrho |\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}| + k_1 \theta \Delta t_k \sigma_y}\right) \operatorname{dev} \boldsymbol{\alpha}_{k-1+\theta} &- \operatorname{dev} \boldsymbol{\alpha}_{h,k-1} \\ &= \frac{k_1 \theta \Delta t_k |\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}|}{2\varrho |\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}| + k_1 \theta \Delta t_k \sigma_y} \operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta} \\ \Rightarrow \left(1 + \frac{k_1 \theta \Delta t_k |\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}|}{2\varrho |\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}| + k_1 \theta \Delta t_k \sigma_y}\right) \operatorname{dev}(\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}) & \\ &= \frac{k_1 \theta \Delta t_k |\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}|}{2\varrho |\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}| + k_1 \theta \Delta t_k \sigma_y} \operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}) \\ \Rightarrow \operatorname{dev}(\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}) & \\ &= \frac{|\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}|}{\frac{2\varrho}{k_1 \theta \Delta t_k} |\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}| + \sigma_y + |\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}|} \operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}) \\ \Rightarrow |\operatorname{dev}(\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1})| &= \frac{\theta \Delta t_k k_1}{2\varrho + \theta \Delta t_k k_1} (|\operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1})| - \sigma_y) \\ \Rightarrow \operatorname{dev}(\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}) &= a \mathbf{A}, \end{aligned}$$

wobei folgende Definitionen gelten,

$a :=$

$$\frac{|\operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1})| - \sigma_y}{\frac{2\varrho}{\theta \Delta t_k k_1} (|\operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1})| - \sigma_y) + \frac{\theta \Delta t_k k_1}{2\varrho + \theta \Delta t_k k_1} \sigma_y + |\operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1})| - \sigma_y},$$

$$\mathbf{A} := \operatorname{dev}(\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}).$$

Entsprechend wird definiert

$b :=$

$$\frac{|\operatorname{dev}(\boldsymbol{\sigma}_{h,k-1+\theta} - \boldsymbol{\alpha}_{h,k-1})| - \sigma_y}{\frac{2\varrho}{\theta \Delta t_k k_1} (|\operatorname{dev}(\boldsymbol{\sigma}_{h,k-1+\theta} - \boldsymbol{\alpha}_{h,k-1})| - \sigma_y) + \frac{\theta \Delta t_k k_1}{2\varrho + \theta \Delta t_k k_1} \sigma_y + |\operatorname{dev}(\boldsymbol{\sigma}_{h,k-1+\theta} - \boldsymbol{\alpha}_{h,k-1})| - \sigma_y},$$

$$\mathbf{B} := \operatorname{dev}(\boldsymbol{\sigma}_{h,k-1+\theta} - \boldsymbol{\alpha}_{h,k-1}).$$

Damit gilt $\boldsymbol{\tau}_1 - \boldsymbol{\tau}_2 := \operatorname{dev}(\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1+\theta}) = a \mathbf{A} - b \mathbf{B}$. Analog zum Fall $\varrho \rightarrow 0$ gilt

$$|\boldsymbol{\tau}_1 - \boldsymbol{\tau}_2|^2 \leq \frac{1}{2} (a + b)^2 |\mathbf{A} - \mathbf{B}|^2 + \frac{1}{2} (a - b)^2 |\mathbf{A} + \mathbf{B}|^2.$$

Der 2. Summand wird abgeschätzt,

$$\frac{1}{2}(a-b)^2|\mathbf{A} + \mathbf{B}|^2 \leq \frac{1}{2}(a-b)^2(|\mathbf{A}| + |\mathbf{B}|)^2.$$

Es wird definiert

$$c_1 := \frac{2\rho}{\theta\Delta t_k k_1} \quad \text{und} \quad c_2 := \frac{\theta\Delta t_k k_1}{2\rho + \theta\Delta t_k k_1}$$

und man erhält

$$\begin{aligned} & \frac{1}{2}(a-b)^2(|\mathbf{A}| + |\mathbf{B}|)^2 \\ &= \frac{1}{2} \frac{(|\mathbf{A}| + |\mathbf{B}|)^2 c_2^2 \sigma_y^2 (|\mathbf{A}| - |\mathbf{B}|)^2}{((1+c_1)|\mathbf{A}| + (-1-c_1+c_2)\sigma_y)^2 ((1+c_1)|\mathbf{B}| + (-1-c_1+c_2)\sigma_y)^2} \\ &= \frac{1}{2} \frac{(|\mathbf{A}| + |\mathbf{B}|)^2 c_2^2 \sigma_y^2 (|\mathbf{A}| - |\mathbf{B}|)^2}{((1+c_1)(|\mathbf{A}| - \sigma_y) + c_2\sigma_y)^2 ((1+c_1)(|\mathbf{B}| - \sigma_y) + c_2\sigma_y)^2}. \end{aligned}$$

Es wird eine obere Schranke für folgenden Ausdruck berechnet,

$$\begin{aligned} & \frac{(|\mathbf{A}| + |\mathbf{B}|)c_2\sigma_y}{((1+c_1)(|\mathbf{A}| - \sigma_y) + c_2\sigma_y)((1+c_1)(|\mathbf{B}| - \sigma_y) + c_2\sigma_y)} \\ &= \frac{(|\mathbf{A}| - \sigma_y + |\mathbf{B}| - \sigma_y + 2\sigma_y)c_2\sigma_y}{((1+c_1)(|\mathbf{A}| - \sigma_y) + c_2\sigma_y)((1+c_1)(|\mathbf{B}| - \sigma_y) + c_2\sigma_y)} \\ &= \frac{(x_1 + x_2 + 2\sigma_y)c_2\sigma_y}{((1+c_1)x_1 + c_2\sigma_y)((1+c_1)x_2 + c_2\sigma_y)} \\ &= f(x_1, x_2), \end{aligned}$$

wobei x_1 und x_2 definiert sind durch

$$x_1 = |\mathbf{A}| - \sigma_y \quad \text{und} \quad x_2 = |\mathbf{B}| - \sigma_y.$$

Dann gilt

$$\begin{aligned} \frac{\partial f}{\partial x_1} &= -\frac{c_2\sigma_y((1+c_1)x_2 + (2+2c_1-c_2)\sigma_y)}{((1+c_1)x_1 + c_2\sigma_y)^2((1+c_1)x_2 + c_2\sigma_y)} \\ \frac{\partial f}{\partial x_2} &= -\frac{c_2\sigma_y((1+c_1)x_1 + (2+2c_1-c_2)\sigma_y)}{((1+c_1)x_2 + c_2\sigma_y)^2((1+c_1)x_1 + c_2\sigma_y)}. \end{aligned}$$

Wegen $c_1 > 0$, $c_2 > 0$ und

$$2 + 2c_1 - c_2 = \frac{8\rho^2 + 8\rho\theta\Delta t_k k_1 + \theta^2\Delta t_k^2 k_1^2}{\theta\Delta t_k k_1(2\rho + \theta\Delta t_k k_1)} > 0$$

gilt

$$\frac{\partial f}{\partial x_1} < 0 \quad \text{und} \quad \frac{\partial f}{\partial x_2} < 0.$$

Das Maximum von $f(x_1, x_2)$ liegt am Randpunkt $(x_1, x_2) = (0, 0)$ und ist

$$f(x_1, x_2) \leq f(0, 0) = \frac{2}{c_2} = \frac{4\varrho + 2\theta\Delta t_k k_1}{\theta\Delta t_k k_1}. \quad (4.37)$$

Der Fehler der internen Spannung läßt sich also durch die Spannung wie folgt abschätzen

$$\begin{aligned} |\boldsymbol{\alpha}_{k-1+\theta} - \boldsymbol{\alpha}_{h,k-1+\theta}|^2 &\leq \left(2 + \frac{1}{2} \left(\frac{4\varrho + 2\theta\Delta t_k k_1}{\theta\Delta t_k k_1}\right)^2\right) |\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta} - \operatorname{dev} \boldsymbol{\sigma}_{h,k-1+\theta}|^2 \\ &\leq \left(2 + \frac{1}{2} \left(\frac{4\varrho + 2\theta\Delta t_k k_1}{\theta\Delta t_k k_1}\right)^2\right) |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|^2. \end{aligned}$$

Satz 4.13. *Unter den Voraussetzungen aus Definition 4.8 gilt fast überall in $\omega \subset \Omega$, daß sich für isotrope Verfestigung der Fehler der plastischen Verzerrung folgendermaßen durch den Spannungsfehler abschätzen läßt,*

$$|\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1+\theta}| \leq \frac{\sqrt{3}\theta\Delta t_k}{\varrho(1 + H^2\sigma_y^2) + \theta\Delta t_k H_1 H^2\sigma_y^2} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|. \quad (4.38)$$

Beweis. In das Fließgesetz wird der Ausdruck (3.31) für die interne Spannung eingesetzt,

$$\begin{aligned} \frac{\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1}}{\theta\Delta t_k} &= \frac{1}{\varrho} \frac{1}{1 + H^2\sigma_y^2} \left(1 - \frac{(1 + \alpha_{k-1+\theta}H)\sigma_y}{|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}|}\right) \operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta} \\ &= \frac{|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}| - (1 + \alpha_{h,k-1}H)\sigma_y}{|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}|(\varrho(1 + H^2\sigma_y^2) + \theta\Delta t_{k-1} H_1 H^2\sigma_y^2)} \operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}. \end{aligned} \quad (4.39)$$

Für den Fehler der plastischen Verzerrung gilt

$$\begin{aligned} &|\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1+\theta}|^2 \\ &= (|\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1}| - |\mathbf{p}_{h,k-1+\theta} - \mathbf{p}_{h,k-1}|)^2 \\ &\quad + 2(|\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1}| |\mathbf{p}_{h,k-1+\theta} - \mathbf{p}_{h,k-1}| - (\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1}) : (\mathbf{p}_{h,k-1+\theta} - \mathbf{p}_{h,k-1})). \end{aligned} \quad (4.40)$$

Der 1. Summand in (4.40) läßt sich mit (4.39) folgendermaßen abschätzen,

$$\begin{aligned} &(|\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1}| - |\mathbf{p}_{h,k-1+\theta} - \mathbf{p}_{h,k-1}|)^2 \\ &= \frac{\theta^2 \Delta t_k^2}{(\varrho(1 + H^2\sigma_y^2) + \theta\Delta t_k H_1 H^2\sigma_y^2)^2} (|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}| - |\operatorname{dev} \boldsymbol{\sigma}_{h,k-1+\theta}|)^2 \\ &\leq \frac{\theta^2 \Delta t_k^2}{(\varrho(1 + H^2\sigma_y^2) + \theta\Delta t_k H_1 H^2\sigma_y^2)} |\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta} - \operatorname{dev} \boldsymbol{\sigma}_{h,k-1+\theta}|^2 \\ &\leq \frac{\theta^2 \Delta t_k^2}{(\varrho(1 + H^2\sigma_y^2) + \theta\Delta t_k H_1 H^2\sigma_y^2)^2} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|^2. \end{aligned}$$

Es gilt wegen der Fließregel

$$\begin{aligned} 0 &\leq \frac{|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}| - (1 + \alpha_{h,k-1}H)\sigma_y}{|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}|} \leq 1 \quad \text{und} \\ 0 &\leq \frac{|\operatorname{dev} \boldsymbol{\sigma}_{h,k-1+\theta}| - (1 + \alpha_{h,k-1}H)\sigma_y}{|\operatorname{dev} \boldsymbol{\sigma}_{h,k-1+\theta}|} \leq 1. \end{aligned}$$

Damit läßt sich der 2. Summand in (4.40) mit (4.39) folgendermaßen abschätzen

$$\begin{aligned} &2(|\boldsymbol{p}_{k-1+\theta} - \boldsymbol{p}_{h,k-1}| |\boldsymbol{p}_{h,k-1+\theta} - \boldsymbol{p}_{h,k-1}| - (\boldsymbol{p}_{k-1+\theta} - \boldsymbol{p}_{h,k-1}) : (\boldsymbol{p}_{h,k-1+\theta} - \boldsymbol{p}_{h,k-1})) \\ &= \frac{2\theta^2 \Delta t_k^2}{(\varrho(1 + H^2\sigma_y^2) + \theta\Delta t_k H_1 H^2\sigma_y^2)^2} \\ &\quad \times \frac{|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}| - (1 + \alpha_{h,k-1}H)\sigma_y}{|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}|} \frac{|\operatorname{dev} \boldsymbol{\sigma}_{h,k-1+\theta}| - (1 + \alpha_{0,h}H)\sigma_y}{|\operatorname{dev} \boldsymbol{\sigma}_{h,k-1+\theta}|} \\ &\quad \times (|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}| |\operatorname{dev} \boldsymbol{\sigma}_{h,k-1+\theta}| - \operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta} : \operatorname{dev} \boldsymbol{\sigma}_{h,k-1+\theta}) \\ &\leq \frac{2\theta^2 \Delta t_k^2}{(\varrho(1 + H^2\sigma_y^2) + \theta\Delta t_k H_1 H^2\sigma_y^2)^2} \\ &\quad \times (|\operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta}| |\operatorname{dev} \boldsymbol{\sigma}_{h,k-1+\theta}| - \operatorname{dev} \boldsymbol{\sigma}_{k-1+\theta} : \operatorname{dev} \boldsymbol{\sigma}_{h,k-1+\theta}) \\ &\leq \frac{2\theta^2 \Delta t_k^2}{(\varrho(1 + H^2\sigma_y^2) + \theta\Delta t_k H_1 H^2\sigma_y^2)^2} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|^2. \end{aligned}$$

Insgesamt ergibt sich

$$|\boldsymbol{p}_{k-1+\theta} - \boldsymbol{p}_{h,k-1+\theta}|^2 \leq \frac{3\theta^2 \Delta t_k^2}{(\varrho(1 + H^2\sigma_y^2) + \theta\Delta t_k H_1 H^2\sigma_y^2)^2} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|^2. \quad \blacksquare$$

Satz 4.14. *Unter den Voraussetzungen aus Definition 4.8 gilt fast überall in $\omega \subset \Omega$, daß sich für kinematische Verfestigung der Fehler der plastischen Verzerrung folgendermaßen durch den Spannungsfehler abschätzen läßt,*

$$|\boldsymbol{p}_{k-1+\theta} - \boldsymbol{p}_{h,k-1+\theta}| \leq \frac{1}{k_1} \sqrt{2 + \frac{1}{2} \left(\frac{4\varrho + 2\theta\Delta t_k k_1}{\theta\Delta t_k k_1} \right)^2} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}| \quad (4.41)$$

Beweis. Der Beweis folgt aus $\boldsymbol{p} = 1/k_1 \boldsymbol{\alpha}$ und Satz 4.12. ■

Satz 4.15. *Unter den Voraussetzungen aus Definition 4.8 gilt fast überall in $\omega \subset \Omega$, daß sich für isotrope Verfestigung der Verzerrungsfehler folgendermaßen durch den Spannungsfehler abschätzen läßt,*

$$|\boldsymbol{\epsilon}_{k-1+\theta} - \boldsymbol{\epsilon}_{h,k-1+\theta}| \leq \left(\frac{1}{2\mu} + \frac{\sqrt{3}\theta\Delta t_k}{\varrho(1 + H^2\sigma_y^2) + \theta\Delta t_k H_1 H^2\sigma_y^2} \right) |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|. \quad (4.42)$$

Beweis. Satz 4.5 zeigt

$$|\boldsymbol{\epsilon}(\mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta})| \leq \frac{1}{2\mu} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}| + |\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1+\theta}|.$$

Mit Satz 4.13 gilt

$$\begin{aligned} & |\boldsymbol{\epsilon}(\mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta})| \\ & \leq \frac{1}{2\mu} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}| + \frac{\sqrt{3}\theta\Delta t_k}{\varrho(1 + H^2\sigma_y^2) + \theta\Delta t_k H_1 H^2\sigma_y^2} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|. \end{aligned}$$

■

Satz 4.16. *Unter den Voraussetzungen aus Definition 4.8 gilt fast überall in $\omega \subset \Omega$, daß sich für kinematische Verfestigung der Verzerrungsfehler folgendermaßen durch den Spannungsfehler abschätzen läßt,*

$$|\boldsymbol{\epsilon}_{k-1+\theta} - \boldsymbol{\epsilon}_{h,k-1+\theta}| \leq \left(\frac{1}{2\mu} + \frac{1}{k_1} \sqrt{2 + \frac{1}{2} \left(\frac{4\varrho + 2\theta\Delta t_k k_1}{\theta\Delta t_k k_1} \right)^2} \right) |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|. \quad (4.43)$$

Beweis. Mit Satz 4.5 und 4.14 folgt die Behauptung. ■

3. Fall: elastisch-plastisch

In diesem Unterabschnitt wird unter den Voraussetzungen der nachfolgenden Definition eine Abschätzung des Verzerrungsfehlers durch den Spannungsfehler bewiesen.

Definition 4.17 (Voraussetzungen 3). *Es wird vorausgesetzt, daß für das orts- und zeitdiskrete Problem 6 sowie für das zeitdiskrete Problem 5 in dem Zeitschritt $[t_{k-1}, t_k]$ dieselben Anfangswerte $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$ und $\boldsymbol{\alpha}_{h,k-1}$ (kinematische Verfestigung) bzw. $\alpha_{h,k-1}$ (isotrope Verfestigung) gegeben sind. Es wird der Fall betrachtet, daß die Lösung des des orts- und zeitdiskreten Problems 6 in einem Gebiet $\omega \subset \Omega$ elastisches Materialverhalten aufweist und die Lösung des zeitdiskreten Problems 5 zum Zeitpunkt $t_{k-1+\theta}$ in diesem Gebiet viskoplastisches Materialverhalten aufweist.*

Mit Satz 4.5 gilt

$$|\boldsymbol{\epsilon}(\mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta})| \leq \frac{1}{2\mu} |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}| + |\mathbf{p}_{k-1+\theta} - \mathbf{p}_{h,k-1+\theta}|. \quad (4.44)$$

Mit den Voraussetzungen aus Definition 4.17 gilt $\mathbf{p}_{h,k-1+\theta} = \mathbf{0}$. Da vorausgesetzt wird, daß die Anfangswerte exakt gegeben sind, müssen diese aus einem elastischen Zeitschritt stammen, da es andernfalls nicht möglich wäre, daß die numerische Lösung $\mathbf{u}_{h,k-1+\theta}$ elastisch ist. D.h. es gilt $\mathbf{p}_{h,k-1} = \mathbf{0}$. Es wird eine verallgemeinerte Spannung $\bar{\boldsymbol{\Sigma}}_{k-1+\theta} = (\bar{\boldsymbol{\sigma}}_{k-1+\theta}, \bar{\boldsymbol{\alpha}}_{k-1+\theta})$ definiert durch

$$\bar{\boldsymbol{\Sigma}}_{k-1+\theta} = \Pi_K \boldsymbol{\Sigma}_{k-1+\theta},$$

wobei Π_K die Projektion aus (2.8) auf die konvexe Menge K ist. Aus dem zeitdiskreten Fließgesetz

$$\frac{1}{\theta \Delta t_k} (\bar{\mathbf{P}}_{k-1+\theta} - \mathbf{P}_{h,k-1}) = \frac{1}{\varrho} (\bar{\boldsymbol{\Sigma}}_{k-1+\theta} - \Pi_K \bar{\boldsymbol{\Sigma}}_{k-1+\theta})$$

folgt dann für $\varrho > 0$, daß $\bar{\mathbf{p}}_{k-1+\theta} = \mathbf{0}$. Damit kann in (4.44) statt $\mathbf{p}_{h,k-1+\theta} = \mathbf{0}$ auch $\bar{\mathbf{p}}_{k-1+\theta} = \mathbf{0}$ eingesetzt werden. Mit den Sätzen 4.13 und 4.14 folgt, daß

$$|\mathbf{p}_{k-1+\theta} - \bar{\mathbf{p}}_{k-1+\theta}| \leq \frac{\sqrt{3}\theta \Delta t_k}{\varrho(1 + H^2\sigma_y^2) + \theta \Delta t_k H_1 H^2 \sigma_y^2} |\boldsymbol{\sigma}_{k-1+\theta} - \bar{\boldsymbol{\sigma}}_{k-1+\theta}| \quad (4.45)$$

und

$$|\mathbf{p}_{k-1+\theta} - \bar{\mathbf{p}}_{k-1+\theta}| \leq \frac{1}{k_1} \sqrt{2 + \frac{1}{2} \left(\frac{4\varrho + 2\theta \Delta t_k k_1}{\theta \Delta t_k k_1} \right)^2} |\boldsymbol{\sigma}_{k-1+\theta} - \bar{\boldsymbol{\sigma}}_{k-1+\theta}|. \quad (4.46)$$

Aufgrund der Definition der Projektion Π_K in (2.8) gilt

$$|\boldsymbol{\sigma}_{k-1+\theta} - \bar{\boldsymbol{\sigma}}_{k-1+\theta}| \leq |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|,$$

denn $\boldsymbol{\Sigma}_{h,k-1+\theta} = (\boldsymbol{\sigma}_{h,k-1+\theta}, \boldsymbol{\alpha}_{h,k-1+\theta})$ ist ein Element der konvexen Menge K . Aus (4.44), (4.45) und (4.46) folgt dann, daß in der Viskoplastizität für $\varrho > 0$ die Abschätzungen der Sätze 4.15 und 4.16 gelten.

Definition 4.18 (Voraussetzungen 4). *Es wird vorausgesetzt, daß für das orts- und zeitdiskrete Problem 6 und das zeitdiskrete Problem 5 in dem Zeitschritt $[t_{k-1}, t_k]$ dieselben Anfangswerte $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$ und $\boldsymbol{\alpha}_{h,k-1}$ (kinematische Verfestigung) bzw. $\alpha_{h,k-1}$ (isotrope Verfestigung) gegeben sind. Es wird der Fall betrachtet, daß die Lösung des orts- und zeitdiskreten Problems 6 in einem Gebiet $\omega \subset \Omega$ viskoplastisches Materialverhalten aufweist und die Lösung des zeitdiskreten Problems 5 zum Zeitpunkt $t_{k-1+\theta}$ in diesem Gebiet elastisches Materialverhalten aufweist.*

Mit den Voraussetzungen aus Definition 4.18 folgt sofort, daß $\mathbf{p}_{h,k-1} = \mathbf{0}$. Außerdem ist $\mathbf{p}_{k-1+\theta} = \mathbf{0}$. Es wird eine verallgemeinerte Spannung $\bar{\boldsymbol{\Sigma}}_{k-1+\theta} = (\bar{\boldsymbol{\sigma}}_{k-1+\theta}, \bar{\boldsymbol{\alpha}}_{k-1+\theta})$ definiert durch

$$\bar{\boldsymbol{\Sigma}}_{k-1+\theta} = \Pi_K \boldsymbol{\Sigma}_{h,k-1+\theta}$$

wobei Π_K die Projektion aus (2.8) ist. Aus dem zeitdiskreten Fließgesetz

$$\frac{1}{\theta \Delta t_k} (\bar{\mathbf{P}}_{k-1+\theta} - \mathbf{P}_{h,k-1}) = \frac{1}{\varrho} (\bar{\boldsymbol{\Sigma}}_{k-1+\theta} - \Pi_K \bar{\boldsymbol{\Sigma}}_{k-1+\theta})$$

folgt für $\varrho > 0$, daß $\bar{\mathbf{p}}_{k-1+\theta} = \mathbf{0}$. Dann kann in (4.44) statt $\mathbf{p}_{k-1+\theta} = \mathbf{0}$ auch $\bar{\mathbf{p}}_{k-1+\theta} = \mathbf{0}$ gesetzt werden. Mit den Sätzen 4.13 und 4.14 folgt, daß

$$|\bar{\mathbf{p}}_{k-1+\theta} - \bar{\mathbf{p}}_{k-1+\theta}| \leq \frac{\sqrt{3}\theta \Delta t_k}{\varrho(1 + H^2\sigma_y^2) + \theta \Delta t_k H_1 H^2 \sigma_y^2} |\bar{\boldsymbol{\sigma}}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}| \quad (4.47)$$

und

$$|\bar{\mathbf{p}}_{k-1+\theta} - \mathbf{p}_{h,k-1+\theta}| \leq \frac{1}{k_1} \sqrt{2 + \frac{1}{2} \left(\frac{4\varrho + 2\theta \Delta t_k k_1}{\theta \Delta t_k k_1} \right)^2} |\bar{\boldsymbol{\sigma}}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|. \quad (4.48)$$

Aufgrund der Definition der Projektion Π_K in (2.8) gilt

$$|\bar{\boldsymbol{\sigma}}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}| \leq |\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}|,$$

denn $\boldsymbol{\Sigma}_{k-1+\theta} = (\boldsymbol{\sigma}_{k-1+\theta}, \boldsymbol{\alpha}_{k-1+\theta})$ ist ein Element der konvexen Menge K . Aus (4.44), (4.47) und (4.48) folgt dann, daß in der Viskoplastizität für $\varrho > 0$ die Abschätzungen der Sätze 4.15 und 4.16 gelten.

4.2 Der Residuenschätzer

4.2.1 Zuverlässigkeit der Fehlerabschätzung

Der Residuenschätzer liefert eine zuverlässige Fehlerabschätzung für den Spannungsfehler in der Viskoplastizität mit Verfestigung. Durch die analytische Darstellung des Spannungstensors in jedem Zeitschritt, mit den in 4.1 gezeigten Abschätzungen für den Verzerrungsfehler durch den Spannungsfehler und mit der Monotonieabschätzung aus Satz 4.4 entspricht der Zuverlässigkeitsbeweis des Residuenschätzers in der Elastoviskoplastizität dem Fall der Elastizität. Der hier angegebene Beweis beruht auf Arbeiten von R. Verfürth [65] für h-verfeinerte Netze sowie J.M. Melenk und B. Wohlmuth [47] für p-Verfeinerung. Dieses Kapitel stellt den zweidimensionalen Spezialfall des ebenen Verzerrungszustandes dar. Da die Ergebnisse in Abschnitt 4.1 auch den dreidimensionalen Fall einschließen, ist eine Verallgemeinerung möglich.

Satz 4.19 ([30]). *Für eine reguläre Triangulierung \mathcal{T} eines Lipschitz-Gebietes Ω existiert eine Abbildung $J_h : H_0^1(\Omega) \rightarrow \mathcal{S}_0^1(\Omega)$ und es gibt Konstanten c_3, c_4 , so daß für alle $\mathbf{v} \in H_0^1(\Omega)$*

$$\sum_{T \in \mathcal{T}} h_T^{-2} \|\mathbf{v} - J_h \mathbf{v}\|_{L^2(T)}^2 \leq c_3^2 \|\nabla \mathbf{v}\|_{L^2(\Omega)}^2, \quad (4.49)$$

$$\sum_{E \in \mathcal{E}} h_E^{-1} \|\mathbf{v} - J_h \mathbf{v}\|_{L^2(E)}^2 \leq c_4^2 \|\nabla \mathbf{v}\|_{L^2(\Omega)}^2. \quad (4.50)$$

Der Operator J_h kann auf mehrere Arten definiert werden. In [25] wird von C. Carstensen und S.A. Funken folgende Definition gegeben.

Definition 4.20 ([25]). *Sei \mathcal{N} die Menge aller Knoten der Triangulierung \mathcal{T} und \mathcal{K} die Menge der freien Knoten. Für jeden Knoten sei φ_z die Hutfunktion zum Knoten z , so daß $\varphi_z(x) = 0$ für alle $x \in \mathcal{N} \setminus \{z\}$ und $\varphi_z(z) = 1$. Der Operator $J_h : H_0^1(\Omega) \rightarrow \mathcal{S}_0^1(\Omega)$ ist definiert durch*

$$J_h \mathbf{v} = \sum_{z \in \mathcal{K}} (\pi_z \mathbf{v}) \varphi_z, \quad (4.51)$$

wobei

$$\pi_z \mathbf{v} := \frac{\int_{\Omega_z} \mathbf{v} d\mathbf{x}}{\int_{\Omega_z} d\mathbf{x}}. \quad (4.52)$$

In [25] werden für die Definition in (4.51) des Interpolationsoperators J_h explizite Schranken für die Konstanten c_3 und c_4 in (4.49) und (4.50) angegeben.

Für den Fall der p-Verfeinerung wird von J.M. Melenk und B. Wohlmuth in [47] der Clément-Interpolationsoperator [30] verallgemeinert. Vorausgesetzt wird eine quasiuniforme Triangulierung \mathcal{T} , das bedeutet, für die Familie $\{\mathcal{T}_i\}_{i=1}^{\infty}$ existieren Konstanten $\kappa_1, \kappa_2 > 0$, unabhängig von i , so daß

$$\kappa_1 h(\mathcal{T}) \leq h_T \leq \kappa_2 \rho_T \quad (4.53)$$

für alle $T \in \mathcal{T}_i$, $i = 1, 2, \dots$. Für jeden Knoten $x \in \mathcal{N}$ der Triangulierung \mathcal{T} wird folgende Notation eingeführt:

$$\omega_x^0 := \{x\}, \quad (4.54)$$

$$\omega_x^j := \cup\{T | T \in \mathcal{T} \text{ und } T \cap \omega_x^{j-1} \neq \emptyset\}, j \geq 1, \quad (4.55)$$

$$h_x := \max\{h_T | x \in T\}, \quad (4.56)$$

$$p_x := \max\{p_T + 1 | x \in T\}, \quad (4.57)$$

$$\mathcal{E}_x := \{E \in \mathcal{E} | x \in E\}. \quad (4.58)$$

In folgendem Satz definieren J.M. Melenk und B. Wohlmuth einen Interpolationsoperator, der grundlegend ist für den Beweis der Zuverlässigkeit des Residuenschätzers unter Berücksichtigung von p-Verfeinerung.

Satz 4.21 ([47]). *Vorausgesetzt wird eine quasiuniforme Triangulierung. Dann existiert eine Konstante c_5 und ein linearer Operator $I_h : H_{loc}^1(\mathbb{R}^2) \rightarrow \mathcal{S}^p(\Omega)$, so daß für alle Knoten $x \in \mathcal{N}$ und alle Kanten $E \in \mathcal{E}$ die folgenden Abschätzungen gelten,*

$$\|\mathbf{v} - I_h \mathbf{v}\|_{L^2(\omega_x^1)} \leq c_5 \frac{h_x}{p_x} \|\nabla v\|_{L^2(\omega_x^4)}, \quad (4.59)$$

$$\|\mathbf{v} - I_h \mathbf{v}\|_{L^2(E)} \leq c_5 \frac{h_x^{1/2}}{p_x^{1/2}} \|\nabla v\|_{L^2(\omega_x^4)}. \quad (4.60)$$

Definition 4.22. *Das Residuum $R(\mathbf{u}_{h,k-1+\theta})$ ist definiert als ein Element des Dualraums von $H^1(\Omega)$,*

$$\langle R(\mathbf{u}_{h,k-1+\theta}), \mathbf{v} \rangle := \int_{\Omega} \mathbf{f}(t_{k-1+\theta}) \cdot \mathbf{v} d\mathbf{x} + \int_{\Gamma_N} \mathbf{g}(t_{k-1+\theta}) \cdot \mathbf{v} ds - \int_{\Omega} \boldsymbol{\sigma}_{h,k-1+\theta} : \boldsymbol{\epsilon}(\mathbf{v}) d\mathbf{x}. \quad (4.61)$$

Definition 4.23 (Kantenresiduum). *Es wird \mathcal{E}_{Ω} als die Menge der inneren Kanten definiert, und \mathcal{E}_N bzw. \mathcal{E}_D werden als die Menge der Kanten auf dem Neumann- bzw. Dirichlet-Rand definiert. Für eine innere Kante $E \in \mathcal{E}_{\Omega}$ werden mit T_E^+ und T_E^- die Elemente bezeichnet, für die $E = \partial T_E^+ \cap \partial T_E^-$ und mit \mathbf{n}_E der äußere Normalenvektor von T_E^- zu dieser Kante. Für eine Kante $E \in \mathcal{E}_N$ auf dem Neumann-Rand ist \mathbf{n}_E der äußere Normalenvektor zur Kante E . Es werden Kantensprünge*

$\mathbf{R} : \mathcal{E} \rightarrow \mathbb{R}^d$ definiert durch

$$\mathbf{R}|_E := [\boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_E] = \begin{cases} \mathbf{0} & \text{für } E \in \mathcal{E}_D, \\ \mathbf{g}(t_{k-1+\theta}) - \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_E & \text{für } E \in \mathcal{E}_N, \\ (\boldsymbol{\sigma}_{h,k-1+\theta}^+ - \boldsymbol{\sigma}_{h,k-1+\theta}^-) \cdot \mathbf{n}_E & \text{für } E \in \mathcal{E}_\Omega. \end{cases} \quad (4.62)$$

Definition 4.24 (Volumenresiduum). Das Volumenresiduum $\mathbf{r} : \mathcal{T} \rightarrow \mathbb{R}^d$ ist definiert durch

$$\mathbf{r}|_T := \mathbf{f}(t_{k-1+\theta})|_T + \operatorname{div} \boldsymbol{\sigma}_{h,k-1+\theta}|_T. \quad (4.63)$$

Die folgenden beiden Sätze finden sich z.B. in [19].

Satz 4.25 (1. Kornsche Ungleichung). Ist Ω eine offene, beschränkte Menge in \mathbb{R}^d mit stückweise glattem Rand, dann gibt es eine Konstante $C_K = C_K(\Omega)$, so daß

$$C_K \|\mathbf{v}\|_{H^1(\Omega)}^2 \leq \|\mathbf{v}\|_{L^2(\Omega)}^2 + \int_{\Omega} \boldsymbol{\epsilon}(\mathbf{v}) : \boldsymbol{\epsilon}(\mathbf{v}) \, d\mathbf{x}. \quad (4.64)$$

Satz 4.26 (2. Kornsche Ungleichung). Ist Ω eine offene, beschränkte Menge in \mathbb{R}^d mit stückweise glattem Rand und hat $\Gamma_D \subset \Gamma$ ein positives Maß, dann gibt es eine Konstante $c_K(\Omega, \Gamma_D)$, so daß für alle $\mathbf{v} \in H_0^1(\Omega)$

$$c_K \|\mathbf{v}\|_{H^1(\Omega)}^2 \leq \int_{\Omega} |\boldsymbol{\epsilon}(\mathbf{v})|^2 \, d\mathbf{x}. \quad (4.65)$$

Definition 4.27 (Fehlerschätzer η_R). Der Residuenschätzer besteht aus Volumentermen η_T und Kantentermen η_E , die definiert sind durch

$$\eta_T := \frac{h_T}{p_T} \|\mathbf{r}\|_{L^2(T)} \quad \text{und} \quad \eta_E := \frac{h_E^{1/2}}{p_E^{1/2}} \|\mathbf{R}\|_{L^2(E)}. \quad (4.66)$$

Für $E = \partial T_1 \cap \partial T_2$ ist $p_E = \max\{p_{T_1}, p_{T_2}\}$. Der Fehlerschätzer η_R ist definiert durch

$$\eta_R := \frac{1}{\sqrt{c_K}} (\sqrt{c_6} (\sum_{T \in \mathcal{T}} \eta_T^2)^{1/2} + \sqrt{c_7} (\sum_{E \in \mathcal{E}} \eta_E^2)^{1/2}). \quad (4.67)$$

Dabei ist c_K die Konstante aus der 2. Kornschen Ungleichung. Für die Konstanten c_6 und c_7 gilt bei h -Verfeinerung $c_6 = c_3$, $c_7 = c_4$ (mit Satz 4.19). Für p -Verfeinerung sind die Konstanten c_6 und c_7 abhängig von c_5 (mit Satz 4.21).

Satz 4.28. Für gegebene Anfangswerte $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$ gilt für einen elastischen Zeitschritt $[t_{k-1}, t_k]$ die Spannungsfehlerabschätzung

$$\|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} \leq \left(1 + \frac{\lambda}{\mu}\right) \eta_R. \quad (4.68)$$

Für gegebene Anfangswerte $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$, α_{k-1} gilt im Zeitschritt $[t_{k-1}, t_k]$ bei Visko-plastizität mit isotroper Verfestigung die Spannungsfehlerabschätzung

$$\|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} \leq (2\lambda + 2\mu) \left(\frac{1}{2\mu} + \frac{\sqrt{3}\theta \Delta t_k}{\varrho(1 + H^2 \sigma_y^2) + \theta \Delta t_k H_1 H^2 \sigma_y^2} \right) \eta_R. \quad (4.69)$$

Für gegebene Anfangswerte $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$, $\boldsymbol{\alpha}_{k-1}$ gilt im Zeitschritt $[t_{k-1}, t_k]$ bei Viskoplastizität mit kinematischer Verfestigung die Spannungsfehlerabschätzung

$$\|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} \leq (2\lambda + 2\mu) \left(\frac{1}{2\mu} + \frac{1}{k_1} \sqrt{2 + \frac{1}{2} \left(\frac{4\varrho + 2\theta \Delta t_k k_1}{\theta \Delta t_k k_1} \right)^2} \right) \eta_R. \quad (4.70)$$

Beweis. Der Beweis folgt [65] und [47] und nutzt die in Abschnitt 4.1 bewiesenen Sätze 4.7, 4.15 bzw. 4.16. Mit Satz 4.4 und $\mathbf{e} := \mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta}$ gilt

$$\frac{1}{2\lambda + 2\mu} \|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)}^2 \leq \langle R(\mathbf{u}_{h,k-1+\theta}), \mathbf{e} \rangle.$$

Die Spannung $\boldsymbol{\sigma}_{h,k-1+\theta}$ ist elementweise stetig und daher elementweise schwach differenzierbar. Es gilt

$$\begin{aligned} & \langle R(\mathbf{u}_{h,k-1+\theta}), \mathbf{e} \rangle \\ &= \int_{\Omega} \mathbf{f}(t_{k-1+\theta}) \cdot \mathbf{e} \, d\mathbf{x} + \int_{\Gamma_N} \mathbf{g}(t_{k-1+\theta}) \cdot \mathbf{e} \, ds + \sum_{T \in \mathcal{T}} \int_T \operatorname{div}_{\mathcal{T}} \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{e} \, d\mathbf{x} \\ & \quad - \int_{\partial T} (\boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}) \cdot \mathbf{e} \, ds \\ &= \int_{\Omega} \mathbf{f}(t_{k-1+\theta}) \cdot \mathbf{e} \, d\mathbf{x} + \int_{\Gamma_N} \mathbf{g}(t_{k-1+\theta}) \cdot \mathbf{e} \, ds + \sum_{T \in \mathcal{T}} \int_T \operatorname{div}_{\mathcal{T}} \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{e} \, d\mathbf{x} \\ & \quad - \sum_{E \in \mathcal{E}_N} \int_E (\boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_E) \cdot \mathbf{e} \, ds - \sum_{E \in \mathcal{E}_\Omega} \int_E ((\boldsymbol{\sigma}_{h,k-1+\theta}|_{T_E^-} - \boldsymbol{\sigma}_{h,k-1+\theta}|_{T_E^+}) \cdot \mathbf{n}_E) \cdot \mathbf{e} \, ds \\ &= \int_{\Omega} \mathbf{r} \cdot \mathbf{e} \, d\mathbf{x} + \int_{\cup \mathcal{E}} \mathbf{R} \cdot \mathbf{e} \, ds. \end{aligned} \quad (4.71)$$

Setze $\mathcal{J} := J_h$ für h-Verfeinerung und $\mathcal{J} := I_h$ für p-Verfeinerung. Dann gilt aufgrund der Galerkin-Orthogonalität

$$\begin{aligned} & \langle R(\mathbf{u}_{h,k-1+\theta}), \mathbf{e} \rangle \\ & \leq \sum_{T \in \mathcal{T}} \left\| \frac{h_T}{p_T} \mathbf{r} \right\|_{2,T} \left\| \frac{p_T}{h_T} (\mathbf{e} - \mathcal{J}\mathbf{e}) \right\|_{2,T} + \sum_{E \in \mathcal{E}} \left\| \frac{h_E^{1/2}}{p_E^{1/2}} \mathbf{R} \right\|_{2,E} \left\| \frac{p_E^{1/2}}{h_E^{1/2}} (\mathbf{e} - \mathcal{J}\mathbf{e}) \right\|_{2,E} \\ & \leq \left(\sum_{T \in \mathcal{T}} \left\| \frac{h_T}{p_T} \mathbf{r} \right\|_{2,T}^2 \right)^{1/2} \left(\sum_{T \in \mathcal{T}} \frac{p_T^2}{h_T^2} \|\mathbf{e} - \mathcal{J}\mathbf{e}\|_{2,T}^2 \right)^{1/2} \\ & \quad + \left(\sum_{E \in \mathcal{E}} \left\| \frac{h_E^{1/2}}{p_E^{1/2}} \mathbf{R} \right\|_{2,\cup \mathcal{E}}^2 \right)^{1/2} \left(\sum_{E \in \mathcal{E}} \frac{p_E}{h_E} \|\mathbf{e} - \mathcal{J}\mathbf{e}\|_{2,E}^2 \right)^{1/2}. \end{aligned}$$

Aus (4.49), (4.50) und mit den Definitionen für η_T und η_E folgt im Fall der h-Verfeinerung

$$\frac{1}{2\lambda + 2\mu} \|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)}^2 \leq \sqrt{c_6} \left(\sum_{T \in \mathcal{T}} \eta_T^2 \right)^{1/2} \|\mathbf{e}\|_{H^1(\Omega)} + \sqrt{c_7} \left(\sum_{E \in \mathcal{E}} \eta_E^2 \right)^{1/2} \|\mathbf{e}\|_{H^1(\Omega)}. \quad (4.72)$$

Im Fall der p-Verfeinerung folgt aufgrund der Quasiuniformität des Netzes aus (4.59), (4.60)

$$\begin{aligned} \frac{p_T}{h_T} \|\mathbf{e} - I_h \mathbf{e}\|_{L^2(T)} &\leq \frac{c_5}{\kappa_1} \|\nabla \mathbf{e}\|_{L^2(\omega_z^4)}, \quad \forall T \in \mathcal{T}, z \in T, \\ \frac{p_E^{1/2}}{h_E^{1/2}} \|\mathbf{e} - I_h \mathbf{e}\|_{L^2(E)} &\leq \frac{c_5}{\kappa_1} \|\nabla \mathbf{e}\|_{L^2(\omega_z^4)}, \quad \forall E \in \mathcal{E}, z \in E, \end{aligned}$$

für $p_E = \max\{p_{T_1}, p_{T_2}\}$, falls $E = \partial T_1 \cap \partial T_2$. Damit gilt (4.72) auch für p-Verfeinerung. Mit der 2. Kornschen Ungleichung folgt,

$$\frac{1}{2\lambda + 2\mu} \|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)}^2 \leq \left(\sqrt{\frac{c_6}{c_K}} \left(\sum_{T \in \mathcal{T}} \eta_T^2 \right)^{1/2} + \sqrt{\frac{c_7}{c_K}} \left(\sum_{E \in \mathcal{E}} \eta_E^2 \right)^{1/2} \right) \|\boldsymbol{\epsilon}(\mathbf{e})\|_{L^2(\Omega)}.$$

Die Sätze 4.7, 4.15 bzw. 4.16 werden zur Absorption der rechten Seite verwendet. ■

4.2.2 Effizienz der Fehlerabschätzung

Effizienzbeweise für den Residuenschätzer in der Elastizität bei h-Verfeinerung beruhen auf Arbeiten von Verfürth [62, 63, 65] und lassen sich auf die Elastoviskoplastizität übertragen, wenn man die explizite Darstellung des Spannungstensors berücksichtigt (Sätze 3.1- 3.3).

Satz 4.29 ([65]). *Es wird vorausgesetzt, daß die Anfangswerte im Zeitschritt exakt gegeben sind. Dann ist die L^2 -Norm des Volumenresiduums, $\|\mathbf{r}\|_{L^2(T)}$, lokal durch den Fehler der Finite-Elemente-Diskretisierung beschränkt,*

$$\|\mathbf{r}\|_{L^2(T)} \leq c_8 (h_T^{-1} \|\mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta}\|_{H^1(T)} + \|\mathbf{r} - \bar{\mathbf{r}}\|_{L^2(T)}). \quad (4.73)$$

Mit $\bar{\mathbf{r}}$ ist eine endlichdimensionale Approximation für das Volumenresiduum \mathbf{r} bezeichnet, die auf jedem Element stetig ist.

Satz 4.30 ([65]). *Vorausgesetzt wird, daß Anfangswerte im Zeitschritt exakt gegeben sind. Die L^2 -Norm des Kantenresiduums, $\|\mathbf{R}\|_{L^2(E)}$, ist dann lokal durch den Fehler der Finite-Elemente-Diskretisierung beschränkt,*

$$\|\mathbf{R}\|_{L^2(E)} \leq c_9 (h_E^{-1/2} \|\mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta}\|_{H^1(\tilde{E})} + h_E^{1/2} \|\mathbf{r} - \bar{\mathbf{r}}\|_{L^2(\tilde{E})} + \|\bar{\mathbf{R}} - \mathbf{R}\|_{L^2(E)}). \quad (4.74)$$

Mit $\bar{\mathbf{R}}$ ist eine endlichdimensionale Näherung für das Kantenresiduum \mathbf{R} bezeichnet, die auf jeder Kante stetig ist. Das Gebiet \tilde{E} ist der Patch zur Kante E , definiert durch $\tilde{E} = \cup\{T' \in \mathcal{T} : E \subset T'\}$.

In den oberen Abschätzungen für Volumen- und Kantenresiduum treten Terme $\|\mathbf{r} - \bar{\mathbf{r}}\|_{L^2(T)}$ und $\|\mathbf{R} - \bar{\mathbf{R}}\|_{L^2(E)}$ auf, die nur dann klein werden, wenn sich Volumen und Kantenresiduum gut durch Polynome approximieren lassen. Dies ist für elastische Elemente bzw. Kanten gegeben und im Fall der Plastizität ($\rho \rightarrow 0$) auch für komplett plastische Elemente bzw. Kanten zu beiden angrenzenden Elementen.

In der Arbeit Arbeit [47] von Melenk und Wohlmuth wird gezeigt, daß bei p-Verfeinerung die Eigenschaften der Zuverlässigkeit und der Effizienz nicht simultan

für den Residuenschätzer bewiesen werden können. Dies liegt daran, daß in inversen Abschätzungen die Abhängigkeit vom Polynomgrad nicht in derselben Potenz vorkommt wie die Abhängigkeit von dem Elementdurchmesser.

4.3 Die Methode der äquilibrierten Residuen

Die Methode der äquilibrierten Residuen beruht auf Arbeiten von D. Ladeveze und D. Leguillon [43], M. Ainsworth und J.T. Oden [4, 1, 2, 3], R.E. Bank und A. Weiser [17], sowie T. Strouboulis, I. Babuška und K. Gangaraj [59].

4.3.1 Zuverlässigkeit der Fehlerabschätzung

Die Methode der äquilibrierten Residuen ist ein impliziter a posteriori Fehlerschätzer, der eine obere Schranke für den Spannungsfehler liefert. Es sind elementweise Neumannprobleme zu lösen, die Lösbarkeit dieser Probleme wird für finite Elemente der Ordnung $p \geq 1$ durch die Konstruktion äquilibrierter Randkräfte garantiert. In diesem Abschnitt wird der zweidimensionale Spezialfall behandelt. Die Technik der Methode der äquilibrierten Residuen wird in diesem Abschnitt in Anlehnung an das Buch [4] beschrieben. Die Anwendbarkeit auf das elastoviskoplastische Problem ist eine Konsequenz aus den Ergebnissen des Abschnitts 4.1. Effizienzbeweise werden finiten Elementen höherer Ordnung angepaßt.

Definition 4.31 (Äquilibrierte Randkräfte). *Es werden Randkräfte $\mathbf{g}_T(t_{k-1+\theta}) \in L^2(\partial T)$ für $T \in \mathcal{T}$ definiert, mit den Eigenschaften*

$$\mathbf{g}_T(t_{k-1+\theta}) = \mathbf{g}(t_{k-1+\theta}) \quad \text{auf } \partial T \cap \Gamma_N, \quad (4.75)$$

$$\mathbf{g}_T(t_{k-1+\theta}) + \mathbf{g}_{T'}(t_{k-1+\theta}) = \mathbf{0} \quad \text{auf } \partial T \cap \partial T'. \quad (4.76)$$

Definition 4.32 (Lokale Probleme). *Mit V_T wird der Funktionenraum $V_T := \{\mathbf{v} \in H^1(T)^2 : \mathbf{v} = \mathbf{0} \text{ auf } \Gamma_D \cap \partial T\}$ bezeichnet. Das Funktional $l_{k-1+\theta,T}(\mathbf{v}) : V_T \rightarrow \mathbb{R}$ wird definiert als*

$$l_{k-1+\theta,T}(\mathbf{v}) := \int_T \mathbf{f}(t_{k-1+\theta}) \cdot \mathbf{v} \, d\mathbf{x} + \int_{\partial T} \mathbf{g}_T(t_{k-1+\theta}) \cdot \mathbf{v} \, ds - \int_T \boldsymbol{\sigma}_{h,k-1+\theta} : \boldsymbol{\epsilon}(\mathbf{v}) \, d\mathbf{x}, \quad (4.77)$$

und die Bilinearform $a_T(\mathbf{y}_T, \mathbf{v}) : V_T \times V_T \rightarrow \mathbb{R}$ wird definiert als

$$a_T(\mathbf{y}_T, \mathbf{v}) := \int_T \boldsymbol{\epsilon}(\mathbf{y}_T) : \boldsymbol{\epsilon}(\mathbf{v}) \, d\mathbf{x}. \quad (4.78)$$

Es werden lokale Probleme folgendermaßen definiert. Man bestimme $\mathbf{y}_T \in V_T$, so daß für alle $\mathbf{v} \in V_T$

$$a_T(\mathbf{y}_T, \mathbf{v}) = l_{k-1+\theta,T}(\mathbf{v}). \quad (4.79)$$

Definition 4.33 (Fehlerschätzer η_{EQ} und Verfeinerungsindikator $\eta_{EQ,T}$). *Der Fehlerschätzer η_{EQ} und der Verfeinerungsindikator $\eta_{EQ,T}$ werden definiert durch*

$$\eta_{EQ,T}^2 := a_T(\mathbf{y}_T, \mathbf{y}_T), \quad (4.80)$$

$$\eta_{EQ} := \left(\sum_{T \in \mathcal{T}} \eta_{EQ,T}^2 \right)^{1/2}. \quad (4.81)$$

Satz 4.34. Für gegebene Anfangswerte $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$ gilt für einen elastischen Zeitschritt $[t_{k-1}, t_k]$ die Spannungsfehlerabschätzung

$$\|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} \leq \left(1 + \frac{\lambda}{\mu}\right) \eta_{EQ}. \quad (4.82)$$

Für gegebene Anfangswerte $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$, α_{k-1} gilt im Zeitschritt $[t_{k-1}, t_k]$ bei Viskoplastizität mit isotroper Verfestigung die Spannungsfehlerabschätzung

$$\|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} \leq (2\lambda + 2\mu) \left(\frac{1}{2\mu} + \frac{\sqrt{3}\theta\Delta t_k}{\varrho(1 + H^2\sigma_y^2) + \theta\Delta t_k H_1 H^2\sigma_y^2} \right) \eta_{EQ}. \quad (4.83)$$

Für gegebene Anfangswerte $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$, α_{k-1} gilt im Zeitschritt $[t_{k-1}, t_k]$ bei Viskoplastizität mit kinematischer Verfestigung die Spannungsfehlerabschätzung

$$\|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} \leq (2\lambda + 2\mu) \left(\frac{1}{2\mu} + \frac{1}{k_1} \sqrt{2 + \frac{1}{2} \left(\frac{4\varrho + 2\theta\Delta t_k k_1}{\theta\Delta t_k k_1} \right)^2} \right) \eta_{EQ}. \quad (4.84)$$

Beweis. Mit Satz 4.4 und $\mathbf{e} := \mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta}$ gilt

$$\frac{1}{2\lambda + 2\mu} \|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_2^2 \leq \langle R(\mathbf{u}_{h,k-1+\theta}), \mathbf{e} \rangle. \quad (4.85)$$

Es gilt mit Definition 4.32, Definition 4.31 sowie Anwendung von Hölder-Ungleichung und diskreter Hölder-Ungleichung

$$\begin{aligned} \langle R(\mathbf{u}_{h,k-1+\theta}), \mathbf{e} \rangle &= \sum_{T \in \mathcal{T}} l_{k-1+\theta,T}(\mathbf{e}) = \sum_{T \in \mathcal{T}} \int_T \boldsymbol{\epsilon}(\mathbf{y}_T) : \boldsymbol{\epsilon}(\mathbf{e}) \, d\mathbf{x} \\ &\leq \sum_{T \in \mathcal{T}} \|\boldsymbol{\epsilon}(\mathbf{y}_T)\|_{2,T} \|\boldsymbol{\epsilon}(\mathbf{e})\|_{2,T} \leq \left(\sum_{T \in \mathcal{T}} \|\boldsymbol{\epsilon}(\mathbf{y}_T)\|_{2,T}^2 \right)^{1/2} \left(\sum_{T \in \mathcal{T}} \|\boldsymbol{\epsilon}(\mathbf{e})\|_{2,T}^2 \right)^{1/2} = \eta_{EQ} \|\boldsymbol{\epsilon}(\mathbf{e})\|_2. \end{aligned}$$

Zur Absorption der rechten Seite werden die Sätze 4.7, 4.15 bzw. 4.16 verwendet. ■

4.3.2 Lösung der lokalen Probleme

Lösbarkeit

Falls $|\partial T \cap \Gamma_D| \neq 0$ folgt die eindeutige Lösbarkeit der lokalen Probleme aus dem Lax-Milgram-Lemma. Falls $|\partial T \cap \Gamma_D| = 0$ handelt es sich bei den lokalen Problemen (4.79) um Neumann-Probleme. In diesem Fall kann keine eindeutige Lösung existieren, denn addiert man zu einer Lösung \mathbf{y}_T eine beliebige Starrkörperbewegung $\mathbf{v} \in Z_T$ mit $Z_T := \{\mathbf{z} \in V_T : a(\mathbf{v}, \mathbf{z}) = 0 \quad \forall \mathbf{v} \in V_T\}$, erhält man wieder eine Lösung. Eine Anwendung des Lax-Milgram-Lemmas benötigt das Konzept der Quotientenräume [48]. Es wird der Raum $X_T = H^1(T)/Z_T$ definiert, das ist $H^1(T)$

modulo Starrkörperbewegungen. In diesem Raum soll das Lax-Milgram-Lemma angewendet werden, um eine eindeutige Lösung in dem Quotientenraum zu beweisen. In X_T ist eine Norm definiert durch

$$\|[\mathbf{u}]\|_{X_T} = \|[\mathbf{u}]\|_{H^1(T)/Z_T} := \inf_{\mathbf{z} \in Z_T} \|\mathbf{u} + \mathbf{z}\|_{H^1(T)}. \quad (4.86)$$

Die Bilinearform im Quotientenraum ist unabhängig von einem bestimmten Repräsentanten gegeben durch

$$a_T([\mathbf{u}], [\mathbf{v}]) = \int_T \boldsymbol{\epsilon}(\mathbf{u}) : \boldsymbol{\epsilon}(\mathbf{v}) d\mathbf{x} \quad \text{mit } \mathbf{u} \in [\mathbf{u}], \mathbf{v} \in [\mathbf{v}]. \quad (4.87)$$

Die Bilinearform ist stetig, denn aus

$$a_T(\mathbf{u}, \mathbf{v}) \leq \|\boldsymbol{\epsilon}(\mathbf{u})\|_{L^2(T)} \|\boldsymbol{\epsilon}(\mathbf{v})\|_{L^2(T)} \leq \|\mathbf{u}\|_{H^1(T)} \|\mathbf{v}\|_{H^1(T)}$$

folgt

$$a_T([\mathbf{u}], [\mathbf{v}]) \leq \inf_{\mathbf{u} \in [\mathbf{u}]} \|\mathbf{u}\|_{H^1(T)} \inf_{\mathbf{v} \in [\mathbf{v}]} \|\mathbf{v}\|_{H^1(T)}. \quad (4.88)$$

Der Beweis der Elliptizität der Bilinearform in dem Quotientenraum X_T benötigt die 1. Kornsche Ungleichung und eine Variante davon.

Satz 4.35 (Variante der Kornschen Ungleichung). *Ist Ω eine offene, beschränkte Menge in \mathbb{R}^2 mit stückweise glattem Rand und Z_Ω die Menge aller Starrkörperbewegungen auf Ω , dann gibt es eine Konstante $c_{10} > 0$, so daß*

$$c_{10} \|\mathbf{v}\|_{H^1(\Omega)^2/Z_\Omega} \leq \|\boldsymbol{\epsilon}(\mathbf{v})\|_{L^2(\Omega)}. \quad (4.89)$$

Beweis. Angenommen, die Behauptung sei falsch. Dann gibt es eine Folge $\mathbf{v}_n \in H^1(\Omega)^2/Z_\Omega$ mit

$$\|\boldsymbol{\epsilon}(\mathbf{v}_n)\|_{L^2(\Omega)}^2 < \frac{1}{n} \quad \text{und} \quad \|\mathbf{v}_n\|_{H^1(\Omega)^2/Z_\Omega} = 1.$$

Weil $H^1(\Omega)$ in $H^0(\Omega)$ kompakt ist, gibt es eine Teilfolge der beschränkten Folge \mathbf{v}_n , die in $H^0(\Omega) = L^2(\Omega)$ konvergiert. Aufgrund der 1. Kornschen Ungleichung gilt

$$\begin{aligned} C_K \|\mathbf{v}_{n_i} - \mathbf{v}_{n_j}\|_{H^1(\Omega)}^2 &\leq \|\boldsymbol{\epsilon}(\mathbf{v}_{n_i} - \mathbf{v}_{n_j})\|_{L^2(\Omega)}^2 + \|\mathbf{v}_{n_i} - \mathbf{v}_{n_j}\|_{L^2(\Omega)}^2 \\ &\leq \frac{2}{n_i} + \frac{2}{n_j} + \|\mathbf{v}_{n_i} - \mathbf{v}_{n_j}\|_{L^2(\Omega)}^2. \end{aligned}$$

Damit ist die Teilfolge \mathbf{v}_{n_i} eine Cauchy-Folge in $H^1(\Omega)$ und, da $H^1(\Omega)$ ein Banach-Raum ist, auch konvergent gegen ein $\mathbf{u}_0 \in H^1(\Omega)$. Es gilt

$$\lim_{n_i \rightarrow \infty} \|\boldsymbol{\epsilon}(\mathbf{v}_{n_i})\|_{L^2(\Omega)} = \|\boldsymbol{\epsilon}(\mathbf{u}_0)\|_{L^2(\Omega)} = 0.$$

Daher ist \mathbf{u}_0 eine Starrkörperbewegung, ein Widerspruch zu

$$1 = \|\mathbf{v}_{n_i}\|_{H^1(\Omega)^2/Z_\Omega} \rightarrow \|\mathbf{u}_0\|_{H^1(\Omega)^2/Z_\Omega} = \inf_{\mathbf{z} \in Z_\Omega} \|\mathbf{u}_0 + \mathbf{z}\|_{H^1(\Omega)^2} = 0.$$

■

Die Elliptizität der Bilinearform $a_T([\mathbf{u}], [\mathbf{u}])$ folgt nun aus Satz 4.35. Es gilt

$$a_T([\mathbf{u}], [\mathbf{u}]) = \int_T \boldsymbol{\epsilon}(\mathbf{u}) : \boldsymbol{\epsilon}(\mathbf{u}) \, d\mathbf{x} \geq c_{10} \|[\mathbf{u}]\|_{H^1(T)/Z_T}^2. \quad (4.90)$$

Der Beweis des Lax-Milgram Lemmas im Quotientenraum X_T erfordert weiterhin die Stetigkeit des Funktionals der rechten Seite, $l_{k-1+\theta,T}(\mathbf{v})$. Damit dieses Funktional wie die Bilinearform unabhängig von einem bestimmten Repräsentanten in dem Quotientenraum definiert ist, muß gefordert werden, daß

$$l_{k-1+\theta,T}(\mathbf{v}) = 0 \quad \text{für alle } \mathbf{v} \in Z_T. \quad (4.91)$$

Das Funktional $l_{k-1+\theta,T}(\mathbf{v})$ ist stetig. Der Beweis benötigt den Spursatz.

Satz 4.36 (Spursatz [33]). *Für ein beschränktes Gebiet Ω mit einem C^1 -Rand $\partial\Omega$ existiert ein beschränkter linearer Operator*

$$\gamma : W^{1,p}(\Omega) \rightarrow L^p(\partial\Omega),$$

so daß

$$\gamma u = u|_{\partial\Omega} \quad \text{falls } u \in W^{1,p}(\Omega) \cap C(\bar{\Omega}) \quad (4.92)$$

und

$$\|\gamma u\|_{L^p(\partial\Omega)} \leq c_{11} \|u\|_{W^{1,p}(\Omega)}. \quad (4.93)$$

Beweis. [33] ■

Mit der Cauchy-Ungleichung und der Spurgleichung (4.93) gilt

$$\begin{aligned} & |l_{k-1+\theta,T}([\mathbf{v}])| \\ &= \left| \int_T \mathbf{f}(t_{k-1+\theta}) \cdot \mathbf{v} \, d\mathbf{x} + \int_{\partial T} \mathbf{g}_T(t_{k-1+\theta}) \cdot \gamma \mathbf{v} \, ds - \int_T \boldsymbol{\sigma}_{h,k-1+\theta} : \boldsymbol{\epsilon}(\mathbf{v}) \, d\mathbf{x} \right| \\ &\leq \|\mathbf{f}(t_{k-1+\theta})\|_{L^2(T)} \|\mathbf{v}\|_{L^2(T)} + \|\mathbf{g}_T(t_{k-1+\theta})\|_{L^2(\partial T)} \|\gamma \mathbf{v}\|_{L^2(\partial T)} \\ &\quad + \|\boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(T)} \|\boldsymbol{\epsilon}(\mathbf{v})\|_{L^2(T)} \\ &\leq (\|\mathbf{f}(t_{k-1+\theta})\|_{L^2(T)} + c_{11} \|\mathbf{g}_T(t_{k-1+\theta})\|_{L^2(\partial T)} + \|\boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(T)}) \|\mathbf{v}\|_{H^1(T)}. \end{aligned} \quad (4.94)$$

Mit (4.88), (4.90) und (4.94) folgt aus dem Lax-Milgram-Lemma die eindeutige Lösung des folgenden lokalen Neumann-Problems im Quotientenraum $H^1(T)/Z_T$: Finde ein $[\mathbf{y}_T] \in H^1(T)/Z_T$, so daß

$$a(\mathbf{y}_T, \mathbf{v}) = l_{k-1+\theta,T}(\mathbf{v}) = 0 \quad \text{für alle } \mathbf{v} \in H^1(T). \quad (4.95)$$

Der Raum Z_T enthält die linearen Starrkörperbewegungen, d.h. $\mathbf{z} \in Z_T$, falls $\mathbf{z}(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x} + \mathbf{b}$, wobei \mathbf{A} eine Drehmatrix und \mathbf{b} ein konstanter Vektor ist. Mit \mathcal{N} sind die Knoten der Triangulierung \mathcal{T} bezeichnet. Es gilt $\sum_{n \in \mathcal{N}} \varphi_n(x) = 1$ in Ω . Mit $\mathcal{N}(T)$ sind die Knoten von Element T bezeichnet und mit $\mathcal{N}(\gamma)$ die Knoten der Kante γ . Die Randkräfte $\mathbf{g}(t_{k-1+\theta})_T$ sind so zu konstruieren, daß die

Äquilibrierungsbedingung 1. Ordnung erfüllt ist. Der Raum Z_T ist ein Teilraum von $\text{span}\{\phi_i\}$, $i \in I_1(T)$. Es werden Randkräfte $\mathbf{g}(t_{k-1+\theta})_T$ konstruiert, so daß

$$l_{k-1+\theta,T}\left(\begin{pmatrix} \varphi_l \\ 0 \end{pmatrix}\right) = l_{k-1+\theta,T}\left(\begin{pmatrix} 0 \\ \varphi_l \end{pmatrix}\right) = 0 \quad \forall l \in \mathcal{N}(T), \quad (4.96)$$

$$\mathbf{g}_T(t_{k-1+\theta}) + \mathbf{g}_{T'}(t_{k-1+\theta}) = 0 \quad \text{auf } \partial T \cap \partial T', \quad (4.97)$$

$$\mathbf{g}_T(t_{k-1+\theta}) = \mathbf{g}(t_{k-1+\theta}) \quad \text{auf } \partial T \cap \Gamma_N. \quad (4.98)$$

Der folgende Algorithmus nach M. Ainsworth und J. T. Oden [4] dient der Bestimmung der Randkräfte $\mathbf{g}_T(t_{k-1+\theta})$. Die Berechnung der Randkräfte erfolgt dabei durch Lösung lokaler Probleme.

Definition 4.37. Mit $I_1(\gamma)$ ist die Menge der linearen Freiheitsgrade von Kante γ bezeichnet. Die Momente der Randkräfte sind definiert durch

$$\mu_{T,l}^\gamma = \int_\gamma \mathbf{g}_T(t_{k-1+\theta}) \cdot \phi_l \, ds \quad \text{für } l \in I_1(\gamma). \quad (4.99)$$

Für die Randkräfte wird der Ansatz

$$\mathbf{g}_T(t_{k-1+\theta})_T|_\gamma := \sum_{i=1}^4 \alpha_{l_i} \phi_{l_i} \quad \text{für } l_i \in I_1(\gamma) \quad (4.100)$$

verwendet. Damit erhält man für die Momente der Randkräfte

$$\mu_{T,l_j}^\gamma = \sum_{i=1}^4 \alpha_{l_i} \int_\gamma \phi_{l_j} \cdot \phi_{l_i} \quad \text{für } l_i \in I_1(\gamma). \quad (4.101)$$

Zur Berechnung der Koeffizienten α_{l_i} , $i = 1, \dots, 4$ wird die Massenmatrix M_γ für jede Kante γ definiert,

$$M_\gamma = \begin{bmatrix} \int_\gamma \phi_{l_1} \cdot \phi_{l_1} \, ds & \dots & \int_\gamma \phi_{l_1} \cdot \phi_{l_4} \, ds \\ \vdots & \dots & \vdots \\ \int_\gamma \phi_{l_4} \cdot \phi_{l_1} \, ds & \dots & \int_\gamma \phi_{l_4} \cdot \phi_{l_4} \, ds \end{bmatrix} = \frac{h}{6} \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 \end{bmatrix}, \quad (4.102)$$

und man erhält die Koeffizienten damit aus der Lösung eines linearen Gleichungssystems,

$$M_\gamma[\alpha_{l_1}, \alpha_{l_2}, \alpha_{l_3}, \alpha_{l_4}]^T = [\mu_{T,l_1}^\gamma, \mu_{T,l_2}^\gamma, \mu_{T,l_3}^\gamma, \mu_{T,l_4}^\gamma]^T, \quad (4.103)$$

$$\alpha_{l_j} = \frac{2}{h} (2\mu_{T,l_j}^\gamma - \mu_{T,l_{\text{mod}(j+1,3)+1}}^\gamma). \quad (4.104)$$

Definition 4.38. Es wird die Funktion $\Delta_{k-1+\theta,T}(\phi_l)$ für $l \in I_1(T)$ definiert durch

$$\Delta_{k-1+\theta,T}(\phi_l) := \int_T \boldsymbol{\sigma}_{h,k-1+\theta} : \boldsymbol{\epsilon}(\phi_l) \, dx - \int_T \mathbf{f}(t_{k-1+\theta}) \cdot \phi_l \, dx.$$

Die Äquilibrierungsbedingungen (4.96)-(4.98) lassen sich in Abhängigkeit der Momente ausdrücken. Für alle Elemente $T \in \mathcal{T}$ der Triangulierung muß gelten

$$\sum_{\gamma \subset \partial T} \mu_{T,l}^\gamma = \Delta_{k-1+\theta,T}(\phi_l) \quad \forall l \in I_1(T), \quad (4.105)$$

$$\mu_{T,l}^\gamma = \mu_{T',l}^\gamma \quad \forall l \in I_1(\gamma), \quad \gamma = \partial T \cap \partial T', \quad (4.106)$$

$$\mu_{T,l}^\gamma = \int_\gamma \mathbf{g}(t_{k-1+\theta}) \cdot \phi_l \, ds \quad \forall l \in I_1(\gamma), \quad \gamma = \partial T \cap \Gamma_N. \quad (4.107)$$

Der Patch zum Knoten n ist das Gebiet $\mathcal{P}_n = \cup\{T \in \mathcal{T} : n \in \mathcal{N}(T)\}$. Die Äquilibrierungsbedingungen für die Momente lassen sich als lokale Probleme zu jedem Patch \mathcal{P}_n , $n \in \mathcal{N}$, formulieren. Die Menge der Kanten, die Knoten n enthalten, ist definiert durch $\mathcal{E}_n = \{\gamma \in \partial\mathcal{T} : n \in \mathcal{N}(\gamma)\}$. Zu Knoten n gehören die Freiheitsgrade l_1 und l_2 mit $l_1 = 2n - 1$, $l_2 = 2n$. Je nachdem ob Knoten n ein innerer Knoten ist oder auf dem Rand liegt, ergeben sich Unterschiede für die Äquilibrierungsbedingungen. Bei Randknoten muß noch unterschieden werden, in welcher Kombination die an Knoten n angrenzenden Elementkanten zu Dirichlet-Rändern oder Neumann-Rändern gehören.

Bemerkung 3. Durch zweimalige partielle Integration in (4.79) erhält man

$$\begin{aligned} & - \int_T \operatorname{div} \mathbf{D}\mathbf{y}_T \cdot \mathbf{v} \, dx + \int_{\partial T} \mathbf{D}\mathbf{y}_T \cdot \mathbf{n} \, ds = \\ & \int_T (\mathbf{f}(t_{k-1+\theta}) + \operatorname{div} \boldsymbol{\sigma}_{h,k-1+\theta}) \cdot \mathbf{v} \, dx + \int_{\partial T} (\mathbf{g}_T - \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T) \cdot \mathbf{v} \, ds, \end{aligned} \quad (4.108)$$

wobei \mathbf{n}_T der Normalenvektor zum Element T ist. Wählt man nun für die Randkräfte $\mathbf{g}_T = \boldsymbol{\sigma}_{k-1+\theta} \cdot \mathbf{n}_T$, lautet das Problem zur Bestimmung von \mathbf{y}_T in der starken Formulierung

$$- \operatorname{div} \mathbf{D}\mathbf{y}_T = \mathbf{f}(t_{k-1+\theta}) + \operatorname{div} \boldsymbol{\sigma}_{h,k-1+\theta} \quad \text{auf } T, \quad (4.109)$$

$$\mathbf{D}\mathbf{y}_T \cdot \mathbf{n}_T = (\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}) \cdot \mathbf{n}_T \quad \text{auf } \partial T \cap \Gamma_N, \quad (4.110)$$

$$\mathbf{y}_T = \mathbf{0} \quad \text{auf } \partial T \cap \Gamma_D. \quad (4.111)$$

Das heißt $\mathbf{D}\mathbf{y}_T$ ist der Gradientenanteil von $\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}$ in einer Helmholtz-Zerlegung [21]. Eine optimale Wahl der Randkräfte wäre daher $\mathbf{g}_T = \boldsymbol{\sigma}_{k-1+\theta} \cdot \mathbf{n}_T$.

Numerische Lösung

Die lokalen Probleme 4.32 sind auf einem endlichdimensionalen Teilraum zu lösen.

Definition 4.39 (Endlichdimensionale lokale Probleme). Mit $V_{T,h}$ wird der Funktionenraum $V_{T,h} := \{\mathbf{v}_h \in H^1(T)^d \cap \mathcal{S}^q(T) : \mathbf{v}_h = \mathbf{0} \text{ auf } \Gamma_D \cap \partial T\}$ bezeichnet. Der Polynomgrad q dieses Finite-Elemente-Raumes ist höher zu wählen als der Polynomgrad p zur Lösung des Modellproblems. Mit (4.77) für das Funktional l und (4.78) für die Bilinearform a sind folgende endlichdimensionale Probleme zu lösen. Bestimme $\mathbf{y}_{T,h} \in V_{T,h}$, so daß für alle $\mathbf{v}_h \in V_{T,h}$

$$a_T(\mathbf{y}_{T,h}, \mathbf{v}_h) = l_{k-1+\theta,T}(\mathbf{v}_h). \quad (4.112)$$

Falls $|\partial T \cap \Gamma_D| = 0$, ist die Steifigkeitsmatrix \mathbf{A}_T des lokalen Problems singulär. In diesem Fall ist ein Gleichungssystem $\mathbf{A}_T \mathbf{x} = \mathbf{b}$ zu lösen, für das $\text{rg}(\mathbf{A}_T) = \text{rg}(\mathbf{A}_T | \mathbf{b})$ gilt, da aufgrund der Äquilibrierung eine nichteindeutige Lösung existiert. Der Vektor \mathbf{b} liegt also im Bildraum der Matrix \mathbf{A}_T . Eine Lösung wird mit Hilfe der QR-Zerlegung bestimmt. Es existiert eine Zerlegung

$$\mathbf{A}_T = \mathbf{Q}\mathbf{R} \quad (4.113)$$

der Matrix \mathbf{A} in ein Matrixprodukt aus einer unitären Matrix \mathbf{Q} und einer rechten oberen Dreiecksmatrix \mathbf{R} . Damit gilt

$$\mathbf{R}\mathbf{x} = \mathbf{Q}^T \mathbf{b} =: \mathbf{c}. \quad (4.114)$$

Die Diagonale von \mathbf{R} hat 3 Nulleinträge, entsprechend den 3 verschiedenen Starrkörperbewegungen im \mathbb{R}^2 . Die entsprechenden Indizes werden mit i_1, i_2, i_3 bezeichnet. Die Komponenten $x(i_1), x(i_2), x(i_3)$ von \mathbf{x} zu diesen Indizes sind beliebig und werden zu null gesetzt, $x(i_1) = x(i_2) = x(i_3) = 0$. Dann wird das Gleichungssystem

$$\tilde{\mathbf{R}}\tilde{\mathbf{x}} = \tilde{\mathbf{c}} \quad (4.115)$$

gelöst, das durch Streichen der Zeilen und Spalten (i_1, i_2, i_3) entsteht. Für $I = [1 : 2N(T)]$ und $\tilde{I} = I \setminus \{i_1, i_2, i_3\}$ gilt dann $\mathbf{x}(\tilde{I}) = \tilde{\mathbf{x}}$ und $\mathbf{x}((i_1, i_2, i_3)) = (0, 0, 0)$. Dieses Lösungsverfahren ist erheblich weniger aufwendig als etwa eine Lösung über eine Singulärwertzerlegung.

Definition 4.40 (Fehlerschätzer $\eta_{EQ,h}$ und Verfeinerungsindikator $\eta_{EQ,T,h}$). Die Lösung der endlichdimensionalen Probleme in Definition 4.39 definiert den Fehlerschätzer $\eta_{EQ,h}$ und den Verfeinerungsindikator $\eta_{EQ,T,h}$ durch

$$\eta_{EQ,T,h}^2 := a_T(\mathbf{y}_{T,h}, \mathbf{y}_{T,h}), \quad (4.116)$$

$$\eta_{EQ,h} := \left(\sum_{T \in \mathcal{T}} \eta_{EQ,T,h}^2 \right)^{1/2}. \quad (4.117)$$

4.3.3 Konstruktion der Momente der äquilibrierten Randkräfte

Es wird der Lösungsalgorithmus zur Bestimmung der Momente nach [4] angegeben. Eine optimale Wahl für die Randkräfte wäre $\mathbf{g}_T(t_{k-1+\theta}) = \boldsymbol{\sigma}_{k-1+\theta} \cdot \mathbf{n}$ und damit $\mu_{T,l}^\gamma = \int_\gamma \boldsymbol{\phi}_l \cdot (\boldsymbol{\sigma}_{k-1+\theta} \cdot \mathbf{n}_T) \mathbf{d}s$. Die exakten Spannungen sind nicht bekannt. Daher wählt man die Näherung

$$\mu_{T,l}^\gamma \approx \tilde{\mu}_{T,l}^\gamma = \int_\gamma \boldsymbol{\phi}_l \cdot (\boldsymbol{\sigma}_{h,k-1+\theta}|_T \cdot \mathbf{n}_T) \mathbf{d}s. \quad (4.118)$$

Es wird der Ausdruck

$$\frac{1}{2} \sum_{T \in \mathcal{P}_n} \sum_{\gamma \subset \partial T} \sum_{i=1}^2 (\mu_{T,n_i}^\gamma - \tilde{\mu}_{T,n_i}^\gamma)^2 \quad (4.119)$$

minimiert mit den Äquilibrierungsbedingungen (4.105)-(4.107) als Nebenbedingungen. Zu minimieren ist

$$\begin{aligned}
L(\{\mu_{T,n_i}^\gamma\}, \{\lambda_{\gamma,i}\}, \{\sigma_{T,i}\}) &= \frac{1}{2} \sum_{T \in \mathcal{P}_n} \sum_{\gamma \subset \partial T} \sum_{i=1}^2 (\mu_{T,n_i}^\gamma - \tilde{\mu}_{T,n_i}^\gamma)^2 \\
&+ \sum_{T \in \mathcal{P}_n} \sum_{i=1}^2 \sigma_{T,n_i} \left(\Delta_{k-1+\theta,T}(\phi_{n_i}) - \sum_{\gamma \subset \partial T} \mu_{T,n_i}^\gamma \right) + \sum_{\gamma = \partial T \cap \partial T'} \sum_{i=1}^2 \lambda_{\gamma,n_i} (\mu_{T,n_i}^\gamma + \mu_{T',n_i}^\gamma) \\
&+ \sum_{\gamma = \partial T \cap \Gamma_N} \sum_{i=1}^2 \lambda_{\gamma,n_i} \left(\mu_{T,n_i}^\gamma - \int_\gamma \mathbf{g}(t_{k-1+\theta}) \cdot \phi_{n_i} \, ds \right).
\end{aligned} \tag{4.120}$$

Eine notwendige Bedingung für ein Minimum lautet

$$\frac{\partial L(\{\mu_{T,n_i}^\gamma\}, \{\lambda_{\gamma,i}\}, \{\sigma_{T,i}\})}{\partial \mu_{T,n_i}^\gamma} = 0. \tag{4.121}$$

Daraus erhält man

$$\mu_{T,n_i}^\gamma - \tilde{\mu}_{T,n_i}^\gamma - \sigma_{T,n_i} + \lambda_{\gamma,n_i} = 0. \tag{4.122}$$

Man setzt

$$\lambda_{\gamma,n_i} = 0 \quad \text{auf } \gamma \subset \Gamma_D. \tag{4.123}$$

Es ergibt sich für die Lagrange-Multiplikatoren

$$\lambda_{\gamma,n_i} = \begin{cases} \frac{1}{2}(\sigma_{T,n_i} + \sigma_{T',n_i} + \tilde{\mu}_{T,n_i}^\gamma + \tilde{\mu}_{T',n_i}^\gamma) & \gamma = \partial T \cap \partial T', \\ \sigma_{T,n_i} + \tilde{\mu}_{T,n_i}^\gamma - \int_\gamma \mathbf{g} \cdot \phi_{n_i} \, ds & \gamma = \partial T \cap \Gamma_N, \\ 0 & \gamma = \partial T \cap \Gamma_D \end{cases} \tag{4.124}$$

und für die Momente der Randkräfte

$$\mu_{T,n_i}^\gamma = \begin{cases} \frac{1}{2}(\sigma_{T,n_i} - \sigma_{T',n_i} + \tilde{\mu}_{T,n_i}^\gamma - \tilde{\mu}_{T',n_i}^\gamma) & \gamma = \partial T \cap \partial T', \\ \int_\gamma \mathbf{g}(t_{k-1+\theta}) \cdot \phi_{n_i} \, ds & \gamma = \partial T \cap \Gamma_N, \\ \sigma_{T,n_i} + \tilde{\mu}_{T,n_i}^\gamma & \gamma = \partial T \cap \Gamma_D. \end{cases} \tag{4.125}$$

Damit erhält man durch Einsetzen in die Äquilibrierungsbedingung 1. Ordnung für die Momente ein lineares Gleichungssystem zur Bestimmung der Lagrange-Multiplikatoren $\{\sigma_{T,n_i} : T \in \mathcal{P}_n\}$:

$$\frac{1}{2} \sum_{\gamma = \partial T \cap \partial T'} (\sigma_{T,n_i} - \sigma_{T',n_i}) + \sum_{\gamma \subset \partial T \cap \Gamma_D} \sigma_{T,n_i} = \tilde{\Delta}_{k-1+\theta,T}(\phi_{n_i}) \quad \forall T \in \mathcal{P}_n \tag{4.126}$$

mit

$$\begin{aligned}
\tilde{\Delta}_{k-1+\theta,T}(\phi_{n_i}) &= \Delta_{k-1+\theta,T}(\phi_{n_i}) - \frac{1}{2} \sum_{\gamma = \partial T \cap \partial T'} (\tilde{\mu}_{T,n_i}^\gamma - \tilde{\mu}_{T',n_i}^\gamma) \\
&- \sum_{\gamma \subset \partial T \cap \Gamma_D} \tilde{\mu}_{T,n_i}^\gamma - \sum_{\gamma \subset \partial T \cap \Gamma_N} \int_\gamma \mathbf{g}(t_{k-1+\theta}) \cdot \phi_{n_i} \, ds.
\end{aligned} \tag{4.127}$$

Dieser Ausdruck kann auch in der folgenden Form dargestellt werden,

$$\tilde{\Delta}_{k-1+\theta,T}(\boldsymbol{\phi}_{n_i}) = \Delta_{k-1+\theta,T}(\boldsymbol{\phi}_{n_i}) - \int_{\partial T} \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle \cdot \boldsymbol{\phi}_{n_i} ds \quad (4.128)$$

mit

$$\langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle = \begin{cases} \frac{1}{2}(\boldsymbol{\sigma}_{h,k-1+\theta}|_T + \boldsymbol{\sigma}_{h,k-1+\theta}|_{T'}) \cdot \mathbf{n}_T & \text{auf } \partial T \cap \partial T', \\ \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T & \text{auf } \partial T \cap \Gamma_D, \\ \mathbf{g}(t_{k-1+\theta}) & \text{auf } \partial T \cap \Gamma_N. \end{cases} \quad (4.129)$$

Innerer Knoten:

Es wird das Gleichungssystem zur Bestimmung der Lagrange-Multiplikatoren $\{\sigma_{T,n_i} : T \in \mathcal{P}_n\}$ für ein Patch zu einem inneren Knoten aufgestellt.

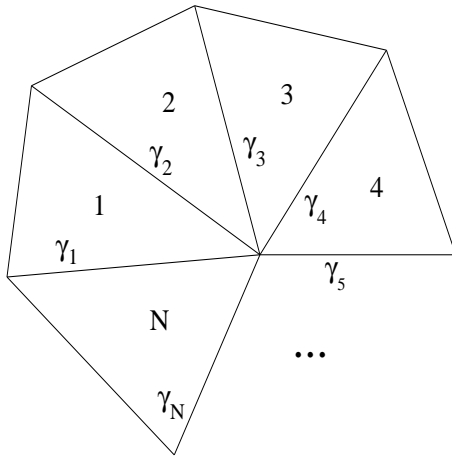


Abbildung 4.1: Patch zu einem inneren Knoten

Das Gleichungssystem (4.126) lautet:

$$\frac{1}{2} \begin{bmatrix} 2 & -1 & \dots & -1 \\ -1 & 2 & -1 & \dots & 0 \\ \vdots & & & & \\ 0 & \dots & -1 & 2 & -1 \\ -1 & \dots & & -1 & 2 \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_{N-1} \\ \sigma_N \end{bmatrix} = \begin{bmatrix} \tilde{\Delta}_1(\boldsymbol{\phi}_{n_i}) \\ \tilde{\Delta}_2(\boldsymbol{\phi}_{n_i}) \\ \vdots \\ \tilde{\Delta}_{N-1}(\boldsymbol{\phi}_{n_i}) \\ \tilde{\Delta}_N(\boldsymbol{\phi}_{n_i}) \end{bmatrix}. \quad (4.130)$$

Randknoten:

Es wird das Gleichungssystem zur Bestimmung der Lagrange-Multiplikatoren $\{\sigma_{T,n_i} : T \in \mathcal{P}_n\}$ für ein Patch zu einem Randknoten aufgestellt. Dabei sind drei Fälle zu unterscheiden. Der Knoten n gehört zu zwei Elementkanten, die von Typ Neumann-Neumann, Dirichlet-Neumann oder Dirichlet-Dirichlet sein können.

1. Neumann-Neumann: $\gamma_1 \subset \Gamma_N, \gamma_{N+1} \subset \Gamma_N$

$$\frac{1}{2} \begin{bmatrix} 1 & -1 & \dots & 0 \\ -1 & 2 & -1 & \dots & 0 \\ \vdots & & & & \\ 0 & \dots & -1 & 2 & -1 \\ 0 & \dots & & -1 & 1 \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_{N-1} \\ \sigma_N \end{bmatrix} = \begin{bmatrix} \tilde{\Delta}_1(\varphi_{n_i}) \\ \tilde{\Delta}_2(\varphi_{n_i}) \\ \vdots \\ \tilde{\Delta}_{N-1}(\varphi_{n_i}) \\ \tilde{\Delta}_N(\varphi_{n_i}) \end{bmatrix} \quad (4.131)$$

2. Neumann-Dirichlet: $\gamma_1 \subset \Gamma_N$, $\gamma_{N+1} \subset \Gamma_D$

$$\frac{1}{2} \begin{bmatrix} 3 & -1 & \dots & 0 \\ -1 & 2 & -1 & \dots & 0 \\ \vdots & & & & \\ 0 & \dots & -1 & 2 & -1 \\ 0 & \dots & & -1 & 1 \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_{N-1} \\ \sigma_N \end{bmatrix} = \begin{bmatrix} \tilde{\Delta}_1(\varphi_{n_i}) \\ \tilde{\Delta}_2(\varphi_{n_i}) \\ \vdots \\ \tilde{\Delta}_{N-1}(\varphi_{n_i}) \\ \tilde{\Delta}_N(\varphi_{n_i}) \end{bmatrix} \quad (4.132)$$

3. Dirichlet-Dirichlet: $\gamma_1 \subset \Gamma_D$, $\gamma_{N+1} \subset \Gamma_D$

$$\frac{1}{2} \begin{bmatrix} 3 & -1 & \dots & 0 \\ -1 & 2 & -1 & \dots & 0 \\ \vdots & & & & \\ 0 & \dots & -1 & 2 & -1 \\ 0 & \dots & & -1 & 3 \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_{N-1} \\ \sigma_N \end{bmatrix} = \begin{bmatrix} \tilde{\Delta}_1(\varphi_{n_i}) \\ \tilde{\Delta}_2(\varphi_{n_i}) \\ \vdots \\ \tilde{\Delta}_{N-1}(\varphi_{n_i}) \\ \tilde{\Delta}_N(\varphi_{n_i}) \end{bmatrix} \quad (4.133)$$

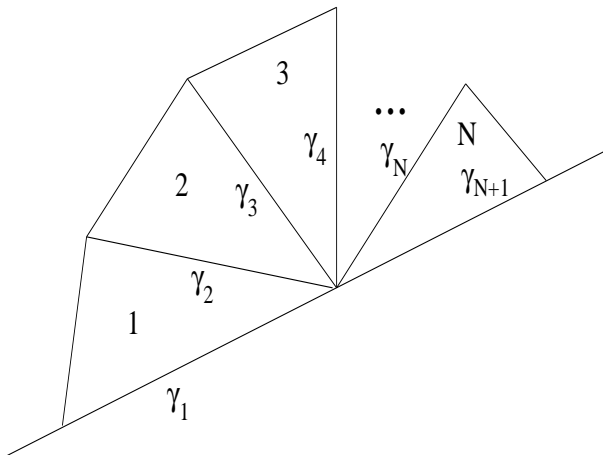


Abbildung 4.2: Randknoten

Es sind lineare Gleichungssysteme $M \cdot \sigma = b$, $M \in \mathbb{R}^{N \times N}$, $\sigma, b \in \mathbb{R}^N$ zu lösen. Die Matrix M ist für innere Knoten und Neumann-Neumann Randknoten singularär mit Kern $[1, 1, \dots, 1]^T$. Es gilt also $\dim(\text{Kern}(M)) = 1$ und damit $\text{rg}(M) = N - 1$. Das Gleichungssystem hat eine Lösung, falls $\text{rg}(M) = \text{rg}(M|b)$. Da $M = M^T$ folgt aus $\text{Kern}(M) = [1, 1, \dots, 1]^T$, daß das Gleichungssystem dann und nur dann eine Lösung besitzt, falls die Summe der Komponenten der rechten Seite b null ist, denn

nur dann können M und $M|b$ durch diesselben elementaren Zeilenumformungen in Zeilenstufenform gebracht werden. Es gilt

$$\sum_{T \in \mathcal{P}_n} \tilde{\Delta}_{k-1+\theta, T}(\phi_{n_i}) = \sum_{T \in \mathcal{P}_n} \Delta_{k-1+\theta, T}(\phi_{n_i}) - \int_{\partial \mathcal{P}_n \cap \Gamma_N} \mathbf{g}(t_{k-1+\theta}) \cdot \phi_{n_i} \, ds = 0 \quad (4.134)$$

aufgrund der Galerkin-Orthogonalität für P_1 -Finite-Elemente, und damit sind die linearen Gleichungssysteme lösbar. Zur Festlegung der Lösung der nicht eindeutig lösbareren Gleichungssysteme wird eine Methode aus [2] verwendet. Man minimiert $\sigma^T \sigma / 2$ unter der Nebenbedingung $M\sigma = b$. Das Variationsfunktional lautet

$$L(\sigma, \lambda) = \frac{1}{2} \sigma^T \sigma - (M\sigma - b)^T \lambda. \quad (4.135)$$

Durch Ableiten nach σ und λ erhält man

$$\sigma - M^T \lambda = 0 \quad \text{und} \quad M\sigma = b \quad (4.136)$$

und damit das Gleichungssystem

$$MM^T \lambda = b \quad (4.137)$$

zur Bestimmung von λ . Man definiert $T := MM^T$. T und b haben die Eigenschaft $\sum_{i=1}^N T_{ij} = 0$ und $\sum_{i=1}^N b_i = 0$. Man setzt $D := 1 \otimes 1$ und löst das Gleichungssystem

$$(T + D)\lambda = b. \quad (4.138)$$

Gleichungssysteme (4.137) und (4.138) haben diesselbe Lösung, denn aus (4.138) folgt

$$T\lambda + \left(\sum_{i=1}^N \lambda_i \right) \mathbf{1} = b, \quad (4.139)$$

und Aufsummieren der Vektorkomponenten auf beiden Seiten ergibt

$$\sum_{j=1}^N \lambda_j \sum_{i=1}^N T_{ij} + N \left(\sum_{i=1}^N \lambda_i \right) = 0. \quad (4.140)$$

Wegen $\sum_{i=1}^N T_{ij} = 0$ erhält man $\sum_{i=1}^N \lambda_i = 0$.

Durch die Äquilibrierungsprozedur wurde eine elementweise Galerkin-Orthogonalität für Ansatzfunktionen 1. Ordnung konstruiert. Besteht der FE-Ansatzraum aus Polynomen der Ordnung p , kann man auch Randkräfte $\mathbf{g}(t_{k-1+\theta})_T$ konstruieren, so daß

$$l_{k-1+\theta, T}(\phi_l) = 0 \quad \forall l \in I(T), \quad (4.141)$$

$$\mathbf{g}_{T_1}(t_{k-1+\theta}) + \mathbf{g}_{T_2}(t_{k-1+\theta}) = 0 \quad \text{auf } \partial T_1 \cap \partial T_2, \quad (4.142)$$

$$\mathbf{g}_T(t_{k-1+\theta}) = \mathbf{g}(t_{k-1+\theta}) \quad \text{auf } \partial T \cap \Gamma_N. \quad (4.143)$$

Man erhält eine elementweise Galerkin-Orthogonalität der Ordnung p . Es wird eine Äquilibration höherer Ordnung durchgeführt. Für die Randkräfte $\mathbf{g}(t_{k-1+\theta})_T$ macht man den Ansatz

$$\mathbf{g}(t_{k-1+\theta})_T|_\gamma = \sum_{l \in I_1(\gamma) \cup I_2(\gamma)} \alpha_l \boldsymbol{\phi}_l \quad (4.144)$$

und erhält zur Bestimmung der Koeffizienten α_l ein lineares Gleichungssystem,

$$\sum_{k \in I(\gamma)} \left(\int_\gamma \boldsymbol{\phi}_j \cdot \boldsymbol{\phi}_k \, d\mathbf{s} \right) \alpha_k = \mu_{T,j}^\gamma \quad \forall j \in I_1(\gamma) \cup I_2(\gamma). \quad (4.145)$$

Es zeigt sich, daß die Momente bezüglich der linearen Basisfunktionen unabhängig durch den bereits beschriebenen Algorithmus zur Äquilibration 1. Ordnung berechnet werden können. Es verbleiben die Momente bezüglich der Kantenbasisfunktionen zu bestimmen. Für $l \in I_2(\gamma)$ sind die folgenden Äquilibrationsbedingungen zu erfüllen,

$$\mu_{T,l}^\gamma = \Delta_{k-1+\theta,T}(\boldsymbol{\phi}_l) \quad \forall T \text{ mit } \gamma \subset \partial T, \quad (4.146)$$

$$\mu_{T_1,l}^\gamma + \mu_{T_2,l}^\gamma = 0 \quad \gamma = \partial T_1 \cap \partial T_2, \quad (4.147)$$

$$\mu_{T,l}^\gamma = \int_\gamma \mathbf{g}(t_{k-1+\theta}) \cdot \boldsymbol{\phi}_l \, d\mathbf{s} \quad \gamma = \partial T \cap \Gamma_N. \quad (4.148)$$

Innere Kante:

Falls $\gamma = \partial T_1 \cap \partial T_2$ lauten die Äquilibrationsbedingungen (4.146)-(4.147)

$$\mu_{T_1,l}^\gamma = \Delta_{k-1+\theta,T_1}(\boldsymbol{\phi}_l), \quad (4.149)$$

$$\mu_{T_2,l}^\gamma = \Delta_{k-1+\theta,T_2}(\boldsymbol{\phi}_l), \quad (4.150)$$

$$\mu_{T_1,l}^\gamma + \mu_{T_2,l}^\gamma = 0. \quad (4.151)$$

Die ersten beiden Bedingungen bestimmen die Momente eindeutig und die dritte Bedingung gilt wegen der Galerkin-Orthogonalität.

Neumann-Kante:

Falls $\gamma = \partial T \cap \Gamma_N$ lauten die Äquilibrationsbedingungen (4.146)-(4.147),

$$\mu_{T,l}^\gamma = \Delta_{k-1+\theta,T}(\boldsymbol{\phi}_l), \quad (4.152)$$

$$\mu_{T,l}^\gamma = \int_\gamma \mathbf{g}(t_{k-1+\theta}) \cdot \boldsymbol{\phi}_l \, d\mathbf{s}. \quad (4.153)$$

Diese Bedingungen bestimmen die Momente eindeutig, da die Differenz der rechten Seiten aufgrund der Galerkin-Orthogonalität verschwindet.

Dirichlet-Kante:

Für $\gamma \subset \Gamma_D$, sind die Momente eindeutig durch die folgende Bedingung bestimmt,

$$\mu_{T,l}^\gamma = \Delta_{k-1+\theta,T}(\boldsymbol{\phi}_l). \quad (4.154)$$

4.3.4 Effizienz der Fehlerabschätzung

Die Effizienz der Methode der äquilibrierten Residuen ist in [4] für die Finite-Elemente-Lösung des Poisson-Problems mit P_1 -Elementen bewiesen. Im Zusammenhang der Lösung des elastoviskoplastischen Problems mit einer Finite-Elemente-Methode höherer Ordnung ergeben sich zusätzliche Schwierigkeiten. In Effizienzbeweisen treten üblicherweise approximierende Terme auf, die unter bestimmten Voraussetzungen von höherer Ordnung sind. Bei der Methode der äquilibrierten Residuen tritt dabei die Orthogonalprojektion $\Pi_p^\gamma : L_2(\gamma) \rightarrow \mathcal{P}_p(\gamma)$ auf. Um den approximierenden Term in Satz 4.44 möglichst klein zu halten, wird ausgenutzt, daß p in der Orthogonalprojektion als der Polynomgrad der Finite-Elemente-Methode gewählt werden kann. Für Finite-Elemente-Methoden höherer Ordnung ist damit $p > 1$ und es sind im Vergleich zu [4] Modifikationen nötig.

Die Menge der Kanten $E \in \mathcal{E}$, die den Knoten $n \in \mathcal{N}$ enthalten, ist definiert als

$$\mathcal{E}_n := \{E \in \mathcal{E} : n \subset E\}. \quad (4.155)$$

Satz 4.41. *Es wird vorausgesetzt, daß die Randkräfte \mathbf{g}_T nach einer Äquilibrierung p -ter Ordnung bestimmt sind. Mit $\Pi_p^\gamma : L_2(\gamma) \rightarrow \mathcal{P}_p(\gamma)$ ist eine Orthogonalprojektion bezeichnet. Es gilt*

$$\|\Pi_p^\gamma(\mathbf{g}_T - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle)\|_{L^2(\gamma)}^2 \leq c_{12} \left(\sum_{n \in \mathcal{N} \cap \gamma} \left(\sum_{T' \subset \Omega_n} h_{T'} \|\mathbf{r}\|_{L^2(T')}^2 + \sum_{\gamma' \subset \partial T' \cap \mathcal{E}_n} \|\mathbf{R}\|_{L^2(\gamma')}^2 \right) \right). \quad (4.156)$$

Beweis. Es wird das Element $T \in \mathcal{T}$ betrachtet. Aufgrund der κ -Regularität der Triangulierung und (3.52) ist der Durchmesser von jedem Element in dem Patch \tilde{T} äquivalent zu h_T [4]. Daher kann während des Beweises h für jeden auftretenden Elementdurchmesser geschrieben werden.

Die Momente der Größe $\Pi_p^\gamma(\mathbf{g}_T - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle) \in \mathcal{P}_p(\gamma)$ bezüglich Basisfunktionen ϕ_l sind

$$\mu_{T,l}^{*\gamma} = \int_\gamma \Pi_p^\gamma(\mathbf{g}_T - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle) \cdot \phi_l ds.$$

Die Größe $\Pi_p^\gamma(\mathbf{g}_T - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle)$ kann in eine Legendre-Reihe entwickelt werden. Die Orthogonalitätseigenschaft der Legendre-Polynome zeigt dann

$$\int_\gamma \Pi_p^\gamma(\mathbf{g}_T - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle) \cdot \phi_l ds = \int_\gamma (\mathbf{g}_T - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle) \cdot \phi_l ds,$$

wobei ϕ_l Kantenbasisfunktionen bis zum Grad p sind. Die Größe $\Pi_p^\gamma(\mathbf{g}_T - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle)$ wird als Linearkombination der Kantenbasisfunktionen dargestellt,

$$\Pi_p^\gamma(\mathbf{g}_T - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle) = \sum_{l \in I(\gamma)} \beta_l \phi_l.$$

Die Koeffizienten $\boldsymbol{\beta} = \{\beta_1, \dots, \beta_N\}$ bestimmen sich aus dem linearen Gleichungssystem

$$\mathbf{M}_\gamma \boldsymbol{\beta} = \boldsymbol{\mu}_T^{*\gamma}$$

zu

$$\boldsymbol{\beta} = \mathbf{M}_\gamma^{-1} \boldsymbol{\mu}_T^{*\gamma}. \quad (4.157)$$

Hierbei ist \mathbf{M}_γ die Massenmatrix zur Kante γ und der Vektor $\boldsymbol{\mu}_T^{*\gamma}$ hat die Momente $\mu_{T,i}^{*\gamma}$ als Komponenten. Damit lässt sich wie folgt abschätzen:

$$\begin{aligned} \|\Pi_p^\gamma(\mathbf{g}_T - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle)\|_{L^2(\gamma)} &= \left\| \sum_{l \in I(\gamma)} \beta_l \boldsymbol{\phi}_l \right\|_{L^2(\gamma)} \\ &= \left(\int_\gamma \left| \sum_{l \in I(\gamma)} \beta_l \boldsymbol{\phi}_l \right|^2 ds \right)^{1/2} \\ &\leq \left(\int_\gamma \left(\left(\sum_{l \in I(\gamma)} \beta_l^2 \right)^{1/2} \left(\sum_{l \in I(\gamma)} \boldsymbol{\phi}_l \cdot \boldsymbol{\phi}_l \right)^{1/2} \right)^2 ds \right)^{1/2} = \left(\int_\gamma \left(\sum_{l \in I(\gamma)} \beta_l^2 \right) \left(\sum_{l \in I(\gamma)} \boldsymbol{\phi}_l \cdot \boldsymbol{\phi}_l \right) ds \right)^{1/2} \\ &= |\boldsymbol{\beta}| \left(\int_\gamma \sum_{l \in I(\gamma)} \boldsymbol{\phi}_l \cdot \boldsymbol{\phi}_l ds \right)^{1/2} \leq c_{13} h^{1/2} |\boldsymbol{\beta}| \leq c_{13} h^{1/2} \underbrace{|\mathbf{M}_\gamma^{-1}|}_{\leq c_{14} h^{-1}} |\boldsymbol{\mu}_T^{*\gamma}| \\ &\leq c_{13} c_{14} h^{-1/2} |\boldsymbol{\mu}_T^{*\gamma}|. \end{aligned}$$

Damit gilt also

$$\|\Pi_p^\gamma(\mathbf{g}_T - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle)\|_{L^2(\gamma)}^2 \leq c_{13}^2 c_{14}^2 h^{-1} |\boldsymbol{\mu}_T^{*\gamma}|^2 = c_{13}^2 c_{14}^2 h^{-1} \sum_{i \in I(\gamma)} (\mu_{T,i}^{*\gamma})^2.$$

Zunächst werden die Momente zu Basisfunktionen 1. Ordnung abgeschätzt, wobei hier die Ausführungen [4] folgen. Es gilt

$$\mu_{T,i}^{*\gamma} = \int_\gamma (\mathbf{g}_T - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle) \cdot \boldsymbol{\phi}_i ds.$$

Mit (4.129) erhält man

$$\int_\gamma \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle \cdot \boldsymbol{\phi}_{n_i} ds = \begin{cases} \frac{1}{2}(\tilde{\mu}_{T,n_i}^\gamma - \tilde{\mu}_{T',n_i}^\gamma) & \text{auf } \gamma = \partial T \cap \partial T', \\ \tilde{\mu}_{T,n_i}^\gamma & \text{auf } \gamma = \partial T \cap \Gamma_D, \\ \int_\gamma \mathbf{g}(t_{k-1+\theta}) \cdot \boldsymbol{\phi}_{n_i} ds & \text{auf } \gamma \cap \partial T \cap \Gamma_N. \end{cases}$$

Mit (4.125) folgt

$$\mu_{T,n_i}^{*\gamma} = \begin{cases} \frac{1}{2}(\sigma_{T,n_i} - \sigma_{T',n_i}) & \text{auf } \gamma = \partial T \cap \partial T', \\ \sigma_{T,n_i} & \text{auf } \gamma = \partial T \cap \Gamma_D, \\ 0 & \text{auf } \gamma = \partial T \cap \Gamma_N. \end{cases}$$

Es folgt

$$|\mu_{T,n_i}^{*\gamma}|^2 \leq c_{15} \sum_{T' \in \Omega_n} \sigma_{T',n_i}^2 \leq c_{15} \sum_{T' \in \Omega_n} \tilde{\Delta}_{k-1+\theta,T'}(\boldsymbol{\phi}_{n_i})^2 \quad (4.158)$$

für die Momente zu linearen Basisfunktionen. Weiter gilt

$$\begin{aligned}
& \tilde{\Delta}_{k-1+\theta, T'}(\phi_{n_i}) \\
&= \int_{T'} \boldsymbol{\sigma}_{h, k-1+\theta} : \boldsymbol{\epsilon}(\phi_{n_i}) \mathbf{d}\mathbf{x} - \int_{T'} \mathbf{f}(t_{k-1+\theta}) \cdot \phi_{n_i} \mathbf{d}\mathbf{x} - \int_{\partial T'} \langle \boldsymbol{\sigma}_{h, k-1+\theta} \cdot \mathbf{n}_{T'} \rangle \cdot \phi_{n_i} \mathbf{d}\mathbf{s} \\
&= - \int_{T'} (\operatorname{div} \boldsymbol{\sigma}_{h, k-1+\theta} + \mathbf{f}(t_{k-1+\theta})) \cdot \phi_{n_i} \mathbf{d}\mathbf{x} + \int_{\partial T'} (\boldsymbol{\sigma}_{h, k-1+\theta} \cdot \mathbf{n}_{T'}) \cdot \phi_{n_i} \mathbf{d}\mathbf{s} \\
&\quad \int_{\partial T'} \langle \boldsymbol{\sigma}_{h, k-1+\theta} \cdot \mathbf{n}_{T'} \rangle \cdot \phi_{n_i} \mathbf{d}\mathbf{s} \\
&= - \int_{T'} \mathbf{r} \cdot \phi_{n_i} \mathbf{d}\mathbf{x} + \int_{\partial T'} \tilde{\mathbf{R}} \cdot \phi_{n_i} \mathbf{d}\mathbf{s}.
\end{aligned}$$

Auf einer Kante $\gamma = \partial T' \cap \Gamma_N$ oder $\gamma = \partial T' \cap \Gamma_D$ gilt dabei $\tilde{\mathbf{R}} = \mathbf{R}$ und auf inneren Kanten γ gilt $\tilde{\mathbf{R}} = \frac{1}{2} \mathbf{R}$. Damit erhält man

$$\begin{aligned}
|\tilde{\Delta}_{k-1+\theta, T'}(\phi_{n_i})| &\leq \|\mathbf{r}\|_{L^2(T')} \|\phi_{n_i}\|_{L^2(T')} + \sum_{\gamma' \subset \partial T'} \|\mathbf{R}\|_{L^2(\gamma')} \|\phi_{n_i}\|_{L^2(\gamma')} \\
&\leq c_{16} (h \|\mathbf{r}\|_{L^2(T')} + \sum_{\gamma' \subset \partial T' \cap \mathcal{E}_n} h^{1/2} \|\mathbf{R}\|_{L^2(\gamma)}).
\end{aligned}$$

Hierbei ist \mathcal{E}_n die Menge der Kanten, die Knoten n enthalten. Damit gilt

$$(\mu_{T, n_i}^{*\gamma})^2 \leq c_{17} \sum_{T' \in \Omega_n} (h^2 \|\mathbf{r}\|_{L^2(T')}^2 + \sum_{\gamma' \subset \partial T' \cap \mathcal{E}_n} h \|\mathbf{R}\|_{L^2(\gamma')}^2).$$

Es sollen nun die Momente $\mu_{T, l}^{*\gamma}$ zu Basisfunktionen höherer Ordnung abgeschätzt werden. Es gilt

$$\mu_{T, l}^{*\gamma} = \int_{\gamma} (\mathbf{g}_T - \langle \boldsymbol{\sigma}_{h, k-1+\theta} \cdot \mathbf{n}_T \rangle) \cdot \phi_l \mathbf{d}\mathbf{s} = \mu_{T, l}^{\gamma} - \int_{\gamma} \langle \boldsymbol{\sigma}_{h, k-1+\theta} \cdot \mathbf{n}_T \rangle \cdot \phi_l \mathbf{d}\mathbf{s}.$$

Für $\gamma \subset \Gamma_N$ gilt

$$\mu_{T, l}^{\gamma} = \int_{\gamma} \mathbf{g}(t_{k-1+\theta}) \cdot \phi_l \mathbf{d}\mathbf{s} \quad \text{und} \quad \langle \boldsymbol{\sigma}_{h, k-1+\theta} \cdot \mathbf{n}_T \rangle = \mathbf{g}(t_{k-1+\theta}).$$

Daher ist $\mu_{T, l}^{*\gamma} = 0$. Für $\gamma \subset \Gamma_D$ gilt mit (4.154)

$$\mu_{T, l}^{\gamma} = \Delta_{k-1+\theta, T}(\phi_l) \quad \text{und} \quad \langle \boldsymbol{\sigma}_{h, k-1+\theta} \cdot \mathbf{n}_T \rangle = \boldsymbol{\sigma}_{h, k-1+\theta} \cdot \mathbf{n}_T.$$

Damit gilt

$$\begin{aligned}
\mu_{T, l}^{*\gamma} &= \Delta_{k-1+\theta, T}(\phi_l) - \int_{\gamma} (\boldsymbol{\sigma}_{h, k-1+\theta} \cdot \mathbf{n}_T) \cdot \phi_l \mathbf{d}\mathbf{s} \\
&= \int_T \boldsymbol{\sigma}_{h, k-1+\theta} : \boldsymbol{\epsilon}(\phi_l) \mathbf{d}\mathbf{x} - \int_T \mathbf{f} \cdot \phi_l \mathbf{d}\mathbf{x} - \int_{\gamma} (\boldsymbol{\sigma}_{h, k-1+\theta} \cdot \mathbf{n}_T) \cdot \phi_l \mathbf{d}\mathbf{s}.
\end{aligned}$$

Partielle Integration zeigt

$$\begin{aligned} \mu_{T,l}^{*\gamma} &= - \int_T (\operatorname{div} \boldsymbol{\sigma}_{h,k-1+\theta} + \mathbf{f}(t_{k-1+\theta})) \cdot \boldsymbol{\phi}_l \, d\mathbf{x} + \int_{\partial T} (\boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T) \cdot \boldsymbol{\phi}_l \, ds \\ &\quad - \int_{\gamma} (\boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T) \cdot \boldsymbol{\phi}_l \, ds \\ &= - \int_T (\operatorname{div} \boldsymbol{\sigma}_{h,k-1+\theta} + \mathbf{f}(t_{k-1+\theta})) \cdot \boldsymbol{\phi}_l \, d\mathbf{x} = - \int_T \mathbf{r} \cdot \boldsymbol{\phi}_l \, d\mathbf{x}. \end{aligned}$$

Falls $\gamma = \partial T \cap \partial T'$, gilt

$$\mu_{T,l}^{\gamma} = \Delta_{k-1+\theta,T}(\boldsymbol{\phi}_l) \quad \text{und} \quad \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle = \frac{1}{2} (\boldsymbol{\sigma}_{h,k-1+\theta}|_T + \boldsymbol{\sigma}_{h,k-1+\theta}|_{T'}) \cdot \mathbf{n}_T.$$

Damit folgt

$$\begin{aligned} \mu_{T,l}^{*\gamma} &= \Delta_{k-1+\theta,T}(\boldsymbol{\phi}_l) - \int_{\gamma} \left(\frac{1}{2} (\boldsymbol{\sigma}_{h,k-1+\theta}|_T + \boldsymbol{\sigma}_{h,k-1+\theta}|_{T'}) \cdot \mathbf{n}_T \right) \cdot \boldsymbol{\phi}_l \, ds = \\ &= \int_T \boldsymbol{\sigma}_{h,k-1+\theta} : \boldsymbol{\epsilon}(\boldsymbol{\phi}_l) \, d\mathbf{x} - \int_T \mathbf{f} \cdot \boldsymbol{\phi}_l \, d\mathbf{x} - \int_{\gamma} \left(\frac{1}{2} (\boldsymbol{\sigma}_{h,k-1+\theta}|_T + \boldsymbol{\sigma}_{h,k-1+\theta}|_{T'}) \cdot \mathbf{n}_T \right) \cdot \boldsymbol{\phi}_l \, ds. \end{aligned}$$

Partielle Integration zeigt

$$\begin{aligned} \mu_{T,l}^{*\gamma} &= - \int_T (\operatorname{div} \boldsymbol{\sigma}_{h,k-1+\theta} + \mathbf{f}(t_{k-1+\theta})) \cdot \boldsymbol{\phi}_l \, d\mathbf{x} + \int_{\partial T} (\boldsymbol{\sigma}_{h,k-1+\theta}|_T \cdot \mathbf{n}_T) \cdot \boldsymbol{\phi}_l \, ds \\ &\quad - \int_{\gamma} \left(\frac{1}{2} (\boldsymbol{\sigma}_{h,k-1+\theta}|_T + \boldsymbol{\sigma}_{h,k-1+\theta}|_{T'}) \cdot \mathbf{n}_T \right) \cdot \boldsymbol{\phi}_l \, ds \\ &= - \int_T (\operatorname{div} \boldsymbol{\sigma}_{h,k-1+\theta} + \mathbf{f}(t_{k-1+\theta})) \cdot \boldsymbol{\phi}_l \, d\mathbf{x} \\ &\quad + \frac{1}{2} \int_{\gamma} ((\boldsymbol{\sigma}_{h,k-1+\theta}|_T - \boldsymbol{\sigma}_{h,k-1+\theta}|_{T'}) \cdot \mathbf{n}_T) \cdot \boldsymbol{\phi}_l \, ds \\ &= - \int_T \mathbf{r} \cdot \boldsymbol{\phi}_l \, d\mathbf{x} + \frac{1}{2} \int_{\gamma} \mathbf{R} \cdot \boldsymbol{\phi}_l \, ds. \end{aligned}$$

Damit gilt für Momente zu Kantenbasisfunktionen

$$|\mu_{T,l}^{*\gamma}| \leq \|\mathbf{r}\|_{L^2(T)} \|\boldsymbol{\phi}_l\|_{L^2(T)} + \|\mathbf{R}\|_{L^2(\gamma)} \|\boldsymbol{\phi}_l\|_{L^2(\gamma)} \leq c_{18} (h \|\mathbf{r}\|_{L^2(T)} + h^{1/2} \|\mathbf{R}\|_{L^2(\gamma)}),$$

und man erhält

$$(\mu_{T,l}^{*\gamma})^2 \leq c_{19} (h^2 \|\mathbf{r}\|_{L^2(T)}^2 + h \|\mathbf{R}\|_{L^2(\gamma)}^2). \quad \blacksquare$$

Satz 4.42 (Variante der Kornschen Ungleichung). *Ist Ω eine offene, beschränkte Menge in \mathbb{R}^2 mit stückweise glattem Rand, dann gibt es eine positive Konstante c_{20} , so daß*

$$c_{20} \|\mathbf{v}\|_{H^1(\Omega)}^2 \leq \int_{\Omega} \boldsymbol{\epsilon}(\mathbf{v}) : \boldsymbol{\epsilon}(\mathbf{v}) \, d\mathbf{x} \quad \text{für alle } \mathbf{v} \in H^1(\Omega)^2 \setminus Z_{\Omega}, \quad (4.159)$$

wobei Z_{Ω} die Menge aller Starrkörperbewegungen auf Ω ist.

Beweis. Angenommen, die Ungleichung sei falsch. Dann gibt es eine Folge $\mathbf{v}_n \in H^1(\Omega)^2 \setminus Z_\Omega$ mit

$$\|\boldsymbol{\epsilon}(\mathbf{v}_n)\|_{L^2(\Omega)}^2 \leq \frac{1}{n} \quad \text{und} \quad \|\mathbf{v}_n\|_{H^1(\Omega)} = 1.$$

Weil $H^1(\Omega)$ in $H^0(\Omega)$ kompakt ist, gibt es eine Teilfolge der beschränkten Folge \mathbf{v}_n , die in $H^0(\Omega) = L^2(\Omega)$ konvergiert. Aufgrund der 1. Kornschen Ungleichung gilt,

$$\begin{aligned} c_K \|\mathbf{v}_{n_i} - \mathbf{v}_{n_j}\|_{H^1(\Omega)}^2 &\leq \|\boldsymbol{\epsilon}(\mathbf{v}_{n_i} - \mathbf{v}_{n_j})\|_{L^2(\Omega)}^2 + \|\mathbf{v}_{n_i} - \mathbf{v}_{n_j}\|_{L^2(\Omega)}^2 \\ &\leq \frac{2}{n_i} + \frac{2}{n_j} + \|\mathbf{v}_{n_i} - \mathbf{v}_{n_j}\|_{L^2(\Omega)}^2. \end{aligned}$$

Damit ist die Teilfolge von \mathbf{v}_n Cauchy-Folge in $H^1(\Omega)$ und da $H^1(\Omega)$ ein Banachraum ist, auch konvergent in $H^1(\Omega)$. Es gibt daher ein $\mathbf{u}_0 \in H^1(\Omega)^2 \setminus Z_\Omega$, so daß

$$\lim_{n_i \rightarrow \infty} \|\boldsymbol{\epsilon}(\mathbf{v}_{n_i})\|_{H^1(\Omega)} = \|\boldsymbol{\epsilon}(\mathbf{u}_0)\|_{H^1(\Omega)} = 0.$$

Daher ist $\mathbf{u}_0 = 0$, ein Widerspruch zu $1 = \|\mathbf{v}_{n_i}\|_{H^1(\Omega)} \rightarrow \|\mathbf{u}_0\|_{H^1(\Omega)}$. ■

Satz 4.43 ([20]). *Unter der Voraussetzung, daß das Gebiet Ω einen Lipschitz-Rand hat und $1 \leq p \leq \infty$ eine reelle Zahl ist, existiert eine Konstante $c_{21} > 0$, so daß für alle $\mathbf{v} \in W^{1,p}(\Omega)$*

$$\|\mathbf{v}\|_{L^p(\partial\Omega)} \leq c_{21} \|\mathbf{v}\|_{L^p(\Omega)}^{1-1/p} \|\mathbf{v}\|_{W^{1,p}(\Omega)}^{1/p}. \quad (4.160)$$

Der folgende Satz zeigt, wie der Effizienzbeweis für die Methode der äquilibrierten Residuen in [4] zu modifizieren ist, um in der Elastoviskoplastizität eine obere Abschätzung der elementweisen Fehlerindikatoren zu erhalten.

Satz 4.44. *Ist \mathbf{y}_T eine Lösung des lokalen Problems aus Definition 4.32, und wurden die äquilibrierten Randkräfte $\{\mathbf{g}_T\}$ nach einer Äquilibrierung p -ter Ordnung bestimmt, dann gilt mit $\bar{\mathbf{r}}$ und $\bar{\mathbf{R}}$ aus den Sätzen 4.29 und 4.30 für den Fehler $\mathbf{e} := \mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta}$ mit einer Konstanten $c_{24} > 0$ die Abschätzung*

$$\begin{aligned} a_T(\mathbf{y}_T, \mathbf{y}_T)^{1/2} &\leq c_{24} \left(\|\mathbf{e}\|_{H^1(\tilde{T})} + h_T \|\mathbf{r} - \bar{\mathbf{r}}\|_{L^2(\tilde{T})} + h_T^{1/2} \|\mathbf{R} - \bar{\mathbf{R}}\|_{L^2(\cup\{\mathcal{E}_n: n \in \mathcal{N}, n \subset T\})} \right. \\ &\quad \left. + h_T^{1/2} \|\Pi_p^\gamma \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle\|_{L^2(\partial T \setminus \Gamma_N)} \right). \end{aligned} \quad (4.161)$$

Beweis. Die lokalen Probleme sind definiert durch

$$\begin{aligned} &a_T(\mathbf{y}_T, \mathbf{v}) \\ &= l_{k-1+\theta}(\mathbf{v}) \\ &= \int_T \mathbf{f}(t_{k-1+\theta}) \cdot \mathbf{v} \, d\mathbf{x} + \int_{\partial T} \mathbf{g}_T(t_{k-1+\theta}) \cdot \mathbf{v} \, d\mathbf{s} - \int_T \boldsymbol{\sigma}_{h,k-1+\theta} : \boldsymbol{\epsilon}(\mathbf{v}) \, d\mathbf{x} \\ &= \int_T (\mathbf{f}(t_{k-1+\theta}) + \operatorname{div} \boldsymbol{\sigma}_{h,k-1+\theta}) \cdot \mathbf{v} \, d\mathbf{x} + \int_{\partial T} (\mathbf{g}_T(t_{k-1+\theta}) - \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T) \cdot \mathbf{v} \, d\mathbf{s} \\ &= \int_T \mathbf{r} \cdot \mathbf{v} \, d\mathbf{x} + \int_{\partial T} c_\gamma \mathbf{R} \cdot \mathbf{v} \, d\mathbf{s} + \int_{\partial T} (\mathbf{g}_T(t_{k-1+\theta}) - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle) \cdot \mathbf{v} \, d\mathbf{s}, \end{aligned}$$

wobei $c_\gamma = 1/2$, falls γ eine innere Kante ist, und $c_\gamma = 1$, falls γ eine Kante auf dem Rand $\partial\Omega$ ist. Aufgrund der Äquilibrierung 1. Ordnung gilt $l_{k-1+\theta}(1) = 0$, und daher erhält man mit $\bar{\mathbf{v}} = \frac{1}{T} \int_T \mathbf{v} d\mathbf{x}$,

$$\begin{aligned} a_T(\mathbf{y}_T, \mathbf{v}) &= a(\mathbf{y}_T, \mathbf{v} - \bar{\mathbf{v}}) = \int_T \mathbf{r} \cdot (\mathbf{v} - \bar{\mathbf{v}}) d\mathbf{x} + \int_{\partial T} c_\gamma \mathbf{R} \cdot (\mathbf{v} - \bar{\mathbf{v}}) ds \\ &\quad + \int_{\partial T} (\mathbf{g}_T(t_{k-1+\theta}) - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle) \cdot (\mathbf{v} - \bar{\mathbf{v}}) ds. \end{aligned}$$

Mit der Cauchy-Ungleichung und Satz 4.43 folgt

$$\begin{aligned} &a_T(\mathbf{y}_T, \mathbf{v}) \\ &\leq \|\mathbf{r}\|_{L^2(T)} \|\mathbf{v} - \bar{\mathbf{v}}\|_{L^2(T)} + \|\mathbf{R}\|_{L^2(\partial T)} \|\mathbf{v} - \bar{\mathbf{v}}\|_{L^2(\partial T)} \\ &\quad + \|\mathbf{g}_T(t_{k-1+\theta}) - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle\|_{L^2(\partial T)} \|\mathbf{v} - \bar{\mathbf{v}}\|_{L^2(\partial T)} \\ &\leq \|\mathbf{r}\|_{L^2(T)} \|\mathbf{v} - \bar{\mathbf{v}}\|_{L^2(T)} + c_{21} \|\mathbf{R}\|_{L^2(\partial T)} \|\mathbf{v} - \bar{\mathbf{v}}\|_{L^2(T)}^{1/2} \|\mathbf{v} - \bar{\mathbf{v}}\|_{H^1(T)}^{1/2} \\ &\quad + c_{21} \|\mathbf{g}_T(t_{k-1+\theta}) - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle\|_{L^2(\partial T)} \|\mathbf{v} - \bar{\mathbf{v}}\|_{L^2(T)}^{1/2} \|\mathbf{v} - \bar{\mathbf{v}}\|_{H^1(T)}^{1/2}. \end{aligned}$$

Mit der Poincaré-Ungleichung $\|\mathbf{v} - \bar{\mathbf{v}}\|_{L^2(T)} \leq c_{22} h |\mathbf{v}|_{H^1(T)}$ folgt

$$\begin{aligned} a_T(\mathbf{y}_T, \mathbf{v}) &\leq c_{22} (h \|\mathbf{r}\|_{L^2(T)} + c_{21} h^{1/2} \|\mathbf{R}\|_{L^2(\partial T)} + c_{21} h^{1/2} \|\mathbf{g}_T - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle\|_{L^2(\partial T)}) \\ &\quad \times \|\mathbf{v}\|_{H^1(T)}. \end{aligned}$$

Mit Satz 4.42 und der Wahl $\mathbf{v} = \mathbf{y}_T$ erhält man mit $c_{23} = c_{22} \max\{1, c_{21}\}$

$$\begin{aligned} &a_T(\mathbf{y}_T, \mathbf{y}_T)^{1/2} \\ &\leq c_{23} (h \|\mathbf{r}\|_{L^2(T)} + h^{1/2} \|\mathbf{R}\|_{L^2(\partial T)} + h^{1/2} \|\mathbf{g}_T - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle\|_{L^2(\partial T)}) \\ &= c_{23} (h \|\mathbf{r}\|_{L^2(T)} + h^{1/2} \|\mathbf{R}\|_{L^2(\partial T)} \\ &\quad + h^{1/2} \|\mathbf{g}_T - \Pi_p^\gamma \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle + \Pi_p^\gamma \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \rangle\|_{L^2(\partial T \setminus \Gamma_N)}) \\ &\leq c_{23} (h \|\mathbf{r}\|_{L^2(T)} + h^{1/2} \|\mathbf{R}\|_{L^2(\partial T)} + h^{1/2} \|\mathbf{g}_T - \Pi_p^\gamma \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle\|_{L^2(\partial T \setminus \Gamma_N)}) \\ &\quad + h^{1/2} \|\Pi_p^\gamma \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle\|_{L^2(\partial T \setminus \Gamma_N)}. \end{aligned}$$

Mit Satz 4.41 folgt

$$\begin{aligned} &a_T(\mathbf{y}_T, \mathbf{y}_T)^{1/2} \leq c_{23} \times \\ &\left(h_T \|\mathbf{r}\|_{L^2(T)} + h_T^{1/2} \|\mathbf{R}\|_{L^2(\partial T)} + h^{1/2} \|\Pi_p^\gamma \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle\|_{L^2(\partial T \setminus \Gamma_N)} \right. \\ &\quad \left. + \sum_{\gamma \subset \partial T} \sum_{n \in \mathcal{N} \cap \gamma} \sum_{T' \subset \Omega_n} (h_{T'} \|\mathbf{r}\|_{L^2(T')} + \sum_{\gamma' \subset \partial T' \cap \varepsilon_n} h_{T'}^{1/2} \|\mathbf{R}\|_{L^2(\gamma')}) \right). \end{aligned}$$

Mit \tilde{T} ist der Patch zum Element T bezeichnet, wie in (3.51) definiert, und ω_γ bezeichnet den Patch zur Kante γ , d.h. $\omega_\gamma = \cup\{T' \in \mathcal{T} : \gamma \subset T'\}$. Mit den Sätzen 4.29, 4.30 und $\mathbf{e} := \mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta}$ folgt

$$\begin{aligned} a_T(\mathbf{y}_T, \mathbf{y}_T)^{1/2} &\leq c_{24} (\|\mathbf{e}\|_{H^1(\tilde{T})} + h_T \|\mathbf{r} - \bar{\mathbf{r}}\|_{L^2(\tilde{T})} + h_T^{1/2} \|\mathbf{R} - \bar{\mathbf{R}}\|_{L^2(\cup\{\varepsilon_n : n \in \mathcal{N}, n \subset T\})}) \\ &\quad + h_T^{1/2} \|\Pi_p^\gamma \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle - \langle \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}_T \rangle\|_{L^2(\partial T \setminus \Gamma_N)} \end{aligned}$$

■

4.4 Die Methode der Patch-Residuen

Ein zweiter impliziter Fehlerschätzer ist die Methode der Patch-Residuen. Im Gegensatz zur Methode der äquilibrierten Residuen werden lokale Probleme nicht elementweise, sondern für jeden Patch gelöst. Dadurch entfällt die Äquilibrierungsprozedur, man hat es aber mit größeren lokalen Problemen zu tun. Für das Poisson Problem wurde diese Methode von C. Carstensen und S.A. Funken in [26] beschrieben. In der Kontinuumsmechanik treten bei dieser Methode lokale Probleme auf, die nur bei Verwendung finiter Elemente höherer Ordnung lösbar sind, d.h für $p \geq 2$. Dieser Abschnitt beschreibt die Methode der Patch-Residuen für den ebenen Verzerungszustand. Da die Ergebnisse in Abschnitt 4.1 auch den dreidimensionalen Fall einschließen, ist eine Verallgemeinerung möglich.

4.4.1 Zuverlässigkeit der Fehlerabschätzung

Die Methode der Patch-Residuen liefert eine obere Schranke für den Spannungsfehler. Es werden für jeden Patch lokale Probleme definiert.

Definition 4.45 (Lokale Probleme). *Mit V_z wird der Funktionenraum $V_z := \{\mathbf{v} \in H^1(\Omega_z) : \mathbf{v} = \mathbf{0} \text{ auf } \Gamma_D \cap \partial\Omega_z\}$ bezeichnet. Das Funktional $l_z(\mathbf{v}) \rightarrow \mathbb{R}$ wird definiert als*

$$l_z(\mathbf{v}) := \int_{\Omega_z} \varphi_z \mathbf{r} \cdot \mathbf{v} \, d\mathbf{x} + \int_{\cup \mathcal{E} \cap \Omega_z} \varphi_z \mathbf{R} \cdot \mathbf{v} \, d\mathbf{s}, \quad (4.162)$$

mit $\cup \mathcal{E} = \cup\{E : E \in \mathcal{E}\} = \{x : \text{für alle } E \in \mathcal{E}, x \in E\}$. Die Bilinearform $a_z(\mathbf{y}_z, \mathbf{v}) : V_z \times V_z \rightarrow \mathbb{R}$ wird definiert als

$$a_z(\mathbf{y}_z, \mathbf{v}) := \int_{\Omega_z} \varphi_z \boldsymbol{\epsilon}(\mathbf{y}_z) : \boldsymbol{\epsilon}(\mathbf{v}) \, d\mathbf{x}. \quad (4.163)$$

Es werden lokale Probleme für jeden Patch Ω_z definiert. Man bestimme $\mathbf{y}_z \in V_z$, so daß für alle $\mathbf{v} \in V_z$

$$a_z(\mathbf{y}_z, \mathbf{v}) = l_z(\mathbf{v}). \quad (4.164)$$

Definition 4.46 (Fehlerschätzer η_L und Verfeinerungsindikator $\eta_{L,z}$). *Der Fehlerschätzer η_L und der Verfeinerungsindikator $\eta_{L,z}$ sind definiert durch*

$$\eta_{L,z}^2 = a_z(\mathbf{y}_z, \mathbf{y}_z), \quad (4.165)$$

$$\eta_L := \left(\sum_{z \in \mathcal{N}} \eta_{L,z}^2 \right)^{1/2}. \quad (4.166)$$

Satz 4.47. *Für gegebene Anfangswerte $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$ gilt für einen elastischen Zeitschritt $[t_{k-1}, t_k]$ die Spannungsfehlerabschätzung*

$$\|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} \leq \left(1 + \frac{\lambda}{\mu}\right) \eta_L. \quad (4.167)$$

Für gegebene Anfangswerte $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$, $\boldsymbol{\alpha}_{k-1}$ gilt im Zeitschritt $[t_{k-1}, t_k]$ bei Visko-
plastizität mit isotroper Verfestigung die Spannungsfehlerabschätzung

$$\|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} \leq (2\lambda + 2\mu) \left(\frac{1}{2\mu} + \frac{\sqrt{3}\theta\Delta t_k}{\varrho(1 + H^2\sigma_y^2) + \theta\Delta t_k H_1 H^2\sigma_y^2} \right) \eta_L. \quad (4.168)$$

Für gegebene Anfangswerte $\mathbf{u}_{h,k-1}$, $\boldsymbol{\sigma}_{h,k-1}$, $\boldsymbol{\alpha}_{k-1}$ gilt im Zeitschritt $[t_{k-1}, t_k]$ bei
Viskoplastizität mit kinematischer Verfestigung die Spannungsfehlerabschätzung

$$\|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_{L^2(\Omega)} \leq (2\lambda + 2\mu) \left(\frac{1}{2\mu} + \frac{1}{k_1} \sqrt{2 + \frac{1}{2} \left(\frac{4\varrho + 2\theta\Delta t_k k_1}{\theta\Delta t_k k_1} \right)^2} \right) \eta_L. \quad (4.169)$$

Beweis. Mit Satz 4.4 und $\mathbf{e} := \mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta}$ gilt

$$\frac{1}{2\lambda + 2\mu} \|\boldsymbol{\sigma}_{k-1+\theta} - \boldsymbol{\sigma}_{h,k-1+\theta}\|_2^2 \leq \langle R(\mathbf{u}_{h,k-1+\theta}), \mathbf{e} \rangle.$$

Es gilt mit Definition 4.45

$$\langle R(\mathbf{u}_{h,k-1+\theta}), \mathbf{e} \rangle = \int_{\Omega} \mathbf{f}(t_{k-1+\theta}) \cdot \mathbf{e} \, d\mathbf{x} + \int_{\Gamma_N} \mathbf{g}(t_{k-1+\theta}) \cdot \mathbf{e} \, d\mathbf{s} - \int_{\Omega} \boldsymbol{\sigma}_{h,k-1+\theta} : \boldsymbol{\epsilon}(\mathbf{e}) \, d\mathbf{x}.$$

Elementweise Auswertung der rechten Seite und partielle Integration liefert

$$\begin{aligned} \langle R(\mathbf{u}_{h,k-1+\theta}), \mathbf{e} \rangle &= \sum_{T \in \mathcal{T}} \int_T (\mathbf{f}(t_{k-1+\theta}) + \operatorname{div} \boldsymbol{\sigma}_{h,k-1+\theta}) \cdot \mathbf{e} \, d\mathbf{x} \\ &\quad - \int_{\partial T} (\boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}) \cdot \mathbf{e} \, d\mathbf{s} + \int_{\Gamma_N} \mathbf{g}(t_{k-1+\theta}) \cdot \mathbf{e} \, d\mathbf{s}. \end{aligned}$$

Für $E = T^+ \cap T^-$ werden die zu den Elementen T^+ und T^- zugehörigen Spannungen
mit $\boldsymbol{\sigma}_{h,k-1+\theta}^+$ und $\boldsymbol{\sigma}_{h,k-1+\theta}^-$ bezeichnet. Der Vektor \mathbf{n}_E ist in diesem Fall der äußere
Normalenvektor von Element T^- zur Kante E . Für $E \subset \partial\Omega$ ist \mathbf{n}_E der äußere
Normalenvektor zur Kante E . Damit folgt

$$\begin{aligned} &\langle R(\mathbf{u}_{h,k-1+\theta}), \mathbf{e} \rangle \\ &= \sum_{T \in \mathcal{T}} \int_T (\mathbf{f}(t_{k-1+\theta}) + \operatorname{div} \boldsymbol{\sigma}_{h,k-1+\theta}) \cdot \mathbf{e} \, d\mathbf{x} + \int_{\Gamma_N} (\mathbf{g}(t_{k-1+\theta}) - \boldsymbol{\sigma}_{h,k-1+\theta} \cdot \mathbf{n}) \cdot \mathbf{e} \, d\mathbf{s} \\ &\quad + \sum_{E \in \mathcal{E}} \int_E ((\boldsymbol{\sigma}_{h,k-1+\theta}^+ - \boldsymbol{\sigma}_{h,k-1+\theta}^-) \cdot \mathbf{n}_E) \cdot \mathbf{e} \, d\mathbf{s}. \end{aligned}$$

Mit den Definitionen 4.23 und 4.24 für die Kantensprünge und das Volumenresiduum
erhält man

$$\langle R(\mathbf{u}_{h,k-1+\theta}), \mathbf{e} \rangle = \int_{\Omega} \mathbf{r} \cdot \mathbf{e} \, d\mathbf{x} + \int_{\cup \mathcal{E}} \mathbf{R} \cdot \mathbf{e} \, d\mathbf{s} = \sum_{z \in \mathcal{N}} \int_{\Omega_z} \varphi_z \mathbf{r} \cdot \mathbf{e} \, d\mathbf{x} + \int_{\cup \mathcal{E} \cap \Omega_z} \varphi_z \mathbf{R} \cdot \mathbf{e} \, d\mathbf{s}.$$

Anwendung von Hölderungleichung und diskreter Hölderungleichung zeigt

$$\begin{aligned}
& \frac{1}{2\lambda + 2\mu} \|\sigma_{k-1+\theta} - \sigma_{h,k-1+\theta}\|_2^2 \leq \sum_{z \in \mathcal{N}} l_z(\mathbf{e}) = \sum_{z \in \mathcal{N}} \int_{\Omega_z} \varphi_z \boldsymbol{\epsilon}(\mathbf{y}_z) : \boldsymbol{\epsilon}(\mathbf{e}) \, d\mathbf{x} \\
& \leq \sum_{z \in \mathcal{N}} \|\varphi_z^{1/2} \boldsymbol{\epsilon}(\mathbf{y}_z)\|_{2,\Omega_z}^2 \|\varphi_z^{1/2} \boldsymbol{\epsilon}(\mathbf{e})\|_{2,\Omega_z}^2 \\
& \leq \left(\sum_{z \in \mathcal{N}} \|\varphi_z^{1/2} \boldsymbol{\epsilon}(\mathbf{y}_z)\|_{2,\Omega_z}^2 \right)^{1/2} \left(\sum_{z \in \mathcal{N}} \|\varphi_z^{1/2} \boldsymbol{\epsilon}(\mathbf{e})\|_{2,\Omega_z}^2 \right)^{1/2} \\
& = \eta_L \|\boldsymbol{\epsilon}(\mathbf{e})\|_2.
\end{aligned}$$

Die Sätze 4.7, 4.15 bzw. 4.16 werden zur Absorption der rechten Seite benutzt. ■

4.4.2 Lösung der lokalen Probleme

Lösbarkeit

Ein Beweis für die Existenz von Lösungen der lokalen Probleme in Definition 4.45 ist ein offenes Problem, da die Elliptizität der Bilinearform $a(\mathbf{v}, \mathbf{v})$ in (4.163) nicht klar ist. Es kann aber eine notwendige Bedingung für die Existenz von Lösungen angegeben werden. Falls $|\partial\Omega_z \cap \Gamma_D| = 0$, handelt es sich bei den lokalen Problemen (4.164) um Neumann-Probleme. In diesem Fall lautet eine notwendige Bedingung für die Existenz von Lösungen, daß für $Z_z := \{\mathbf{z} \in V_z : a_z(\mathbf{v}, \mathbf{z}) = 0 \quad \forall \mathbf{v} \in V_z\}$ gilt

$$l_z(\mathbf{v}) = 0 \quad \text{für alle } \mathbf{v} \in Z_z. \quad (4.170)$$

Dabei enthält Z_z die linearen Starrkörperbewegungen, d.h. $\mathbf{z} \in Z_z$, falls $\mathbf{z}(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x} + \mathbf{b}$, wobei \mathbf{A} eine Drehmatrix und \mathbf{b} ein konstanter Vektor ist.

Satz 4.48. *Die lokalen Probleme aus Definition (4.45) können für $|\partial\Omega_z \cap \Gamma_D| = 0$ nur dann eine Lösung besitzen, falls die Galerkin-Orthogonalität 2. Grades erfüllt ist.*

Beweis. Es ist zu überprüfen, daß $l_z(\mathbf{v}) = 0$, falls $\mathbf{v} \in Z_z$. Die Menge Z_z besteht aus den Starrkörperbewegungen. Für $\mathbf{v} \in Z_z$ gilt $\varphi_z \mathbf{v} \in \mathcal{S}^2(\Omega)$. Mit partieller Integration erhält man die Darstellung

$$l_z(\mathbf{v}) = \int_{\Omega_z} \mathbf{f}(t_{k-1+\theta}) \cdot (\varphi_z \mathbf{v}) \, d\mathbf{x} - \int_{\Omega_z} \sigma_{h,k-1+\theta} : D(\varphi_z \mathbf{v}) \, d\mathbf{x} + \int_{\partial\Omega_z \cap \Gamma_N} \mathbf{g}(t_{k-1+\theta}) \cdot (\varphi_z \mathbf{v}) \, ds \quad (4.171)$$

für das Funktional (4.162). Die Forderung $l_z(\mathbf{v}) = 0$ für $\mathbf{v} \in Z_z$ in (4.171) ist identisch mit der Forderung, daß die Galerkin-Orthogonalität zweiten Grades erfüllt ist. ■

Die Galerkin-Orthogonalität 2. Grades ist für Finite-Elemente-Methoden höherer Ordnung gegeben. Erforderlich sind Ansatzfunktionen mindestens 2. Grades für die Verschiebung.

Numerische Lösung

Die lokalen Probleme (4.164) sind auf einem endlichdimensionalen Teilraum zu lösen.

Definition 4.49 (Endlichdimensionale lokale Probleme). Mit $V_{z,h}$ wird der Funktionenraum $V_{z,h} := \{\mathbf{v}_h \in H^1(\Omega_z) \cap \mathcal{S}^q(\Omega) : \mathbf{v}_h = \mathbf{0} \text{ auf } \Gamma_D \cap \partial\Omega_z\}$ bezeichnet. Der Polynomgrad q dieses Finite-Elemente-Raumes ist höher zu wählen als der Polynomgrad p zur Lösung des Modellproblems. Mit (4.162) für das Funktional l_z und (4.163) für die Bilinearform a_z sind folgende endlichdimensionale Probleme zu lösen. Bestimme $\mathbf{y}_{z,h} \in V_{z,h}$, so daß für alle $\mathbf{v}_h \in V_{z,h}$

$$a_z(\mathbf{y}_{z,h}, \mathbf{v}_h) = l_z(\mathbf{v}_h). \quad (4.172)$$

Falls $|\partial\Omega_z \cap \Gamma_D| = 0$, ist die zugehörige Steifigkeitsmatrix \mathbf{A}_z des lokalen Problems singulär. In diesem Fall ist ein Gleichungssystem $\mathbf{A}_z \mathbf{x} = \mathbf{b}$ zu lösen, für das $\text{rg}(\mathbf{A}_z) = \text{rg}(\mathbf{A}_z \mathbf{b})$ gilt, da aufgrund der Betrachtungen im letzten Abschnitt eine nichteindeutige Lösung existiert. Der Vektor \mathbf{x} wird analog zum letzten Abschnitt mit Hilfe einer QR-Zerlegung bestimmt.

Definition 4.50 (Fehlerschätzer $\eta_{L,h}$ und Verfeinerungsindikator $\eta_{L,z,h}$). Die Lösungen der endlichdimensionalen lokalen Probleme in Definition 4.49 definieren den Fehlerschätzer $\eta_{L,h}$ und den Verfeinerungsindikator $\eta_{L,z,h}$ durch

$$\eta_{L,z,h}^2 = a_z(\mathbf{y}_{z,h}, \mathbf{y}_{z,h}), \quad (4.173)$$

$$\eta_L := \left(\sum_{z \in \mathcal{N}} \eta_{L,z,h}^2 \right)^{1/2}. \quad (4.174)$$

4.4.3 Effizienz der Fehlerabschätzung

Effizienz für die Methode der Patch-Residuen kann mit Hilfe der Sätze 4.29 und 4.30 gezeigt werden, die für h-Verfeinerung gelten. Der Durchmesser h_z von Ω_z ist definiert als

$$h_z := \max\{h_T : T \in \mathcal{T}, T \subset \Omega_z\}. \quad (4.175)$$

Satz 4.51. Ist $\mathbf{y}_{z,h}$ eine Lösung des lokalen Problems aus Definition 4.49, dann gilt mit $\bar{\mathbf{r}}$ und $\bar{\mathbf{R}}$ aus den Sätzen 4.29 und 4.30 für den Fehler $\mathbf{e} := \mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta}$ mit einer Konstanten c_{31} die Abschätzung

$$a(\mathbf{y}_{z,h}, \mathbf{y}_{z,h})^{1/2} \leq c_{31} (\|\mathbf{e}\|_{H^1(\Omega_z)} + h_z \|\mathbf{r} - \bar{\mathbf{r}}\|_{L^2(\Omega_z)} + h_z^{1/2} \|\mathbf{R} - \bar{\mathbf{R}}\|_{L^2(\cup \mathcal{E}_z)}). \quad (4.176)$$

Beweis. Falls für die Lösung des lokalen Problems in Definition 4.49 $\mathbf{y}_{z,h} \in Z_z$ gilt, d.h. $\mathbf{y}_{z,h}$ eine Starrkörperbewegung ist, gilt $a(\mathbf{y}_{z,h}, \mathbf{y}_{z,h}) = 0$ und die Behauptung ist bewiesen. Im folgenden wird daher angenommen, daß $\mathbf{y}_{z,h} \in V_{z,h} \setminus Z_z$. Dann gilt für alle $\mathbf{v}_h \in V_{z,h}$ und im speziellen für alle $\mathbf{v}_h \in Z_z$

$$\begin{aligned} a(\mathbf{y}_{z,h}, \mathbf{v}_h) &= a(\mathbf{y}_{z,h}, \mathbf{v}_h - \bar{\mathbf{v}}_h) = l_z(\mathbf{v}_h - \bar{\mathbf{v}}_h) \\ &= \int_{\Omega_z} \varphi_z \mathbf{r} \cdot (\mathbf{v}_h - \bar{\mathbf{v}}_h) d\mathbf{x} + \int_{\cup \mathcal{E} \cap \Omega_z} \varphi_z \mathbf{R} \cdot (\mathbf{v}_h - \bar{\mathbf{v}}_h) ds \end{aligned}$$

wegen der Galerkin-Orthogonalität 1. Ordnung, wenn mit $\bar{\mathbf{v}}_h$ das Integralmittel von \mathbf{v}_h über den Patch Ω_z bezeichnet ist. Mit der Cauchy-Ungleichung und Satz 4.43 folgt

$$\begin{aligned} a(\mathbf{y}_{z,h}, \mathbf{v}_h) &\leq \|\varphi_z \mathbf{r}\|_{L^2(\Omega_z)} \|\mathbf{v}_h - \bar{\mathbf{v}}_h\|_{L^2(\Omega_z)} + \|\varphi_z \mathbf{R}\|_{L^2(\cup \mathcal{E} \cap \Omega_z)} \|\mathbf{v}_h - \bar{\mathbf{v}}_h\|_{L^2(\cup \mathcal{E} \cap \Omega_z)} \\ &\leq \|\varphi_z \mathbf{r}\|_{L^2(\Omega_z)} \|\mathbf{v}_h - \bar{\mathbf{v}}_h\|_{L^2(\Omega_z)} + c_{25} \|\varphi_z \mathbf{R}\|_{L^2(\cup \mathcal{E} \cap \Omega_z)} \|\mathbf{v}_h - \bar{\mathbf{v}}_h\|_{L^2(\Omega_z)}^{1/2} \|\mathbf{v}_h - \bar{\mathbf{v}}_h\|_{H^1(\Omega_z)}^{1/2}. \end{aligned}$$

Mit der Friedrichs-Ungleichung $\|\mathbf{v}_h - \bar{\mathbf{v}}_h\|_{H^1(\Omega_z)} \leq c_{25} |\mathbf{v}_h|_{H^1(\Omega_z)}$ und der Poincaré-Ungleichung $\|\mathbf{v}_h - \bar{\mathbf{v}}_h\|_{L^2(\Omega_z)} \leq c_{22} h_z |\mathbf{v}_h|_{H^1(\Omega_z)}$ folgt

$$\|\mathbf{v}_h - \bar{\mathbf{v}}_h\|_{L^2(\Omega_z)}^{1/2} \|\mathbf{v}_h - \bar{\mathbf{v}}_h\|_{H^1(\Omega_z)}^{1/2} \leq \sqrt{c_{22} c_{25}} h_z^{1/2} |\mathbf{v}_h|_{H^1(\Omega_z)}$$

und

$$a(\mathbf{y}_{z,h}, \mathbf{v}_h) \leq c_{26} (h_z \|\varphi_z \mathbf{r}\|_{L^2(\Omega_z)} + h_z^{1/2} \|\varphi_z \mathbf{R}\|_{L^2(\cup \mathcal{E} \cap \Omega_z)}) \|\mathbf{D}\mathbf{v}_h\|_{L^2(\Omega_z)}.$$

Für alle $\mathbf{v}_h \in V_{z,h} \setminus Z_{\Omega_z}$ gilt dann auch mit Satz 4.42

$$a(\mathbf{y}_z, \mathbf{v}_h) \leq c_{27} (h_z \|\varphi_z \mathbf{r}\|_{L^2(\Omega_z)} + h_z^{1/2} \|\varphi_z \mathbf{R}\|_{L^2(\cup \mathcal{E} \cap \Omega_z)}) \|\boldsymbol{\epsilon}(\mathbf{v}_h)\|_{L^2(\Omega_z)}.$$

Die Abbildung

$$\mathbf{v}_h \rightarrow \left(\int_{\Omega_z} \varphi_z^{1/2} |\mathbf{v}_h|^2 d\mathbf{x} \right)^{1/2}$$

definiert eine Norm auf dem endlichdimensionalen Raum $V_{z,h}$. Aufgrund der Äquivalenz von Normen auf endlichdimensionalen Räumen gilt dann

$$a(\mathbf{y}_{z,h}, \mathbf{v}_h) \leq c_{28} (h_z \|\varphi_z \mathbf{r}\|_{L^2(\Omega_z)} + h_z^{1/2} \|\varphi_z \mathbf{R}\|_{L^2(\cup \mathcal{E} \cap \Omega_z)}) \|\varphi_z^{1/2} \boldsymbol{\epsilon}(\mathbf{v}_h)\|_{L^2(\Omega_z)}.$$

Für $\mathbf{v}_h = \mathbf{y}_{z,h}$ gilt

$$\begin{aligned} a(\mathbf{y}_{z,h}, \mathbf{y}_{z,h})^{1/2} &\leq c_{28} (h_z \|\varphi_z \mathbf{r}\|_{L^2(\Omega_z)} + h_z^{1/2} \|\varphi_z \mathbf{R}\|_{L^2(\cup \mathcal{E} \cap \Omega_z)}) \\ &\leq c_{29} (h \|\mathbf{r}\|_{L^2(\Omega_z)} + h_z^{1/2} \|\mathbf{R}\|_{L^2(\cup \mathcal{E}_z)}). \end{aligned}$$

Mit ω_E wird der Patch zur Kante E bezeichnet, d.h. $\omega_E = \cup \{T' \in \mathcal{T} : E \subset T'\}$. Mit (3.52), den Sätzen 4.29, 4.30 und $\mathbf{e} := \mathbf{u}_{k-1+\theta} - \mathbf{u}_{h,k-1+\theta}$ folgt

$$\begin{aligned} a(\mathbf{y}_{z,h}, \mathbf{y}_{z,h})^{1/2} &\leq c_{30} (\|\mathbf{e}\|_{H^1(\Omega_z)} + h_z \|\mathbf{r} - \bar{\mathbf{r}}\|_{L^2(\Omega_z)} + \|\mathbf{e}\|_{H^1(\cup \{\omega_E : E \in \mathcal{E}_z\})} \\ &\quad + h_z \|\mathbf{r} - \bar{\mathbf{r}}\|_{L^2(\cup \{\omega_E : E \in \mathcal{E}_z\})} + h_z^{1/2} \|\mathbf{R} - \bar{\mathbf{R}}\|_{L^2(\cup \mathcal{E}_z)}) \\ &\leq c_{31} (\|\mathbf{e}\|_{H^1(\Omega_z)} + h_z \|\mathbf{r} - \bar{\mathbf{r}}\|_{L^2(\Omega_z)} + h_z^{1/2} \|\mathbf{R} - \bar{\mathbf{R}}\|_{L^2(\cup \mathcal{E}_z)}) \end{aligned}$$

■

Kapitel 5

Numerische Ergebnisse

In diesem Kapitel wird der Lösungsalgorithmus für das elastoviskoplastische Modellproblem anhand von numerischen Beispielen getestet. Die Ergebnisse sind mit den im Rahmen dieser Arbeit implementierten Programmen berechnet worden, die im Anhang aufgeführt sind.

Kapitel 4 hat gezeigt, daß für das elastoviskoplastische Problem in jedem Zeitschritt eine berechenbare Schranke für den Spannungsfehler existiert. In der Fehlerabschätzung treten materialabhängige Konstanten auf. In Abschnitt 5.2 wird untersucht, wie der globale Effektivitätsindex der Fehlerschätzer von Materialparametern abhängt.

In Abschnitt 5.3 wird anhand von numerischen Beispielen die Approximationsqualität verschiedener adaptiv generierter Netze untersucht.

5.1 Definition von Testproblemen

Beispiel 5.1 (*Das elastische L-Gebiet-Problem*). Die Geometrie des L-Gebiet-Problems zusammen mit der Ausgangstriangulierung ist in Abbildung 5.1 gezeigt. Dieses Standard-Benchmarkproblem löst das Modellproblem für den elastischen Fall bei verschwindender Volumenkraft und Dirichlet-Randbedingungen. Die Verschiebung \mathbf{w} auf dem Dirichlet-Rand $\Gamma_D = \Gamma$ ist durch die bekannte exakte Lösung \mathbf{u} vorgegeben. In Polarkoordinaten ($r \in [0, \infty)$, $\varphi \in [-\pi, \pi]$) ist \mathbf{u} gegeben durch

$$\mathbf{u}(r, \varphi) = u_r(r, \varphi)\mathbf{e}_r + u_\varphi(r, \varphi)\mathbf{e}_\varphi, \quad (5.1)$$

mit

$$\begin{aligned} u_r(r, \varphi) &= \frac{1}{2\mu} r^\alpha (-(\alpha + 1) \cos((\alpha + 1)\varphi) + (C_2 - (\alpha + 1))C_1 \cos((\alpha - 1)\varphi)), \\ u_\varphi(r, \varphi) &= \frac{1}{2\mu} r^\alpha ((\alpha + 1) \sin((\alpha + 1)\varphi) + (C_2 + \alpha - 1)C_1 \sin((\alpha - 1)\varphi)). \end{aligned} \quad (5.2)$$

Der Exponent $\alpha = 0.54483737$ ist die Lösung der Gleichung

$$\alpha \sin(2\omega) + \sin(2\omega\alpha) = 0$$

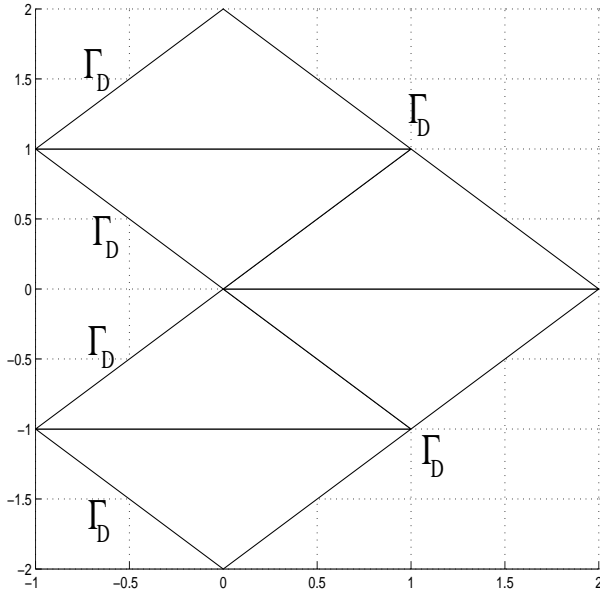


Abbildung 5.1: Ausgangstriangulierung für Beispiel 5.1.

mit

$$\omega = 3\pi/4, \quad C_1 = -(\cos((\alpha+1)\omega))/(\cos((\alpha-1)\omega)), \quad C_2 = (2(\lambda+2\mu))/(\lambda+\mu).$$

Beispiel 5.2 (Der radialbelastete Ring). Die Geometrie des radialbelasteten Ringes ist in Abbildung 5.2 gezeigt. Dargestellt wird der zweidimensionale Ausschnitt eines in z-Richtung unendlich ausgedehnten Zylinders mit innerem Radius r_1 und äußerem Radius r_2 . Der Zylinder ist zeitabhängigen radialen Oberflächenkräften $\mathbf{g}_1(t) = t\mathbf{e}_r$ und $\mathbf{g}_2(t) = -t/4\mathbf{e}_r$ ausgesetzt, die an dem gesamten Zylindermantel angreifen und zu einer elasto-plastischen Verformung führen. Aufgrund der Symmetrie des Problems braucht nur ein Viertel des Gebietes diskretisiert zu werden. Auf der x-Achse und der y-Achse sind Gleitrandbedingungen vorgegeben. Die Finite-Elemente-Triangulierung ist eine polygonale Näherung des Gebietes. Es wird das Modell der kinematischen Verfestigung betrachtet. Für dieses Problem existiert eine explizit bekannte Lösung [7] von Problem 1 für $\varrho \rightarrow 0$,

$$\begin{aligned} u(r, \phi, t) &= u_r(r, t)\mathbf{e}_r, \\ \sigma(r, \phi, t) &= \sigma_r(r, t)\mathbf{e}_r \otimes \mathbf{e}_r + \sigma_\phi\mathbf{e}_\phi \otimes \mathbf{e}_\phi, \\ p(r, \phi, t) &= P_r(r, t)(\mathbf{e}_r \otimes \mathbf{e}_r - \mathbf{e}_\phi \otimes \mathbf{e}_\phi), \end{aligned} \quad (5.3)$$

mit $\mathbf{e}_r = (\cos \phi, \sin \phi)$, $\mathbf{e}_\phi = (-\sin \phi, \cos \phi)$ und, mit $a = \mu + \lambda$, $\kappa = 2\mu/(2\mu + \lambda)$,

$$u_r(r, t) = \begin{cases} \frac{t}{2\mu r} - \frac{1}{3}\kappa I(R(t)) \left(r + \frac{4a}{\mu r} \right) & \text{für } r \geq R(t), \\ \frac{t}{2\mu r} - \frac{1}{3}\kappa I(R(t)) \left(4r + \frac{4a}{\mu r} \right) + \kappa I(r)r & \text{für } r < R(t), \end{cases} \quad (5.4)$$

$$\sigma_r(r, t) = \begin{cases} -\frac{t}{r^2} - \frac{8}{3}a\kappa I(R(t)) \left(\frac{1}{4} - \frac{1}{r^2} \right) & \text{für } r \geq R(t) \\ -\frac{t}{r^2} - \frac{8}{3}a\kappa I(R(t)) \left(1 - \frac{1}{r^2} \right) + 2a\kappa I(r) & \text{für } r < R(t), \end{cases} \quad (5.5)$$

$$\sigma_\phi(r, t) = \partial(r\sigma_r)/\partial r, \quad (5.6)$$

$$P_r(r, t) = \begin{cases} 0 & \text{für } r \geq R(t), \\ \frac{\sigma_y}{\sqrt{2}(a\kappa + k_1)}(1 - R^2/r^2) & \text{für } r < R(t), \end{cases} \quad (5.7)$$

$$I(r) = \frac{\sigma_y}{\sqrt{2}(a\kappa + k_1)}(\ln r + 1/2(R^2/r^2 - R^2)). \quad (5.8)$$

Mit $\alpha = 4a\kappa/(3(a\kappa + k_1))$ ist der Radius der plastischen Grenze $R(t)$ die Lösung der Gleichung

$$f(R) = -2\alpha \ln R + (\alpha - 1)R^2 - \alpha + \frac{\sqrt{2}}{\sigma_y}t. \quad (5.9)$$

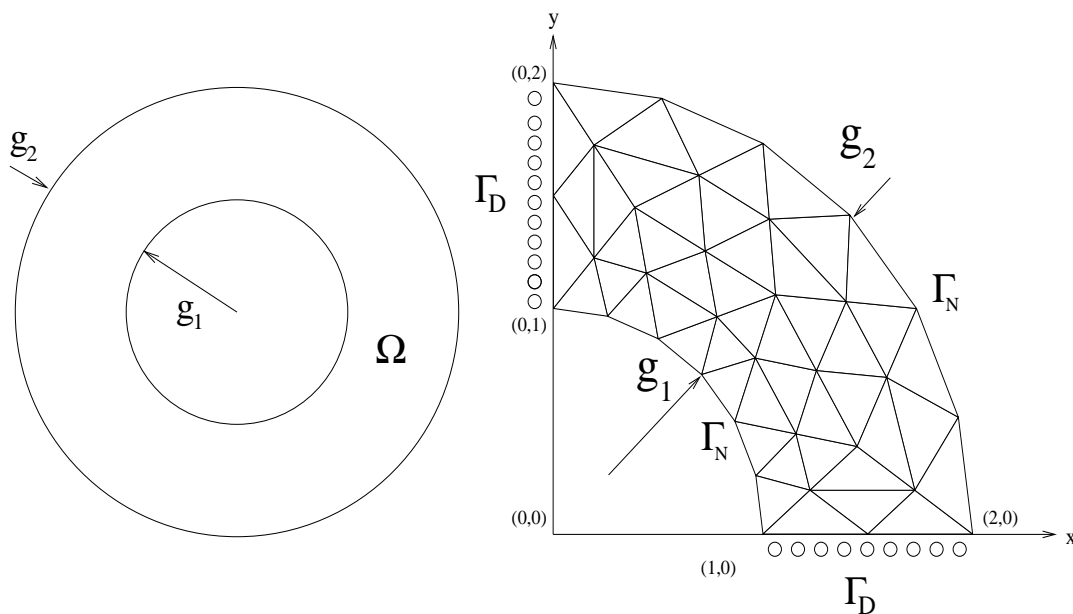


Abbildung 5.2: Radialsymmetrischer Ring (linkes Bild). Geometrie (rechtes Bild) mit Oberflächenkräften $g_1(t) = te_r$ und $g_2(t) = -t/4e_r$, die auf den gesamten Neumann-Rand wirken.

Das betrachtete Zeitintervall ist $[0, 4]$. Für die Materialparameter werden folgende Werte gewählt:

$$E = 210000, \quad \nu = 0.28, \quad k_1 = 28000, \quad \sigma_y = 5$$

Beispiel 5.3 (Cooks Membrane). Eine zweidimensionale Scheibe aus Plexiglas, beschrieben durch das Gebiet $\Omega = \text{conv}\{(0, 0), (48, 44), (48, 60), (0, 44)\}$ ist an der Kante $(x = 0)$ befestigt und bei verschwindender Volumenkraft ($\mathbf{f} = \mathbf{0}$) an der gesamten gegenüberliegenden Kante $(x = 48)$ einer zeitabhängigen Scherkraft $\mathbf{g} = (0, 8.0t)$ ausgesetzt. Abbildung 5.3 zeigt die Geometrie des Problems. Die Materialparameter im Modell der Plastizität mit kinematischer Verfestigung werden gewählt als

$$E = 2900, \quad \nu = 0.4, \quad k_1 = 100, \quad \sigma_y = 50, \quad \varrho = 0.$$

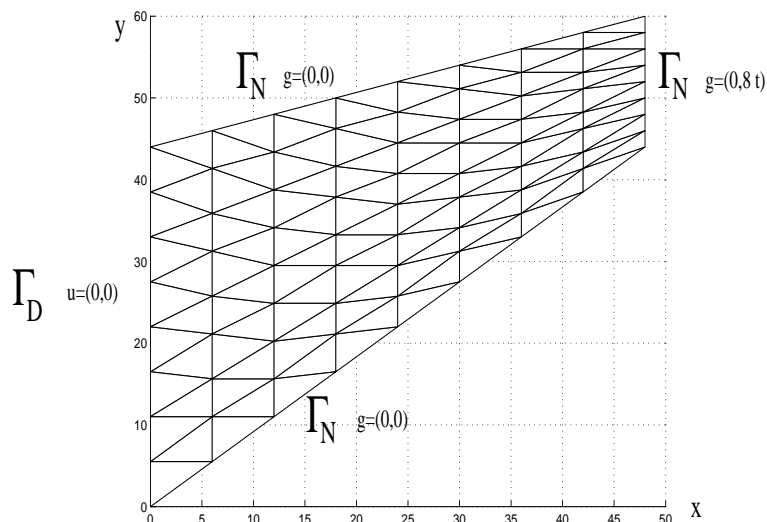


Abbildung 5.3: Geometrie für Beispiel 5.3.

Beispiel 5.4 (L-Gebiet). Ein L-Gebiet mit den Eckpunkten $\{(0.5, 0.5), (0.5, 0.0), (1.0, 0.0), (1.0, 1.0), (0.5, 1.0), (0.0, 1.0), (0.0, 0.5)\}$ ist an der Kante $y = 0$ befestigt und an der gesamten Kante $y = 1$ einer zeitabhängigen Oberflächenkraft $g = (0, 1.2t)$ ausgesetzt. Die Geometrie des Problems zeigt Abbildung 5.4. Das betrachtete Zeitintervall ist $[0, 0.1]$. Die Materialparameter im Modell der kinematischen Verfestigung werden gewählt als

$$E = 206900, \quad \nu = 0.29, \quad k_1 = 50000, \quad \sigma_y = 100.$$

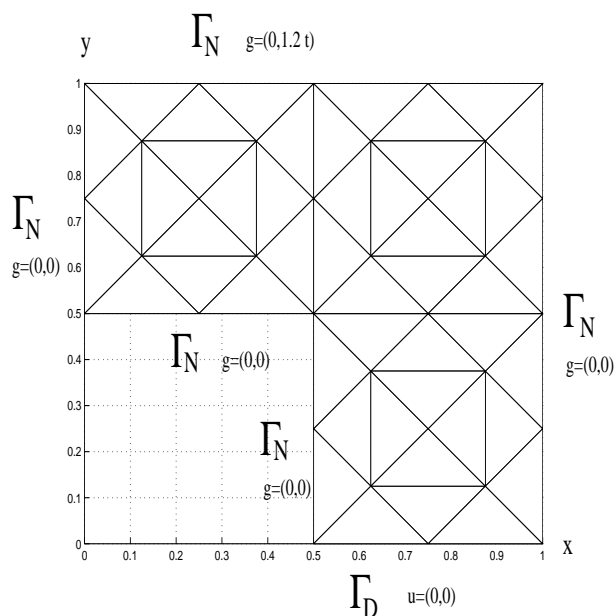


Abbildung 5.4: Geometrie für Beispiel 5.4.

Beispiel 5.5 (Lochscheibe). Eine zweidimensionale Lochscheibe $\Omega = (-3, 3)^2 \setminus B(0, 1)$, wie in Abbildung 5.5 gezeigt, ist einer zeitabhängigen Oberflächenkraft $\mathbf{g}(t) = 30t\mathbf{n}$ an der oberen Kante ($y = 3$) und der unteren Kante ($y = -3$) ausgesetzt. Hierbei ist \mathbf{n} der äußere Normalenvektor von $\partial\Omega$. Der Rest des Randes ist kräftefrei. Aufgrund der Symmetrie des Problems braucht nur ein Viertel des Gebietes diskretisiert zu werden. Die Finite-Elemente-Triangulierung nähert das Gebiet Ω durch ein polygonales Gebiet an. Die Materialparameter im Modell der kinematischen Verfestigung werden gewählt als

$$E = 210000, \quad \nu = 0.28, \quad \sigma_y = 50, \quad k_1 = 60000, \quad \varrho = 0.$$

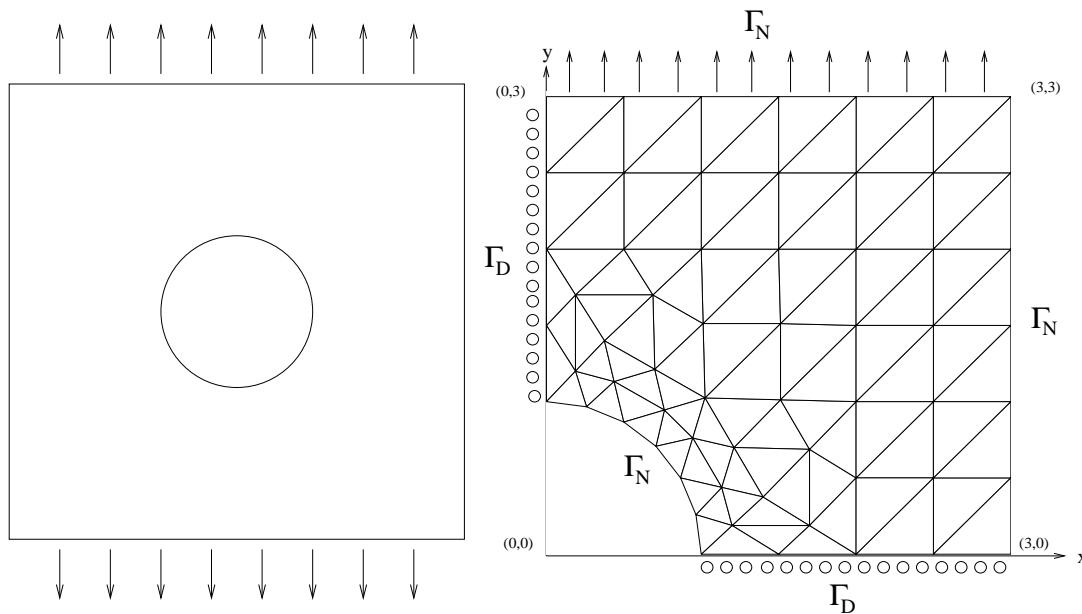


Abbildung 5.5: Geometrie und Ausgangstriangulierung für Beispiel 5.5.

5.2 Der Effektivitätsindex in Abhängigkeit von Materialparametern

Kapitel 4 hat gezeigt, daß in jedem Zeitschritt der in Kapitel 3 vorgestellten Lösungsmethode eine Fehlerabschätzung mit garantierter oberer Schranke für den Spannungsfehler existiert. Fehlerabschätzungen nach der Methode der äquilibrierten Residuen und der Patch-Residuen-Methode sind nach Satz 4.34 und 4.47 von Materialparametern abhängig. Der Effektivitätsindex dieser Schätzer soll im folgenden anhand der Beispiele 5.1 und 5.2 bestimmt werden.

Das elastische Problem aus Beispiel 5.1 wird mit einer h -uniformen P_2 -Finite-Elemente-Methode gelöst. Es werden der Spannungsfehler sowie die Fehlerschätzer $\eta_{EQ,h}$ und $\eta_{L,h}$ berechnet. Die lokalen Probleme in den Fehlerschätzern werden mit einer p -Methode mit Polynomgrad $p = 4$ berechnet. Nach Satz 4.34 und Satz 4.47

gelten für den Spannungsfehler die Abschätzungen

$$\|\boldsymbol{\sigma} - \boldsymbol{\sigma}_h\|_{L^2(\Omega)} \leq \underbrace{\left(1 + \frac{\lambda}{\mu}\right)}_{c_M} \eta_{EQ} \quad \text{und} \quad \|\boldsymbol{\sigma} - \boldsymbol{\sigma}_h\|_{L^2(\Omega)} \leq \underbrace{\left(1 + \frac{\lambda}{\mu}\right)}_{c_M} \eta_L.$$

Die Fehlerabschätzungen sind abhängig von der Materialkonstante

$$c_M = 1 + \frac{\lambda}{\mu}. \quad (5.10)$$

Aus den Beziehungen [27]

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad \text{und} \quad \mu = \frac{E}{2(1+\nu)}$$

wird ersichtlich, daß die Querkontraktion ν ein kritischer Parameter ist. Für ν nahe 0.5 werden die Fehlerschätzer $\eta_{EQ,h}$ und $\eta_{L,h}$ eine sehr hohe Überschätzung haben. Tabelle 5.1 zeigt für die Materialparameter von Beton, Stahl, Kupfer und Blei, den Spannungsfehler, die Werte für die Fehlerschätzer $\eta_{EQ,h}$, $\eta_{L,h}$, die Materialkonstante c_M , sowie den Effektivitätsindex beider Fehlerabschätzungen. Abbildung 5.6 stellt den Spannungsfehler, und die Werte $c_M\eta_{EQ,h}$ sowie $c_M\eta_{L,h}$ in Abhängigkeit der Freiheitsgrade dar. In allen vier Rechnungen ist der Effektivitätsindex kleiner als 10, für $\eta_{L,h}$ ist er jeweils kleiner als für $\eta_{EQ,h}$.

ν	$\ \boldsymbol{\sigma} - \boldsymbol{\sigma}_h\ _2$	$\eta_{EQ,h}$	$\eta_{L,h}$	c_M	$C_{Eff,1}$	$C_{Eff,2}$
0.20	0.4786	0.6131	0.4949	1.6667	2.1353	1.7237
0.28	0.5210	0.6412	0.5252	2.2727	2.7972	2.2911
0.34	0.6001	0.7073	0.5835	3.1250	3.6834	3.0390
0.44	1.0895	1.1787	0.9489	8.3333	9.0152	7.2582

Tabelle 5.1: Effektivitätskonstanten $C_{Eff,1} = c_M\eta_{EQ,h}/\|\boldsymbol{\sigma} - \boldsymbol{\sigma}_h\|_{L^2(\Omega)}$ und $C_{Eff,2} = c_M\eta_{L,h}/\|\boldsymbol{\sigma} - \boldsymbol{\sigma}_h\|_{L^2(\Omega)}$ in Abhängigkeit von der Querkontraktion ν in Testbeispiel 5.1. Finite-Elemente-Methode mit $p = 2$, 25090 Freiheitsgrade.

Das plastische Problem aus Beispiel 5.2 wird mit einer P_2 -Finite-Elemente-Methode gelöst. Auf dem Neumannrand werden die Randkräfte $\mathbf{g} = \boldsymbol{\sigma} \cdot \mathbf{n}$ vorgegeben, wobei $\boldsymbol{\sigma}$ die exakte Lösung ist und \mathbf{n} der äußere Normalenvektor. Der polygonale Rand wird nach jeder Verfeinerung angepaßt, sodaß Knoten auf dem inneren Neumannrand auf einem Kreis mit dem Radius 1 um den Koordinatenursprung liegen und Knoten auf dem äußeren Neumannrand auf einem Kreis mit dem Radius 2. Als Ausgangsnetz wird eine Triangulierung mit 113 Knoten verwendet. Es werden für verschiedene Werte des plastischen Parameters k_1 je vier Experimente mit steigender Vorgabe an die Genauigkeit durchgeführt. Das betrachtete Zeitintervall ist in allen Experimenten $[0, 4]$. Mit n ist die Anzahl der Zeitschritte in einem Experiment bezeichnet. Berechnet werden der Spannungsfehler $\|\boldsymbol{\sigma}(t_{n-1+\theta}) - \boldsymbol{\sigma}_{h,n-1+\theta}\|_{L^2(\Omega)}$ sowie die Fehlerschätzer $\eta_{EQ,h}$ und $\eta_{L,h}$. Nach Satz 4.34 und Satz 4.47 gelten für den

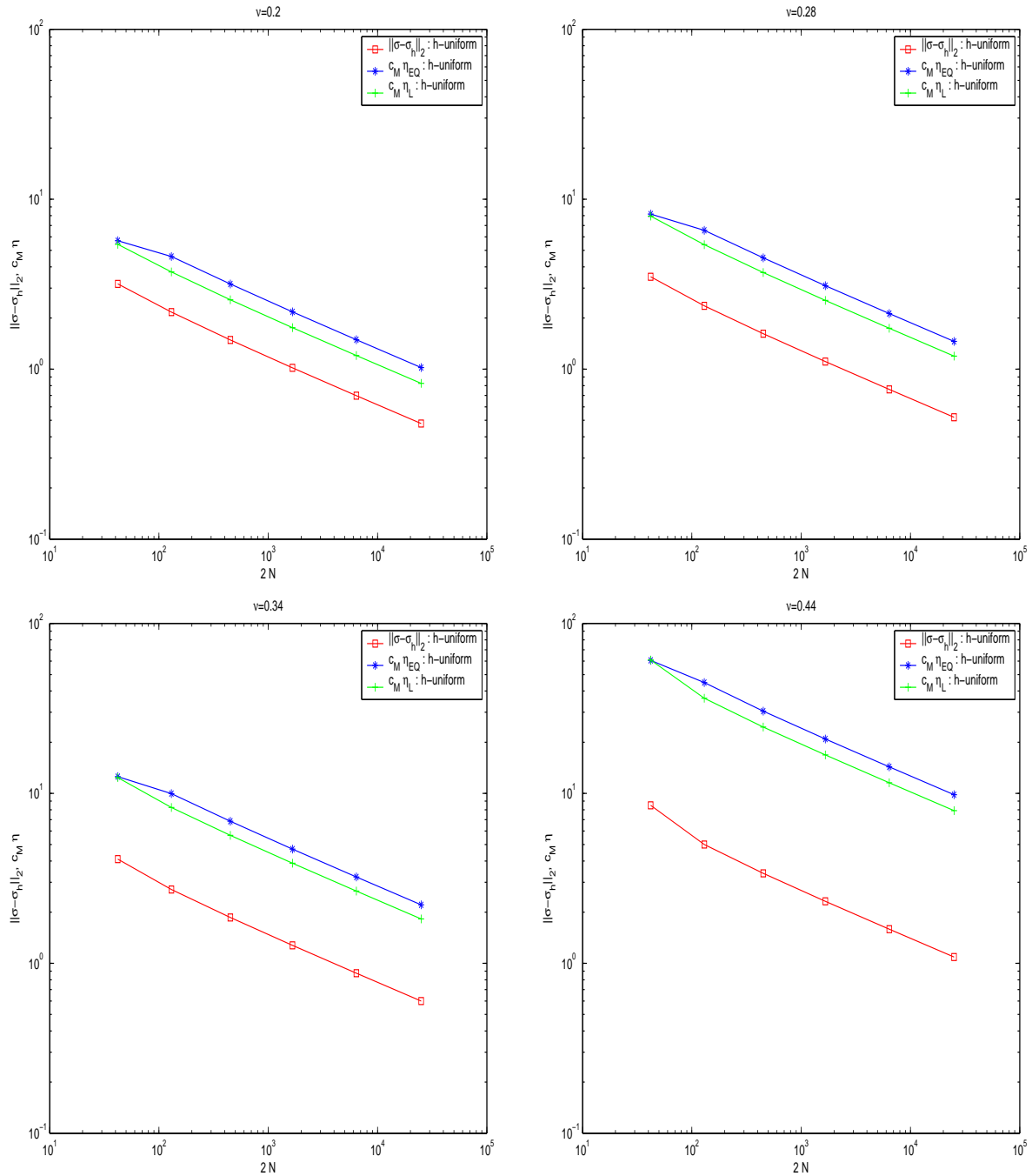


Abbildung 5.6: Fehlerabschätzungen für Beispiel 5.1 in Abhängigkeit der Freiheitsgrade für verschiedene Werte der Querkontraktion ν .

Spannungsfehler im letzten Zeitschritt $[t_{n-1}, t_{n-1+\theta}]$ die Fehlerabschätzungen

$$\|\sigma_{n-1+\theta} - \sigma_{h,n-1+\theta}\|_{L^2(\Omega)} \leq \underbrace{(2\lambda + 2\mu)\left(\frac{1}{2\mu} + \frac{1}{k_1}\right)}_{c_M} \eta_{EQ,h}$$

und

$$\|\sigma_{n-1+\theta} - \sigma_{h,n-1+\theta}\|_{L^2(\Omega)} \leq \underbrace{(2\lambda + 2\mu)\left(\frac{1}{2\mu} + \frac{1}{k_1}\right)}_{c_M} \eta_{L,h}.$$

Der Gesamtfehler $\|\sigma(t_{n-1+\theta}) - \sigma_{h,n-1+\theta}\|_{L^2(\Omega)}$ ist im Gegensatz zu dem Fehler des letzten Zeitschrittes von dem Zeitdiskretisierungsfehler und der Fehlerakkumulation der Ortsdiskretisierungsfehler vorheriger Zeitschritte beeinflusst. Daher wird der Gesamtfehler größer sein, als der Fehler des letzten Zeitschrittes. Durch den Vergleich der berechneten Fehlerschätzer mit dem Gesamtfehler ergeben sich daher Aussagen über den globalen Effektivitätsindex, die als ein Mindestwert zu verstehen sind.

Die Fehlerabschätzungen sind von dem Faktor

$$c_M = (2\lambda + 2\mu)\left(\frac{1}{2\mu} + \frac{2}{k_1}\right) \quad (5.11)$$

abhängig, der von den Lamékonstanten λ und μ sowie dem Verfestigungsparameter k_1 abhängt. Neben dem in der Elastizität untersuchten kritischen Parameter ν , tritt hier eine problematische Abhängigkeit von der plastischen Variablen k_1 auf. Entscheidend ist der Wert dieser Variablen im Vergleich zum Elastizitätsmodul. Es treten Werte $0 \leq k_1 \leq E$ in der Praxis auf [46]. Bei isotroper Verfestigung spielen die Variablen H_1 , H und σ_y eine Rolle. In diesem Fall liefert die Abschätzung nur dann annehmbare Genauigkeiten, falls das Produkt $H_1 H^2 \sigma_y^2$ nicht viel kleiner als der Elastizitätsmodul ist. Der materialabhängige Faktor ist in diesem Fall

$$c_M = (2\lambda + 2\mu)\left(\frac{1}{2\mu} + \frac{\sqrt{3}}{H_1 H^2 \sigma_y^2}\right). \quad (5.12)$$

Tabelle 5.2 zeigt für verschiedene Werte der plastischen Variablen k_1 den Spannungsfehler, die Werte für die Fehlerschätzer $\eta_{EQ,h}$, $\eta_{L,h}$, die Materialkonstante c_M sowie den Effektivitätsindex beider Fehlerabschätzungen. Abbildung 5.8 stellt den Spannungsfehler, und die Werte $c_M \eta_{EQ,h}$ sowie $c_M \eta_{L,h}$ in Abhängigkeit der Freiheitsgrade dar.

Es zeigt sich, daß der Effektivitätsindex nur für $k_1 \approx E$ einen annehmbaren Wert ergibt. Für kleine Werte von k_1 im Vergleich zu dem Elastizitätsmodul E ist der Effektivitätsindex sehr hoch und für $k_1 = 0$ wäre er unendlich. Das ist der Fall der perfekten Plastizität. Der Schätzer $\eta_{L,h}$ ist in den Experimenten immer kleiner als der Schätzer $\eta_{EQ,h}$. Die lokalen Probleme wurden dabei mit einer p-Methode gelöst, wobei der Polynomgrad als $p = 4$ gewählt wird.

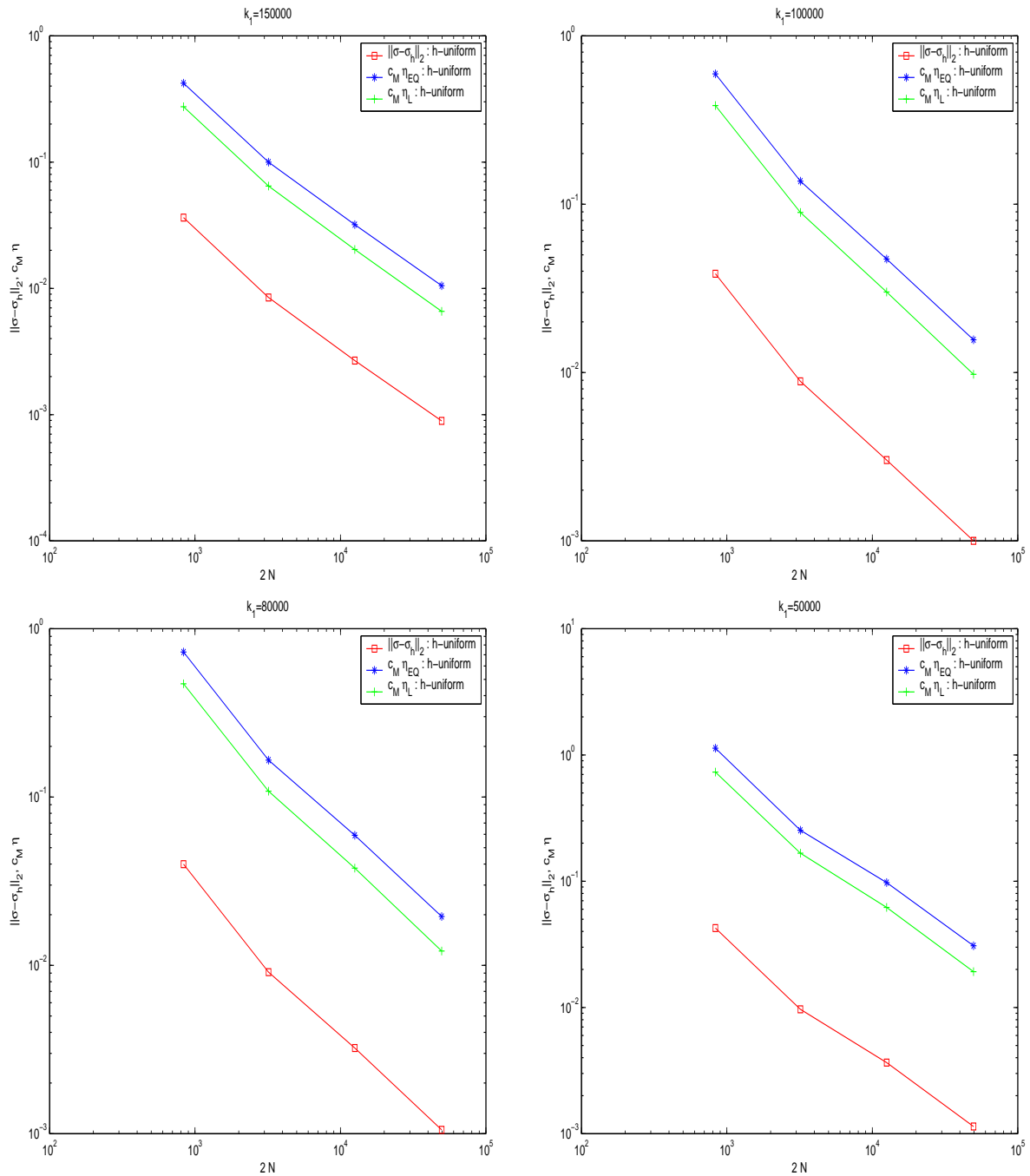


Abbildung 5.7: Fehlerabschätzungen für Beispiel 5.2 in Abhängigkeit der Freiheitsgrade für verschiedene Werte der plastischen Verfestigungsvariable k_1

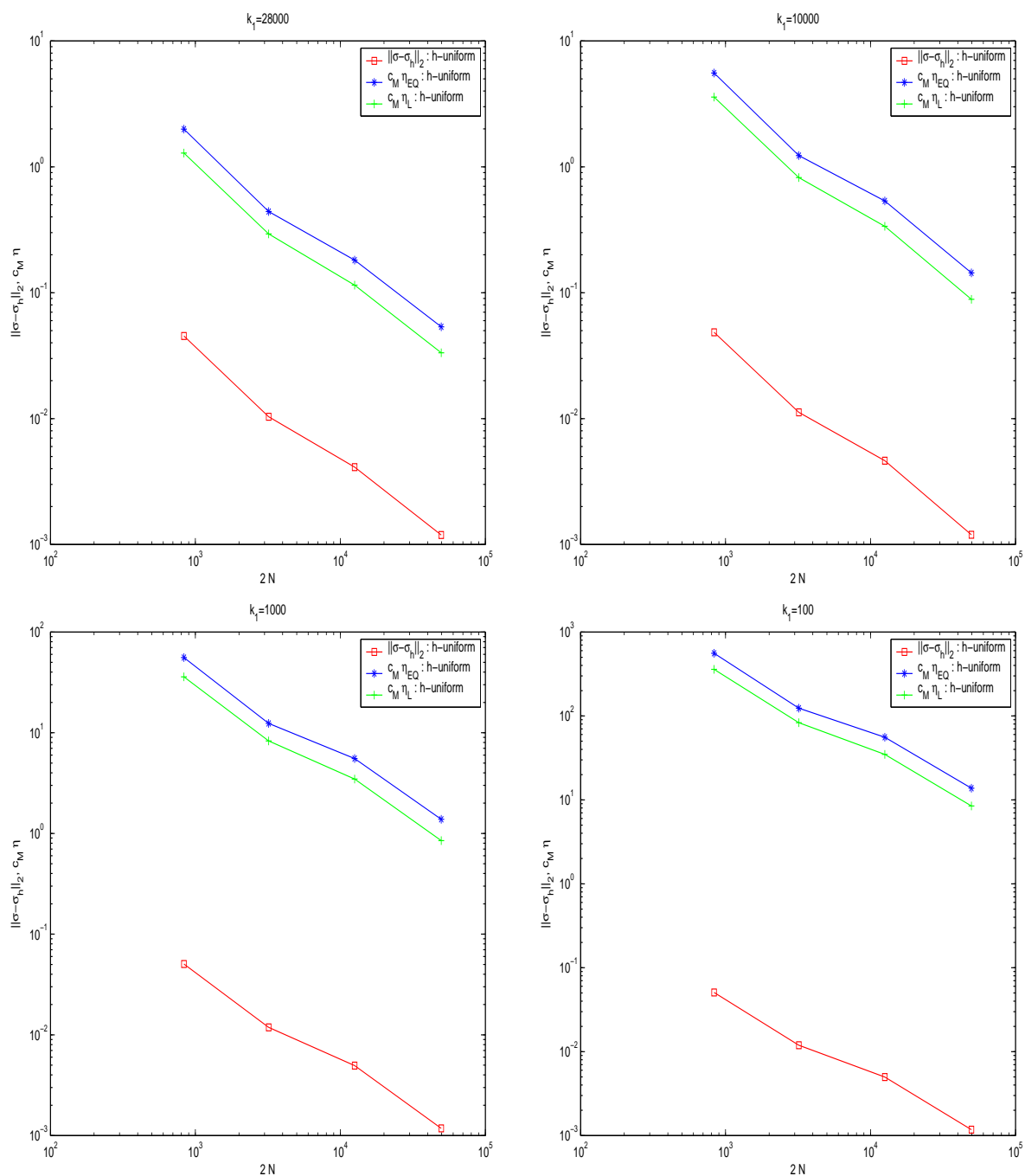


Abbildung 5.8: Fehlerabschätzungen für Beispiel 5.2 in Abhängigkeit der Freiheitsgrade für verschiedene Werte der plastischen Verfestigungsvariable k_1

k_1	$\ \boldsymbol{\sigma} - \boldsymbol{\sigma}_h\ _2$	$\eta_{EQ,h}$	$\eta_{L,h}$	c_M	$C_{Eff,1}$	$C_{Eff,2}$
150000	0.000893	0.001449	0.000907	7.2443	11.75	7.36
100000	0.001001	0.001608	0.001003	9.7301	15.63	9.75
80000	0.001055	0.001683	0.001050	11.594	18.49	11.53
50000	0.001141	0.001791	0.001117	17.188	26.97	16.82
28000	0.001188	0.001853	0.001153	28.906	45.06	28.05
10000	0.001194	0.001869	0.001152	76.847	120.3	74.14
1000	0.001179	0.001848	0.001135	748.01	1172	720.3
100	0.001177	0.001846	0.001133	7460	11695	7177

Tabelle 5.2: Effektivitätskonstanten $C_{Eff,1} = c_M \eta_{EQ,h} / \|\boldsymbol{\sigma}(t_{n-1+\theta}) - \boldsymbol{\sigma}_{h,n-1+\theta}\|_{L^2(\Omega)}$ und $C_{Eff,2} = c_M \eta_L / \|\boldsymbol{\sigma}(t_{n-1+\theta}) - \boldsymbol{\sigma}_{h,n-1+\theta}\|_{L^2(\Omega)}$ in Abhängigkeit von der plastischen Variablen k_1 in Testbeispiel 5.2. Berechnet mit einer P_2 -Finite-Elemente-Methode, 49666 Freiheitsgrade.

5.3 Experimentelle Konvergenzraten

Der in Kapitel 3 entwickelte Lösungsalgorithmus für das elastoviskoplastische Modellproblem berechnet in jedem Zeitschritt einen a posteriori Fehlerschätzer. Bei Überschreiten einer gegebenen Fehlertoleranz ist eine Verfeinerung der Ortsdiskretisierung erforderlich. Verfeinerungsindikatoren generieren adaptive Netze. Im folgenden werden verschiedene Verfeinerungsstrategien getestet.

Um Fehlerschätzer als Verfeinerungsindikatoren einzusetzen, werden ihnen elementweise Beiträge zugeordnet. In der Methode der äquilibrierten Residuen ist dieser Beitrag durch

$$\eta_{EQ,T,h} = \sqrt{a(\mathbf{y}_{T,h}, \mathbf{y}_{T,h})} \quad (5.13)$$

nach Definition 4.40 gegeben. Bei $\mathbf{y}_{T,h}$ handelt es sich um die Lösung des endlichdimensionalen Problems in Definition 4.39.

In der Patch-Residuen-Methode müssen knotenweise Beiträge auf elementweise Beiträge umgerechnet werden. Die elementweisen Fehlerindikatoren sind nach Definition 4.49 gegeben durch

$$\eta_{L,T,h} = \frac{1}{3} \sum_{z \in \mathcal{N} \cap T} \sqrt{a_z(\mathbf{y}_{z,h}, \mathbf{y}_{z,h})}. \quad (5.14)$$

Bei $\mathbf{y}_{z,h}$ handelt es sich um die Lösung des endlichdimensionalen Problems in Definition 4.49.

Die elementweisen Beiträge des Residuenschätzers werden mit Definition 4.66 berechnet,

$$\eta_{R,T,h} = (\eta_T^2 + \sum_{\substack{E \in \mathcal{E} \\ E \subset \partial T \cap \Gamma_N}} \eta_E^2 + \frac{1}{2} \sum_{\substack{E \in \mathcal{E} \\ E \subset \partial T \cap \Gamma}} \eta_E^2)^{1/2}. \quad (5.15)$$

Die lokalen Probleme in der Methode der äquilibrierten Residuen nach Definition 4.39 und in der Methode der Patch-Residuen nach Definition 4.49 werden

mit einer p -Methode gelöst, wobei der Polynomgrad $p + 2$ beträgt, wenn p der Polynomgrad der Finite-Elemente-Methode zur Lösung des Modellproblems ist. Im Lösungsalgorithmus und in den impliziten Fehlerschätzern sind dieselben Quadraturformeln zu verwenden, da sonst in den lokalen Problemen der Fehlerschätzer keine Lösbarkeit gegeben ist. Da in den lokalen Problemen ein höherer Polynomgrad verwendet wird, als der maximale Polynomgrad in der Finite-Elemente-Methode zur Lösung des Modellproblems, ist die Quadraturformel dem Polynomgrad der lokalen Probleme anzupassen.

Der adaptive Algorithmus ist wie in Abschnitt 3.3 beschrieben durchgeführt worden. Als Zeitdiskretisierungsschema wurde das implizite Euler-Verfahren eingesetzt. Der gesamte zeitabhängige Algorithmus wird mehrmals durchlaufen. Bei jedem Durchlauf wird eine kleinere Toleranz für den Ortsdiskretisierungsfehler vorgegeben. Daher werden je nach Erhöhung der Genauigkeit mehr Freiheitsgrade nötig. Die dargestellten Ergebnisse für Fehlerschätzer in Abhängigkeit von Freiheitsgraden können sich bei zwei aufeinanderfolgenden Genauigkeiten um mehr als eine Verfeinerungsstufe voneinander unterscheiden.

Eine Steuerung der Zeitschrittweite kann über die Anzahl der benötigten Iterationsschritte im Newton-Verfahren erfolgen. Dies setzt im Gegensatz zu dem in Algorithmus 3.8 beschriebenen Verfahren eine andere Abbruchbedingung voraus. Die Iteration muß dann abbrechen, wenn die Norm des Residuums kleiner als eine vorgegebene Schranke ist oder wenn der Betrag der Differenz von neuem und altem Residuum kleiner als eine vorgegebene Schranke ist. Der Startwert für das Newton-Verfahren ist der Anfangswert der Verschiebung. Daher wird für eine sehr kleine Zeitschrittweite dieser Startwert so gut sein, daß im plastischen Fall nur wenige Newton-Schritte nötig sind. Legt man die Anzahl der maximal erlaubten Newton-Schritte fest und verkleinert die Zeitschrittweite, falls diese Anzahl überschritten wird, erhält man einen groben Anhaltspunkt für die erforderliche Zeitschrittweite.

Alternativ kann die Zeitschrittweite empirisch vorgegeben werden, wobei eine Erhöhung in der geforderten Genauigkeit der Ortsdiskretisierung mit einer Verkleinerung der Zeitschrittweite zu koppeln ist. In den hier dargestellten Experimenten wurde mit uniformen Zeitschrittweiten gerechnet, die bei einer Verringerung der Fehlertoleranz für den Ortsdiskretisierungsfehler halbiert worden sind. Die größte Zeitschrittweite ist ein Viertel der Länge des Zeitintervalls.

In jedem Zeitschritt wird einer der Fehlerschätzer $\eta_{EQ,h}$, $\eta_{L,h}$, η_R zur Kontrolle des Ortsdiskretisierungsfehlers eingesetzt. Die Konstanten des Residuenschätzers wurden als 1 festgelegt. Da in diesem Abschnitt Konvergenzraten untersucht werden sollen, sind absolute Größen der Fehlerschätzer nicht relevant. Wenn in einem Zeitschritt der Ortsdiskretisierungsfehler größer als eine bestimmte Toleranz ist, wird das Finite-Elemente-Netz verfeinert und der aktuelle Zeitschritt nochmals berechnet. Die numerischen Experimente haben gezeigt, daß der absolute Wert des Fehlerschätzers mit der Zeit sehr stark anwächst, wogegen ein relativer Wert $\eta/\|\sigma_h\|_2$ für elastische Zeitschritte nahezu konstant bleibt und erst in plastischen Zeitschritten ansteigt. Dieser relative Wert wurde als Kriterium benutzt, ob in einem Zeitschritt eine Ortsverfeinerung durchzuführen ist. Zur Darstellung der Konvergenzrate werden nur absolute Werte für Fehlerschätzer und Fehler verwendet. Falls eine Verfeinerung für $t > 0$ nötig ist, sind Anfangswerte auf eine feinere Triangulierung zu interpolieren,

wie in Kapitel 3 beschrieben.

In den folgenden Unterabschnitten sind Abbildungen gezeigt, in denen für die verschiedenen Testbeispiele die Werte für die Fehlerschätzer $\eta_{EQ,h}$, $\eta_{L,h}$ und η_R gegenüber der Anzahl $2N$ der Freiheitsgrade in doppelt-logarithmischem Maßstab aufgetragen sind. Dargestellt ist mit einem Marker der Wert für den Fehlerschätzer und gegebenenfalls den Spannungsfehler $\|\sigma(t_{n-1+\theta}) - \sigma_{h,n-1+\theta}\|_{L^2(\Omega)}$ aus dem letzten Zeitschritt. Zu sehen sind Steigungsdreiecke, die sich in allen Fällen auf die letzten beiden Meßpunkte beziehen. Für die Anzahl $2N$ der Freiheitsgrade gilt in zweidimensionalen Rechnungen $N \sim 1/h^2$, so daß eine Steigung $-\alpha$ als eine experimentelle Konvergenzrate 2α interpretiert wird. In den Abbildungen werden folgende Abkürzungen verwendet.

Bezeichnung	Finite-Elemente-Methode
hu1	h-uniforme Verfeinerung, P_1 -FEM
η -ha1	η -adaptive h-Verfeinerung, P_1 -FEM
hu2	h-uniforme Verfeinerung, P_2 -FEM
η -ha2	η -adaptive h-Verfeinerung, P_2 -FEM
pu	uniforme p-Verfeinerung
η -hap	η -adaptive h-Verfeinerung+ uniforme p-Verfeinerung

Beispiel 5.2: Plastischer Ring

Das Modellproblem 1 ist für die Geometrie und die Materialparameter aus Beispiel 5.2 mit verschiedenen FE-Methoden gelöst worden. Betrachtet wird das Zeitintervall $[0, 4]$. Die mit einer P_2 -FE-Methode mit 12546 Freiheitsgraden ermittelte von-Mises-Vergleichsspannung $|\text{dev}(\sigma_h - \alpha_h)|^2$ zum Zeitpunkt $t = 4.0$ ist in Abbildung 5.9 zu sehen. Ein Teil des inneren Ringgebietes ist plastifiziert.

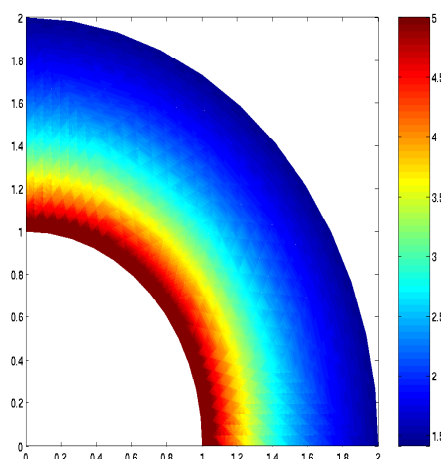


Abbildung 5.9: Darstellung der von-Mises-Vergleichsspannung in Beispiel 5.2 (Ringbeispiel) zum Zeitpunkt $t = 4$.

In Finite-Elemente-Rechnungen wird das Ringgebiet durch ein polygonales Gebiet angenähert. Bei der Näherung der exakten Lösung (5.3) entsteht ein Fehler durch die Randapproximation. Numerische Tests haben ergeben, daß die Approximation der exakten Lösung am besten gelingt, wenn die Randkräfte \mathbf{g} auf dem Neumannrand Γ_N durch die Normalenkomponente der exakten Spannung, $\mathbf{g} = \boldsymbol{\sigma} \cdot \mathbf{n}$ vorgegeben werden. Um den Einfluß von Eckensingularitäten durch die polygonale Gebietsnäherung möglichst gering zu halten, wird mit einem feinen Ausgangsnetz mit 226 Elementen gerechnet (Abb. 5.10). Nach h-Verfeinerungen wird in diesem Test keine Anpassung des Neumannrandes wie im letzten Abschnitt durchgeführt, damit im Vergleich mit der p-Methode derselbe Fehler durch die Randapproximation auftritt.

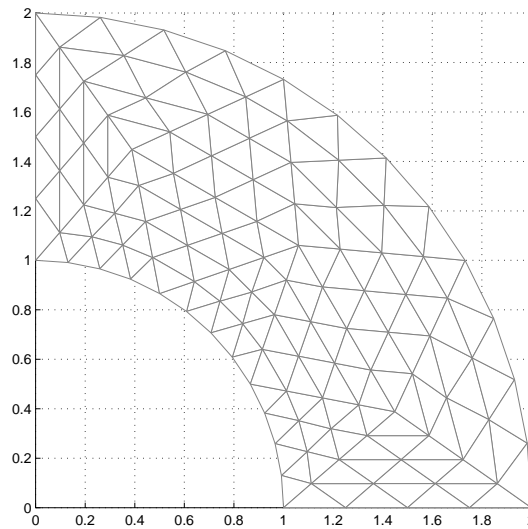


Abbildung 5.10: Ausgangsnetz \mathcal{T}_0 in Beispiel 5.2 (Plastischer Ring)

In den Abbildungen 5.11- 5.13 sind der Spannungsfehler sowie jeweils einer der Fehlerschätzer $\eta_{EQ,h}$, $\eta_{L,h}$, η_R für den letzten Zeitschritt $[t_{n-1}, t_n]$ gegenüber der Anzahl $2N$ an Freiheitsgraden im doppeltlogarithmischen Maßstab gezeigt. Grüne Linien beziehen sich auf P_1 -Elemente, rote Linien auf P_2 -Elemente. Schwarze Linien gehören zu einer p-Methode, wobei auf dem Ausgangsnetz \mathcal{T}_0 mit dem Polynomgrad $p = 2$ gestartet wurde. Die Lösung des Problems ist im Intervall $[0, 4]$ berechnet worden. Die experimentellen Konvergenzraten für die P_1 -FEM sind anhand des Schätzers $\eta_{EQ,h}$ und des Schätzers η_R ermittelt und betragen $2\alpha = 1.1$ bzw. $2\alpha = 1.04$. Die experimentellen Konvergenzraten für die P_2 -FEM sind anhand der Schätzer $\eta_{EQ,h}$, $\eta_{L,h}$ und η_R ermittelt, und betragen $2\alpha = 1.82$, $2\alpha = 1.84$ bzw. $2\alpha = 1.84$. Es zeigt sich, daß uniforme und adaptive Verfeinerung dieselben experimentellen Konvergenzraten liefern, bedingt durch die hohe Regularität des Problems. Die p-Methode hat nur im präasymptotischen Bereich eine bessere Konvergenzrate als die P_2 -Methode, konvergiert dann aber mit derselben algebraischen Konvergenzrate wie die P_2 -Methode, da durch die polygonale Gebietsnäherung Eckensingularitäten vorhanden sind.

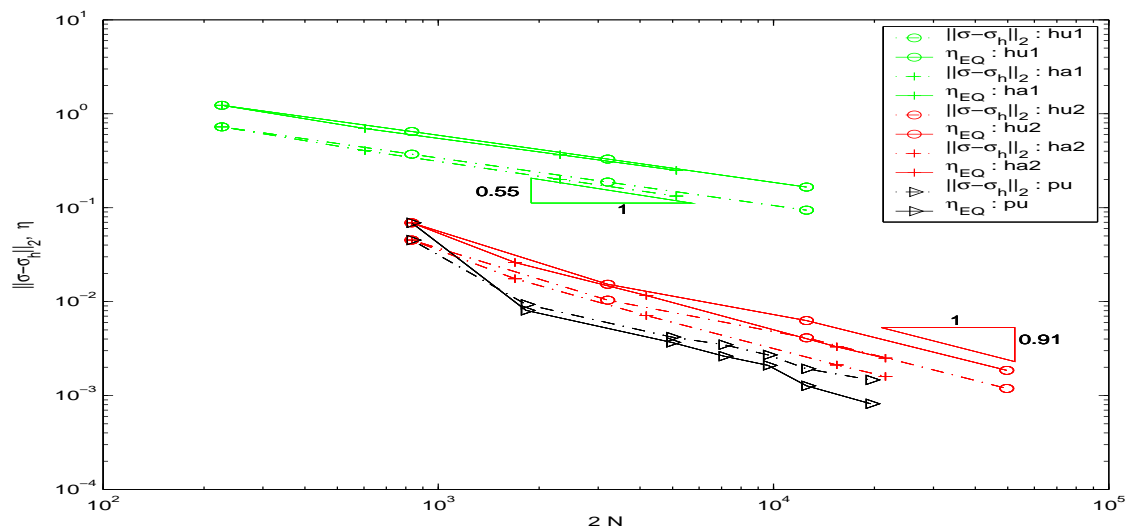


Abbildung 5.11: Fehler $\|\sigma(t_{n-1+\theta}) - \sigma_{h,n-1+\theta}\|_2$ und Fehlerschätzer $\eta_{EQ,h}$ aus dem letzten Zeitschritt für das Ring-Beispiel (Beispiel 5.2) gegenüber der Anzahl $2N$ an Freiheitsgraden.

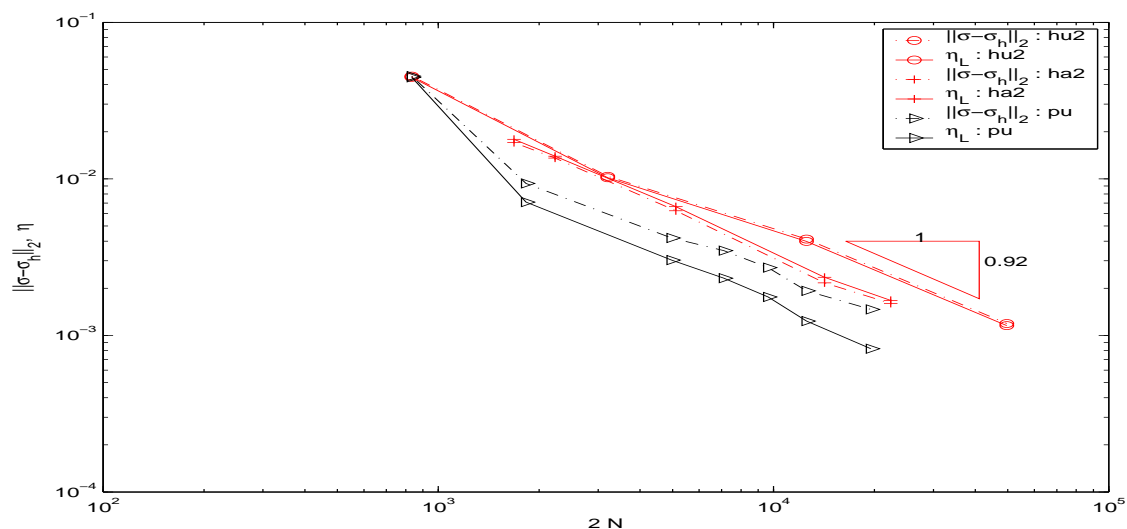


Abbildung 5.12: Fehler $\|\sigma(t_{n-1+\theta}) - \sigma_{h,n-1+\theta}\|_2$ und Fehlerschätzer $\eta_{L,h}$ aus dem letzten Zeitschritt für das Ring-Beispiel (Beispiel 5.2) gegenüber der Anzahl $2N$ an Freiheitsgraden.

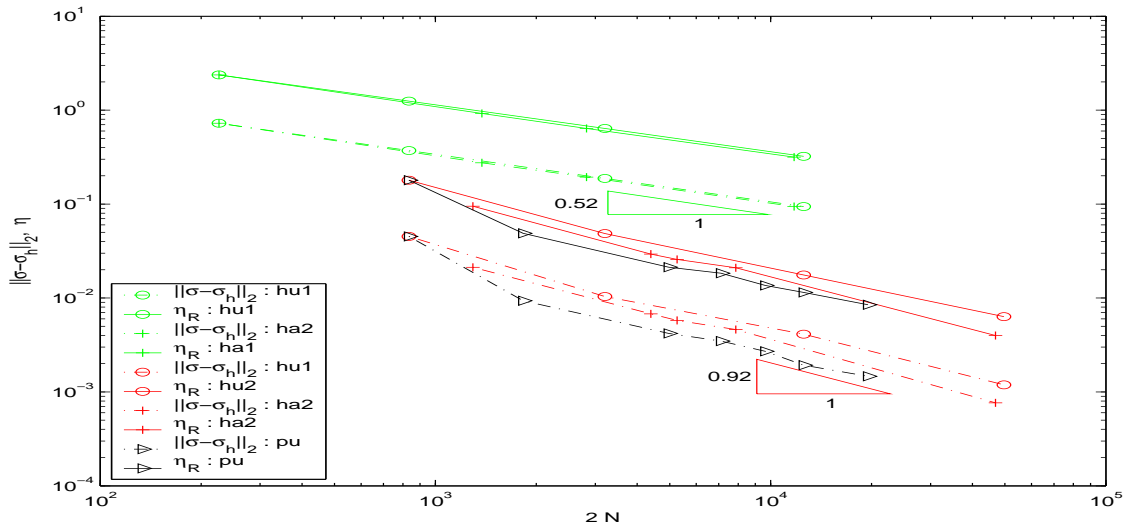


Abbildung 5.13: Fehler $\|\sigma(t_{n-1+\theta}) - \sigma_{h,n-1+\theta}\|_2$ und Fehlerschätzer η_R aus dem letzten Zeitschritt für das Ring-Beispiel (Beispiel 5.2) gegenüber der Anzahl $2N$ an Freiheitsgraden.

Beispiel 5.3: Cooks Membrane

Problem 1 ist mit einer P_2 -Finite-Elemente-Methode im Intervall $[0, 1]$ für die Geometrie und die Materialparameter aus Beispiel 5.3 gelöst worden. Die Ausgangstriangulierung hat 162 Elemente. Das Material ist um einen kleinen Bereich an einer Ecke der Auflagekante plastifiziert. Abbildung 5.14 zeigt die von-Mises-Vergleichsspannung $|\text{dev}(\sigma_h - \alpha_h)|^2$. Die Abbildungen 5.15-5.17 zeigen die Fehlerschätzer $\eta_{EQ,h}$, $\eta_{L,h}$,

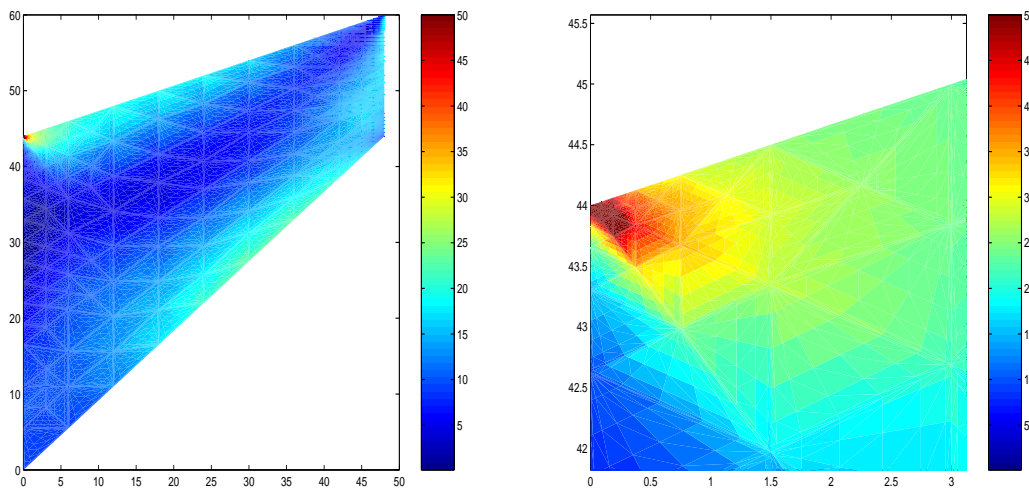


Abbildung 5.14: Darstellung der von-Mises-Vergleichsspannung in Beispiel 5.3 (Cooks Membrane) zum Zeitpunkt $t=1.0$

η_R gegenüber der Anzahl $2N$ an Freiheitsgraden für das Testbeispiel 5.3 im doppellogarithmischen Maßstab.

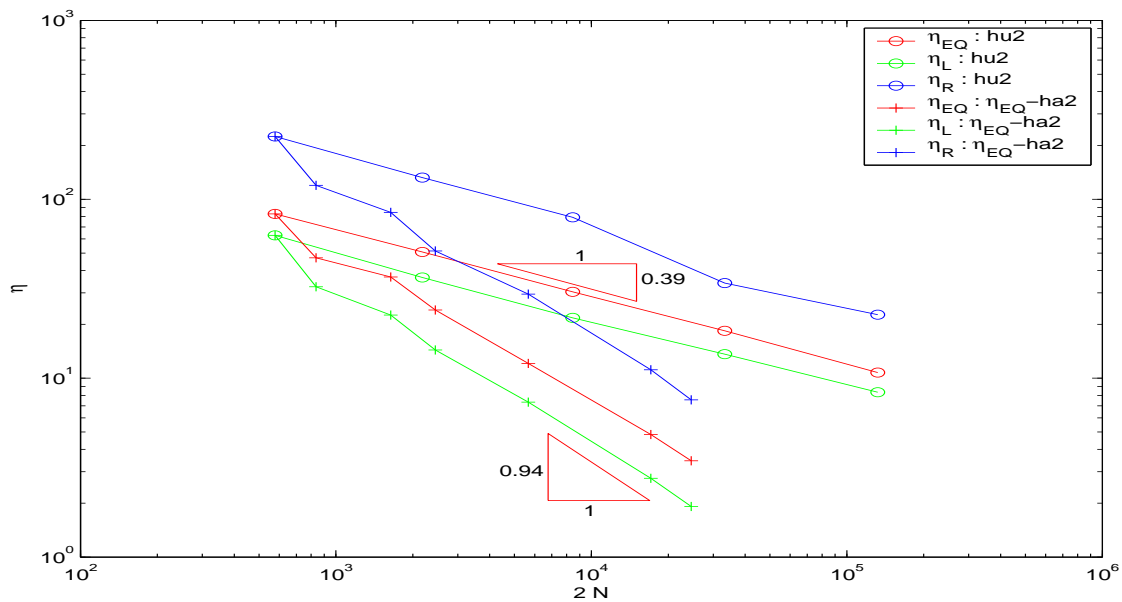


Abbildung 5.15: Verschiedene Fehlerschätzer in Beispiel 5.3 (Cooks Membrane) für eine P_2 -FEM mit uniformer und $\eta_{EQ,h}$ -adaptiver h-Verfeinerung

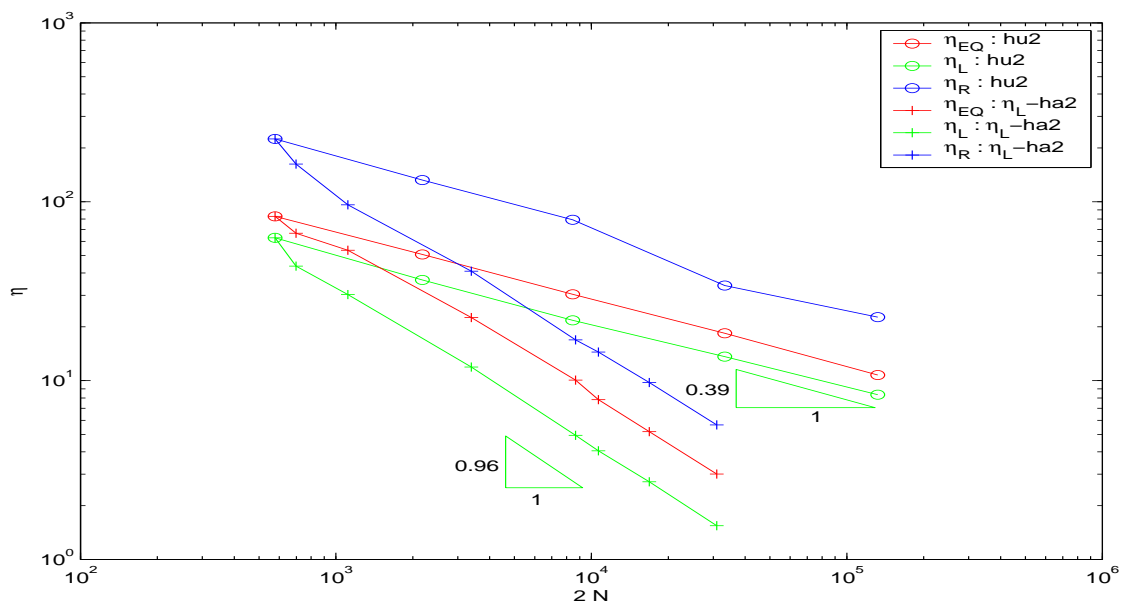


Abbildung 5.16: Verschiedene Fehlerschätzer in Beispiel 5.3 (Cooks Membrane) für eine P_2 -FEM mit uniformer und $\eta_{L,h}$ -adaptiver h-Verfeinerung.

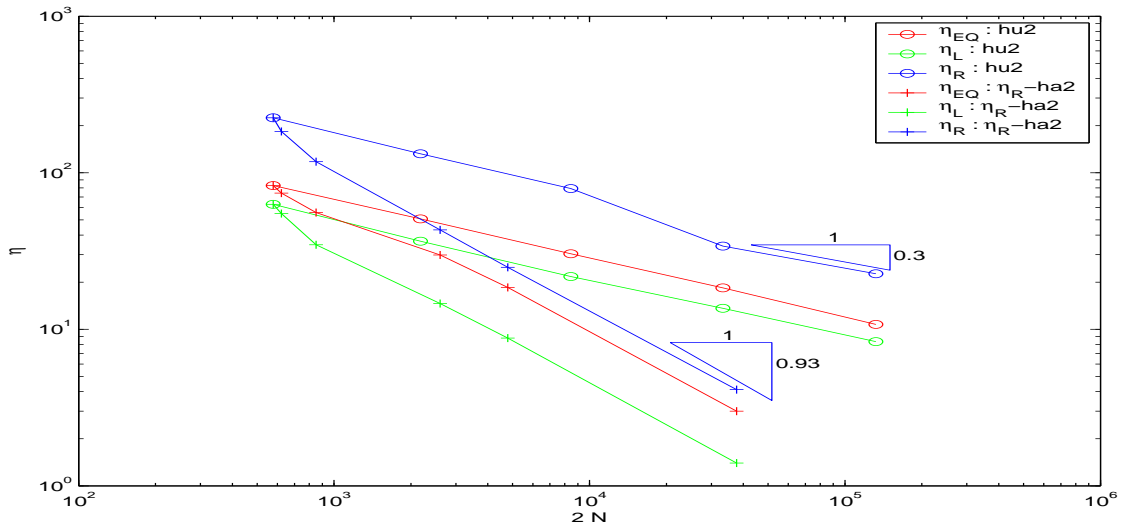


Abbildung 5.17: Verschiedene Fehlerschätzer in Beispiel 5.3 (Cooks Membrane) für eine P_2 -FEM mit uniformer und η_R -adaptiver h-Verfeinerung.

Dargestellt ist mit einem Marker der jeweilige Wert für die Fehlerschätzer aus dem letzten Zeitschritt. In Abbildung 5.15 wird der adaptive Algorithmus durch den Fehlerindikator $\eta_{EQ,T,h}$ gesteuert, in Abbildung 5.16 durch $\eta_{L,T,h}$ und in Abbildung 5.17 durch $\eta_{R,T,h}$. Die durch die Schätzer $\eta_{EQ,h}$, $\eta_{L,h}$ bzw. η_R gegebenen experimentellen Konvergenzraten betragen für uniforme Verfeinerung $2\alpha = 0.78$, $2\alpha = 0.78$ bzw. $2\alpha = 0.6$. Adaptive Algorithmen verbessern die experimentelle Konvergenzraten auf die Werte $2\alpha = 1.88$, $2\alpha = 1.92$ bzw. $2\alpha = 1.86$, bei $\eta_{EQ,T,h}$, $\eta_{L,T,h}$ bzw. $\eta_{R,T}$ -adaptiven Verfeinerungen. Alle drei Verfeinerungsindikatoren liefern im asymptotischen Bereich vergleichbare Konvergenzraten. Der Fehlerschätzer $\eta_{L,h}$ liefert in allen drei Tests kleinere Werte als der Schätzer $\eta_{EQ,h}$. Abbildung 5.22 zeigt die durch den Verfeinerungsindikator $\eta_{EQ,T,h}$ erzeugten adaptiven Netze.

In einer zweiten Testreihe ist das elastoplastische Problem 1 für die Geometrie und die Materialparameter aus Beispiel 5.3 im Zeitintervall $[0, 0.4]$ gelöst worden. Abbildung 5.18 zeigt die numerisch ermittelte von-Mises-Vergleichsspannung zum Zeitpunkt $t = 0.4$ in einem Ausschnitt um den plastifizierten Bereich. Das Material ist nur an einer kleinen Stelle an der Auflagekante plastifiziert.

Die Abbildungen 5.19- 5.21 zeigen einen Vergleich experimenteller Konvergenzraten für verschiedene Finite-Elemente-Methoden. Gezeigte Werte für Fehlerschätzer beziehen sich auf den letzten Zeitschritt. Durch grüne, rote bzw. blaue Marker und Linien sind die Ergebnisse der P_1 -FEM, P_2 -FEM bzw. der hp-FEM gekennzeichnet. Die hp-FEM ist als die Hintereinanderausführung einer adaptiven h-Methode und einer uniformen p-Methode zu verstehen. In den Abbildungen 5.19, 5.20 bzw. 5.21 sind adaptive Netze durch die Verfeinerungsindikatoren $\eta_{EQ,T,h}$, $\eta_{L,T,h}$ bzw. $\eta_{R,T}$ generiert. Es zeigt sich in den Abbildungen 5.19 und 5.21, daß sich die experimentellen Konvergenzraten bei uniformer und adaptiver h-Verfeinerung für die P_1 -Finite-Elemente-Methode nur geringfügig unterscheiden. Die experimentelle Konvergenzrate der uniformen P_1 -FEM beträgt, durch den Schätzer $\eta_{EQ,h}$ bzw. durch den Schätzer η_R ermittelt, $2\alpha = 0.92$ bzw. $2\alpha = 0.8$. Die experimentellen Konver-

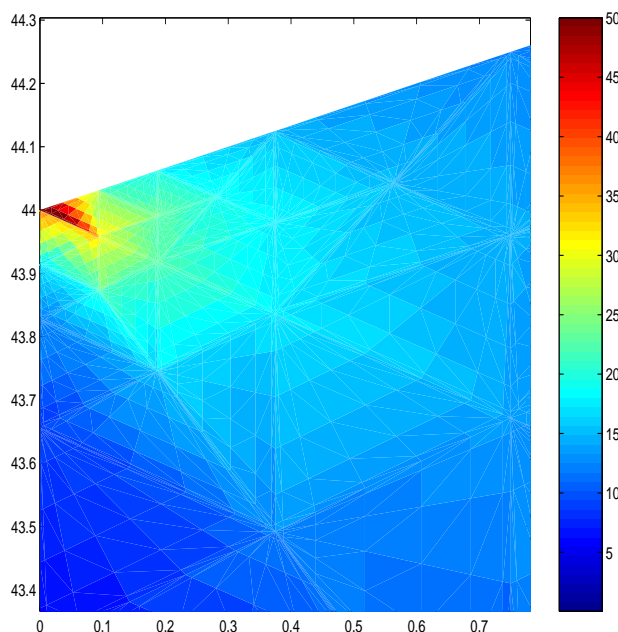


Abbildung 5.18: Darstellung der von-Mises-Vergleichsspannung in Beispiel 5.3 (Cooks Membrane) für den Zeitpunkt $t=0.4$. Zu sehen ist ein vergrößerter Bereich um die plastische Zone.

genzraten der $\eta_{EQ,h}$ -adaptiven bzw. der η_R -adaptiven P_1 -FEM betragen $2\alpha = 1.1$ bzw. $2\alpha = 0.88$. Es fällt auf, daß bei uniformer Verfeinerung eine P_2 -FEM die Konvergenzrate gegenüber einer P_1 -FEM nicht verbessern kann, sondern daß im Gegenteil die Konvergenzraten etwas schlechter sind. Durch die Schätzer $\eta_{EQ,h}$, $\eta_{L,h}$ und η_R ermittelte experimentelle Konvergenzraten betragen $2\alpha = 0.7$, $2\alpha = 0.7$ bzw. $2\alpha = 0.68$. Durch eine adaptive P_2 -FEM ergibt sich eine deutlich verbesserte Konvergenzrate. Die experimentellen Konvergenzraten der $\eta_{EQ,h}$ -adaptiven, $\eta_{L,h}$ -adaptiven bzw. der η_R -adaptiven P_2 -FEM betragen $2\alpha = 1.68$, $2\alpha = 1.9$ bzw. $2\alpha = 1.86$. Die Abbildungen 5.19- 5.21 zeigen, daß die verwendete hp-Methode besser als algebraisch konvergiert. Der Betrag der ermittelte Steigung zwischen zwei berechneten Werten für den Fehlerschätzer nimmt mit der Anzahl der Freiheitsgrade zu. In Abbildung 5.19 sind für die hp-Methode Ergebnisse mit bis zu 33970 Freiheitsgraden dargestellt. Der Betrag der Steigung zwischen Wert 7 und 8 für den Schätzer $\eta_{EQ,h}$ beträgt 2.6 und ist damit größer als der Betrag der Steigungen bei alle anderen Verfeinerungsstrategien.

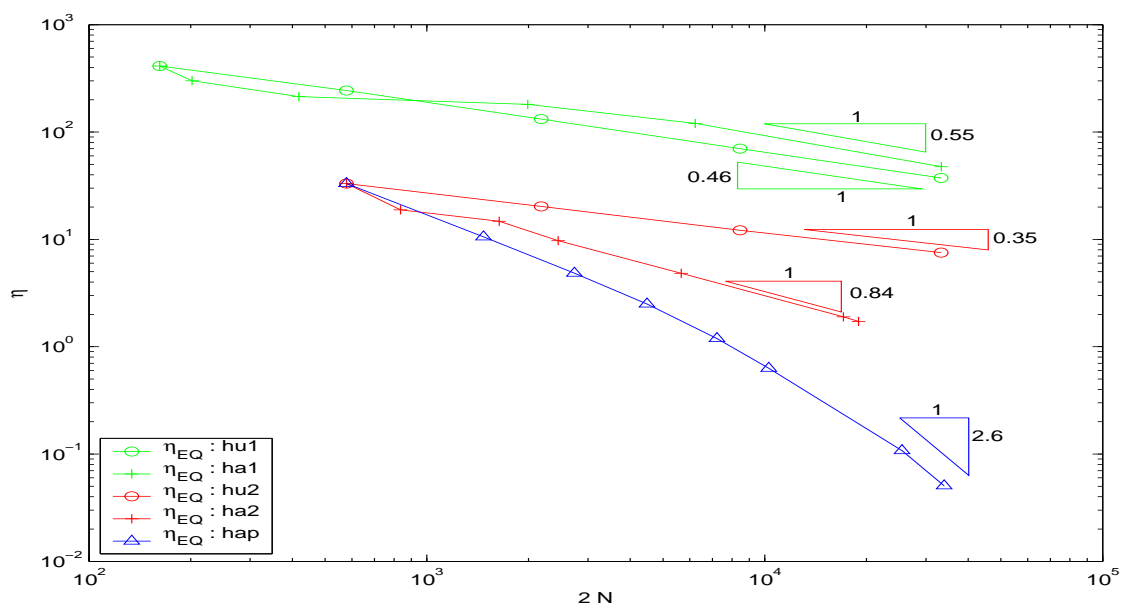


Abbildung 5.19: Fehlerschätzer $\eta_{EQ,h}$ für verschiedene FE-Methoden in Beispiel 5.3 (Cooks Membrane).

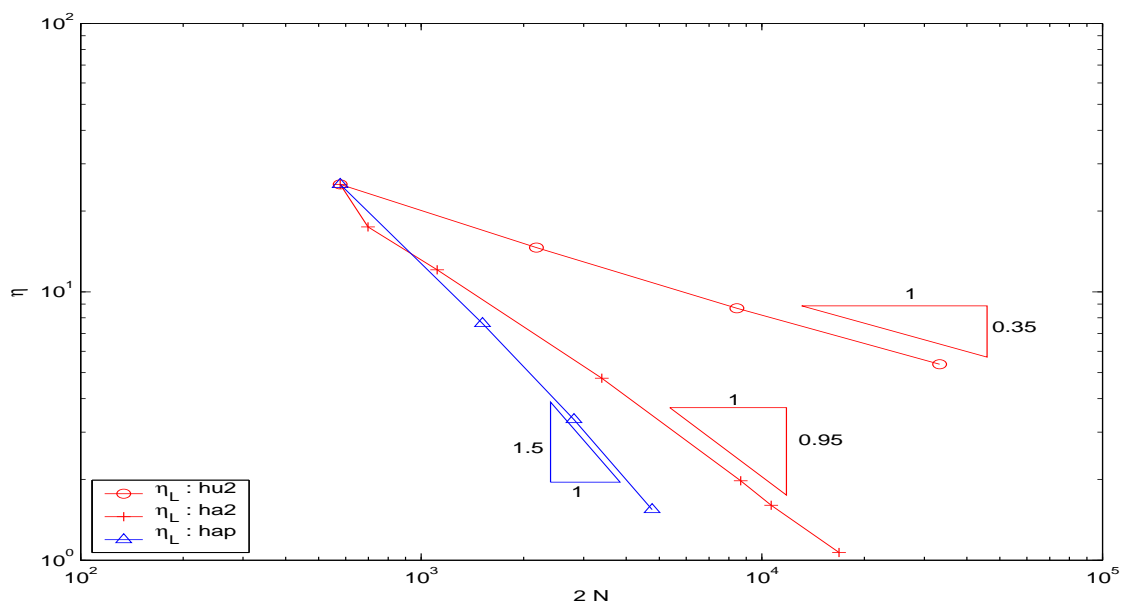


Abbildung 5.20: Fehlerschätzer $\eta_{L,h}$ für verschiedene FE-Methoden in Beispiel 5.3 (Cooks Membrane).

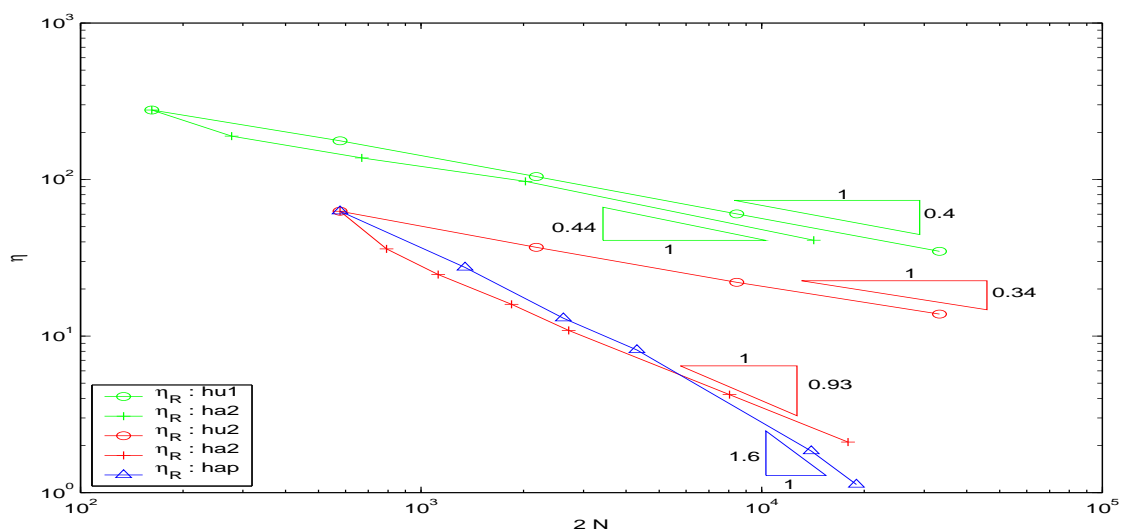


Abbildung 5.21: Fehlerschätzer η_R für verschiedene FE-Methoden in Beispiel 5.3 (Cooks Membrane).

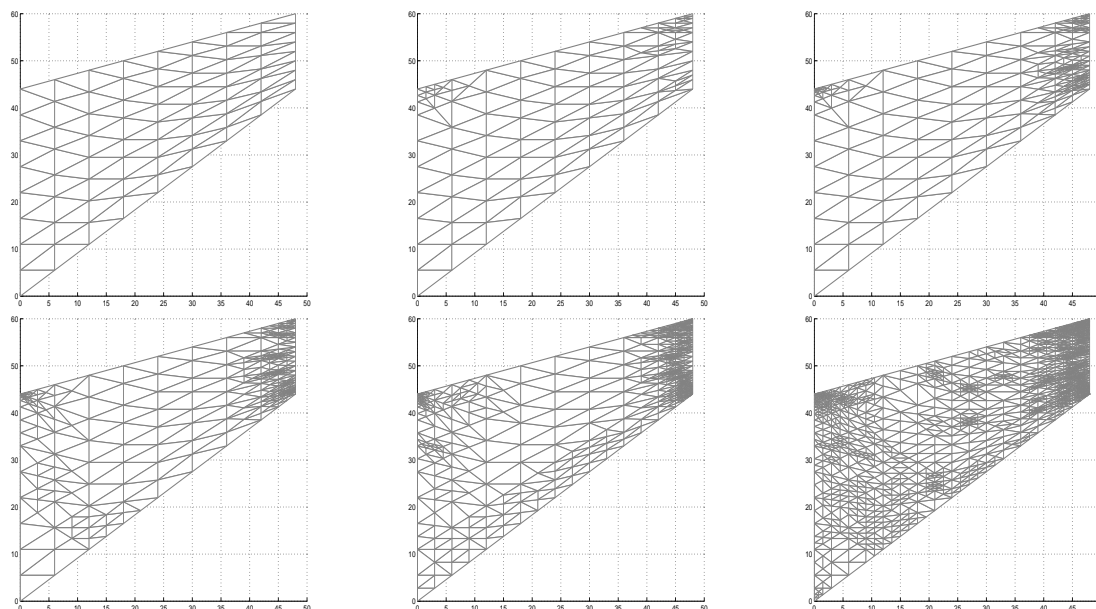


Abbildung 5.22: Adaptiv verfeinerte Netze für Beispiel 5.3. Verfeinerungsstufen 1-6.

Beispiel 5.4: L-Gebiet

Das Problem 1 ist für die Geometrie und die Materialparameter aus Beispiel 5.4 mit einer P_2 -Finite-Elemente-Methode im Zeitintervall $[0, 0.1]$ gelöst worden. Abbildung 5.23 zeigt die von-Mises-Vergleichsspannung. Das Material ist um einen kleinen Bereich an der einspringenden Ecke plastifiziert. Dort hat die Spannung einen deutlich höheren Betrag als auf dem übrigen Gebiet.

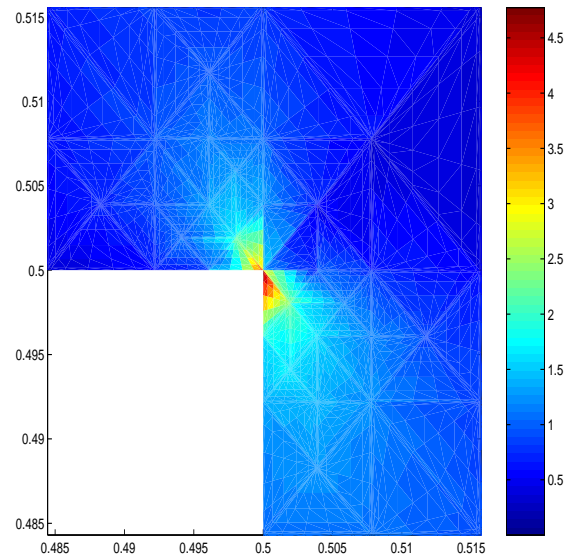


Abbildung 5.23: Darstellung der von-Mises-Vergleichsspannung für das L-Gebiet-Problem zum Zeitpunkt $t = 0.1$.

Die Abbildungen 5.25 - 5.27 zeigen Testrechnungen für das zeitabhängige Problem aus Beispiel 5.4. Für P_2 -Finite-Elemente sind die Fehlerschätzer $\eta_{EQ,h}$, $\eta_{L,h}$ und η_R im letzten Zeitschritt gegenüber der Anzahl $2N$ an Freiheitsgraden im doppeltlogarithmischen Maßstab gezeigt. Dargestellt ist mit einem Marker der jeweilige Wert für die Fehlerschätzer aus dem letzten Zeitschritt. In Abbildung 5.25 wird der adaptive Algorithmus durch den Verfeinerungsindikator $\eta_{EQ,T,h}$ gesteuert, in Abbildung 5.26 durch $\eta_{L,T,h}$ und in Abbildung 5.27 durch $\eta_{R,T,h}$. Die durch die Schätzer $\eta_{EQ,h}$, $\eta_{L,h}$ bzw. η_R gegebenen experimentellen Konvergenzraten betragen für uniforme Verfeinerung $2\alpha = 0.56$, $2\alpha = 0.56$ bzw. $2\alpha = 0.58$. Adaptive Algorithmen verbessern die experimentellen Konvergenzraten auf die Werte $2\alpha = 2.4$, $2\alpha = 1.98$ bzw. $2\alpha = 1.94$.

Ein Vergleich der Fehlerschätzer zeigt, daß die durch die Verfeinerungsindikatoren $\eta_{EQ,T,h}$, $\eta_{L,T,h}$ und $\eta_{R,T}$ generierten Netze zu nahezu identische Konvergenzraten führen. Der Schätzer $\eta_{EQ,h}$ ist um den Faktor 1.4-2.3 mal größer als der Schätzer $\eta_{L,h}$.

In den Abbildungen 5.24 ist zu erkennen, daß die durch den Verfeinerungsindikator $\eta_{EQ,T,h}$ adaptiv erzeugten Netze die einspringende Ecke auflösen. Die Verfeinerungsindikatoren erzeugen Netze, die an der einspringenden Ecke verstärkt verfeinert werden.

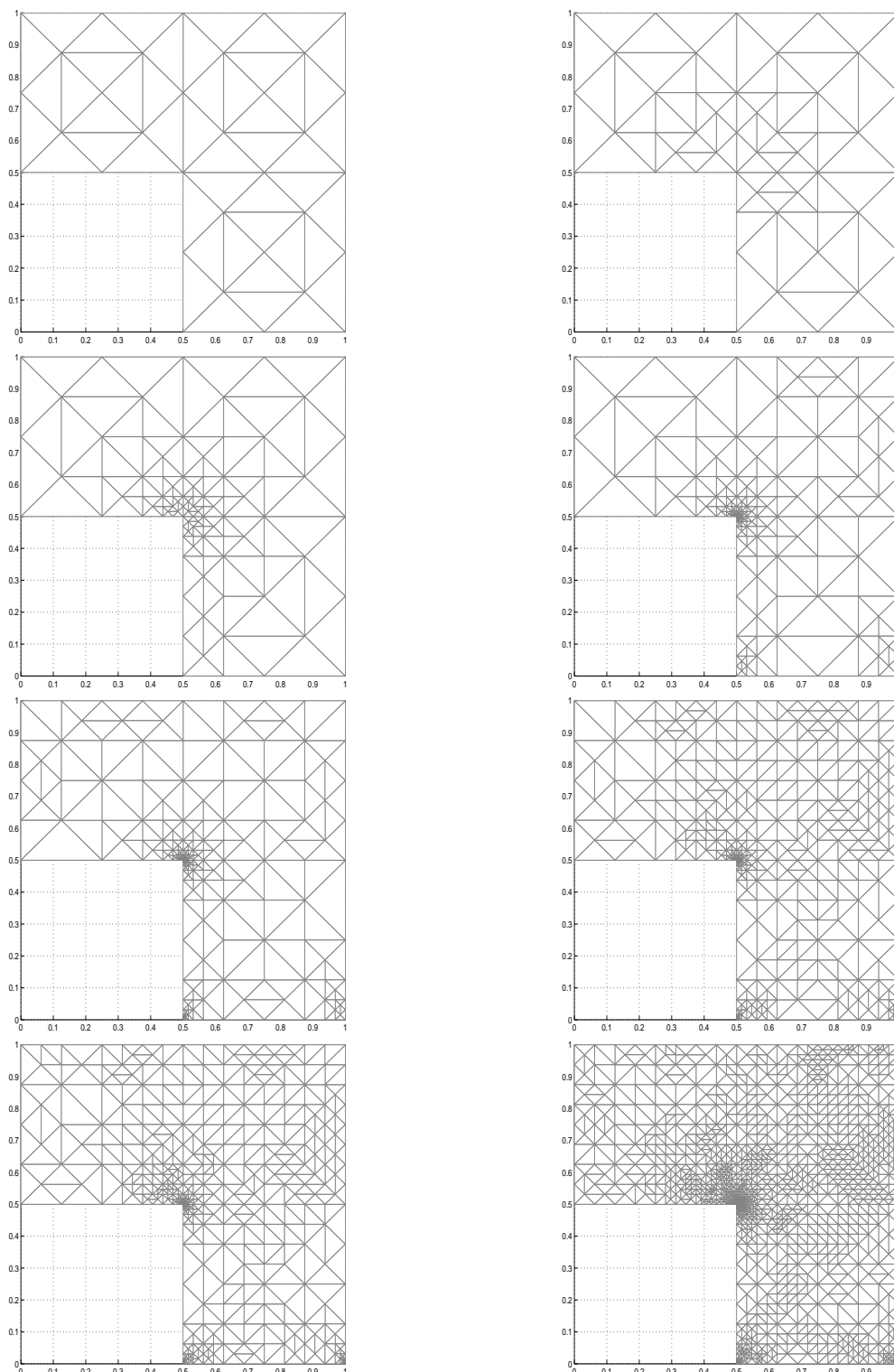


Abbildung 5.24: Adaptiv verfeinerte Netze für Beispiel 5.4. Verfeinerungsstufen 1-8.

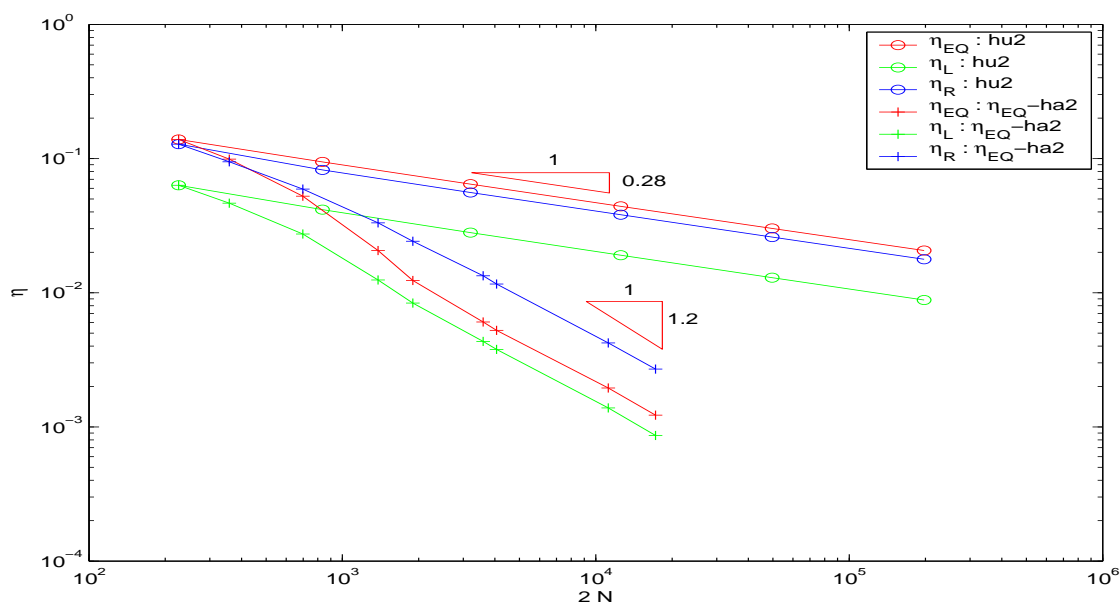


Abbildung 5.25: Verschiedene Fehlerschätzer in Beispiel 5.4 (L-Gebiet) für eine P_2 -FEM mit uniformer und $\eta_{EQ,h}$ -adaptiver h-Verfeinerung.

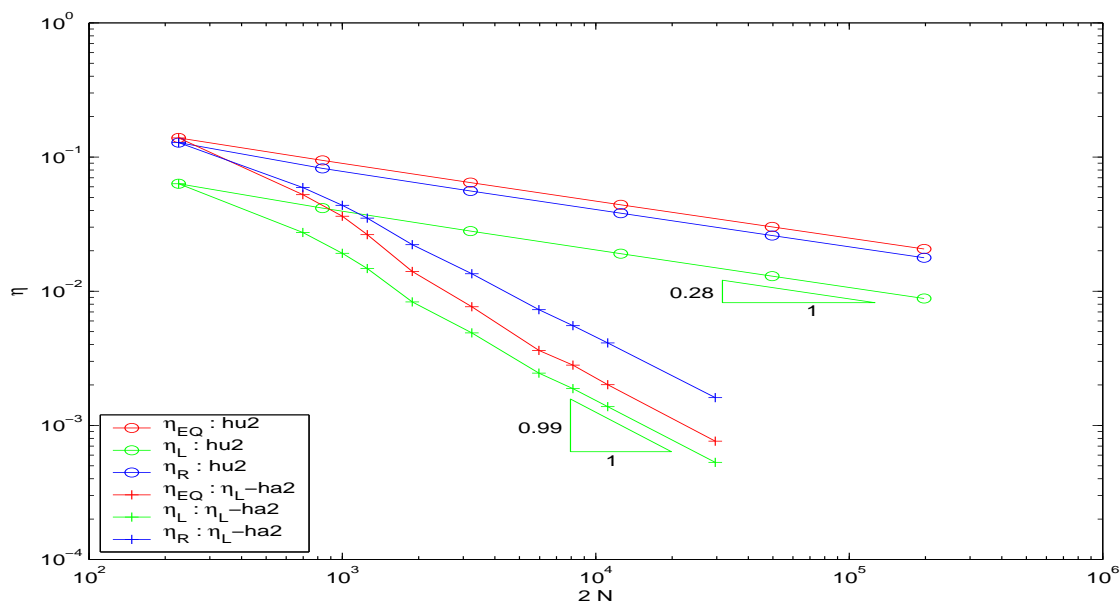


Abbildung 5.26: Verschiedene Fehlerschätzer in Beispiel 5.4 (L-Gebiet) für eine P_2 -FEM mit uniformer und $\eta_{L,h}$ -adaptiver h-Verfeinerung.

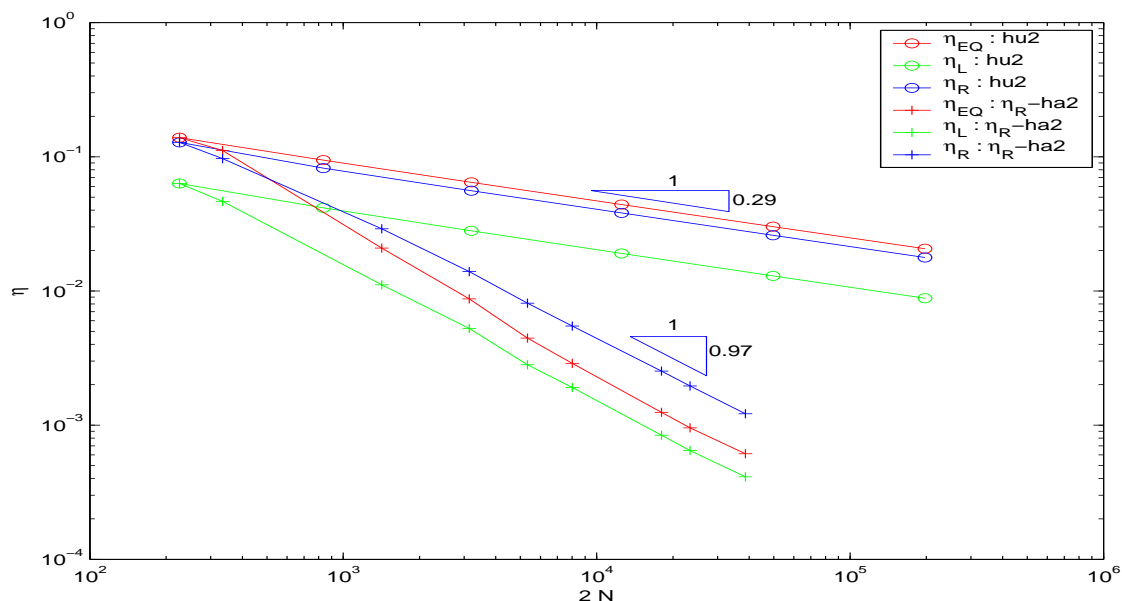


Abbildung 5.27: Verschiedene Fehlerschätzer in Beispiel 5.4 (L-Gebiet) für eine P_2 -FEM mit uniformer und η_R -adaptiver h-Verfeinerung.

In einer weiteren Testreihe wird das elastoplastische Problem 1 für die Geometrie und die Materialparameter aus Beispiel 5.4 mit verschiedenen Finite-Elemente-Methoden gelöst.

Die Abbildungen 5.28-5.30 zeigen einen Vergleich experimenteller Konvergenzraten für verschiedene Finite-Elemente-Methoden. Die gezeigten Werte für Fehlerschätzer beziehen sich auf den letzten Zeitschritt in dem Zeitintervall $[0, 0.1]$. Durch grüne, rote bzw. blaue Marker und Linien sind die Ergebnisse der P_1 -FEM, P_2 -FEM bzw. der hp-FEM gekennzeichnet. Die in Legenden mit „hap“ bezeichnete hp-FEM ist eine Hintereinanderausführung einer adaptiven h-Methode und einer uniformen p-Methode. Zusätzlich wurde eine hp-Finite-Elemente-Methode getestet, bei der zunächst eine adaptive h-Methode durchgeführt wurde und anschließend auf den nichtverfeinerten Elementen eine p-Methode. Diese hp-Methode ist in der Legende von Abbildung 5.28 mit „hpa“ bezeichnet und ist in den Abbildungen mit hellblauer Farbe dargestellt. In den Abbildungen 5.28, 5.29 bzw. 5.30 sind adaptive Netze durch die Verfeinerungsindikatoren $\eta_{EQ,T,h}$, $\eta_{L,T,h}$ bzw. $\eta_{R,T}$ generiert. Für die P_1 -FE-Methode betragen die mit η_{EQ} bzw. η_R ermittelten experimentellen Konvergenzraten bei uniformer Verfeinerung $2\alpha = 0.6$ bzw. $2\alpha = 0.62$. Die entsprechenden adaptiven Algorithmen verbessern die experimentellen Konvergenzraten auf $2\alpha = 0.94$ bzw. $2\alpha = 0.96$. Wie schon im letzten Beispiel zeigt sich, daß sich die experimentelle Konvergenzrate durch Verwendung einer uniformen P_2 -FE-Methode nicht gegenüber der einer uniformen P_1 -FE-Methode verbessern läßt. Die experimentellen Konvergenzraten, ermittelt durch die Schätzer $\eta_{EQ,h}$, $\eta_{L,h}$ bzw. η_R betragen $2\alpha = 0.54$, $2\alpha = 0.56$ bzw. $2\alpha = 0.56$. Die adaptive P_2 -FEM erzielt deutlich verbesserte experimentelle Konvergenzraten, die im präasymptotischen Bereich sogar besser sind, als die der hp-FEM. Die experimentellen Konvergenzraten der $\eta_{EQ,h}$ -adaptiven, $\eta_{L,h}$ -adaptiven bzw. der η_R -adaptiven P_2 -FEM betragen $2\alpha = 2.2$,

$2\alpha = 1.98$ bzw. $2\alpha = 1.92$. Es zeigt sich, daß die hp-Methode im präasymptotischen Bereich eine etwas schlechtere Konvergenz aufweist als die adaptive P_2 -FE-Methode, im asymptotischen Bereich aber besser als algebraisch konvergiert. Abbildung 5.28 zeigt, daß im Bereich der gerechneten Freiheitsgrade eine hp-Methode bei der eine adaptive h-Verfeinerung mit einer uniformen p-Verfeinerung kombiniert wird („hap“), im präasymptotischen Bereich eine bessere Konvergenz aufweist als die oben beschriebene und mit „hpa“ bezeichnete zweite Möglichkeit einer Kombination aus h- und p-Methode.

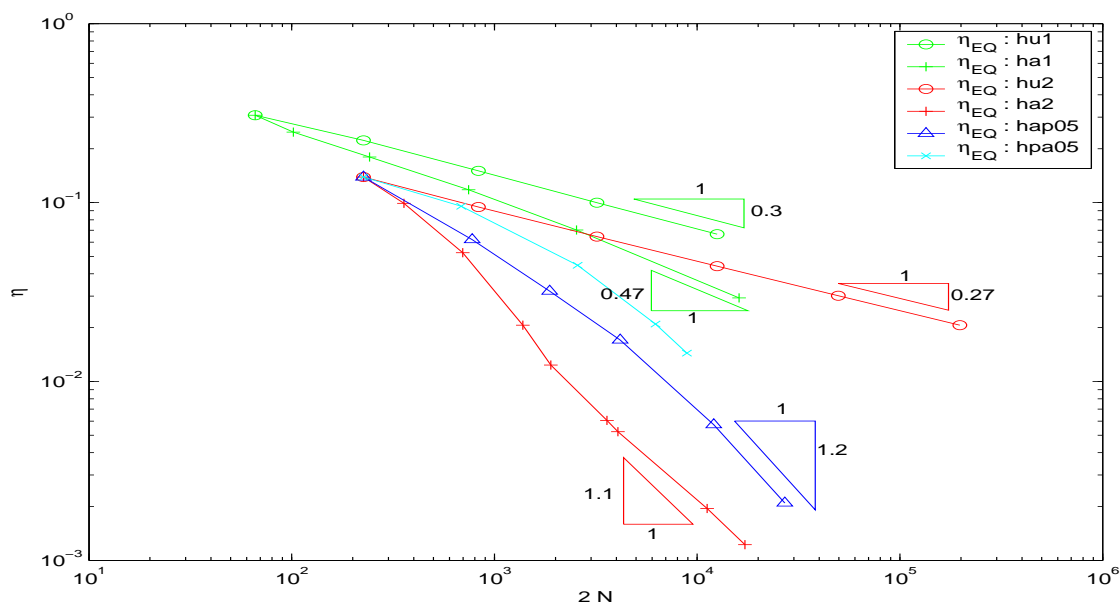


Abbildung 5.28: Fehlerschätzer $\eta_{EQ,h}$ für verschiedene FE-Methoden in Beispiel 5.4 (L-Gebiet Problem).

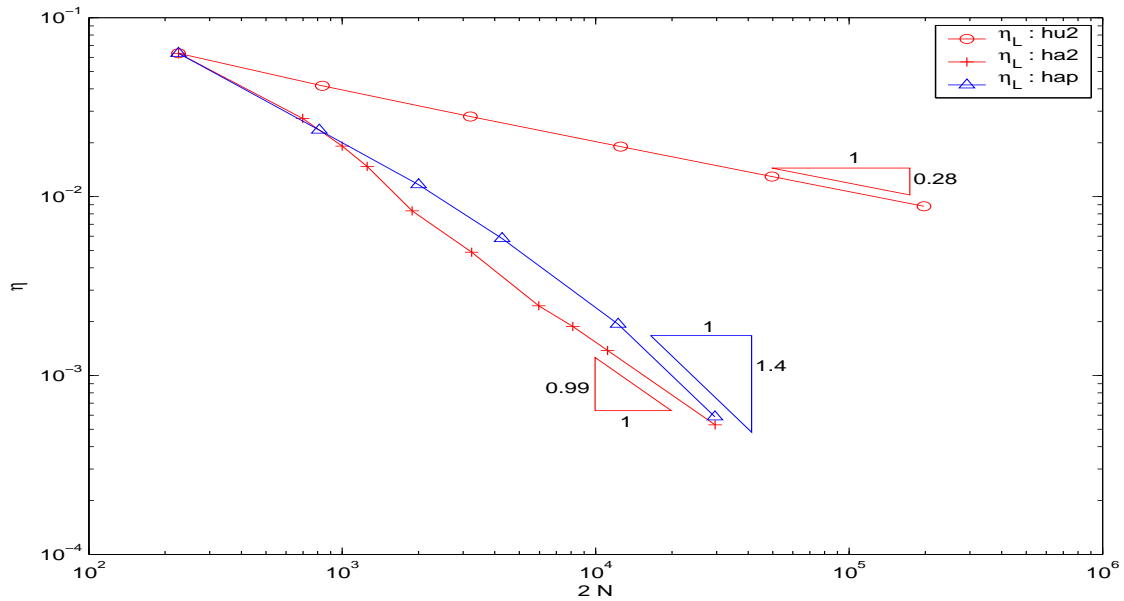


Abbildung 5.29: Fehlerschätzer $\eta_{L,h}$ für verschiedene FE-Methoden in Beispiel 5.4 (L-Gebiet Problem).

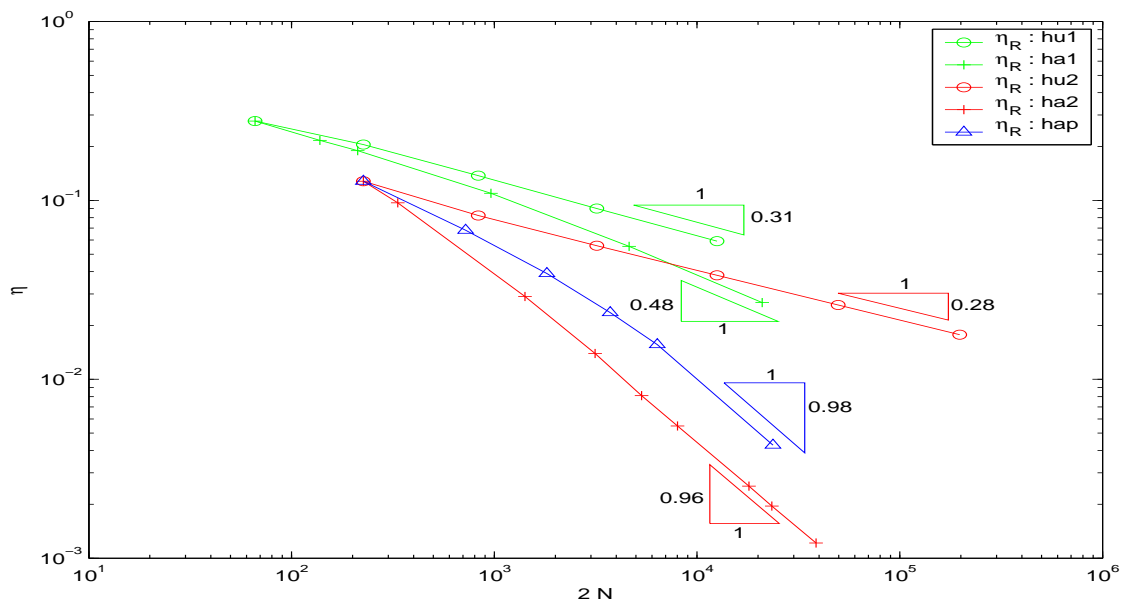


Abbildung 5.30: Fehlerschätzer η_R für verschiedene FE-Methoden in Beispiel 5.4 (L-Gebiet-Problem).

Beispiel 5.5: Lochscheibe

Das Problem 1 wird für die Geometrie und die Materialparameter aus Beispiel 5.5 für verschiedene Finite-Elemente-Methoden im Zeitintervall $[0, 1]$ gelöst. Abbildung 5.31 zeigt die numerisch ermittelte von-Mises-Vergleichsspannung zur Zeit $t = 1.0$. Das Material ist auf einem Gebiet um den Punkt $(1, 0)$ plastifiziert.

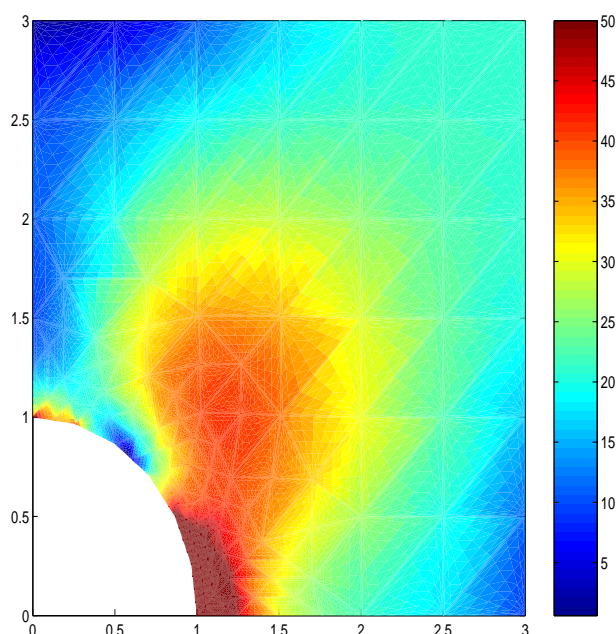


Abbildung 5.31: Darstellung der von-Mises-Vergleichsspannung für Testbeispiel 5.5 zum Zeitpunkt $t = 1.0$.

Die Abbildungen 5.32-5.34 zeigen einen Vergleich experimenteller Konvergenzraten für verschiedene Finite-Elemente-Methoden. Die gezeigten Werte beziehen sich auf den letzten Zeitschritt im Zeitintervall $[0, 1]$. Durch grüne, rote bzw. blaue Marker und Linien sind die Ergebnisse der P_1 -FEM bzw. P_2 -FEM gekennzeichnet. Die p-Methode ist durch schwarze Marker und Linien gekennzeichnet. Die hp-FEM als Hintereinanderausführung einer h-adaptiven und einer p-uniformen FE-Methode ist durch blaue Marker und Linien gekennzeichnet und in den Legenden mit „hap“ aufgeführt. Zusätzlich ist eine weitere hp-Methode getestet worden, die auch von anderen Autoren schon angewandt worden ist [39]. Dabei handelt es sich um eine Verfeinerungsstrategie, die durch zwei Parameter $\Theta_1 = 0.8$ und $\Theta_2 = 0.2$ bestimmt ist. Zunächst wird eine h-Verfeinerung auf allen Elementen durchgeführt, auf denen $\eta_T \geq \Theta_1 \max\{T' \in \mathcal{T}\} \eta_{T'}$ gilt. Anschließend wird eine p-Verfeinerung auf allen Elementen durchgeführt, auf denen $\Theta_2 \max\{T' \in \mathcal{T}\} \eta_{T'} \leq \eta_T < \Theta_1 \max\{T' \in \mathcal{T}\} \eta_{T'}$ gilt. In der Legende von Abbildung 5.32 ist diese Verfeinerung mit „hpa0802“ bezeichnet. Abbildung 5.32 zeigt, daß eine Verfeinerung nach dieser Methode ähnliche Werte ergibt wie eine Kombination aus h-adaptiver und p-uniformer Verfeinerung („hap“). Es zeigt sich, daß uniforme und adaptive P_1 -FE-Methoden nahezu dieselbe experimentelle Konvergenzrate haben. Für die uniforme

P_1 -FEM betragen die experimentellen Konvergenzraten, ermittelt durch die Schätzer $\eta_{EQ,h}$ und η_R , $2\alpha = 0.96$ bzw. $2\alpha = 0.9$. Die experimentellen Konvergenzraten der adaptiven P_1 -FEM, ermittelt durch die Schätzer $\eta_{EQ,h}$ und η_R betragen $2\alpha = 0.82$ bzw. $2\alpha = 0.92$. Wie schon in den vorhergehenden Beispielen kann durch Verwendung einer uniformen P_2 -FEM die Konvergenz gegenüber der uniformen P_1 -FEM nicht wesentlich verbessert werden. Es ergibt sich die experimentelle Konvergenzrate $2\alpha = 0.92$ für alle drei Schätzer. Durch die Verwendung einer adaptiven P_2 -FEM verbessert sich die experimentelle Konvergenzrate. Hierfür ergeben sich die Werte $2\alpha = 1.8$, $2\alpha = 1.74$ und $2\alpha = 1.72$ für die Schätzer $\eta_{EQ,h}$, $\eta_{L,h}$ bzw. η_R . Die Abbildungen 5.32-5.34 zeigen, daß die p-Methode nur im präasymptotischen Bereich schneller konvergiert als die adaptive P_2 -FE-Methode, bei höherer Anzahl an Freiheitsgraden dagegen gleich schnell. Die beste experimentelle Konvergenz zeigt die Kombination aus adaptiver h-Verfeinerung und uniformer p-Verfeinerung (blaue Linien und Marker, „hap“). Für die gerechnete Anzahl an Freiheitsgraden ist in den Abbildungen 5.32 und 5.33 nur schwach zu erkennen, daß diese Methode besser als algebraisch konvergiert. Durch η_R gesteuert, scheint sich bei dieser Methode eine algebraische Konvergenz zu ergeben (Abbildung 5.32).

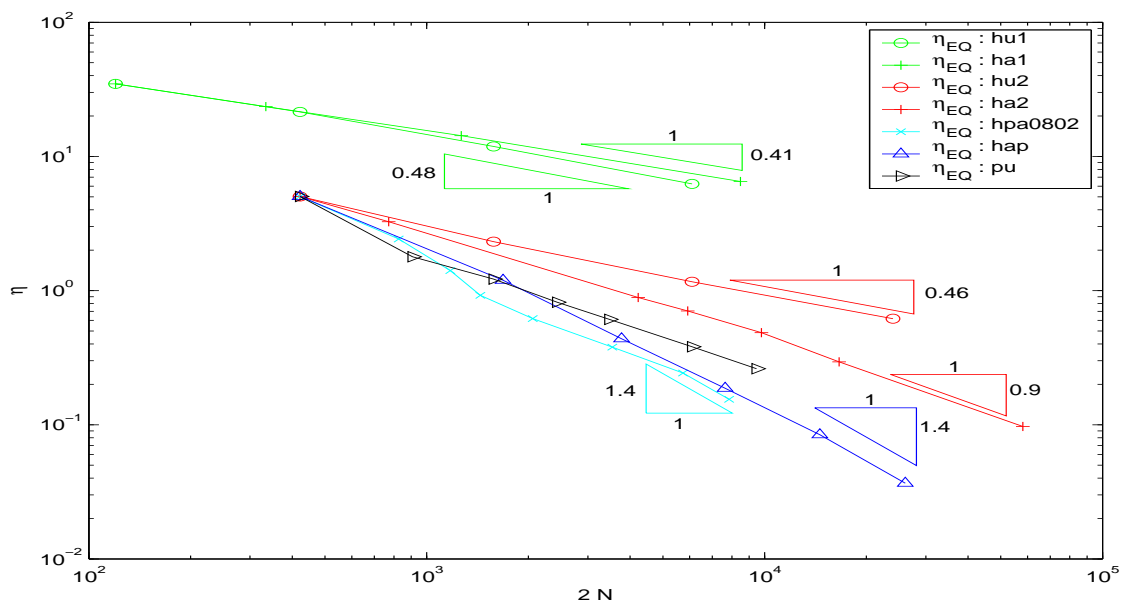


Abbildung 5.32: Fehlerschätzer $\eta_{EQ,h}$ für verschiedene FE-Methoden in Beispiel 5.5 (Lochscheibe).

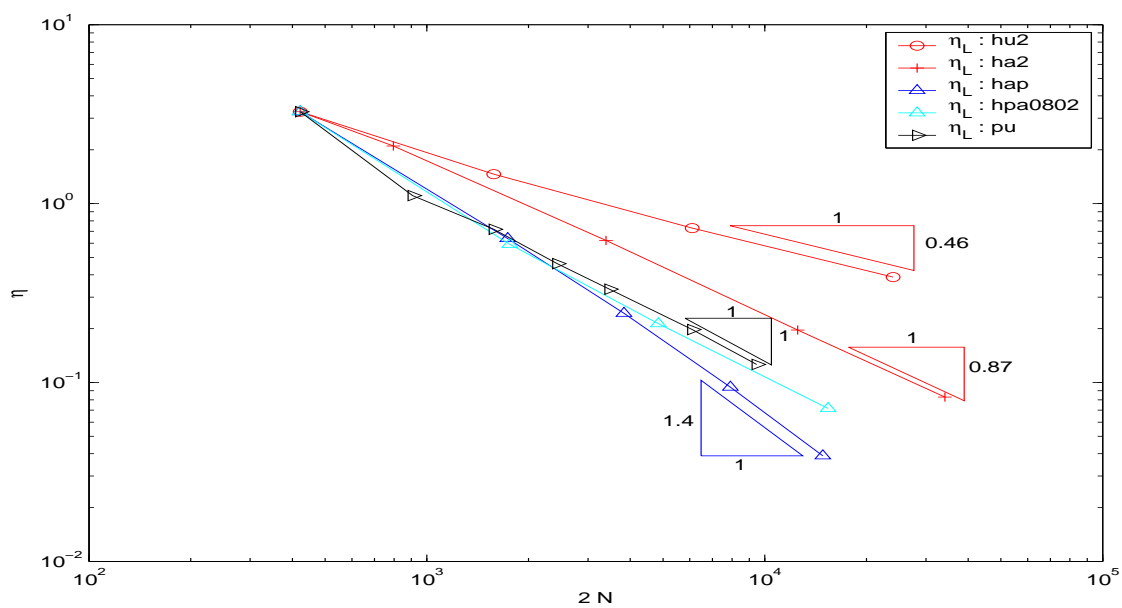


Abbildung 5.33: Fehlerschätzer $\eta_{L,h}$ für verschiedene FE-Methoden in Beispiel 5.5 (Lochscheibe).

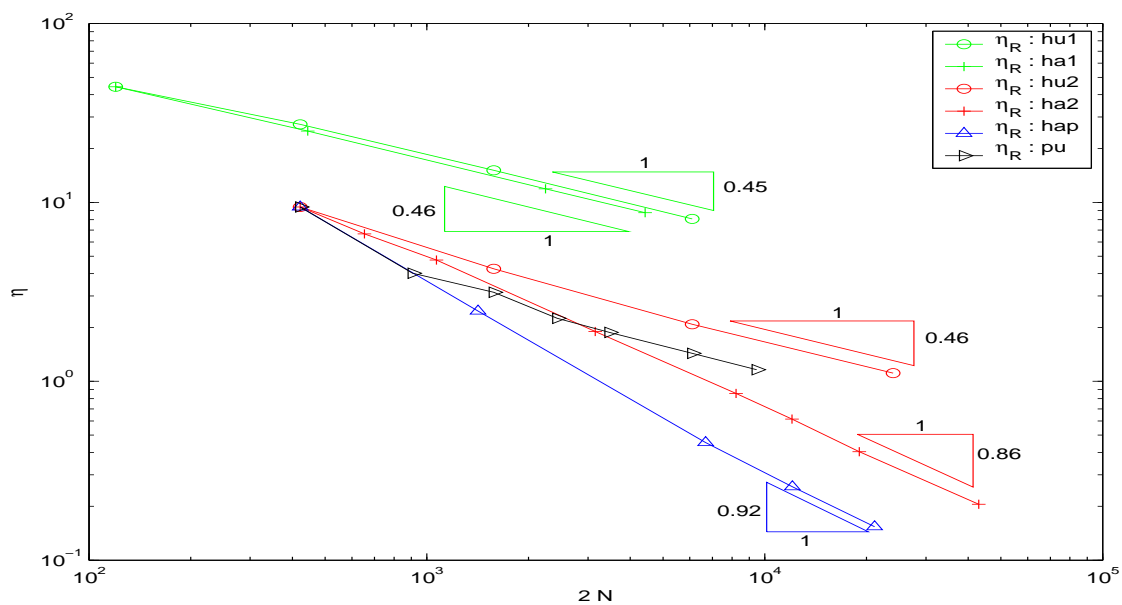


Abbildung 5.34: Fehlerschätzer η_R für verschiedene FE-Methoden in Beispiel 5.5 (Lochscheibe).

5.4 Bewertung der Resultate

In Abschnitt 5.3 wurden anhand von zweidimensionalen numerischen Testbeispielen mehrere Ergebnisse offensichtlich. In den Beispielen 5.3-5.5 haben die hier vorgestellten adaptiven Verfahren zu einer erheblichen Verbesserung der Konvergenzrate gegenüber den uniformen Verfahren geführt. Außerdem erreichten Finite-Elemente-Methoden höherer Ordnung gegenüber der P_1 -FE-Methode verbesserte experimentelle Konvergenzraten. Es zeigte sich der Vorteil der P_2 -FE-Methode gegenüber der P_1 -FE-Methode aber nur in Verbindung mit einer adaptiven Verfeinerung. Eine hp-Methode, die hier aus einer Kombination aus adaptiver h-Methode und uniformer p-Methode besteht, konvergierte besser als algebraisch.

Die Konvergenzrate allein kann aber nicht für die Bewertung einer bestimmten Verfeinerungsstrategie ausreichen. Erforderlich ist die Berücksichtigung der benötigten Rechenzeit.

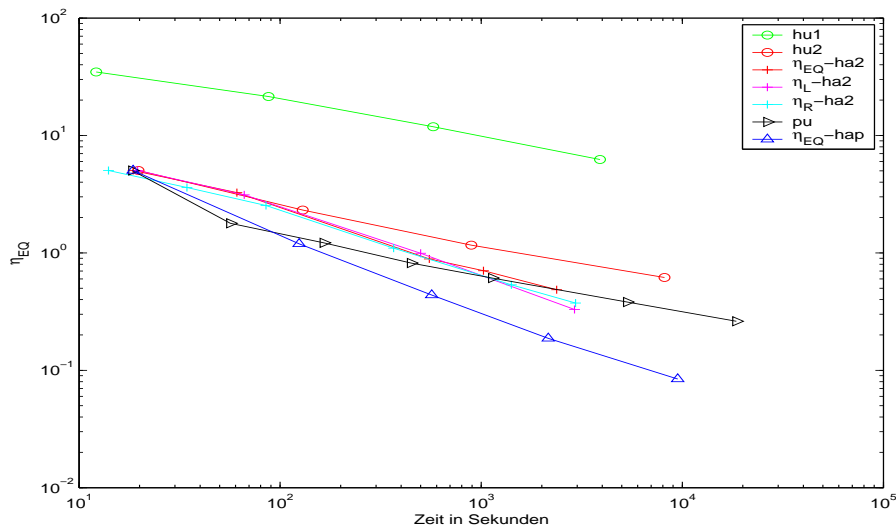


Abbildung 5.35: Fehlerschätzer $\eta_{EQ,h}$ in Abhängigkeit der Rechenzeit, die sich für eine vorgegebene Genauigkeit ergibt für verschiedene Finite-Elemente-Methoden in Beispiel 5.5.

Die hier diskutierten Ergebnisse beziehen sich auf die im Anhang dargestellte Implementierung. Abb. 5.35 zeigt für das Beispiel 5.5 den Fehlerschätzer $\eta_{EQ,h}$ gegenüber der Gesamtrechenzeit für verschiedene Finite-Elemente-Methoden. Jedes eingetragene Symbol entspricht einer dem Algorithmus vorgegebenen Genauigkeit. Bei einer Steigerung der Genauigkeit auf den nächsten Wert wurde die Zeitschrittweite halbiert. Aus der Grafik ist ersichtlich, daß die getestete hp-FE-Methode auch unter Berücksichtigung der Rechenzeit den Vorteil gegenüber den anderen Methoden innerhalb der berücksichtigten Anzahl von Freiheitsgraden behält. Der Fehlerschätzer η_{EQ} wird in der getesteten hp-FEM am schnellsten mit der Zeit kleiner. Weiterhin zeigt sich der Vorteil der P_2 -FEM gegenüber der P_1 -FEM und der Vorteil der adaptiven gegenüber der uniformen Strategie. Eine generelle Aussage über den Vorteil einzelner Verfeinerungsstrategien zu treffen ist aber schwer. Es wäre erforderlich mehrere Beispiele zu testen und mehr Freiheitsgrade zu berücksichtigen.

Es hat sich gezeigt, daß der größte Anteil der Rechenzeit für die Berechnung der lokalen Steifigkeitsmatrizen benötigt wird. Typischerweise sind hierfür bei einer P_2 -FE-Methode ca. 65% der Gesamtrechenzeit nötig, wobei die genaue Prozentzahl von der Anzahl der Freiheitsgrade abhängt. Erst bei ca. 100000 Freiheitsgraden spielt der Zeitaufwand für das Lösen des Gleichungssystems im Newton-Verfahren eine Rolle. Die Hauptrechenzeit bei den impliziten Fehlerschätzern nach der Methode der äquilibrierten Residuen und nach der Patch-Residuen-Methode wird für die Berechnung der dort auftretenden Steifigkeitsmatrizen benötigt. Der Aufwand zur Berechnung von lokalen Steifigkeitsmatrizen steigt mit der Anzahl der Quadraturpunkte und somit bei einer Erhöhung der Ordnung der Finite-Elemente-Methode.

Die in Abbildung 5.35 angegebenen Rechenzeiten ließen sich auf verschiedene Art und Weise optimieren. So ist eine Parallelisierung in dem dargestellten Algorithmus an verschiedenen Stellen möglich. Es kann die Berechnung der lokalen Steifigkeitsmatrizen parallel durchgeführt werden, ebenso wie die Lösung lokaler Probleme in den impliziten Fehlerschätzern.

Eine weitere Möglichkeit, die Implementierung von Finite-Elemente-Methoden höherer Ordnung effizienter zu gestalten, ist in [57] beschrieben. Es ist möglich, die zu Knoten und Kanten gehörenden Freiheitsgrade aus einem Gleichungssystem reduzierter Größe zu berechnen. Aus dem Ergebnis können die inneren Freiheitsgrade aus lokalen Problemen bestimmt werden, die sich wieder für eine Parallelisierung anbieten. Die Anzahl der Freiheitsgrade pro Element ist proportional zu $p_T^2 + O(p_T)$. Durch die in [57] beschriebene Methode der Reduzierung des globalen Systems auf die Knoten- und Kantenfreiheitsgrade sinkt die Anzahl der Freiheitsgrade pro Element auf $O(p_T)$.

Kapitel 6

Zusammenfassung der Ergebnisse

Die Hauptergebnisse dieser Arbeit lassen sich wie folgt zusammenfassen.

- Für das Modellproblem der regularisierten Viskoplastizität wurde ein Lösungsalgorithmus entwickelt, der auf einer analytischen Darstellung des Spannungstensors in jedem Zeitschritt einer verallgemeinerten Mittelpunktmethode beruht. Das ursprünglich spannungs- und verschiebungsabhängige Problem ist damit nur noch von der Verschiebung sowie Größen aus dem letzten Zeitschritt abhängig. Dadurch ist es möglich, den Ausdruck für die Spannung im quasi-statischen Kräftegleichgewicht zur Bestimmung der unbekanntenen Verschiebung zu verwenden. Die Lösung erfolgt mit Hilfe eines Newton-Verfahrens, wobei in jedem Newton-Schritt die Verschiebung mit einer Finite-Elemente-Methode bestimmt wird.
- Es wurde eine zuverlässige a posteriori Fehlerabschätzung für den Spannungsfehler elastoviskoplastischer Materialien mit Verfestigung in einem Zeitschritt bewiesen. Die a posteriori Fehleranalyse läßt finite Elemente höherer Ordnung zu und liefert unter Verwendung impliziter Fehlerschätzer garantierte, berechenbare obere Schranken für den Spannungsfehler. In den Fehlerabschätzungen auftretende Konstanten sind bestimmt worden. Somit stehen Fehlerschätzer zur Verfügung, die als Abbruchkriterium in Finite-Elemente-Rechnungen eingesetzt werden können.
- Numerischen Ergebnisse haben gezeigt, daß der globale Effektivitätsindex der betrachteten Fehlerabschätzungen bei Auftreten von plastischem Materialverhalten für hohe Werte der Verfestigungsvariablen akzeptable Ergebnisse liefert. Bei kleinen Werten der Verfestigungsvariable werden die getesteten Fehlerschätzer jedoch als Abbruchkriterium unbrauchbar. In weiteren numerischen Experimenten wurden anhand von zweidimensionalen elastoplastischen Problemen die Eigenschaften der Fehlerschätzer bei der adaptiven Netzgenerierung untersucht. Es wurden h- p- und hp-Verfeinerungen eingesetzt. Es hat sich der Vorteil von adaptiven Methoden sowie Finite-Elemente-Methoden höherer Ordnung bezüglich der Konvergenzrate innerhalb eines Zeitschritts gezeigt. Eine hp-Methode, die aus einer Hintereinanderausführung von adaptiver h-Verfeinerung und uniformer p-Verfeinerung besteht, konvergierte besser als algebraisch.

Anhang A

Implementierung

Dieses Kapitel beschreibt die im Rahmen dieser Arbeit entwickelten Computerprogramme zur numerischen Lösung des Modellproblems sowie zur a posteriori Fehlerabschätzung. Es soll unter anderem ermöglichen, die in vorherigen Kapiteln beschriebenen Algorithmen bestimmten Programmzeilen zuzuordnen. Die Darstellung beschränkt sich auf die wesentlichen Programmteile im Fall der kinematischen Verfestigung. Die Implementierung ist in Matlab 6.5 und C durchgeführt. Matlab ist in der Lage, C-Programme über sogenannte mex-files einzubinden. Im vorliegenden Programmpaket wurden mex-files eingesetzt bei der Berechnung von Steifigkeitsmatrizen und Vektoren der rechten Seite, bei der Spannungsberechnung und bei der Berechnung von Spannungsableitungen. Die Matlab-Programmteile basieren zum Teil auf Ideen aus [8], [28] und [18]. Zur Verwaltung der Netzgeometrie über die Dateien `Koordinaten.dat`, `Elemente.dat`, `Dirichlet.dat` und `Neumann.dat` wird auf [8] verwiesen, für Details zur Matlab-Finite-Elemente-Programmierung mit Ansatzfunktionen höherer Ordnung auf [18].

A.1 Hauptprogramm

Dieser Abschnitt beschreibt den adaptiven Algorithmus zur Lösung des Modellproblems. Die Implementierung der zugehörigen Matlab-Routine ist angegeben (Programm `start.m`).

Dem Programm werden eine Zeitschrittweite `dt`, der Parameter `tt` der θ -Regel und eine Fehlertoleranz (`toleranz`) übergeben, die in keinem Zeitschritt überschritten werden soll.

Zunächst werden Netzgeometrie, Basisfunktionen, Materialparameter und Quadraturformeln initialisiert (Zeilen 13-31).

```

1  % Adaptive 2D-FEM fuer Viskoplastizitaet
2  function [Freimax,NF,eta_N,eta_L,eta_R,RefTyp,Koordinaten,Elemente,u,p]=...
3  start(toleranz,tt,dt,ep)
4  format long
5  global Koordinaten Elemente Dirichlet Neumann p N dof Kantennr ...
6  Aussenkanten NJ NQ Gewichte BaryK Gewichte1 alpha1 beta1 NQ1 ...
7  P_ref basis_p basis_IP basis_P dx dy dx_K dy_K dx_P dy_P DX DY ...
8  dxx dxy dyy dxx_K dxy_K dyy_K dxx_P dxy_P dyy_P lbasis lbasis_IP ...

```

```

9          lbasis_IP_K lbasis_IP_P IPdof DOF sy lambda mu k1 eta C1 C2 C3 C5 R R0
10
11  iter=1;step=1;
12
13  [tmax, pmax, IPmax, Pmax, max_iter, RefNr, RefTyp, Frei, q_anz]=...
14  Approximation_initialisieren;
15
16  [Gewichte, BaryK, NQ, Gewichte1, alpha1, beta1, NQ1, P_ref]=...
17  Quadratur_initialisieren(q_anz);
18
19  [Koordinaten, Elemente, Dirichlet, Neumann, p, Kantennr, Aussenkanten, P, NJ]= ...
20  Netz_initialisieren(pmax, Pmax, DatenDir);
21
22  [lambda, mu, k1, C1, C5, sy, eta]=Material_initialisieren;
23
24  [dx, dy, dx_K, dy_K, dx_P, dy_P, DX, DY, dxx, dxy, dyy, dxx_K, dxy_K, dyy_K, ...
25   dxx_P, dxy_P, dyy_P, basis_p, basis_IP, basis_P, lbasis, lbasis_IP, lbasis_IP_K, ...
26   lbasis_IP_P, N, pKanten, dof, IPdof, DOF]=...
27  Basis_initialisieren(p, pmax, Pmax, IPmax, BaryK, alpha1, beta1);
28
29  [S0, A0, u0, S0_K, A0_K, S0_P, A0_P, DSOX, DSOY, DAOX, DAOY, DSOX_K, DSOY_K, DAOX_K, DAOY_K, ...
30   DSOX_P, DSOY_P, DAOX_P, DAOY_P, phase0, phase00, t0, t1, t]=...
31  Eulerverfahren_initialisieren(size(Elemente, 1), NQ, NQ1, N, tt, dt);
32
33  R=rad(t);R0=rad(t0);

```

In einer Schleife über alle Zeitschritte wird das diskrete Modellproblem (Problem 6) gelöst, in dem in jedem Zeitschritt eine Finite-Elemente-Rechnung durchgeführt wird (Zeilen 45,46). Als Ergebnis werden der Verschiebungsvektor u geliefert, die Anzahl der Newton-Iterationen nit , sowie ein Vektor $phase$, der für jedes Element die Information enthält, ob es in der plastischen oder elastischen Phase ist. Beim Übergang in die plastische Phase wird die Zeitschrittweite, abhängig von der Stufe $iter$ der Ortsverfeinerungen angepaßt.

```

34  dtadapt=0;term=0;
35  iter
36  while (term==0)
37    sprintf('Zeitschritt : %d\n ', step)
38    zeititer=0;zit=0;nit=0;
39    while zeititer==0
40      zit=zit+1
41      sprintf('t1 : %8.4f\n ', t1)
42      sprintf('Schrittweite : %12.10f\n ', dt)
43      C2=K2(tt, dt); C3=K3(tt, dt);
44      sv=u0;
45      [u, nit, phase]=FEM4(Koordinaten, Elemente, Dirichlet, Neumann, Kantennr, ...
46       Aussenkanten, sv, u0, S0, A0, N, p, dof, t, t0, C2, C3, pmax, NJ);
47      sprintf('Newtonschnitte : %d\n ', nit)
48      if (((max(phase)>0) & (max(phase0)==0)) & ((zit==1) & (dtadapt==0)))
49        %Zeitschrittweite beim elastisch-plastischen Uebergang anpassen
50        dt=dt/(2*iter); t1=t0+dt; t=tt*t1+(1-tt)*t0; dtadapt=1;
51      else
52        zeititer=1;
53    end

```


54 end

Abhängig vom gewählten Fehlerschätzertyp (RefNr(1)) wird ein a posteriori Fehlerschätzer aufgerufen (Zeilen 57-78). Der Residuenschätzer erfordert die Berechnung von Spannungsableitungen (Zeilen 69-72). Abhängig davon, ob der Fehlerschätzer kleiner als eine vorgegebene Toleranz ist (Zeile 82), wird zum nächsten Zeitschritt übergegangen (Zeilen 83-103) bzw. das Programm beendet oder der aktuelle Zeitschritt mit einer verfeinerten Diskretisierung erneut berechnet.

```

55 [Fehler,NM]=FSigma(u,u0,S0,A0,t,t0);
56
57 if RefNr(1)<=1
58   eta_NT=Aposteriori_EQ1(u,u0,S0,A0,S0_K,A0_K,t,t0,Pmax,Pmax);
59   eta_N=sqrt(sum(eta_NT.^2));
60   eta_l=eta_NT;
61   eta_S=sqrt(sum(eta_NT.^2)/NM)
62 end
63 if RefNr(1)==2
64   eta_LK=Aposteriori_L1(u,u0,S0,A0,t,t0,Pmax);
65   eta_L=sqrt(sum(eta_LK.^2));
66   eta_l=eta_LK;
67   eta_S=sqrt(sum(eta_LK.^2)/NM)
68 end
69 [DSX,DSY,DAX,DAY,DSX_K,DSY_K,DAX_K,DAY_K,DSX_P,DSY_P,DAX_P,DAY_P]=...
70 DSigma(u,u0,S0,A0,S0_K,A0_K,S0_P,A0_P,DSOX,DSOY,DAOX,DAOY,DSOX_K,DSOY_K,...
71   DAOX_K,DAOY_K,DSOX_P,DSOY_P,DAOX_P,DAOY_P,Koordinaten,Elemente,dof,...
72   Gewichte,NQ,NQ1);
73 if RefNr(1)==3
74   eta_RT=Aposteriori_R(u,u0,S0,A0,S0_K,A0_K,DSX,DSY,p,t,t0);
75   eta_R=sqrt(sum(eta_RT.^2));
76   eta_l=eta_RT;
77   eta_S=sqrt(sum(eta_RT.^2)/NM)
78 end
79 Frei(iter,1)=size(Koordinaten,1);Frei(iter,2)=2*max(max(abs(dof)));
80
81
82 if ((eta_S<toleranz) | (iter==max_iter))
83   step=step+1;
84   if t1+dt/2<tmax
85     t00=t0;t0=t1;t1=t1+dt;t=tt*t1+(1-tt)*t0;
86     phase00=phase0;phase0=phase;
87     R0=rad(t0);R=rad(t);
88
89     S00=S0;A00=A0;u00=u0;
90     S00_K=S0_K;A00_K=A0_K;S00_P=S0_P;A00_P=A0_P;
91
92     DS00X=DSOX;DS00Y=DSOY;DA00X=DAOX;DA00Y=DAOY;
93     DS00X_K=DSOX_K;DS00Y_K=DSOY_K;DA00X_K=DAOX_K;DA00Y_K=DAOY_K;
94     DS00X_P=DSOX_P;DS00Y_P=DSOY_P;DA00X_P=DAOX_P;DA00Y_P=DAOY_P;
95
96     [S0,A0,S0_K,A0_K,S0_P,A0_P,DSOX,DSOY,DAOX,DAOY,DSOX_K,DSOY_K,...
97     DAOX_K,DAOY_K,DSOX_P,DSOY_P,DAOX_P,DAOY_P]=...
98     QPD_updaten(u,u0,S0,A0,S0_K,A0_K,S0_P,A0_P,DSX,DSY,DAX,DAY,DSX_K,DSY_K,...
99     DAX_K,DAY_K,DSX_P,DSY_P,DAX_P,DAY_P,DSOX,DSOY,DAOX,DAOY,...
100    DSOX_K,DSOY_K,DAOX_K,DAOY_K,DSOX_P,DSOY_P,DAOX_P,DAOY_P,tt,...

```

```

101         Koordinaten,Elemente,dof,NQ,NQ1);
102     u1=1/tt*(u+(tt-1)*u0);u0=u1;
103     else
104         term=1;
105     end

```

Falls der Fehlerschätzer größer als die vorgegebene Schranke ist, wird eine Verfeinerung der Ortsdiskretisierung nötig, die abhängig vom gewählten Verfeinerungstyp (RefNr(2)) erfolgt. Möglich sind h-Verfeinerung (Zeilen 110-135) und p-Verfeinerung (Zeilen 138-172) sowie eine Kombination aus beiden Verfeinerungen.

Die h-Verfeinerung des Finite-Elemente-Netzes (Zeile 110-111) erfolgt nach der Rot-Blau-Grün-Verfeinerung (s. Abschnitt 3.3.1). Benötigt werden Anfangswerte für die verfeinerte Triangulierung, die im ersten Zeitschritt exakt gegeben sind (Zeilen 116-118) und in späteren Zeitschritten durch Interpolation berechnet werden (Zeilen 120-127 und Abschnitt 3.3.2).

```

106     else
107         iter=iter+1
108         if ((RefNr(2)==1) | (RefNr(2)==3))
109             [Koordinaten_n,Elemente_n,Dirichlet,Neumann,p,VK,phase,eta_l]= ...
110             Netz_h_verfeinern(eta_l,RefNr, ...
111                 Koordinaten,Elemente,Dirichlet,Neumann,p,Kantenr,phase);
112             [Kantenr_n,Aussenkanten_n]=GeneriereKantenr(Elemente_n,Koordinaten_n);
113             [pKanten,dof_n]=erstelldof(Koordinaten_n,Elemente_n,p,Kantenr_n);
114             N=max(max(abs(dof_n)));
115             if (step==1)
116                 [SO,A0,u0,SO_K,A0_K,SO_P,A0_P,DSOX,DSOY,DAOX,DAOY,DSOX_K,DSOY_K, ...
117                 DAOX_K,DAOY_K,DSOX_P,DSOY_P,DAOX_P,DAOY_P,phase0,phase00,t0,t1,t]=...
118                 Eulerverfahren_initialisieren(size(Elemente_n,1),NQ,NQ1,N,tt,dt);
119             else
120                 [u0,S0,A0,SO_K,A0_K,SO_P,A0_P,DSOX,DSOY,DAOX,DAOY,DSOX_K,DSOY_K, ...
121                 DAOX_K,DAOY_K,DSOX_P,DSOY_P,DAOX_P,DAOY_P,u00,S00,S00_K,S00_P, ...
122                 A00,A00_K,A00_P,DS00X,DS00Y,DA00X,DA00Y,DS00X_K,DS00Y_K,DA00X_K, ...
123                 DA00Y_K,DS00X_P,DS00Y_P,DA00X_P,DA00Y_P,phase0,phase00]=...
124                 AW_interpolieren_h(VK,S00,A00,S00_K,A00_K,S00_P,A00_P,u00,u0, ...
125                 DS00X,DS00Y,DA00X,DA00Y,DS00X_K,DS00Y_K,DA00X_K,DA00Y_K,...
126                 DS00X_P,DS00Y_P,DA00X_P,DA00Y_P,Koordinaten_n,Elemente_n, ...
127                 Kantenr_n,Aussenkanten_n,N,pmax,dof_n,tt,t0,t00,phase0,phase00);
128             end
129             Koordinaten=Koordinaten_n;Elemente=Elemente_n;
130             Kantenr=Kantenr_n;Aussenkanten=Aussenkanten_n;dof=dof_n;
131             clear Koordinaten_n;clear Elemente_n;clear Kantenr_n;clear Aussenkanten_n;
132             clear dof_n;NJ=size(Elemente,1);
133             IPdof=erstelldof2(Koordinaten,Elemente,IPmax*ones(NJ,1),Kantenr);
134             D0F=erstelldof2(Koordinaten,Elemente,Pmax*ones(NJ,1),Kantenr);
135         end

```

Die p-Verfeinerung der Finite-Elemente-Triangulierung (Zeilen 137-172) erfolgt, indem der Polynomgrad der Basisfunktionen erhöht wird (Zeile 137,138), die Quadraturformeln angepaßt werden (Zeile 139-146) und neue Basisfunktionen berechnet werden (Zeilen 150-153). Für eine Verfeinerung im ersten Zeitschritt sind Anfangswerte bekannt (Zeile 155-157). Verfeinerungen in späteren Zeitschritten erfordern

eine Interpolation von Anfangswerten in den neuen Quadraturpunkten (Zeile 162-168). Die Verschiebung ist wegen des erhöhten Polynomgrades in der neuen Basis darzustellen (Zeile 159,160).

```

136   if ((RefNr(2)==2) | (RefNr(2)==3))
137       [p_n,pmax_n,Pmax,IPmax_n]=...
138       Netz_p_verfeinern(eta_l,RefNr,Elemente,p,pmax,Pmax,Kantennr,phase);
139       q_anz=q_anz+1;
140       if (q_anz>16) & (q_anz<33)
141           q_anz=32;
142       elseif (q_anz>=33) & (q_anz<=64)
143           q_anz=64;
144       elseif q_anz>64
145           q_anz=64;
146       end
147       [Gewichte_n,BaryK_n,NQ_n,Gewichte1_n,alpha1_n,beta1_n,NQ1_n,P_ref]=...
148       Quadratur_initialisieren(q_anz);
149
150       [dx,dy,dx_K,dy_K,dx_P,dy_P,DX,DY,dxx,dxy,dyy,dxx_K,dxy_K,dyy_K,dxx_P,...
151       dxy_P,dyy_P,basis_p,basis_IP,basis_P,lbasis,lbasis_IP,lbasis_IP_K,...
152       lbasis_IP_P,N_n,pKanten_n,dof_n,IPdof,DOF]=...
153       Basis_initialisieren(p_n,pmax_n,Pmax,IPmax,BaryK_n,alpha1_n,beta1_n);
154       if (step==1)
155           [SO,A0,u0,SO_K,A0_K,SO_P,A0_P,DSOX,DSOY,DAOX,DAOY,DSOX_K,DSOY_K,...
156           DAOX_K,DAOY_K,DSOX_P,DSOY_P,DAOX_P,DAOY_P,phase0,phase00,t0,t1,t]=...
157           Eulerverfahren_initialisieren(size(Elemente,1),NQ_n,NQ1_n,N_n,tt,dt);
158       else
159           [u0,u00]=U_verfeinern(u0,u00,p,pKanten,dof,p_n,pKanten_n,dof_n,N_n,...
160           size(Koordinaten,1),Kantennr);
161
162           [SO,A0,SO_K,A0_K,DSOX,DSOY,DAOX,DAOY,DSOX_K,DSOY_K,DAOX_K,DAOY_K,...
163           SO0,SO0_K,A00,A00_K,DS00X,DS00Y,DA00X,DA00Y,DS00X_K,DS00Y_K,...
164           DA00X_K,DA00Y_K,]=...
165           AW_interpolieren_p(SO0,A00,SO0_K,A00_K,SO0_P,A00_P,u00,u0,DS00X,DS00Y,...
166           DA00X,DA00Y,DS00X_K,DS00Y_K,DA00X_K,DA00Y_K,DS00X_P,DS00Y_P,...
167           DA00X_P,DA00Y_P,Gewichte_n,BaryK_n,NQ_n,alpha1_n,beta1_n,NQ1_n,...
168           pmax_n,dof_n,IPmax,tt,t0,t00,phase0,phase00);
169       end
170       p=p_n;pmax=pmax_n;pKanten=pKanten_n;dof=dof_n;clear dof_n;N=N_n;
171       clear p_Kanten_n;IPmax=IPmax_n;Gewichte=Gewichte_n;BaryK=BaryK_n;NQ=NQ_n;
172       Gewichte1=Gewichte1_n;alpha1=alpha1_n;beta1=beta1_n;NQ1=NQ1_n;
173   end
174 end
175 end

```

Im letzten Zeitschritt werden zum Vergleich weitere Fehlerschätzer aufgerufen (Zeilen 176-193).

```

176   Freimax=Frei(iter,:);NF=Fehler;
177   if RefNr(1)<=1
178       eta_LK=AposterioriL1(u,u0,SO,A0,t,t0,Pmax);
179       eta_L=sqrt(sum(eta_LK.^2));
180       eta_RT=AposterioriR(u,u0,SO,A0,SO_K,A0_K,DSX,DSY,p,t,t0);
181       eta_R=sqrt(sum(eta_RT.^2));
182   end

```

```

183 if RefNr(1)==2
184   eta_NT=Aposteriori_EQ1(u,u0,S0,A0,S0_K,A0_K,t,t0,pmax,Pmax);
185   eta_N=sqrt(sum(eta_NT.^2));
186   eta_RT=AposterioriR(u,u0,S0,A0,S0_K,A0_K,DSX,DSY,p,t,t0);
187   eta_R=sqrt(sum(eta_RT.^2));
188 end
189 if RefNr(1)==3
190   eta_NT=Aposteriori_EQ1(u,u0,S0,A0,S0_K,A0_K,t,t0,pmax,Pmax);
191   eta_N=sqrt(sum(eta_NT.^2));
192   eta_LK=AposterioriL1(u,u0,S0,A0,t,t0,Pmax);
193   eta_L=sqrt(sum(eta_LK.^2));
194 end

```

A.2 Finite-Elemente-Methode

Die Finite-Elemente-Methode ist nach dem Algorithmus in Abschnitt 3.2.4 implementiert und baut auf die Arbeit [18] von C. Boecker für elastische Probleme auf.

```

1 function [u,nit,phase]=FEM4(Koordinaten,Elemente,Dirichlet, Neumann, ...
2                             Kantennr, Aussenkanten,u,u0,S0,A0,N,p,dof, ...
3                             t,t0,C2,C3,pmax,NJ)
4 global NQ Gewichte BaryK NQ1 Gewichte1 alpha1 beta1 P_ref ...
5         dx dy basis_p lambda mu C1 sy
6
7 nit=0;phase=zeros(NJ,1);dim=zeros(NJ,1);
8 F=zeros(2*N,1);Q=zeros(2*N,1);P=zeros(2*N,1);
9 for j=1:size(Elemente,1)
10  jdof=dof(j,:);nz=find(jdof);
11  basis_ref=(sign(jdof(nz))'*ones(NQ,1))'*basis_p(nz,:);
12  lokKoordinaten=Koordinaten(Elemente(j,:),:);
13  f_wert=f(BaryK'*lokKoordinaten,t);
14  T=det([1,1,1;lokKoordinaten])/2;
15  I=reshape([2*abs(jdof(nz))-1;2*abs(jdof(nz))],1,2*size(nz,2));
16  P(I)=P(I)+reshape((basis_ref*((Gewichte'*[1,1]).*f_wert)*T)',2*size(nz,2),1);
17 end
18
19 if ~isempty(Neumann)
20  Basiswerte=fwertkante(ones(1,NQ1)-2*alpha1,pmax);
21  KV=Koordinaten(Neumann(:,2),:)-Koordinaten(Neumann(:,1),:);
22  Lg=sqrt(sum(KV.*KV,2));KV=KV./[Lg,Lg];NV=KV*[0,-1;1,0];
23  for j=1:size(Neumann,1)
24    T=Aussenkanten(Neumann(j,1),Neumann(j,2));
25    Seite=Aussenkanten(Neumann(j,2),Neumann(j,1));
26    IK=[Seite,rem(Seite,3)+1,(Seite-1)*(pmax-1)+4:Seite*(pmax-1)+3];
27    nz = find(dof(T,IK));
28    g_wert=g([alpha1;beta1]*Koordinaten(Neumann(j,:),:),NV(j,:),t);
29    Kantewerte=Basiswerte*((Gewichte1'*[1,1]).*g_wert)*Lg(j,:)/2;
30    I=reshape([2*abs(dof(T,IK(nz)))-1;2*abs(dof(T,IK(nz)))],1,2*size(IK(nz),2));
31    P(I)=...
32    P(I)+reshape([1;1]*sign(dof(T,IK(nz))).*Kantewerte(nz,:)',2*size(nz,2),1);
33  end
34 end

```

Dirichletrandbedingungen und Verschiebungsrandbedingungen werden wie in Abschnitt 3.2.3 beschrieben implementiert.

```

35 Dknoten=[unique(Dirichlet)];B=sparse(2*N,2*N);
36 [M,L]=u_rand(Koordinaten(Dknoten,:));
37 for k=0:1
38   for l=0:1
39     B(2*Dknoten-1+l,2*Dknoten-1+k)=diag(M(1+l:2:size(M,1),1+k));
40   end
41 end
42 I=find(sum(B));I0=find(sum(M,2));
43 W=zeros(2*N,1);W(I)=L(I0);u(I)=L(I0);
44
45 Basiswerte=fwertkante(ones(1,size(Gewichte1,2))-2*alpha1,max(p));
46 Basiswerte=Basiswerte(3:end,:);
47 KV=Koordinaten(Dirichlet(:,2),:)-Koordinaten(Dirichlet(:,1),:);
48 Laenge=sqrt(sum((KV.*KV)')));
49 for j=1:size(Dirichlet,1)
50   T = Aussenkanten(Dirichlet(j,1),Dirichlet(j,2));
51   Seite=Aussenkanten(Dirichlet(j,2),Dirichlet(j,1));
52   I=[(Seite-1)*(max(p)-1)+4:Seite*(max(p)-1)+3];nz=find(dof(T,I));
53   if length(nz)>0
54     QuadPunkte=[alpha1;beta1]'*Koordinaten(Dirichlet(j,:),:);
55     [M,L]=u_rand(QuadPunkte);
56     V11=sum(reshape((M*[1;0]).*(M*[1;0])),2,size(Gewichte1,2)));
57     V12=sum(reshape((M*[1;0]).*(M*[0;1])),2,size(Gewichte1,2)));
58     V21=sum(reshape((M*[0;1]).*(M*[1;0])),2,size(Gewichte1,2)));
59     V22=sum(reshape((M*[0;1]).*(M*[0;1])),2,size(Gewichte1,2)));
60     LG=Laenge(j)*Gewichte1;
61     for i=1:length(nz)
62       for k=1:length(nz)
63         V=(Basiswerte(nz(i),:)*sign(dof(T,I(nz(i)))))* ...
64           (Basiswerte(nz(k),:)*sign(dof(T,I(nz(k))))));
65         B(2*abs(dof(T,I(nz(i))))-1,2*abs(dof(T,I(nz(k))))-1)=LG*(V11.*V)';
66         B(2*abs(dof(T,I(nz(i))))-1,2*abs(dof(T,I(nz(k)))))=LG*(V12.*V)';
67         B(2*abs(dof(T,I(nz(i)))) ,2*abs(dof(T,I(nz(k))))-1)=LG*(V21.*V)';
68         B(2*abs(dof(T,I(nz(i)))) ,2*abs(dof(T,I(nz(k)))))=LG*(V22.*V)';
69       end
70       W0=reshape(u_D(QuadPunkte)',2*size(Gewichte1,2),1);
71       W1=reshape([u(2*Dirichlet(j,1)-1);u(2*Dirichlet(j,1))]*alpha1+ ...
72                 [u(2*Dirichlet(j,2)-1);u(2*Dirichlet(j,2))]*beta1, ...
73                 2*size(Gewichte1,2),1);
74       W(2*abs(dof(T,I(nz(i))))-1)= ...
75         LG*(sum(reshape((M*[1;0]).*(W0-W1),2,size(Gewichte1,2)))* ...
76             (sign(dof(T,I(nz(i))))*Basiswerte(nz(i),:)))');
77       W(2*abs(dof(T,I(nz(i)))))= ...
78         LG*(sum(reshape((M*[0;1]).*(W0-W1),2,size(Gewichte1,2)))* ...
79             (sign(dof(T,I(nz(i))))*Basiswerte(nz(i),:)))');
80     end
81   end
82 end
83 freieKnoten=find(~sum(abs(B)));I=find(sum(abs(B),2));B=B(I,:);W=W(I,:);

```

Als Startvektor wird dem folgenden Newton-Verfahren der Vektor F übergeben, der abhängig ist von dem Startwert u für die Verschiebung.

```

85 for j=1:size(Elemente,1)
86     J=Funktionalmatrix(Koordinaten,Elemente,j);
87     jdof=dof(j,:);nz=find(jdof);dim(j)=size(nz,2);
88     [F0,ph]=VEKTOR3(u,u0,S0(j,:),A0(j,:),jdof,nz,dx,dy,reshape(J,1,4), ...
89         Gewichte,lambda,mu,C1,C2,C3,sy);
90     I0=reshape([2*abs(jdof(nz))-1;2*abs(jdof(nz))],1,2*size(nz,2));
91     Q(I0)=Q(I0)+F0;phase(j)=ph;
92 end
93 F=Q-P;

```

Das Newton-Verfahren (Zeile 97-137) ist nach dem Algorithmus 3.8 in Abschnitt 3.2.4 implementiert. Die Assemblierung der globalen Steifigkeitsmatrix erfolgt nach einem Verfahren von A. Hecht, das in [38] für die Matlab-Implementierung entwickelt wurde und ohne zeitintensive Speicherreorganisationen auskommt. Dabei werden die lokalen Steifigkeitsmatrizen der Dimension $2N(T) \times 2N(T)$ zunächst in einem Hilfsvektor `vektor` der Länge $4N(T)^2$ gespeichert. Die Indizes, an denen die Einträge der lokalen Steifigkeitsmatrizen in der globalen Steifigkeitsmatrix auftreten, werden in den Vektoren `I1` für die Zeilen und `I2` für die Spalten gespeichert. Der Aufbau der globalen Steifigkeitsmatrix erfolgt dann durch den einmaligen Aufruf des `sparse`-Befehls in Matlab (Zeile 116).

```

95 nF=norm(F(freieKnoten));n0=nF;n00=nF;
96
97 while (nF/n00>10^-4) | (nF<n0)
98     nit=nit+1;dim0=0;
99
100    vektor=zeros(1,4*sum(dim.^2));
101    I1=zeros(1,4*sum(dim.^2));I2=zeros(1,4*sum(dim.^2));
102    for j=1:size(Elemente,1)
103        J=Funktionalmatrix(Koordinaten,Elemente,j);
104        jdof=dof(j,:);nz=find(jdof);
105        M0=STEMAL(u,u0,S0(j,:),A0(j,:),jdof,nz,dx,dy,reshape(J,1,4), ...
106            Gewichte,lambda,mu,C1,C2,C3,sy);
107        vektor(dim0+1:dim0+4*dim(j)*dim(j))=reshape(M0,1,4*dim(j)*dim(j));
108        I0=reshape([2*abs(jdof(nz))-1;2*abs(jdof(nz))],1,2*size(nz,2));
109        I01=repmat(I0,1,2*dim(j));
110        I02=reshape(repmat(I0,2*dim(j),1),1,4*dim(j)*dim(j));
111        I1(dim0+1:dim0+4*dim(j)*dim(j))=I01;
112        I2(dim0+1:dim0+4*dim(j)*dim(j))=I02;
113        dim0=dim0+4*dim(j)*dim(j);
114    end
115
116    DF=sparse(I1,I2,vektor,2*N,2*N);clear vektor;clear I1;clear I2;
117    FE=[DF*u-F;W];DF=[DF,B';B,sparse(size(B,1),size(B,1))];

```

In jedem Iterationsschritt des Newton-Verfahrens wird ein lineares Gleichungssystem mit dem Matlab-Löser gelöst (Zeile 118).

```

118 x=DF\FE;u=x(1:2*N);
119 Q=zeros(2*N,1);
120
121 for j=1:size(Elemente,1)
122     J=Funktionalmatrix(Koordinaten,Elemente,j);
123     jdof=dof(j,:);nz=find(jdof);

```

```

124     [F0,ph]=VEKTOR3(u,u0,S0(j,:),A0(j,:),jdof,nz,dx,dy,reshape(J,1,4), ...
125                 Gewichte,lambda,mu,C1,C2,C3,sy);
126     I0=reshape([2*abs(jdof(nz))-1;2*abs(jdof(nz))],1,2*size(nz,2));
127     Q(I0)=Q(I0)+F0;phase(j)=ph;
128     end
129
130     F=Q-P;
131     n0=nF;nF=norm(F(freieKnoten))
132     if nit==20
133         disp('Fails to converge within maximum number of iterations!');
134         break
135     end
136
137     end
138     return

```

In den Routinen `STEMAL.c` und `VEKTOR3.c` sind die Steifigkeitsmatrix und der Vektor der rechten Seite für kinematische Verfestigung implementiert. Dabei ist entscheidend, daß die gleiche Numerik bei der Unterscheidung zwischen elastischem und plastischem Bereich in `STEMAL.c` und `VEKTOR3.c` verwendet wird. Ansonsten können Rundungsfehler dazu führen, daß eine Routine elastisch, die andere plastisch rechnet.

```

1  #include <stdio.h>
2  #include <math.h>
3  #include <stdlib.h>
4  #include "mex.h"
5
6  double *CreateDoubleVector(int len)
7  {
8      return malloc(len*sizeof(double));
9  }
10
11 int *CreateIntVector(int len)
12 {
13     return malloc(len*sizeof(int));
14 }
15
16 double tr(double x[4])
17 {
18     double t;
19     t=x[0]+x[3];
20     return t;
21 }
22
23 double skal(double x[4], double y[4])
24 {
25     double s;
26     s=x[0]*y[0]+x[1]*y[1]+x[2]*y[2]+x[3]*y[3];
27     return s;
28 }
29
30
31 void symgrad(double y[4], double x[4])
32 {

```

```

33   y[0]=x[0];y[1]=(x[1]+x[2])/2.0;y[2]=y[1];y[3]=x[3];
34   }
35
36   void dev(double y[4], double x[4])
37   {
38     y[0]=x[0]-(x[0]+x[3])/2.0;y[1]=x[1];y[2]=x[2];y[3]=x[3]-(x[0]+x[3])/2.0;
39   }
40
41   double norm(double x[4])
42   {
43     double n;
44     n=sqrt(pow(x[0],2)+pow(x[1],2)+pow(x[2],2)+pow(x[3],2));
45     return n;
46   }
47
48   double dsign(double x)
49   {
50     double s;
51     if (x>=0) s=1; else s=-1;
52     return s;
53   }
54
55
56   void STEMA(double M[], int n, double u[], double u0[],
57             double s0[], double a0[], double idof[], double nz[],
58             double dx[], double dy[], double jm[], double Gewichte[],
59             double *lambda, double *mu, double *c1, double *c2, double *c3,
60             double *sy, int nq, int nd, int nd0)
61   {
62     double dtm,phi[2][2],*dxz,*dyz;
63     double dux1,duy1,dux2,duy2,Du[4],Eu[4];
64     double ds[4],da[4],v0[4],dv0[4],dv1[4],dv[4];
65     double w1[2],w2[2],d1[4],d2[4],e1[4],e2[4],de1[4],de2[4];
66     double f1,f2,delta;
67     int *I0,l,i,j,k,iw1,iw2;
68
69     I0=CreateIntVector(nd);
70     dxz=CreateDoubleVector(nd0);dyz=CreateDoubleVector(nd0);
71
72     for (i=0;i<4*nd*nd;i++) M[i]=0.0;
73     dtm=jm[0]*jm[3]-jm[1]*jm[2];
74     phi[0][0]=jm[3]/dtm; phi[0][1]=-jm[2]/dtm;
75     phi[1][0]=-jm[1]/dtm; phi[1][1]=jm[0]/dtm;
76
77     for (j=0;j<nd;j++) I0[j]=abs((int) idof[(int) nz[j]-1]);
78     for (l=0;l<nq;l++)
79     {
80       for (j=0;j<nd0;j++)
81       {
82         dxz[j]=dx[j+nd0*1]*dsign(idof[j]);
83         dyz[j]=dy[j+nd0*1]*dsign(idof[j]);
84       }
85       dux1=0;duy1=0;dux2=0;duy2=0;
86       for (j=0;j<nd;j++)
87       {

```



```

88     dux1=dux1+(u[2*I0[j]-1-1]-u0[2*I0[j]-1-1])*dxz[(int) nz[j]-1];
89     duy1=duy1+(u[2*I0[j]-1-1]-u0[2*I0[j]-1-1])*dyz[(int) nz[j]-1];
90     dux2=dux2+(u[2*I0[j]-1]-u0[2*I0[j]-1])*dxz[(int) nz[j]-1];
91     duy2=duy2+(u[2*I0[j]-1]-u0[2*I0[j]-1])*dyz[(int) nz[j]-1];
92     }
93     Du[0]=dux1*phi[0][0]+duy1*phi[0][1];
94     Du[1]=dux1*phi[1][0]+duy1*phi[1][1];
95     Du[2]=dux2*phi[0][0]+duy2*phi[0][1];
96     Du[3]=dux2*phi[1][0]+duy2*phi[1][1];
97     symgrad(Eu,Du);
98     ds[0]=s0[1]-(s0[1]+s0[3*nq+1])/2.0;
99     ds[1]=s0[nq+1];
100    ds[2]=s0[2*nq+1];
101    ds[3]=s0[3*nq+1]-(s0[1]+s0[3*nq+1])/2.0;
102    da[0]=a0[1]-(a0[1]+a0[3*nq+1])/2.0;
103    da[1]=a0[nq+1];
104    da[2]=a0[2*nq+1];
105    da[3]=a0[3*nq+1]-(a0[1]+a0[3*nq+1])/2.0;
106    v0[0]=Eu[0]+(s0[1]+s0[3*nq+1])/(4.0*(lambda)+4.0*(mu))+
107          ds[0]/(2.0*(mu));
108    v0[1]=Eu[1]+ds[1]/(2.0*(mu));
109    v0[2]=Eu[2]+ds[2]/(2.0*(mu));
110    v0[3]=Eu[3]+(s0[1]+s0[3*nq+1])/(4.0*(lambda)+4.0*(mu))+
111          ds[3]/(2.0*(mu));
112    dev(dv0,v0);
113    dv[0]=dv0[0]-da[0]/(2*(mu));dv[1]=dv0[1]-da[1]/(2*(mu));
114    dv[2]=dv0[2]-da[2]/(2*(mu));dv[3]=dv0[3]-da[3]/(2*(mu));
115
116    if (norm(dv)-(*sy)/(2*(mu))>0)
117    {
118        f1>(*c2)+(*c3)/norm(dv);f2>(*c3)/pow(norm(dv),3);
119        delta=1.0;
120    }
121    else
122    {
123        f1=2*(mu);f2=0.0;delta=0.0;
124    }
125
126    for (i=0;i<2*nd;i++)
127    {
128        iw1=(int) nz[(int) floor(i/2)]-1;
129        w1[0]=dxz[iw1]*phi[0][0]+dyz[iw1]*phi[0][1];
130        w1[1]=dxz[iw1]*phi[1][0]+dyz[iw1]*phi[1][1];
131        if (fmod(i+1,2)==0)
132        {
133            d1[0]=0.0;d1[1]=0.0;d1[2]=w1[0];d1[3]=w1[1];
134        }
135        else
136        {
137            d1[0]=w1[0];d1[1]=w1[1];d1[2]=0.0;d1[3]=0.0;
138        }
139        symgrad(e1,d1);dev(de1,e1);
140        for (k=0;k<2*nd;k++)
141        {
142            iw2=(int) nz[(int) floor(k/2)]-1;

```

```

143         w2[0]=dxz[iw2]*phi[0][0]+dyz[iw2]*phi[0][1];
144         w2[1]=dxz[iw2]*phi[1][0]+dyz[iw2]*phi[1][1];
145         if (fmod(k+1,2)==0)
146             {
147                 d2[0]=0.0;d2[1]=0.0;d2[2]=w2[0];d2[3]=w2[1];
148             }
149         else
150             {
151                 d2[0]=w2[0];d2[1]=w2[1];d2[2]=0.0;d2[3]=0.0;
152             }
153         symgrad(e2,d2);dev(de2,e2);
154
155         M[k*2*nd+i]=M[k*2*nd+i]+dtm*sqrt(3.0)*Gewichte[1]*
156             ((*c1)*tr(e1)*tr(e2)+f1*skal(de1,de2)-f2*skal(dv,e2)*
157             skal(dv,e1));
158     }
159 }
160 }
161 free(I0);I0=NULL;
162 free(dxz);dxz=NULL;free(dyz);dyz=NULL;
163 }
164
165 /* Gateway function */
166
167 void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[])
168 {
169     double *M,*u,*u0,*s0,*a0,*idof,*nz,*dx,*dy,*J,*Gewichte,*lambda,*mu,
170         *c1,*c2,*c3,*sy;
171     int nd,nd0,nq,n;
172
173     /*Check for proper number of arguments */
174     if (nrhs!=16)
175         mexErrMsgTxt("16 input required!");
176     if (nlhs!=1)
177         mexErrMsgTxt("1 output required!");
178
179     n =mxGetM(prhs[0])/2; nq=mxGetN(prhs[9]);
180     nd=mxGetN(prhs[5]); nd0=mxGetN(prhs[4]);
181
182     u      =mxGetPr(prhs[0]); u0      =mxGetPr(prhs[1]);
183     s0     =mxGetPr(prhs[2]); a0     =mxGetPr(prhs[3]);
184     idof   =mxGetPr(prhs[4]); nz     =mxGetPr(prhs[5]);
185     dx     =mxGetPr(prhs[6]); dy     =mxGetPr(prhs[7]);
186     J      =mxGetPr(prhs[8]); Gewichte=mxGetPr(prhs[9]);
187     lambda =mxGetPr(prhs[10]);u      =mxGetPr(prhs[11]);
188     c1     =mxGetPr(prhs[12]);c2     =mxGetPr(prhs[13]);
189     c3     =mxGetPr(prhs[14]);sy     =mxGetPr(prhs[15]);
190
191
192     /*Set the output pointer to the output matrix. */
193     plhs[0]=mxCreateDoubleMatrix(2*nd,2*nd,mxREAL);
194
195     /*Create a C pointer to a copy of the output matrix */
196     M=mxGetPr(plhs[0]);
197

```

```
198  /* STEMA */
199  STEMA(M,n,u,u0,s0,a0,idof,nz,dx,dy,J,Gewichte,lambda,mu,c1,c2,c3,sy,
200      nq,nd,nd0);
201  }

1  #include <stdio.h>
2  #include <math.h>
3  #include <stdlib.h>
4  #include "mex.h"
5
6  double *CreateDoubleVector(int len)
7  {
8      return malloc(len*sizeof(double));
9  }
10
11  int *CreateIntVector(int len)
12  {
13      return malloc(len*sizeof(int));
14  }
15
16  double tr(double x[4])
17  {
18      double t;
19      t=x[0]+x[3];
20      return t;
21  }
22
23  double skal(double x[4], double y[4])
24  {
25      double s;
26      s=x[0]*y[0]+x[1]*y[1]+x[2]*y[2]+x[3]*y[3];
27      return s;
28  }
29
30
31  void symgrad(double y[4], double x[4])
32  {
33      y[0]=x[0];y[1]=(x[1]+x[2])/2.0;y[2]=y[1];y[3]=x[3];
34  }
35
36  void dev(double y[4], double x[4])
37  {
38      y[0]=x[0]-(x[0]+x[3])/2.0;y[1]=x[1];y[2]=x[2];y[3]=x[3]-(x[0]+x[3])/2.0;
39  }
40
41  double norm(double x[4])
42  {
43      double n;
44      n=sqrt(pow(x[0],2)+pow(x[1],2)+pow(x[2],2)+pow(x[3],2));
45      return n;
46  }
47
48  double dsign(double x)
49  {
50      double s;
51      if (x>=0) s=1; else s=-1;
```

```

52     return s;
53 }
54
55
56 void VEKTOR(double F[], double p[], int n, double u[], double u0[],
57             double s0[], double a0[], double idof[], double nz[],
58             double dx[], double dy[], double jm[], double Gewichte[],
59             double *lambda, double *mu, double *c1, double *c2,
60             double *c3, double *sy, int nq, int nd, int nd0)
61 {
62     double dtm, phi[2][2], *dxz, *dyz; dux1, duy1, dux2, duy2, Du[4], Eu[4];
63     double ds[4], da[4], v0[4], dv0[4], v1[4], dv1[4], dv[4]; w1[2], w2[2];
64     double d1[4], d2[4], e1[4], e2[4], de1[4], de2[4], f1, f2, delta;
65     int *I0, l, i, j, k, anzp, iw1;
66
67     p[0]=0.0; anzp=0;
68     I0=CreateIntVector(nd);
69     dxz=CreateDoubleVector(nd0); dyz=CreateDoubleVector(nd0);
70
71     for (i=0; i<2*nd; i++) F[i]=0.0;
72
73     dtm=jm[0]*jm[3]-jm[1]*jm[2];
74     phi[0][0]=jm[3]/dtm; phi[0][1]=-jm[2]/dtm;
75     phi[1][0]=-jm[1]/dtm; phi[1][1]=jm[0]/dtm;
76
77     for (j=0; j<nd; j++) I0[j]=abs((int) idof[(int) nz[j]-1]);
78     for (l=0; l<nq; l++)
79     {
80         for (j=0; j<nd0; j++)
81         {
82             dxz[j]=dx[j+nd0*1]*dsign(idof[j]);
83             dyz[j]=dy[j+nd0*1]*dsign(idof[j]);
84         }
85         dux1=0; duy1=0; dux2=0; duy2=0;
86         for (j=0; j<nd; j++)
87         {
88             dux1=dux1+(u[2*I0[j]-1-1]-u0[2*I0[j]-1-1])*dxz[(int) nz[j]-1];
89             duy1=duy1+(u[2*I0[j]-1-1]-u0[2*I0[j]-1-1])*dyz[(int) nz[j]-1];
90             dux2=dux2+(u[2*I0[j]-1]-u0[2*I0[j]-1])*dxz[(int) nz[j]-1];
91             duy2=duy2+(u[2*I0[j]-1]-u0[2*I0[j]-1])*dyz[(int) nz[j]-1];
92         }
93
94         Du[0]=dux1*phi[0][0]+duy1*phi[0][1];
95         Du[1]=dux1*phi[1][0]+duy1*phi[1][1];
96         Du[2]=dux2*phi[0][0]+duy2*phi[0][1];
97         Du[3]=dux2*phi[1][0]+duy2*phi[1][1];
98         symgrad(Eu, Du);
99         ds[0]=s0[1]-(s0[1]+s0[3*nq+1])/2.0;
100        ds[1]=s0[nq+1];
101        ds[2]=s0[2*nq+1];
102        ds[3]=s0[3*nq+1]-(s0[1]+s0[3*nq+1])/2.0;
103        da[0]=a0[1]-(a0[1]+a0[3*nq+1])/2.0;
104        da[1]=a0[nq+1];
105        da[2]=a0[2*nq+1];
106        da[3]=a0[3*nq+1]-(a0[1]+a0[3*nq+1])/2.0;

```

```

107     v0[0]=Eu[0]+(s0[1]+s0[3*nq+1])/(4.0*(lambda)+4.0*(mu))+
108         ds[0]/(2.0*(mu));
109     v0[1]=Eu[1]+ds[1]/(2.0*(mu));
110     v0[2]=Eu[2]+ds[2]/(2.0*(mu));
111     v0[3]=Eu[3]+(s0[1]+s0[3*nq+1])/(4.0*(lambda)+4.0*(mu))+
112         ds[3]/(2.0*(mu));
113     dev(dv0,v0);
114     dv[0]=dv0[0]-da[0]/(2*(mu)); dv[1]=dv0[1]-da[1]/(2*(mu));
115     dv[2]=dv0[2]-da[2]/(2*(mu)); dv[3]=dv0[3]-da[3]/(2*(mu));
116     v1[0]=-a0[1]/(2.0*(mu)); v1[1]=-a0[nq+1]/(2.0*(mu));
117     v1[2]=-a0[2*nq+1]/(2.0*(mu)); v1[3]=-a0[3*nq+1]/(2.0*(mu));
118     dev(dv1,v1);
119
120     if (norm(dv)-(*sy)/(2*(mu))>0)
121     {
122         f1>(*c2)+(*c3)/norm(dv); f2>(*c3)/pow(norm(dv),3);
123         delta=1.0;
124         p[0]=1.0; anzp++;
125     }
126     else
127     {
128         f1=2*(mu); f2=0.0; delta=0.0;
129     }
130     for (i=0; i<2*nd; i++)
131     {
132         iw1=(int) nz[(int) floor(i/2)]-1;
133         w1[0]=dxz[iw1]*phi[0][0]+dyz[iw1]*phi[0][1];
134         w1[1]=dxz[iw1]*phi[1][0]+dyz[iw1]*phi[1][1];
135         if (fmod(i+1,2)==0)
136         {
137             d1[0]=0.0; d1[1]=0.0; d1[2]=w1[0]; d1[3]=w1[1];
138         }
139         else
140         {
141             d1[0]=w1[0]; d1[1]=w1[1]; d1[2]=0.0; d1[3]=0.0;
142         }
143         symgrad(e1,d1); dev(de1,e1);
144         F[i]=F[i]+dtm*sqrt(3.0)*Gewichte[1]*((c1)*tr(v0)*tr(e1)+
145             f1*skal(dv0,e1)+delta*(f1*skal(dv1,e1)+skal(da,e1)));
146     }
147 }
148 if (anzp==nq) p[0]=2;
149 free(dxz); dxz=NULL; free(dyz); dyz=NULL;
150 free(I0); I0=NULL;
151 }
152
153 /* Gateway function */
154
155 void mexFunction(int nlhs, mxArray *plhs[], int nrhs,
156                 const mxArray *prhs[])
157 {
158     double *F,*p,*u,*u0,*s0,*a0,*idof,*nz,*dx,*dy,*J,*Gewichte,*lambda,*mu,
159         *c1,*c2,*c3,*sy;
160     int nd,nd0,nq,n;
161

```

```

162
163  /*Check for proper number of arguments */
164  if (nrhs!=16)
165      mexErrMsgTxt("16 input required!");
166  if (nlhs!=2)
167      mexErrMsgTxt("2 output required!");
168
169  n =mxGetM(prhs[0])/2; nq=mxGetN(prhs[9]);
170  nd=mxGetN(prhs[5]);  nd0=mxGetN(prhs[4]);
171
172  u      =mxGetPr(prhs[0]); u0      =mxGetPr(prhs[1]);
173  s0     =mxGetPr(prhs[2]); a0     =mxGetPr(prhs[3]);
174  idof   =mxGetPr(prhs[4]); nz     =mxGetPr(prhs[5]);
175  dx     =mxGetPr(prhs[6]); dy     =mxGetPr(prhs[7]);
176  J      =mxGetPr(prhs[8]); Gewichte=mxGetPr(prhs[9]);
177  lambda =mxGetPr(prhs[10]);mu     =mxGetPr(prhs[11]);
178  c1     =mxGetPr(prhs[12]);c2     =mxGetPr(prhs[13]);
179  c3     =mxGetPr(prhs[14]);sy     =mxGetPr(prhs[15]);
180
181
182  /*Set the output pointer to the output matrix. */
183  plhs[0]=mxCreateDoubleMatrix(2*nd,1,mxREAL);
184  plhs[1]=mxCreateDoubleMatrix(1,1,mxREAL);
185
186  /*Create a C pointer to a copy of the output matrix */
187  F=mxGetPr(plhs[0]);
188  p=mxGetPr(plhs[1]);
189
190  /* VEKTOR */
191  VEKTOR(F,p,n,u,u0,s0,a0,idof,nz,dx,dy,J,Gewichte,lambda,mu,c1,c2,c3,sy,
192        nq,nd,nd0);
193  }

```

A.3 Interpolation von Anfangswerten

Die folgende Routine `AW_interpolieren_h.m` interpoliert Anfangswerte bei einer h -Verfeinerung nach dem in Abschnitt 3.3.2 beschriebenen Verfahren.

```

1  function [u0,S0,A0,S0_K,A0_K,S0_P,A0_P,DSOX,DSOY,DAOX,DAOY,DSOX_K,DSOY_K,...
2          DAOX_K,DAOY_K,DSOX_P,DSOY_P,DAOX_P,DAOY_P,u00,S00,S00_K, S00_P, ...
3          A00,A00_K, A00_P, DS00X, DS00Y, DA00X, DA00Y, DS00X_K, DS00Y_K, ...
4          DA00X_K,DA00Y_K,DS00X_P,DS00Y_P,DA00X_P,DA00Y_P,ph0,ph00]=...
5  AW_interpolieren_h(VK,S00,A00,S00_K,A00_K,S00_P,A00_P,u00,u0,DS00X,DS00Y,...
6          DA00X,DA00Y,DS00X_K,DS00Y_K,DA00X_K,DA00Y_K,DS00X_P,DS00Y_P,DA00X_P,...
7          DA00Y_P,Koordinaten_n,Elemente_n,Kantennr_n,Aussenkanten_n,N,pmax,dof_n,...
8          tt,t0,t00,phase0,phase00)
9
10 global Koordinaten Elemente Dirichlet Neumann dof Kantennr Gewichte BaryK NQ ...
11      alpha1 beta1 NQ1 lbasis IPdof lbasis_IP lbasis_IP_K ...
12      lbasis_IP_P basis_p basis_IP NJ C2 C3
13
14 NE_n=size(Elemente_n,1);ph0=zeros(NE_n,1);ph00=zeros(NE_n,1);
15 M1=sparse(2*N,2*N);b_11=zeros(2*N,1);b_12=zeros(2*N,1);
16 S=zeros(NE_n,4*NQ);A=zeros(NE_n,4*NQ);

```

```

17 S_K{1}=zeros(NE_n,4*NQ1);A_K{1}=zeros(NE_n,4*NQ1);
18 S_K{2}=zeros(NE_n,4*NQ1);A_K{2}=zeros(NE_n,4*NQ1);
19 S_K{3}=zeros(NE_n,4*NQ1);A_K{3}=zeros(NE_n,4*NQ1);
20 S_P=zeros(NE_n,12);A_P=zeros(NE_n,12);
21
22 DSX=zeros(NE_n,4*NQ);DSY=zeros(NE_n,4*NQ);
23 DAX=zeros(NE_n,4*NQ);DAY=zeros(NE_n,4*NQ);
24 DSX_K{1}=zeros(NE_n,4*NQ1);DSY_K{1}=zeros(NE_n,4*NQ1);
25 DSX_K{2}=zeros(NE_n,4*NQ1);DSY_K{2}=zeros(NE_n,4*NQ1);
26 DSX_K{3}=zeros(NE_n,4*NQ1);DSY_K{3}=zeros(NE_n,4*NQ1);
27 DAX_K{1}=zeros(NE_n,4*NQ1);DAY_K{1}=zeros(NE_n,4*NQ1);
28 DAX_K{2}=zeros(NE_n,4*NQ1);DAY_K{2}=zeros(NE_n,4*NQ1);
29 DAX_K{3}=zeros(NE_n,4*NQ1);DAY_K{3}=zeros(NE_n,4*NQ1);
30 DSX_P=zeros(NE_n,12);DSY_P=zeros(NE_n,12);
31 DAX_P=zeros(NE_n,12);DSY_P=zeros(NE_n,12);
32
33 I21=4*[1:NQ]-3;I22=4*[1:NQ]-2;I23=4*[1:NQ]-1;I24=4*[1:NQ];
34
35 j=0;
36 for i=1:size(Elemente,1)
37     idof=dof(i,:);nz01=find(idof);iIPdof=IPdof(i,:);nziIP=find(iIPdof);
38     S00_i=reshape(S00(i,:),NQ,4);A00_i=reshape(A00(i,:),NQ,4);
39     S00_P_i=reshape(S00_P(i,:),3,4);A00_P_i=reshape(A00_P(i,:),3,4);
40
41     DS00X_i=reshape(DS00X(i,:),NQ,4);DS00Y_i=reshape(DS00Y(i,:),NQ,4);
42     DA00X_i=reshape(DA00X(i,:),NQ,4);DA00Y_i=reshape(DA00Y(i,:),NQ,4);
43     DS00X_P_i=reshape(DS00X_P(i,:),3,4);DS00Y_P_i=reshape(DS00Y_P(i,:),3,4);
44     DA00X_P_i=reshape(DA00X_P(i,:),3,4);DA00Y_P_i=reshape(DA00Y_P(i,:),3,4);
45
46     for l1=1:3
47         S00_K_i{l1}=reshape(S00_K{l1}(i,:),NQ1,4);
48         A00_K_i{l1}=reshape(A00_K{l1}(i,:),NQ1,4);
49         DS00X_K_i{l1}=reshape(DS00X_K{l1}(i,:),NQ1,4);
50         DS00Y_K_i{l1}=reshape(DS00Y_K{l1}(i,:),NQ1,4);
51         DA00X_K_i{l1}=reshape(DA00X_K{l1}(i,:),NQ1,4);
52         DA00Y_K_i{l1}=reshape(DA00Y_K{l1}(i,:),NQ1,4);
53     end
54
55
56 I=diag(Kantennr(Elemente(i,:),Elemente(i,[2,3,1])));
57 RefineEdge=find(VK(I));
58 NeuKnoten=VK(I(RefineEdge));
59 if size(RefineEdge,1)==3
60     z3=1;
61     locNeueElemente=[NeuKnoten([2,3,1]);
62                     [Elemente(i,1) NeuKnoten(1) NeuKnoten(3)];
63                     [NeuKnoten(1) Elemente(i,2) NeuKnoten(2)];
64                     [NeuKnoten(3) NeuKnoten(2) Elemente(i,3)]];
65 elseif size(RefineEdge,1)==2
66     z3=RefineEdge(2)-1;
67     locNeueElemente=...
68     [[NeuKnoten(1), Elemente(i,RefineEdge(2)),NeuKnoten(2)];
69     [Elemente(i,5-RefineEdge(2)),Elemente(i,rem(5-RefineEdge(2),3)+1),...
70     NeuKnoten(1)]];
71     [Elemente(i,rem(RefineEdge(2),3)+1),NeuKnoten(1),NeuKnoten(2)]];

```

```

72     elseif size(RefineEdge,1)==1
73         z3=1;
74         locNeueElemente=[[Elemente(i,[2,3]),NeuKnoten];
75                         [Elemente(i,[3,1]),NeuKnoten]];
76     else
77         z3=1;
78         locNeueElemente=Elemente(i,:);
79     end
80     z1=5-size(locNeueElemente,1);
81
82     if (size(locNeueElemente,1)>1)
83         if ((phase00(i)==0) | (phase00(i)==2))
84             M2=MMatrix1(basis_IP,iIPdof,nziIP);
85             b2=(basis_IP.*(sign(iIPdof)'*Gewichte))*S00_i;
86             b3=(basis_IP.*(sign(iIPdof)'*Gewichte))*A00_i;
87
88             b6=(basis_IP.*(sign(iIPdof)'*Gewichte))*DS00X_i;
89             b7=(basis_IP.*(sign(iIPdof)'*Gewichte))*DS00Y_i;
90             b8=(basis_IP.*(sign(iIPdof)'*Gewichte))*DA00X_i;
91             b9=(basis_IP.*(sign(iIPdof)'*Gewichte))*DA00Y_i;
92
93
94             [L,U]=lu(M2);
95             S00_basis=U\(L\b2);A00_basis=U\(L\b3);
96
97             DS00X_basis=U\(L\b6);DS00Y_basis=U\(L\b7);
98             DA00X_basis=U\(L\b8);DA00Y_basis=U\(L\b9);
99         else
100     % Koordinaten der Quadraturpunkte im Element i
101     xyi_I=BaryK'*Koordinaten(Elemente(i,:),:);
102     xyi_K1=[alpha1',beta1']*Koordinaten(Elemente(i,[1,2]),:);
103     xyi_K2=[alpha1',beta1']*Koordinaten(Elemente(i,[2,3]),:);
104     xyi_K3=[alpha1',beta1']*Koordinaten(Elemente(i,[3,1]),:);
105     xyi_P=Koordinaten(Elemente(i,:),:);
106     % Bestimmung eines lokalen FE-Netzes im Element i mit den Quadraturpunkten
107     % als Knoten
108     Koordinaten_i=zeros((NQ1+2)*(NQ1+1)+1,2);
109     Elemente_i=zeros(2*((NQ1+1)*NQ1)+NQ1+1,3);
110     Koordinaten_i(1,:)=xyi_P(1,:);
111     Koordinaten_i(NQ1+2,:)=xyi_P(2,:);
112     Koordinaten_i((NQ1+2)*(NQ1+1)+1,:)=xyi_P(3,:);
113     Koordinaten_i(2:NQ1+1,:)=xyi_K1;
114     Koordinaten_i((NQ1+2)*2:NQ1+2:(NQ1+2)*(NQ1+1),:)=xyi_K2;
115     Koordinaten_i((NQ1+2)*NQ1+1:-(NQ1+2):NQ1+3,:)=xyi_K3;
116     num=0;
117     for l1=1:NQ1
118         for l2=1:NQ1
119             num=num+1;
120             Koordinaten_i(1+l1+l2*(NQ1+2),:)=xyi_I(num,:);
121         end
122     end
123     num=0;
124     for l2=1:NQ1
125         for l1=1:(NQ1+1)
126             num=num+1;

```



```

127     Elemente_i(num, :)= [l2*(NQ1+2)+l1, (l2-1)*(NQ1+2)+l1, (l2-1)*(NQ1+2)+l1+l1];
128     num=num+1;
129     Elemente_i(num, :)= [l2*(NQ1+2)+l1, (l2-1)*(NQ1+2)+l1+l1, l2*(NQ1+2)+l1+l1];
130     end
131 end
132 for l1=1:(NQ1+1)
133     num=num+1;
134     Elemente_i(num, :)= [(NQ1+2)*(NQ1+1)+1, (NQ1+2)*NQ1+l1, (NQ1+2)*NQ1+l1+l1];
135 end
136 % Speicherung aller Variablen in Reihenfolge der Interpolationspunkte
137 S00_k(1, :)=S00_P_i(1, :);
138 S00_k(NQ1+2, :)=S00_P_i(2, :);
139 S00_k((NQ1+2)*(NQ1+1)+1, :)=S00_P_i(3, :);
140 S00_k(2:NQ1+1, :)=S00_K_i{1};
141 S00_k((NQ1+2)*2:NQ1+2:(NQ1+2)*(NQ1+1), :)=S00_K_i{2};
142 S00_k((NQ1+2)*NQ1+1:-(NQ1+2):NQ1+3, :)=S00_K_i{3};
143
144 A00_k(1, :)=A00_P_i(1, :);
145 A00_k(NQ1+2, :)=A00_P_i(2, :);
146 A00_k((NQ1+2)*(NQ1+1)+1, :)=A00_P_i(3, :);
147 A00_k(2:NQ1+1, :)=A00_K_i{1};
148 A00_k((NQ1+2)*2:NQ1+2:(NQ1+2)*(NQ1+1), :)=A00_K_i{2};
149 A00_k((NQ1+2)*NQ1+1:-(NQ1+2):NQ1+3, :)=A00_K_i{3};
150
151 DS00X_k(1, :)=DS00X_P_i(1, :);
152 DS00X_k(NQ1+2, :)=DS00X_P_i(2, :);
153 DS00X_k((NQ1+2)*(NQ1+1)+1, :)=DS00X_P_i(3, :);
154 DS00X_k(2:NQ1+1, :)=DS00X_K_i{1};
155 DS00X_k((NQ1+2)*2:NQ1+2:(NQ1+2)*(NQ1+1), :)=DS00X_K_i{2};
156 DS00X_k((NQ1+2)*NQ1+1:-(NQ1+2):NQ1+3, :)=DS00X_K_i{3};
157
158 DS00Y_k(1, :)=DS00Y_P_i(1, :);
159 DS00Y_k(NQ1+2, :)=DS00Y_P_i(2, :);
160 DS00Y_k((NQ1+2)*(NQ1+1)+1, :)=DS00Y_P_i(3, :);
161 DS00Y_k(2:NQ1+1, :)=DS00Y_K_i{1};
162 DS00Y_k((NQ1+2)*2:NQ1+2:(NQ1+2)*(NQ1+1), :)=DS00Y_K_i{2};
163 DS00Y_k((NQ1+2)*NQ1+1:-(NQ1+2):NQ1+3, :)=DS00Y_K_i{3};
164
165 DA00X_k(1, :)=DA00X_P_i(1, :);
166 DA00X_k(NQ1+2, :)=DA00X_P_i(2, :);
167 DA00X_k((NQ1+2)*(NQ1+1)+1, :)=DA00X_P_i(3, :);
168 DA00X_k(2:NQ1+1, :)=DA00X_K_i{1};
169 DA00X_k((NQ1+2)*2:NQ1+2:(NQ1+2)*(NQ1+1), :)=DA00X_K_i{2};
170 DA00X_k((NQ1+2)*NQ1+1:-(NQ1+2):NQ1+3, :)=DA00X_K_i{3};
171
172 DA00Y_k(1, :)=DA00Y_P_i(1, :);
173 DA00Y_k(NQ1+2, :)=DA00Y_P_i(2, :);
174 DA00Y_k((NQ1+2)*(NQ1+1)+1, :)=DA00Y_P_i(3, :);
175 DA00Y_k(2:NQ1+1, :)=DA00Y_K_i{1};
176 DA00Y_k((NQ1+2)*2:NQ1+2:(NQ1+2)*(NQ1+1), :)=DA00Y_K_i{2};
177 DA00Y_k((NQ1+2)*NQ1+1:-(NQ1+2):NQ1+3, :)=DA00Y_K_i{3};
178
179 num=0;
180 for l1=1:NQ1
181     for l2=1:NQ1

```

```

182     num=num+1;
183     S00_k(1+l1+12*(NQ1+2),:)=S00_i(num,:);
184     A00_k(1+l1+12*(NQ1+2),:)=A00_i(num,:);
185     DS00X_k(1+l1+12*(NQ1+2),:)=DS00X_i(num,:);
186     DS00Y_k(1+l1+12*(NQ1+2),:)=DS00Y_i(num,:);
187     DA00X_k(1+l1+12*(NQ1+2),:)=DA00X_i(num,:);
188     DA00Y_k(1+l1+12*(NQ1+2),:)=DA00Y_i(num,:);
189     end
190     end
191     end
192     end
193
194
195     for k=1:size(locNeueElemente,1)
196         j=j+1;
197         ph0(j)=phase0(i);ph00(j)=phase00(i);
198         lokKoordinaten=BaryK'*Koordinaten_n(Elemente_n(j,:),:);
199         T=1/2*det([1,1,1;Koordinaten_n(locNeueElemente(k,:),:)]');
200         jdof=dof_n(j,:);nz1=find(jdof);
201         ME1=MMatrix1(basis_p,jdof,nz1);
202         I11=2*abs(jdof(nz1))-1;I12=2*abs(jdof(nz1));
203         I1=reshape([I11;I12],2*size(nz1,2),1);
204         M1(I11,I11)=M1(I11,I11)+T*ME1;M1(I12,I12)=M1(I12,I12)+T*ME1;
205         IO1=2*abs(idof(nz01))-1;IO2=2*abs(idof(nz01));
206         basis_ref=basis_p.*(sign(jdof)'*ones(1,NQ));
207         l_basis_ref=(lbasis{z1,k,z3}(nz01,:).*(sign(idof(nz01))'*ones(1,NQ)))';
208         b_11(I1)=b_11(I1)+T*reshape((basis_ref(nz1,:).*(l_basis_ref*...
209             [u0(I01),u0(I02)]).*(Gewichte'*[1,1]))',2*size(nz1,2),1);
210         b_12(I1)=b_12(I1)+T*reshape((basis_ref(nz1,:).*(l_basis_ref*...
211             [u0(I01),u0(I02)]).*(Gewichte'*[1,1]))',2*size(nz1,2),1);
212
213
214         if (size(locNeueElemente,1)==1)
215             S(j,:)=S00(i,:);A(j,:)=A00(i,:);
216
217             DSX(j,:)=DS00X(i,:);DSY(j,:)=DS00Y(i,:);
218             DAX(j,:)=DA00X(i,:);DAY(j,:)=DA00Y(i,:);
219             S_P(j,:)=S00_P(i,:);A_P(j,:)=A00_P(i,:);
220             DSX_P(j,:)=DS00X_P(i,:);DSY_P(j,:)=DS00Y_P(i,:);
221             DAX_P(j,:)=DA00X_P(i,:);DAY_P(j,:)=DA00Y_P(i,:);
222
223             for l=1:3
224                 S_K{l}(j,:)=S00_K{l}(i,:);A_K{l}(j,:)=A00_K{l}(i,:);
225                 DSX_K{l}(j,:)=DS00X_K{l}(i,:);DSY_K{l}(j,:)=DS00Y_K{l}(i,:);
226                 DAX_K{l}(j,:)=DA00X_K{l}(i,:);DAY_K{l}(j,:)=DA00Y_K{l}(i,:);
227             end
228         else
229             if ((phase00(i)==0) | (phase00(i)==2))
230 % Interpolation der Quadraturpunktdaten auf elastischen+
231 % komplett plastischen Elementen
232                 vz1=sign(iIPdof)'*ones(1,NQ);
233                 vz2=sign(iIPdof)'*ones(1,NQ1);
234                 vz3=sign(iIPdof)'*ones(1,3);
235                 S(j,:)=reshape((lbasis_IP{z1,k,z3}.*vz1)'*S00_basis,1,4*NQ);
236                 A(j,:)=reshape((lbasis_IP{z1,k,z3}.*vz1)'*A00_basis,1,4*NQ);

```

```

237 DSX(j,:)=reshape((lbasis_IP{z1,k,z3}.*vz1)'*DS00X_basis,1,4*NQ);
238 DSY(j,:)=reshape((lbasis_IP{z1,k,z3}.*vz1)'*DS00Y_basis,1,4*NQ);
239 DAX(j,:)=reshape((lbasis_IP{z1,k,z3}.*vz1)'*DA00X_basis,1,4*NQ);
240 DAY(j,:)=reshape((lbasis_IP{z1,k,z3}.*vz1)'*DA00Y_basis,1,4*NQ);
241
242 S_P(j,:)=reshape((lbasis_IP_P{z1,k,z3}.*vz3)'*S00_basis,1,12);
243 A_P(j,:)=reshape((lbasis_IP_P{z1,k,z3}.*vz3)'*A00_basis,1,12);
244 DSX_P(j,:)=reshape((lbasis_IP_P{z1,k,z3}.*vz3)'*DS00X_basis,1,12);
245 DSY_P(j,:)=reshape((lbasis_IP_P{z1,k,z3}.*vz3)'*DS00Y_basis,1,12);
246 DAX_P(j,:)=reshape((lbasis_IP_P{z1,k,z3}.*vz3)'*DA00X_basis,1,12);
247 DAY_P(j,:)=reshape((lbasis_IP_P{z1,k,z3}.*vz3)'*DA00Y_basis,1,12);
248
249 for l=1:3
250   S_K{l}(j,:)=reshape((lbasis_IP_K{z1,k,z3,l}.*vz2)'*S00_basis,1,4*NQ1);
251   A_K{l}(j,:)=reshape((lbasis_IP_K{z1,k,z3,l}.*vz2)'*A00_basis,1,4*NQ1);
252   DSX_K{l}(j,:)=reshape((lbasis_IP_K{z1,k,z3,l}.*vz2)'*DS00X_basis,1,4*NQ1);
253   DSY_K{l}(j,:)=reshape((lbasis_IP_K{z1,k,z3,l}.*vz2)'*DS00Y_basis,1,4*NQ1);
254   DAX_K{l}(j,:)=reshape((lbasis_IP_K{z1,k,z3,l}.*vz2)'*DA00X_basis,1,4*NQ1);
255   DAY_K{l}(j,:)=reshape((lbasis_IP_K{z1,k,z3,l}.*vz2)'*DA00Y_basis,1,4*NQ1);
256   end
257 else
258 % Interpolation der Quadraturpunktdaten auf elastisch-plastischen Elementen
259 xyj_I=BaryK'*Koordinaten_n(Elemente_n(j,:),:);
260 xyj_K{1}=[alpha1',beta1']*Koordinaten_n(Elemente_n(j,[1,2]),:);
261 xyj_K{2}=[alpha1',beta1']*Koordinaten_n(Elemente_n(j,[2,3]),:);
262 xyj_K{3}=[alpha1',beta1']*Koordinaten_n(Elemente_n(j,[3,1]),:);
263 xyj_P=Koordinaten_n(Elemente_n(j,:),:);
264 % Schleife ueber alle Quadraturpunkte im neuen Element Nr.j
265 % 1. Quadraturpunkte im Innern
266 for qnr=1:size(xyj_I,1)
267 % Quadraturpunkt Nr. qnr liegt in Element Nr. i_nr
268 i_nr=T_nr(xyj_I(qnr,:),Elemente_i,Koordinaten_i,1);
269 T_i=det([1,1,1;Koordinaten_i(Elemente_i(i_nr,:),:)]);
270 a1=(1/T_i)*det([1,1,1;Koordinaten_i(Elemente_i(i_nr,2),:)]',...
271   Koordinaten_i(Elemente_i(i_nr,3),:)]',xyj_I(qnr,:)]);
272 a2=(1/T_i)*det([1,1,1;Koordinaten_i(Elemente_i(i_nr,3),:)]',...
273   Koordinaten_i(Elemente_i(i_nr,1),:)]',xyj_I(qnr,:)]);
274 a3=(1/T_i)*det([1,1,1;Koordinaten_i(Elemente_i(i_nr,1),:)]',...
275   Koordinaten_i(Elemente_i(i_nr,2),:)]',xyj_I(qnr,:)]);
276 I_qnr=[qnr,NQ+qnr,2*NQ+qnr,3*NQ+qnr];
277 S(j,I_qnr)=[a1,a2,a3]*S00_k(Elemente_i(i_nr,:),:);
278 A(j,I_qnr)=[a1,a2,a3]*A00_k(Elemente_i(i_nr,:),:);
279 DSX(j,I_qnr)=[a1,a2,a3]*DS00X_k(Elemente_i(i_nr,:),:);
280 DSY(j,I_qnr)=[a1,a2,a3]*DS00Y_k(Elemente_i(i_nr,:),:);
281 DAX(j,I_qnr)=[a1,a2,a3]*DA00X_k(Elemente_i(i_nr,:),:);
282 DAY(j,I_qnr)=[a1,a2,a3]*DA00Y_k(Elemente_i(i_nr,:),:);
283 end
284 % 2. Quadraturpunkte auf den Kanten
285 for l1=1:3
286   for qnr=1:size(xyj_K{l1},1)
287     i_nr=T_nr(xyj_K{l1}(qnr,:),Elemente_i,Koordinaten_i,1);
288     T_i=det([1,1,1;Koordinaten_i(Elemente_i(i_nr,:),:)]);
289     a1=(1/T_i)*det([1,1,1;Koordinaten_i(Elemente_i(i_nr,2),:)]',...
290       Koordinaten_i(Elemente_i(i_nr,3),:)]',xyj_K{l1}(qnr,:)]);
291     a2=(1/T_i)*det([1,1,1;Koordinaten_i(Elemente_i(i_nr,3),:)]',...

```

```

292     Koordinaten_i(Elemente_i(i_nr,1),:)',xyj_K{11}(qnr,:)]');
293     a3=(1/T_i)*det([1,1,1;Koordinaten_i(Elemente_i(i_nr,1),:)', ...
294     Koordinaten_i(Elemente_i(i_nr,2),:)',xyj_K{11}(qnr,:)]');
295     I_qnr=[qnr,NQ1+qnr,2*NQ1+qnr,3*NQ1+qnr];
296     S_K{11}(j,I_qnr)=[a1,a2,a3]*S00_k(Elemente_i(i_nr,:),:);
297     A_K{11}(j,I_qnr)=[a1,a2,a3]*A00_k(Elemente_i(i_nr,:),:);
298     DSX_K{11}(j,I_qnr)=[a1,a2,a3]*DS00X_k(Elemente_i(i_nr,:),:);
299     DSY_K{11}(j,I_qnr)=[a1,a2,a3]*DS00Y_k(Elemente_i(i_nr,:),:);
300     DAX_K{11}(j,I_qnr)=[a1,a2,a3]*DA00X_k(Elemente_i(i_nr,:),:);
301     DAY_K{11}(j,I_qnr)=[a1,a2,a3]*DA00Y_k(Elemente_i(i_nr,:),:);
302     end
303 end
304 % Quadraturpunkte auf den Eckpunkten
305 for qnr=1:size(xyj_P,1)
306 % Quadraturpunkt Nr. qnr liegt in Element Nr. i_nr
307     i_nr=T_nr(xyj_P(qnr,:),Elemente_i,Koordinaten_i,1);
308     T_i=det([1,1,1;Koordinaten_i(Elemente_i(i_nr,:),:)',:]);
309     a1=(1/T_i)*det([1,1,1;Koordinaten_i(Elemente_i(i_nr,2),:)', ...
310     Koordinaten_i(Elemente_i(i_nr,3),:)',xyj_P(qnr,:)]');
311     a2=(1/T_i)*det([1,1,1;Koordinaten_i(Elemente_i(i_nr,3),:)', ...
312     Koordinaten_i(Elemente_i(i_nr,1),:)',xyj_P(qnr,:)]');
313     a3=(1/T_i)*det([1,1,1;Koordinaten_i(Elemente_i(i_nr,1),:)', ...
314     Koordinaten_i(Elemente_i(i_nr,2),:)',xyj_P(qnr,:)]');
315     I_qnr=[qnr,3+qnr,2*3+qnr,3*3+qnr];
316     S_P(j,I_qnr)=[a1,a2,a3]*S00_k(Elemente_i(i_nr,:),:);
317     A_P(j,I_qnr)=[a1,a2,a3]*A00_k(Elemente_i(i_nr,:),:);
318     DSX_P(j,I_qnr)=[a1,a2,a3]*DS00X_k(Elemente_i(i_nr,:),:);
319     DSY_P(j,I_qnr)=[a1,a2,a3]*DS00Y_k(Elemente_i(i_nr,:),:);
320     DAX_P(j,I_qnr)=[a1,a2,a3]*DA00X_k(Elemente_i(i_nr,:),:);
321     DAY_P(j,I_qnr)=[a1,a2,a3]*DA00Y_k(Elemente_i(i_nr,:),:);
322     end
323     end
324     end
325     end % Schleife ueber lokale neue Elemente
326 end
327
328 u0=M1\b_11;u0=M1\b_12;u0=(1-tt)*u0+tt*u0;
329 S00=S;S00_K=S_K;S00_P=S_P;A00=A;A00_K=A_K;A00_P=A_P;
330
331 DS00X=DSX;DS00X_K=DSX_K;DS00X_P=DSX_P;DS00Y=DSY;DS00Y_K=DSY_K;DS00Y_P=DSY_P;
332 DA00X=DAX;DA00X_K=DAX_K;DA00X_P=DAX_P;DA00Y=DAY;DA00Y_K=DAY_K;DA00Y_P=DAY_P;
333
334 [DSOX,DSOY,DAOX,DAOY,DSOX_K,DSOY_K,DAOX_K,DAOY_K,DSOX_P,DSOY_P,...
335     DAOX_P,DAOY_P]=...
336 DSigma(u0,u00,S00,A00,S00_K,A00_K,S00_P,A00_P,DS00X,DS00Y,DA00X,DA00Y,...
337     DS00X_K,DS00Y_K,DA00X_K,DA00Y_K,DS00X_P,DS00Y_P,DA00X_P,DA00Y_P, ...
338     Koordinaten_n,Elemente_n,dof_n,Gewichte,NQ,NQ1);
339
340 [S0,A0,SO_K,A0_K,SO_P,A0_P,DSOX,DSOY,DAOX,DAOY,DSOX_K,DSOY_K,DAOX_K,DAOY_K,...
341     DSOX_P,DSOY_P,DAOX_P,DAOY_P]=...
342 QPD_updaten(u0,u00,S00,A00,S00_K,A00_K,S00_P,A00_P,DSOX,DSOY,DAOX,DAOY, ...
343     DSOX_K,DSOY_K,DAOX_K,DAOY_K,DSOX_P,DSOY_P,DAOX_P,DAOY_P,DS00X,DS00Y, ...
344     DA00X,DA00Y,DS00X_K,DS00Y_K,DA00X_K,DA00Y_K,DS00X_P,DS00Y_P, ...
345     DA00X_P,DA00Y_P,tt,Koordinaten_n,Elemente_n,dof_n,NQ,NQ1);

```

Die folgende Routine `AW_interpolieren_p.m` interpoliert Anfangswerte bei einer p -Verfeinerung nach dem in Abschnitt 3.3.2 beschriebenen Verfahren.

```

1  function [S0,A0,S0_K,A0_K,DS0X,DS0Y,DA0X,DA0Y,DS0X_K,DS0Y_K,DA0X_K,DA0Y_K,...
2          S00,S00_K,A00,A00_K,DS00X,DS00Y, DA00X, DA00Y, DS00X_K, DS00Y_K, ...
3          DA00X_K,DA00Y_K]=...
4  AW_interpolieren(S00,A00,S00_K,A00_K,S00_P,A00_P,u00,u0,DS00X,DS00Y, ...
5          DA00X, DA00Y,DS00X_K,DS00Y_K,DA00X_K,DA00Y_K,DS00X_P,DS00Y_P,...
6          DA00X_P,DA00Y_P,Gewichte_n, BaryK_n, NQ_n,alpha1_n,beta1_n,...
7          NQ1_n,pmax,dof_n,IPmax,tt,t0,t00,phase0,phase00)
8
9  global Koordinaten Elemente Dirichlet Neumann dof Kantenr P_ref ...
10         Gewichte BaryK NQ alpha1 beta1 NQ1 basis_p NJ C2 C3
11
12  S=zeros(NJ,4*NQ_n);A=zeros(NJ,4*NQ_n);
13  S_K{1}=zeros(NJ,4*NQ1_n);A_K{1}=zeros(NJ,4*NQ1_n);
14  S_K{2}=zeros(NJ,4*NQ1_n);A_K{2}=zeros(NJ,4*NQ1_n);
15  S_K{3}=zeros(NJ,4*NQ1_n);A_K{3}=zeros(NJ,4*NQ1_n);
16
17  DSX=zeros(NJ,4*NQ_n);DSY=zeros(NJ,4*NQ_n);
18  DAX=zeros(NJ,4*NQ_n);DAY=zeros(NJ,4*NQ_n);
19  DSX_K{1}=zeros(NJ,4*NQ1_n);DSY_K{1}=zeros(NJ,4*NQ1_n);
20  DSX_K{2}=zeros(NJ,4*NQ1_n);DSY_K{2}=zeros(NJ,4*NQ1_n);
21  DSX_K{3}=zeros(NJ,4*NQ1_n);DSY_K{3}=zeros(NJ,4*NQ1_n);
22  DAX_K{1}=zeros(NJ,4*NQ1_n);DAY_K{1}=zeros(NJ,4*NQ1_n);
23  DAX_K{2}=zeros(NJ,4*NQ1_n);DAY_K{2}=zeros(NJ,4*NQ1_n);
24  DAX_K{3}=zeros(NJ,4*NQ1_n);DAY_K{3}=zeros(NJ,4*NQ1_n);
25
26  basis_IP=fwertbasis(BaryK'*P_ref,IPmax);
27  basis_IP_K{1}=fwertbasis([alpha1',beta1']*[-1,0;1,0],IPmax);
28  basis_IP_K{2}=fwertbasis([alpha1',beta1']*[1,0;0,sqrt(3)],IPmax);
29  basis_IP_K{3}=fwertbasis([alpha1',beta1']*[0,sqrt(3);-1,0],IPmax);
30  IPdof=erstelledof2(Koordinaten,Elemente,IPmax*ones(NJ,1),Kantenr);
31
32  basis_IP_n=fwertbasis(BaryK_n'*P_ref,IPmax);
33  basis_IP_K_n{1}=fwertbasis([alpha1_n',beta1_n']*[-1,0;1,0],IPmax);
34  basis_IP_K_n{2}=fwertbasis([alpha1_n',beta1_n']*[1,0;0,sqrt(3)],IPmax);
35  basis_IP_K_n{3}=fwertbasis([alpha1_n',beta1_n']*[0,sqrt(3);-1,0],IPmax);
36
37  for i=1:size(Elemente,1)
38      iIPdof=IPdof(i,:);nziIP=find(iIPdof);
39      S00_i=reshape(S00(i,:),NQ,4);A00_i=reshape(A00(i,:),NQ,4);
40      S00_P_i=reshape(S00_P(i,:),3,4);A00_P_i=reshape(A00_P(i,:),3,4);
41
42      DS00X_i=reshape(DS00X(i,:),NQ,4);DS00Y_i=reshape(DS00Y(i,:),NQ,4);
43      DA00X_i=reshape(DA00X(i,:),NQ,4);DA00Y_i=reshape(DA00Y(i,:),NQ,4);
44      DS00X_P_i=reshape(DS00X_P(i,:),3,4);DS00Y_P_i=reshape(DS00Y_P(i,:),3,4);
45      DA00X_P_i=reshape(DA00X_P(i,:),3,4);DA00Y_P_i=reshape(DA00Y_P(i,:),3,4);
46
47  for l1=1:3
48      S00_K_i{l1}=reshape(S00_K{l1}(i,:),NQ1,4);
49      A00_K_i{l1}=reshape(A00_K{l1}(i,:),NQ1,4);
50      DS00X_K_i{l1}=reshape(DS00X_K{l1}(i,:),NQ1,4);
51      DS00Y_K_i{l1}=reshape(DS00Y_K{l1}(i,:),NQ1,4);
52      DA00X_K_i{l1}=reshape(DA00X_K{l1}(i,:),NQ1,4);

```

```

53     DA00Y_K_i{l1}=reshape(DA00Y_K{l1}(i,:),NQ1,4);
54 end
55 if (phase00(i)~=1)
56     M2=MMatrix1(basis_IP,iIPdof,nziIP);
57     b2=(basis_IP.*(sign(iIPdof))*Gewichte)*S00_i;
58     b3=(basis_IP.*(sign(iIPdof))*Gewichte)*A00_i;
59     b6=(basis_IP.*(sign(iIPdof))*Gewichte)*DS00X_i;
60     b7=(basis_IP.*(sign(iIPdof))*Gewichte)*DS00Y_i;
61     b8=(basis_IP.*(sign(iIPdof))*Gewichte)*DA00X_i;
62     b9=(basis_IP.*(sign(iIPdof))*Gewichte)*DA00Y_i;
63
64     [L,U]=lu(M2);
65     S00_basis=U\(L\b2);A00_basis=U\(L\b3);
66     DS00X_basis=U\(L\b6);DS00Y_basis=U\(L\b7);
67     DA00X_basis=U\(L\b8);DA00Y_basis=U\(L\b9);
68 else
69 % Koordinaten der Quadraturpunkte im Element i
70     xyi_I=BaryK'*Koordinaten(Elemente(i,:),:);
71     xyi_K1=[alpha1',beta1']*Koordinaten(Elemente(i,[1,2]),:);
72     xyi_K2=[alpha1',beta1']*Koordinaten(Elemente(i,[2,3]),:);
73     xyi_K3=[alpha1',beta1']*Koordinaten(Elemente(i,[3,1]),:);
74     xyi_P=Koordinaten(Elemente(i,:),:);
75 % Bestimmung eines lokalen FE-Netzes im Element i mit den alten
76 % Quadraturpunkten als Knoten
77     Koordinaten_i=zeros((NQ1+2)*(NQ1+1)+1,2);
78     Elemente_i=zeros(2*((NQ1+1)*NQ1)+NQ1+1,3);
79     Koordinaten_i(1,:)=xyi_P(1,:);
80     Koordinaten_i(NQ1+2,:)=xyi_P(2,:);
81     Koordinaten_i((NQ1+2)*(NQ1+1)+1,:)=xyi_P(3,:);
82     Koordinaten_i(2:NQ1+1,:)=xyi_K1;
83     Koordinaten_i((NQ1+2)*2:NQ1+2:(NQ1+2)*(NQ1+1),:)=xyi_K2;
84     Koordinaten_i((NQ1+2)*NQ1+1:-(NQ1+2):NQ1+3,:)=xyi_K3;
85     num=0;
86     for l1=1:NQ1
87         for l2=1:NQ1
88             num=num+1;
89             Koordinaten_i(1+l1+l2*(NQ1+2),:)=xyi_I(num,:);
90         end
91     end
92     num=0;
93     for l2=1:NQ1
94         for l1=1:(NQ1+1)
95             num=num+1;
96             Elemente_i(num,:)= [l2*(NQ1+2)+l1,(l2-1)*(NQ1+2)+l1,(l2-1)*(NQ1+2)+l1+1];
97             num=num+1;
98             Elemente_i(num,:)= [l2*(NQ1+2)+l1,(l2-1)*(NQ1+2)+l1+1,l2*(NQ1+2)+l1+1];
99         end
100    end
101    for l1=1:(NQ1+1)
102        num=num+1;
103        Elemente_i(num,:)= [(NQ1+2)*(NQ1+1)+1,(NQ1+2)*NQ1+l1,(NQ1+2)*NQ1+l1+1];
104    end
105 %Speicherung aller Variablen in Reihenfolge der Interpolationspunkte
106     S00_k(1,:)=S00_P_i(1,:);
107     S00_k(NQ1+2,:)=S00_P_i(2,:);

```

```

108     S00_k((NQ1+2)*(NQ1+1)+1,:) = S00_P_i(3,:);
109     S00_k(2:NQ1+1,:) = S00_K_i{1};
110     S00_k((NQ1+2)*2:NQ1+2:(NQ1+2)*(NQ1+1),:) = S00_K_i{2};
111     S00_k((NQ1+2)*NQ1+1:-(NQ1+2):NQ1+3,:) = S00_K_i{3};
112
113     A00_k(1,:) = A00_P_i(1,:);
114     A00_k(NQ1+2,:) = A00_P_i(2,:);
115     A00_k((NQ1+2)*(NQ1+1)+1,:) = A00_P_i(3,:);
116     A00_k(2:NQ1+1,:) = A00_K_i{1};
117     A00_k((NQ1+2)*2:NQ1+2:(NQ1+2)*(NQ1+1),:) = A00_K_i{2};
118     A00_k((NQ1+2)*NQ1+1:-(NQ1+2):NQ1+3,:) = A00_K_i{3};
119
120     DS00X_k(1,:) = DS00X_P_i(1,:);
121     DS00X_k(NQ1+2,:) = DS00X_P_i(2,:);
122     DS00X_k((NQ1+2)*(NQ1+1)+1,:) = DS00X_P_i(3,:);
123     DS00X_k(2:NQ1+1,:) = DS00X_K_i{1};
124     DS00X_k((NQ1+2)*2:NQ1+2:(NQ1+2)*(NQ1+1),:) = DS00X_K_i{2};
125     DS00X_k((NQ1+2)*NQ1+1:-(NQ1+2):NQ1+3,:) = DS00X_K_i{3};
126
127     DS00Y_k(1,:) = DS00Y_P_i(1,:);
128     DS00Y_k(NQ1+2,:) = DS00Y_P_i(2,:);
129     DS00Y_k((NQ1+2)*(NQ1+1)+1,:) = DS00Y_P_i(3,:);
130     DS00Y_k(2:NQ1+1,:) = DS00Y_K_i{1};
131     DS00Y_k((NQ1+2)*2:NQ1+2:(NQ1+2)*(NQ1+1),:) = DS00Y_K_i{2};
132     DS00Y_k((NQ1+2)*NQ1+1:-(NQ1+2):NQ1+3,:) = DS00Y_K_i{3};
133
134     DA00X_k(1,:) = DA00X_P_i(1,:);
135     DA00X_k(NQ1+2,:) = DA00X_P_i(2,:);
136     DA00X_k((NQ1+2)*(NQ1+1)+1,:) = DA00X_P_i(3,:);
137     DA00X_k(2:NQ1+1,:) = DA00X_K_i{1};
138     DA00X_k((NQ1+2)*2:NQ1+2:(NQ1+2)*(NQ1+1),:) = DA00X_K_i{2};
139     DA00X_k((NQ1+2)*NQ1+1:-(NQ1+2):NQ1+3,:) = DA00X_K_i{3};
140
141     DA00Y_k(1,:) = DA00Y_P_i(1,:);
142     DA00Y_k(NQ1+2,:) = DA00Y_P_i(2,:);
143     DA00Y_k((NQ1+2)*(NQ1+1)+1,:) = DA00Y_P_i(3,:);
144     DA00Y_k(2:NQ1+1,:) = DA00Y_K_i{1};
145     DA00Y_k((NQ1+2)*2:NQ1+2:(NQ1+2)*(NQ1+1),:) = DA00Y_K_i{2};
146     DA00Y_k((NQ1+2)*NQ1+1:-(NQ1+2):NQ1+3,:) = DA00Y_K_i{3};
147
148     num=0;
149     for l1=1:NQ1
150         for l2=1:NQ1
151             num=num+1;
152             S00_k(1+l1+l2*(NQ1+2),:) = S00_i(num,:);
153             A00_k(1+l1+l2*(NQ1+2),:) = A00_i(num,:);
154             DS00X_k(1+l1+l2*(NQ1+2),:) = DS00X_i(num,:);
155             DS00Y_k(1+l1+l2*(NQ1+2),:) = DS00Y_i(num,:);
156             DA00X_k(1+l1+l2*(NQ1+2),:) = DA00X_i(num,:);
157             DA00Y_k(1+l1+l2*(NQ1+2),:) = DA00Y_i(num,:);
158         end
159     end
160 end
161
162

```

```

163     if (phase00(i)~=0)
164     % Interpolation der Quadraturpunktdaten auf elastischen Elementen
165         vz1=sign(iIPdof)'*ones(1,NQ_n);
166         vz2=sign(iIPdof)'*ones(1,NQ1_n);
167
168         S(i,:)=reshape((basis_IP_n.*vz1)'*S00_basis,1,4*NQ_n);
169         A(i,:)=reshape((basis_IP_n.*vz1)'*A00_basis,1,4*NQ_n);
170         DSX(i,:)=reshape((basis_IP_n.*vz1)'*DS00X_basis,1,4*NQ_n);
171         DSY(i,:)=reshape((basis_IP_n.*vz1)'*DS00Y_basis,1,4*NQ_n);
172         DAX(i,:)=reshape((basis_IP_n.*vz1)'*DA00X_basis,1,4*NQ_n);
173         DAY(i,:)=reshape((basis_IP_n.*vz1)'*DA00Y_basis,1,4*NQ_n);
174
175
176     for l=1:3
177         S_K{l}(i,:)=reshape((basis_IP_K_n{l}.*vz2)'*S00_basis,1,4*NQ1_n);
178         A_K{l}(i,:)=reshape((basis_IP_K_n{l}.*vz2)'*A00_basis,1,4*NQ1_n);
179         DSX_K{l}(i,:)=reshape((basis_IP_K_n{l}.*vz2)'*DS00X_basis,1,4*NQ1_n);
180         DSY_K{l}(i,:)=reshape((basis_IP_K_n{l}.*vz2)'*DS00Y_basis,1,4*NQ1_n);
181         DAX_K{l}(i,:)=reshape((basis_IP_K_n{l}.*vz2)'*DA00X_basis,1,4*NQ1_n);
182         DAY_K{l}(i,:)=reshape((basis_IP_K_n{l}.*vz2)'*DA00Y_basis,1,4*NQ1_n);
183     end
184     else
185     % Interpolation der Quadraturpunktdaten auf plastischen Elementen
186         xyn_I=BaryK_n'*Koordinaten(Elemente(i,:),:);
187         xyn_K{1}=[alpha1_n',beta1_n']*Koordinaten(Elemente(i,[1,2]),:);
188         xyn_K{2}=[alpha1_n',beta1_n']*Koordinaten(Elemente(i,[2,3]),:);
189         xyn_K{3}=[alpha1_n',beta1_n']*Koordinaten(Elemente(i,[3,1]),:);
190
191     % Schleife ueber alle neuen Quadraturpunkte
192     % 1. Quadraturpunkte im Innern
193     for qnr=1:size(xyn_I,1)
194     % Quadraturpunkt Nr. qnr liegt in Element Nr. i_nr
195         i_nr=T_nr(xyn_I(qnr,:),Elemente_i,Koordinaten_i,1);
196         T_i=det([1,1,1;Koordinaten_i(Elemente_i(i_nr,:),:)]);
197         a3=(1/T_i)*det([1,1,1;Koordinaten_i(Elemente_i(i_nr,1),:)]', ...
198             Koordinaten_i(Elemente_i(i_nr,2),:)',xyn_I(qnr,:)]');
199         a1=(1/T_i)*det([1,1,1;Koordinaten_i(Elemente_i(i_nr,2),:)]', ...
200             Koordinaten_i(Elemente_i(i_nr,3),:)',xyn_I(qnr,:)]');
201         a2=(1/T_i)*det([1,1,1;Koordinaten_i(Elemente_i(i_nr,3),:)]', ...
202             Koordinaten_i(Elemente_i(i_nr,1),:)',xyn_I(qnr,:)]');
203         I_qnr=[qnr,NQ_n+qnr,2*NQ_n+qnr,3*NQ_n+qnr];
204
205         S(i,I_qnr)=[a1,a2,a3]*S00_k(Elemente_i(i_nr,:),:);
206         A(i,I_qnr)=[a1,a2,a3]*A00_k(Elemente_i(i_nr,:),:);
207         DSX(i,I_qnr)=[a1,a2,a3]*DS00X_k(Elemente_i(i_nr,:),:);
208         DSY(i,I_qnr)=[a1,a2,a3]*DS00Y_k(Elemente_i(i_nr,:),:);
209         DAX(i,I_qnr)=[a1,a2,a3]*DA00X_k(Elemente_i(i_nr,:),:);
210         DAY(i,I_qnr)=[a1,a2,a3]*DA00Y_k(Elemente_i(i_nr,:),:);
211     end
212     % 2. Quadraturpunkte auf den Kanten
213     for l1=1:3
214         for qnr=1:size(xyn_K{l1},1)
215             i_nr=T_nr(xyn_K{l1}(qnr,:),Elemente_i,Koordinaten_i,1);
216             T_i=det([1,1,1;Koordinaten_i(Elemente_i(i_nr,:),:)]);
217             a3=(1/T_i)*det([1,1,1;Koordinaten_i(Elemente_i(i_nr,1),:)]', ...

```



```

218     Koordinaten_i(Elemente_i(i_nr,2),:)',xyn_K{11}(qnr,:)]');
219     a1=(1/T_i)*det([1,1,1;Koordinaten_i(Elemente_i(i_nr,2),:)', ...
220     Koordinaten_i(Elemente_i(i_nr,3),:)',xyn_K{11}(qnr,:)]');
221     a2=(1/T_i)*det([1,1,1;Koordinaten_i(Elemente_i(i_nr,3),:)', ...
222     Koordinaten_i(Elemente_i(i_nr,1),:)',xyn_K{11}(qnr,:)]');
223     I_qnr=[qnr,NQ1+qnr,2*NQ1+qnr,3*NQ1+qnr];
224     S_K{11}(i,I_qnr)=[a1,a2,a3]*S00_k(Elemente_i(i_nr,:),:);
225     A_K{11}(i,I_qnr)=[a1,a2,a3]*A00_k(Elemente_i(i_nr,:),:);
226     DSX_K{11}(i,I_qnr)=[a1,a2,a3]*DS00X_k(Elemente_i(i_nr,:),:);
227     DSY_K{11}(i,I_qnr)=[a1,a2,a3]*DS00Y_k(Elemente_i(i_nr,:),:);
228     DAX_K{11}(i,I_qnr)=[a1,a2,a3]*DA00X_k(Elemente_i(i_nr,:),:);
229     DAY_K{11}(i,I_qnr)=[a1,a2,a3]*DA00Y_k(Elemente_i(i_nr,:),:);
230     end
231     end
232     end
233     end
234
235     S00=S;S00_K=S_K;A00=A;A00_K=A_K;
236
237     DS00X=DSX;DS00X_K=DSX_K;DS00Y=DSY;DS00Y_K=DSY_K;
238     DA00X=DAX;DA00X_K=DAX_K;DA00Y=DAY;DA00Y_K=DAY_K;
239
240     [DSOX,DSOY,DAOX,DAOY,DSOX_K,DSOY_K,DAOX_K,DAOY_K,DSOX_P,DSOY_P,...
241     DAOX_P,DAOY_P]=...
242     DSigma(u0,u00,S00,A00,S00_K,A00_K,S00_P,A00_P,DS00X,DS00Y,DA00X,DA00Y,...
243     DS00X_K,DS00Y_K,DA00X_K,DA00Y_K,DS00X_P,DS00Y_P,DA00X_P,DA00Y_P,...
244     Koordinaten,Elemente,dof_n,Gewichte_n,NQ_n,NQ1_n);
245
246
247     [S0,A0,SO_K,A0_K,SO_P,A0_P,DSOX,DSOY,DAOX,DAOY,DSOX_K,DSOY_K,DAOX_K,DAOY_K,...
248     DSOX_P,DSOY_P,DAOX_P,DAOY_P]=...
249     QPD_updaten(u0,u00,S00,A00,S00_K,A00_K,S00_P,A00_P,DSOX,DSOY,DAOX,DAOY,...
250     DSOX_K,DSOY_K,DAOX_K,DAOY_K,DSOX_P,DSOY_P,DAOX_P,DAOY_P,DS00X,DS00Y,...
251     DA00X,DA00Y,DS00X_K,DS00Y_K,DA00X_K,DA00Y_K,DS00X_P,DS00Y_P,DA00X_P,DA00Y_P,...
252     tt,Koordinaten,Elemente,dof_n,NQ_n,NQ1_n);

```

A.4 Spannungsberechnung

Die Berechnung der Spannung erfolgt nach Satz 3.3 durch

$$\sigma_{k-1+\theta} = \left(\lambda + \frac{2\mu}{d}\right) \text{tr}(\theta \Delta t_k \mathbf{A}_{k-1+\theta}) \mathbf{I} \\ + 2\mu h(|\text{dev } \theta \Delta t_k \mathbf{A}_{k-1+\theta} - \beta \boldsymbol{\alpha}_{k-1}|) \text{dev}(\theta \Delta t_k \mathbf{A}_{k-1+\theta} - \beta \boldsymbol{\alpha}_{k-1}) + \text{dev } \boldsymbol{\alpha}_{k-1}$$

und ist in der folgenden Routine `SpannungC4.c` implementiert.

```

1  #include <stdio.h>
2  #include <math.h>
3  #include <stdlib.h>
4  #include "mex.h"
5
6  double *CreateDoubleVector(int len)
7  {
8  return malloc(len*sizeof(double));

```

```
9  }
10
11  int *CreateIntVector(int len)
12  {
13      return malloc(len*sizeof(int));
14  }
15
16  double tr(double x[4])
17  {
18      double t;
19      t=x[0]+x[3];
20      return t;
21  }
22
23  double skal(double x[4], double y[4])
24  {
25      double s;
26      s=x[0]*y[0]+x[1]*y[1]+x[2]*y[2]+x[3]*y[3];
27      return s;
28  }
29
30  void symgrad(double y[4], double x[4])
31  {
32      y[0]=x[0];y[1]=(x[1]+x[2])/2.0;y[2]=y[1];y[3]=x[3];
33  }
34
35  void dev(double y[4], double x[4])
36  {
37      y[0]=x[0]-(x[0]+x[3])/2.0;
38      y[1]=x[1];
39      y[2]=x[2];
40      y[3]=x[3]-(x[0]+x[3])/2.0;
41  }
42
43  double norm(double x[4])
44  {
45      double n;
46      n=sqrt(pow(x[0],2)+pow(x[1],2)+pow(x[2],2)+pow(x[3],2));
47      return n;
48  }
49
50  double dsign(double x)
51  {
52      double s;
53      if (x>=0) s=1; else s=-1;
54      return s;
55  }
56
57
58  void SPANNUNG(double SP[], int n, double u[], double u0[],
59              double S0[], double a0[], double idof[], double nz[],
60              double dx[], double dy[], double jm[],
61              double *lambda, double *mu,
62              double *c1, double *c2, double *c3,
63              double *sy, int nq, int nd, int nd0)
```

```

64  {
65      double dtm,phi[2][2],*dxz,*dyz;
66      double dux1,duy1,dux2,duy2,Du[4],Eu[4];
67      double ds[4],da[4],v0[4],dv0[4],v1[4],dv1[4],v[4],dv[4];
68      double w1[2],w2[2],d1[4],d2[4],e1[4],e2[4],de1[4],de2[4];
69      double f1,f2,delta;
70      int *I0,l,i,j,k;
71
72      I0=CreateIntVector(nd);
73      dxz=CreateDoubleVector(nd0);dyz=CreateDoubleVector(nd0);
74
75      for (i=0;i<4*nq;i++) SP[i]=0.0;
76      dtm=jm[0]*jm[3]-jm[1]*jm[2];
77      phi[0][0]=jm[3]/dtm;
78      phi[0][1]=-jm[2]/dtm;
79      phi[1][0]=-jm[1]/dtm;
80      phi[1][1]=jm[0]/dtm;
81      for (j=0;j<nd;j++) I0[j]=abs((int) idof[(int) nz[j]-1]);
82      for (l=0;l<nq;l++)
83      {
84          for (j=0;j<nd0;j++)
85          {
86              dxz[j]=dx[j+nd0*1]*dsign(idof[j]);
87              dyz[j]=dy[j+nd0*1]*dsign(idof[j]);
88          }
89          dux1=0;duy1=0;dux2=0;duy2=0;
90          for (j=0;j<nd;j++)
91          {
92              dux1=dux1+(u[2*I0[j]-1-1]-u0[2*I0[j]-1-1])*dxz[(int) nz[j]-1];
93              duy1=duy1+(u[2*I0[j]-1-1]-u0[2*I0[j]-1-1])*dyz[(int) nz[j]-1];
94              dux2=dux2+(u[2*I0[j]-1]-u0[2*I0[j]-1])*dxz[(int) nz[j]-1];
95              duy2=duy2+(u[2*I0[j]-1]-u0[2*I0[j]-1])*dyz[(int) nz[j]-1];
96          }
97
98          Du[0]=dux1*phi[0][0]+duy1*phi[0][1];
99          Du[1]=dux1*phi[1][0]+duy1*phi[1][1];
100         Du[2]=dux2*phi[0][0]+duy2*phi[0][1];
101         Du[3]=dux2*phi[1][0]+duy2*phi[1][1];
102         symgrad(Eu,Du);
103         ds[0]=S0[1]-(S0[1]+S0[3*nq+1])/2.0;
104         ds[1]=S0[nq+1];
105         ds[2]=S0[2*nq+1];
106         ds[3]=S0[3*nq+1]-(S0[1]+S0[3*nq+1])/2.0;
107         da[0]=a0[1]-(a0[1]+a0[3*nq+1])/2.0;
108         da[1]=a0[nq+1];
109         da[2]=a0[2*nq+1];
110         da[3]=a0[3*nq+1]-(a0[1]+a0[3*nq+1])/2.0;
111         v0[0]=Eu[0]+(S0[1]+S0[3*nq+1])/(4.0*(lambda)+4.0*(mu))+
112             ds[0]/(2.0*(mu));
113         v0[1]=Eu[1]+ds[1]/(2.0*(mu));
114         v0[2]=Eu[2]+ds[2]/(2.0*(mu));
115         v0[3]=Eu[3]+(S0[1]+S0[3*nq+1])/(4.0*(lambda)+4.0*(mu))+
116             ds[3]/(2.0*(mu));
117         dev(dv0,v0);
118         v1[0]=-a0[1]/(2.0*(mu));v1[1]=-a0[nq+1]/(2.0*(mu));

```

```

119     v1[2]=-a0[2*nq+1]/(2.0*(mu));v1[3]=-a0[3*nq+1]/(2.0*(mu));
120     dev(dv1,v1);
121     v[0]=v0[0]+v1[0];v[1]=v0[1]+v1[1];
122     v[2]=v0[2]+v1[2];v[3]=v0[3]+v1[3];
123     dev(dv,v);
124
125     if (norm(dv)-(*sy)/(2*(mu))>0)
126     {
127         f1>(*c2)+(*c3)/norm(dv);f2>(*c3)/pow(norm(dv),3);
128         delta=1.0;
129     }
130     else
131     {
132         f1=2*(mu);f2=0.0;delta=0.0;
133     }
134
135     SP[1]=(*c1)*tr(v0)+f1*dv0[0]+delta*(f1*dv1[0]+da[0]);
136     SP[nq+1]=f1*dv0[1]+delta*(f1*dv1[1]+da[1]);
137     SP[2*nq+1]=f1*dv0[2]+delta*(f1*dv1[2]+da[2]);
138     SP[3*nq+1]=(*c1)*tr(v0)+f1*dv0[3]+delta*(f1*dv1[3]+da[3]);
139
140 }
141 free(I0);I0=NULL;
142 free(dxz);dxz=NULL;free(dyz);dyz=NULL;
143 }
144
145 /* Gateway function */
146
147 void mexFunction(int nlhs, mxArray *plhs[], int nrhs,
148                 const mxArray *prhs[])
149 {
150     double *S,*p,*u,*u0,*S0,*a0,*idof,*nz,*dx,*dy,*J,*Gewichte,*lambda,*mu,
151           *c1,*c2,*c3,*sy;
152     int nd,nd0,nq,n;
153
154
155     /*Check for proper number of arguments */
156     if (nrhs!=16)
157         mexErrMsgTxt("16 input required!");
158     if (nlhs!=1)
159         mexErrMsgTxt("1 output required!");
160
161     n=mxGetM(prhs[0])/2;
162     nq=mxGetN(prhs[9]);
163     nd=mxGetN(prhs[5]);
164     nd0=mxGetN(prhs[4]);
165
166     u      =mxGetPr(prhs[0]); u0      =mxGetPr(prhs[1]);
167     S0     =mxGetPr(prhs[2]); a0      =mxGetPr(prhs[3]);
168     idof   =mxGetPr(prhs[4]); nz     =mxGetPr(prhs[5]);
169     dx     =mxGetPr(prhs[6]); dy     =mxGetPr(prhs[7]);
170     J      =mxGetPr(prhs[8]); Gewichte=mxGetPr(prhs[9]);
171     lambda =mxGetPr(prhs[10]);mu     =mxGetPr(prhs[11]);
172     c1     =mxGetPr(prhs[12]);c2     =mxGetPr(prhs[13]);
173     c3     =mxGetPr(prhs[14]);sy     =mxGetPr(prhs[15]);

```

```

174
175
176 /*Set the output pointer to the output matrix. */
177 plhs[0]=mxCreateDoubleMatrix(nq,4,mxREAL);
178
179
180 /*Create a C pointer to a copy of the output matrix */
181 S=mxGetPr(plhs[0]);
182
183 /* SPANNUNG */
184 SPANNUNG(S,n,u,u0,S0,a0,idof,nz,dx,dy,J,lambda,mu,c1,c2,c3,sy,
185         nq,nd,nd0);
186 }

```

A.5 Fehlerschätzer η_R : Residuenschätzer

Berechnung der Spannungsableitungen

Die Berechnung des Volumenresiduums erfordert die Bestimmung der Ableitungen der Spannungsfunktion. Der Term η_T in (4.66) enthält den Ausdruck

$$\operatorname{div}_{\mathcal{T}} \boldsymbol{\sigma}_{h,k-1+\theta} = \begin{pmatrix} \frac{\partial(\boldsymbol{\sigma}_{h,k-1+\theta})_{11}}{\partial x} & \frac{\partial(\boldsymbol{\sigma}_{h,k-1+\theta})_{12}}{\partial y} \\ \frac{\partial(\boldsymbol{\sigma}_{h,k-1+\theta})_{21}}{\partial x} & \frac{\partial(\boldsymbol{\sigma}_{h,k-1+\theta})_{22}}{\partial y} \end{pmatrix}.$$

Die zweiten Ableitungen der Kantenbasisfunktionen auf dem Referenzelement lauten für $\ell = 1, 2, 3$ und mit $\ell + 1$ modulo 3

$$\begin{aligned} & \frac{\partial^2 \hat{N}_j^\ell(\hat{x}, \hat{y})}{\partial \hat{x}^2} \\ &= \frac{\partial^2}{\partial \hat{x}^2} (\hat{N}_\ell(\hat{x}, \hat{y}) \hat{N}_{\ell+1}(\hat{x}, \hat{y})) \psi_j (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \\ & \quad + 2 \frac{\partial}{\partial \hat{x}} (\hat{N}_\ell(\hat{x}, \hat{y}) \hat{N}_{\ell+1}(\hat{x}, \hat{y})) \psi_j' (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \frac{\partial}{\partial \hat{x}} (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \\ & \quad + \hat{N}_\ell(\hat{x}, \hat{y}) \hat{N}_{\ell+1}(\hat{x}, \hat{y}) \psi_j'' (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \left(\frac{\partial}{\partial \hat{x}} (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \right)^2 \\ & \quad + \hat{N}_\ell(\hat{x}, \hat{y}) \hat{N}_{\ell+1}(\hat{x}, \hat{y}) \psi_j' (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \frac{\partial^2}{\partial \hat{x}^2} (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})), \\ & \frac{\partial^2 \hat{N}_j^\ell(\hat{y}, \hat{y})}{\partial \hat{y}^2} \\ &= \frac{\partial^2}{\partial \hat{y}^2} (\hat{N}_\ell(\hat{x}, \hat{y}) \hat{N}_{\ell+1}(\hat{x}, \hat{y})) \psi_j (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \\ & \quad + 2 \frac{\partial}{\partial \hat{y}} (\hat{N}_\ell(\hat{x}, \hat{y}) \hat{N}_{\ell+1}(\hat{x}, \hat{y})) \psi_j' (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \frac{\partial}{\partial \hat{y}} (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \\ & \quad + \hat{N}_\ell(\hat{x}, \hat{y}) \hat{N}_{\ell+1}(\hat{x}, \hat{y}) \psi_j'' (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \left(\frac{\partial}{\partial \hat{y}} (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \right)^2 \\ & \quad + \hat{N}_\ell(\hat{x}, \hat{y}) \hat{N}_{\ell+1}(\hat{x}, \hat{y}) \psi_j' (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \frac{\partial^2}{\partial \hat{y}^2} (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})), \end{aligned}$$

$$\begin{aligned}
& \frac{\partial^2 \hat{N}_j^\ell(\hat{x}, \hat{y})}{\partial \hat{y} \partial \hat{x}} \\
&= \frac{\partial^2}{\partial \hat{y} \partial \hat{x}} (\hat{N}_\ell(\hat{x}, \hat{y}) \hat{N}_{\ell+1}(\hat{x}, \hat{y})) \psi_j (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \\
&+ \frac{\partial}{\partial \hat{x}} (\hat{N}_\ell(\hat{x}, \hat{y}) \hat{N}_{\ell+1}(\hat{x}, \hat{y})) \psi_j' (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \frac{\partial}{\partial \hat{y}} (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \\
&+ \frac{\partial}{\partial \hat{y}} (\hat{N}_\ell(\hat{x}, \hat{y}) \hat{N}_{\ell+1}(\hat{x}, \hat{y})) \psi_j' (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \frac{\partial}{\partial \hat{x}} (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \\
&+ \hat{N}_\ell(\hat{x}, \hat{y}) \hat{N}_{\ell+1}(\hat{x}, \hat{y}) \psi_j'' (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \\
&\quad \times \frac{\partial}{\partial \hat{y}} (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \frac{\partial}{\partial \hat{x}} (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \\
&+ \hat{N}_\ell(\hat{x}, \hat{y}) \hat{N}_{\ell+1}(\hat{x}, \hat{y}) \psi_j' (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})) \frac{\partial^2}{\partial \hat{y} \partial \hat{x}} (\hat{N}_{\ell+1}(\hat{x}, \hat{y}) - \hat{N}_\ell(\hat{x}, \hat{y})).
\end{aligned}$$

Die zweiten Ableitungen der internen Basisfunktionen auf dem Referenzelement lauten

$$\begin{aligned}
\frac{\partial^2 \hat{N}_{i,j}(\hat{x}, \hat{y})}{\partial \hat{x}^2} &= \frac{\partial^2}{\partial \hat{x}^2} (\hat{N}_1(\hat{x}, \hat{y}) \hat{N}_2(\hat{x}, \hat{y}) \hat{N}_3(\hat{x}, \hat{y})) P_i(\hat{x}) P_j\left(\frac{2}{\sqrt{3}} \hat{y} - 1\right) \\
&+ 2 \frac{\partial}{\partial \hat{x}} (\hat{N}_1(\hat{x}, \hat{y}) \hat{N}_2(\hat{x}, \hat{y}) \hat{N}_3(\hat{x}, \hat{y})) \left(\frac{\partial}{\partial \hat{x}} P_i(\hat{x})\right) P_j\left(\frac{2}{\sqrt{3}} \hat{y} - 1\right) \\
&+ \hat{N}_1(\hat{x}, \hat{y}) \hat{N}_2(\hat{x}, \hat{y}) \hat{N}_3(\hat{x}, \hat{y}) \left(\frac{\partial^2}{\partial \hat{x}^2} P_i(\hat{x})\right) P_j\left(\frac{2}{\sqrt{3}} \hat{y} - 1\right),
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 \hat{N}_{i,j}(\hat{x}, \hat{y})}{\partial \hat{y}^2} &= \frac{\partial^2}{\partial \hat{y}^2} (\hat{N}_1(\hat{x}, \hat{y}) \hat{N}_2(\hat{x}, \hat{y}) \hat{N}_3(\hat{x}, \hat{y})) P_i(\hat{x}) P_j\left(\frac{2}{\sqrt{3}} - 1\right) \\
&+ 2 \frac{\partial}{\partial \hat{y}} (\hat{N}_1(\hat{x}, \hat{y}) \hat{N}_2(\hat{x}, \hat{y}) \hat{N}_3(\hat{x}, \hat{y})) P_x(\hat{x}) \frac{\partial}{\partial \hat{y}} P_j\left(\frac{2}{\sqrt{3}} \hat{y} - 1\right) \\
&+ \hat{N}_1(\hat{x}, \hat{y}) \hat{N}_2(\hat{x}, \hat{y}) \hat{N}_3(\hat{x}, \hat{y}) P_i(\hat{x}) \frac{\partial^2}{\partial \hat{y}^2} P_j\left(\frac{2}{\sqrt{3}} \hat{y} - 1\right),
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 \hat{N}_{i,j}(\hat{x}, \hat{y})}{\partial \hat{x} \partial \hat{y}} &= \frac{\partial^2}{\partial \hat{x} \partial \hat{y}} (\hat{N}_1(\hat{x}, \hat{y}) \hat{N}_2(\hat{x}, \hat{y}) \hat{N}_3(\hat{x}, \hat{y})) P_i(\hat{x}) P_j\left(\frac{2}{\sqrt{3}} \hat{y} - 1\right) \\
&+ \frac{\partial}{\partial \hat{y}} (\hat{N}_1(\hat{x}, \hat{y}) \hat{N}_2(\hat{x}, \hat{y}) \hat{N}_3(\hat{x}, \hat{y})) \left(\frac{\partial}{\partial \hat{x}} P_i(\hat{x})\right) P_j\left(\frac{2}{\sqrt{3}} \hat{y} - 1\right) \\
&+ \frac{\partial}{\partial \hat{x}} (\hat{N}_1(\hat{x}, \hat{y}) \hat{N}_2(\hat{x}, \hat{y}) \hat{N}_3(\hat{x}, \hat{y})) P_i(\hat{x}) \frac{\partial}{\partial \hat{y}} P_j\left(\frac{2}{\sqrt{3}} \hat{y} - 1\right) \\
&+ \hat{N}_1(\hat{x}, \hat{y}) \hat{N}_2(\hat{x}, \hat{y}) \hat{N}_3(\hat{x}, \hat{y}) \left(\frac{\partial}{\partial \hat{x}} P_i(\hat{x})\right) \left(\frac{\partial}{\partial \hat{y}} P_j\left(\frac{2}{\sqrt{3}} \hat{y} - 1\right)\right).
\end{aligned}$$

Die Routine `Gradient2.m` berechnet die 2. Ableitungen der Basisfunktionen auf dem Referenzelement. Die Implementierung baut auf die Arbeit [18] von C. Boecker

auf, in der die Implementierung der ersten Ableitungen der Basisfunktionen mit den benötigten Unterroutrinen zur Ableitung der Legendre-Polynome dargestellt ist.

```

1  function [dxx,dxy,dy] = Gradient2(xy,p);
2  % Berechnet die Werte der 2.Ableitungen der Basisfunktionen an der Stelle (x,y)
3  % bis zum Polynomgrad p
4  % xy=[x,y]
5  max_p=p;
6  w3=sqrt(3);
7  % Dimension der zu berechnenden Felder
8  dy=zeros(3*max_p+((max_p-1)*(max_p-2)/2),size(xy,1));
9  dxx=zeros(3*max_p+((max_p-1)*(max_p-2)/2),size(xy,1));
10 dxy=zeros(3*max_p+((max_p-1)*(max_p-2)/2),size(xy,1));
11 G=zeros(3,2);
12 P=zeros(3,1);
13 S=zeros(3,1);
14 GS=zeros(3,2);
15 % Ableitungen an den Ecken
16 dxx(1,:)=0;dxy(1,:)=0;dy(1,:)=0;
17 dxx(2,:)=0;dxy(2,:)=0;dy(2,:)=0;
18 dxx(3,:)=0;dxy(3,:)=0;dy(3,:)=0;
19 %2.Ableitungen vom Produkt zweier Basisfunktionen Ni*Nj, i,j=1..3
20 GP2(1,:)=[-1/2,1/6];
21 GP2(2,:)=[0,-1/3];
22 GP2(3,:)=[0,-1/3];
23 %Gemischte Ableitungen vom Produkt zweier Basisfunktionen
24 GPxy(1)=0;
25 GPxy(2)=1/(2*w3);
26 GPxy(3)=-1/(2*w3);
27 %1.Ableitungen der Differenz zweier Basisfunktionen Ni-Nj, i,j=1..3
28 GD1(1,:)=[1,0];
29 GD1(2,:)=[-0.5,w3/2];
30 GD1(3,:)=[-0.5,-w3/2];
31 %2.Ableitungen der Differenz zweier Basisfunktionen Ni-Nj, i,j=1..3
32 GD2(1,:)=[0,0];
33 GD2(2,:)=[0,0];
34 GD2(3,:)=[0,0];
35 %Gemischte Ableitungen der Differenz zweier Basisfunktionen
36 GDxy(1)=0;
37 GDxy(2)=0;
38 GDxy(3)=0;
39 for l=1:size(xy,1)
40 %1. Ableitungen vom Produkt zweier Basisfunktionen Ni*Nj, i,j=1..3,
41 GP1(1,:)=[-xy(1,1)/2 , xy(1,2)/6 - w3/6]; % Gradient(N1*N2)
42 GP1(2,:)=[xy(1,2)*w3/6 , w3/6 + xy(1,1)*w3/6 - xy(1,2)/3]; % Gradient(N2*N3)
43 GP1(3,:)=[-xy(1,2)*w3/6 , w3/6 - xy(1,1)*w3/6 - xy(1,2)/3]; % Gradient(N3*N1)
44 % Produkt zweier Basisfunktionen Ni*Nj, i,j=1..3, i<>j
45 P(1)=.25*(1-(2/w3)*xy(1,2)-xy(1,1)^2+xy(1,2)^2/3); % N1*N2
46 P(2)=.5*( xy(1,2)/w3+xy(1,1)*xy(1,2)/w3-xy(1,2)^2/3); % N2*N3
47 P(3)=.5*(xy(1,2)/w3-xy(1,1)*xy(1,2)/w3-xy(1,2)^2/3); % N3*N1
48 % Summe zweier Basisfunktionen Ni-Nj, i,j=1..3, i<>j
49 S(1,1)=xy(1,1); % N2-N1
50 S(2,1)=.5*(-1-xy(1,1)+w3*xy(1,2)); % N3-N2
51 S(3,1)=.5*(1-xy(1,1)-w3*xy(1,2)); % N1-N3
52 lfd=4;

```

```

53 % Berechne gradient auf Kanten
54 for i=1:3
55     phi_j=phi(S(i,1),max_p);
56     abl_phi_j=abl_phi2(S(i,1),max_p);
57     abl2_phi_j=abl2_phi2(S(i,1),max_p);
58     for j=2:max_p
59         dxx(lfd,l)=GP2(i,1)*phi_j(j)+2*GP1(i,1)*GD1(i,1)*abl_phi_j(j)+...
60             P(i)*abl2_phi_j(j)*GD1(i,1)^2+P(i)*abl_phi_j(j)*GD2(i,1);
61         dyy(lfd,l)=GP2(i,2)*phi_j(j)+2*GP1(i,2)*GD1(i,2)*abl_phi_j(j)+...
62             P(i)*abl2_phi_j(j)*GD1(i,2)^2+P(i)*abl_phi_j(j)*GD2(i,2);
63         dxy(lfd,l)=GPxy(i)*phi_j(j)+GP1(i,1)*abl_phi_j(j)*GD1(i,2)+...
64             GP1(i,2)*abl_phi_j(j)*GD1(i,1)+P(i)*abl2_phi_j(j)*...
65             GD1(i,2)*GD1(i,1);
66         lfd=lfd+1;
67     end
68 end
69 % Berechne gradient im Innern
70 P123=P(1,1)*xy(1,2)/w3; % N1*N2*N3
71 GP123=[-w3/6*xy(1,1)*xy(1,2), -w3/36*(4*w3*xy(1,2)-3*xy(1,2)^2-3*xy(1,1)^2)];
72 P_i=Legend(xy(1,1),max_p-2);
73 P_j=Legend(2*xy(1,2)/w3-1,max_p-2);
74 GP_i=abl_legend(xy(1,1),max_p-2);
75 GP_j=abl_legend(2*xy(1,2)/w3-1,max_p-2)*(2/w3);
76 G2P_i=abl2_legend(xy(1,1),max_p-2);
77 G2P_j=abl2_legend(2*xy(1,2)/w3-1,max_p-2)*4/3;
78 G2P123=[-xy(1,2)/(2*w3), (-2+w3*xy(1,2))/6];
79 G2xyP123=-xy(1,1)/(2*w3);
80 for k=0:max_p-3
81     i=0;
82     for j=k:-1:0
83         dxx(lfd,l)=G2P123(1)*P_i(i+1)*P_j(j+1)+2*GP123(1)*GP_i(i+1)*P_j(j+1)+...
84             P123*G2P_i(i+1)*P_j(j+1);
85         dyy(lfd,l)=G2P123(2)*P_i(i+1)*P_j(j+1)+2*GP123(2)*P_i(i+1)*GP_j(j+1)+...
86             P123*P_i(i+1)*G2P_j(j+1);
87         dxy(lfd,l)=G2xyP123*P_i(i+1)*P_j(j+1)+GP123(2)*GP_i(i+1)*P_j(j+1)+...
88             GP123(1)*P_i(i+1)*GP_j(j+1)+P123*GP_i(i+1)*GP_j(j+1);
89         lfd=lfd+1;
90         i=i+1;
91     end
92 end
93 end

```

Ist $\hat{u}(\hat{x}, \hat{y})$ eine Funktion, die auf dem Referenzelement \hat{T} definiert ist und $J : \hat{T} \rightarrow T$ die Transformationsmatrix, erhält man die 1. Ableitungen bezüglich Element T durch

$$\begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{pmatrix} = J^{-1} \begin{pmatrix} \frac{\partial \hat{u}}{\partial \hat{x}} \\ \frac{\partial \hat{u}}{\partial \hat{y}} \end{pmatrix} = \begin{pmatrix} (J^{-1})_{11} \frac{\partial \hat{u}}{\partial \hat{x}} + (J^{-1})_{12} \frac{\partial \hat{u}}{\partial \hat{y}} \\ (J^{-1})_{21} \frac{\partial \hat{u}}{\partial \hat{x}} + (J^{-1})_{22} \frac{\partial \hat{u}}{\partial \hat{y}} \end{pmatrix}.$$

Die 2. Ableitungen transformieren sich analog durch

$$\begin{pmatrix} \frac{\partial}{\partial x} \left(\frac{\partial u}{\partial x} \right) \\ \frac{\partial}{\partial y} \left(\frac{\partial u}{\partial x} \right) \end{pmatrix} = J^{-1} \begin{pmatrix} \frac{\partial}{\partial \hat{x}} \left((J^{-1})_{11} \frac{\partial \hat{u}}{\partial \hat{x}} + (J^{-1})_{12} \frac{\partial \hat{u}}{\partial \hat{y}} \right) \\ \frac{\partial}{\partial \hat{y}} \left((J^{-1})_{11} \frac{\partial \hat{u}}{\partial \hat{x}} + (J^{-1})_{12} \frac{\partial \hat{u}}{\partial \hat{y}} \right) \end{pmatrix} = J^{-1} \begin{pmatrix} (J^{-1})_{11} \frac{\partial^2 \hat{u}}{\partial \hat{x}^2} + (J^{-1})_{12} \frac{\partial^2 \hat{u}}{\partial \hat{x} \partial \hat{y}} \\ (J^{-1})_{11} \frac{\partial^2 \hat{u}}{\partial \hat{y} \partial \hat{x}} + (J^{-1})_{12} \frac{\partial^2 \hat{u}}{\partial \hat{y}^2} \end{pmatrix}$$

$$\begin{pmatrix} \frac{\partial}{\partial x} \left(\frac{\partial u}{\partial y} \right) \\ \frac{\partial}{\partial y} \left(\frac{\partial u}{\partial x} \right) \end{pmatrix} = J^{-1} \begin{pmatrix} (J^{-1})_{21} \frac{\partial^2 \hat{u}}{\partial \hat{x}^2} + (J^{-1})_{22} \frac{\partial^2 \hat{u}}{\partial \hat{x} \partial \hat{y}} \\ (J^{-1})_{11} \frac{\partial^2 \hat{u}}{\partial \hat{y} \partial \hat{x}} + (J^{-1})_{12} \frac{\partial^2 \hat{u}}{\partial \hat{y}^2} \end{pmatrix}.$$

Die Transformation läßt sich in Matrixschreibweise angeben,

$$\begin{pmatrix} \frac{\partial^2 u}{\partial x^2} & \frac{\partial^2 u}{\partial x \partial y} \\ \frac{\partial^2 u}{\partial y \partial x} & \frac{\partial^2 u}{\partial y^2} \end{pmatrix} = J^{-1} \begin{pmatrix} \frac{\partial^2 \hat{u}}{\partial \hat{x}^2} & \frac{\partial^2 \hat{u}}{\partial \hat{x} \partial \hat{y}} \\ \frac{\partial^2 \hat{u}}{\partial \hat{y} \partial \hat{x}} & \frac{\partial^2 \hat{u}}{\partial \hat{y}^2} \end{pmatrix} J^{-T}.$$

Die Unteroutine `DSigma.m` berechnet die partiellen Ableitungen der Spannung und der internen Spannung in den Quadraturpunkten der Elemente im Innern, auf den Kanten und auf den Eckpunkten der Elemente.

```

1 function [DSX,DSY,DAX,DAY,DSX_K,DSY_K,DAX_K,DAY_K,DSX_P,DSY_P,DAX_P,DAY_P]=...
2 DSigma(u,u0,S0,A0,S0_K,A0_K,S0_P,A0_P,DSOX,DSOY,DAOX,DAOY,DSOX_K,DSOY_K,...
3   DAOX_K,DAOY_K,DSOX_P,DSOY_P,DAOX_P,DAOY_P,Koordinaten,Elemente,dof)
4
5 global NQ NQ1 dx dy dx_K dy_K dx_P dy_P dxx dxy dyy dxx_K dxy_K dyy_K ...
6   dxx_P dxy_P dyy_P sy C1 C2 C3 k1 lambda mu Gewichte
7
8 NE=size(Elemente,1);
9
10 DSX=zeros(NE,4*NQ);DSY=zeros(NE,4*NQ);
11 DAX=zeros(NE,4*NQ);DAY=zeros(NE,4*NQ);
12
13 DSX2=zeros(NE,4*NQ);DSY2=zeros(NE,4*NQ);
14 DAX2=zeros(NE,4*NQ);DAY2=zeros(NE,4*NQ);
15
16
17 DSX_K{1}=zeros(NE,4*NQ1);DSY_K{1}=zeros(NE,4*NQ1);
18 DSX_K{2}=zeros(NE,4*NQ1);DSY_K{2}=zeros(NE,4*NQ1);
19 DSX_K{3}=zeros(NE,4*NQ1);DSY_K{3}=zeros(NE,4*NQ1);
20
21 DSX_P=zeros(NE,12);DSY_P=zeros(NE,12);
22 DAX_P=zeros(NE,12);DAY_P=zeros(NE,12);
23
24
25 for j=1:size(Elemente,1)
26   J=Funktionalmatrix(Koordinaten,Elemente,j);
27   jdof=dof(j,:);nz=find(jdof);I=abs(jdof(nz));
28
29   [DSX(j,:),DSY(j,:)] = DSigmaC3(u,u0,S0(j,:),A0(j,:),...
30     reshape(DSOX(j,:),NQ,4),reshape(DSOY(j,:),NQ,4),...
31     reshape(DAOX(j,:),NQ,4),reshape(DAOY(j,:),NQ,4),...
32     J,jdof,nz,I,dx,dy,dxx,dxy,dyy,NQ,lambda,mu,...
33     C1,C2,C3,sy,k1);
34
35   [DAX(j,:),DAY(j,:)] = DAlphaC3(u,u0,S0(j,:),A0(j,:),...
36     reshape(DSX(j,:),NQ,4),reshape(DSY(j,:),NQ,4),...
37     reshape(DSOX(j,:),NQ,4),reshape(DSOY(j,:),NQ,4),...
38     reshape(DAOX(j,:),NQ,4),reshape(DAOY(j,:),NQ,4),...
39     J,jdof,nz,I,dxx,dxy,dyy,NQ,mu,k1);
40
41 for k=1:3
42   [DSX_K{k}(j,:),DSY_K{k}(j,:)] = ...

```

```

43   DSigmaC3(u,u0,S0_K{k}(j,:),A0_K{k}(j,:),...
44   reshape(DSOX_K{k}(j,:),NQ1,4),reshape(DSOY_K{k}(j,:),NQ1,4),...
45   reshape(DAOX_K{k}(j,:),NQ1,4),reshape(DAOY_K{k}(j,:),NQ1,4),...
46   J,jdof,nz,I,dx_K{k},dy_K{k},dxx_K{k},dxy_K{k},dyy_K{k},NQ1,...
47   lambda,mu,C1,C2,C3,sy,k1);
48
49   [DAX_K{k}(j,:),DAY_K{k}(j,.)]=DAlphac3(u,u0,S0_K{k}(j,:),A0_K{k}(j,:),...
50   reshape(DSX_K{k}(j,:),NQ1,4),reshape(DSY_K{k}(j,:),NQ1,4),...
51   reshape(DSOX_K{k}(j,:),NQ1,4),reshape(DSOY_K{k}(j,:),NQ1,4),...
52   reshape(DAOX_K{k}(j,:),NQ1,4),reshape(DAOY_K{k}(j,:),NQ1,4),...
53   J,jdof,nz,I,dxx_K{k},dxy_K{k},dyy_K{k},NQ1,mu,k1);
54
55   end
56   [DSX_P(j,:),DSY_P(j,.)]=DSigmaC3(u,u0,S0_P(j,:),A0_P(j,:),...
57   reshape(DSOX_P(j,:),3,4),reshape(DSOY_P(j,:),3,4),...
58   reshape(DAOX_P(j,:),3,4),reshape(DAOY_P(j,:),3,4),...
59   J,jdof,nz,I,dx_P,dy_P,dxx_P,dxy_P,dyy_P,3,lambda,mu,...
60   C1,C2,C3,sy,k1);
61
62   [DAX_P(j,:),DAY_P(j,.)]=DAlphac3(u,u0,S0_P(j,:),A0_P(j,:),...
63   reshape(DSX_P(j,:),3,4),reshape(DSY_P(j,:),3,4),...
64   reshape(DSOX_P(j,:),3,4),reshape(DSOY_P(j,:),3,4),...
65   reshape(DAOX_P(j,:),3,4),reshape(DAOY_P(j,:),3,4),...
66   J,jdof,nz,I,dxx_P,dxy_P,dyy_P,3,mu,k1);
67
68
69   end

```

Die Unteroutine `DSigmaC3c` berechnet die Ableitung der Spannung. Im Fall der kinematischen Verfestigung hat der Spannungstensor nach (3.3) die Darstellung

$$\begin{aligned}
\boldsymbol{\sigma}_{h,k-1+\theta} = & (\lambda + \mu) \operatorname{tr}(\theta \Delta t_k \mathbf{A}_{h,k-1+\theta}) \mathbf{I} \\
& + 2\mu f \left(\underbrace{|\operatorname{dev} \theta \Delta t_k \mathbf{A}_{h,k-1+\theta} - \beta \boldsymbol{\alpha}_{h,k-1}|}_{=: \mathbf{Z}_h} \right) \underbrace{\operatorname{dev}(\theta \Delta t_k \mathbf{A}_{h,k-1+\theta} - \beta \boldsymbol{\alpha}_{h,k-1})}_{=: \mathbf{Z}_h} \\
& + \operatorname{dev} \boldsymbol{\alpha}_{h,k-1},
\end{aligned} \tag{1.1}$$

mit $\theta \Delta t_k \mathbf{A}_{h,k-1+\theta} = \boldsymbol{\epsilon}(\mathbf{u}_{h,k-1+\theta} - \mathbf{u}_{h,k-1}) + \mathbb{C}^{-1} \boldsymbol{\sigma}_{h,k-1}$. Die Ortsableitung der Spannung erfordert die Berechnung der Ausdrücke

$$\frac{\partial}{\partial x} \boldsymbol{\epsilon}(\bar{\mathbf{u}}_h) = \begin{pmatrix} \frac{\partial^2 \bar{u}_1}{\partial x^2} & \frac{1}{2} \left(\frac{\partial^2 \bar{u}_2}{\partial x^2} + \frac{\partial^2 \bar{u}_1}{\partial x \partial y} \right) \\ \frac{1}{2} \left(\frac{\partial^2 \bar{u}_2}{\partial x^2} + \frac{\partial^2 \bar{u}_1}{\partial x \partial y} \right) & \frac{\partial^2 \bar{u}_2}{\partial x \partial y} \end{pmatrix}$$

und

$$\frac{\partial}{\partial y} \boldsymbol{\epsilon}(\bar{\mathbf{u}}_h) = \begin{pmatrix} \frac{\partial^2 \bar{u}_1}{\partial y \partial x} & \frac{1}{2} \left(\frac{\partial^2 \bar{u}_2}{\partial y \partial x} + \frac{\partial^2 \bar{u}_1}{\partial y^2} \right) \\ \frac{1}{2} \left(\frac{\partial^2 \bar{u}_2}{\partial y \partial x} + \frac{\partial^2 \bar{u}_1}{\partial y^2} \right) & \frac{\partial^2 \bar{u}_2}{\partial y^2} \end{pmatrix}$$

mit $\bar{\mathbf{u}} = \mathbf{u}_{h,k-1+\theta} - \mathbf{u}_{h,k-1}$. Dies erfolgt in den Zeilen 183-203 in dem folgenden Programm. Dabei werden die Ableitungen in den Zeilen 183-202 zunächst für das Referenzelement berechnet und dann in der Zeile 203 auf das aktuelle Element transformiert. Die Ableitung der Größe \mathbf{Z}_h (in (1.1) definiert) wird in den folgenden

Zeilen 205-225 bestimmt. In den Zeilen 229-234 wird die Ableitung des Spuranteils des Spannungstensors berechnet, in den Zeilen 237-245 die Ableitung des letzten Summanden in (1.1). Der Deviatoranteil des Spannungstensors bis auf den Anteil $\text{dev}\boldsymbol{\alpha}_{h,k-1}$ wird im folgenden mit $\text{dev}\tilde{\boldsymbol{\sigma}}_{h,k-1+\theta}$ bezeichnet. Es gilt

$$\text{dev}\tilde{\boldsymbol{\sigma}}_{h,k-1+\theta} = 2\mu h(|\mathbf{Z}_h|)\mathbf{Z}_h = 2\mu(1 - \max\{0, \frac{1}{1 + \Gamma_1}(1 - \frac{\Gamma_2}{2\mu|\mathbf{Z}_h|})\})\mathbf{Z}_h. \quad (1.2)$$

Nach der Kettenregel gilt

$$\frac{\partial \text{dev}\tilde{\boldsymbol{\sigma}}_{h,k-1+\theta}}{\partial x} = \sum_{i,j=1}^2 \frac{\partial \text{dev}\tilde{\boldsymbol{\sigma}}_{h,k-1+\theta}}{\partial \mathbf{Z}_{h,ij}} \frac{\partial \mathbf{Z}_{h,ij}}{\partial x}. \quad (1.3)$$

Die Ausdrücke $\partial \text{dev}\tilde{\boldsymbol{\sigma}}_{h,k-1+\theta}/\partial \mathbf{Z}_{h,ij}$ berechnet die Unteroutine `Ab1Sigma`, die in den Zeilen 247-248 aufgerufen wird. Die Zeilen 101-112 berechnen die Größen $\partial \bar{\mathbf{u}}/\partial x$ und $\partial \bar{\mathbf{u}}/\partial y$ zunächst für das Referenzelement und dann durch Transformation auf das aktuelle Element. Die Zeilen 114-121 berechnen die Komponenten der Matrix \mathbf{Z}_h , Zeile 125 den Betrag.

Im elastischen Fall gilt

$$\frac{\partial \text{dev}\tilde{\boldsymbol{\sigma}}_{ij}}{\partial \mathbf{Z}_{h,kl}} = 2\mu \varrho_{ik} \varrho_{jl}, \quad (1.4)$$

wie in Zeile 123 berechnet. Im plastischen Fall werden die Ableitungen

$$\frac{\partial f(|\mathbf{Z}_h|)}{\partial \mathbf{Z}_{h,ij}} \quad (1.5)$$

in den Zeilen 128-135 berechnet, wobei f wie in (1.1) definiert ist.

Schließlich wird in der aufrufenden Funktion `DSigma3` der Ausdruck (1.3) in den Zeilen 250-266 ausgewertet.

```

1  #include <stdio.h>
2  #include <math.h>
3  #include <stdlib.h>
4  #include "mex.h"
5
6  double **CreateDoubleMatrix(int m, int n)
7  {
8      double **mat=malloc(m*sizeof(double*));
9      int row=m;
10     while (row-->0) mat[row]=malloc(n*sizeof(double));
11     return mat;
12 }
13
14 double *CreateDoubleVector(int len)
15 {
16     return malloc(len*sizeof(double));
17 }
18
19 int *CreateIntVector(int len)
20 {
21     return malloc(len*sizeof(int));

```

```
22  }
23
24  double tr(double x[4])
25  {
26    double t;
27    t=x[0]+x[3];
28    return t;
29  }
30
31  double skal(double x[4], double y[4])
32  {
33    double s;
34    s=x[0]*y[0]+x[1]*y[1]+x[2]*y[2]+x[3]*y[3];
35    return s;
36  }
37
38  void Matmult(double Du[2][2],double x[2][2], double y[2][2])
39  {
40    double tmp[2][2];
41
42    tmp[0][0]=x[0][0]*y[0][0]+x[0][1]*y[0][1];
43    tmp[1][0]=x[1][0]*y[0][0]+x[1][1]*y[0][1];
44    tmp[0][1]=x[0][0]*y[1][0]+x[0][1]*y[1][1];
45    tmp[1][1]=x[1][0]*y[1][0]+x[1][1]*y[1][1];
46
47    Du[0][0]=y[0][0]*tmp[0][0]+y[0][1]*tmp[1][0];
48    Du[0][1]=y[0][0]*tmp[0][1]+y[0][1]*tmp[1][1];
49    Du[1][0]=y[1][0]*tmp[0][0]+y[1][1]*tmp[1][0];
50    Du[1][1]=y[1][0]*tmp[0][1]+y[1][1]*tmp[1][1];
51
52  }
53
54
55  void symgrad(double y[4], double x[4])
56  {
57    y[0]=x[0];y[1]=(x[1]+x[2])/2.0;y[2]=y[1];y[3]=x[3];
58  }
59
60  void dev(double y[4], double x[4])
61  {
62    y[0]=x[0]-(x[0]+x[3])/2.0;y[1]=x[1];y[2]=x[2];y[3]=x[3]-(x[0]+x[3])/2.0;
63  }
64
65  double norm(double x[4])
66  {
67    double n;
68    n=sqrt(pow(x[0],2)+pow(x[1],2)+pow(x[2],2)+pow(x[3],2));
69    return n;
70  }
71
72  double dsign(double x)
73  {
74    double s;
75    if (x>=0) s=1; else s=-1;
76    return s;
```

```

77 }
78
79 void AblSigma(double DSZ11[], double DSZ12[], double DSZ21[], double DSZ22[],
80             int l, double u[], double u0[], double S0[], double A0[],
81             double jdof[], double nz[], int IO[], double dx[], double dy[],
82             double IJM[2][2], int NQ, int ND, int ND0, double C1, double C2,
83             double C3, double lambda, double mu, double k1, double sy)
84 {
85     double g1,g2,g3,f11,f12,f21,f22,b,f;
86     double *dxz,*dyz,dux1,dux2,duy1,duy2,Z[4],Du1[2],Du2[2];
87     int i,j;
88
89     dxz=CreateDoubleVector(ND0);dyz=CreateDoubleVector(ND0);
90
91     g1=k1/(2*mu);g2=sy;g3=g2/(1+g1);
92     for (i=0;i<4;i++)
93     {
94         DSZ11[i]=0.0;DSZ12[i]=0.0;DSZ21[i]=0.0;DSZ22[i]=0.0;
95     }
96
97     for (j=0;j<ND0;j++)
98     {
99         dxz[j]=dx[j+ND0*1]*dsign(jdof[j]);dyz[j]=dy[j+ND0*1]*dsign(jdof[j]);
100    }
101    dux1=0;duy1=0;dux2=0;duy2=0;
102    for (j=0;j<ND;j++)
103    {
104        dux1=dux1+(u[2*IO[j]-1-1]-u0[2*IO[j]-1-1])*dxz[(int) nz[j]-1];
105        duy1=duy1+(u[2*IO[j]-1-1]-u0[2*IO[j]-1-1])*dyz[(int) nz[j]-1];
106        dux2=dux2+(u[2*IO[j]-1]-u0[2*IO[j]-1])*dxz[(int) nz[j]-1];
107        duy2=duy2+(u[2*IO[j]-1]-u0[2*IO[j]-1])*dyz[(int) nz[j]-1];
108    }
109    Du1[0]=dux1*IJM[0][0]+duy1*IJM[0][1];
110    Du1[1]=dux1*IJM[1][0]+duy1*IJM[1][1];
111    Du2[0]=dux2*IJM[0][0]+duy2*IJM[0][1];
112    Du2[1]=dux2*IJM[1][0]+duy2*IJM[1][1];
113
114    Z[0]=(Du1[0]-Du2[1])/2.0+1.0/(2.0*mu)*(S0[1]-S0[3*NQ+1])/2.0-
115        1.0/(2.0*mu)*(A0[1]-A0[3*NQ+1])/2.0;
116    Z[1]=(Du1[1]+Du2[0])/2.0+
117        1.0/(2.0*mu)*S0[NQ+1]-1.0/(2.0*mu)*A0[NQ+1];
118    Z[2]=(Du1[1]+Du2[0])/2.0+
119        1.0/(2.0*mu)*S0[2*NQ+1]-1.0/(2.0*mu)*A0[2*NQ+1];
120    Z[3]=(Du2[1]-Du1[0])/2.0+1.0/(2.0*mu)*(S0[3*NQ+1]-S0[1])/2.0-
121        1.0/(2.0*mu)*(A0[3*NQ+1]-A0[1])/2.0;
122
123    DSZ11[0]=2*mu;DSZ12[1]=2*mu;DSZ21[2]=2*mu;DSZ22[3]=2*mu;
124
125    b=sqrt(pow(Z[0],2.0)+pow(Z[1],2.0)+pow(Z[2],2.0)+pow(Z[3],2.0));
126    if (b-sy/(2.0*mu)>0.0)
127    {
128        f11=-g3*Z[0]/pow(b,3.0);f12=-g3*Z[1]/pow(b,3.0);
129        f21=-g3*Z[2]/pow(b,3.0);f22=-g3*Z[3]/pow(b,3.0);
130        f=2*mu*(1.0-1.0/(1.0+g1)*(1.0-g2/(2*mu*b)));
131

```

```

132     DSZ11[0]=f+f11*Z[0];DSZ11[1]=f12*Z[0];DSZ11[2]=f21*Z[0];DSZ11[3]=f22*Z[0];
133     DSZ12[0]=f11*Z[1];DSZ12[1]=f+f12*Z[1];DSZ12[2]=f21*Z[1];DSZ12[3]=f22*Z[1];
134     DSZ21[0]=f11*Z[2];DSZ21[1]=f12*Z[2];DSZ21[2]=f+f21*Z[2];DSZ22[3]=f22*Z[2];
135     DSZ22[0]=f11*Z[3];DSZ22[1]=f12*Z[3];DSZ22[2]=f21*Z[3];DSZ22[3]=f+f22*Z[3];
136 }
137 free(dxz);dxz=NULL;free(dyz);dyz=NULL;
138 }
139
140
141 void DSigma3(double DSX[],double DSY[],double u[],double u0[],
142             double S0[],double A0[],double DSOX[],
143             double DSOY[],double DAOX[],double DAOY[],
144             double JM[],double jdof[],double nz[],double I[],
145             double dx[],double dy[],
146             double dxx[],double dxy[], double dyy[],
147             int NQ,int ND,int NDO,
148             double *lambda, double *mu,
149             double *C1,double *C2,double *C3,double *sy,double *k1)
150 {
151     double DT,IJM[2][2],Du1[2][2],Du2[2][2],Du1j[2][2],Du2j[2][2],
152           **dxxz,**dxyz,**dyyz,Zx[4],Zy[4];
153     double DSZ11[4],DSZ12[4],DSZ21[4],DSZ22[4];
154     int *I0,i,j,l;
155
156     I0=CreateIntVector(ND);
157     dxxz=CreateDoubleMatrix(NDO,NQ);
158     dxyz=CreateDoubleMatrix(NDO,NQ);
159     dyyz=CreateDoubleMatrix(NDO,NQ);
160
161     for (i=0;i<4*NQ;i++)
162     {
163         DSX[i]=0.0;DSY[i]=0.0;
164     }
165
166     DT=JM[0]*JM[3]-JM[1]*JM[2];
167     IJM[0][0]=JM[3]/DT; IJM[0][1]=-JM[2]/DT;
168     IJM[1][0]=-JM[1]/DT;IJM[1][1]=JM[0]/DT;
169
170     for (j=0;j<ND;j++) I0[j]=(int) I[j];
171     for (j=0;j<NDO;j++)
172     {
173         for (l=0;l<NQ;l++)
174         {
175             dxxz[j][l]=dxx[j+NDO*1]*dsign(jdof[j]);
176             dxyz[j][l]=dxy[j+NDO*1]*dsign(jdof[j]);
177             dyyz[j][l]=dyy[j+NDO*1]*dsign(jdof[j]);
178         }
179     }
180
181     for (l=0;l<NQ;l++)
182     {
183         Du1[0][0]=0;Du1[1][0]=0;Du1[0][1]=0;Du1[1][1]=0;
184         Du2[0][0]=0;Du2[1][0]=0;Du2[0][1]=0;Du2[1][1]=0;
185
186         for (j=0;j<ND;j++)

```

```

187     {
188     Du1[0][0]=Du1[0][0]+
189     (u[2*I0[j]-1-1]-u0[2*I0[j]-1-1])*dxxz[(int) nz[j]-1][1];
190     Du1[0][1]=Du1[0][1]+
191     (u[2*I0[j]-1-1]-u0[2*I0[j]-1-1])*dxyz[(int) nz[j]-1][1];
192     Du1[1][1]=Du1[1][1]+
193     (u[2*I0[j]-1-1]-u0[2*I0[j]-1-1])*dyyz[(int) nz[j]-1][1];
194
195     Du2[0][0]=Du2[0][0]+
196     (u[2*I0[j]-1]-u0[2*I0[j]-1])*dxxz[(int) nz[j]-1][1];
197     Du2[0][1]=Du2[0][1]+
198     (u[2*I0[j]-1]-u0[2*I0[j]-1])*dxyz[(int) nz[j]-1][1];
199     Du2[1][1]=Du2[1][1]+
200     (u[2*I0[j]-1]-u0[2*I0[j]-1])*dyyz[(int) nz[j]-1][1];
201     }
202     Du1[1][0]=Du1[0][1];Du2[1][0]=Du2[0][1];
203     Matmult(Du1j,Du1,IJM);Matmult(Du2j,Du2,IJM);
204
205     Zx[0]=(Du1j[0][0]-Du2j[0][1])/2.0+
206     1.0/(2.0*(mu))*(DSOX[1]-DSOX[3*NQ+1])/2.0-
207     1.0/(2.0*(mu))*(DAOX[1]-DAOX[3*NQ+1])/2.0;
208     Zx[1]=(Du2j[0][0]+Du1j[0][1])/2.0+
209     1.0/(2.0*(mu))*DSOX[NQ+1]-1.0/(2.0*(mu))*DAOX[NQ+1];
210     Zx[2]=(Du2j[0][0]+Du1j[0][1])/2+
211     1.0/(2.0*(mu))*DSOX[2*NQ+1]-1.0/(2.0*(mu))*DAOX[2*NQ+1];
212     Zx[3]=(Du2j[0][1]-Du1j[0][0])/2+
213     1.0/(2.0*(mu))*(DSOX[3*NQ+1]-DSOX[1])/2.0-
214     1.0/(2.0*(mu))*(DAOX[3*NQ+1]-DAOX[1])/2.0;
215
216     Zy[0]=(Du1j[1][0]-Du2j[1][1])/2+
217     1.0/(2.0*(mu))*(DSOY[1]-DSOY[3*NQ+1])/2.0-
218     1.0/(2.0*(mu))*(DAOY[1]-DAOY[3*NQ+1])/2.0;
219     Zy[1]=(Du2j[1][0]+Du1j[1][1])/2+
220     1.0/(2.0*(mu))*DSOY[NQ+1]-1.0/(2.0*(mu))*DAOY[NQ+1];
221     Zy[2]=(Du2j[1][0]+Du1j[1][1])/2+
222     1.0/(2.0*(mu))*DSOY[2*NQ+1]-1.0/(2.0*(mu))*DAOY[2*NQ+1];
223     Zy[3]=(Du2j[1][1]-Du1j[1][0])/2+
224     1.0/(2.0*(mu))*(DSOY[3*NQ+1]-DSOY[1])/2.0-
225     1.0/(2.0*(mu))*(DAOY[3*NQ+1]-DAOY[1])/2.0;
226
227
228     /* Spuranteil */
229     DSX[1]=(*lambda+mu)*(Du1j[0][0]+Du2j[0][1]+
230     1.0/(2.0*(lambda)+2.0*(mu))*(DSOX[1]+DSOX[1+3*NQ]));
231     DSY[1]=(*lambda+mu)*(Du1j[1][0]+Du2j[1][1]+
232     1.0/(2.0*(lambda)+2.0*(mu))*(DSOY[1]+DSOY[1+3*NQ]));
233     DSX[3*NQ+1]=DSX[1];
234     DSY[3*NQ+1]=DSY[1];
235
236     /* letzter Summand im Spannungstensor */
237     DSX[1]=DSX[1]+DAOX[1];
238     DSX[1+NQ]=DSX[1+NQ]+DAOX[1+NQ];
239     DSX[1+2*NQ]=DSX[1+2*NQ]+DAOX[1+2*NQ];
240     DSX[1+3*NQ]=DSX[1+3*NQ]+DAOX[1+3*NQ];
241

```

```

242     DSY [1]=DSY [1]+DAOY [1];
243     DSY [1+NQ]=DSY [1+NQ]+DAOY [1+NQ];
244     DSY [1+2*NQ]=DSY [1+2*NQ]+DAOY [1+2*NQ];
245     DSY [1+3*NQ]=DSY [1+3*NQ]+DAOY [1+3*NQ];
246
247     AblSigma(DSZ11,DSZ12,DSZ21,DSZ22,l,u,u0,S0,A0,jdof,nz,I0,dx,dy,IJM,
248             NQ,ND,NDO,(*C1),(*C2),(*C3),(*lambda),(*mu),(*k1),(*sy));
249
250     DSX [1]=DSX [1]+
251             (DSZ11 [0]*Zx [0]+DSZ11 [1]*Zx [1]+DSZ11 [2]*Zx [2]+DSZ11 [3]*Zx [3]);
252     DSX [1+NQ]=DSX [1+NQ]+
253             (DSZ12 [0]*Zx [0]+DSZ12 [1]*Zx [1]+DSZ12 [2]*Zx [2]+DSZ12 [3]*Zx [3]);
254     DSX [1+2*NQ]=DSX [1+2*NQ]+
255             (DSZ21 [0]*Zx [0]+DSZ21 [1]*Zx [1]+DSZ21 [2]*Zx [2]+DSZ21 [3]*Zx [3]);
256     DSX [1+3*NQ]=DSX [1+3*NQ]+
257             (DSZ22 [0]*Zx [0]+DSZ22 [1]*Zx [1]+DSZ22 [2]*Zx [2]+DSZ22 [3]*Zx [3]);
258
259     DSY [1]=DSY [1]+
260             (DSZ11 [0]*Zy [0]+DSZ11 [1]*Zy [1]+DSZ11 [2]*Zy [2]+DSZ11 [3]*Zy [3]);
261     DSY [1+NQ]=DSY [1+NQ]+
262             (DSZ12 [0]*Zy [0]+DSZ12 [1]*Zy [1]+DSZ12 [2]*Zy [2]+DSZ12 [3]*Zy [3]);
263     DSY [1+2*NQ]=DSY [1+2*NQ]+
264             (DSZ21 [0]*Zy [0]+DSZ21 [1]*Zy [1]+DSZ21 [2]*Zy [2]+DSZ21 [3]*Zy [3]);
265     DSY [1+3*NQ]=DSY [1+3*NQ]+
266             (DSZ22 [0]*Zy [0]+DSZ22 [1]*Zy [1]+DSZ22 [2]*Zy [2]+DSZ22 [3]*Zy [3]);
267 }
268
269     free(I0);I0=NULL;
270     j=ND0;
271     while (j--)
272     {
273         free(dxz[j]);free(dxy[j]);free(dyy[j]);
274     }
275     free(dxz);dxz=NULL;free(dxy);dxy=NULL;free(dyy);dyy=NULL;
276
277
278 }
279
280 /* Gateway function */
281
282 void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[])
283 {
284     double *DSX,*DSY,*u,*u0,*S0,*A0,*DSOX,*DSOY,*DAOX,*DAOY,*J,*jdof,*nz,*I,
285             *dx,*dy,*dxx,*dxy,*dyy,*NQP,*lambda,*mu,*C1,*C2,*C3,*sy,*k1;
286     int ND,ND0,n,NQ;
287
288
289     /*Check for proper number of arguments */
290     if (nrhs!=25)
291         mexErrMsgTxt("25 input required!");
292     if (nlhs!=2)
293         mexErrMsgTxt("2 output required!");
294
295     n =mxGetM(prhs [0])/2;
296     ND0=mxGetN(prhs [9]);

```



```

297 ND=mxGetN(prhs[10]);
298
299 u      =mxGetPr(prhs[0]); u0      =mxGetPr(prhs[1]);
300 S0     =mxGetPr(prhs[2]); A0     =mxGetPr(prhs[3]);
301 DS0X  =mxGetPr(prhs[4]); DS0Y  =mxGetPr(prhs[5]);
302 DA0X  =mxGetPr(prhs[6]); DA0Y  =mxGetPr(prhs[7]);
303 J     =mxGetPr(prhs[8]); jdof  =mxGetPr(prhs[9]);
304 nz    =mxGetPr(prhs[10]); I    =mxGetPr(prhs[11]);
305 dx    =mxGetPr(prhs[12]); dy   =mxGetPr(prhs[13]);
306 dxx   =mxGetPr(prhs[14]); dxy  =mxGetPr(prhs[15]);
307 dyy   =mxGetPr(prhs[16]); NQP  =mxGetPr(prhs[17]);
308 lambda =mxGetPr(prhs[18]); mu  =mxGetPr(prhs[19]);
309 C1    =mxGetPr(prhs[20]); C2    =mxGetPr(prhs[21]);
310 C3    =mxGetPr(prhs[22]); sy    =mxGetPr(prhs[23]);
311 k1    =mxGetPr(prhs[24]);
312
313
314 NQ=(int) *NQP;
315 /*Set the output pointer to the output matrix. */
316 plhs[0]=mxCreateDoubleMatrix(1,4*NQ,mxREAL);
317 plhs[1]=mxCreateDoubleMatrix(1,4*NQ,mxREAL);
318
319 /*Create a C pointer to a copy of the output matrix */
320 DSX=mxGetPr(plhs[0]);
321 DSY=mxGetPr(plhs[1]);
322
323 /* DSigma3 */
324 DSigma3(DSX,DSY,u,u0,S0,A0,DS0X,DS0Y,DA0X,DA0Y,J,jdof,nz,I,dx,dy,
325         dxx,dxy,dyy,NQ,ND,ND0,lambda,mu,C1,C2,C3,sy,k1);
326 }

```

Die Unteroutine `DALphaC3.c` berechnet die Ableitung der internen Spannung, wobei (3.41) aus Kapitel 3 ausgenutzt wird. Es gilt

$$\frac{\partial}{\partial x} \alpha_{k-1+\theta} = \frac{\partial}{\partial x} \alpha_{h,k-1} + k_1 (\theta \Delta t_k \frac{\partial}{\partial x} \mathbf{A}_{k-1+\theta} - \beta \frac{\partial}{\partial x} \text{dev } \boldsymbol{\sigma}_{k-1+\theta}), \quad (1.6)$$

$$\frac{\partial}{\partial y} \alpha_{k-1+\theta} = \frac{\partial}{\partial y} \alpha_{h,k-1} + k_1 (\theta \Delta t_k \frac{\partial}{\partial y} \mathbf{A}_{k-1+\theta} - \beta \frac{\partial}{\partial y} \text{dev } \boldsymbol{\sigma}_{k-1+\theta}). \quad (1.7)$$

```

1 #include <stdio.h>
2 #include <math.h>
3 #include <stdlib.h>
4 #include "mex.h"
5
6 double **CreateDoubleMatrix(int m, int n)
7 {
8     double **mat=malloc(m*sizeof(double*));
9     int row=m;
10    while (row-->0) mat[row]=malloc(n*sizeof(double));
11    return mat;
12 }
13
14 double *CreateDoubleVector(int len)
15 {

```

```
16     return malloc(len*sizeof(double));
17 }
18
19 int *CreateIntVector(int len)
20 {
21     return malloc(len*sizeof(int));
22 }
23
24 double tr(double x[4])
25 {
26     double t;
27     t=x[0]+x[3];
28     return t;
29 }
30
31 double skal(double x[4], double y[4])
32 {
33     double s;
34     s=x[0]*y[0]+x[1]*y[1]+x[2]*y[2]+x[3]*y[3];
35     return s;
36 }
37
38 void Matmult(double Du[2][2],double x[2][2], double y[2][2])
39 {
40     double tmp[2][2];
41
42     tmp[0][0]=x[0][0]*y[0][0]+x[0][1]*y[0][1];
43     tmp[1][0]=x[1][0]*y[0][0]+x[1][1]*y[0][1];
44     tmp[0][1]=x[0][0]*y[1][0]+x[0][1]*y[1][1];
45     tmp[1][1]=x[1][0]*y[1][0]+x[1][1]*y[1][1];
46
47     Du[0][0]=y[0][0]*tmp[0][0]+y[0][1]*tmp[1][0];
48     Du[0][1]=y[0][0]*tmp[0][1]+y[0][1]*tmp[1][1];
49     Du[1][0]=y[1][0]*tmp[0][0]+y[1][1]*tmp[1][0];
50     Du[1][1]=y[1][0]*tmp[0][1]+y[1][1]*tmp[1][1];
51
52 }
53
54
55 void symgrad(double y[4], double x[4])
56 {
57     y[0]=x[0];y[1]=(x[1]+x[2])/2.0;y[2]=y[1];y[3]=x[3];
58 }
59
60 void dev(double y[4], double x[4])
61 {
62     y[0]=x[0]-(x[0]+x[3])/2.0;y[1]=x[1];y[2]=x[2];y[3]=x[3]-(x[0]+x[3])/2.0;
63 }
64
65 double norm(double x[4])
66 {
67     double n;
68     n=sqrt(pow(x[0],2)+pow(x[1],2)+pow(x[2],2)+pow(x[3],2));
69     return n;
70 }
```

```

71
72 double dsign(double x)
73 {
74     double s;
75     if (x>=0) s=1; else s=-1;
76     return s;
77 }
78
79 void DAlpha3(double DAX[],double DAY[],double u[],double u0[],
80             double S0[],double A0[],double DSX[], double DSY[],
81             double DSOX[],double DSOY[],double DAOX[],double DAOY[],
82             double JM[],double jdof[],double nz[],double I[],
83             double dxx[],double dxy[], double dyy[],
84             int NQ,int ND,int NDO,
85             double *mup,double *k1p)
86 {
87     double DT,IJM[2][2],Du1[2][2],Du2[2][2],Du1j[2][2],Du2j[2][2],
88         **dxxz,**dxyz,**dyyz,Zx[4],Zy[4];
89     double DSZ11[4],DSZ12[4],DSZ21[4],DSZ22[4];
90     double mu=*mup,k1=*k1p;
91     int *I0,i,j,l;
92
93     I0=CreateIntVector(ND);
94     dxxz=CreateDoubleMatrix(NDO,NQ);
95     dxyz=CreateDoubleMatrix(NDO,NQ);
96     dyyz=CreateDoubleMatrix(NDO,NQ);
97
98
99     for (i=0;i<4*NQ;i++)
100     {
101         DAX[i]=0.0;DAY[i]=0.0;
102     }
103
104     DT=JM[0]*JM[3]-JM[1]*JM[2];
105     IJM[0][0]=JM[3]/DT; IJM[0][1]=-JM[2]/DT;
106     IJM[1][0]=-JM[1]/DT;IJM[1][1]=JM[0]/DT;
107
108     for (j=0;j<ND;j++) I0[j]=(int) I[j];
109     for (j=0;j<NDO;j++)
110     {
111         for (l=0;l<NQ;l++)
112         {
113             dxxz[j][l]=dxx[j+NDO*l]*dsign(jdof[j]);
114             dxyz[j][l]=dxy[j+NDO*l]*dsign(jdof[j]);
115             dyyz[j][l]=dyy[j+NDO*l]*dsign(jdof[j]);
116         }
117     }
118
119     for (l=0;l<NQ;l++)
120     {
121         Du1[0][0]=0;Du1[1][0]=0;Du1[0][1]=0;Du1[1][1]=0;
122         Du2[0][0]=0;Du2[1][0]=0;Du2[0][1]=0;Du2[1][1]=0;
123
124         for (j=0;j<ND;j++)
125         {

```

```

126         Du1[0][0]=Du1[0][0]+
127         (u[2*I0[j]-1-1]-u0[2*I0[j]-1-1])*dxxz[(int) nz[j]-1][1];
128         Du1[0][1]=Du1[0][1]+
129         (u[2*I0[j]-1-1]-u0[2*I0[j]-1-1])*dxyz[(int) nz[j]-1][1];
130         Du1[1][1]=Du1[1][1]+
131         (u[2*I0[j]-1-1]-u0[2*I0[j]-1-1])*dyyz[(int) nz[j]-1][1];
132
133         Du2[0][0]=Du2[0][0]+
134         (u[2*I0[j]-1]-u0[2*I0[j]-1])*dxxz[(int) nz[j]-1][1];
135         Du2[0][1]=Du2[0][1]+
136         (u[2*I0[j]-1]-u0[2*I0[j]-1])*dxyz[(int) nz[j]-1][1];
137         Du2[1][1]=Du2[1][1]+
138         (u[2*I0[j]-1]-u0[2*I0[j]-1])*dyyz[(int) nz[j]-1][1];
139     }
140     Du1[1][0]=Du1[0][1]; Du2[1][0]=Du2[0][1];
141     Matmult(Du1j, Du1, IJM); Matmult(Du2j, Du2, IJM);
142
143     Zx[0]=(Du1j[0][0]-Du2j[0][1])/2.0+
144         1.0/(2.0*mu)*(DSOX[1]-DSOX[3*NQ+1])/2.0-
145         1.0/(2.0*mu)*(DAOX[1]-DAOX[3*NQ+1])/2.0;
146     Zx[1]=(Du2j[0][0]+Du1j[0][1])/2.0+
147         1.0/(2.0*mu)*DSOX[NQ+1]-1.0/(2.0*mu)*DAOX[NQ+1];
148     Zx[2]=(Du2j[0][0]+Du1j[0][1])/2+
149         1.0/(2.0*mu)*DSOX[2*NQ+1]-1.0/(2.0*mu)*DAOX[2*NQ+1];
150     Zx[3]=(Du2j[0][1]-Du1j[0][0])/2+
151         1.0/(2.0*mu)*(DSOX[3*NQ+1]-DSOX[1])/2.0-
152         1.0/(2.0*mu)*(DAOX[3*NQ+1]-DAOX[1])/2.0;
153
154     Zy[0]=(Du1j[1][0]-Du2j[1][1])/2+
155         1.0/(2.0*mu)*(DSOY[1]-DSOY[3*NQ+1])/2.0-
156         1.0/(2.0*mu)*(DAOY[1]-DAOY[3*NQ+1])/2.0;
157     Zy[1]=(Du2j[1][0]+Du1j[1][1])/2+
158         1.0/(2.0*mu)*DSOY[NQ+1]-1.0/(2.0*mu)*DAOY[NQ+1];
159     Zy[2]=(Du2j[1][0]+Du1j[1][1])/2+
160         1.0/(2.0*mu)*DSOY[2*NQ+1]-1.0/(2.0*mu)*DAOY[2*NQ+1];
161     Zy[3]=(Du2j[1][1]-Du1j[1][0])/2+
162         1.0/(2.0*mu)*(DSOY[3*NQ+1]-DSOY[1])/2.0-
163         1.0/(2.0*mu)*(DAOY[3*NQ+1]-DAOY[1])/2.0;
164
165     DAX[1]=DAOX[1]+k1*Zx[0]-k1/(2.0*mu)*(DSX[1]-DSX[3*NQ+1])/2.0;
166     DAX[NQ+1]=DAOX[NQ+1]+k1*Zx[1]-k1/(2.0*mu)*DSX[NQ+1];
167     DAX[2*NQ+1]=DAOX[2*NQ+1]+k1*Zx[2]-k1/(2.0*mu)*DSX[2*NQ+1];
168     DAX[3*NQ+1]=DAOX[3*NQ+1]+k1*Zx[3]-k1/(2.0*mu)*(DSX[3*NQ+1]-DSX[1])/2.0;
169
170     DAY[1]=DAOY[1]+k1*Zy[0]-k1/(2.0*mu)*(DSY[1]-DSY[3*NQ+1])/2.0;
171     DAY[NQ+1]=DAOY[NQ+1]+k1*Zy[1]-k1/(2.0*mu)*DSY[NQ+1];
172     DAY[2*NQ+1]=DAOY[2*NQ+1]+k1*Zy[2]-k1/(2.0*mu)*DSY[2*NQ+1];
173     DAY[3*NQ+1]=DAOY[3*NQ+1]+k1*Zy[3]-k1/(2.0*mu)*(DSY[3*NQ+1]-DSY[1])/2.0;
174
175 }
176
177 free(IO); IO=NULL;
178 j=NDO;
179 while(j--)
180 {

```

```

181     free(dxxz[j]);free(dxyz[j]);free(dyyz[j]);
182 }
183     free(dxxz);dxxz=NULL;free(dxyz);dxyz=NULL;free(dyyz);dyyz=NULL;
184
185
186 }
187
188 /* Gateway function */
189
190 void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[])
191 {
192     double *DAX,*DAY,*u,*u0,*S0,*A0,*DSX,*DSY,*DSOX,*DSOY,*DAOX,*DAOY,
193           *J,*jdof,*nz,*I,*dxx,*dxy,*dyy,*NQP,*mu,*k1;
194     int ND,NDO,n,NQ;
195
196
197     /*Check for proper number of arguments */
198     if (nrhs!=20)
199         mexErrMsgTxt("20 input required!");
200     if (nlhs!=2)
201         mexErrMsgTxt("2 output required!");
202
203     n =mxGetM(prhs[0])/2;
204     NDO=mxGetN(prhs[11]);
205     ND=mxGetN(prhs[12]);
206
207     u      =mxGetPr(prhs[0]); u0      =mxGetPr(prhs[1]);
208     S0     =mxGetPr(prhs[2]); A0     =mxGetPr(prhs[3]);
209     DSX    =mxGetPr(prhs[4]); DSY    =mxGetPr(prhs[5]);
210     DSOX   =mxGetPr(prhs[6]); DSOY   =mxGetPr(prhs[7]);
211     DAOX   =mxGetPr(prhs[8]); DAOY   =mxGetPr(prhs[9]);
212     J      =mxGetPr(prhs[10]); jdof  =mxGetPr(prhs[11]);
213     nz     =mxGetPr(prhs[12]); I     =mxGetPr(prhs[13]);
214     dxx    =mxGetPr(prhs[14]); dxy    =mxGetPr(prhs[15]);
215     dyy    =mxGetPr(prhs[16]); NQP   =mxGetPr(prhs[17]);
216     mu     =mxGetPr(prhs[18]); k1    =mxGetPr(prhs[19]);
217
218
219     NQ=(int) *NQP;
220     /*Set the output pointer to the output matrix. */
221     plhs[0]=mxCreateDoubleMatrix(1,4*NQ,mxREAL);
222     plhs[1]=mxCreateDoubleMatrix(1,4*NQ,mxREAL);
223
224     /*Create a C pointer to a copy of the output matrix */
225     DAX=mxGetPr(plhs[0]);
226     DAY=mxGetPr(plhs[1]);
227
228     /* DAlpha3 */
229     DAlpha3(DAX,DAY,u,u0,S0,A0,DSX,DSY,DSOX,DSOY,DAOX,DAOY,J,jdof,nz,I,
230           dxx,dxy,dyy,NQ,ND,NDO,mu,k1);
231 }

```

Fehlerschätzer η_R

```

1 function eta_R=APosterioriR(u,u0,S0,A0,S0_K,A0_K,DSX,DSY,p,t,t0)

```

```

2
3 global Kantennr Koordinaten Elemente Dirichlet Neumann Gewichte BaryK NQ ...
4     P_ref Gewichte1 alpha1 beta1 NQ1 dx_K dy_K dof C1 C2 C3 lambda mu sy
5
6 eta_R=zeros(size(Elemente,1),1);eta_E=zeros(max(max(Kantennr)),1);
7 Kantennr2Elemente=zeros(max(max(Kantennr)),2);
8 Kantennr2Kante=zeros(max(max(Kantennr)),2);
9 Kante=2*ones(max(max(Kantennr)),1);
10 Kante2=2*ones(max(max(Kantennr)),1);
11 if size(Neumann,2)>0
12     Kante(diag(Kantennr(Neumann(:,1),Neumann(:,2)')))=ones(size(Neumann,1),1);
13     Kante3(diag(Kantennr(Neumann(:,1),Neumann(:,2)')))=ones(size(Neumann,1),1);
14 end
15 Kante(diag(Kantennr(Dirichlet(:,1),Dirichlet(:,2)')))=...
16 zeros(size(Dirichlet,1),1);
17 Kante2(diag(Kantennr(Dirichlet(:,1),Dirichlet(:,2)')))=...
18 ones(size(Dirichlet,1),1);
19
20 for i=1:size(Elemente,1)
21     for k=1:3
22         Kantennr2Elemente(Kantennr(Elemente(i,k),Elemente(i,rem(k,3)+1)),...
23 nnz(Kantennr2Elemente(Kantennr(Elemente(i,k),Elemente(i,rem(k,3)+1)),:))+1)=i;
24         Kantennr2Kante(Kantennr(Elemente(i,k),Elemente(i,rem(k,3)+1)),...
25 nnz(Kantennr2Kante(Kantennr(Elemente(i,k),Elemente(i,rem(k,3)+1)),:))+1)=k;
26     end
27 end
28 %Kantenspruenge
29 for i=1:full(max(max(Kantennr)))
30     s1=Kantennr2Kante(i,1);
31     j1=Kantennr2Elemente(i,1);
32     J=Funktionalmatrix(Koordinaten,Elemente,j1);
33     jdof=dof(j1,:);nz=find(jdof);
34     Sigma1=SpannungC4(u,u0,S0_K{s1}(j1,:),A0_K{s1}(j1,:),jdof,nz,...
35     dx_K{s1},dy_K{s1},reshape(J,1,4),Gewichte1,lambda,mu,C1,C2,C3,sy);
36     KV=(Koordinaten(Elemente(j1,rem(s1,3)+1),:)-Koordinaten(Elemente(j1,s1),:));
37     NV=[0,1;-1,0]*KV'/norm(KV);
38     if nnz(Kantennr2Elemente(i,:))==2
39         %Innenkante
40         s2=Kantennr2Kante(i,2);
41         j2=Kantennr2Elemente(i,2);
42         J=Funktionalmatrix(Koordinaten,Elemente,j2);
43         jdof=dof(j2,:);nz=find(jdof);
44         Sigma2=SpannungC4(u,u0,S0_K{s2}(j2,:),A0_K{s2}(j2,:),jdof,nz,...
45         dx_K{s2},dy_K{s2},reshape(J,1,4),Gewichte1,lambda,mu,C1,C2,C3,sy);
46         eta_E(i)=(norm(KV)/max(p(j1),p(j2)))*(norm(KV)/2*Gewichte1*...
47         sum([(-Sigma1(:,1:2)+Sigma2(NQ1:-1:1,1:2))*NV, ...
48         (-Sigma1(:,3:4)+Sigma2(NQ1:-1:1,3:4))*NV].^2,2));
49     else
50         %Neumann
51         if Kante(i)==1
52             Quadpunkte=[alpha1',beta1']*...
53             [Koordinaten(Elemente(j1,s1),:);Koordinaten(Elemente(j1,rem(s1,3)+1),:)];
54             eta_E(i)=norm(KV)/p(j1)*(norm(KV)/2*Gewichte1*...
55             sum([Sigma1(:,1:2)*NV,Sigma1(:,3:4)*NV]-g(Quadpunkte,NV',t)).^2,2));
56         end

```

```

57   end
58   end
59
60   for j=1:size(Elemente,1)
61     DSXj=reshape(DSX(j,:),NQ,4);DSYj=reshape(DSY(j,:),NQ,4);
62     Quadraturpunkte=BaryK'*Koordinaten(Elemente(j,:),:);
63     KV=Koordinaten(Elemente(j,[2,3,1]),:)-Koordinaten(Elemente(j,:),:);
64     h=max(sum(KV.^2,2).^(1/2));
65     ft=f(Quadraturpunkte,t);
66     T=det([1 1 1;Koordinaten(Elemente(j,:),:)]')/2;
67     I=diag(Kantennr(Elemente(j,:),Elemente(j,[2,3,1])))';
68     eta_R(j)=sqrt(h^2/p(j)^2*(T*Gewichte*...
69       ((DSXj(:,1)+DSYj(:,2)+ft(:,1)).^2+(DSXj(:,3)+DSYj(:,4)+ft(:,2)).^2))+...
70       sum(eta_E(I)./Kante2(I)));
71   end

```

A.6 Fehlerschätzer η_{EQ} : Methode der äquilibrierten Residuen

Der Fehlerschätzer nach der Methode der äquilibrierten Residuen ist nach dem Algorithmus aus Abschnitt 4.3 implementiert. Das Programm `Aposteriori_EQ1.m` teilt sich in zwei Teile auf. Der erste Teil, Zeile 7 - Zeile 221, bestimmt die Momente der äquilibrierten Randkräfte zu linearen Basisfunktionen, führt also die Äquilibrierung 1. Ordnung durch. Der zweite Teil, Zeile 222 - Zeile 309, löst die elementweisen lokalen Probleme.

Die Äquilibrierung 1. Ordnung läßt sich unabhängig für die Komponenten der linearen Basisfunktionen durchführen. Im folgenden bezeichnen Indizes $\{1\}$ und $\{2\}$ Größen, die sich auf die x- bzw. y-Komponente der linearen Basisfunktionen beziehen. In der Datei `Elemente.dat` sind die Knoten zu jedem Element so angeordnet, daß die Elementknoten entgegen dem Uhrzeigersinn durchlaufen werden. Dadurch ist jeder Elementkante eindeutig ein 1. und ein 2. Knoten zugewiesen. Zeile 8-13 initialisieren die genäherten Momente 1. Ordnung, $\tilde{\mu}_{T,l}^\gamma$, abhängig von der Elementnummer j und der Kantenummer k bezeichnet durch die Variablen

```

mu0_1{1}(j,k) (Basisfunktion 1, Komponente 1)
mu0_1{2}(j,k) (Basisfunktion 1, Komponente 2)
mu0_2{1}(j,k) (Basisfunktion 2, Komponente 1)
mu0_2{2}(j,k) (Basisfunktion 2, Komponente 2)

```

die Momente 1. Ordnung, $\mu_{T,l}^\gamma$, bezeichnet durch die Variablen

```

mu0_1{1}(j,k) (Basisfunktion 1, Komponente 1)
mu0_1{2}(j,k) (Basisfunktion 1, Komponente 2)
mu0_2{1}(j,k) (Basisfunktion 2, Komponente 1)
mu0_2{2}(j,k) (Basisfunktion 2, Komponente 2)

```

sowie die Variablen σ_{T,n_i} , abhängig von der Elementnummer j und der Knotennummer n , bezeichnet durch die Variablen

```

sigma{1}(j,n) (Komponente 1)
sigma{2}(j,n) (Komponente 2).

```

Zeilen 39-64 berechnen die genäherten Momente 1. Ordnung, wie in (4.118), Abschnitt 4.3 definiert. In den Zeilen 65-192 werden die Variablen $\sigma\{1\}(j,n)$ und $\sigma\{2\}(j,n)$ ermittelt, nach dem Verfahren aus Abschnitt 4.3.3. Dazu ist ein Durchlauf über alle Knoten n nötig (Zeile 66), da zu jedem Patch ein Problem gelöst werden soll, sowie ein Durchlauf über die beiden Komponenten (Zeile 73). In den Zeilen 73-114 werden für den aktuellen Patch und die aktuelle Komponente die rechten Seiten der lokalen Probleme bestimmt, definiert in (4.128). In den Zeilen 116-191 wird die Matrix des Gleichungssystems (4.126) aufgestellt und das Gleichungssystem gelöst, wobei die in Abschnitt 4.3.3 beschriebene Lösungsstrategie aus [2] implementiert ist. In den Zeilen 194-220 werden nach (4.125) die Momente der äquilibrierten Randkräfte berechnet.

Der restliche Programmteil, Zeilen 222-309, löst die elementweisen lokalen Probleme 4.39 zur Bestimmung der Fehlerindikatoren. In den Zeilen 236-296 wird die rechte Seite $l_{k-1+\theta,T}(\mathbf{v}_h)$ aus (4.79) aufgestellt und in Zeile 245 eine Unterroutine zur Berechnung der zugehörigen Steifigkeitsmatrix aufgerufen. Zeilen 260-296 berechnen die äquilibrierte Randkraft aus den Momenten. Dabei wird auch die Äquilibrierung höherer Ordnung berücksichtigt. Die äquilibrierte Randkraft wird nach (4.144) als Linearkombination der Basisfunktionen zu den Freiheitsgraden aus der Indexmenge $I_1(\gamma) \cup I_2(\gamma)$ berechnet. Das Gleichungssystem (4.145) zur Bestimmung der Koeffizienten läßt sich in zwei Gleichungssysteme aufteilen und wird in den Zeilen 281,282 gelöst. Zeilen 302-306 lösen das lokale Probleme mit Hilfe einer QR-Zerlegung für Elemente die keine Kante auf dem Dirichletrand haben, Zeile 300 löst das lokale Problem für Elemente mit Kante auf dem Dirichletrand.

```

1  function eta_N=Aposteriori_EQ1(u,u0,S0,A0,S0_K,A0_K,t,t0,pmax,Pmax);
2
3  global Koordinaten Elemente Dirichlet Neumann p dof Kantennr Aussenkanten ...
4      NJ Gewichte BaryK NQ Gewichte1 alpha1 beta1 NQ1 basis_p dx dy ...
5      dx_K dy_K basis_P DX DY D0F P_ref R C1 C2 C3 lambda mu k1 sy
6
7  maxK=max(max(Kantennr));
8  mu_1{1}=sparse(NJ,maxK);mu_1{2}=sparse(NJ,maxK);
9  mu_2{1}=sparse(NJ,maxK);mu_2{2}=sparse(NJ,maxK);
10 mu0_1{1}=sparse(NJ,maxK);mu0_1{2}=sparse(NJ,maxK);
11 mu0_2{1}=sparse(NJ,maxK);mu0_2{2}=sparse(NJ,maxK);
12 sigma{1}=sparse(NJ,size(Koordinaten,1));
13 sigma{2}=sparse(NJ,size(Koordinaten,1));
14 Gewichte_Kante=Gewichte1/2;
15 alpha=BaryK(1,:);beta=BaryK(2,:);gamma=BaryK(3,:);
16 P_dim=3+3*(Pmax-1)+(Pmax-1)*(Pmax-2)/2;
17 eta_N=zeros(size(Elemente,1),1);
18 perm=[1,2,3;2,3,1;3,1,2];
19 perm2=[1 2;2 3;3 1];
20 Knoten=2*ones(size(Koordinaten,1),1);
21 Knoten(unique(Neumann))=1;
22 Knoten(unique(Dirichlet))=0;
23 Kantenbasis_p=fwertkante(ones(1,NQ1)-2*alpha1,pmax);
24 Kante=2*ones(max(max(Kantennr)),1);
25 if size(Neumann,2)>0
26     Kante(diag(Kantennr(Neumann(:,1),Neumann(:,2)')))=ones(size(Neumann,1),1);
27 end

```



```

28 Kante(diag(KantenNr(Dirichlet(:,1),Dirichlet(:,2)')))=...
29 zeros(size(Dirichlet,1),1);
30 x2Elemente=zeros(size(Koordinaten,1),1);
31 KantenNr2Elemente=zeros(max(max(KantenNr)),2);
32 for i=1:size(Elemente,1)
33     for k=1:3
34         x2Elemente(Elemente(i,k),nanz(x2Elemente(Elemente(i,k),:))+1)=i;
35         KantenNr2Elemente(KantenNr(Elemente(i,k),Elemente(i,rem(k,3)+1)),...
36             nanz(KantenNr2Elemente(KantenNr(Elemente(i,k),Elemente(i,rem(k,3)+1)),:))+1)=i;
37     end
38 end
39 %Bestimmung der genaherten Fluss-Momente und Stemas
40 for j=1:size(Elemente,1)
41     T(j)=det([1,1,1;Koordinaten(Elemente(j,:),:)])/2;
42     JM=Funktionalmatrix(Koordinaten,Elemente,j);
43     jdof=dof(j,:);nz=find(jdof);
44     for i=1:3
45         s1=Elemente(j,i);s2=Elemente(j,rem(i,3)+1);
46         lk=norm(Koordinaten(s1,:)-Koordinaten(s2,:));
47         nk=[0 1;-1 0]*(Koordinaten(s2,:)-Koordinaten(s1,:))'/lk;
48         if (Kante(KantenNr(s1,s2))==1)
49             g_wert=g([alpha1;beta1]'*[Koordinaten(s1,:);Koordinaten(s2,)] ,nk',t);
50             mu0_1{1}(j,KantenNr(s1,s2))=lk*Gewichte_Kante*(g_wert(:,1).*alpha1');
51             mu0_1{2}(j,KantenNr(s1,s2))=lk*Gewichte_Kante*(g_wert(:,2).*alpha1');
52             mu0_2{1}(j,KantenNr(s1,s2))=lk*Gewichte_Kante*(g_wert(:,1).*beta1');
53             mu0_2{2}(j,KantenNr(s1,s2))=lk*Gewichte_Kante*(g_wert(:,2).*beta1');
54         else
55             Sigma=SpannungC4(u,u0,S0_K{i}(j,:),A0_K{i}(j,:),jdof,nz,dx_K{i},dy_K{i},...
56                 reshape(JM,1,4),Gewichte1,lambda,mu,C1,C2,C3,sy);
57
58             mu0_1{1}(j,KantenNr(s1,s2))=lk*Gewichte_Kante*((Sigma(:,[1,2])*nk).*alpha1');
59             mu0_1{2}(j,KantenNr(s1,s2))=lk*Gewichte_Kante*((Sigma(:,[3,4])*nk).*alpha1');
60             mu0_2{1}(j,KantenNr(s1,s2))=lk*Gewichte_Kante*((Sigma(:,[1,2])*nk).*beta1');
61             mu0_2{2}(j,KantenNr(s1,s2))=lk*Gewichte_Kante*((Sigma(:,[3,4])*nk).*beta1');
62         end
63     end
64 end
65 %Bestimmung der Funktionen sigma
66 for i=1:size(Koordinaten,1)
67     X=x2Elemente(i,1:nanz(x2Elemente(i,:)));
68     Patch=Elemente(X,:);
69     for k=1:size(Patch,1)
70         Patch(k,:)=Patch(k,perm(find(Patch(k,:))==i,:));
71     end
72 %Schleife ueber die Komponenten (fuer jedes Patch 2 LGS zu loesen)
73 for kp=1:2
74     rk=zeros(size(Patch,1),1);
75     for k=1:size(Patch,1)
76         JM=Funktionalmatrix(Koordinaten,Elemente,X(k));
77         IJM=inv(JM)';
78         fwert=f(BaryK'*Koordinaten(Elemente(X(k),:),:),t);
79         rk(k)=-T(X(k))*Gewichte*(fwert(:,kp).*alpha1');
80         z=find(Patch(k,1)==Elemente(X(k),:));
81         idof=dof(X(k),:);nz=find(idof);
82         dxi=dx(nz,:).*(sign(idof(nz))'*ones(1,NQ));

```

```

83     dyi=dy(nz,:).*(sign(idof(nz))'*ones(1,NQ));
84     Sigma=SpannungC4(u,u0,S0(X(k,:),),A0(X(k,:),),idof,nz,dx,dy,...
85         reshape(JM,1,4),Gewichte,lambda,mu,C1,C2,C3,sy);
86     if kp==1
87         rk(k)=rk(k)+T(X(k))*Gewichte*...
88         sum(Sigma.*[dxi(z,:)',dyi(z,:)]*IJM,zeros(NQ,2)],2);
89     end
90     if kp==2
91         rk(k)=rk(k)+T(X(k))*Gewichte*...
92         sum(Sigma.*[zeros(NQ,2),[dxi(z,:)',dyi(z,:)]*IJM],2);
93     end
94     for q=1:2:3
95         p1=Patch(k,perm2(q,1));p2=Patch(k,perm2(q,2));
96         if (Kante(Kantennr(p1,p2))==2)
97             T1=X(k);
98             TN=find(Kantennr2Elemente(Kantennr(p1,p2),:)==T1);
99             T2=Kantennr2Elemente(Kantennr(p1,p2),TN);
100            if (q==1)
101                rk(k)=rk(k)-(mu0_1{kp}(T1,Kantennr(p1,p2))-...
102                    mu0_2{kp}(T2,Kantennr(p2,p1)))/2;
103            else
104                rk(k)=rk(k)-(mu0_2{kp}(T1,Kantennr(p1,p2))-...
105                    mu0_1{kp}(T2,Kantennr(p2,p1)))/2;
106            end
107            else
108                if (q==1)
109                    rk(k)=rk(k)-mu0_1{kp}(X(k),Kantennr(p1,p2));
110                else
111                    rk(k)=rk(k)-mu0_2{kp}(X(k),Kantennr(p1,p2));
112                end
113            end
114        end
115    end
116    if (Knoten(i)==2)
117        I1=zeros(size(Patch,1),1);
118        I1(1)=1;
119        for k=1:size(Patch,1)-1
120            Nachbarn=Kantennr2Elemente(Kantennr(Patch(I1(k),3),Patch(I1(k),1)),:);
121            NElement=Nachbarn(find(Nachbarn~=X(I1(k))));
122            I1(k+1)=find(NElement==X);
123        end
124        b=rk(I1);
125        KN1=diag(Kantennr(Patch(I1,1),Patch(I1,2)));
126        KN2=diag(Kantennr(Patch(I1,3),Patch(I1,1)));
127        A=spdiags(1/2*ones(size(Patch,1),1)*[-1,2,-1],[-1,0,1],...
128            size(Patch,1),size(Patch,1));
129        A(size(Patch,1),1)=-1/2;A(1,size(Patch,1))=-1/2;
130        b=b-sum(b)/size(Patch,1);
131        x=(A'*A+ones(size(Patch,1),size(Patch,1)))\b;
132        sigma{kp}(X(I1),i)=A'*x;
133    else
134        I1=zeros(size(Patch,1),1);
135        E=diag(Aussenkanten(Patch(:,3),Patch(:,1)'));
136        I1(1)=find(E);
137        for k=1:size(Patch,1)-1;

```

```

138     Nachbarn=Kantennr2Elemente(Kantennr(Patch(I1(k),1),Patch(I1(k),2)),:);
139     NElement=Nachbarn(find(Nachbarn~=x2Elemente(i,I1(k))));
140     I1(k+1)=find(NElement==X);
141     end
142     b=rk(I1);
143     KN1=diag(Kantennr(Patch(I1(1:length(I1)),1),Patch(I1(1:length(I1)),2)));
144     KN2=diag(Kantennr(Patch(I1(1:length(I1)),3),Patch(I1(1:length(I1)),1)));
145     if ((Kante(KN1(length(I1)))==1)&(Kante(KN2(1))==1)) & ...
146         (size(Patch,1)>1)
147     %Neumann-Neumann
148     A=spdiags(1/2*ones(size(Patch,1),1)*[-1,2,-1],[-1,0,1], ...
149         size(Patch,1),size(Patch,1));
150     A(1,1)=1/2;A(size(Patch,1),size(Patch,1))=1/2;
151     b=b-sum(b)/size(Patch,1);
152     x=(A'*A+ones(size(Patch,1),size(Patch,1)))\b;
153     sigma{kp}(X(I1),i)=A'*x;
154     end
155     if ((Kante(KN1(length(I1)))==0)&(Kante(KN2(1))==0))
156     %Dirichlet-Dirichlet
157     if (size(Patch,1)>1)
158         A=spdiags(1/2*ones(size(Patch,1),1)*[-1,2,-1],[-1,0,1], ...
159             size(Patch,1),size(Patch,1));
160         A(1,1)=3/2;A(size(Patch,1),size(Patch,1))=3/2;
161         x=A\b;
162         sigma{kp}(X(I1),i)=x;
163     else
164         sigma{kp}(X(I1),i)=b/2;
165     end
166     end
167     if ((Kante(KN2(1))==0)&(Kante(KN1(length(I1)))==1))
168     %Dirichlet-Neumann
169     if (size(Patch,1)>1)
170         A=spdiags(1/2*ones(size(Patch,1),1)*[-1,2,-1],[-1,0,1], ...
171             size(Patch,1),size(Patch,1));
172         A(1,1)=3/2;A(size(Patch,1),size(Patch,1))=1/2;
173         x=A\b;
174         sigma{kp}(X(I1),i)=x;
175     else
176         sigma{kp}(X(I1),i)=b;
177     end
178     end
179     if ((Kante(KN2(1))==1)&(Kante(KN1(length(I1)))==0))
180     %Neumann-Dirichlet
181     if (size(Patch,1)>1)
182         A=spdiags(1/2*ones(size(Patch,1),1)*[-1,2,-1],[-1,0,1], ...
183             size(Patch,1),size(Patch,1));
184         A(1,1)=1/2;A(size(Patch,1),size(Patch,1))=3/2;
185         x=A\b;
186         sigma{kp}(X(I1),i)=x;
187     else
188         sigma{kp}(X(I1),i)=b;
189     end
190     end
191     end
192     end

```

```

193 end
194 %Bestimmung der Funktion mu
195 for j=1:size(Elemente,1)
196     for kp=1:2
197         for i=1:3
198             p1=Elemente(j,i);p2=Elemente(j,rem(i,3)+1);
199             if (Kante(Kantennr(p1,p2))==2)
200                 % Innenkante
201                     T1=j;
202                     TN=find(Kantennr2Elemente(Kantennr(p1,p2),: )~=T1);
203                     T2=Kantennr2Elemente(Kantennr(p1,p2),TN);
204                     mu_1{kp}(j,Kantennr(p1,p2))=(sigma{kp}(T1,p1)-sigma{kp}(T2,p1)+ ...
205                         mu0_1{kp}(T1,Kantennr(p1,p2))-mu0_2{kp}(T2,Kantennr(p1,p2)))/2;
206                     mu_2{kp}(j,Kantennr(p1,p2))=(sigma{kp}(T1,p2)-sigma{kp}(T2,p2)+ ...
207                         mu0_2{kp}(T1,Kantennr(p1,p2))-mu0_1{kp}(T2,Kantennr(p1,p2)))/2;
208                 end
209             if (Kante(Kantennr(p1,p2))==1)
210                 % Neumannkante
211                     mu_1{kp}(j,Kantennr(p1,p2))=mu0_1{kp}(j,Kantennr(p1,p2));
212                     mu_2{kp}(j,Kantennr(p1,p2))=mu0_2{kp}(j,Kantennr(p1,p2));
213                 end
214             if (Kante(Kantennr(p1,p2))==0)
215                 % Dirichletkante
216                     mu_1{kp}(j,Kantennr(p1,p2))=sigma{kp}(j,p1)+mu0_1{kp}(j,Kantennr(p1,p2));
217                     mu_2{kp}(j,Kantennr(p1,p2))=sigma{kp}(j,p2)+mu0_2{kp}(j,Kantennr(p1,p2));
218                 end
219             end
220         end
221     end
222 %Loesung der lokalen Probleme
223 Kantenbasis_P=fwertkante(ones(1,NQ1)-2*alpha1,Pmax);
224 if pmax>2
225     In=[1:3,4:4+(pmax-2),3+Pmax:3+Pmax+pmax-2,2+2*Pmax:...
226         2+2*Pmax+pmax-2,3*Pmax+1:3*Pmax+(pmax-1)*(pmax-2)/2];
227 else
228     if pmax>1
229         In=[1:3,4:4+(pmax-2),3+Pmax:3+Pmax+pmax-2,...
230             2+2*Pmax:2+2*Pmax+pmax-2];
231     else
232         In=[1:3];
233     end
234 end
235 for j=1:size(Elemente,1)
236     b=zeros(2*P_dim,1);b_Kante=zeros(2*P_dim,1);
237     x=zeros(2*P_dim,1);
238     Quadraturpunkte=BaryK'*Koordinaten(Elemente(j,:),:);
239     ft=f(Quadraturpunkte,t);
240     JM=Funktionalmatrix2(Koordinaten(Elemente(j,:),:));
241     IJM=inv(JM)';
242     iDOF=DOF(j,:);idof=dof(j,:);nz=find(idof);
243     DXi=DX.*(sign(iDOF)')*ones(1,NQ));
244     DYi=DY.*(sign(iDOF)')*ones(1,NQ));
245     A=stema2l(JM,iDOF,DXi,DYi,Gewichte);
246     basis_ref=(sign(iDOF)')*ones(1,NQ)).*basis_P;
247     b=b+T(j)*reshape((basis_ref*(Gewichte'*[1,1].*ft))',2*P_dim,1);

```

```

248 basis_ref=(sign(idof(nz))'*ones(1,NQ)).*basis_p(nz,:);
249 Sigma=SpannungC4(u,u0,S0(j,:),A0(j,:),idof,nz,dx,dy,reshape(JM,1,4), ...
250 Gewichte,lambda,mu,C1,C2,C3,sy);
251 for i=1:size(idof,2)
252     b(2*i-1)=b(2*i-1)-T(j)*Gewichte* ...
253         sum(Sigma.*[[DXi(i,:)'],DYi(i,:)']*IJM,zeros(NQ,2)),2);
254     b(2*i)=b(2*i)-T(j)*Gewichte* ...
255         sum(Sigma.*[zeros(NQ,2),[DXi(i,:)'],DYi(i,:)']*IJM,2);
256 end
257 B=sparse(2*P_dim,2*P_dim);
258 W=zeros(2*P_dim,1);
259 maske=zeros(2*P_dim,1);
260 for s=1:3
261     Is=[s,rem(s,3)+1,(s-1)*(Pmax-1)+4:s*(Pmax-1)+3];
262     Ik=[s,rem(s,3)+1,(s-1)*(pmax-1)+4:s*(pmax-1)+3];
263     nz_Kante=find(dof(j,Ik));
264     s1=Elemente(j,s);s2=Elemente(j,rem(s,3)+1);
265     lk=norm(Koordinaten(s1,:)-Koordinaten(s2,:));
266     M_Kante=lk*MMatrix(Kantenbasis_p(nz_Kante,:),dof(j,Ik(nz_Kante)),...
267         Gewichte_Kante);
268     r_Kante1=zeros(length(dof(j,nz_Kante)),1);
269     r_Kante2=zeros(length(dof(j,nz_Kante)),1);
270     Seite=Kantennr(Elemente(j,s),Elemente(j,rem(s,3)+1));
271     if (Kante(Seite)==1)
272         nk=[0 1;-1 0]*(Koordinaten(s2,:)-Koordinaten(s1,:))'/lk;
273         gk=g([alpha1;beta1]'*[Koordinaten(s1,:);Koordinaten(s2,:)],nk',t);
274     else
275         r_Kante1(1)=mu_1{1}(j,Seite);
276         r_Kante1(2)=mu_2{1}(j,Seite);
277         r_Kante1(3:end)=-b(2*ln(Ik(nz_Kante(3:end))))-1);
278         r_Kante2(1)=mu_1{2}(j,Seite);
279         r_Kante2(2)=mu_2{2}(j,Seite);
280         r_Kante2(3:end)=-b(2*ln(Ik(nz_Kante(3:end)))));
281         x_Kante1=M_Kante\r_Kante1;
282         x_Kante2=M_Kante\r_Kante2;
283         gk1=(sign(dof(j,Ik(nz_Kante)))'*ones(1,NQ1)).* ...
284             Kantenbasis_p(nz_Kante,:))'*x_Kante1;
285         gk2=(sign(dof(j,Ik(nz_Kante)))'*ones(1,NQ1)).* ...
286             Kantenbasis_p(nz_Kante,:))'*x_Kante2;
287         gk=[gk1,gk2];
288     end
289     Is2=reshape([2*Is-1;2*Is],2*length(Is),1);
290     b_Kante(Is2)=b_Kante(Is2)+lk*reshape(((sign(DOF(j,Is))'*ones(1,NQ1)).* ...
291         Kantenbasis_P*(Gewichte_Kante'*[1,1].*gk))',2*length(Is),1);
292     if (Kante(Kantennr(Elemente(j,s),Elemente(j,rem(s,3)+1)))==0)
293         maske(2*Is-1)=ones(size(Is,2),1);
294         maske(2*Is)=ones(size(Is,2),1);
295     end
296 end
297 b=b+b_Kante;
298 if length(find(maske))>0
299     freieKnoten=find(~maske);
300     x(freieKnoten)=A(freieKnoten,freieKnoten)\b(freieKnoten);
301 else
302     [QM,RM]=qr(A);

```

```

303     c=QM'*b;
304     II2=find(abs(diag(RM)/max(abs(diag(RM))))<10^(-10));
305     I=[1:size(RM,2)]';I(II2)=[];
306     x(I)=RM(I,I)\c(I);
307     end
308     eta_N(j)=sqrt(x'*A*x);
309     end
310     return

```

Das folgende in Zeile 242 aufgerufene Unterprogramm `stema21.c` berechnet die bei der Lösung der lokalen Probleme auftretenden Steifigkeitsmatrix.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4  #include "mex.h"
5
6  double *CreateDoubleVector(int len)
7  {
8      return malloc(len*sizeof(double));
9  }
10
11 void STEMA(double M[],double jm[], double iDOF[], double DXi[], double DYi[],
12           double Gewichte[],int nd, int nq)
13 {
14     double *Nx,*Ny;
15     double f,dtm,IJ[2][2],tmp[2];
16     int i,j,k,i1,j1;
17
18     Nx=CreateDoubleVector(nd); Ny=CreateDoubleVector(nd);
19     dtm=jm[0]*jm[3]-jm[1]*jm[2];
20     IJ[0][0]=jm[3]/dtm; IJ[0][1]=-jm[2]/dtm;
21     IJ[1][0]=-jm[1]/dtm;IJ[1][1]=jm[0]/dtm;
22
23     for (j=0;j<4*nd*nd;j++) M[j]=0;
24
25     for (k=0;k<nq;k++)
26     {
27         f=dtm*sqrt(3.0)*Gewichte[k];
28         for (j=0;j<nd;j++)
29         {
30             Nx[j]=IJ[0][0]*DXi[j+nd*k]+IJ[0][1]*DYi[j+nd*k];
31             Ny[j]=IJ[1][0]*DXi[j+nd*k]+IJ[1][1]*DYi[j+nd*k];
32         }
33
34         for (i=0;i<2*nd;i+=2)
35         {
36             i1=(int) floor(i/2);
37             for (j=0;j<2*nd;j+=2)
38             {
39                 j1=(int) floor(j/2);
40                 M[i+j*2*nd]=M[i+j*2*nd]+f*(Nx[i1]*Nx[j1]+0.5*Ny[i1]*Ny[j1]);
41                 M[i+(j+1)*2*nd]=M[i+(j+1)*2*nd]+f*0.5*Ny[i1]*Nx[j1];
42                 M[(i+1)+j*2*nd]=M[(i+1)+j*2*nd]+f*0.5*Nx[i1]*Ny[j1];
43                 M[(i+1)+(j+1)*2*nd]=M[(i+1)+(j+1)*2*nd]+

```

```

44                                     f*(0.5*Nx[i1]*Nx[j1]+Ny[i1]*Ny[j1]);
45     }
46 }
47 }
48 free(Nx);Nx=NULL;free(Ny);Ny=NULL;
49 }
50
51
52 /* Gateway function */
53
54 void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[])
55 {
56     double *M,*jm,*iDOF,*DXi,*DYi,*Gewichte;
57     int nd,nq,n;
58
59
60     /*Check for proper number of arguments */
61     if (nrhs!=5)
62         mexErrMsgTxt("5 input required!");
63     if (nlhs!=1)
64         mexErrMsgTxt("1 output required!");
65
66
67     nd=mxGetN(prhs[1]);nq=mxGetN(prhs[4]);
68
69     jm         =mxGetPr(prhs[0]); iDOF         =mxGetPr(prhs[1]);
70     DXi        =mxGetPr(prhs[2]); DYi        =mxGetPr(prhs[3]);
71     Gewichte   =mxGetPr(prhs[4]);
72
73
74     /*Set the output pointer to the output matrix. */
75     plhs[0]=mxCreateDoubleMatrix(2*nd,2*nd,mxREAL);
76
77     /*Create a C pointer to a copy of the output matrix */
78     M=mxGetPr(plhs[0]);
79
80     /* Assemble STEMA */
81     STEMA(M,jm,iDOF,DXi,DYi,Gewichte,nd,nq);
82 }

```

A.7 Fehlerschätzer η_L : Patch-Residuen-Schätzer

Dieser Abschnitt beschreibt die Implementierung des Patch-Residuen-Schätzers aus Abschnitt 4.4. Das Programm `APosterioriL1.m` löst die lokalen Probleme aus Definition 4.49 für jeden Patch.

Zeilen 19-41 ermitteln lokale Größen zur Anwendung der üblichen Finite Elemente-Methode auf ein Problem, das für den Patch mit der Knotennummer z gestellt ist. Zeilen 19-37 verwenden die Implementierung aus [26] von S. Funken.

Die lokalen Probleme sind in (4.172) definiert, wobei für die rechte Seite in der

Implementierung der Ausdruck

$$l_z(\mathbf{v}_h) = \int_{\Omega_z} \varphi_z \mathbf{f}(t_{k-1+\theta}) \cdot \mathbf{v}_h \, d\mathbf{x} + \int_{\partial\Omega_z \cap \Gamma_N} \varphi_z \mathbf{g}(t_{k-1+\theta}) \cdot \mathbf{v}_h \, ds \quad (1.8)$$

$$- \int_{\Omega_z} \varphi_z \boldsymbol{\sigma}_{h,k-1+\theta} : \mathbf{D}\mathbf{v} \, d\mathbf{x} - \int_{\Omega_z} (\boldsymbol{\sigma}_{h,k-1+\theta} \cdot \nabla \varphi_z) \cdot \mathbf{v} \, d\mathbf{x}$$

verwendet wird. Der 1. Summand von (1.8) wird in Zeile 63 berechnet, der 4. Summand in den Zeilen 64,65, der 3. Summand in den Zeilen 66-71 und der 2. Summand in den Zeilen 74-97. Zeile 59 ruft eine C-Routine zur Bestimmung der lokalen Steifigkeitsmatrix auf, Zeile 60,61 führen die Assemblierung zur Gesamtsteifigkeitsmatrix für den Patch durch. Zeilen 98-116 lösen die lokalen Probleme, wie in Definition 4.49 beschrieben.

```

1  function eta_L=APosterioriL(u,u0,S0,A0,t,t0,Pmax)
2
3  global Koordinaten Elemente Dirichlet Neumann p dof Kantennr ...
4      lambda mu Gewichte BaryK NQ Gewichte1 alpha1 beta1 NQ1 P_ref DX DY ...
5      dx dy basis_P C1 C2 C3 lambda mu sy
6
7  eta_L=zeros(size(Koordinaten,1),1);
8
9
10 perm=[1,2,3;2,3,1;3,1,2];
11 x2Elemente=zeros(size(Koordinaten,1),8);
12 for i=1:size(Elemente,1)
13     for k=1:3
14         x2Elemente(Elemente(i,k),nnz(x2Elemente(Elemente(i,k),:))+1)=i;
15     end
16 end
17
18 for z=1:size(Koordinaten,1)
19     X=x2Elemente(z,1:nnz(x2Elemente(z,:)));
20     Patch=Elemente(X,:);
21     Z=unique(Patch);
22     locKoordinaten=Koordinaten(Z,:);
23     i2fi=1:max(Z);
24     i2fi(Z)=1:length(Z);
25     Patch=i2fi(Patch);
26     if ~isempty(Dirichlet)
27         D=find(Dirichlet(:,1)==z | Dirichlet(:,2)==z);
28         locDirichlet=i2fi(Dirichlet(D,:));
29     else
30         locDirichlet=[];
31     end
32     if ~isempty(Neumann)
33         D=find(Neumann(:,1)==z | Neumann(:,2)==z);
34         locNeumann=i2fi(Neumann(D,:));
35     else
36         locNeumann=[];
37     end
38
39     [locKantennr,locAussenkanten]=GeneriereKantennr(Patch,locKoordinaten);
40     DOF=erstelledof2(locKoordinaten,Patch,Pmax*ones(size(Patch,1),1),locKantennr);

```



```

41 N=max(max(abs(DOF)));
42 A=zeros(2*N,2*N);L=zeros(2*N,2);b=zeros(2*N,1);x=zeros(2*N,1);
43 for i=1:size(Patch,1)
44     T=det([1,1,1;locKoordinaten(Patch(i,:),:)])/2;
45     Quadraturpunkte=BaryK'*locKoordinaten(Patch(i,:),:);
46     ft=f(Quadraturpunkte,t);
47     JM=Funktionalmatrix2(locKoordinaten(Patch(i,:),:));
48     IJM=inv(JM)';
49     iDOF=DOF(i,:);idof=dof(X(i,:),:);nz=find(idof);
50     DXi=DX.*(sign(iDOF)'*ones(1,NQ));
51     DYi=DY.*(sign(iDOF)'*ones(1,NQ));
52     dxi=dx(nz,:).*(sign(idof(nz))'*ones(1,NQ));
53     dyi=dy(nz,:).*(sign(idof(nz))'*ones(1,NQ));
54     Sigma=SpannungC4(u,u0,S0(X(i,:),:),A0(X(i,:),:),idof,nz,dx,dy,reshape(JM,1,4), ...
55         Gewichte,lambda,mu,C1,C2,C3,sy);
56     jj=find(Elemente(X(i,:),:)==z);
57     dphi=IJM'*[dxi(jj,1);dyi(jj,1)];
58     basis_ref=(sign(iDOF)'*ones(1,NQ)).*basis_P;
59     S=stemall(JM,iDOF,DXi,DYi,basis_ref(jj,:),Gewichte);
60     I1=[2*abs(iDOF)-1];I2=[2*abs(iDOF)];I=reshape([I1;I2],1,2*size(iDOF,2));
61     A(I,I)=A(I,I)+S;
62     gb_ref=T*Gewichte.*basis_ref(jj,:);
63     b(I)=b(I)+reshape((basis_ref*((gb_ref)'*[1,1].*ft))',2*length(iDOF),1);
64     b(I)=b(I)-T*reshape((basis_ref*(((Gewichte)'*[1,1]).* ...
65         [Sigma*[dphi;0;0],Sigma*[0;0;dphi]]))',length(I),1);
66     for j=1:size(iDOF,2)
67         b(I(2*j-1))=b(I(2*j-1))-gb_ref* ...
68             sum(Sigma.*[DXi(j,:)',DYi(j,:)]*IJM,zeros(NQ,2)),2);
69         b(I(2*j))=b(I(2*j))-gb_ref* ...
70             sum(Sigma.*[zeros(NQ,2),[DXi(j,:)',DYi(j,:)]*IJM],2);
71     end
72 end
73 %Kantenanteil
74 if ~isempty(locNeumann)
75     KV=locKoordinaten(locNeumann(:,2),:)-locKoordinaten(locNeumann(:,1),:);
76     Laenge=sqrt(sum((KV.*KV)')));
77     Basiswerte=fwertkante(ones(1,NQ1)-2*alpha1,Pmax);
78     for i=1:size(locNeumann,1)
79         n=KV(i,:)*[0,-1;1,0]/Laenge(i);
80         E=locAussenkanten(locNeumann(i,1),locNeumann(i,2));
81         Seite=locAussenkanten(locNeumann(i,2),locNeumann(i,1));
82         jj=find(Elemente(X(E),:)==z);
83         s1=Seite;s2=rem(Seite,3)+1;
84         pos=find([s1,s2]==jj);
85         if length(pos)>0
86             if pos==1 v=alpha1; end
87             if pos==2 v=beta1; end
88         else
89             v=zeros(1,size(Gewichte1,2));
90         end
91         g_wert=g([alpha1;beta1]'*locKoordinaten(locNeumann(i,:),:),n,t);
92         kante_werte=Basiswerte*(((Gewichte1.*v)'*[1,1]).*g_wert)*Laenge(i)/2;
93         II=[Seite,rem(Seite,3)+1,(Seite-1)*(Pmax-1)+4:Seite*(Pmax-1)+3];
94         I=reshape([2*abs(DOF(E,II))-1;2*abs(DOF(E,II))],2*length(II),1);
95         b(I)=b(I)+reshape([1;1]*sign(DOF(E,II)).*kante_werte',2*length(II),1);

```

```

96     end
97 end
98 if ~isempty(locDirichlet)
99     maske=zeros(2*N,1);
100    for i=1:size(locDirichlet,1)
101        E=locAussenkanten(locDirichlet(i,1),locDirichlet(i,2));
102        Seite=locAussenkanten(locDirichlet(i,2),locDirichlet(i,1));
103        I=[find(locDirichlet(i,1)==Patch(E,:)),find(locDirichlet(i,2)==...
104            Patch(E,:)),(Seite-1)*(Pmax-1)+4:Seite*(Pmax-1)+3];
105        II=reshape([2*abs(DOF(E,I))-1;2*abs(DOF(E,I))],1,2*size(I,2));
106        maske(II)=ones(length(II),1);
107    end
108    freieKnoten=find(~maske);
109    x(freieKnoten)=A(freieKnoten,freieKnoten)\b(freieKnoten);
110 else
111     [QM,RM]=qr(A);
112     c=QM'*b;
113     II2=find(abs(diag(RM)/max(abs(diag(RM))))<10^(-10));
114     I=[1:size(RM,2)];I(II2)=[];
115     x(I)=RM(I,I)\c(I);
116 end
117 %Bestimmen der Norm
118     eta_L(z)=sqrt(x'*A*x);
119 end

```

Das folgende in Zeile 59 aufgerufene Unterprogramm `stema11.c` berechnet die bei der Lösung der lokalen Probleme auftretende Steifigkeitsmatrix.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4  #include "mex.h"
5
6  double *CreateDoubleVector(int len)
7  {
8      return malloc(len*sizeof(double));
9  }
10
11
12 void STEMA(double M[],double jm[], double iDOF[], double DXi[], double DYi[],
13           double basis[], double Gewichte[],int nd, int nq)
14 {
15     double *Nx,*Ny,f,dtm,IJ[2][2],tmp[2];
16     int i,j,k,i1,j1;
17
18     Nx=CreateDoubleVector(nd);Ny=CreateDoubleVector(nd);
19
20     dtm=jm[0]*jm[3]-jm[1]*jm[2];
21     IJ[0][0]=jm[3]/dtm; IJ[0][1]=-jm[2]/dtm;
22     IJ[1][0]=-jm[1]/dtm;IJ[1][1]=jm[0]/dtm;
23
24     for (j=0;j<4*nd*nd;j++) M[j]=0;
25
26     for (k=0;k<nq;k++)
27     {

```

```

28     f=dtm*sqrt(3.0)*Gewichte[k]*basis[k];
29
30     for (j=0;j<nd;j++)
31     {
32         Nx[j]=IJ[0][0]*DXi[j+nd*k]+IJ[0][1]*DYi[j+nd*k];
33         Ny[j]=IJ[1][0]*DXi[j+nd*k]+IJ[1][1]*DYi[j+nd*k];
34     }
35
36     for (i=0;i<2*nd;i+=2)
37     {
38         i1=(int) floor(i/2);
39         for (j=0;j<2*nd;j+=2)
40         {
41             j1=(int) floor(j/2);
42             M[i+j*2*nd]=M[i+j*2*nd]+f*(Nx[i1]*Nx[j1]+0.5*Ny[i1]*Ny[j1]);
43             M[i+(j+1)*2*nd]=M[i+(j+1)*2*nd]+f*0.5*Ny[i1]*Nx[j1];
44             M[(i+1)+j*2*nd]=M[(i+1)+j*2*nd]+f*0.5*Nx[i1]*Ny[j1];
45             M[(i+1)+(j+1)*2*nd]=M[(i+1)+(j+1)*2*nd]+f*(0.5*Nx[i1]*Nx[j1]
46                                     +Ny[i1]*Ny[j1]);
47         }
48     }
49 }
50 free(Nx);Nx=NULL;free(Ny);Ny=NULL;
51 }
52
53
54 /* Gateway function */
55
56 void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[])
57 {
58     double *M,*jm,*iDOF,*DXi,*DYi,*basis_ref_jj,*Gewichte;
59     int nd,nq,n;
60
61
62     /*Check for proper number of arguments */
63     if (nrhs!=6)
64         mexErrMsgTxt("6 input required!");
65     if (nlhs!=1)
66         mexErrMsgTxt("1 output required!");
67
68
69     nd=mxGetN(prhs[1]); nq=mxGetN(prhs[5]);
70
71     jm          =mxGetPr(prhs[0]); iDOF          =mxGetPr(prhs[1]);
72     DXi         =mxGetPr(prhs[2]); DYi          =mxGetPr(prhs[3]);
73     basis_ref_jj=mxGetPr(prhs[4]); Gewichte     =mxGetPr(prhs[5]);
74
75
76     /*Set the output pointer to the output matrix. */
77     plhs[0]=mxCreateDoubleMatrix(2*nd,2*nd,mxREAL);
78
79     /*Create a C pointer to a copy of the output matrix */
80     M=mxGetPr(plhs[0]);
81
82     /* Assemble STEMA */

```

```
83     STEMA(M,jm,iDOF,DXi,DYi,basis_ref_jj,Gewichte,nd,nq);  
84 }
```

Literaturverzeichnis

- [1] Ainsworth, M. and Oden, J.T. A posteriori error estimators for second order elliptic systems: Part 1. Theoretical foundations and a posteriori error analysis. *Computers Math. Applic.*, 25(2):101–113, 1993.
- [2] Ainsworth, M. and Oden, J.T. A posteriori error estimators for second order elliptic systems: Part 2. An optimal order process for calculating self-equilibrating fluxes. *Computers Math. Applic.*, 26(9):75–87, 1993.
- [3] Ainsworth, M. and Oden, J.T. A unified approach to a posteriori error estimation using element residual methods. *Numer. Math.*, 65:23–50, 1993.
- [4] Ainsworth, M. and Oden, J.T. *A Posteriori Error Estimation in Finite Element Analysis*. Wiley, 2001.
- [5] Ainsworth, M. and Senior, B. Aspects of an adaptive hp-finite element method: Adaptive strategy, conforming approximation and efficient solvers. *Comput. Methods Appl. Mech. Engrg.*, 150:65–87, 1997.
- [6] Ainsworth, M. and Senior, B. An adaptive strategy for hp-finite element computations. *Applied Numerical Mathematics*, 26:165–178, 1998.
- [7] Albery, J. *Zeitdiskretisierungsverfahren für elastoplastische Probleme der Kontinuumsmechanik*. Dissertation, Lehrstuhl für Wissenschaftliches Rechnen, Christian-Albrechts-Universität zu Kiel, 2001.
- [8] Albery, J., Carstensen, C., Funken, S.A., and Klose, R. Matlab implementation of the finite element method in elasticity. *Computing*, 69:239–263, 2002.
- [9] Albery, J., Carstensen, C., and Zarrabi, D. Adaptive numerical analysis in primal elastoplasticity with hardening. *Comput. Methods Appl. Mech. Engrg.*, (171):175–204, 1998.
- [10] Babuška, I. and Dorr, M.R. Error estimates for the combined h and p versions of the finite element method. *Numer. Math.*, 37:257–277, 1981.
- [11] Babuška, I. and Guo, B.Q. Approximation properties of the h-p version of the finite element method. *Comput. Methods Appl. Mech. Engrg.*, 133:319–346, 1996.
- [12] Babuška, I. and Rheinboldt, W.C. A-posteriori error estimates for the finite element method. *Int. J. Numer. Methods Eng.*, 12:1597–1615, 1978.

- [13] Babuška, I. and Rheinboldt, W.C. Error estimates for adaptive finite element computations. *SIAM J. Numer. Anal.*, 15:763–754, 1978.
- [14] Babuška, I. and Suri, M. The optimal convergence rate of the p-version of the finite element method. *SIAM J. Numer. Anal.*, (24):750–776.
- [15] Babuška, I. and Suri, M. The h-p version of the finite element method with quasi uniform meshes. *Math. Modelling Numer. Anal.*, (21):199–238, 1987.
- [16] Babuška, I. and Suri, M. The p and hp-version of the finite element method, basic principles and properties. *SIAM Rev.*, (36(4)):578–632, 1994.
- [17] Bank, R.E. and Weiser, A. Some a posteriori error estimators for elliptic partial differential equations. *Math. Comp.*, 44(170):283–301, 1985.
- [18] Boecker, C. *Zur hp-Methode der FEM: Effiziente Implementation und Anwendungen*. Diplomarbeit, Lehrstuhl für Wissenschaftliches Rechnen, Christian-Albrechts-Universität zu Kiel, 1999.
- [19] Braess, D. *Finite Elemente: Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*. Springer, 1997.
- [20] Brenner, S.C. and Scott, L.R. *The mathematical theory of finite element methods*. Springer, 2002.
- [21] Brink, U. and Stein, E. A posteriori error estimation in large-strain elasticity using equilibrated local neumann problems. *Comput. Methods Appl. Mech. Eng.*, (161):77–101, 1998.
- [22] Carstensen, C. *Einführung in Computational Plasticity*. Sommerschule der Technischen Fakultät, Christian-Albrechts-Universität zu Kiel, 1999.
- [23] Carstensen, C. Numerical analysis of the primal problem of elastoplasticity with hardening. *Numerische Mathematik*, 82:577–597, 1999.
- [24] Carstensen, C. and Albery, J. Averaging techniques for reliable a posteriori fe-error control in elastoplasticity with hardening. *Comput. Methods Appl. Mech. Engrg.*, 2003.
- [25] Carstensen, C. and Funken, S.A. Constants in Clément-interpolation error and residual based a posteriori estimates in finite element methods. *East-West J. Numer. Math.*, 8(3):153–175, 2000.
- [26] Carstensen, C. and Funken, S.A. Fully reliable localised error control in the FEM. *SIAM J. Sci. Comp.*, 21 (4):1465–1484, 2000.
- [27] Carstensen, C. and Klose, R. *Wissenschaftliches Rechnen: Elastische Festkörper*. Vorlesungsmitschrift, Lehrstuhl für Wissenschaftliches Rechnen, Christian-Albrechts-Universität zu Kiel, 2000.

- [28] Carstensen, C. and Klose, R. Elastoviscoplastic finite element analysis in 100 lines of matlab. *Journal of Numerical Mathematics*, 10(3):157–192, 2002.
- [29] Carstensen, C. and Müller, S. Local stress regularity in scalar non-convex variational problems. *Preprint, Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig*, 2002.
- [30] Clément, Ph. Approximation by finite element functions using local regularisation. *RAIRO Anal. Numérique*, 9:77–84, 1999.
- [31] Düster, A. and Rank, E. A p-version finite element approach for two- and three dimensional problems of the j_2 flow theory with non-linear isotropic hardening. *International Journal for Numerical Methods in Engineering*, 53:49–63, 2001.
- [32] Düster, A. and Rank, E. The p-version of the finite element method compared to an adaptive h-version for the deformation theory of plasticity. *Computer Methods in Applied Mechanics and Engineering*, 190:1925–1935, 2001.
- [33] Evans, L.C. *Partial Differential Equations*. American Mathematica Society, 1998.
- [34] Hackbusch, W. *Iterative Lösung großer schwachbesetzter Gleichungssysteme*. Teubner, 1993.
- [35] Hackbusch, W. *Theorie und Numerik elliptischer Differentialgleichungen*. Teubner, 1996.
- [36] Han, W. and Reddy, B.D. *Plasticity: Mathematical Theory and Numerical Analysis*. Springer, 1999.
- [37] Haupt, P. *Continuum Mechanics and Theory of materials*. Springer, 1999.
- [38] Hecht, A. *Adaptive Finite-Elemente-Methoden für ein elastoplastisches 7-Parameter-Zylinderschalenmodell*. Dissertation (in Vorbereitung), Lehrstuhl für Wissenschaftliches Rechnen, Christian-Albrechts-Universität zu Kiel, 2003.
- [39] Heuer, N., Mellado, M.E., and Stephan, E.P. hp-adaptive two-level methods for boundary integral equations. *Computing*, 67:305–334, 2001.
- [40] Holzer, S. and Yosibash, Z. The p-version of the finite element method in incremental elasto-plastic analysis. *International Journal for Numerical Methods in Engineering*, 39:1859–1878, 1996.
- [41] Jeremic, B. and Xenophontos, C. Application of the p-version of the finite element method to elasto-plasticity with localization of deformation. *Communications in Numerical Methods in Engineering*, 15(12):867–876, 1999.
- [42] Johnson, C. and Hansbo, P. Adaptive finite element methods in computational mechanics. *Comput. Methods Appl. Mech. Eng.*, 101(1-3):143–181, 1992.

- [43] Ladeveze, D. and Leguillon, D. Error estimate procedure in the finite element method and applications. *SIAM J. Numer. Anal.*, 20(3):485–509, 1983.
- [44] Lemaitre, J. and Chaboche, J.-L. *Mechanics of solid materials*. Cambridge University Press, 1990.
- [45] MathWorks, Inc. *Matlab Function Reference*, volume 1-3.
- [46] Maugin, G.A. *The Thermomechanics of Plasticity and Fracture*. Cambridge University Press, 1992.
- [47] Melenk, J.M. and Wohlmuth, B.I. On residual-based a-posteriori error estimation in hp-fem. *Advances in Comput. Math.*, 15:311–331, 2001.
- [48] Oden, J.T. and Demkowicz, L.F. *Applied functional analysis*. CRC Press, 1996.
- [49] Ohnimus, S. *Theorie und Numerik residualer Fehlerschätzer für die Finite Elemente Methode*. Habilitationsschrift am Fachbereich Bauingenieur- und Vermessungswesen der Universität Hannover, 2000.
- [50] Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P. *Numerical Recipes in Fortran 77*. Cambridge University Press, 1996.
- [51] Quarteroni, A., Sacco, R., and Saleri, F. *Numerical Mathematics*. Springer, 2000.
- [52] Quarteroni, A. and Valli, A. *Numerical Approximation of Partial Differential Equations*. Springer, 1997.
- [53] Rannacher, R. and Suttmeier, F.-T. A posteriori error control in finite element methods via duality techniques: Application to perfect plasticity. *Comput. Mech.*, (21):123–133, 1998.
- [54] Rannacher, R. and Suttmeier, F.-T. An adaptive finite element methods for problems in perfect plasticity. *ZAMM*, 79(1):143–146, 1999.
- [55] Rannacher, R. and Suttmeier, F.-T. A posteriori error estimation and mesh adaption for finite element models in elasto-plasticity. *Comput. Methods Appl. Mech. Eng.*, 176(1-4):333–361, 1999.
- [56] Repin, S.I. and Xanthis, L.S. A posteriori error estimation for elasto-plastic problems based on duality theory. *Compu. Methods Appl. Mech. Engrg.*, (138):317–339, 1996.
- [57] Schwab, C. *p- and hp-finite element methods, theory and applications in solid and fluid mechanics*. Oxford Science Publications, 1998.
- [58] Simo, J.C. and Hughes, T.J.R. *Computational Inelasticity*. Springer, 1998.

- [59] Strouboulis, T., Babuška, I., and Gangaraj, K. Guaranteed computable bounds for the exact error in the finite element solution - part ii: Bounds for the energy norm of the error in two-dimensions. *Int. J. Numer. Methods Eng.*, (1-3):427–475, 2000.
- [60] Suquet, P.-M. Discontinuities and plasticity. In Moreau, J.J. and P.D. Paniotopoulos, editors, *Nonsmooth Mechanics and Applications, CISM Courses*, volume 302, pages 279–341. Springer-Verlag New York, 1988.
- [61] Szabo, B., Actis, R., and Holzer, S. volume 75, pages 395–416. Springer: New York, 1995.
- [62] Verfürth, R. A posteriori error estimation and adaptive mesh refinement techniques. *J. Comput. Appl. Math.*, 50:67–83, 1994.
- [63] Verfürth, R. *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*. Wiley-Teubner, Stuttgart, 1996.
- [64] Verfürth, R. Sharp error estimates for some quasi-interpolation operators. *Bericht, Ruhr-Universität Bochum*, 1997.
- [65] Verfürth, R. A review of a posteriori error estimation techniques for elasticity problems. *Comput. Methods Appl. Mech. Engrg.*, 176:419–440, 1999.
- [66] Wieners, C. An abstract interface for the description of material laws for the simulation of elastoplastic problems. *Preprint, Institut für Computeranwendungen III, Universität Stuttgart*, 2002.
- [67] Wieners, C. Efficient elasto-plastic simulation. *Preprint, Institut für Computeranwendungen III, Universität Stuttgart*, 2002.

Veröffentlichungsliste

Teile dieser Arbeit sind in [6] veröffentlicht.

1. J. Albery, C. Carstensen, S.A. Funken, R. Klose, *Matlab Implementation of the Finite-Element-Method in Elasticity*. Computing 69, 239-263 (2002)
2. C. Carstensen, R. Klose, *Guaranteed a posteriori finite element error control for the p -Laplace problem*. Preprint 23. Juli 2001 (Zur Veröffentlichung eingereicht).
3. C. Carstensen, R. Klose, *A posteriori error estimation of the p -Laplace problem*. In: Proceedings in Applied Mathematics and Mechanics, Wiley InterScience 1 (2001).
4. C. Carstensen, S. Bartels, R. Klose, *An experimental survey of a posteriori courant finite element error control for the poisson equation*. Advances in Computational Mathematics (2001).
5. C. Carstensen, R. Klose, *Wissenschaftliches Rechnen: Elastische Festkörper*. Vorlesungsmitschrift, Lehrstuhl für Wissenschaftliches Rechnen, Christian-Albrechts-Universität zu Kiel, (31. Juli 2000).
6. C. Carstensen, R. Klose, *Elastoviscoplastic Finite Element Analysis in 100 lines of Matlab*. Journal of Numerical Mathematics, 10,3 (2002), 157-192