

Hocheffiziente Formulierung und Implementierung finiter Elemente für transiente Analysen mit expliziter Zeitintegration

Zur Erlangung des akademischen Grades eines

DOKTOR-INGENIEURS

an der Fakultät für

Bauingenieur-, Geo- und Umweltwissenschaften
des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

Dipl.-Ing. Steffen Mattern
aus Neustadt a. d. Weinstraße

Tag der mündlichen Prüfung : 20. Juli 2012
Hauptreferent : Prof. Dr.-Ing. K. Schweizerhof
Korreferent : Prof. Dr.-Ing. habil. M. Bischoff

Karlsruhe (2012)

Kurzfassung

Forschung und Entwicklung auf dem Gebiet der Finite-Elemente-Methode ist geprägt von der theoretischen Herleitung numerischer Modelle sowie deren algorithmischen Aufbereitung und Umsetzung in ProgrammROUTINEN. Die Beseitigung ungewollter Versteifungseffekte, die vor allem bei finiten Elementen niedriger Ansatzordnung auftreten, wird häufig über eine Modifikation von Verschiebungs-, Verzerrungs- oder Spannungsverläufen durch entsprechende Annahmen erreicht, und erfordert somit einen Eingriff an verschiedenen Stellen der zugrundeliegenden ElementROUTINE. Eine übersichtliche Programmierung, die Fehler vermeiden und die Lesbarkeit der ROUTINEN steigern soll, wirkt sich hierbei oftmals ungünstig auf deren Effizienz aus. Umgekehrt kann eine sehr kompakte Implementierung unter Verwendung möglichst weniger Programmschritte die Fehleranfälligkeit begünstigen und somit die für die Umsetzung erforderliche Zeit erheblich steigern. Die konsequente Anwendung einer symbolischen Programmierumgebung wie ACEGEN [71, 73] erlaubt eine Programmierung, die den Schreibweisen der theoretischen Zusammenhänge sehr nahe ist und führt durch automatische Differentiation und Optimierung der Programmschritte gleichzeitig zu sehr effizienten ROUTINEN.

Im Zusammenhang mit expliziter Zeitintegration mit diagonalen (zusammengefassten) Massenmatrizen, bei der die erforderliche Rechenzeit für eine Simulation wegen kleiner Zeitschrittweiten und sehr effizienten globalen Operationen im Wesentlichen durch die Rechenschritte innerhalb der ElementROUTINE bestimmt wird, ist eine effiziente Implementierung obligatorisch. Die Anwendung von ACEGEN steigert dann sowohl die Effizienz des Entwicklungsprozesses – Modifikationen können schnell und fehlerfrei umgesetzt werden – als auch die der resultierenden Formulierung, was eine weniger zeitintensive Durchführung numerischer Testbeispiele ermöglicht.

Das vorgestellte Implementierungskonzept wird auf verschiedene Schalen und Volumenschalenformulierungen mit linearer und quadratischer Geometrie- und Verschiebungsinterpolation angewendet, wobei insbesondere die Umsetzung unterschiedlicher – teilweise aus dem Schrifttum für implizite Anwendungen bekannter – Strategien zur Beseitigung ungewollter Versteifungseffekte im Mittelpunkt steht. Hier eignet sich eine symbolische Programmierung besonders, da z. B. modifizierte Verzerrungsverläufe symbolisch angegeben werden können, erforderliche Ableitungen automatisch bestimmt und schließlich in Programmcode umgesetzt werden. An ausgewählten numerischen Beispielen werden die Effekte händischer und symbolischer Programmierung gegenübergestellt, sowie die verschiedenen gewählten Methoden hinsichtlich ihrer Effizienz, insbesondere im Kontext expliziter Zeitintegration, bewertet.

Abstract

Research and Development regarding finite element methods is characterized by a theoretical formulation of numerical models and an algorithmic preparation and implementation into program routines. The elimination of unwanted locking effects, present in low-order finite element approximations, is usually achieved by modifying displacement, strain or stress distributions with appropriate assumptions and hence requires changes in various parts of the underlying element routine. A well-arranged implementation in order to avoid programming errors and improve readability often increases the computational effort. On the other hand, a consolidated routine involving as few operations as possible, may increase the likelihood of errors and consequently the time necessary for an implementation. The consequent application of a symbolic programming environment such as ACEGEN [71, 73], enables an implementation very close to the syntax used for the theoretical relations. At the same time, the software provides automatic differentiation and simultaneous optimization of expressions, leading to very efficient routines.

In the context of explicit time integration with diagonal (lumped) mass matrices, the required computational time is dominated by the operations on element level, due to a small time step size and very efficient global operations. Thus, an efficient implementation of the element routines is mandatory. The application of ACEGEN simultaneously improves the efficiency of the development process – modifications can be realized fast and error-free – and of the computations applying the resulting formulations, thus reducing the computational effort regarding numerical examples.

The presented implementation concept is applied to standard and volumetric shell formulations with linear and quadratic interpolation of geometry and displacements, while it is especially focused on different strategies – partially known from literature for implicit applications – in order to eliminate artificial stiffness effects. A symbolic programming approach is particularly interesting here, as e. g. a modified strain distribution can be implemented directly and necessary derivatives are computed and transferred into the program code automatically. Specially selected numerical examples are chosen to compare the effort of manual and software supported implementations, as well as the different strategies against locking regarding efficiency, especially in the context of explicit time integration.

Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als akademischer Mitarbeiter am Institut für Mechanik des Karlsruher Instituts für Technologie. Die Grundlage

Meinen besonderen Dank möchte ich Herrn Prof. Dr.-Ing. K. Schweizerhof aussprechen. Seine wissenschaftliche Kompetenz und engagierte Unterstützung bei der Bearbeitung sowie seine stete Diskussionsbereitschaft haben diese Arbeit wesentlich geprägt.

Herrn Prof. Dr.-Ing. habil. Manfred Bischoff danke ich für die Übernahme des Korreferats, die zügige Durchsicht des Manuskripts sowie die wertvollen Hinweise.

Herrn Prof. Dr.-Ing. habil. Thomas Seelig und allen Kolleginnen und Kollegen des Instituts für Mechanik danke ich für ihre fachliche und persönliche Unterstützung sowie für das angenehme und produktive Arbeitsklima.

Ganz besonders möchte ich mich bei meiner Familie bedanken, die mich stets unterstützt und ermutigt hat. Die Geduld und der Rückhalt durch meine Frau Susanne und meine Tochter Sophia, haben erheblich zum Gelingen dieser Arbeit beigetragen.

Karlsruhe, 6. September 2012

Steffen Mattern

Inhalt

Bezeichnungen	V
Einleitung	1
1. Kontinuumsmechanische Grundlagen	7
1.1. Kinematik und Verzerrungen	7
1.2. Bilanzgleichungen	9
1.2.1. Massenbilanz	9
1.2.2. Impulsbilanz	9
1.2.3. Drallbilanz	10
1.3. Spannungsmaße und Stoffgesetze	11
1.3.1. Hyperelastizität	11
1.3.2. Elastoplastizität	12
1.4. Schwache Form des Gleichgewichts	15
1.5. Räumliche Diskretisierung	18
2. Lösung zeitabhängiger Probleme	21
2.1. Diskretisierung der Bewegungsgleichung	21
2.2. Explizite Zeitintegration	22
2.2.1. Das zentrale Differenzenverfahren	22
2.2.2. Zeitschrittweite	24
2.2.3. Implementierung	24
2.2.4. Dämpfung und dynamische Relaxation	25
2.2.5. Bulk Viskosität	26
2.2.6. Massenskalierung	29
2.3. Implizite Zeitintegration	30
2.3.1. Das NEWMARK-Verfahren	30
2.3.2. Wahl der Parameter	31
2.4. Effizienzaspekte expliziter und impliziter Verfahren	31
3. Finite Elemente für Schalenprobleme	33
3.1. Degenerierte Schalenelemente	33
3.1.1. Das Degenerationskonzept	33
3.1.2. Diskretisierung mit degenerierten Schalenelementen	35
3.2. Volumenschalenelemente	40
3.2.1. Das 'Solid-Shell'-Konzept	40
3.2.2. Diskretisierung mit Solid-Shell-Elementen	42
3.3. Massenmatrix	43
3.3.1. Grundlagen	43
3.3.2. Konsistente Massenmatrix	43

3.3.3. Diagonalisierte Massenmatrix	44
3.4. Vermeidung von Versteifungseffekten	48
3.4.1. Definition und Klassifizierung von ‘Locking’	48
3.4.2. Die Methode der angenommenen Verzerrungen	49
3.4.3. Die Methode der erweiterten Verschiebungen	56
3.4.4. Die Methode der erweiterten Verzerrungen	62
3.4.5. Die Methode der inkompatiblen Moden	69
3.5. Kontakt zwischen verformbaren und starren Strukturen (STAS)	77
4. Symbolische Programmierung	79
4.1. Definition: ‘Effiziente Programmierung’	79
4.2. Einführung in ACEGEN	81
4.3. Elementroutinen für explizite Zeitintegration	83
4.3.1. Steifigkeitsmatrix und Residuum	83
4.3.2. Massenmatrix und Diagonalisierung	87
4.3.3. Spannungsrückrechnung	88
4.4. Implementierte Elementformulierungen	89
4.4.1. Nomenklatur	90
4.4.2. Degenerierte Schalenelemente	90
4.4.3. Solid-Shell-Elemente	96
4.4.4. Kontaktelemente	101
4.4.5. Tabellarische Zusammenstellung der verwendeten Elementformulierungen	103
5. Numerische Beispiele	107
5.1. Kragarm unter Einzellast	107
5.1.1. Problembeschreibung und Diskretisierung	107
5.1.2. Bewertung der Effizienz	108
5.1.3. Lineare quasi-statische Analysen	111
5.1.4. Nichtlineare quasi-statische Analysen	113
5.1.5. Dynamische Analysen	119
5.2. Halbkugelschale unter dynamischer Biegebeanspruchung	124
5.2.1. Problembeschreibung	124
5.2.2. Simulation mit degenerierten Schalenelementen	124
5.2.3. Simulation mit Volumenschalenelementen	128
5.2.4. Vergleich beider Elementfamilien	134
5.3. Vollkugelschale mit Loch	136
5.3.1. Problembeschreibung und Diskretisierung	136
5.3.2. Diskussion der Ergebnisse bei elastischem Materialverhalten	137
5.3.3. Diskussion der Ergebnisse mit Plastizität	142
Zusammenfassung und Ausblick	149
Schrifttum	153
Anhang	163

A. Elemente der Tensoralgebra	163
A.1. Grundlegende Rechenoperationen	163
A.2. Ableitungsregeln für Tensoren	164
A.3. Invarianten und Eigenwerte eines Tensors	164
A.4. Eigenrichtungen eines symmetrischen reellwertigen Tensors	164
B. Programmierung mit ACEGEN	167
B.1. Prinzipbeispiel – Matrizenoperation	167
B.2. Prinzipbeispiel zum Implementierungskonzept	170
B.2.1. 4-Knoten-Scheibenelement	170
B.2.2. Händische Implementierung	171
B.2.3. Implementierung mit ACEGEN	174
B.3. Implementierung einer Materialformulierung mit Plastizität	181
C. Beispiel zur Diagonalisierung der Elementmassenmatrix	185
C.1. Konsistente Massenmatrix	185
C.2. 1-Element Beispiel – regelmäßige Geometrie	187
C.3. 1-Element Beispiel – unregelmäßige Geometrie	189
C.4. Folgerung für die Integration der Massenmatrix	191

Bezeichnungen

Da im Schrifttum verschiedene Notationen verbreitet sind, wird in der folgenden Liste ein Überblick über die in der Arbeit verwendeten Formelzeichen gegeben. Grundsätzlich werden Vektoren, Matrizen sowie Tensoren zweiter Stufe mit fettgedruckten Buchstaben bezeichnet, die Unterscheidung ergibt sich aus dem jeweiligen Kontext. Skalare sind in Normalschrift angegeben. Einzelne Symbole treten in der Arbeit nur lokal auf, etwa bei der Beschreibung von Beispielen. Diese sind in der folgenden Liste nicht aufgeführt, sondern werden an entsprechender Stelle erläutert.

a	Schalendirektor in der Momentankonfiguration
A	Schalendirektor in der Referenzkonfiguration
α	Bezeichnung für verschiedene Skalierungsfaktoren (Masse, Zeitschritt)
α	Vektor der erweiterten Verzerrungsfreiheitsgrade
b	linker CAUCHY-GREEN-Tensor
b^e, b^p	elastischer, bzw. plastischer Anteil des linken CAUCHY-GREEN-Tensors
c_m, c_K	masse-, bzw. steifigkeitsproportionaler Dämpfungsfaktor
c_s	Wellenausbreitungsgeschwindigkeit im Festkörper
C_0, C_1	quadratischer, bzw. linearer Parameter für Bulk-Viskosität
C	rechter CAUCHY-GREEN-Tensor
C^e, C^p	elastischer, bzw. plastischer Anteil des rechten CAUCHY-GREEN-Tensors
C^d	Dämpfungsmatrix
\mathbb{C}	vierstufiger Stofftensor
d	Vektor aller Knotenfreiheitsgrade eines Elements
d_i	Vektor der Knotenfreiheitsgrade des Knotens i bei Schalenelementen
d_{io}, d_{iu}	Knotenfreiheitsgrade oben/ unten des Elementknotenpaares i bei Volumenschalenelementen
\bar{d}	Vektor aller Verschiebungsfreiheitsgrade eines Schalenelements
d_{IM}	Vektor der inkompatiblen Freiheitsgrade
\tilde{d}	Vektor aller kompatiblen und inkompatiblen Freiheitsgrade

$\dot{\mathbf{d}}, \ddot{\mathbf{d}}$	Vektor der Knotengeschwindigkeiten, bzw. -beschleunigungen
\mathbf{e}	ALMANZI-Verzerrungstensor
\mathbf{E}	GREEN-LAGRANGE-Verzerrungstensor
$\mathbf{E}_{\text{lin}}, \mathbf{E}_{\text{nl}}$	linearer, bzw. nichtlinearer Anteil des GREEN-LAGRANGE-Verzerrungstensors
E_{ij}	Komponente $[i, j]$ des GREEN-LAGRANGE-Verzerrungstensors
$[\mathbf{E}]$	Komponentenmatrix des GREEN-LAGRANGE-Verzerrungstensors
$\boldsymbol{\varepsilon}$	Vektor der Verzerrungskomponenten
$\boldsymbol{\varepsilon}^{tr}$	'trial'-Verzerrungen für Plastizität
$\boldsymbol{\varepsilon}^e, \boldsymbol{\varepsilon}^p$	elastische, bzw. plastische Verzerrungen
$\mathbf{f}^{\text{int}}, \mathbf{f}^{\text{ext}}$	Vektor der inneren, bzw. äußeren Knotenkräfte
$\mathbf{f}_{\text{Kontakt}}^{\text{int}}$	Vektor der inneren Knotenkräfte in Folge Kontakt
\mathbf{F}	Deformationsgradient
$\mathbf{F}^e, \mathbf{F}^p$	elastischer, bzw. plastischer Anteil des Deformationsgradienten
$\mathbf{g}_i, \mathbf{g}^i$	ko-, bzw. kontravarianter Basisvektor in der Momentankonfiguration
$\mathbf{G}_i, \mathbf{G}^i$	ko-, bzw. kontravarianter Basisvektor in der Referenzkonfiguration
\mathbf{J}	JACOBI-Matrix (in Kapitel 1: Drehimpuls)
\mathbf{J}_0	JACOBI-Matrix ausgewertet am Elementmittelpunkt
\mathbf{L}	Impuls
λ	LAMÉ-Konstante
m	Masse
\mathbf{M}, \mathbf{M}^e	System-, bzw. Elementmassenmatrix
$\mathbf{M}_c, \mathbf{M}_d$	konsistente, bzw. diagonalisierte Massenmatrix
$[\mathbf{M}_c^e]_{ij}$	Eintrag $[i, j]$ der konsistenten Elementmassenmatrix ¹
μ	LAMÉ-Konstante
\mathbf{N}, \mathbf{N}_i	Matrix der Ansatzfunktionen, bzw. Untermatrix für den Knoten i
N_i	Ansatzfunktion, dem Knoten i zugeordnet
\mathbf{N}_{EAS}	Ansätze für die erweiterten Verzerrungen
\mathbf{N}_{IM}	Ansätze für die inkompatiblen Freiheitsgrade
p	Penetration

¹hier wird aus Gründen der Übersichtlichkeit wegen der verschiedenen Indices von der üblichen Konvention, skalare Größen in Normalschrift zu notieren abgewichen

\mathbf{P}	1. PIOLA-KIRCHHOFF-Spannungstensor
Π^{int}	innere Energie
$\Pi^{\text{ext}}, \Pi^{\text{kin}}$	Arbeit der äußeren Kräfte, kinetische Energie
$\delta \Pi$	erste Variation des Potentials
ρ	Dichte
\mathbf{S}	2. PIOLA-KIRCHHOFF-Spannungstensor
$\boldsymbol{\sigma}$	CAUCHY-Spannungstensor
$\boldsymbol{\sigma}^{\text{tr}}$	'trial'-Spannungen für Plastizität
$\boldsymbol{\tau}$	KIRCHHOFF Spannungstensor
\mathbf{u}, \mathbf{u}^h	Verschiebungsvektor, bzw. diskretisierter Verschiebungsvektor
$\nabla \mathbf{u}$	lokaler Verschiebungsgradient nach ξ, η und ζ
V, v	Volumen eines Körpers in der Referenz- und Momentankonfiguration
$\Phi(\boldsymbol{\sigma}, \xi_\alpha)$	Fließbedingung
$\boldsymbol{\varphi}(\mathbf{X}, t)$	Konfiguration eines Körpers \mathcal{B}
\mathcal{W}	Formänderungsenergie
\mathbf{x}, \mathbf{x}^h	Ortsvektor, bzw. diskretisierter Ortsvektor in der Momentankonfiguration
\mathbf{X}, \mathbf{X}^h	Ortsvektor, bzw. diskretisierter Ortsvektor in der Referenzkonfiguration
\mathbf{X}_i	Knotenkoordinaten des Elementknotens i bei Schalenelementen
$\mathbf{X}_{io}, \mathbf{X}_{iu}$	Knotenkoordinaten oben/ unten des Elementknotenpaares i bei Volumenschalenelementen
ξ, η, ζ	natürliche Elementkoordinaten
ξ_α	Innere Variablen für Plastizität
Ψ	Gleichungssystem für Plastizität

Einleitung

Motivation

Forschungs- und Entwicklungsarbeit auf dem Gebiet der Finiten-Elemente-Methoden für strukturmechanische Probleme lässt sich in drei wesentliche Schritte gliedern. Zunächst wird ein mechanisches Modell oder ein Diskretisierungsmodell – also z. B. eine Material- oder Elementformulierung – theoretisch entwickelt, indem die zugehörigen Beziehungen aus kontinuumsmechanischen Grundlagen hergeleitet und für eine Programmierung aufbereitet werden. In einem zweiten Schritt erfolgt üblicherweise eine algorithmische Umsetzung und eine programmtechnische Implementierung, meist in eine bestehende Software in einer gewählten Programmiersprache. Dies ist notwendig, um die entwickelte Methode im dritten Schritt an Beispielen zu validieren und deren Funktionalität zu testen. Üblicherweise lassen sich die drei genannten Schritte innerhalb eines Forschungsprozesses nicht klar voneinander trennen, da Mängel bei der Modellbildung teilweise erst nach der Programmierung, bei deren Anwendung an entsprechenden Beispieluntersuchungen, deutlich werden. Kleinere Änderungen an der theoretischen Formulierung können dann wieder umfangreiche Programmmodifikationen mit sich bringen.

Der zweite Schritt erfordert zunächst eine sehr kleinteilige algorithmische Aufbereitung der hergeleiteten Formeln, um deren Umsetzung in Programmcode zu ermöglichen. Hierbei müssen beispielsweise Ableitungen vorab ausgewertet, für eine Programmierung aufbereitet und anschließend in die jeweilige Programmiersprache umgesetzt werden. Neben der händischen Programmierung der hergeleiteten Zusammenhänge sind auch Umsetzungen mit mathematischer Software wie z. B. MATHEMATICA oder MAPLE denkbar, mit deren Hilfe erforderliche Ableitungen symbolisch berechnet werden können. Dies führt jedoch oft zu langen Termen und somit zu umfangreichem und somit unhandlichem und ineffizientem Programmcode. Demgegenüber ist die Aufbereitung für eine manuelle Programmierung in der Regel sehr zeitaufwändig und insbesondere fehleranfällig. Hierbei sind nicht nur algorithmische Fehler gemeint, die bei der Entwicklung eines Programmablaufs entstehen können, sondern auch einfache Programmierfehler im Sinne von ‘Tippfehlern’, die bei der Eingabe des Programmcodes auftreten und beim sogenannten *debugging* zeitaufwändig gesucht und korrigiert werden müssen. Für den dritten Entwicklungsschritt – das Testen der Formulierung sowie der Effizienz des Programmcodes an entsprechenden Beispielen – ist eine möglichst optimale Umsetzung der hergeleiteten Methode wünschenswert. Häufig wird bei der händischen Programmierung jedoch nicht ausdrücklich auf Effizienz geachtet und so kann selbst die Simulation sehr einfacher Testbeispiele, die z. B. bei der Verwendung kommerzieller Programme nur wenig Zeit in Anspruch nehmen, sehr lange dauern.

Ziel der vorliegenden Arbeit ist die Entwicklung eines Programmierkonzeptes zur effizienten Umsetzung von Finite-Elemente-Formulierungen, insbesondere im Kontext expliziter Zeitintegration. Wegen üblicherweise sehr kleinen Zeitschritten und effizienten Vektoroperationen auf Systemebene fällt hier ein vergleichsweise hoher Anteil der Gesamtsimulati-

onszeit auf die Elementroutine ab. Daher ist gerade hier eine hinsichtlich der Operationen optimierte Umsetzung besonders wünschenswert. Die Programmierung wird durch die Verwendung des auf MATHEMATICA basierenden Programmpaketes ACEGEN [71, 73] unterstützt, mit dessen Hilfe Programmierfehler vermieden und die Umsetzung von Änderungen und Korrekturen an der Formulierung erheblich vereinfacht werden sollen. Die symbolischen Eigenschaften der Programmierumgebung haben eine teilweise Verschmelzung der ersten beiden Arbeitsschritte – Modellbildung und Implementierung – zur Folge und erlauben so eine frühe Durchführung numerischer Testbeispiele. Die von ACEGEN automatisch durchgeführte Optimierung der Programmschritte führt außerdem ohne zusätzliche Überlegungen zu sehr effizienten ProgrammROUTINEN. Das Programmierkonzept wurde im Rahmen dieser Arbeit entwickelt und auf degenerierte Schalen- und Volumenschalenelemente mit linearer und quadratischer Interpolation von Geometrie und Verschiebungen in Schalenebene angewendet. Neben bestehender – im Wesentlichen für implizite Statik und Dynamik vorgestellter – Strategien zur Beseitigung ungewollter Versteifungseffekte, konnten neue Methoden entwickelt werden, deren Umsetzung durch die symbolische Programmierung erheblich erleichtert wurde. Zusätzlich zu den Strukturelementen konnte die Implementierung einer Kontaktformulierung ebenfalls mit Hilfe von ACEGEN durchgeführt werden.

Stand der Forschung

Das symbolische Programmierpaket ACEGEN [71, 73] wird als sogenanntes Plug-In zum Computeralgebraprogramm Mathematica [127] angeboten. Es wurde von J. KORELC in seiner Dissertation [65] erstmals vorgestellt und wird zur Zeit von seiner Arbeitsgruppe weiterentwickelt [66–69, 72, 74, 77]. Die genannten Veröffentlichungen behandeln dabei verschiedene Bereiche der Strukturmechanik und verwenden neben der Schnittstelle zu Finite-Elemente-Programmen wie ABAQUS [118] im Wesentlichen das ebenfalls von der Gruppe KORELC entwickelte und auf MATHEMATICA basierende FE-Programm ACEFEM [70]. Wegen der Fähigkeit der symbolischen Programmierumgebung, Ableitungen komplexer Funktionen aufzustellen und deren Ergebnisse in sehr effizienten Programmcode zu überführen, bieten sich unterschiedlichste Anwendungsgebiete im Bereich der computergestützten Mechanik an, da dort behandelte Probleme üblicherweise durch Differentialgleichungen beschrieben sind.

Das aus dem ursprünglich für Molekulardynamik vorgestellten Verlet-Algorithmus [124] entwickelte zentrale Differenzenverfahren, ist heute *das* explizite Zeitintegrationsverfahren im Bereich der Strukturmechanik, insbesondere in Kombination mit diagonalen Massenmatrizen. Auf Grund effizienter globaler Gleichungen, die ausschließlich Vektoroperationen beinhalten, werden im Vergleich zu einem impliziten Lösungsverfahren verhältnismäßig wenige Rechenschritte pro Zeitschritt durchgeführt. Im Zusammenhang mit der Begrenzung der Zeitschrittweite eignet sich die Methode daher insbesondere für kurzzeit-dynamische, durch starke Nichtlinearitäten charakterisierte Prozesse. In der Praxis wird das zentrale Differenzenverfahren in kommerziellen Programmen wie LS-DYNA [83], PAMCRASH [42], ABAQUS/EXPLICIT [118], etc. im Wesentlichen zur Abbildung von Impakt- und Crashvorgängen etwa im Automobilbau aber auch für Umformsimulationen eingesetzt.

Bei schalenartigen Strukturen, wie sie sowohl im Bauwesen als auch im Maschinenbau in vielen Anwendungen vorkommen, ist die geometrische Ausdehnung eines im Allgemeinen gekrümmten Bauteils in zwei Richtungen erheblich größer als die in der dritten; man spricht

von der Schalenebene und der Dickenrichtung. Über die Geschichte der Schalentheorie in der Mechanik und deren Anwendung für die Methode der finiten Elemente existieren im Schrifttum zahlreiche Fach- und Lehrbücher. Ein umfassender und sehr guter Überblick mit vielen Referenzen liegt beispielsweise in [25] vor. Zur Beschreibung solcher Strukturen eignet sich das bekannte *Degenerationskonzept* [1] und die daraus resultierenden degenerierten Schalenelemente, bei denen die Schalendicke als Parameter, nicht wie bei Kontinuumsformulierungen als geometrische Größe, verwendet wird. Umsetzungen verschiedener Elementformulierungen, insbesondere solcher mit linearer Geometrie- und Verschiebungsinterpolation, sind heute praktisch in allen kommerziellen FE-Paketen [10, 42, 83, 118] verfügbar und sind teilweise in den zugehörigen Handbüchern dokumentiert. Elemente mit quadratischer und höherer Ansatzordnung finden sich teilweise in FE-Programmen sowie in wissenschaftlichen Veröffentlichungen im Zusammenhang mit impliziten Lösungsverfahren; kommerzielle Anwendungen im Kontext expliziter Zeitintegration existieren kaum oder nur mit erheblichen Einschränkungen. Eine wesentlich jüngere – und in der kommerziellen Welt noch weitaus weniger vertretene – Elementfamilie stellen die Volumenschalen, oder auch ‘*Solid-Shell*’-Elemente [55, 56, 95, 102, 120] dar. Hier werden die Schreibweisen des Degenerationskonzeptes übernommen, um Schalenebene und Dickenrichtung zu unterscheiden, wobei die Schalendicke jedoch als geometrische Größe erhalten bleibt. Bei den Elementen handelt es sich also prinzipiell um Volumenelemente, die über Verschiebungsfreiheitsgrade formuliert sind, bei denen jedoch die beiden Vorzugsrichtungen der Schalenebene getrennt von der Dickenrichtung behandelt werden können. Dies spielt insbesondere für die Beseitigung ungewollter Versteifungseffekte – den sogenannten ‘*Locking*’-Phänomenen – eine wichtige Rolle spielt.

Die wesentlichen Vorteile der Solid-Shell-Elemente im Vergleich zu degenerierten Schalen lassen sich durch das ausschließliche Auftreten von Verschiebungsfreiheitsgraden begründen. Das Fehlen von Rotationsfreiheitsgraden erlaubt beispielsweise die Anwendung kontinuumsmechanisch formulierter Materialmodelle ohne weitere Annahmen. Effekte in Dickenrichtung, die beispielsweise im Gebiet der Metallumformung von Interesse sind, lassen sich ohne weiteres darstellen, wozu hingegen im Bereich der degenerierten Schalen erweiterte Modelle erforderlich sind, etwa im Rahmen einer 7-Parameter Formulierung [30]. Geschichtete Materialien lassen sich durch Anordnung mehrerer Elemente über die Dicke ebenso leicht umsetzen wie die Kopplung schalenartiger und mit reinen Volumenelementen diskretisierten Strukturteilen. Problematisch im Zusammenhang mit expliziter Zeitintegration ist der Einfluss der Dicke auf die Zeitschrittweite, der bei degenerierten Formulierungen nicht auftritt. Insbesondere bei sehr dünnen Strukturen ergeben sich sehr hohe Eigenfrequenzen, die zu kleinen Zeitschrittweiten und somit einem erheblichen Anstieg der Simulationszeiten führen können.

Der wichtige Forschungsbereich der ‘*Elementtechnologie*’ beschäftigt sich mit der Vermeidung ungewollter Versteifungseffekte, welche im Wesentlichen bei finiten Elementen niederer Ansatzordnungen auftreten und sich je nach deren Ursache in geometrische und materielle Effekte unterteilen lassen (vgl. z. B. [19, 78]). Viele für degenerierte Schalen entwickelte Strategien zur Behandlung solcher Versteifungseffekte können auch auf Volumenschalen angewendet werden. Zur Beseitigung von Querschub- und Membranversteifungen, die etwa bei Strukturproblemen unter Biegung auftreten, wird häufig die für lineare [13] und für quadratische Schalenelemente [29] vorgestellte ‘*Assumed Natural Strains*’ Methode (ANS) angewendet. Eine Strategie zur Behandlung geometrischer und materieller Versteifungseffekte,

stellt die ‘*Enhanced Assumed Strains*’ Methode (EAS) dar [114, 115, 117], deren Entwicklung durch die ‘*Methode der inkompatiblen Moden*’ [123, 126] motiviert wurde. In den Veröffentlichungen der letzten 20 Jahre zu diesen Themen – eine vollständige Liste der wichtigen Arbeiten anzugeben ist nahezu unmöglich – finden sich sowohl reine EAS-Elemente (siehe u. a. [2, 3, 5, 8, 44, 76]), als auch Formulierungen mit einer Kombination aus ANS und EAS (siehe u. a. [23, 52, 55, 61, 62, 120, 121]). In Kombination mit reduzierter Integration und Hourglass-Stabilisierung [32, 105, 106] werden beide Methoden in neueren Publikationen eingesetzt; siehe hierzu z. B. [4, 6, 31, 33, 110, 111]. Weitere Strategien zur Behandlung von Versteifungen wie beispielsweise die selektive Unterintegration bei volumetrisch-deviatorisch entkoppelten Materialformulierungen [39, 52] oder die kompatible quadratische Verschiebungsinterpolation in Dickenrichtung [48, 102] seien hier außerdem erwähnt.

Die oben genannten Publikationen beschäftigen sich fast ausschließlich mit der Anwendung der Elementformulierungen für Statik und Dynamik mit impliziten Lösungsverfahren. Die bereits angesprochenen Eigenschaften eines expliziten Zeitintegrationsverfahrens – sehr effiziente globale Operationen bei limitierter Zeitschrittweite – stellen jedoch besondere Anforderungen an die Elementtechnologie, weshalb sich vollständig reduziert integrierte Volumenelemente [14, 104] in Kombination mit Stabilisierungsverfahren gegen künstliche Elementkinematiken und sehr effiziente Schalenformulierungen (z. B. [16]) in den kommerziellen Anwendungen durchgesetzt haben. Die Anwendbarkeit der oben genannten Methoden für Schalen- und Volumenschalenelemente sowie die Konkurrenzfähigkeit höherer als linearer Interpolationsordnungen hängt also von verschiedenen Faktoren ab. Obgleich zu erwarten ist, dass eine Beseitigung von Versteifungseffekten in gleicher Weise wie bei impliziten Lösungsverfahren möglich ist, werden sich die Methoden doch unterschiedlich auf die Effizienz der jeweiligen Elementformulierung auswirken. Bestimmte Strategien, die sich für implizite Anwendungen bewährt haben, kommen auf Grund ihres hohen numerischen Aufwandes für explizite Verfahren nicht in Frage. Andererseits müssen die Eigenschaften und das Verhalten der Elementformulierungen bei kurzzeit-dynamischen Prozessen untersucht werden, da Simulationen solcher Anwendungen im Wesentlichen mit expliziten FE-Programmen erfolgen.

Gliederung der Arbeit

In *Kapitel 1 – Kontinuumsmechanische Grundlagen* sind die für den weiteren Verlauf erforderlichen, grundlegenden Zusammenhänge der Kontinuumsmechanik zusammengestellt. Aus den Annahmen zu Kinematik, Stoffgesetz und Randbedingungen wird die starke und schließlich die schwache Form des Gleichgewichts hergeleitet, welche am Ende des Kapitels für die Lösung mit der Methode der finiten Elemente in eine räumlich diskretisierte Form überführt wird. Diese Einführung dient ausschließlich der Festlegung der verwendeten Notation und versteht sich keineswegs als eine vollständige Einführung in die Kontinuumsmechanik.

Kapitel 2 – Lösung zeitabhängiger Probleme beschäftigt sich mit der zeitlichen Diskretisierung der zuvor aufgestellten semi-diskreten Bewegungsgleichung. Im Mittelpunkt steht das gewählte explizite zentrale Differenzenverfahren als Lösungsalgorithmus. Eine Abgrenzung zu impliziten Methoden erfolgt am Beispiel des bekannten NEWMARK-Verfahrens. Verschiedene Aspekte zu Effizienz, numerischer Stabilität und Genauigkeit des zentralen

Differenzenverfahrens werden diskutiert und die durchgeführte Implementierung in ein Finite-Elemente-Programm beschrieben.

In *Kapitel 3 – Finite Elemente für Schalenprobleme* werden zunächst die wesentlichen Zusammenhänge für die verwendeten Elementfamilien ‘degenerierte Schalen’ und ‘Volumenschalen’ zusammengefasst. Es folgt die Aufstellung einer diagonalisierten Massenmatrix, welche für die Anwendung der Elemente innerhalb der expliziten Zeitintegration erforderlich ist. Insbesondere bei Elementen mit quadratischer Ansatzordnung in der Schalenebene und von der Würfelform abweichender Geometrien, können die verschiedenen vorgestellten Diagonalisierungsstrategien zu unterschiedlichen Massenverteilungen führen. Dies wird zusätzlich an einem Beispiel im Anhang dieser Arbeit diskutiert. Im Anschluss wird eine kurze Klassifikation der verschiedenen künstlichen Versteifungseffekte angegeben, dann werden die Strategien zu deren Behandlung vorgestellt. Neben den verbreiteten Methoden ANS und EAS, kommt für die Solid-Shell-Elemente die *Methode der inkompatiblen Moden* zum Einsatz, bei der auf Elementebene inkompatible Verschiebungsfreiheitsgrade eingeführt werden. Als Alternative zur ANS-Methode wird – zumindest für ebene Elemente und kleine Deformationen – eine auf einer modifizierten Verschiebungsinterpolation basierende Strategie vorgestellt, die als *Methode der erweiterten Verschiebungen* bezeichnet ist. Am Ende des Kapitels wird eine Möglichkeit zur Beschreibung von Kontakt ereignissen zwischen deformierbaren Strukturteilen und analytisch beschriebenen starren Flächen eingeführt, welche ebenfalls für die Arbeit umgesetzt wurde.

Das entwickelte und verwendete Programmierkonzept wird in *Kapitel 4 – Symbolische Programmierung* eingeführt. Nach einer Definition des Begriffes ‘Effizienz’ im Zusammenhang mit Forschung und Entwicklung auf dem Gebiet der computergestützten Mechanik werden das verwendete Programmierpaket ACEGEN vorgestellt sowie einige Programmbefehle der zugrundeliegenden Meta-Sprache eingeführt. Es folgt die Beschreibung der Implementierungsschritte für die verschiedenen Elementformulierungen. Das verwendete Programmierkonzept lässt sich prinzipiell auch auf andere Elementformen anwenden, was am Beispiel der zuvor eingeführten Kontaktformulierungen deutlich gemacht wird. Eine detaillierte Zusammenstellung der einzelnen Schritte ist nochmal an einem Prinzipbeispiel im Anhang angegeben.

Abschließend werden die implementierten Elementformulierungen in *Kapitel 5 – Numerische Beispiele* getestet. Neben der Beseitigung verschiedener Versteifungseffekte steht hier vor allem die Bewertung des Programmierkonzeptes selbst sowie der verschiedenen Strategien hinsichtlich Effizienz innerhalb des expliziten Zeitintegrationsalgorithmus im Vordergrund. An einem Prinzipbeispiel lassen sich verschiedene Effekte entkoppelt und somit anschaulich darstellen. Es folgen Simulationen gekrümmter Schalenstrukturen, teilweise mit großen elastischen und elasto-plastischen Deformationen, die kurzzeit-dynamisch beansprucht sind, was dem wesentlichen Einsatzgebiet für explizite Zeitintegration entspricht.

Anmerkungen zu besonderen Schreibweisen

Um die Lesbarkeit zu erleichtern, werden verschiedene Vereinbarungen zur Schreibweise getroffen. Diese sind:

- Eigennamen von Personen oder Computerprogrammen werden in Kapitälchen gesetzt (z. B. ZIENKIEWICZ, MATHEMATICA).

- Da der Buchstabe ‘ß’ in Kapitälchen nicht existiert, wird dieser bei Eigennamen durch SS ersetzt (z. B. GAUSS).
- MATHEMATICA-Befehle werden üblicherweise mit einem ‘;’ abgeschlossen, weshalb bei abgesetzten Formeln, die MATHEMATICA-Eingaben beschreiben, auf eine korrekte und fortlaufende Zeichensetzung verzichtet wird.
- MATHEMATICA-Befehle werden in **Schreibmaschinenschrift** gesetzt, zur besseren Abgrenzung sind ACEGEN-Kommandos zusätzlich **blau** dargestellt.

1. Kontinuumsmechanische Grundlagen

In diesem Kapitel werden die für die Arbeit benötigten Grundgleichungen der Kontinuumsmechanik angegeben. Aus den Zusammenhängen für Kinematik, Gleichgewicht und Stoffgesetz wird die starke Form des Gleichgewichtes formuliert und auf dessen Basis eine schwache Form abgeleitet, die für die Verwendung innerhalb der Finite-Elemente-Methode erforderlich ist. Mit der räumlichen Diskretisierung wird schließlich die Grundlage für die Anwendung von Zeitintegrationsverfahren im nächsten Kapitel geschaffen. Da hier nur für die Arbeit unmittelbar relevante Grundlagen angegeben werden können, wird für detaillierte Ausführungen auf das umfassende Schrifttum verwiesen. Exemplarisch seien an dieser Stelle die Bücher zur Kontinuumsmechanik von BONET & WOOD [27], HAUPT [53], HOLZAPFEL [57] und PARISCH [103] genannt. Auch Fachbücher über die Finite-Elemente-Methode bieten ausführliche Einleitungen zur Kontinuumsmechanik, sowie umfassende Ausführungen zur Diskretisierung. Hier seien u. a. COOK et. al. [35], WRIGGERS [130] und ZIENKIEWICZ et. al. [134–136] angegeben, wobei auch diese Liste keinen Anspruch auf Vollständigkeit besitzt. Detaillierte Ausführungen zu den Materialformulierungen – insbesondere zur Implementierung der Stoffgesetze mit Plastizität – können SIMO & HUGHES [116] und DE SOUSA NETO ET. AL. [37] entnommen werden.

1.1. Kinematik und Verzerrungen

Eine Konfiguration $\boldsymbol{\varphi}$ eines Körpers \mathcal{B} ist als eine eindeutige Abbildung $\boldsymbol{\varphi} : \mathcal{B}_0 \rightarrow \mathbb{R}^3$ definiert, bei der jeder Punkt \mathbf{X} einer Referenzkonfiguration $\boldsymbol{\varphi}_0$ auf einen Punkt \mathbf{x} einer Momentankonfiguration abgebildet wird. Wie in Abbildung 1.1 skizziert, ergibt sich eine Bewegung als Abfolge von Konfigurationen über die Zeit t

$$\mathbf{x} = \boldsymbol{\varphi}(\mathbf{X}, t) = \mathbf{X} + \mathbf{u}(\mathbf{X}, t) \quad (1.1)$$

Die Abbildung nutzt die Koordinaten der Referenzkonfiguration und wird als LAGRANGESCHE Darstellung bezeichnet. Ihre Umkehrung liefert die sogenannte EULERSCHE Darstellung $\mathbf{X} = \boldsymbol{\varphi}^{-1}(\mathbf{x}, t)$, die jedoch im Verlauf der Arbeit nicht weiter verwendet wird. Oft wird als Referenzkonfiguration die Ausgangskonfiguration ($t = 0$) verwendet, jedoch ist auch der Bezug auf eine beliebige Referenzkonfiguration möglich, die sich zu keiner Zeit einstellen muss. Die Ortsvektoren \mathbf{X} und \mathbf{x} beziehen sich auf ein kartesisches Bezugssystem $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \}$.

Die Abbildung eines unverformten Längenelementes $d\mathbf{X}$ auf die Momentankonfiguration

$$d\mathbf{x} = \mathbf{F}d\mathbf{X} \quad (1.2)$$

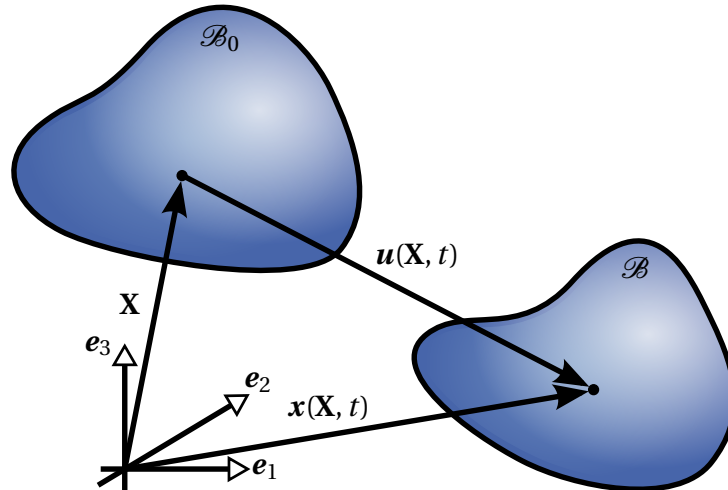


Abbildung 1.1.: Kinematik eines Körpers

und somit die Beschreibung von Deformationen erfolgt über die Einführung des Deformationsgradienten

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \frac{\partial \mathbf{X}}{\partial \mathbf{X}} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} = \mathbf{1} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} = \mathbf{1} + \text{grad } \mathbf{u}(\mathbf{X}, t) = \text{grad } \boldsymbol{\varphi}(\mathbf{X}, t). \quad (1.3)$$

Die Selbstdurchdringung eines Körpers ist durch die Bedingung

$$\det \mathbf{F} > 0 \quad (1.4)$$

ausgeschlossen, wodurch auch die Umkehrbarkeit der Abbildung gewährleistet ist.

Ein Vergleich der Quadrate der Linienelemente in beiden Konfigurationen $d\mathbf{x} d\mathbf{x} - d\mathbf{X} d\mathbf{X}$ führt durch Bezug auf die Referenzkonfiguration zum materiellen GREEN-LAGRANGE-Verzerrungstensor

$$\mathbf{E} = \frac{1}{2} \left(\frac{d\mathbf{x} d\mathbf{x} - d\mathbf{X} d\mathbf{X}}{d\mathbf{X} d\mathbf{X}} \right) = \frac{1}{2} \left(\frac{d\mathbf{X} \mathbf{C} d\mathbf{X} - d\mathbf{X} d\mathbf{X}}{d\mathbf{X} d\mathbf{X}} \right) = \frac{1}{2} (\mathbf{C} - \mathbf{1}). \quad (1.5)$$

Mit dem rechten CAUCHY-GREEN-Tensor

$$\mathbf{C} = \mathbf{F}^T \cdot \mathbf{F} \quad (1.6)$$

führt dies auf die Darstellung

$$\mathbf{E} = \frac{1}{2} (\mathbf{F}^T \cdot \mathbf{F} - \mathbf{1}). \quad (1.7)$$

Wird in Gleichung (1.5) als Bezug die Momentankonfiguration gewählt – im Nenner steht dann also $d\mathbf{x} d\mathbf{x}$ – so ergibt sich der räumliche ALMANSI-Verzerrungstensor

$$\mathbf{e} = \frac{1}{2} (\mathbf{1} - \mathbf{b}^{-1}), \quad (1.8)$$

mit dem linken CAUCHY-GREEN-Tensor

$$\mathbf{b} = \mathbf{F} \cdot \mathbf{F}^T. \quad (1.9)$$

Ein Zusammenhang zwischen den beiden Verzerrungstensoren lässt sich über *push forward* und *pull back*, also ein *Vorschieben* auf die Momentankonfiguration, bzw. ein *Zurückziehen* auf die Referenzkonfiguration herstellen. Es ergibt sich der Zusammenhang

$$\mathbf{e} = \mathbf{F} \cdot \mathbf{E} \cdot \mathbf{F}^T \quad (1.10)$$

$$\mathbf{E} = \mathbf{F}^{-1} \cdot \mathbf{e} \cdot \mathbf{F}^{-T}. \quad (1.11)$$

Durch Einsetzen der Beziehung (1.3) in Gleichung (1.7) ergibt sich eine Darstellung, die die Aufspaltung des Verzerrungstensors in einen linearen und einen nichtlinearen Anteil verdeutlicht

$$\mathbf{E} = \frac{1}{2} \left(\left(\mathbf{1} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} \right)^T \cdot \left(\mathbf{1} + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} \right) - \mathbf{1} \right) = \underbrace{\frac{1}{2} \left(\left(\frac{\partial \mathbf{u}}{\partial \mathbf{X}} \right)^T + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} \right)}_{\text{linear}} + \underbrace{\frac{1}{2} \left(\left(\frac{\partial \mathbf{u}}{\partial \mathbf{X}} \right)^T \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{X}} \right)}_{\text{nichtlinear}}. \quad (1.12)$$

1.2. Bilanzgleichungen

1.2.1. Massenbilanz

Der *Massenerhaltungssatz* besagt, die Masse eines Körpers, also das Integral seiner Rohdichte über sein Volumen, sei unabhängig von seiner Konfiguration:

$$m = \int_{\mathcal{B}} \rho \, dv = \int_{\mathcal{B}_0} \rho_0 \, dV. \quad (1.13)$$

Die Transformation eines Volumenelementes mittels des Deformationsgradienten

$$dv = \det \mathbf{F} \, dV \quad (1.14)$$

führt somit auf den Zusammenhang zwischen der Rohdichte in der Referenz- und der Momentankonfiguration:

$$\rho_0 = \det \mathbf{F} \, \rho. \quad (1.15)$$

1.2.2. Impulsbilanz

Der *Impulserhaltungssatz* fordert die Gleichheit der zeitlichen Ableitung des Impulses \mathbf{L} mit der Summe aller am Kontinuum angreifenden Kräfte \mathbf{f} . Der Impuls in der Momentankonfiguration ist definiert als

$$\mathbf{L} = \int_{\mathcal{B}} \rho \, \dot{\mathbf{u}} \, dv, \quad (1.16)$$

mit der Geschwindigkeit $\dot{\mathbf{u}} = \partial \mathbf{u} / \partial t$ als partielle Zeitableitung des Verschiebungsvektors. Die eingepägten Kräfte berechnen sich mit den eingepägten Volumenkräften \mathbf{b} und den

auf die Flächeneinheit da bezogenen Spannungen \mathbf{t} , welche auf den Rand des Gebietes $\partial\mathcal{B}$ wirken zu

$$\mathbf{f} = \int_{\mathcal{B}} \rho \mathbf{b} \, dv + \int_{\partial\mathcal{B}} \mathbf{t} \, da. \quad (1.17)$$

Das CAUCHY-Theorem $\mathbf{t} = \boldsymbol{\sigma} \cdot \mathbf{n}$ verknüpft den Spannungsvektor mit dem CAUCHY-Spannungstensor $\boldsymbol{\sigma}$ über den Oberflächennormalenvektor \mathbf{n} und führt mit Hilfe des GAUSSSchen Integralsatzes auf ein Volumenintegral

$$\int_{\partial\mathcal{B}} \mathbf{t} \, da = \int_{\partial\mathcal{B}} \boldsymbol{\sigma} \cdot \mathbf{n} \, da = \int_{\mathcal{B}} \operatorname{div} \boldsymbol{\sigma} \, dv. \quad (1.18)$$

Die Zeitableitung von Gleichung (1.16) erfolgt über eine Rücktransformation in die Referenzkonfiguration mit Gleichung (1.15)

$$\dot{\mathbf{L}} = \frac{\partial}{\partial t} \int_{\mathcal{B}} \rho \dot{\mathbf{u}} \, dv = \frac{\partial}{\partial t} \int_{\mathcal{B}_0} \rho_0 \dot{\mathbf{u}} \, dV = \int_{\mathcal{B}_0} \rho_0 \ddot{\mathbf{u}} \, dV. \quad (1.19)$$

Die Impulsbilanz kann jetzt als ein Volumenintegral geschrieben werden:

$$\int_{\mathcal{B}_0} (\operatorname{div} \boldsymbol{\sigma} + \rho \mathbf{b} - \rho \ddot{\mathbf{u}}) \, dV = 0 \quad (1.20)$$

Diese Bedingung muss für jede beliebige Konfiguration erfüllt sein und beschreibt somit als lokale Form

$$\operatorname{div} \boldsymbol{\sigma} + \rho \mathbf{b} - \rho \ddot{\mathbf{u}} = 0 \quad \text{auf } \mathcal{B} \quad (1.21)$$

das Gleichgewicht in der Momentankonfiguration, das auch als 1. CAUCHYSches Bewegungsgesetz bezeichnet wird. Transformiert auf die Referenzkonfiguration führt dies auf

$$\operatorname{div} \mathbf{P} + \rho_0 \mathbf{b} - \rho_0 \ddot{\mathbf{u}} = 0 \quad \text{auf } \mathcal{B}_0 \quad (1.22)$$

mit dem unsymmetrischen 1. PIOLA-KIRCHHOFF-Spannungstensor

$$\mathbf{P} = \det \mathbf{F} \boldsymbol{\sigma} \cdot \mathbf{F}^T. \quad (1.23)$$

1.2.3. Drallbilanz

Der *Drallsatz* fordert die Gleichheit der zeitlichen Ableitung des Drehimpulses \mathbf{J} bezüglich eines Punktes P mit dem Ortsvektor \mathbf{X}_P

$$\mathbf{J} = \int_{\mathcal{B}} (\mathbf{x} - \mathbf{X}_P) \times \rho \dot{\mathbf{x}} \, dv \quad (1.24)$$

und der Summe der Momente um Punkt P in Folge aller äußeren Kräfte. Es folgt die Bedingung

$$\dot{\mathbf{J}} = \frac{\partial}{\partial t} \int_{\mathcal{B}} (\mathbf{x} - \mathbf{X}_P) \times \rho \dot{\mathbf{x}} \, dv = \int_{\mathcal{B}} (\mathbf{x} - \mathbf{X}_P) \times \rho \mathbf{b} \, dv + \int_{\partial\mathcal{B}} (\mathbf{x} - \mathbf{X}_P) \times \mathbf{t} \, da, \quad (1.25)$$

die nach einigen Rechenschritten auf die Symmetrie des CAUCHY-Spannungstensor $\boldsymbol{\sigma}$ führt.

1.3. Spannungsmaße und Stoffgesetze

Ein zum im Abschnitt 1.1 eingeführten GREEN-LAGRANGE-Verzerrungstensor konjugiertes Spannungsmaß ergibt sich durch vollständige Rücktransformation des bereits im vorigen Abschnitt verwendeten CAUCHY-Spannungstensors $\boldsymbol{\sigma}$ auf die Referenzkonfiguration. Es folgt der materielle 2. PIOLA-KIRCHHOFF-Spannungstensor

$$\mathbf{S} = \mathbf{F}^{-1} \cdot \mathbf{P} = \det \mathbf{F} \mathbf{F}^{-1} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-T}. \quad (1.26)$$

Analog lässt sich durch vollständige Transformation auf die Momantankonfiguration der räumliche KIRCHHOFF Spannungstensor bestimmen

$$\boldsymbol{\tau} = \mathbf{P} \cdot \mathbf{F}^T = \det \mathbf{F} \boldsymbol{\sigma}. \quad (1.27)$$

Die räumlichen und materiellen Spannungstensoren lassen sich wie bereits in Abschnitt 1.1 für die Verzerrungen beschrieben über *push forward* und *pull back* Operationen ineinander überführen. Es gelten analog zu Gleichung (1.10) die Zusammenhänge

$$\boldsymbol{\tau} = \mathbf{F} \cdot \mathbf{S} \cdot \mathbf{F}^T \quad (1.28)$$

$$\mathbf{S} = \mathbf{F}^{-1} \cdot \boldsymbol{\tau} \cdot \mathbf{F}^{-T}. \quad (1.29)$$

Im Gegensatz zum unsymmetrischen 1. PIOLA-KIRCHHOFF-Spannungstensor \mathbf{P} gilt

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^T, \quad \mathbf{S} = \mathbf{S}^T \quad \text{und} \quad \boldsymbol{\tau} = \boldsymbol{\tau}^T. \quad (1.30)$$

Zur Bestimmung der Spannungen aus den kinematischen Beziehungen ist die Formulierung eines Stoffgesetzes erforderlich. In den folgenden Abschnitten werden kurz die für die vorliegende Arbeit umgesetzten Materialgesetze beschrieben.

1.3.1. Hyperelastizität

Im Fall hyperelastischen Materialverhaltens – die Annahmen sind für viele Schaumstoffe und Gummimaterialien zulässig – lässt sich der Zusammenhang zwischen Spannungen und Verzerrungen aus einem Potential $\mathcal{W}(\mathbf{E})$ als Funktion einer kinematischen Größe, etwa den in Abschnitt 1.1 eingeführten GREEN-LAGRANGE-Verzerrungen ableiten. Dieses beschreibt die im Körper gespeicherte Energie aus der Formänderung und wird somit auch als *Formänderungsenergiefunktion* oder *Verzerrungsenergiefunktion* bezeichnet. Es existiert kein Einfluss der Materialgeschichte, die Spannungen lassen sich also immer direkt aus dem aktuellen Deformationszustand berechnen. Je nachdem in welcher kinematischen Größe die Formänderungsenergie gegeben ist, ergibt sich ein differentieller Zusammenhang für die Spannungen:

$$\mathbf{S} = \frac{\partial \mathcal{W}(\mathbf{E})}{\partial \mathbf{E}} = 2 \frac{\partial \mathcal{W}(\mathbf{C})}{\partial \mathbf{C}}, \quad \text{bzw.} \quad \boldsymbol{\tau} = \frac{\partial \mathcal{W}(\mathbf{e})}{\partial \mathbf{e}} = 2 \frac{\partial \mathcal{W}(\mathbf{b})}{\partial \mathbf{b}} \cdot \mathbf{b}. \quad (1.31)$$

Für die Arbeit wurden exemplarisch zwei verschiedene hyperelastische Materialgesetze für die verschiedenen Elementformulierungen implementiert.

Der lineare Zusammenhang zwischen GREEN-LAGRANGE-Verzerrungen und 2. PIOLA-KIRCHHOFF-Spannungen führt auf das ST. VENANT-KIRCHHOFF-Materialgesetz, welches durch die quadratisch in den Verzerrungen vorliegende Formänderungsenergiefunktion

$$\mathcal{W}(\mathbf{E}) = \frac{1}{2} \lambda (\text{tr } \mathbf{E})^2 + \mu \mathbf{E} : \mathbf{E} \quad (1.32)$$

mit den LAMÉ-Konstanten

$$\lambda = \frac{E \nu}{(1 + \nu)(1 - 2\nu)}, \quad \text{und} \quad \mu = \frac{E}{2(1 + \nu)} \quad (1.33)$$

beschrieben wird. Es stellt die Erweiterung des HOOKESchen Gesetzes auf die endliche Elastizitätstheorie dar und kann somit große Verschiebungen nur bei kleinen Verzerrungen sinnvoll abbilden. Es kann insbesondere gezeigt werden, dass eine Kompression eines Körpers im Grenzfall $\nu \rightarrow 0$ auf eine Spannung $\boldsymbol{\sigma} = \mathbf{0}$ führt, weshalb dieses Materialgesetz für Großdeformationssimulationen unbrauchbar ist.

Ein besonders zur Simulation gummiartiger Materialien mit großen elastischen Verzerrungen weit verbreitetes Materialgesetz stellt das NEO-HOOKE-Modell dar, welches ebenfalls für die implementierten Elementroutinen umgesetzt wurde. Die Formänderungsenergiefunktion kann ebenfalls mit den LAMÉ-Konstanten aus Gleichung (1.33) in Abhängigkeit der Invarianten des rechten bzw. des linken CAUCHY-GREEN-Tensors geschrieben werden:

$$\begin{aligned} \mathcal{W}(\mathbf{C}) &= \frac{\mu}{2} (\text{I}_{\mathbf{C}} - 3) - \mu \ln \sqrt{\text{III}_{\mathbf{C}}} + \frac{\lambda}{2} \left(\ln \sqrt{\text{III}_{\mathbf{C}}} \right)^2, \quad \text{bzw.} \\ \mathcal{W}(\mathbf{b}) &= \frac{\mu}{2} (\text{I}_{\mathbf{b}} - 3) - \mu \ln \sqrt{\text{III}_{\mathbf{b}}} + \frac{\lambda}{2} \left(\ln \sqrt{\text{III}_{\mathbf{b}}} \right)^2, \end{aligned} \quad (1.34)$$

Das Materialgesetz stellt eine nichtlineare Beziehung zwischen \mathbf{C} und \mathbf{S} her und ist geeignet um große Deformationen bei großen Verschiebungen abzubilden.

In der vorliegenden Arbeit wurde überwiegend mit den beiden beschriebenen hyperelastischen Materialgesetzen gearbeitet. Die verwendete Art der symbolischen Programmierung, wie sie im weiteren Verlauf in Kapitel 4, bzw. in Anhang B beschrieben wird, erlaubt jedoch die Umsetzung weiterer Formulierungen ohne großen Aufwand durch einfaches Ersetzen der Formänderungsenergiefunktion \mathcal{W} . Eine Sammlung verschiedener Formänderungsenergiefunktionen und deren Diskussion an numerischen Beispielen ist beispielsweise der Arbeit von DOLL [38] zu entnehmen.

1.3.2. Elastoplastizität

Viele Werkstoffe weisen bei Belastungsversuchen bleibende Deformationen auch nach Wegnahme der Belastung auf. Eine realistische Beschreibung solcher Materialien erfordert die Formulierung *elastoplastischer* Materialgesetze, deren Entwicklung einen wichtigen Forschungsaspekt in der Strukturmechanik darstellt. Eine detaillierte Herleitung der Zusammenhänge der Elastoplastizität würde den vorgesehenen Rahmen der vorliegenden Arbeit sprengen, es werden also im Folgenden nur die wesentlichen Zusammenhänge dargestellt und die implementierten Stoffgesetze erläutert. Im weiteren Verlauf der Arbeit wird in Kapitel 4 auch auf die programmtechnische Umsetzung eingegangen, die bei diesen Algorithmen im Bezug auf Effizienz von besonderem Interesse ist. Für ausführliche Beschreibungen

der verschiedenen Modelle und Hinweise zu deren Umsetzung sei auf das umfangreiche Schrifttum zu dieser Thematik verwiesen (siehe z. B. [37, 84, 116] u. v. a.).

Grundlage der Elastoplastizität ist die Aufspaltung der Gesamtverzerrungen in einen elastischen und einen plastischen Anteil. Hinsichtlich der getroffenen Annahmen muss zwischen kleinen und großen Deformationen unterschieden werden. Für die Elementformulierungen der vorliegenden Arbeit wurden jeweils beide Versionen umgesetzt. Ein wesentlicher Unterschied im Gegensatz zur Hyperelastizität ist die Abhängigkeit des momentanen Materialverhaltens von der Belastungsgeschichte, eine Bestimmung der Spannungen nur aus dem aktuellen Deformationszustand ist also im Allgemeinen nicht möglich. Um dieser Tatsache gerecht zu werden, müssen bei Materialformulierungen mit Plastizität stets bestimmte Größen als sogenannte *Geschichtsvariablen* abgespeichert werden.

Elastoplastizität für kleine Deformationen

Der Elastoplastizität für kleine Deformationen liegt eine additive Aufspaltung der Gesamtverzerrungen in einen elastischen und einen plastischen Anteil

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p \quad (1.35)$$

zugrunde, wobei $\boldsymbol{\varepsilon}$ der lineare Anteil des Verzerrungstensors aus Gleichung (1.12), also der symmetrische Anteil des Verschiebungsgradienten ist. Das Auftreten bleibender Deformationen beim Überschreiten einer Grenzspannung σ_y , der sogenannten *Fließspannung*, ist über die Fließbedingung

$$\Phi(\boldsymbol{\sigma}, \xi_\alpha) \begin{cases} < 0 & \text{elastischer Bereich} & \Rightarrow & \text{keine plastischen Deformationen} \\ > 0 & \text{unzulässiger Bereich} & \Rightarrow & \Phi = 0 \text{ muß iterativ erfüllt werden} \end{cases} \quad (1.36)$$

definiert. Sie ist eine Funktion der Spannungen $\boldsymbol{\sigma}$ und innerer Variablen ξ_α . Im vorliegenden Fall lautet die Fließbedingung

$$\Phi(\boldsymbol{\sigma}, \gamma^p) = \sigma_V - \sigma_y(\gamma^p) \quad \text{z. B. mit} \quad \sigma_y(\gamma^p) = \sigma_y^0 + c_1 \gamma^p + c_2 \left(1 - e^{-c_3 \gamma^p}\right) \quad (1.37)$$

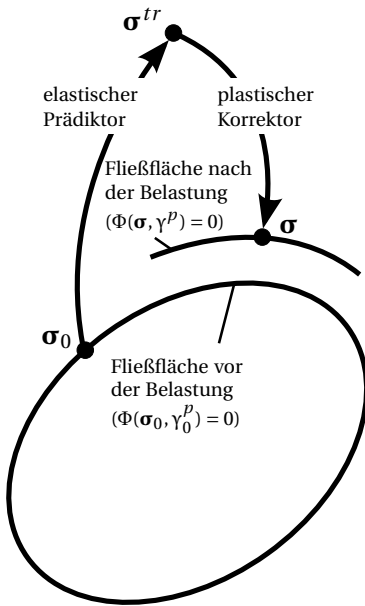
mit der VON MISES *Vergleichsspannung*

$$\sigma_V = \sqrt{\frac{3}{2} \mathbf{s} : \mathbf{s}}. \quad (1.38)$$

Die innere Variable zur Steuerung der Verfestigung kann als *akkumulierte plastische Verzerrung* γ^p interpretiert werden. Es wurde hier eine nichtlineare, isotrope Verfestigung gewählt, die über die initiale Fließspannung σ_y^0 , sowie über drei weitere – vom Berechner zu wählende Parameter c_1 , c_2 und c_3 – gesteuert wird. Andere Annahmen für die Ver- bzw. Entfestigung sind je nach dem zu beschreibendem Material möglich.

Zunächst werden im sogenannten *elastischen Prädiktorschritt trial* Verzerrungen $\boldsymbol{\varepsilon}^{tr}$ und – unter der Annahme rein elastischen Materialverhaltens – *trial* Spannungen $\boldsymbol{\sigma}^{tr}$ berechnet. Letztere ergeben sich über das Elastizitätsgesetz, also im vorliegenden Fall analog zu Gleichung (1.31)

$$\boldsymbol{\sigma}^{tr} = \frac{\partial \mathcal{W}}{\partial \boldsymbol{\varepsilon}^{tr}} \quad \text{mit} \quad \boldsymbol{\varepsilon}^{tr} = \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_0^p \quad (1.39)$$



- $\boldsymbol{\epsilon}^{tr}$ und $\boldsymbol{\sigma}^{tr}$ aus aktuellem Verschiebungszustand
 $\gamma^{p,tr} = \gamma_0^p$ aus Materialgeschichte
- wenn $\Phi(\boldsymbol{\sigma}^{tr}, \gamma^{p,tr}) \leq 0$, dann $\boldsymbol{\epsilon}^p = \boldsymbol{\epsilon}_0^p$, $\gamma^p = \gamma_0^p$
- sonst: *return mapping* Verfahren
(exemplarisch für Schritt $i \rightarrow i+1$)
 - $\boldsymbol{\epsilon}_i^e = \boldsymbol{\epsilon}^{tr} - \boldsymbol{\epsilon}_i^p$, $\Delta \gamma_i = \gamma_i^p - \gamma^{p,tr}$
 - $\boldsymbol{\sigma}_i(\boldsymbol{\epsilon}_i^e) = \frac{\partial \mathcal{W}(\boldsymbol{\epsilon}_i^e)}{\partial \boldsymbol{\epsilon}_i^e}$, $\mathbf{s}_i = \boldsymbol{\sigma}_i - \frac{\text{tr } \boldsymbol{\sigma}_i}{3} \mathbf{1}$, $\sigma_{v,i} = \sqrt{\frac{3}{2} \mathbf{s}_i : \mathbf{s}_i}$
 - Verfestigung z. B.: $\sigma_{y,i} = \sigma_y^0 + c_1 \gamma_i^p + c_2 (1 - e^{-c_3 \gamma_i^p})$
 - Fließregel: $\Phi(\boldsymbol{\sigma}_i, \gamma_i) = \sigma_{v,i}(\boldsymbol{\sigma}_i) - \sigma_{y,i}(\gamma_i^p)$
 - $\boldsymbol{\Psi}(\boldsymbol{\epsilon}_i^p, \gamma_i^p) = \begin{cases} \boldsymbol{\epsilon}_i^p - \boldsymbol{\epsilon}_0^p - \Delta \lambda_i \frac{\partial \Phi(\boldsymbol{\sigma}_i, \gamma_i^p)}{\partial \boldsymbol{\sigma}_i} \\ \Phi(\boldsymbol{\sigma}_i, \gamma_i^p) \end{cases}$
 - Newton-Iterationsschritt zur Lösung von $\boldsymbol{\Psi} = \mathbf{0}$
 $\Rightarrow \boldsymbol{\epsilon}_{i+1}^p, \gamma_{i+1}^p$
- Abbruchkriterium erreicht $\Rightarrow \boldsymbol{\epsilon}^p = \boldsymbol{\epsilon}_{i+1}^p$, $\gamma^p = \gamma_i^p$
- Berechnung der Spannungen $\boldsymbol{\sigma}$
Abspeichern von $\boldsymbol{\epsilon}^p$ und γ^p als Geschichtsvariablen

Abbildung 1.2.: Grafische Interpretation der Prädiktor-Korrektor Methode (vgl. [37, 116]) und algorithmische Umsetzung des *return mapping* Verfahrens

mit den plastischen Verzerrungen $\boldsymbol{\epsilon}_0^p$ aus der Materialgeschichte und z. B. der Formänderungsenergiefunktion aus Gleichung (1.32). Ergibt sich $\Phi(\boldsymbol{\sigma}^{tr}, \gamma^{p,tr}) < 0$ mit den akkumulierten plastischen Verzerrungen $\gamma^{p,tr}$ ebenfalls aus der Materialgeschichte, so sind die tatsächlichen Spannungen $\boldsymbol{\sigma} = \boldsymbol{\sigma}^{tr}$.

Liefert die Auswertung $\Phi > 0$, d. h. einen Zustand für welchen die Fließbedingung nicht definiert ist, so müssen die plastischen Verzerrungen so bestimmt werden, dass Gleichung (1.39) Spannungen liefert, die für Gleichung (1.37) genau den Wert $\Phi = 0$ liefern. Dies wird über den sogenannten *plastischen Korrektorschritt* umgesetzt, bei dem die eingangs berechneten *trial* Spannungen iterativ durch *return mapping* auf die Fließfläche zurückgeführt werden. Eine geometrische Interpretation des Verfahrens, die auch den Einfluss der veränderlichen inneren Variable im Verfestigungsgesetz berücksichtigt, wurde DE SOUSA NETO ET. AL. [37] entnommen und ist in Abbildung 1.2 angegeben. Die algorithmische Struktur des *Prädiktor-Korrektor Verfahrens* für kleine Deformationen, auf deren programmtechnische Umsetzung in Kapitel 4 noch genauer eingegangen wird, ist ebenfalls Abbildung 1.2 zu entnehmen.

Die bereits angesprochenen *Geschichtsvariablen*, die zur Berücksichtigung der Belastungshistorie gespeichert werden müssen, sind im vorliegenden Fall die plastischen Verzerrungen $\boldsymbol{\epsilon}^p$, sowie die akkumulierte plastische Dehnung γ^p . Für eine diskretisierte Kontinuumsformulierung fallen also an jedem Integrationspunkt 7 Geschichtsvariablen an, die entsprechend dem in Abbildung 1.2 skizzierten Verfahren aufdatiert werden müssen.

Elastoplastizität für endliche Deformationen

Bei der Erweiterung der Elastoplastizität auf große Deformationen hat sich die von LEE & LIU [82] und LEE [81] vorgeschlagene, multiplikative Aufspaltung des Deformationsgradienten aus Gleichung (1.3) durchgesetzt. Diese entspricht der Einführung einer plastischen Zwischenkonfiguration $\tilde{\mathbf{x}}$. Es ist also

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \frac{\partial \mathbf{x}}{\partial \tilde{\mathbf{x}}} \cdot \frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{X}}, \quad \text{mit} \quad \frac{\partial \mathbf{x}}{\partial \tilde{\mathbf{x}}} = \mathbf{F}^e \quad \text{und} \quad \frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{X}} = \mathbf{F}^p \quad \Rightarrow \quad \mathbf{F} = \mathbf{F}^e \cdot \mathbf{F}^p \quad (1.40)$$

Eine Möglichkeit der Spannungsberechnung, die für die vorliegende Implementierung eines isotropen Materials gewählt wurde, ist die Bestimmung der KIRCHHOFF Spannungen aus dem elastischen Anteil des linken CAUCHY-GREEN-Tensors, also

$$\boldsymbol{\tau} = 2 \frac{\partial \mathcal{W}}{\partial \mathbf{b}^e} \mathbf{b}^e \quad \text{mit} \quad \mathbf{b}^e = \mathbf{F}^e \cdot \mathbf{F}^{eT} = \mathbf{F} \cdot \mathbf{C}^{p-1} \cdot \mathbf{F}^T \quad \text{und} \quad \mathbf{C}^p = \mathbf{F}^{pT} \cdot \mathbf{F}^p, \quad (1.41)$$

mit der Formänderungsenergiefunktion z. B. des NEO-HOOKE-Gesetzes $\mathcal{W}(\mathbf{b})$ aus Gleichung (1.34).

Mit den Einträgen des inversen rechten CAUCHY-GREEN-Tensors und der akkumulierten plastischen Verzerrung γ^p als Geschichtsvariablen lässt sich bei isotropem Materialverhalten und isotroper Verfestigung eine der für kleine Verzerrungen äquivalente Vorgehensweise mit dem *Prädiktor-Korrektor-Verfahren* angeben. Für detaillierte Herleitungen sei auf das eingangs zitierte Schrifttum verwiesen. Die algorithmische Aufbereitung, analog zu der in Abbildung 1.2 skizzierten, ist Abbildung 1.3 zu entnehmen.

Wie zuvor beim Vorgehen für kleine Deformationen müssen für den Kontinuums-Fall 7 Geschichtsvariablen gespeichert werden. Bei der Programmierung ist jedoch zu beachten, dass für den elastischen Fall die plastische Zwischenkonfiguration $\tilde{\mathbf{x}}$ mit der Ausgangskonfiguration zusammenfällt und somit

$$\mathbf{F}_0 = \mathbf{F}_0^e \cdot \mathbf{F}_0^p = \mathbf{F}_0^e \quad \Leftrightarrow \quad \mathbf{F}_0^p = \frac{\partial \tilde{\mathbf{x}}_0}{\partial \mathbf{X}} = \frac{\partial \mathbf{X}}{\partial \mathbf{X}} = \mathbf{1} \quad (1.42)$$

gilt. Für den rechten CAUCHY-GREEN-Tensor und dessen Inverse bedeutet dies

$$\mathbf{C}_0^p = \mathbf{C}_0^{p-1} = \mathbf{1}, \quad (1.43)$$

es muss also bei der Initialisierung des Geschichtsfeldes für die Hauptdiagonaleinträge der initiale Wert 1 gesetzt werden.

1.4. Schwache Form des Gleichgewichts

Die Beziehungen (1.7), (1.22) und (1.31) aus den Abschnitten 1.1 – 1.3

$$\text{Kinematik:} \quad \mathbf{E} = \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{1}) \quad \text{auf } \mathcal{B}_0$$

$$\text{Gleichgewicht:} \quad \text{div } \mathbf{P} + \rho_0 \mathbf{b} - \rho_0 \ddot{\mathbf{u}} = 0 \quad \text{auf } \mathcal{B}_0$$

$$\text{Stoffgesetz:} \quad \mathbf{S} = \frac{\partial \mathcal{W}(\mathbf{E})}{\partial \mathbf{E}} \quad \text{auf } \mathcal{B}_0$$

- \mathbf{F} aus aktuellem Verschiebungszustand
 \mathbf{C}_0^{p-1} und $\gamma^{tr} = \gamma_0$ aus Materialgeschichte
 $\Rightarrow \mathbf{b}^{e,tr} = \mathbf{F} \cdot \mathbf{C}_0^{p-1} \cdot \mathbf{F}^{tr} \Rightarrow \boldsymbol{\tau}^{tr} = 2 \frac{\partial \mathcal{W}}{\partial \mathbf{b}^{e,tr}} \mathbf{b}^{e,tr}$
- wenn $\Phi(\boldsymbol{\tau}^{tr}, \gamma^{tr}) \leq 0$, dann $\mathbf{C}^{p-1} = \mathbf{C}_0^{p-1}$, $\gamma = \gamma_0$
- sonst: *return mapping* Verfahren
(exemplarisch für Schritt $i \rightarrow i+1$)
 - $\mathbf{b}_i^e = \mathbf{F} \cdot \mathbf{C}_i^{p-1} \cdot \mathbf{F}^{tr}$, $\Delta \gamma_i = \gamma_i - \gamma_0$
 - $\boldsymbol{\tau}_i(\mathbf{b}_i^e) = 2 \frac{\partial \mathcal{W}(\mathbf{b}_i^e)}{\partial \mathbf{b}_i^e} \mathbf{b}_i^e$, $\mathbf{s}_i = \boldsymbol{\tau}_i - \frac{\text{tr } \boldsymbol{\tau}_i}{3} \mathbf{1}$, $\sigma_{v,i} = \sqrt{\frac{3}{2} \mathbf{s}_i : \mathbf{s}_i}$
 - Verfestigung z. B.: $\sigma_{y,i} = \sigma_y^0 + c_1 \gamma_i + c_2 (1 - e^{-c_3 \gamma_i^p})$
 - Fließregel: $\Phi(\boldsymbol{\tau}_i, \gamma_i) = \sigma_{v,i}(\boldsymbol{\tau}_i) - \sigma_{y,i}(\gamma_i)$
 - $\boldsymbol{\Psi}(\mathbf{C}_i^{p-1}, \gamma_i) = \begin{cases} \mathbf{F} \cdot \mathbf{C}_i^{p-1} - \exp \left[-2 \Delta \gamma_i \frac{\partial \Phi(\boldsymbol{\tau}_i, \gamma_i)}{\partial \boldsymbol{\tau}_i} \right] \cdot \mathbf{F} \cdot \mathbf{C}_0^{p-1} \\ \Phi(\boldsymbol{\tau}_i, \gamma_i) \end{cases}$
 - Newton-Iterationsschritt zur Lösung von $\boldsymbol{\Psi} = \mathbf{0}$
 $\Rightarrow \mathbf{C}_{i+1}^{p-1}, \gamma_{i+1}$
- Abbruchkriterium erreicht $\Rightarrow \mathbf{C}^{p-1} = \mathbf{C}_{i+1}^{p-1}$, $\gamma = \gamma_{i+1}$
- Berechnung der Spannungen $\boldsymbol{\tau}$
Abspeichern von \mathbf{C}_{i+1}^{p-1} und γ als Geschichtsvariablen

Abbildung 1.3.: Algorithmische Umsetzung des *Prädiktor-Korrektor Verfahrens* für finite Elastoplastizität bei isotropem Materialverhalten und isotroper Verfestigung

lassen sich mit vorgegebenen Verschiebungs- oder DIRICHLET-Randbedingungen

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{auf } \partial \mathcal{B}_D \quad \text{für } 0 \leq t \leq T, \quad (1.44)$$

Spannungs- oder NEUMANN-Randbedingungen

$$\mathbf{P} \cdot \mathbf{N} = \bar{\mathbf{t}} \quad \text{auf } \partial \mathcal{B}_\sigma \quad \text{für } 0 \leq t \leq T \quad (1.45)$$

und Anfangsbedingungen für Verschiebungen und/ oder Geschwindigkeiten

$$\dot{\mathbf{u}} = \dot{\mathbf{u}}_0 \quad \text{bzw.} \quad \mathbf{u} = \mathbf{u}_0 \quad \text{auf } \mathcal{B}_0 \quad \text{für } t = 0 \quad (1.46)$$

zu einem Anfangsrandwertproblem zusammenfassen. Eine analytische Lösung dieser, auch als *starke Form des Gleichgewichts* bezeichneten Feldgleichungen ist nur für einige einfache Sonderfälle möglich. Die näherungsweise Lösung, hier zum Beispiel mit der Methode der finiten Elemente, erfordert zunächst den Übergang auf eine sogenannte *schwache Form*, die das Gleichgewicht nicht in jedem Punkt, sondern nur im integralen Mittel erfüllt.

Einen möglichen Zugang stellt das *Prinzip der virtuellen Verschiebungen* dar, bei dem eine vektorwertige Testfunktion – hier eine virtuelle Verschiebung $\delta \mathbf{u}$ – eingeführt wird, mit der jeder Term der Gleichgewichtsbeziehung skalar multipliziert wird. Die virtuellen Verschiebungen müssen dabei lediglich die DIRICHLET-Randbedingungen aus Gleichung (1.44)

exakt erfüllen. Anschließende Integration über das Gebiet \mathcal{B}_0 führt auf

$$\int_{\mathcal{B}_0} \operatorname{div} \mathbf{P} \cdot \delta \mathbf{u} \, dV + \int_{\mathcal{B}_0} \rho_0 (\mathbf{b} - \ddot{\mathbf{u}}) \cdot \delta \mathbf{u} \, dV = 0, \quad \text{für } 0 \leq t \leq T \quad (1.47)$$

Die Anwendung der Beziehungen aus Kinematik und Stoffgesetz und einige Zwischenschritte, die hier aus Gründen der Übersichtlichkeit übersprungen, jedoch dem Schrifttum entnommen werden können, führt schließlich auf die *schwache Form des Gleichgewichts* in der Form

$$\underbrace{\int_{\mathcal{B}_0} \mathbf{S} : \delta \mathbf{E} \, dV}_{\text{innere Energie}} - \underbrace{\int_{\mathcal{B}_0} \rho_0 \mathbf{b} \cdot \delta \mathbf{u} \, dV}_{\text{Arbeit der Volumenkräfte}} - \underbrace{\int_{\partial \mathcal{B}_\sigma} \bar{\mathbf{t}} \cdot \delta \mathbf{u} \, dA}_{\text{Arbeit der Oberflächenkräfte}} + \underbrace{\int_{\mathcal{B}_0} \rho_0 \ddot{\mathbf{u}} \cdot \delta \mathbf{u} \, dV}_{\text{Arbeit der Trägheitskräfte}} = 0. \quad (1.48)$$

Eine weitere Methode der Überführung der starken in eine schwache Form stellt z. B. das *Prinzip vom stationären Wert des Gesamtpotentials* dar. Es besagt, die gesamte potentielle Energie eines konservativen mechanischen Systems nehme in einer Gleichgewichtslage einen stationären Wert an. Dies ist gleichbedeutend mit der Aussage

$$\delta \Pi = 0, \quad \text{für } 0 \leq t \leq T, \quad (1.49)$$

für einen beliebigen aber festen Zeitpunkt t . Wird hier unter Voraussetzung der Existenz einer Formänderungsenergiefunktion $\mathcal{W}(\mathbf{E})$ das Gesamtpotential

$$\Pi = \int_{\mathcal{B}_0} \mathcal{W}(\mathbf{E}) \, dV - \int_{\mathcal{B}_0} \rho_0 \mathbf{b} \cdot \mathbf{u} \, dV - \int_{\partial \mathcal{B}_\sigma} \bar{\mathbf{t}} \cdot \mathbf{u} \, dA + \int_{\mathcal{B}_0} \rho_0 \ddot{\mathbf{u}} \cdot \mathbf{u} \, dV \quad (1.50)$$

aufgestellt, so ergibt sich durch Bildung der ersten Variation $\delta \Pi(\mathbf{u}, \delta \mathbf{u})$ wieder die schwache Form aus Gleichung (1.48). In dieser Betrachtung taucht als einzige zu diskretisierende Größe die Verschiebung \mathbf{u} auf, die daraus resultierenden Elementformulierungen werden auch als *Verschiebungselemente* bezeichnet.

Eine allgemeinere Form des Funktionals liefert das Prinzip nach HU-WASHIZU, in welchem neben den Verschiebungen auch die Verzerrungen \mathbf{E} und die Spannungen \mathbf{S} lediglich im integralen Mittel erfüllt werden. Ohne weitere Herleitung ergibt sich das Dreifeldfunktional nach HU-WASHIZU zu

$$\Pi_{\text{HW}}(\mathbf{u}, \mathbf{E}, \mathbf{S}) = \int_{\mathcal{B}_0} \mathcal{W}(\mathbf{E}) + \mathbf{S} : (\mathbf{E}^u - \mathbf{E}) \, dV - \left[\int_{\mathcal{B}_0} \rho_0 \mathbf{b} \cdot \delta \mathbf{u} \, dV + \int_{\partial \mathcal{B}_\sigma} \bar{\mathbf{t}} \cdot \delta \mathbf{u} \, dA \right] + \int_{\mathcal{B}_0} \rho_0 \ddot{\mathbf{u}} \cdot \delta \mathbf{u} \, dV. \quad (1.51)$$

Dabei werden die Verzerrungen \mathbf{E}^u direkt aus den Verschiebungen \mathbf{u} berechnet, wobei \mathbf{E} ohne Index sowie \mathbf{S} unabhängige Größen darstellen. Diese Form eignet sich besonders zur Entwicklung sogenannter gemischter Elementformulierungen, bei denen neben der Verschiebungsinterpolation auch Ansätze für die Verzerrungen und/ oder Spannungen gemacht werden. Durch Anwendung der ersten Variation auf Gleichung (1.51), lässt sich auch hier formal eine schwache Form $\delta \Pi(\mathbf{u}, \delta \mathbf{u}, \mathbf{E}, \delta \mathbf{E}, \mathbf{S}, \delta \mathbf{S})$ angeben, wobei in der Anwendung (vgl.

Abschnitte 3.4.2 und 3.4.4) üblicherweise zunächst eine Reduzierung auf ein Zweifeldfunktional durch Formulierung entsprechender Bedingungen erfolgt.

Mit der *schwachen Form des Gleichgewichts* (1.48) liegt nun die Grundlage für eine Diskretisierung im Rahmen der Methode der finiten Elemente vor. Die Diskretisierung der räumlichen Größen – also der Geometrie und der Verschiebungen – wird im nächsten Abschnitt besprochen, die zeitliche Diskretisierung erfolgt zusammen mit der Vorstellung zweier Zeitintegrationsverfahren in Kapitel 2.

1.5. Räumliche Diskretisierung

Zur Lösung der im vorigen Abschnitt hergeleiteten Gleichgewichtsbeziehung im Rahmen der Methode der finiten Elemente werden die im Gebiet \mathcal{B}_0 auftretenden geometrischen Größen näherungsweise abgebildet, um das Differentialgleichungssystem (1.48) in ein System algebraischer Gleichungen zu überführen. Durch Unterteilung des Gebietes \mathcal{B}_0 in n_{el} einfache nicht überlappende Teilgebiete $\mathcal{B}_{0,h}^i$ – den finiten Elementen – ergibt sich die Näherung

$$\mathcal{B}_0 \approx \mathcal{B}_{0,h} = \bigcup_{i=1}^{n_{el}} \mathcal{B}_{0,h}^i \quad \text{mit} \quad \mathcal{B}_{0,h}^i \cap \mathcal{B}_{0,h}^j = \emptyset \quad \forall i \neq j. \quad (1.52)$$

Es werden diskrete Knoten eingeführt, deren geometrische Lage durch Ortsvektoren \mathbf{X}_i und deren Freiheitsgrade (Verschiebungen, Rotationen) durch \mathbf{d}_i repräsentiert werden. Zwischen den Knoten werden die diskreten Werte \mathbf{X}_i und \mathbf{d}_i durch geeignete Ansatzfunktionen N_i interpoliert. Für ein finites Element mit n Knoten ergibt sich für die Interpolation der Geometrie und der Verschiebungen allgemein

$$\mathbf{X}_h^e = \sum_{i=1}^n N_i \mathbf{X}_i, \quad \text{bzw.} \quad \mathbf{u}_h^e = \sum_{i=1}^n N_i \mathbf{d}_i, \quad (1.53)$$

wobei der Index h die Diskretisierung der kontinuierlichen Größen \mathbf{X} und \mathbf{u} anzeigt. Die virtuellen Verschiebungen werden mit den gleichen Ansatzfunktionen zu

$$\delta \mathbf{u}_h^e = \sum_{i=1}^n N_i \delta \mathbf{d}_i \quad (1.54)$$

interpoliert. Die Verwendung identischer Ansatzfunktionen N_i für Geometrie und Freiheitsgrade wird auch als *isoparametrisches Konzept* bezeichnet. Durch Einsetzen der diskretisierten Größen in die einzelnen Terme von Gleichung (1.48) ergibt sich für die innere Energie

$$\int_{\mathcal{B}_0} \mathbf{S} : \delta \mathbf{E} \, dV = \delta \Pi^{\text{int}}(\mathbf{u}) \approx \delta \Pi_h^{\text{int}}(\mathbf{d}) = \frac{\partial \Pi_h^{\text{int}}}{\partial \mathbf{d}} \delta \mathbf{d}, \quad (1.55)$$

für die Arbeit der äußeren Kräfte

$$\int_{\mathcal{B}_0} \rho_0 \mathbf{b} \cdot \delta \mathbf{u} \, dV + \int_{\partial \mathcal{B}_\sigma} \bar{\mathbf{t}} \cdot \delta \mathbf{u} \, dA = \delta \Pi^{\text{ext}}(\mathbf{u}) \approx \delta \Pi_h^{\text{ext}}(\mathbf{d}) = \frac{\partial \Pi_h^{\text{ext}}}{\partial \mathbf{d}} \delta \mathbf{d} \quad (1.56)$$

und für die Arbeit der D'ALEMBERTSchen Trägheitskräfte

$$\int_{\mathcal{B}_0} \rho_0 \ddot{\mathbf{u}} \cdot \delta \mathbf{u} \, dV \approx \mathbf{M} \cdot \ddot{\mathbf{d}} \cdot \delta \mathbf{d} \quad (1.57)$$

wobei in \mathbf{d} alle Freiheitsgrade des Gesamtsystems zusammengefasst sind. Mit dem Fundamentallemma der Variationsrechnung, für das die Gleichgewichtsbedingung für beliebige Werte $\delta \mathbf{d} \neq \mathbf{0}$ erfüllt sein muss, folgt schließlich die Bewegungsgleichung ohne Berücksichtigung von Dämpfung in semi-diskreter Form zu

$$\mathbf{M} \ddot{\mathbf{d}} + \mathbf{f}^{\text{int}}(\mathbf{d}) = \mathbf{f}^{\text{ext}}. \quad (1.58)$$

Durch Assemblierung über alle Elemente (nel) gemäß Gleichung (1.52) ergeben sich die Vektoren der inneren und äußeren Kräfte

$$\mathbf{f}^{\text{int}} = \bigcup_{k=1}^{\text{nel}} \frac{\partial \Pi_h^{\text{int},k}}{\partial \mathbf{d}^k}, \quad \text{bzw.} \quad \mathbf{f}^{\text{ext}} = \bigcup_{k=1}^{\text{nel}} \frac{\partial \Pi_h^{\text{ext},k}}{\partial \mathbf{d}^k}, \quad (1.59)$$

wobei der Vektor \mathbf{d}^k mit Index oben die Freiheitsgrade aller Knoten des k -ten Elementes beinhaltet. Weiters ist die *konsistente Massenmatrix* definiert als

$$\mathbf{M} = \bigcup_{k=1}^{\text{nel}} \int_{\mathcal{B}_0^k} \rho_{0,k} \mathbf{N}^T \cdot \mathbf{N} \, dV, \quad \text{mit} \quad \mathbf{N} = \left(\begin{array}{c|ccc|c|ccc} \mathbf{N}_1 & & \mathbf{0} & \dots & \mathbf{N}_n & & \mathbf{0} \\ & \ddots & & & & \ddots & \\ \mathbf{0} & & \mathbf{N}_1 & \dots & \mathbf{0} & & \mathbf{N}_n \end{array} \right), \quad (1.60)$$

auf deren Eigenschaften in Abschnitt 3.3 noch genauer eingegangen wird. Diese lässt sich formell auch aus der Definition der kinetischen Energie herleiten

$$\Pi^{\text{kin}} = \frac{1}{2} \dot{\mathbf{d}}^T \mathbf{M} \dot{\mathbf{d}} \quad \Rightarrow \quad \mathbf{M} = \frac{\partial^2 \Pi^{\text{kin}}}{\partial \dot{\mathbf{d}}^2}, \quad (1.61)$$

worauf beispielsweise in [43] genauer eingegangen wird.

2. Lösung zeitabhängiger Probleme

Die für die vorliegende Arbeit implementierten Elementformulierungen werden innerhalb eines expliziten Zeitintegrationsverfahrens verwendet, was hinsichtlich ihrer effizienten Umsetzungen zu besonderen Anforderungen führt. Im folgenden Abschnitt wird daher das zentrale Differenzenverfahren als Lösungsverfahren der diskretisierten Bewegungsgleichung und dessen Implementierung in das am Institut verwendete quelloffene Finite-Elemente-Programm FEAP-MEKA [112] beschrieben. Die wesentlichen Unterschiede zu impliziten Verfahren, welche insbesondere die Notwendigkeit effizienter Elementroutinen motivieren, werden durch Vergleich mit dem in kommerziellen FE-Codes häufig eingesetzten NEWMARK-Verfahren herausgearbeitet.

Eine umfassende Übersicht über Zeitintegrationsverfahren wird von WOOD [129] angegeben, weitere Details insbesondere zum zentralen Differenzenverfahren finden sich u. a. bei BELYTSCHKO, LIU & MORAN [15] und HALLQUIST [49].

2.1. Diskretisierung der Bewegungsgleichung

Wie in Abschnitt 1.5 beschrieben führt die räumliche Diskretisierung der Bewegungsgleichungen auf die Form

$$\mathbf{M}\ddot{\mathbf{d}} + \mathbf{f}^{\text{int}}(\mathbf{d}) = \mathbf{f}^{\text{ext}}, \quad (2.1)$$

mit der konsistenten Massenmatrix \mathbf{M} und den inneren und äußeren Kräften \mathbf{f}^{int} bzw. \mathbf{f}^{ext} . Häufig wird zur Beschreibung von Energieverlusten durch innere Reibung oder Reibung in Verbindungsmitteln eine geschwindigkeitsproportionale Dämpfung angenommen, die durch die Einführung einer konstanten Dämpfungsmatrix \mathbf{C}^d beschrieben wird. Es ergibt sich dann

$$\mathbf{M}\ddot{\mathbf{d}} + \mathbf{C}^d \dot{\mathbf{d}} + \mathbf{f}^{\text{int}}(\mathbf{d}) = \mathbf{f}^{\text{ext}}, \quad (2.2)$$

wobei sich die Dämpfungsmatrix durch das sog. RAYLEIGH-Modell als eine Kombination aus masse- und steifigkeitsproportionalen Termen beschreiben lässt, worauf in Abschnitt 2.2.4 noch genauer eingegangen wird. Diese Bewegungsgleichung wird auch als *semidiskret* bezeichnet, da sie diskret im Raum aber kontinuierlich in der Zeit ist. Eine Diskretisierung in der Zeit erfolgt nun durch Auswertung der Verschiebungsgrößen zu einem bestimmten Zeitpunkt $t_{n+1} = t_n + \Delta t$

$$\mathbf{M}\ddot{\mathbf{d}}_{n+1} + \mathbf{C}^d \dot{\mathbf{d}}_{n+1} + \mathbf{f}_{n+1}^{\text{int}}(\mathbf{d}_{n+1}) = \mathbf{f}_{n+1}^{\text{ext}}. \quad (2.3)$$

Die vollständige Beschreibung des Anfangswertproblems erfordert nun noch die Vorgabe von Anfangsbedingungen etwa zur Zeit $t = 0$

$$\mathbf{d}_0 = \bar{\mathbf{d}}, \quad \dot{\mathbf{d}}_0 = \bar{\mathbf{v}} \text{ bzw. } \ddot{\mathbf{d}}_0 = \bar{\mathbf{a}}. \quad (2.4)$$

Zur Integration der Bewegungsgleichungen stehen nun grundsätzlich zwei verschiedene Typen von Verfahren zur Verfügung.

Bei *expliziten Zeitintegrationsverfahren* hängt die Lösung \mathbf{d}_{n+1} zur Zeit t_{n+1} nur von bekannten Größen zur Zeit t_n ab. In Kombination mit einer diagonalisierten Massenmatrix – eine ausführliche Diskussion erfolgt in Abschnitt 3.3 – führt dies ausschließlich zu Vektoroperationen auf globaler Ebene, da die Bewegungsgleichungen (2.3) dann vollständig entkoppelt sind. Das Verfahren ist somit sehr effizient, der numerische Aufwand pro Zeitschritt also sehr gering. Allerdings ist bei expliziten Verfahren der Zeitschritt beschränkt, darf also einen maximalen Wert nicht überschreiten, der an Struktureigenschaften (Masse, Steifigkeit) gekoppelt ist. Im Abschnitt 2.2 wird das *zentrale Differenzenverfahren* beschrieben, das bereits 1967 von VERLET [124] zur Beschreibung des dynamischen Verhaltens von Partikeln formuliert und später auf die Methode der finiten Elemente angewendet wurde. Es ist heute in kommerziellen FE-Programmen LS-DYNA [83], PAM-CRASH [42], ABAQUS/EXPLICIT [118], etc. implementiert. Auf die Bestimmung des kritischen Zeitschrittes wird in Abschnitt 2.2.2 noch genauer eingegangen.

Implizite Verfahren sind dadurch charakterisiert, dass die Lösung zur Zeit t_{n+1} sowohl von bekannten Größen zur Zeit t_n als auch von unbekanntem Werten zur Zeit t_{n+1} abhängt. Es muss also zu jedem Zeitschritt ein im Allgemeinen nichtlineares Gleichungssystem gelöst werden, wodurch die Zahl der erforderlichen Rechenoperationen und somit der numerische Aufwand je Zeitschritt deutlich größer ist als beim expliziten Verfahren. Bei bestimmten impliziten Verfahren ist jedoch die Zeitschrittweite nicht an ein Kriterium gekoppelt und kann theoretisch beliebig groß gewählt werden. Dennoch muss die Wahl des Zeitschrittes physikalisch sinnvoll sein, weshalb insbesondere bei stark nichtlinearen Problemen bzw. Impakt- oder Crashsimulationen auch hier ein sehr kleiner Zeitschritt erforderlich wird, um das Problem abbilden zu können. Als Beispiel für ein implizites Zeitintegrationsverfahren, welches in der Strukturmechanik überwiegend in kommerziellen impliziten Programmen wie ANSYS [10], ABAQUS/STANDARD [118] aber auch LS-DYNA [83] Anwendung findet, wird in Abschnitt 2.3 das NEWMARK-Verfahren, das auf NEWMARK [97] zurückgeht, kurz skizziert. Weitere Verfahren können dem eingangs zitierten Schrifttum entnommen werden.

2.2. Explizite Zeitintegration

2.2.1. Das zentrale Differenzenverfahren

Für ein ungedämpftes System lassen sich die Beschleunigungen zum Zeitpunkt t_n bestimmen über

$$\ddot{\mathbf{d}}_n = \mathbf{M}^{-1} (\mathbf{f}_n^{\text{ext}} - \mathbf{f}_n^{\text{int}}) \quad (2.5)$$

mit der inversen Massenmatrix und den äußeren und inneren Kräften. Die Methode der *zentralen Differenzen* basiert nun auf der Annahme einer *zentralen* gleichmäßigen Gewichtung der Geschwindigkeiten um den Zeitpunkt t_n , welche hierzu in den halben Zeitschritten $t_{n-1/2}$

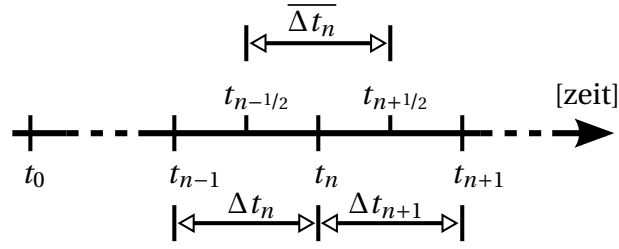


Abbildung 2.1.: Zeitschrittweiten im zentralen Differenzenverfahren

und $t_{n+1/2}$ ausgewertet werden. Mit der mittleren Zeitschrittweite $\overline{\Delta t_n}$ aus Abbildung 2.1 folgt schließlich

$$\ddot{\mathbf{d}}_n = \frac{\dot{\mathbf{d}}_{n+1/2} - \dot{\mathbf{d}}_{n-1/2}}{\overline{\Delta t_n}} \Rightarrow \dot{\mathbf{d}}_{n+1/2} = \dot{\mathbf{d}}_{n-1/2} + \mathbf{a}_n \overline{\Delta t_n}. \quad (2.6)$$

In gleicher Weise lassen sich die Verschiebungen bestimmen. Durch zentrale Differenzen um den Zeitpunkt $t_{n+1/2}$ ergibt sich

$$\dot{\mathbf{d}}_{n+1/2} = \frac{\mathbf{d}_{n+1} - \mathbf{d}_n}{\Delta t_n} \Rightarrow \mathbf{d}_{n+1} = \mathbf{d}_n + \dot{\mathbf{d}}_{n+1/2} \Delta t_n. \quad (2.7)$$

Die Beschleunigungen lassen sich auch direkt in Abhängigkeit der Verschiebungen angeben. Durch Einsetzen von (2.7) für $t_{n+1/2}$ und $t_{n-1/2}$ in (2.6) ergibt sich

$$\ddot{\mathbf{d}}_n = \frac{\frac{\mathbf{d}_{n+1} - \mathbf{d}_n}{\Delta t_n} - \frac{\mathbf{d}_n - \mathbf{d}_{n-1}}{\Delta t_{n-1}}}{\overline{\Delta t_n}} = \frac{\Delta t_{n-1} (\mathbf{d}_{n+1} - \mathbf{d}_n) - \Delta t_n (\mathbf{d}_n - \mathbf{d}_{n-1})}{\Delta t_n \overline{\Delta t_n} \Delta t_{n-1}}. \quad (2.8)$$

Unter der Annahme äquidistanter Zeitschritte ($\Delta t_{n-1} = \overline{\Delta t_n} = \Delta t_n$) ergibt sich schließlich die bekannte Beziehung der zentralen Differenzen für die zweite Ableitung einer Funktion

$$\ddot{\mathbf{d}}_n = \frac{\mathbf{d}_{n+1} - 2\mathbf{d}_n + \mathbf{d}_{n-1}}{(\Delta t_n)^2}, \quad (2.9)$$

deren Genauigkeit von der Ordnung $\mathcal{O}(\Delta t^2)$ ist.

Zur Berücksichtigung von Dämpfung wird wie in Abschnitt 2.1 beschrieben ein geschwindigkeitsproportionales Dämpfungsmodell gewählt und eine Dämpfungsmatrix \mathbf{C}^d als Kombination aus Massen- und Steifigkeitsmatrix eingeführt. Aufgrund der sehr kleinen Zeitschritte wird die ‐Dämpfungskraft‐ im zentralen Differenzenverfahren *nachgezogen*, also mit den Geschwindigkeiten aus dem Zeitschritt $t_{n-1/2}$ bestimmt. Dies hat eine Reduktion der Genauigkeit des Verfahrens zu Folge. Für Gleichung (2.5) ergibt sich dann

$$\ddot{\mathbf{d}}_n = \mathbf{M}^{-1} \left(\mathbf{f}_n^{\text{ext}} - \mathbf{f}_n^{\text{int}} - \mathbf{C}^d \dot{\mathbf{d}}_{n-1/2} \right). \quad (2.10)$$

Der erste Schritt des Zeitintegrationsalgorithmus zur Zeit t_0 erfordert bei Vorgabe von Anfangsbedingungen wie in Gleichung (2.4) auch Geschwindigkeiten zur Zeit $t_{0-1/2}$. WRIGGERS gibt in [130] an, die Verschiebungen \mathbf{d}_{0-1} über eine Taylorreihenentwicklung

$$\mathbf{d}_{0-1} = \mathbf{d}_0 - \Delta t_0 \dot{\mathbf{d}}_0 + \frac{(\Delta t_0)^2}{2} \ddot{\mathbf{d}}_0 \quad (2.11)$$

zu berechnen. In der vorliegenden in Abschnitt 2.2 beschriebenen Implementierung wurde auf diesen Schritt verzichtet und die Geschwindigkeiten $\dot{\mathbf{d}}_{0+1/2}$ vereinfacht über

$$\dot{\mathbf{d}}_{0+1/2} = \dot{\mathbf{d}}_0 + \mathbf{a}_0 \Delta t_0 \quad (2.12)$$

berechnet. Diese Vereinfachung führte wegen der sehr kleinen Zeitschritte in keiner der betrachteten Anwendungen zu Problemen.

2.2.2. Zeitschrittweite

Explizite Zeitintegrationsverfahren sind sogenannte *bedingt stabile* Verfahren, das heißt die Lösung steigt unbegrenzt an, wenn der verwendete Zeitschritt größer ist als ein bestimmter Wert. Diese auch als *kritischer Zeitschritt* bezeichnete Größe wird über die größte in der Struktur auftretende Eigenfrequenz ω_{\max} bestimmt, welche sich über die Eigenfrequenzen des maßgebenden Elementes abschätzen mit $\omega_{\max} \leq \max \omega_{e,i}$ abschätzen lässt. Das COURANT-Kriterium nach COURANT, FRIEDRICHS & LEWY [36] lautet schließlich

$$\Delta t = \alpha_{\Delta t} \Delta t_{\text{krit}}, \quad \text{mit} \quad \Delta t_{\text{krit}} = \frac{2}{\omega_{\max}} \leq \frac{2}{\max \omega_{e,i}} = \min_e \frac{\ell_e}{c_{s,e}}, \quad (2.13)$$

mit einer charakteristischen Elementlänge ℓ_e und der Wellenausbreitungsgeschwindigkeit $c_{s,e}$ jeweils des Elementes e . Zur Berücksichtigung von Nichtlinearitäten, die zu einer Reduktion des kritischen Zeitschrittes führen, wird dieser in Gleichung (2.13) mit dem Faktor $\alpha_{\Delta t}$, für den üblicherweise $0.2 \leq \alpha_{\Delta t} \leq 0.98$ gilt gemindert. Es ist ersichtlich, dass sich die Größe des erforderlichen Zeitschrittes im Laufe einer Simulation ändert, daher muss dieser innerhalb des Algorithmus ständig neu bestimmt und aktualisiert werden. Dies geschieht bei der Aufstellung des Vektors der inneren Knotenkräfte und wird daher in Kapitel 4 bei der Beschreibung der Implementierung nochmals für die jeweiligen Elementformulierungen erläutert. Basierend auf einer genaueren Ermittlung der Eigenfrequenzen, lassen sich weniger konservative Abschätzungen der Zeitschrittweite formulieren, was sich günstig auf den numerischen Aufwand der Zeitintegrationsverfahren auswirken kann; exemplarisch sei auf BENSON [17] verwiesen.

2.2.3. Implementierung

Da die Implementierung des zentralen Differenzenverfahrens keine komplexen Gleichungslösungsalgorithmen erfordert, ist der Programmieraufwand vergleichsweise gering. Die Methode kann problemlos auch auf nichtlineare Probleme angewendet werden, da sich die Nichtlinearität lediglich auf die Bestimmung der inneren Kräfte bezieht, also nur auf Elementebene Einfluss auf die Simulation hat. Eine Skizze der erforderlichen Implementierungsschritte, nach der der Algorithmus in FEAP-MEKA umgesetzt wurde, ist in Abbildung 2.2 angegeben.

In kommerziellen FE-Programmen wird die Stabilität des Zeitintegrationsverfahrens durch eine Überprüfung des Energiegleichgewichts überprüft, die dann ebenfalls in jedem Zeitschritt durchgeführt wird (vgl. Abbildung 2.2 Punkt ⑥). Instabilitäten des Verfahrens werden dann insbesondere bei nichtlinearen Problemen, bei denen die Einhaltung der Bedingung (2.13) alleine nicht ausreicht, durch künstliche Erzeugung von Energie, also eine

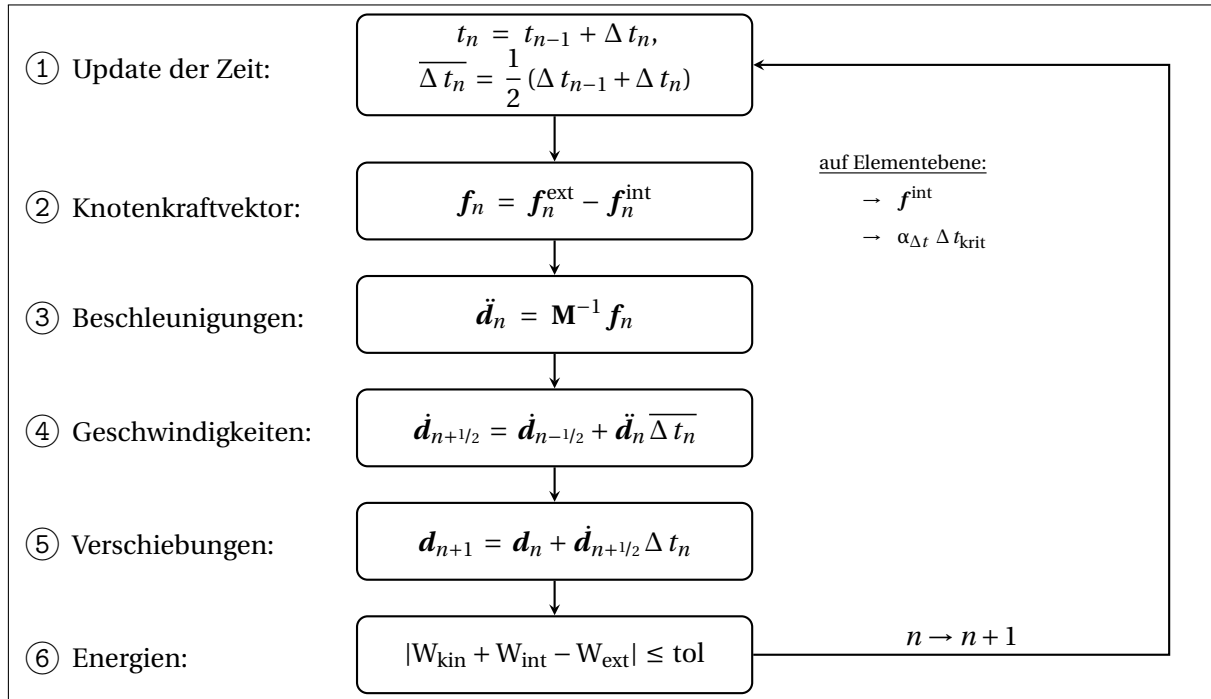


Abbildung 2.2.: Implementierung des zentralen Differenzenverfahrens (VERLET-Algorithmus [124]) für den Zeitschritt $n + 1$

Verletzung der Energieerhaltungsbedingung identifiziert. Die Vorgehensweise zur Berechnung der Energien und zur Wahl der Toleranz wird von BELYTSCHKO ET. AL. [15] ausführlich beschrieben, für die relativ einfachen Beispiele der vorliegenden Arbeit wurde auf eine solche Überprüfung verzichtet.

2.2.4. Dämpfung und dynamische Relaxation

Eine weit verbreitete Methode zur Abbildung von Dämpfungseffekten stellt, wie bereits in Abschnitt 2.1 angesprochen, das Modell der RAYLEIGH-Dämpfung dar. Es basiert auf der Annahme einer viskosen Dämpfung und wird in Gleichung (2.2) durch eine konstante Dämpfungsmatrix

$$\mathbf{C}^d = c_m \mathbf{M} + c_K \mathbf{K}_0, \quad \text{mit} \quad \mathbf{K}_0 = \left. \frac{\partial \mathbf{f}^{\text{int}}}{\partial \mathbf{d}} \right|_{\mathbf{u}=0} \quad (2.14)$$

repräsentiert, die sich aus einem masse- und einem steifigkeitsproportionalen Anteil zusammensetzt. Um den Rechenaufwand möglichst gering zu halten, wird die lineare Steifigkeitsmatrix verwendet. Die Wahl der Parameter c_m und c_K wird ausführlich z. B. bei CLOUGH & PENZIEN [34] oder MAGNUS ET. AL. [85] diskutiert.

Eine Möglichkeit, statische Lösungen mit dem zentralen Differenzenverfahren zu bestimmen stellt die *dynamische Relaxation* dar und wird u. a. auch bei HALLQUIST [49] beschrieben. Die Dämpfungsmatrix aus Gleichung (2.14) beinhaltet dann nur den masseproportionalen Anteil und wird mit einer diagonalisierten Massenmatrix bestimmt

$$\mathbf{C}^{\text{dr}} = c_{m,\text{dr}} \mathbf{M}_d. \quad (2.15)$$

Der Parameter $c_{m,dr}$ muss nun so gewählt werden, dass gerade die kritische Dämpfung erreicht wird, das dynamische System also in eine Kriechbewegung übergeht. Bei WRIGGERS [130] wird eine Möglichkeit beschrieben, den Eigenvektor zur kleinsten Eigenfrequenz im Zeitschritt $n + 1$ durch das Verschiebungsinkrement $\Delta \mathbf{d}_{n+1} = \mathbf{d}_{n+1} - \mathbf{d}_n$ zu approximieren. Die Näherung der kleinsten Eigenfrequenz und somit des gewünschten Dämpfungsparameters lautet somit

$$\omega_{n+1}^2 \approx \frac{\Delta \mathbf{d}_{n+1}^T \cdot (\mathbf{f}_n^{\text{ext}} - \mathbf{f}_n^{\text{int}})}{\Delta \mathbf{d}_{n+1}^T \cdot \mathbf{M}_d \cdot \Delta \mathbf{d}_{n+1}} \rightsquigarrow c_{m,dr,n+1} = 2 \omega_{n+1} \quad (2.16)$$

Das Verfahren ist in verschiedenen kommerziellen expliziten FE-Programmen umgesetzt, die in LS-DYNA verfügbare Implementierung basiert beispielsweise auf PAPADRAKAKIS [101].

2.2.5. Bulk Viskosität

Die Simulation von Strukturproblemen mit Stoßwellen mit dem zentralen Differenzenverfahren kann zu numerischen Problemen führen. Insbesondere Wellenlängen, die kleiner sind als eine kritische Elementlänge, können durch die Zeitdiskretisierung schlecht abgebildet werden. Eine Möglichkeit zur Verbesserung der Abbildungseigenschaften – die sogenannte ‘*bulk viscosity*’ – ist heute in allen kommerziellen Programmen implementiert, die sich auf die Simulation hoch-dynamischer Prozesse im Bereich Struktur-/ Fluidmechanik spezialisiert haben. Mangels einer passenden deutschen Übersetzung des englischen Fachbegriffes, wird die Methode im Folgenden als *Bulk Viskosität* bezeichnet. Details zur Implementierung sind in den Handbüchern der Simulationsprogramme beschrieben [42, 83, 118], in denen zum theoretischen Hintergrund im Wesentlichen auf die Arbeiten von VON NEUMANN & RICHTMYER [96], RICHTMYER & MORTON [107], sowie auf Publikationen von NOH und WILKINS verwiesen wird (siehe hierzu auch HALLQUIST [49]). Im angegebenen Schrifttum wird auch auf die genaue Definition und Beschreibung von Schockwellen eingegangen, auf die an dieser Stelle verzichtet werden soll. Eine übersichtliche Darstellung der Methode findet sich auch bei ANDERSON in [9]. Im Folgenden werden die wesentlichen Aspekte der Methode, wie sie in den kommerziellen Programmen implementiert ist und auch für die vorliegende Arbeit für FEAP-MEKA umgesetzt wurde, zusammengefasst, sowie der Effekt der Methode an einem einfachen Beispiel veranschaulicht.

Die grundsätzliche Idee der Bulk Viskosität ist die Einführung einer dissipativen Größe zur Verteilung bzw. ‘*Verschmierung*’ der Schockwelle auf mehrere Elemente. Dies erfolgt über die Berechnung eines künstlichen Drucks

$$q = \begin{cases} C_0 \rho \ell^2 \dot{\epsilon}_{kk}^2 - C_1 \rho \ell c_s \dot{\epsilon}_{kk}, & \text{wenn } \dot{\epsilon}_{kk} < 0 \\ 0 & \text{wenn } \dot{\epsilon}_{kk} \geq 0 \end{cases} \quad (2.17)$$

Hierbei ist ρ die Rohdichte, c_s die Wellenausbreitungsgeschwindigkeit im jeweiligen Medium und $\dot{\epsilon}_{kk}$ die Spur des Dehnratentensors $\dot{\epsilon}$. Es ist zu erkennen, dass der künstliche Druck nur bei Kompression berechnet wird, also wenn $\dot{\epsilon}_{kk} < 0$ ist. Die Länge ℓ repräsentiert eine charakteristische Elementabmessung, die bei Schalenelementen durch die Quadratwurzel der Elementfläche, bzw. bei Volumenelementen durch die dritte Wurzel aus dem Volumen definiert ist. Bei den in der Arbeit verwendeten Volumenschalenformulierungen wurden verschiedene Varianten für die charakteristische Länge getestet. Neben der Verallgemeinerung

der oben genannten Methoden mit

$$\ell = \max \left\{ \sqrt[3]{V}, \sqrt{A_o}, \sqrt{A_u} \right\}, \quad (2.18)$$

wobei A_o und A_u die Flächen der Schalenober- bzw. -unterseite sind, ist die Wahl einer mittleren Dicke denkbar, da sich dann die Behandlung der Schockwellen bei biegedominanten Simulation – z. B. von Impaktproblemen – insbesondere auf die Dickenrichtung konzentriert. Diese letztere Variante hat sich schließlich für Volumenschalenelemente als praktikabel erwiesen und wird somit in den Beispielen ausschließlich verwendet.

Gleichung (2.17) setzt sich aus einem in $\dot{\epsilon}_{kk}$ quadratischen und einem linearen Term zusammen, die jeweils durch einen Faktor C_0 , bzw. C_1 gesteuert werden. Nach ANDERSON [9] wurde der lineare Anteil hinzugefügt, um die Oszillationen zu glätten, die durch den quadratischen Anteil nicht vollständig gedämpft werden konnten. Die Faktoren C_0 und C_1 können in den kommerziellen Programmen frei gewählt werden, wobei z. B. in LS-DYNA die Wahl von $C_0 = C_1 = 0$ nicht zulässig ist und automatisch mit den Vorgabewerten gerechnet wird. Diese sind z. B.

$$\begin{array}{llll} C_0 = 1.50, & \text{bzw.} & C_1 = 0.06 & \text{in LS-DYNA [83]} \\ C_0 = 1.44, & \text{bzw.} & C_1 = 0.06 & \text{in ABAQUS-EXPLICIT [118]} \end{array} \quad (2.19)$$

Um den Einfluss der Bulk Viskosität in LS-DYNA dennoch auszuschalten, müssen für C_0 und C_1 sehr kleine, aber von Null verschiedene Werte gesetzt werden.

Der Einfluss der Bulk Viskosität kann am Beispiel eines Stabes mit quadratischem Querschnitt gezeigt werden, der von einer Druckwelle in Längsrichtung durchlaufen wird. Ein ähnliches Beispiel ist auch in der Arbeit von LARCHER [80] zu finden. Durch eine sprunghafte Änderung der Steifigkeit des Materials ändert sich auch die Wellenausbreitungsgeschwindigkeit sprunghaft. Die Geometrie- und Materialeigenschaften sind in Abbildung 2.3 angegeben, wobei auf die Angabe von Einheiten bewusst verzichtet wurde. In den Stab wird durch Vorgabe einer Verschiebung an den Knoten der linken Seite eine Druckwelle eingebracht – die Geschwindigkeit, mit der die Knoten verschoben werden, beträgt $v_0 = 200 \cdot 10^3$. Das Weg-Zeit Diagramm ist ebenfalls in Abbildung 2.3 angegeben.

Aus den gegebenen Eigenschaften der zwei verschiedenen Materialien, lassen sich die jeweiligen Wellenausbreitungsgeschwindigkeiten für Longitudinalwellen in elastischen Festkörpern

$$c_s = \sqrt{\frac{E(1-\nu)}{\rho(1+\nu)(1-2\nu)}} \Rightarrow c_{s,1} = 316.2 \cdot 10^3 \quad \text{bzw.} \quad c_{s,2} = 100.0 \cdot 10^3 \quad (2.20)$$

bestimmen. Durch die Steifigkeitsänderung kommt es also zu einer Reduktion der Wellenausbreitungsgeschwindigkeit an der Übergangsstelle, wodurch die tatsächliche Ausbreitungsgeschwindigkeit im Moment des Übergangs höher ist als die physikalische Wellenausbreitungsgeschwindigkeit des Materials. Es tritt an der Übergangsstelle also eine Schockwelle auf. Das vorliegende Beispiel ist so gewählt, dass bei der Simulation ohne Bulk Viskosität keine numerische Schwierigkeiten auftreten, der Effekt der Methode jedoch verdeutlicht werden kann. Für das Beispiel wurde eine feste Zeitschrittweite von $\Delta t = 1.00 \cdot 10^{-8}$ gewählt, die vorgegebene Verschiebung wird also über 125 Zeitschritte aufgebracht und dann für den Rest der Simulation konstant gehalten.

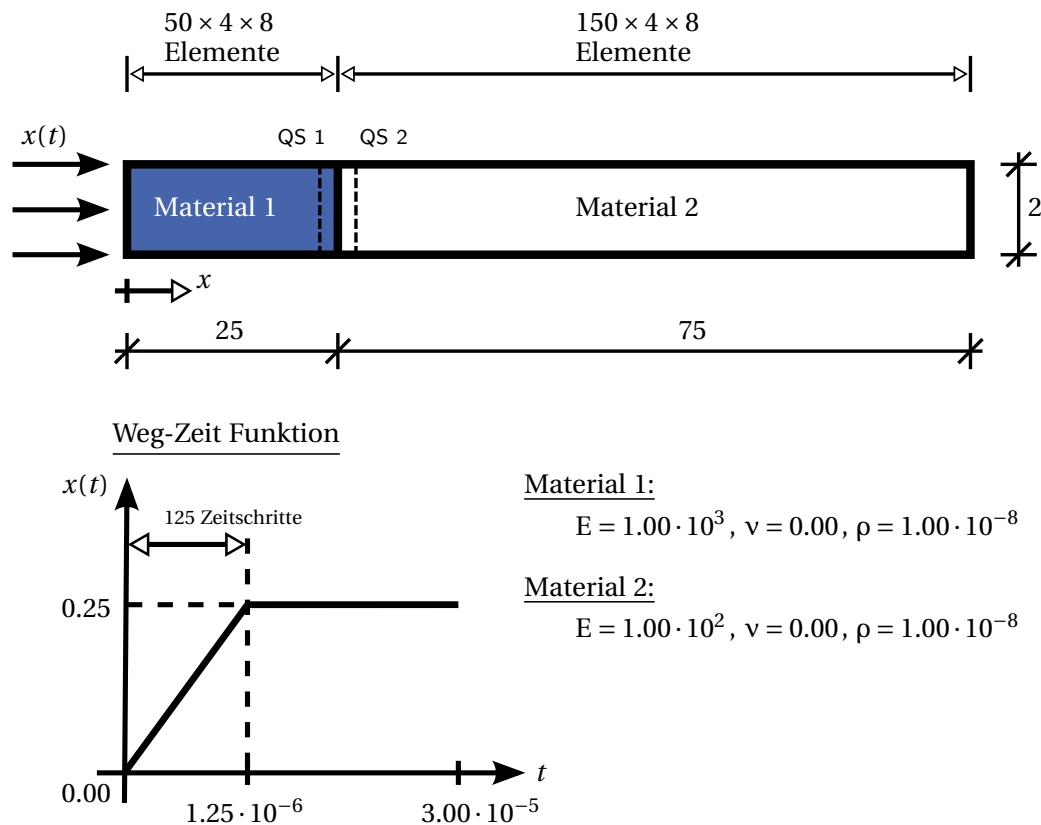


Abbildung 2.3.: Beispiel für Bulk-Viskosität – Stab mit Longitudinalwelle

In Abbildung 2.4 sind die resultierenden Schnittkräfte in den in Abbildung 2.3 angegebenen Querschnitten QS 1 und QS 2 über die Berechnungszeit aufgetragen. Die Simulation wurde ohne Bulk-Viskosität sowie mit den in Gleichung (2.19) angegebenen Standard-Werten aus LS-DYNA durchgeführt. Des weiteren wurde jeweils nur einer der beiden Standard-Werte gewählt und der jeweils andere zu Null gesetzt, um den Einfluss der beiden Anteile zu untersuchen. Abbildung 2.4a zeigt die Schnittkraft im Querschnitt direkt vor dem Übergang zu Material 2, die Welle und ihre Reflexion an der Übergangsstelle treten also unmittelbar nacheinander auf, was zu deutlichen Oszillationen führt. Der quadratische Parameter C_0 vermindert hier überwiegend die Amplitude der Kraft, wobei der lineare Term C_1 eine Dämpfung bewirkt, die zu einem Abklingen der Oszillationen führt. Dieses Verhalten kann insbesondere im Ausschnitt in Abbildung 2.4b abgelesen werden. Im zweiten Querschnitt dessen Schnittkraftverlauf in Abbildung 2.4c angegeben ist, kommt zunächst nur die Welle selbst an, bzw. der Teil des Signals, der nicht an der Übergangsstelle reflektiert wird. Kurz vor QS 1 erfolgt die sprunghafte Reduktion der Wellenausbreitungsgeschwindigkeit, die bei deutlicheren Steifigkeitsunterschieden zu erheblichen numerischen Problemen führen kann. Die Bulk-Viskosität führt hier zu einer flacheren und etwas breiteren Kurve, die als 'Verschmierung' der Welle über mehrere Elemente interpretiert werden kann. Beide Parameter C_0 und C_1 haben hier etwa den gleichen Einfluss. Im weiteren Verlauf der Simulation erreicht die Reflexion der Welle ein weiteres mal die Stelle QS 2, wobei dann wieder eine Dämpfung der Oszillationen durch C_1 zu erkennen ist. Es sei angemerkt, dass bei Simulationen ohne sprunghafte Änderung der Steifigkeit sowie bei kleinen Dehnraten kein Einfluss des qua-

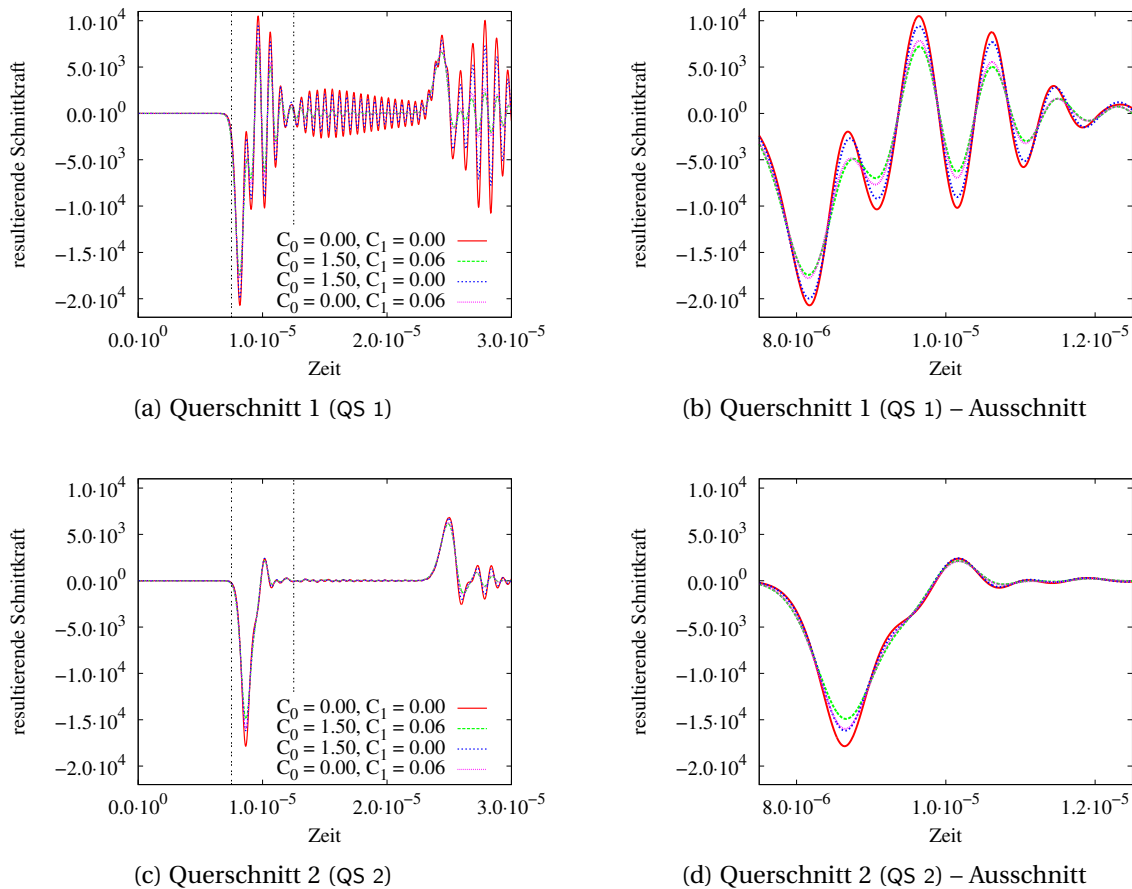


Abbildung 2.4.: Auswertung der resultierenden Schnittkraft in zwei Querschnitten (QS 1 und QS 2 – siehe Abbildung 2.3) für verschiedene Parameter C_0 und C_1

dratischen Anteils erkennbar ist. Da in diesem durch C_0 gesteuerten Term die Dehnrate $|\dot{\epsilon}|$ quadratisch eingeht, ist er nur im unmittelbaren Bereich von Schockwellen von Bedeutung.

2.2.6. Massenskalierung

Die Effizienz einer Simulation mit dem zentralen Differenzenverfahren wird im Wesentlichen von der Größe des kritischen Zeitschrittes bestimmt, die durch das in Abschnitt 2.2.2 eingeführte COURANT-Kriterium aus Gleichung (2.13) begrenzt ist. Die Größe des kritischen Zeitschrittes wird dabei von dem Element bestimmt, bei welchem das Verhältnis aus Steifigkeit und Masse besonders ungünstig ist. Liegt also eine Diskretisierung vor, bei der einige wenige Elemente deutlich kleiner bzw. steifer sind als die anderen, so kann die Simulation nur mit einem sehr kleinen Zeitschritt durchgeführt werden, es sind also insgesamt mehr Zeitschritte erforderlich. Für Anwendungen, bei denen die dynamischen Eigenschaften des Systems von untergeordnetem Interesse sind, wird z. B. bei BELYTSCHKO ET AL. [15] die Möglichkeit diskutiert, die Masse solcher Elemente zu erhöhen, um einen größeren Wert für den kritischen Zeitschritt zu erreichen. Die statischen Eigenschaften, die etwa bei Anwendungen etwa in der Metallumformung wichtig sind, werden davon – im Gegensatz zu Veränderungen an den Steifigkeiten – kaum beeinflusst und eine Simulation kann in deutlich kürzerer Zeit

durchgeführt werden. Für Probleme, bei denen dynamische Effekte wichtiger sind, ist eine solche Skalierung der Elementmasse nicht empfehlenswert.

Für verschiebungsbasierten Elementformulierungen wird in der vorliegenden Arbeit keine Massenskalierung verwendet, weshalb nicht weiter auf diese Option eingegangen wird. Es sei jedoch für die Anwendung einer selektiven Massenskalierung zusammen mit einer Verbesserung der dynamischen Eigenschaften auf die Arbeiten von OLOVSSON ET. AL. [99, 100] und OLOVSSON & SIMONSSON [98] verwiesen. Die dort veröffentlichte Strategie ist mittlerweile auch in der kommerziellen Anwendung LS-DYNA verfügbar. In Abschnitt 3.4.5 wird im Zusammenhang mit der *Methode der inkompatiblen Verschiebungen* eine unterschiedliche Behandlung und somit selektive Skalierung einzelner Elementknotenmassen diskutiert, die ebenfalls der Steuerung der kritischen Zeitschrittweite dient.

2.3. Implizite Zeitintegration

2.3.1. Das NEWMARK-Verfahren

Im Folgenden wird das NEWMARK-Verfahren als das im Bereich der Strukturmechanik am weitesten verbreitete implizite Zeitintegrationsverfahren kurz beschrieben. Es wird in vielen FE-Programmen verwendet, so auch im institutseigenen Code FEAP-MEKA [112]. In der vorliegenden Arbeit wird nicht mit dem NEWMARK-Verfahren gearbeitet, es sollen hier aber die Beziehungen kurz zusammengefasst werden, um die Unterschiede zur expliziten Zeitintegration herauszuarbeiten. Auf detaillierte Ausführungen wird hier verzichtet, für vertiefende Informationen siehe das eingangs zitierte Schrifttum.

Für den Zeitschritt $n + 1$ gelten die folgenden Berechnungsvorschriften für die Verschiebungen und Geschwindigkeiten

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta t_n \dot{\mathbf{d}}_n + \frac{\Delta t_n^2}{2} (1 - \beta) \ddot{\mathbf{d}}_n + \beta (\Delta t_n)^2 \ddot{\mathbf{d}}_{n+1} \quad (2.21)$$

$$\dot{\mathbf{d}}_{n+1} = \dot{\mathbf{d}}_n + (1 - \gamma) \Delta t_n \ddot{\mathbf{d}}_n + \gamma \Delta t_n \ddot{\mathbf{d}}_{n+1} \quad (2.22)$$

mit den beiden NEWMARK-Parametern β und γ , auf deren Wahl in Abschnitt 2.3.2 eingegangen wird. Es ist direkt zu erkennen, dass in den Gleichungen (2.21) und (2.22) sowohl bekannte (Zeitpunkt t_n) als auch unbekannte (Zeitpunkt t_{n+1}) kinematischen Größen auftreten. Durch Auflösen von Gleichung (2.21) nach den Beschleunigungen und Einsetzen in Gleichung (2.22), ergeben sich die Beziehungen

$$\ddot{\mathbf{d}}_{n+1} = \frac{1}{\beta \Delta t_n^2} (\mathbf{d}_{n+1} - \mathbf{d}_n) - \frac{1}{\beta \Delta t_n} \dot{\mathbf{d}}_n - \frac{1 - 2\beta}{2\beta} \ddot{\mathbf{d}}_n \quad (2.23)$$

$$\dot{\mathbf{d}}_{n+1} = \frac{\gamma}{\beta \Delta t_n} (\mathbf{d}_{n+1} - \mathbf{d}_n) + \left(1 - \frac{\gamma}{\beta}\right) \dot{\mathbf{d}}_n + \left(1 - \frac{\gamma}{2\beta}\right) \Delta t_n \ddot{\mathbf{d}}_n. \quad (2.24)$$

Einsetzen in die diskretisierte Bewegungsgleichung (2.3) liefert ein im Allgemeinen nichtli-

neares Gleichungssystem für die unbekanntten Verschiebungen \mathbf{d}_{n+1}

$$\begin{aligned} \mathcal{G}(\mathbf{d}_{n+1}) := & \mathbf{M} \left[\frac{1}{\beta \Delta t_n^2} (\mathbf{d}_{n+1} - \mathbf{d}_n) - \frac{1}{\beta \Delta t_n} \dot{\mathbf{d}}_n - \frac{1-2\beta}{2\beta} \ddot{\mathbf{d}}_n \right] \\ & + \mathbf{C}^d \left[\frac{\gamma}{\beta \Delta t_n} (\mathbf{d}_{n+1} - \mathbf{d}_n) + \left(1 - \frac{\gamma}{\beta}\right) \dot{\mathbf{d}}_n + \left(1 - \frac{\gamma}{2\beta}\right) \Delta t_n \ddot{\mathbf{d}}_n \right] \\ & + \mathbf{f}_{n+1}^{\text{int}}(\mathbf{d}_{n+1}) - \mathbf{f}_{n+1}^{\text{ext}} = \mathbf{0}, \end{aligned} \quad (2.25)$$

welches in jedem Zeitschritt für die Verschiebungen z. B. mit einem NEWTON-Verfahren gelöst werden muss. Hierzu wird Gleichung (2.25) in eine Taylor-Reihe entwickelt, die nach dem linearen Glied abgebrochen wird. Es ergibt sich die Iterationsvorschrift $i \rightarrow i+1$

$$\begin{aligned} \left(\frac{1}{\beta \Delta t_n^2} \mathbf{M} + \frac{\gamma}{\beta \Delta t_n} \mathbf{C}^d + \frac{\partial \mathbf{f}_{n+1}^{\text{int}}}{\partial \mathbf{d}_{n+1}} \bigg|_{\mathbf{d}_{n+1}^i} \right) \cdot \Delta \mathbf{d}_{n+1}^{i+1} = -\mathcal{G}(\mathbf{d}_{n+1}^i) \\ \mathbf{d}_{n+1}^{i+1} = \mathbf{d}_{n+1}^i + \Delta \mathbf{d}_{n+1}^{i+1}. \end{aligned} \quad (2.26)$$

Die Größe

$$\mathbf{K}_{n+1,i}^{\text{tang}} = \frac{\partial \mathbf{f}_{n+1}^{\text{int}}}{\partial \mathbf{d}} \bigg|_{\mathbf{d}_{n+1}^i} = \frac{\partial^2 \Pi_{n+1}^{\text{int}}}{\partial \mathbf{d}^2} \bigg|_{\mathbf{d}_{n+1}^i} \quad (2.27)$$

wird als *tangentielle Steifigkeitsmatrix* bezeichnet. Die Iteration wird bis zum Erreichen eines Konvergenzkriteriums fortgesetzt. Durch Einsetzen in die Gleichungen (2.23) und (2.24) erhält man schließlich die Geschwindigkeiten und Beschleunigungen und das Verfahren kann mit dem nächsten Zeitschritt fortgesetzt werden. Als Startwert für den nächsten Zeitschritt wird die konvergierte Verschiebung verwendet. Das Gleichungssystem aus Gleichung (2.26) mit den symmetrischen Matrizen \mathbf{M} , \mathbf{C}^d und $\mathbf{K}_{n+1,i}^{\text{tang}}$ muss in jedem Iterationsschritt gelöst werden, was insbesondere bei starken Nichtlinearitäten zu einem hohen Rechenaufwand pro Zeitschritt führt.

2.3.2. Wahl der Parameter

Die Wahl der NEWMARK-Parameter β und γ ist von besonderer Bedeutung für die Genauigkeit und die Stabilität des Zeitintegrationsverfahrens. Die sogenannte Trapezregel erhält man für $\beta = 0.25$ und $\gamma = 0.5$. Diese ist im linearen Fall unbedingt stabil – der Zeitschritt kann also im Gegensatz zum zentralen Differenzenverfahren beliebig groß sein. Im ungedämpften Fall ($\mathbf{C}^d = \mathbf{0}$) ist die Trapezregel ebenso wie das zentrale Differenzenverfahren energieerhaltend und von zweiter Ordnung genau ($\mathcal{O}(\Delta t^2)$). Andere Kombinationen der Parameter β und γ führen zu numerischer Dämpfung oder Anfachung und können Verluste bei der Genauigkeit und der numerischen Stabilität zur Folge haben. Bemerkenswert ist, dass sich für $\beta = 0.0$ und $\gamma = 0.5$ gerade wieder das zentrale Differenzenverfahren, jedoch mit einer anderen Startprozedur ergibt. Die in Abschnitt 2.2 beschriebenen Kriterien – etwa für die Wahl des kritischen Zeitschritts – sind dann identisch.

2.4. Effizienzaspekte expliziter und impliziter Verfahren

Ein wesentlicher Unterschied zwischen den vorgestellten Zeitintegrationsverfahren hinsichtlich ihrer Anwendung für dynamische Probleme ist der *numerische Aufwand pro Zeitschritt*.

Das zentrale Differenzenverfahren zeichnet sich durch sehr effiziente Rechenoperationen auf globaler Ebene aus, da wegen der Verwendung diagonalisierter Massenmatrizen lediglich Vektoroperationen durchzuführen sind. Das NEWMARK-Verfahren hingegen erfordert die Lösung eines nichtlinearen Gleichungssystems zu jedem Zeitschritt, was je nach Komplexität der Struktur sehr zeit- und speicherintensiv sein kann. Aufwändig zu implementierende moderne Gleichungslösungsalgorithmen – eine gut aufbereitete Übersicht mit entsprechenden Quellenangaben ist beispielsweise [130] zu entnehmen – steigern die Effizienz des Verfahrens, dennoch sind deutlich mehr Rechenoperationen pro Zeitschritt erforderlich.

Die *Wahl der Zeitschrittweite* ist der zweite Punkt, der die Wahl des Zeitintegrationsverfahrens beeinflusst. Beim NEWMARK-Verfahren kann – bei entsprechender Wahl der Parameter β und γ der Zeitschritt beliebig groß gewählt werden, ohne dass die numerische Stabilität des Algorithmus beeinflusst wird. Dagegen ist die Zeitschrittweite beim zentralen Differenzenverfahren beschränkt und muss entsprechend der Elementgeometrien angepasst werden, um eine numerisch stabile Lösung zu erhalten.

Traditionell führen diese beiden Kriterien zu jeweils speziellen Anwendungsgebieten für die beiden Verfahren. Dominieren kurzzeitdynamische Effekte die Struktur, wie etwa bei der Simulation von Wellenausbreitungsproblemen, Impakt oder Crash-Simulationen, so müssen schon aus physikalischen Gründen kleine Zeitschritte gewählt werden, um zeitlich lokale Effekte abbilden zu können. In diesem Fall ‘stört’ die Limitierung der Zeitschrittweite bei zentralen Differenzen nicht und man profitiert deutlich von der hohen Effizienz des Verfahrens. Langzeitdynamische Probleme ohne plötzliche Einwirkungen hingegen erlauben auch größere Zeitschrittweiten, weshalb hier häufig implizite Verfahren angewendet werden. Es können jedoch für alle strukturmechanischen Anwendungen beide Möglichkeiten der Zeitintegration angewendet werden, so werden z. B. auch quasi-statische Effekte durch die Einführung einer kritischen Dämpfung (*dynamische Relaxation*) mit zentralen Differenzen simuliert, was etwa für die Aufbringung von Gravitationslasten angewendet wird. Denkbar und in der Praxis verbreitet ist auch eine Kombination beider Verfahren, bei denen innerhalb einer Simulation zwischen impliziten und expliziten Lösungsverfahren umgeschaltet werden kann.

Der geringe numerische Aufwand des globalen Lösungsalgorithmus führt zu einer Dominanz der Operationen auf Elementebene beim zentralen Differenzenverfahren. Insbesondere verschiedene Verfahren zur Beseitigung künstlicher Versteifungseffekte, wie sie in Kapitel 3 beschrieben werden, erfordern numerisch aufwändige Modifikationen auf Elementebene. Diese machen sich hinsichtlich der gesamten Simulationszeit innerhalb eines impliziten Algorithmus weitaus weniger bemerkbar, da hier die globale Gleichungslösung dominiert. Die Anwendung von Elementformulierungen bei expliziter Zeitintegration erfordert also eine kritische Bewertung der bekannten Verfahren zur Vermeidung von Versteifungseffekten hinsichtlich ihrer effizienten Umsetzbarkeit. Dieser zentrale Aspekt der vorliegenden Arbeit wird in den folgenden Abschnitten diskutiert.

3. Finite Elemente für Schalenprobleme

Im folgenden Kapitel werden die Finite-Elemente-Formulierungen zur Simulation von Schalenstrukturen beschrieben, welche im Rahmen der Arbeit implementiert wurden. Im ersten Abschnitt wird kurz das Degenerationskonzept und die verwendete 5-Parameter-Schalenformulierung vorgestellt, welche u. a. auf Arbeit von AHMAD ET. AL. [1] zurückgeht. Heute können die wesentlichen Zusammenhänge praktisch allen Standardwerken zur Finite-Elemente-Methode entnommen werden [12, 15, 35, 58, 134], einen sehr umfassenden Überblick über die verschiedenen Theorien gibt auch die Arbeit von BISCHOFF ET. AL. [25]. Es folgt eine Einführung in das 'Solid-Shell'-Konzept, welches zu den nur auf Verschiebungsfreiheitsgraden basierenden Volumenschalen- oder Solid-Shell-Elementen führt. Die Grundlagen dieser Elementfamilie basieren auf den Arbeiten von PARISCH [102], MIEHE [95], HAUPTMANN & SCHWEIZERHOF [55], HAUPTMANN ET. AL. [56] und SZE & YAO [120].

Von besonderer Bedeutung innerhalb des expliziten Zeitintegrationsverfahrens ist die Formulierung einer diagonalisierten Massenmatrix. Verschiedene Strategien zu deren Aufstellung, die dem Schrifttum entnommen wurden [58, 136], werden im folgenden Abschnitt angegeben. Bei verschiebungsbasierten finiten Elementen auftretende unerwünschte Versteifungseffekte werden klassifiziert und verschiedene Konzepte zusammengefasst, die sich mit der Reduzierung bzw. Beseitigung solcher Versteifungen befassen. Die verwendeten Methoden ANS [13, 29] und EAS [113, 114] sind aus dem Schrifttum bekannt und wurden bereits für Schalen- und Volumenschalenelemente eingesetzt. Alternativ werden zwei auf einer Erweiterung der Verschiebungsinterpolation basierende Strategien vorgeschlagen und deren Umsetzung insbesondere bei expliziter Zeitintegration diskutiert. Eine Zusammenstellung der für die vorliegende Arbeit umgesetzten Konzepte wird hier angegeben, um im weiteren Verlauf auf die Besonderheiten bei der Verwendung symbolischer Programmierung im Zusammenhang mit expliziter Zeitintegration einzugehen und Aspekte der Effizienz der verschiedenen Methoden diskutieren zu können.

Abschließend wird kurz eine im Rahmen der Arbeit verwendete Formulierung zur Beschreibung von Kontakt ereignissen skizziert, die auf der Arbeit von KONYUKHOV [64] beruht.

3.1. Degenerierte Schalenelemente

3.1.1. Das Degenerationskonzept

Zur Simulation von Schalenstrukturen mit der Methode der finiten Elemente hat sich das Degenerationskonzept in den Anwendungen durchgesetzt. Hierbei werden mechanische Annahmen formuliert, die insbesondere für Schalenstrukturen gültig sind, also Strukturen deren Dicke deutlich kleiner ist als die beiden anderen Abmessungen. Das Konzept basiert

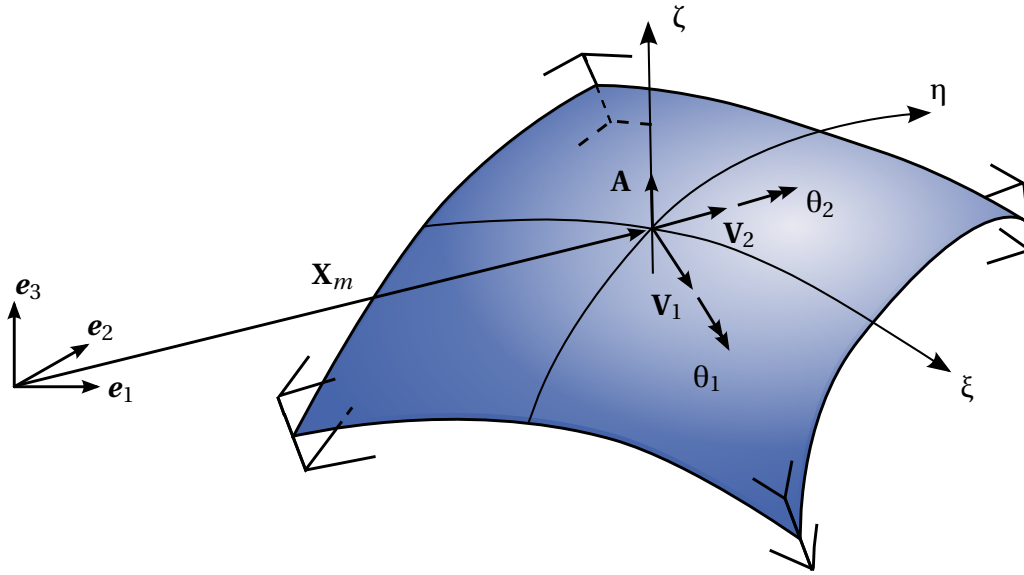


Abbildung 3.1.: Beschreibung der Schengeometrie nach dem Degenerationskonzept

auf der Beschreibung der Geometrie über deren Mittelfläche, also einer *Degeneration* des Kontinuums. Die getroffenen Annahmen lauten im Einzelnen:

- Die Normale zur Mittelfläche der Ausgangskonfiguration bleibt gerade.
- Die Länge der Normale zur Mittelfläche ändert sich nicht.
- Es treten keine Normalspannungen in Dickenrichtung auf.

Die ersten beiden Punkte – die kinematischen Annahmen – führen auf die Beschreibung der Geometrie in der Ausgangskonfiguration gemäß Abbildung 3.1

$$\mathbf{X}(\xi, \eta, \zeta) = \mathbf{X}_m(\xi, \eta) + \frac{1}{2} \zeta t(\xi, \eta) \mathbf{A}(\xi, \eta), \quad (3.1)$$

mit dem Schalendirektor \mathbf{A} , dem Ortsvektor der Mittelebene \mathbf{X}_m , und der über die Schalenebene veränderlichen Schalendicke $t(\xi, \eta)$. Entsprechend ergibt sich für die Momentankonfiguration

$$\mathbf{x}(\xi, \eta, \zeta) = \mathbf{x}_m(\xi, \eta) + \frac{1}{2} \zeta t(\xi, \eta) \mathbf{a}(\xi, \eta) \quad (3.2)$$

und somit für die Verschiebung

$$\mathbf{u}(\xi, \eta, \zeta) = \mathbf{x}(\xi, \eta, \zeta) - \mathbf{X}(\xi, \eta, \zeta) = \mathbf{u}_m(\xi, \eta) + \frac{1}{2} \zeta t(\xi, \eta) (\mathbf{a}(\xi, \eta) - \mathbf{A}(\xi, \eta)). \quad (3.3)$$

Punkte, die sich in der Ausgangskonfiguration auf dem Direktor \mathbf{A} befinden, müssen nach der ersten kinematischen Annahme (*Geradebleiben der Normalen*) in der Momentankonfiguration auf dem verformten Direktor \mathbf{a} liegen, welcher jedoch nicht mehr senkrecht auf der Mittelfläche stehen muss. Man spricht hierbei von der REISSNER-MINDLIN Schalentheorie, die im Gegensatz zur KIRCHHOFF-LOVE-Theorie Schubverzerrungen erlaubt. Die beschreibenden natürlichen Koordinaten ξ und η liegen in der Schalenebene, ζ steht dazu senkrecht.

Eine wesentliche Schwierigkeit bei der Anwendung der oben beschriebenen Zusammenhänge, insbesondere bei Problemen mit großen Deformationen ist die Bestimmung des

Direktors in der deformierten Konfiguration \mathbf{a} . In der vorliegenden Arbeit wird eine Beschreibung verwendet, die auf die Arbeit von ARGYRIS [11] zurückgeht. Auf eine Herleitung der Zusammenhänge, die in den Dissertationen von GEBHARDT [46] und HAUPTMANN [54] ausführlich und übersichtlich aufbereitet wurden, wird an dieser Stelle verzichtet. Durch Einführung eines lokalen kartesischen Koordinatensystems mit den in Abbildung 3.1 angegebenen Basisvektoren \mathbf{V}_1 , \mathbf{V}_2 und \mathbf{A} wird der deformierte Direktor über die Beziehung

$$\mathbf{a} = \frac{\sin \varphi}{\varphi} \theta_2 \mathbf{V}_1 - \frac{\sin \varphi}{\varphi} \theta_1 \mathbf{V}_2 + \cos \varphi \mathbf{A}, \quad \text{mit} \quad \varphi = \sqrt{\theta_1^2 + \theta_2^2} \quad (3.4)$$

bestimmt, die wegen

$$\lim_{\varphi \rightarrow 0} \frac{\sin \varphi}{\varphi} = 1 \quad (3.5)$$

singularitätenfrei ist. Für kleine Rotationen lässt sich aus Gleichung (3.4) direkt die Darstellung

$$|\varphi| \ll 1 \Rightarrow \sin \varphi \approx \varphi \quad \text{und} \quad \cos \varphi \approx 1 \Rightarrow \mathbf{a} = \theta_2 \mathbf{V}_1 - \theta_1 \mathbf{V}_2 + \mathbf{A} \quad (3.6)$$

angeben.

Die dritte Annahme des Degenerationskonzeptes lässt sich erfüllen, in dem die Materialformulierung in lokalen Koordinaten ξ , η und ζ formuliert ist, deren ζ -Achse stets senkrecht auf der Mittelfläche steht. Es wird dann bei der Berechnung des Spannungszustandes die Bedingung $S_{\zeta\zeta} = 0$ eingeführt und nach der Verzerrung $E_{\zeta\zeta}$ aufgelöst. Dies ist bei einfachen Materialgesetzen analytisch möglich, bei komplexeren nichtlinearen Materialformulierungen ist eine iterative Erfüllung dieser Bedingung erforderlich.

3.1.2. Diskretisierung mit degenerierten Schalenelementen

Interpolation von Geometrie und Verschiebungen

Die Diskretisierung mit degenerierten Schalenelementen erfolgt durch die Einführung von Ansatzfunktionen, mit denen die diskreten Knotenwerte der Geometrie und der Verschiebungen über das Element interpoliert werden. Gemäß dem isoparametrischen Konzept werden zur Interpolation von Geometrie und Verschiebungen die gleichen Formfunktionen

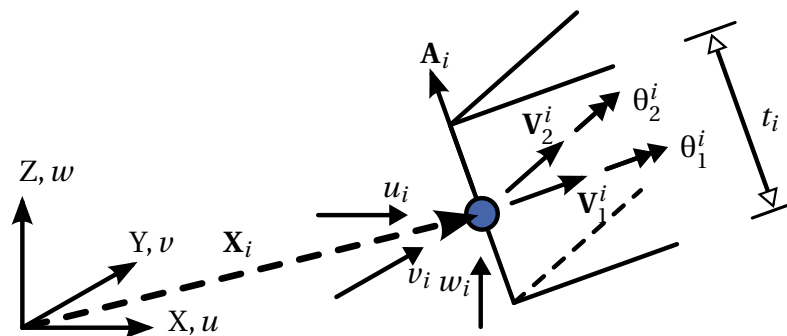


Abbildung 3.2.: Freiheitsgrade am Knoten i eines degenerierten Schalenelements

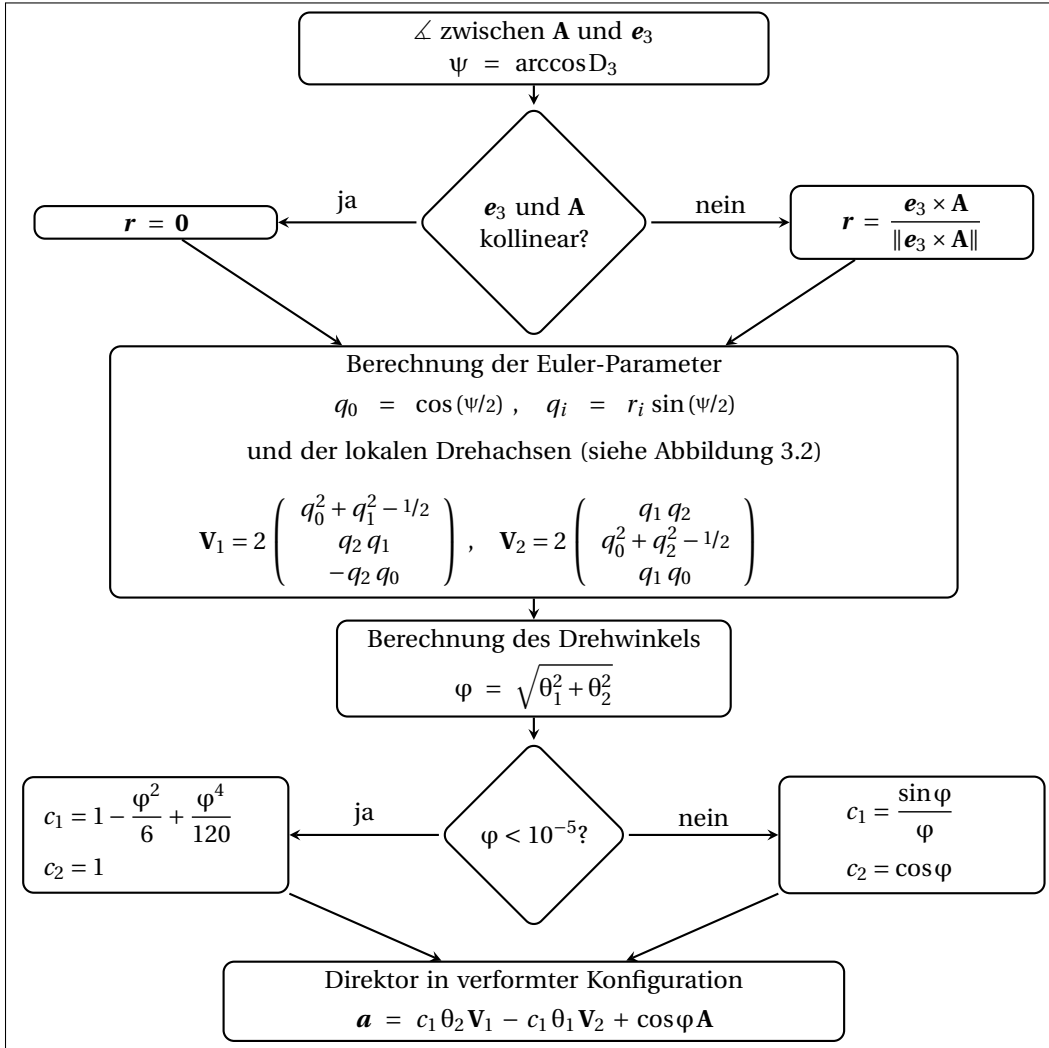


Abbildung 3.3.: Implementierungsschema zur Bestimmung des Schalendirektors in der Momentankonfiguration für einen Elementknoten

verwendet und es ergibt sich für eine Elementformulierung mit n Knoten

$$\mathbf{X}^h(\xi, \eta, \zeta) = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \sum_{i=1}^n N_i(\xi, \eta) \mathbf{X}_i + \frac{1}{2} \zeta t_i N_i(\xi, \eta) \mathbf{A}_i \quad (3.7)$$

für Gleichung (3.1) und entsprechend

$$\mathbf{u}^h(\xi, \eta, \zeta) = \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \sum_{i=1}^n N_i(\xi, \eta) \bar{\mathbf{d}}_i + \frac{1}{2} \zeta t_i N_i(\xi, \eta) (\mathbf{a}_i - \mathbf{A}_i) \quad (3.8)$$

für Gleichung (3.3). Größen mit Index i beziehen sich hierbei auf den i -ten Elementknoten, mit den in Abbildung 3.2 gegebenen Translations- und Rotationsfreiheitsgraden. Der Vektor \mathbf{X}_i beschreibt die geometrische Lage des Elementknotens auf der Mittelebene, \mathbf{A}_i ist der Schalendirektor an diesem Knoten. Der Vektor $\bar{\mathbf{d}}_i = (u_i \ v_i \ w_i)^T$ beinhaltet hier nur die Verschiebungsfreiheitsgrade; die Rotationen werden durch den Direktor in der Momentankonfiguration \mathbf{a}_i beschrieben. Eine Skizze der Implementierung zu dessen Berechnung ist

in Abbildung 3.3 angegeben; weitere Details zur programmtechnischen Umsetzung werden in Abschnitt 4.4.2 angegeben.

In der vorliegenden Arbeit wurden zur Geometrie- und Verschiebungsinterpolation lineare und quadratische LAGRANGE-Polynome verwendet. Die Verwendung dieser Ansatzordnung führt dann zu einer 4-Knoten, bzw. 9-Knoten Schalenformulierung. Die Ansatzfunktionen N_i lauten dann

$$N_i(\xi, \eta) = \frac{1}{2} (1 + \xi \bar{\xi}_i) \frac{1}{2} (1 + \eta \bar{\eta}_i), \quad i = 1 \dots 4 \quad (3.9)$$

bzw.

$$N_i(\xi, \eta) = \left(\frac{1}{2} \xi \bar{\xi}_i (1 + \xi \bar{\xi}_i) + (1 - \xi^2) (1 - \bar{\xi}_i^2) \right) \left(\frac{1}{2} \eta \bar{\eta}_i (1 + \eta \bar{\eta}_i) + (1 - \eta^2) (1 - \bar{\eta}_i^2) \right), \quad i = 1 \dots 9 \quad (3.10)$$

Die Lage der Knoten im lokalen Koordinatensystem am Einheitsselement $\bar{\xi}_i$ und $\bar{\eta}_i$ ist jeweils in den Abbildungen 3.4a und 3.4b angegeben. Einige Untersuchungen wurden auch mit quadratischen *Serendipity*-Ansätzen gemacht, deren Verwendung auf ein 8-Knoten Schalen-element mit den lokalen Knotenkoordinaten aus Abbildung 3.4c führt. Die Ansatzfunktionen für letzteres lauten dann

$$N_i(\xi, \eta) = \begin{cases} \frac{1}{4} (1 + \xi \bar{\xi}_i) (1 + \eta \bar{\eta}_i) (\xi \bar{\xi}_i + \eta \bar{\eta}_i - 1), & i = 1 \dots 4 \\ \frac{1}{2} (1 - \xi^2) (1 + \eta \bar{\eta}_i), & i = 5, 7 \\ \frac{1}{2} (1 - \eta^2) (1 + \xi \bar{\xi}_i), & i = 6, 8 \end{cases} \quad (3.11)$$

Konvektive Basisvektoren

Wie schon in Abschnitt 3.1.1 angesprochen, werden die Materialbeziehungen in lokalen Koordinaten formuliert, um eine getrennte Betrachtung der Spannungen in Dickenrichtung

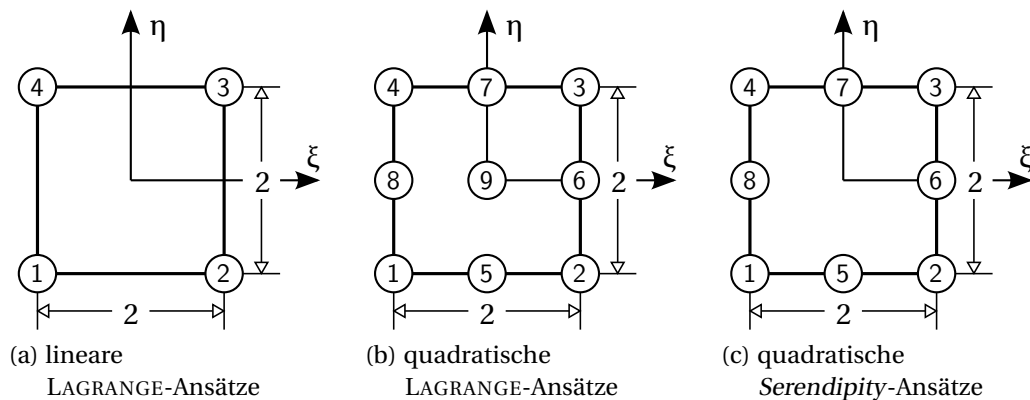


Abbildung 3.4.: Lage und Nummerierung der Knoten im Einheitsselement für unterschiedliche Ansatzordnungen

von solchen in der Schalenebene zu ermöglichen. Es wird im betrachteten Punkt ein konvektives Koordinatensystem eingeführt, dessen Basisvektoren tangential an die natürlichen Koordinaten ξ , η und ζ liegen. Die kovarianten Basisvektoren in der Referenzkonfiguration sind als Ableitungen des Ortsvektors \mathbf{X} definiert

$$\mathbf{G}_i = \frac{\partial \mathbf{X}}{\partial \xi^i}, \quad \text{mit } \boldsymbol{\xi} = \{\xi^1, \xi^2, \xi^3\} = \{\xi, \eta, \zeta\}. \quad (3.12)$$

Entsprechend gilt für die Momentankonfiguration

$$\mathbf{g}_i = \frac{\partial \mathbf{x}}{\partial \xi^i} = \frac{\partial (\mathbf{X} + \mathbf{u})}{\partial \xi^i}. \quad (3.13)$$

Da die ko- und kontravarianten Basisvektoren – letztere werden mit einem oben stehenden Index bezeichnet – paarweise senkrecht aufeinander stehen, also

$$\mathbf{G}_i \cdot \mathbf{G}^j = \delta_i^j \quad (3.14)$$

gilt, lassen sie sich über die Beziehungen

$$\mathbf{G}_i = G_{ij} \mathbf{G}^j, \quad \text{mit} \quad G_{ij} = \mathbf{G}_i \cdot \mathbf{G}_j \quad (3.15)$$

$$\mathbf{G}^i = G^{ij} \mathbf{G}_j, \quad \text{mit} \quad [G^{ij}] = [G_{ij}]^{-1} \quad (3.16)$$

ineinander umrechnen. Aus einer zeilenweisen Anordnung der kovarianten Basisvektoren ergibt sich die JACOBI-Matrix der isoparametrischen Abbildung

$$\mathbf{J} = \left(\frac{\partial \mathbf{X}}{\partial \boldsymbol{\xi}} \right)^T = \begin{pmatrix} - & \mathbf{G}_1 & - \\ - & \mathbf{G}_2 & - \\ - & \mathbf{G}_3 & - \end{pmatrix} \quad (3.17)$$

und es folgt mit Gleichung (3.14)

$$\mathbf{J} \mathbf{J}^{-1} = \mathbf{1} \quad \Rightarrow \quad \mathbf{J}^{-1} = \left(\frac{\partial \mathbf{X}}{\partial \boldsymbol{\xi}} \right)^{-T} = \begin{pmatrix} | & | & | \\ \mathbf{G}^1 & \mathbf{G}^2 & \mathbf{G}^3 \\ | & | & | \end{pmatrix}. \quad (3.18)$$

Die Komponenten der ko- und kontravarianten Metrikensoren aus Gleichung (3.15) lassen sich schließlich auch über die Matrixprodukte

$$[G_{ij}] = \mathbf{J} \cdot \mathbf{J}^T \quad \text{und} \quad [G^{ij}] = \mathbf{J}^{-T} \cdot \mathbf{J}^{-1} \quad (3.19)$$

berechnen.

Verzerrungen und Spannungen in konvektiven Koordinaten

Die in Abschnitt 1.1 eingeführten Verzerrungsmaße lassen sich nun in den definierten konvektiven Koordinaten berechnen, um daraus dann den Zusammenhang zwischen Spannungen und Verzerrungen ebenfalls in der konvektiven Basis zu beschreiben. Um den GREEN-LAGRANGE-Verzerrungstensor aus Gleichung (1.12) in konvektiven Koordinaten darzustellen, kann zunächst der Deformationsgradient \mathbf{F} aus Gleichung (1.3) in einer gemischten Basis

$\mathbf{g}_i \otimes \mathbf{G}^i$ angegeben werden. Aus Gleichung (3.13) folgt mit den Gleichungen (3.12) und (1.3) die sog. *push-forward* Operation

$$\mathbf{g}_i = \frac{\partial \mathbf{x}}{\partial \xi^i} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} \cdot \frac{\partial \mathbf{X}}{\partial \xi^i} = \mathbf{F} \cdot \mathbf{G}_i. \quad (3.20)$$

Auflösen nach \mathbf{F} liefert

$$\mathbf{F} = \mathbf{g}_i \otimes \mathbf{G}^i \quad \text{und entsprechend} \quad \mathbf{F}^T = \mathbf{G}^i \otimes \mathbf{g}_i. \quad (3.21)$$

Zusammen mit einer Darstellung für den Einheitstensor in konvektiven Koordinaten $\mathbf{1} = G_{ij} \mathbf{G}^i \otimes \mathbf{G}^j$ ergibt Einsetzen in Gleichung (1.12) den GREEN-LAGRANGE-Verzerrungstensor in einer kontravarianten Basis:

$$\mathbf{E} = \frac{1}{2} \left[\left(\mathbf{G}^i \otimes \mathbf{g}_i \right) \left(\mathbf{g}_j \otimes \mathbf{G}^j \right) - G_{ij} \mathbf{G}^i \otimes \mathbf{G}^j \right] = \frac{1}{2} (g_{ij} - G_{ij}) \mathbf{G}^i \otimes \mathbf{G}^j. \quad (3.22)$$

Einsetzen der Beziehung

$$\mathbf{g}_i = \frac{\partial \mathbf{x}}{\partial \xi^i} = \frac{\partial \mathbf{X}}{\partial \xi^i} + \frac{\partial \mathbf{u}}{\partial \xi^i} = \mathbf{G}_i + \frac{\partial \mathbf{u}}{\partial \xi^i} \quad (3.23)$$

führt schließlich auf die Komponenten des GREEN-LAGRANGE-Verzerrungstensors in der Form

$$E_{ij} = \frac{1}{2} \left(\mathbf{G}_i \cdot \frac{\partial \mathbf{u}}{\partial \xi^j} + \frac{\partial \mathbf{u}}{\partial \xi^i} \cdot \mathbf{G}_j + \frac{\partial \mathbf{u}}{\partial \xi^i} \cdot \frac{\partial \mathbf{u}}{\partial \xi^j} \right), \quad (3.24)$$

oder in Matrixschreibweise, wie sie später für die symbolische Implementierung benötigt wird

$$[\mathbf{E}] = \frac{1}{2} (\mathbf{J} \cdot \nabla \mathbf{u}^T + \nabla \mathbf{u} \cdot \mathbf{J}^T + \nabla \mathbf{u} \cdot \nabla \mathbf{u}^T), \quad (3.25)$$

mit dem lokalen Verschiebungsgradienten

$$\nabla \mathbf{u} = \left(\frac{\partial \mathbf{u}}{\partial \xi} \right)^T. \quad (3.26)$$

Wie schon in Abschnitt 1.3 beschrieben, bestimmen sich die Spannungen bei der Verwendung hyperelastischer Materialgesetze aus der Ableitung der Verzerrungsenergiefunktion \mathcal{W} nach den Verzerrungen. Durch Ableitung nach dem GREEN-LAGRANGE-Verzerrungstensor ergibt sich also wie in Gleichung (1.31) angegeben der energetisch konjugierte 2. PIOLA-KIRCHHOFF-Spannungstensor entsprechend in einer kovarianten Basis

$$\mathbf{S} = \frac{\partial \mathcal{W}(\mathbf{E})}{\partial \mathbf{E}} = S^{ij} \mathbf{G}_i \otimes \mathbf{G}_j. \quad (3.27)$$

Durch nochmaliges Ableiten nach den Verzerrungen ergibt sich der vierstufige Stofftensor \mathbb{C} , der jedoch als Folge der Anwendung des symbolischen Programmierwerkzeuges ACEGEN, auf das in Kapitel 4 noch genauer eingegangen wird, für die Implementierung nicht explizit aufgestellt werden muss. Wie in Abschnitt 4.3.1 und Anhang B.2 gezeigt wird führt die symbolische Differentiation der Verzerrungsenergie direkt auf die Spannungen

bzw. nach Gleichung (1.59) auf die inneren Knotenkräfte, sodass eine Berechnung und Implementierung der Komponenten des Stofftensors \mathbb{C} nicht erforderlich ist.

Die in Abschnitt 1.3 beschriebenen hyperelastischen Materialgesetze wurden für die implementierten Schalenelemente umgesetzt. Bei der linearen ST. VENANT-KIRCHHOFF-Formulierung kann die Annahme verschwindender Normalspannungen in Dickenrichtung $S_{\zeta\zeta} = 0$ direkt erfüllt und analytisch nach $E_{\zeta\zeta}$ aufgelöst werden. Beim nichtlinearen Zusammenhang zwischen Spannungen und Verzerrungen der NEO-HOOKE-Formulierung muss diese Bedingung iterativ erfüllt werden, was in der Implementierung durch ein NEWTON-Verfahren realisiert wird.

3.2. Volumenschalenelemente

3.2.1. Das 'Solid-Shell'-Konzept

Eine Alternative zum Degenerationskonzept für die Beschreibung von Schalenstrukturen, stellt das *Solid-Shell*-Konzept dar. Durch eine spezielle Beschreibung der Kinematik ergeben sich sog. *Volumenschalen* oder *Solid-Shell*-Elemente, Formulierungen ohne Degeneration der Dicke, also mit vollständigen 3D-Eigenschaften. Diese erlauben etwa eine Kopplung von Schalenstrukturen und Kontinua ohne die Definition von Übergangsbedingungen für die Rotationsfreiheitsgrade. Auch eine direkte Anwendung kontinuumsmechanisch formulierter Materialgesetze ist möglich, da keine Annahmen z. B. zur Erfüllung eines ebenen Spannungszustandes erforderlich sind. Die Möglichkeit der Abbildung eines vollständigen 3D-Spannungszustandes bringt etwa bei der Simulation von Laminaten Vorteile mit sich, da hier Normalspannungen in Dickenrichtung als Kriterium für eine Delamination direkt in der Formulierung enthalten sind und diese nicht durch zusätzliche Beziehungen nachträglich berechnet werden müssen.

Im Kontext einer expliziten Zeitintegration werden Solid-Shell-Elemente – insbesondere in kommerziellen Anwendungen – bisher kaum eingesetzt. Bei der groben Diskretisierung einer dünnen Schalenstruktur oder eines Laminates mit vielen Schichten kann die Dicke eines Elementes deutlich kleiner sein als dessen Ausdehnung in Schalenrichtung. Daraus resultiert im zentralen Differenzenverfahren ein kleinerer Zeitschritt verglichen mit dem einer degenerierten Schalenformulierung, da in dessen Berechnung nach Gleichung (2.13) die kleinste geometrische Abmessung eingeht. Betrachtet man jedoch kommerzielle Simulationen etwa im Bereich der Automobilindustrie, bei denen derzeit Elementkantenlängen von 6 – 8 mm bei Bauteildicken von 2 – 4 mm die Regel sind, so fällt dieses Verhältnis kaum ins Gewicht und die Zeitschrittweite einer Vernetzung mit Volumenschalen liegt in der gleichen Größenordnung wie die einer Schalendiskretisierung. Beispielsweise sind bei der Simulation von Gussbauteilen lokal auch größere Dicken durchaus üblich, sodass die Verwendung von Volumenschalenelementen hier insbesondere von Vorteil sein kann, und damit die Größe des kritischen Zeitschrittes keine Einschränkung darstellt.

Die Geometrie der Volumenschale wird durch die in Abbildung 3.5 angegebenen Ortsvektoren der Schalenober- und -unterseite und durch lineare Interpolation in Dickenrichtung beschrieben:

$$\mathbf{X}(\xi, \eta, \zeta) = \frac{1}{2} \left((1 + \zeta) \mathbf{X}_o(\xi, \eta) + (1 - \zeta) \mathbf{X}_u(\xi, \eta) \right). \quad (3.28)$$

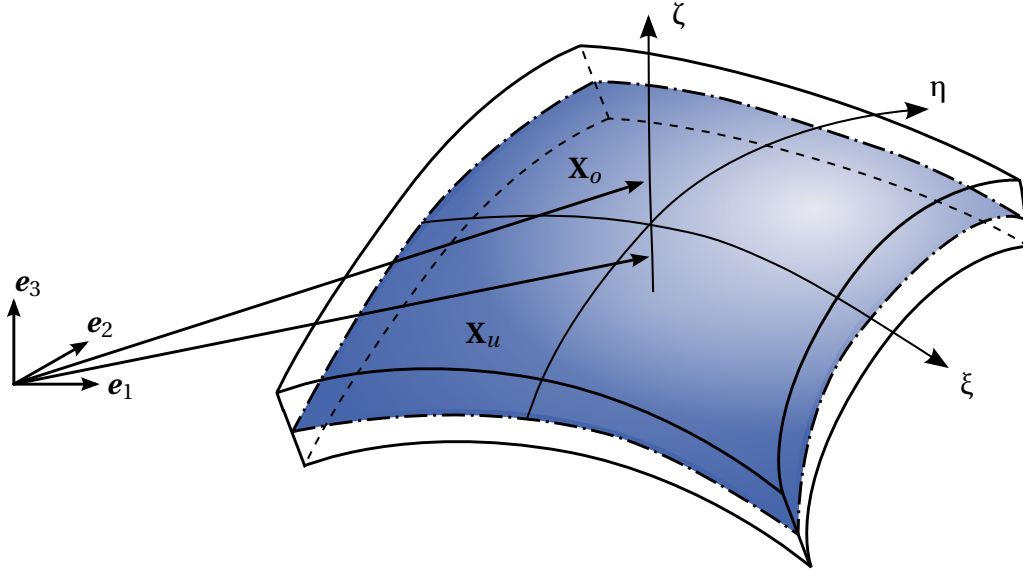


Abbildung 3.5.: Beschreibung der Schalen geometrie nach dem Solid-Shell-Konzept

Wird die Verschiebung ebenfalls linear in Dickenrichtung angenommen, so ergibt sich analog

$$\mathbf{u}(\xi, \eta, \zeta) = \frac{1}{2} \left((1 + \zeta) \mathbf{u}_o(\xi, \eta) + (1 - \zeta) \mathbf{u}_u(\xi, \eta) \right), \quad (3.29)$$

mit den Verschiebungen der Schalenober- und -unterseite \mathbf{u}_o und \mathbf{u}_u . Im Schrifttum wird bei GRUTTMANN [48] und PARISCH [102] eine hierarchische Erhöhung der Interpolationsordnung für die Verschiebungen in Dickenrichtung vorgeschlagen, um bestimmte Abbildungseigenschaften zu verbessern (siehe auch [50, 54]). Diese Möglichkeit soll hier nicht vertieft werden, eine vergleichbare Erhöhung der Interpolationsordnung durch Einführung zusätzlicher Freiheitsgrade – nicht ausschließlich für die Dickenrichtung – wird durch die Anwendung der *Methode der inkompatiblen Moden* in Abschnitt 3.4.5 beschrieben.

Eine Umstellung von Gleichung (3.28) führt auf eine alternative Darstellung, welche die Analogie zur degenerierten Schalenformulierung deutlich macht. Es gilt dann für die Ortsvektoren in der Ausgangs- und Momentankonfiguration

$$\mathbf{X}(\xi, \eta, \zeta) = \frac{1}{2} \underbrace{(\mathbf{X}_o(\xi, \eta) + \mathbf{X}_u(\xi, \eta))}_{\equiv \mathbf{X}_m(\xi, \eta)} + \frac{1}{2} \zeta \underbrace{(\mathbf{X}_o(\xi, \eta) - \mathbf{X}_u(\xi, \eta))}_{\equiv t(\xi, \eta) \mathbf{D}(\xi, \eta)}, \quad \text{und} \quad (3.30)$$

$$\mathbf{x}(\xi, \eta, \zeta) = \frac{1}{2} \underbrace{(\mathbf{x}_o(\xi, \eta) + \mathbf{x}_u(\xi, \eta))}_{\equiv \mathbf{x}_m(\xi, \eta)} + \frac{1}{2} \zeta \underbrace{(\mathbf{x}_o(\xi, \eta) - \mathbf{x}_u(\xi, \eta))}_{\equiv t(\xi, \eta) \mathbf{d}(\xi, \eta)}, \quad \text{mit} \quad (3.31)$$

$$\mathbf{x}_o(\xi, \eta) = \mathbf{X}_o(\xi, \eta) + \mathbf{u}_o(\xi, \eta) \quad \text{und} \quad \mathbf{x}_u(\xi, \eta) = \mathbf{X}_u(\xi, \eta) + \mathbf{u}_u(\xi, \eta), \quad (3.32)$$

wobei sich die Ausdrücke unter den Klammern auf die Geometriebeschreibung des Degenerationskonzeptes aus Gleichung (3.1) beziehen. Hier wird auch deutlich, dass eine aufwändige Beschreibung der Rotationen wie in Abschnitt 3.1.1 nicht erforderlich ist, da sich der Schalendirektor in der Momentankonfiguration direkt aus der Differenz aus Orts- bzw. Verschiebungsvektoren der Schalenoberflächen bestimmen lässt.

3.2.2. Diskretisierung mit Solid-Shell-Elementen

Interpolation von Geometrie und Verschiebungen

Mit den Ansatzfunktionen für die Schalenebene aus den Gleichungen (3.9) – (3.11) ergibt sich für die Diskretisierung des Ortsvektors für die Volumenschale

$$\mathbf{X}^h(\xi, \eta, \zeta) = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \sum_{i=1}^n \left(\frac{1}{2} N_i(\xi, \eta) [(1 + \zeta) \mathbf{X}_{i_o} + (1 - \zeta) \mathbf{X}_{i_u}] \right) \quad (3.33)$$

und für die Verschiebungen

$$\mathbf{u}^h(\xi, \eta, \zeta) = \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \sum_{i=1}^n \left(\frac{1}{2} N_i(\xi, \eta) [(1 + \zeta) \mathbf{d}_{i_o} + (1 - \zeta) \mathbf{d}_{i_u}] \right), \quad (3.34)$$

wobei der Zähler i hier für ein *Knotenpaar* mit jeweils einem Knoten auf der Elementober- und -unterseite steht. Wie in Abbildung 3.6 dargestellt beschreiben die Ortsvektoren \mathbf{X}_{i_u} und \mathbf{X}_{i_o} die Lage des Knotenpaares i , und in den Vektoren $\mathbf{d}_{i_u} = [u_{i_u} \ v_{i_u} \ w_{i_u}]^T$ und $\mathbf{d}_{i_o} = [u_{i_o} \ v_{i_o} \ w_{i_o}]^T$ sind die sechs Verschiebungsfreiheitsgrade des Knotenpaares zusammengefasst. Es ergibt sich also bei der Verwendung linearer LAGRANGE-Ansätze eine Formulierung mit 8 Knoten ($n = 4$) und 24 Freiheitsgraden; bei quadratischen LAGRANGE-Ansätzen erhält man 18 Knoten ($n = 9$) und entsprechend 54 Freiheitsgrade. Die Interpolation mit den quadratischen Serendipity-Ansätzen aus Gleichung (3.11) führt auf 16 Knoten ($n = 8$) und 48 Freiheitsgrade.

Verzerrungen und Spannungen

Wie in Abschnitt 3.1.2 für die degenerierten Schalenelemente beschrieben, werden zur Beschreibung der Kinematik und des Stoffgesetzes bei Solid-Shell-Elementen konvektive Koordinaten verwendet. Die in Abschnitt 3.1.2 angegebene Notation gilt für die Solid-Shell-Elemente völlig analog.

Wie bereits erwähnt entfällt bei der Volumenschale die separate Erfüllung etwa eines ebenen Spannungszustandes, das Materialgesetz stellt also die Beziehung zwischen den vollständigen Verzerrungs- und Spannungstensoren her, wie in Abschnitt 1.3 beschrieben.

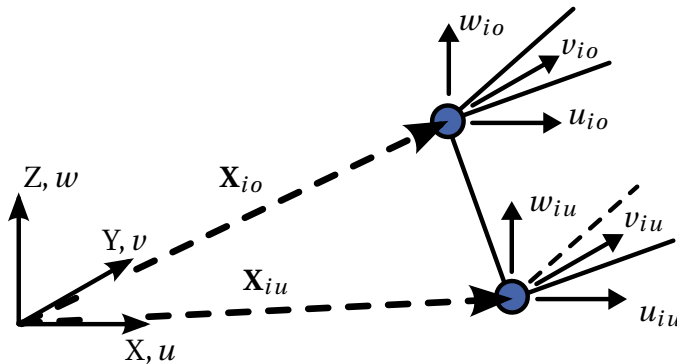


Abbildung 3.6.: Freiheitsgrade am Knoten i eines Solid-Shell-Elements

Die Volumenschalenelemente wurden mit linearen und nichtlinearen hyperelastische Materialformulierungen (siehe Gln. (1.32) und (1.34)) sowie teilweise mit den in Abschnitt 1.3.2 eingeführten Plastizitätsmodellen umgesetzt.

3.3. Massenmatrix

3.3.1. Grundlagen

Die Masse einer Struktur wird in der räumlich diskretisierten Bewegungsgleichung (2.1) durch die Massenmatrix \mathbf{M} aus Gleichung (1.60) repräsentiert. Da diese im zentralen Differenzenverfahren in Gleichung (2.5) invertiert bzw. zerlegt werden muss, ist sowohl aus Gründen des Speicherplatzes, als auch wegen des Rechenaufwandes die Verwendung einer diagonalisierten Form von erheblichem Vorteil und in der Praxis üblich. Physikalisch bedeutet eine Diagonalisierung der Massenmatrix die Konzentration der Masse auf die Elementknoten. Im zentralen Differenzenverfahren ergibt sich daraus eine Entkopplung der Gleichungen und es sind ausschließlich Vektoroperationen auf globaler Ebene durchzuführen, was zu dem bereits angesprochenen sehr geringen Zeitaufwand pro Zeitschritt führt. FELIPPA gibt in [43] einen umfassenden Überblick über verschiedene Massenmatrizen sowie deren Eigenschaften.

Im Folgenden wird in Abschnitt 3.3.2 zunächst die konsistente Massenmatrix eingeführt, die nach [43] passender als *steifigkeits-konsistent* zu bezeichnen ist. In Abschnitt 3.3.3 werden dann verschiedene Möglichkeiten der Diagonalisierung vorgestellt und deren Einfluss auf die Form der diagonalisierten Massenmatrix für die verschiedenen Elementklassen diskutiert. Ein Beispiel an dem die Einflüsse der verschiedenen Diagonalisierungsalgorithmen sowie die Wahl der Integrationsordnung deutlich wird, befindet sich in Anhang C.

3.3.2. Konsistente Massenmatrix

Für ein Element mit k Knoten werden die Ansatzfunktionen zur Interpolation der Knotenfreiheitsgrade in einer Matrix angeordnet

$$\mathbf{N} = (\mathbf{N}_1 \quad \mathbf{N}_2 \quad \dots \quad \mathbf{N}_k) , \quad (3.35)$$

bestehend aus k Matrizen der Größe “Anzahl der Dimensionen” \times “Anzahl der Freiheitsgrade pro Knoten” ($\text{ndm} \times \text{ndf}$).

Mit den Ansatzfunktionen aus den Gleichungen (3.9), (3.10) oder (3.11) und den x -, y - und z -Komponenten der Vektoren \mathbf{V}_1 und \mathbf{V}_2 aus Abbildung 3.2 gilt für die degenerierte Schalenformulierung aus Abschnitt 3.1

$$\mathbf{N}_i = \begin{pmatrix} N_i(\xi, \eta) & 0 & 0 & -\frac{1}{2} \zeta t_i N_i(\xi, \eta) V_{2,x}^i & \frac{1}{2} \zeta t_i N_i(\xi, \eta) V_{1,x}^i \\ 0 & N_i(\xi, \eta) & 0 & -\frac{1}{2} \zeta t_i N_i(\xi, \eta) V_{2,y}^i & \frac{1}{2} \zeta t_i N_i(\xi, \eta) V_{1,y}^i \\ 0 & 0 & N_i(\xi, \eta) & -\frac{1}{2} \zeta t_i N_i(\xi, \eta) V_{2,z}^i & \frac{1}{2} \zeta t_i N_i(\xi, \eta) V_{1,z}^i \end{pmatrix}. \quad (3.36)$$

Die Einträge in der vierten und fünften Spalte ergeben sich durch die Darstellung des Schalendirektors in der verformten Konfiguration für kleine Rotationen aus Gleichung (3.6), die

zur Aufstellung der Massenmatrix verwendet wird. Die beiden Matrizen für ein Knotenpaar der Volumenschale aus Abschnitt 3.2 lauten entsprechend

$$\mathbf{N}_i = \begin{cases} \begin{pmatrix} \frac{1}{2}(1-\zeta)N_i & 0 & 0 \\ 0 & \frac{1}{2}(1-\zeta)N_i & 0 \\ 0 & 0 & \frac{1}{2}(1-\zeta)N_i \end{pmatrix}, & \text{für } i = 1, \dots, \frac{k}{2} \\ \begin{pmatrix} \frac{1}{2}(1+\zeta)N_{i-k/2} & 0 & 0 \\ 0 & \frac{1}{2}(1+\zeta)N_{i-k/2} & 0 \\ 0 & 0 & \frac{1}{2}(1+\zeta)N_{i-k/2} \end{pmatrix}, & \text{für } i = \frac{k}{2} + 1, \dots, k. \end{cases} \quad (3.37)$$

Die konsistente Elementmassenmatrix ergibt sich dann durch Integration über das Elementvolumen mit der als konstant angenommenen Rohdichte ρ zu

$$\mathbf{M}_c^e = \rho \int_V \mathbf{N}^T \cdot \mathbf{N} \, dV. \quad (3.38)$$

Innerhalb der Elementroutine wird die Integration numerisch, z. B. mit GAUSS-Integration durchgeführt und es ergeben sich im Allgemeinen vollbesetzte symmetrische Elementmatrizen, die auf Systemebene wie in Gleichung (1.60) angegeben zur Gesamtmassenmatrix \mathbf{M} assembliert werden können.

3.3.3. Diagonalisierte Massenmatrix

Im Schrifttum werden verschiedenen Möglichkeiten zur Aufstellung einer diagonalisierten Elementmassenmatrix vorgestellt. Übersichtliche Darstellungen finden sich etwa bei ZIENKIEWICZ ET. AL. [136] und bei HUGHES [58]. Die dort zusammengestellten Techniken wurden in der Arbeit für die verschiedenen Elementtypen implementiert, um deren Auswirkung auf die Abbildungseigenschaften an numerischen Beispielen zu testen. Es ist zu bemerken, dass bei Elementformulierungen mit linearer Ansatzordnung das unterschiedliche Vorgehen zur Diagonalisierung keine Auswirkungen auf die Massenmatrix hat, jeder der vorgestellten Algorithmen also die gleiche diagonalisierte Massenmatrix liefert. Bei höheren Ansatzordnungen und hier insbesondere bei verzerrten Elementgeometrien führen die Algorithmen auf unterschiedliche Massenverteilungen, was im Anhang C am Beispiel eines einzelnen Elementes gezeigt wird. Voraussetzung für sinnvolle Abbildungseigenschaften der Elementformulierung ist die korrekte Repräsentation der gesamten Elementmasse m durch die diagonalisierte Massenmatrix \mathbf{M}_d^e . Für ein ausschließlich auf Verschiebungsfreiheitsgraden basierendes Element mit ndf Freiheitsgraden pro Knoten und insgesamt nst Freiheitsgraden muss also beispielsweise

$$\frac{1}{\text{ndf}} \sum_{i=1}^{\text{nst}} [\mathbf{M}_d^e]_{ii} = \frac{1}{\text{ndf}} \sum_{i=1}^{\text{nst}} \sum_{j=1}^{\text{nst}} [\mathbf{M}_c^e]_{ij} = m \quad (3.39)$$

gelten. Die den Rotationsfreiheitsgraden zugeordnete Masse wird bei entsprechenden Elementtypen hinsichtlich der Diagonalisierung separat behandelt, worauf am Ende dieses Abschnittes noch genauer eingegangen wird. Diagonalisierte Massenmatrizen, die diese Eigenschaft erfüllen, sowie nur positive Einträge auf der Diagonalen aufweisen, werden im Folgenden als *zulässige* Diagonalformen bezeichnet.

Zeilenweise Summation ('row-sum')

Eine diagonalisierte Form der Massenmatrix erhält man, in dem die Einträge der konsistenten Massenmatrix zeilenweise aufsummiert werden und das Ergebnis auf die Diagonale geschrieben wird. Für ein Element mit nst Freiheitsgraden ergibt sich demnach

$$\left[\mathbf{M}_{d,rs}^e \right]_{ij} = \begin{cases} \sum_{k=1}^{nst} [\mathbf{M}_c^e]_{ik}, & \text{für } i = j \\ 0, & \text{für } i \neq j \end{cases}. \quad (3.40)$$

Diese Form der Diagonalisierung führt bei Elementen mit LAGRANGE-Ansätzen immer zu einer zulässigen Diagonalform der Massenmatrix, in der vorliegenden Arbeit wurde sie für die 4- und 9-Knoten Schalen sowie für die 8- und 18-Knoten Volumenschalen eingesetzt. Problematisch ist die Anwendung bei *Serendipity*-Ansätzen, also z. B. bei einer 8-Knoten Schalenformulierung, da hier negative Einträge auf der Diagonalen auftreten können. Da dies einer negativen Knotenmasse entspricht, die physikalisch nicht zulässig ist, müssen bei *Serendipity*-Elementen andere Methoden zur Diagonalisierung eingesetzt werden.

Integration an den Elementknoten ('nodal integration')

Zur Integration der konsistenten Massenmatrix wird wie auch für die übrigen Elementmatrizen üblicherweise eine GAUSS-Quadratur Regel angewendet. Eine wesentliche Eigenschaft der verwendeten Ansatzfunktionen ist

$$N_i(\xi_j, \eta_j, \zeta_j) = \begin{cases} 1, & \text{für } i = j \\ 0, & \text{für } i \neq j \end{cases}, \quad (3.41)$$

sie nehmen also am jeweiligen Elementknoten den Wert 1 an und sind an allen anderen Knoten 0. Wendet man demzufolge eine Integrationsregel an, deren Integrationspunkte mit den Elementknoten zusammenfallen, so ergibt sich automatisch eine Diagonalmatrix $\mathbf{M}_{d,nq}^e$. Solche Integrationsregeln werden allgemein GAUSS-LOBATTO Regeln genannt. Für eine Elementformulierung mit quadratischen Ansätzen aus der LAGRANGE-Klasse wird also beispielsweise eine 3-Punkt GAUSS-LOBATTO Regel angewendet – diese wird im Schrifttum auch als SIMPSON-Integration bezeichnet. Insbesondere bei stark von einer Quaderform abweichenden Elementgeometrien kann Gleichung (3.38) dann nicht mehr exakt integriert werden, da die Wahl einer höheren Integrationsordnung nicht mehr auf eine diagonalisierte Massenmatrix führt. Die Auswirkungen auf die diagonalisierte Massenmatrix werden an einem Beispiel in Anhang C diskutiert. Für *Serendipity*-Elemente ist eine solche Integrationsregel nicht möglich, da der Mittelknoten fehlt, also eine 3-Punkt GAUSS-LOBATTO-Regel nicht zu einer Diagonalmatrix führen würde.

Gleichmäßige Verteilung ('equal lumping')

Eine vergleichsweise einfache Strategie zur Diagonalisierung der Massenmatrix ist die gleichmäßige Verteilung der Elementmasse auf die einzelnen Elementknoten. Für eine Element der Masse m mit nen Knoten ergibt sich

$$\left[\mathbf{M}_{d,eq}^e \right]_{ii} = \frac{m}{nen}, \quad (3.42)$$

wodurch die Bedingung (3.39) wegen

$$\frac{1}{\text{ndf}} \sum_{i=1}^{\text{nst}} [\mathbf{M}_{d,\text{eq}}^e]_{ii} = \frac{1}{\text{ndf}} \frac{\text{nst} \cdot m}{\text{nen}} = \frac{1}{\text{ndf}} \frac{\text{ndf} \cdot \text{nen} \cdot m}{\text{nen}} = m \quad (3.43)$$

automatisch erfüllt ist. Diese Form der Diagonalisierung führt nur für lineare Elemente zu brauchbaren Ergebnissen, bei höheren Ansatzordnungen müssen Eck-, Rand- und Mittelknoten unterschiedlich gewichtet werden.

‘Special Lumping’

Eine weitere Methode zur Diagonalisierung der Massenmatrix wird von HUGHES in [58] als ‘*Special Lumping*’ bezeichnet. Hierbei werden die Einträge proportional zu den Diagonalelementen der konsistenten Massenmatrix berechnet. Es ergibt sich für ein Element der Masse m , ndf Freiheitsgraden pro Knoten und insgesamt nst Freiheitsgraden

$$[\mathbf{M}_{d,\text{sl}}^e]_{ij} = \begin{cases} \alpha [\mathbf{M}_c^e]_{ii}, & \text{für } i = j \\ 0, & \text{für } i \neq j \end{cases}, \quad \text{mit } \alpha = \frac{\sum_{i=1}^{\text{nst}} \sum_{j=1}^{\text{nst}} [\mathbf{M}_c^e]_{ij}}{\sum_{i=1}^{\text{nst}} [\mathbf{M}_c^e]_{ii}}. \quad (3.44)$$

Hier wird die Bedingung (3.39) durch die Wahl des Skalierungsfaktors α automatisch erfüllt.

Skalierung der Diagonalelemente

Eine der ‘*Special Lumping*’ Methode sehr ähnliche Vorgehensweise wird von ZIENKIEWICZ ET. AL. in [136] beschrieben. Sie basiert auf der Einführung verschiedener Skalierungsfaktoren α_k , jeweils für die unterschiedlichen Knotentypen. Bei der Volumenschale mit quadratischer Interpolation in Schalenebene mit den Ansätzen aus Gleichung (3.10) sind dies beispielsweise Eckknoten, Seitenmittelknoten und Mittelknoten. Die Einträge der diagonalisierten Massenmatrix ergeben sich also zu

$$[\mathbf{M}_{d,\text{sd}}^e]_{ij} = \begin{cases} \alpha_k [\mathbf{M}_c^e]_{ii}, & \text{für } i = j \\ 0, & \text{für } i \neq j \end{cases}, \quad \text{mit } k = \begin{cases} 1 & \text{für Eckknoten} \\ 2 & \text{für Seitenmittelknoten} \\ 3 & \text{für Mittelknoten} \end{cases}. \quad (3.45)$$

Die Parameter α_k müssen hierbei so gewählt werden, dass die Bedingung (3.39) erfüllt ist. Werden bewusst größere Werte für die einzelnen α_k gewählt, die Elementmasse also skaliert, so lassen sich die Eigenfrequenzen und folglich auch die Größe des kritischen Zeitschrittes aus Gleichung (2.13) beeinflussen. Die Einflüsse dieser Modifikationen auf das dynamische Abbildungsverhalten müssen dann entsprechende überprüft werden (siehe hierzu auch Abschnitt 2.2.6). Außerdem muss gelten $\alpha_k > 0, \forall k$, da negative Knotenmassen nicht zulässig sind. Die zuvor vorgestellte ‘*Special Lumping*’ Methode lässt sich durch entsprechende Wahl der Größen α_k ($\alpha_1 = \alpha_2 = \dots = \alpha_k$) reproduzieren. Durch Wichtung lässt sich die Masse z. B. stärker auf die Seitenmittelknoten konzentrieren, was zu Änderungen im dynamischen Verhalten der Elemente führt. Von besonderem Interesse ist diese Methode bei der Klasse der Serendipity-Elemente. Diese besitzen keinen Mittelknoten, es muss also nur ein Parameter α_1 frei gewählt werden. Die Größe von α_2 lässt sich dann über die Bedingung (3.39) bestimmen, wobei auch hier $\alpha_1 > 0$ und $\alpha_2 > 0$ gewährleistet sein muss.

Behandlung von Rotationsfreiheitsgraden

Bei Elementen, die nicht ausschließlich Verschiebungsfreiheitsgrade besitzen, wie etwa die für die Arbeit implementierten degenerierten Schalenformulierungen, müssen den Rotationsfreiheitsgraden ebenfalls Massen zugeordnet werden. Bei der Anwendung der oben skizzierten Vorgehensweisen zur Diagonalisierung der Massenmatrix können diese ‘*Rotationsträgheiten*’ sehr klein werden, was zu hohen Eigenfrequenzen und folglich in der expliziten Zeitintegration zu einem sehr kleinen kritischen Zeitschritt führt (vgl. Gleichung (2.13)). HUGHES, LIU & LEVIT schlagen für die Anwendung von Schalenelementen für explizite Zeitintegration in [59] die Verwendung skalierter *Verschiebungsmassen* als *Rotationsträgheiten* vor. Dies wird auch in kommerziellen FE-Codes angewendet, wie z. B. im Theoriehandbuch zu LS-DYNA [49] beschrieben ist. Zunächst werden die Verschiebungsmassen durch zeilenweises Aufsummieren der konsistenten Massenmatrix berechnet. Die Rotationsträgheiten ergeben sich dann für den Knoten i zu

$$m_{\text{rot}}^i = \alpha^i m_{\text{disp}}^i, \quad \text{mit} \quad \alpha^i = \max\{\alpha_1^i, \alpha_2^i\} \quad (3.46)$$

und den Skalierungsfaktoren

$$\begin{aligned} \alpha_1^i &= \bar{z}_i^2 + \frac{1}{12} \tilde{z}_i^2 & \bar{z}_i &= \frac{z_{o,i} + z_{u,i}}{2} \\ \alpha_2^i &= \frac{V}{8 t_i} & \tilde{z}_i &= z_{o,i} - z_{u,i}. \end{aligned} \quad (3.47)$$

Die Koordinaten $z_{o,i}$ und $z_{u,i}$ beschreiben jeweils die Lage der Elementober- und -unterseite, V ist das Elementvolumen und t_i die Dicke am Knoten i .

Bei den in der Arbeit verwendeten Elementformulierungen mit linearen und quadratischen LAGRANGE-Ansätzen führt diese Methode zu zulässigen Massenmatrizen, werden Serendipity-Ansätze verwendet, müssen die Verschiebungsmassen auf andere Weise – etwa durch die zuvor beschriebene ‘*Special Lumping*’ Technik – berechnet werden, da das zeilenweise Aufsummieren wie bereits angesprochen zu negativen Knotenmassen führen kann.

Anmerkung zur Diagonalisierung

Zu den vorgestellten Diagonalisierungstechniken sei angemerkt, dass sie für die in der Arbeit implementierten Elementformulierungen mit LAGRANGE-Ansätzen – mit Ausnahme der gleichmäßigen Massenverteilung – zu sehr ähnlichen Massenmatrizen führen. Für lineare Ansätze sind sie sogar identisch, kleinere Unterschiede ergeben sich lediglich bei höheren Ansatzordnungen und stark verzerrten Elementgeometrien, was anhand eines Beispiels in Anhang C verdeutlicht wird. Für die in der Arbeit durchgeführten Simulationen mit Elementformulierungen mit LAGRANGE-Ansätzen wurde stets die Methode des zeilenweisen Aufsummierens angewendet. Auch bei der Wahl der Integrationsordnung ergeben sich insbesondere bei stark von einer rechtwinkligen Form abweichenden Elementgeometrien Unterschiede in der Massenmatrix. Dieser Einfluss wird ebenfalls in Anhang C verdeutlicht, wobei hinsichtlich der Effizienz der Elementformulierung eine konservative – also ‘*zu hohe*’ – Integrationsordnung für die Massenmatrix unerheblich ist, da diese lediglich einmal zu Beginn einer Simulation berechnet werden muss.

Bei Serendipity-Elementen konnten durch Anwendung der ‘*Special-Lumping*’ Methode oder durch ‘*Skalierung der Diagonalelemente*’ zulässige diagonalisierte Massenmatrizen aufgestellt werden. Von der ‘*row-sum*’ Methode ist wegen des besprochenen Auftretens negativer Knotenmassen aber grundsätzlich abzusehen.

3.4. Vermeidung von Versteifungseffekten

3.4.1. Definition und Klassifizierung von ‘Locking’

Elementformulierungen, die wie in Abschnitt 1.4 beschrieben, auf dem *Prinzip der virtuellen Verschiebungen* basieren – sog. *Verschiebungselemente* – können bekanntermaßen unerwünschte Versteifungseffekte aufweisen. Diese zeichnen sich dadurch aus, dass beispielsweise eine gesuchte Verschiebung im diskretisierten System erheblich kleiner ist als in der analytischen Lösung. In der Strukturmechanik bedeutet dies also ein zu steifes Elementverhalten – daher der Begriff der *Versteifung*. Seit den Anfangszeiten der Finite-Elemente-Methode beschäftigten sich Entwicklungen in der Elementtechnologie daher mit der Behandlung solcher unerwünschter Effekte; eine vollständige Liste der relevanten Veröffentlichungen anzugeben ist praktisch unmöglich. Im internationalen Schrifttum hat sich der Begriff ‘*Locking*’ etabliert, der auch in deutschsprachigen Arbeiten verwendet wird. Daher wird auch im Folgenden oft auf eine deutsche Übersetzung verzichtet.

Künstliche Versteifungen haben verschiedene Ursachen, deren Einfluss jeweils in Abhängigkeit eines kritischen Parameters ausgedrückt werden kann. Nach der Art dieses kritischen Parameters lassen sich Locking-Phänomene in zwei verschiedenen Klassen einteilen. Wird die Versteifung durch eine geometrische Größe gesteuert wie etwa das Seitenverhältnis der Elementkanten, so spricht man von *geometrischem Locking*; hängt der Einfluss des Effekts von einer Materialgröße ab, handelt es sich um einen *materiellen Lockingeffekt*. Der Einfluss eines Versteifungseffektes kann prinzipiell durch Steigerung der Netzfeinheit verringert werden, ein versteifendes System wird also bei Netzverfeinerung immer noch gegen die richtige Lösung konvergieren, jedoch langsamer.

Eine geschlossene Definition des Begriffes Locking lässt sich nur schwierig formulieren. In der Arbeit von BISCHOFF [19] wird das Phänomen durch drei verschiedenen Betrachtungsweisen charakterisiert. Diese Art der Definition wird in der Arbeit von KOSCHNICK [78] nochmals aufgegriffen und ausgearbeitet. Die pragmatischste Erklärung ist die *mechanische Sichtweise*, bei der die Entstehung von Versteifungseffekten durch das Auftreten parasitärer Spannungen begründet wird, die nur im diskretisierten System existieren. Die gesamten Spannungen werden also zu groß abgebildet und es resultiert folglich eine Überschätzung der inneren Energie, also ein zu steifes Systemverhalten. Weitere Definitionen werden in den genannten Arbeiten über eine *numerische* und eine *mathematische Sichtweise* formuliert. Erstere basiert auf einem Ansatz von HUGHES und wird auch in dessen Buch [58] erläutert, die zweite Sichtweise ist im Buch von BRAESS [28] ausführlich beschrieben. An dieser Stelle soll auf eine Wiederholung der Erklärungsansätze verzichtet werden und es sei auf die sehr umfassenden und an Beispielen belegten Ausführungen in [19] und [78] verwiesen.

In den folgenden Abschnitten werden verschiedene Strategien vorgestellt, die sich mit der Beseitigung künstlicher Versteifungseffekte befassen. Teilweise wurden die präsentierten Konzepte bereits in zahlreichen Veröffentlichungen für verschiedene Elementformulierun-

gen angewendet. Nach einer kurzen Zusammenstellung der theoretischen Grundlagen werden die gewählten Methoden jeweils für die in den Abschnitten 3.1 und 3.2 vorgestellten Elementtypen aufbereitet, um in Kapitel 4 die Eigenarten bei deren Implementierung unter spezieller Berücksichtigung der Effizienz im Zusammenhang mit symbolischer Programmierung zu diskutieren.

3.4.2. Die Methode der angenommenen Verzerrungen

Entwicklung

Die *Methode der angenommenen Verzerrungen* ('Assumed Natural Strains' – ANS) wurde als Kollokationsmethode entwickelt und basiert auf dem Wissen um bestimmte Stellen, an denen die parasitären Anteile der Verzerrungen verschwinden. Die Methode und ihre Anwendung auf Plattenelemente wurde erstmals von HUGHES & TEZDUYAR [60] und WEMPNER ET. AL. [125] vorgestellt. Die Publikation der Anwendung auf 4-Knoten Schalenelemente durch BATHE & DVORKIN [13] wird heute am häufigsten im Zusammenhang mit der ANS-Methode zitiert und liefert auch die Grundlage für die in der vorliegenden Arbeit implementierten Verzerrungsmodifikationen. Das Element von BATHE & DVORKIN begründet die Familie der MITC-Elemente, die im Bereich der Schalenelemente sehr bekannt und auch in kommerziellen Anwendungen verbreitet sind. Ursprünglich für Querschubverzerrungen eingeführt, wird die Methode bei BETSCH & STEIN [18] auch für die Normalverzerrungen in Dickenrichtung angewendet. Eine Erweiterung der Elementklasse auf quadratische LAGRANGE-Ansätze – das MITC9-Element – findet sich beispielsweise im Aufsatz von BUCALEM & BATHE [29]. Anwendungen der Methode auf Solid-Shell-Elemente erfolgten zunächst durch HAUPTMANN & SCHWEIZERHOF [55] und SZE & YAO [120], sowie in neueren Publikationen [61, 62, 121]. Im Zusammenhang mit reduzierter Integration bei Solid-Shell-Elementen wird von CARDOSO ET. AL. [33] eine Modifikation vorgeschlagen, die auch bei SCHWARZE & REESE [110, 111] verwendet wird.

Eine der ANS-Methode ähnliche Vorgehensweise, die jedoch für die vorliegende Arbeit nicht implementiert wurde, stellt die '*Discrete Strain Gap*' Methode dar. Sie geht auf eine Veröffentlichung von BLETZINGER ET. AL. [26] zurück, als weitere Arbeiten zu diesem Thema sind [20–22, 79] zu nennen. Auch die bereits zitierten Dissertationen von BISCHOFF [19] und KOSCHNICK [78] beschäftigen sich mit DSG. Bei der DSG-Methode werden wie bei ANS die aus der Verschiebungsinterpolation resultierenden Verzerrungen nachträglich modifiziert, um versteifende Anteile zu eliminieren. Für detaillierte Informationen sei auf das angegebene Schrifttum verwiesen.

Variationelle Basis

Eine variationelle Basis der Methode der angenommenen Verzerrungen wurde 1986 von SIMO & HUGHES in [115] formuliert. Auf Grundlage des HU-WASHIZU-Funktionalen werden hierzu zunächst die aus den Verschiebungen resultierenden Verzerrungen \mathbf{E}^u additiv in einen erwünschten und einen unerwünschten – zu Versteifungen führenden – Anteil aufgeteilt:

$$\mathbf{E}^u = \mathbf{E}^a + \mathbf{E}^s. \quad (3.48)$$

Werden die unabhängigen Verzerrungen \mathbf{E} des HU-WASHIZU-Funktionalen aus Gleichung (1.51) gerade so gewählt, dass sie nur die erwünschten Anteile beinhalten, also

$\mathbf{E} = \mathbf{E}^a = \mathbf{E}^u - \mathbf{E}^s$, so ergibt sich

$$\begin{aligned}
\Pi_{\text{HW}} &= \int_{\mathcal{B}_0} \mathcal{W}(\mathbf{E}) + \mathbf{S} : (\mathbf{E}^u - \mathbf{E}) \, dV - \left[\int_{\mathcal{B}_0} \rho_0 \mathbf{b} \cdot \mathbf{u} \, dV + \int_{\partial \mathcal{B}_\sigma} \bar{\mathbf{t}} \cdot \mathbf{u} \, dA \right] + \int_{\mathcal{B}_0} \rho_0 \ddot{\mathbf{u}} \cdot \mathbf{u} \, dV \\
&= \int_{\mathcal{B}_0} \mathcal{W}(\mathbf{E}^a) + \mathbf{S} : (\mathbf{E}^a + \mathbf{E}^s - \mathbf{E}^a) \, dV - \left[\int_{\mathcal{B}_0} \rho_0 \mathbf{b} \cdot \mathbf{u} \, dV + \int_{\partial \mathcal{B}_\sigma} \bar{\mathbf{t}} \cdot \mathbf{u} \, dA \right] + \int_{\mathcal{B}_0} \rho_0 \ddot{\mathbf{u}} \cdot \mathbf{u} \, dV \\
&= \int_{\mathcal{B}_0} \mathcal{W}(\mathbf{E}^a) + \mathbf{S} : \mathbf{E}^s \, dV - \left[\int_{\mathcal{B}_0} \rho_0 \mathbf{b} \cdot \mathbf{u} \, dV + \int_{\partial \mathcal{B}_\sigma} \bar{\mathbf{t}} \cdot \mathbf{u} \, dA \right] + \int_{\mathcal{B}_0} \rho_0 \ddot{\mathbf{u}} \cdot \mathbf{u} \, dV. \tag{3.49}
\end{aligned}$$

Die versteifenden Verzerrungen dürfen keinen Energiebeitrag liefern, die unabhängigen Spannungen \mathbf{S} des Funktionals werden also so gewählt, dass die Orthogonalitätsbedingung

$$\int_{\mathcal{B}_0} \mathbf{S} : \mathbf{E}^s \, dV = 0 \tag{3.50}$$

erfüllt ist. Demnach folgt für die Methode der angenommenen Verzerrungen

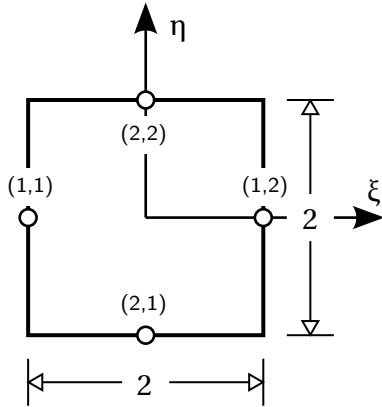
$$\Pi_{\text{HW}}^{\text{ANS}} = \int_{\mathcal{B}_0} \mathcal{W}(\mathbf{E}^a) \, dV - \left[\int_{\mathcal{B}_0} \rho_0 \mathbf{b} \cdot \mathbf{u} \, dV + \int_{\partial \mathcal{B}_\sigma} \bar{\mathbf{t}} \cdot \mathbf{u} \, dA \right] + \int_{\mathcal{B}_0} \rho_0 \ddot{\mathbf{u}} \cdot \mathbf{u} \, dV \rightarrow \text{stat.}, \quad \text{für } 0 \leq t \leq T, \tag{3.51}$$

d. h. die aus den Verschiebungen resultierenden Verzerrungen werden durch die angenommenen Verzerrungen *ersetzt*. Die angenommenen Verzerrungen werden wie bereits erwähnt durch Auswertung des Verzerrungsverlaufs an bestimmten Punkten und Interpolation mit geeigneten Ansätzen erreicht. Die Lage der Punkte und die entsprechende Wahl der Ansätze wird im folgenden Abschnitt beschrieben.

Verzerrungsinterpolation

Im Folgenden wird die Anwendung der Methode der angenommenen Verzerrungen für die implementierten Schalen- und Volumenschalenelemente beschrieben. Die Methode dient der Behandlung verschiedener geometrischer Versteifungseffekte (*Querschublocking*, *Membranlocking*, *Curvature-Thickness Locking*, *Trapezoidallocking*), auf deren Definition, genauen Ursprung und deren Auswirkungen auf das Abbildungsverhalten hier nicht im Detail eingegangen werden soll. Es existieren viele Arbeiten, die sich insbesondere mit der Beschreibung und Behandlung verschiedener Lockingphänomene befassen. Auf einige sei hier ohne Anspruch auf Vollständigkeit verwiesen [7, 19, 50, 54, 78].

Wie eingangs erwähnt basiert die ANS-Methode auf einer Auswertung der Verzerrungen an Kollokationspunkten, an denen die versteifenden Anteile verschwinden, und einer anschließenden Interpolation mit geeigneten Funktionen. In der ursprünglichen MITC-Elementformulierung mit bilinearen Ansätzen für Geometrie und Verschiebungen von BATHÉ & DVORKIN [13] erfolgt eine Auswertung der Querschubverzerrungen an den Seitenmittelpunkten der Elementmittelfläche. Anschließend wird die Komponente $E_{\xi\zeta}$ linear in



Punkt	ξ	η	ζ_1	ζ_2
(1,1)	-1	0	0	± 1
(1,2)	1	0	0	± 1
(2,1)	0	-1	0	± 1
(2,2)	0	1	0	± 1

Abbildung 3.7.: Lage der Kollokationspunkte für bilineare Elementformulierung – in der Mittelebene (ζ_1), bzw. auf der Elementober- und -unterseite (ζ_2)

η und konstant in ξ und entsprechend $E_{\eta\zeta}$ linear in ξ und konstant in η angenommen um den Effekt des *Querschublocking* zu beseitigen. Mit der Lage der Kollokationspunkte in der Elementmittelfläche gemäß Abbildung 3.7 – also bei $\zeta = 0$ – ergibt sich für die interpolierten Verzerrungen

$$E_{\eta\zeta}^a(\xi) = \frac{1}{2} \sum_{i=1}^2 (1 + \xi\xi_i) E_{\eta\zeta}^{(1,i)}, \quad \text{mit } \xi_i = \{-1, 1\} \quad (3.52)$$

$$E_{\xi\zeta}^a(\eta) = \frac{1}{2} \sum_{i=1}^2 (1 + \eta\eta_i) E_{\xi\zeta}^{(2,i)}, \quad \text{mit } \eta_i = \{-1, 1\}. \quad (3.53)$$

Die ursprünglich für Schalenelemente entwickelte Methode lässt sich völlig analog auch auf Volumenschalenformulierungen anwenden.

In einer Arbeit von CARDOSO ET. AL. [33] sowie bei SCHWARZE & REESE [110, 111] wird für lineare Solid-Shell-Elemente bei reduzierter Integration der Elementmatrizen eine Interpolation mit den Kollokationspunkten auf der Elementober- und -unterseite vorgeschlagen, die zu einer Verbesserung hinsichtlich des Auftretens von Elementkinematiken führen soll. Verwendet man also die in Abbildung 3.7 angegebenen Koordinaten ζ_2 und führt die entsprechende Interpolation zusätzlich linear in ζ -Richtung durch, so ergeben sich die Verzerrungen

$$E_{\eta\zeta}^a(\xi, \zeta) = \frac{1}{4} (1 - \zeta) \sum_{i=1}^2 (1 + \xi\xi_i) E_{\eta\zeta}^{(1,i)b} + \frac{1}{4} (1 + \zeta) \sum_{i=1}^2 (1 + \xi\xi_i) E_{\eta\zeta}^{(1,i)\sharp}, \quad \text{mit } \xi_i = \{-1, 1\} \quad (3.54)$$

$$E_{\xi\zeta}^a(\eta, \zeta) = \frac{1}{4} (1 - \zeta) \sum_{i=1}^2 (1 + \eta\eta_i) E_{\xi\zeta}^{(2,i)b} + \frac{1}{4} (1 + \zeta) \sum_{i=1}^2 (1 + \eta\eta_i) E_{\xi\zeta}^{(2,i)\sharp}, \quad \text{mit } \eta_i = \{-1, 1\} \quad (3.55)$$

Hier bedeutet der Zusatz \sharp die Auswertung bei $\zeta = +1$ und entsprechend b die Verzerrungskomponente bei $\zeta = -1$. In der vorliegenden Arbeit wurden ausschließlich Elementformulierungen mit linearer Interpolation der Geometrie und der Verschiebungen in ζ -Richtung

verwendet. Es gilt also mit dem Platzhalter für die Geometrie, bzw. die Verschiebungen $\square = \{\mathbf{X}, \mathbf{u}\}$ für die Verzerrungskomponenten aus Gleichung (1.12)

$$\frac{\partial \square}{\partial \xi} = \mathcal{O}(\zeta^1), \quad \frac{\partial \square}{\partial \eta} = \mathcal{O}(\zeta^1) \quad \text{und} \quad \frac{\partial \square}{\partial \zeta} = \mathcal{O}(\zeta^0), \quad (3.56)$$

somit liegt für die Querschubverzerrungen $E_{\xi\zeta}$ und $E_{\eta\zeta}$ stets ein linearer Verlauf in ζ vor. Daher ist die Vorgehensweise der Gleichungen (3.54) und (3.55) analog zur Strategie, die ζ -Koordinate unverändert zu lassen, da sich durch Auswertung und lineare Interpolation wieder derselbe Verlauf ergibt. Die für die vorliegende Arbeit implementierten Elementformulierungen wurden also mit der Interpolation der Querschubverzerrungen

$$E_{\eta\zeta}^a(\xi, \zeta) = \frac{1}{2} [(1 - \xi) E_{\eta\zeta}(\xi = -1, \eta = 0) + (1 + \xi) E_{\eta\zeta}(\xi = +1, \eta = 0)] \quad (3.57)$$

$$E_{\xi\zeta}^a(\eta, \zeta) = \frac{1}{2} [(1 - \eta) E_{\xi\zeta}(\xi = 0, \eta = -1) + (1 + \eta) E_{\xi\zeta}(\xi = 0, \eta = +1)], \quad (3.58)$$

analog zur Vorgehensweise in den Arbeiten von HAUPTMANN [54] und HARNAU [50] umgesetzt. In entsprechenden Untersuchungen mit linearer Interpolation und vollständiger Integration ($2 \times 2 \times 2$ GAUSS-Punkte), konnte entsprechend kein Unterschied zur Vorgehensweise mit zusätzlicher Interpolation in ζ -Richtung festgestellt werden.

Einen Ansatz zu Behandlung von *Trapezoidal Locking* oder *Curvature-Thickness Locking* bei linearer Ansatzordnung durch Interpolation der Normalverzerrungen in Dickenrichtung wurde von BETSCH & STEIN [18] vorgeschlagen. Die Verzerrungen $E_{\zeta\zeta}$ werden hierzu an den Eckpunkten der Elementmittelebene ausgewertet und in ξ - und η -Richtung mit Hilfe der bilinearen Ansatzfunktionen aus Gleichung (3.9) interpoliert. Es ergibt sich

$$E_{\zeta\zeta}^a(\xi, \eta) = \sum_{i=1}^4 N_i(\xi, \eta) E_{\zeta\zeta}(\xi = \xi_i, \eta = \eta_i, \zeta = 0), \quad (3.59)$$

mit der Lage der Elementknoten ξ_i und η_i aus Abbildung 3.4a.

Bei den quadratischen Elementen wird die ANS-Methode wie in [29] für die MITC9-Formulierung vorgeschlagen zusätzlich zur Behandlung von *Membranlocking* eingesetzt,

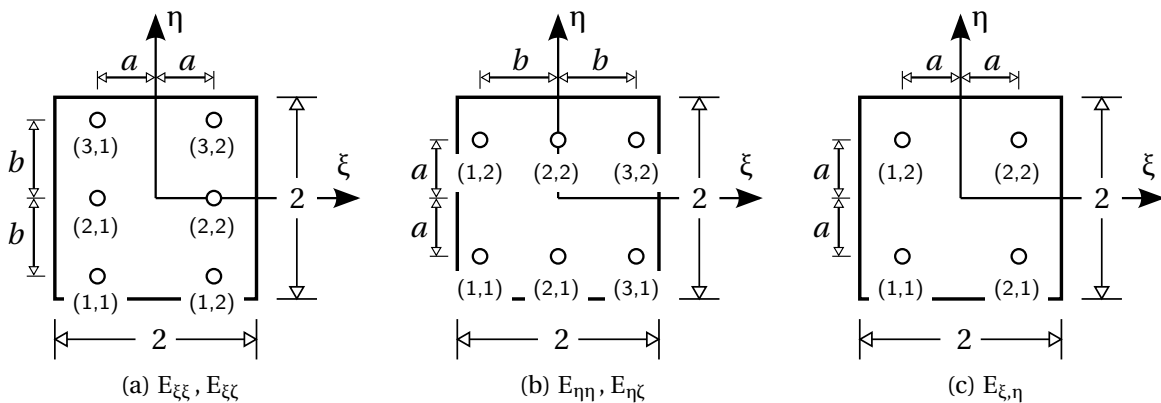


Abbildung 3.8.: Lage der Kollokationspunkte für biquadratische LAGRANGE-Elemente mit $a = \sqrt{1/3}$ und $b = \sqrt{3/5}$

welches nur bei der Behandlung gekrümmter Geometrien relevant ist. Analog zu den Ausführungen für die linearen Ansatzfunktionen besteht wieder die Möglichkeit, die Verzerrungen wie in Gleichungen (3.52) und (3.53) in der Mittelebene auszuwerten, oder sie wie in den Gleichungen (3.54) und (3.55) linear über die Dicke zu interpolieren. Beim Vergleich der verschiedenen Vorgehensweisen an Prinzipbeispielen im Zusammenhang mit vollständiger Integration ($3 \times 3 \times 2$ GAUSS-Punkte) ergab sich auch hier kein Unterschied bei den Abbildungseigenschaften. Insbesondere bei den Querschubverzerrungen liegt wegen der linearen Geometrie- und Verschiebungsinterpolation in ζ -Richtung auch hier ein linearer Verlauf in ζ vor. Für die Verzerrungsinterpolation der Normalverzerrungen in Schalenebene ergibt sich also

$$E_{\xi\xi}^a(\xi, \eta) = \sum_{i=1}^3 \sum_{j=1}^2 q_i(\eta) \ell_j(\xi) E_{\xi\xi}^{(i,j)} \quad (3.60)$$

$$E_{\eta\eta}^a(\xi, \eta) = \sum_{i=1}^3 \sum_{j=1}^2 q_i(\xi) \ell_j(\eta) E_{\eta\eta}^{(i,j)} \quad (3.61)$$

und für die Schub- und Querschubverzerrungen

$$E_{\xi\eta}^a(\xi, \eta) = \sum_{i=1}^2 \sum_{j=1}^2 \ell_i(\xi) \ell_j(\eta) E_{\xi\eta}^{(i,j)} \quad (3.62)$$

$$E_{\eta\zeta}^a(\xi, \eta) = \sum_{i=1}^3 \sum_{j=1}^2 q_i(\xi) \ell_j(\eta) E_{\eta\zeta}^{(i,j)} \quad (3.63)$$

$$E_{\zeta\xi}^a(\xi, \eta) = \sum_{i=1}^3 \sum_{j=1}^2 q_i(\eta) \ell_j(\xi) E_{\zeta\xi}^{(i,j)}, \quad (3.64)$$

mit den linearen und quadratischen Interpolationsfunktionen

$$q_i = \begin{cases} \frac{1}{2} \sqrt{\frac{5}{3}} \left(\sqrt{\frac{5}{3}} z - 1 \right) \\ 1 - \frac{5}{3} z^2 \\ \frac{1}{2} \sqrt{\frac{5}{3}} \left(\sqrt{\frac{5}{3}} z + 1 \right) \end{cases}, \quad \ell_i = \begin{cases} \frac{1}{2} (1 - \sqrt{3} z) \\ \frac{1}{2} (1 + \sqrt{3} z) \end{cases}, \quad z = \xi, \eta. \quad (3.65)$$

Die Lage der Kollokationspunkte für die einzelnen Verzerrungsanteile ist Abbildung 3.8 zu entnehmen.

Analog zur von BETSCH & STEIN [18] vorgeschlagenen Vorgehensweise können die Normalverzerrungen in Dickenrichtung wie bei den linearen Elementen an den Elementknoten der Schalenmittelfläche ausgewertet und mit den Ansatzfunktionen interpoliert werden. Es ergibt sich die Vorschrift

$$E_{\zeta\zeta}^a(\xi, \eta) = \sum_{i=1}^k N_i(\xi, \eta) E_{\zeta\zeta}(\xi = \xi_i, \eta = \eta_i, \zeta = 0), \quad (3.66)$$

mit $k = 9$, den quadratischen LAGRANGE-Ansätzen aus Gleichung (3.10) und den Knotenkoordinaten ξ_i und η_i aus Abbildung 3.4b. Für Serendipity-Elemente ist $k = 8$ und es sind entsprechend die Ansatzfunktionen aus Gleichung (3.11) sowie ξ_i und η_i aus Abbildung 3.4c zu verwenden.

Anwendung für Plastizität mit großen Dehnungen und großen Rotationen

Die vorigen Abschnitte zeigen die ANS-Methode als Modifikation der natürlichen Verzerrungen durch Auswertung an Kollokationspunkten und Interpolation mit speziellen Funktionen. Bei der Verwendung hyperelastischer Materialgesetze, die sich wie in Abschnitt 1.3.1 beschrieben vollständig in Verzerrungsgrößen (\mathbf{E} , \mathbf{C} etc.) angeben lassen, können die modifizierten Verzerrungsverläufe direkt in die jeweilige Materialbeschreibung eingesetzt werden. Auch eine Anwendung der beschriebenen Plastizitätsgesetze für kleine Deformationen ist ohne weitere Annahmen möglich, wie in Abbildung 1.2 deutlich wird. Bei der Formulierung für Plastizität mit endlichen Deformationen (siehe Abbildung 1.3) ist die Kinematik durch den Deformationsgradienten \mathbf{F} aus Gleichung (1.3) charakterisiert. Sollen also die aus der Verschiebung berechneten Verzerrungen durch ANS modifiziert werden, entspricht dies der Beziehung

$$\mathbf{E}^u \xrightarrow{\text{ANS}} \mathbf{E}^a \quad \text{mit} \quad \mathbf{E}^a = \frac{1}{2} (\mathbf{F}^{aT} \cdot \mathbf{F}^a - \mathbf{1}) \quad \Rightarrow \quad \mathbf{F}^a = ? \quad (3.67)$$

Aus diesem Zusammenhang lässt sich jedoch der für die Plastizitätsroutine erforderliche modifizierte Deformationsgradient \mathbf{F}^a nicht direkt bestimmen. Es existieren im Schrifttum verschiedene Ansätze, den Deformationsgradienten zu bestimmen, auf die im Folgenden kurz eingegangen werden soll.

Eine Möglichkeit der Bestimmung des modifizierten Deformationsgradienten \mathbf{F}^a aus Gleichung (3.67) geht auf die Arbeit von EBERLEIN [41] zurück und beruht auf der polaren Zerlegung in den eigentlich orthogonalen Drehtensor \mathbf{R} und den symmetrischen Rechtsstrecktensor \mathbf{U}

$$\mathbf{F} = \mathbf{R} \cdot \mathbf{U}. \quad (3.68)$$

Eingesetzt in Gleichung (1.6) bzw. (1.7) folgt für den rechten CAUCHY-GREEN-Tensor und somit für den GREEN-LAGRANGE-Verzerrungstensor

$$\mathbf{C} = \mathbf{F}^T \cdot \mathbf{F} = \mathbf{U}^T \cdot \underbrace{\mathbf{R}^T \cdot \mathbf{R}}_1 \cdot \mathbf{U} = \mathbf{U}^2 \quad \Rightarrow \quad \mathbf{E} = \frac{1}{2} (\mathbf{U}^2 - \mathbf{1}). \quad (3.69)$$

Die Verzerrungstensoren sind also vom Rotationstensor unabhängig, der modifizierte Deformationsgradient lässt sich mit dem ursprünglichen Rotationstensor und dem modifizierten Rechtsstrecktensor berechnen

$$\mathbf{F}^a = \mathbf{R} \cdot \mathbf{U}^a. \quad (3.70)$$

Es müssen nun die modifizierten und unmodifizierten Rechtsstrecktensoren berechnet werden. Dies kann über Spektralzerlegung aus den jeweiligen rechten CAUCHY-GREEN-Tensoren geschehen; eine ausführliche Diskussion über Spektraldarstellungen verschiedener Tensoren findet sich z. B. bei DOLL [38]. Es folgt

$$\mathbf{C} = \sum_{i=1}^3 \lambda_{Ci} \boldsymbol{\varphi}_{Ci} \otimes \boldsymbol{\varphi}_{Ci} \quad \Rightarrow \quad \mathbf{U} = \sum_{i=1}^3 \sqrt{\lambda_{Ci}} \boldsymbol{\varphi}_{Ci} \otimes \boldsymbol{\varphi}_{Ci} \quad (3.71)$$

$$\mathbf{C}^a = \sum_{i=1}^3 \lambda_{C^a i} \boldsymbol{\varphi}_{C^a i} \otimes \boldsymbol{\varphi}_{C^a i} \quad \Rightarrow \quad \mathbf{U}^a = \sum_{i=1}^3 \sqrt{\lambda_{C^a i}} \boldsymbol{\varphi}_{C^a i} \otimes \boldsymbol{\varphi}_{C^a i} \quad (3.72)$$

mit den Eigenwerten und Eigenvektoren des unmodifizierten $(\lambda_{C_i}, \boldsymbol{\varphi}_{C_i})$ und des modifizierten $(\lambda_{C^a_i}, \boldsymbol{\varphi}_{C^a_i})$ rechten CAUCHY-GREEN-Tensors

$$\mathbf{C} = 2\mathbf{E} + \mathbf{1} \quad \text{bzw.} \quad \mathbf{C}^a = 2\mathbf{E}^a + \mathbf{1}. \quad (3.73)$$

Es folgt schließlich durch Einsetzen der Gleichungen (3.71) und (3.72) in (3.70)

$$\mathbf{F}^a = \mathbf{F} \cdot \mathbf{U}^{-1} \cdot \mathbf{U}^a, \quad \text{mit} \quad \mathbf{R} = \mathbf{F} \cdot \mathbf{U}^{-1} \quad (3.74)$$

der modifizierte Deformationsgradient \mathbf{F}^a , der dann den Materialgleichungen übergeben werden kann.

Eine weitere Vorgehensweise zur Bestimmung des modifizierten Deformationsgradienten \mathbf{F}^a wird bei DVORKIN ET. AL. [40] beschrieben, wurde jedoch für die vorliegende Arbeit nicht umgesetzt und sei hier der Vollständigkeit halber angegeben. Sie basiert auf der Modifikation des HENCKY Verzerrungstensors, der sich ebenfalls über Spektralzerlegung aus dem rechten CAUCHY-GREEN-Tensor bestimmen lässt

$$\mathbf{H} = \ln \mathbf{U} = \ln \mathbf{C}^{1/2} = \sum_{i=1}^3 \ln(\sqrt{\lambda_{C_i}}) \boldsymbol{\varphi}_{C_i} \otimes \boldsymbol{\varphi}_{C_i}. \quad (3.75)$$

Nach einer Transformation in ein lokales konvexives Koordinatensystem kann dieser analog zu der für die GREEN-LAGRANGE-Verzerrungen beschriebenen Vorgehensweise an den Kollokationspunkten ausgewertet und mit den im vorigen Abschnitt eingeführten ANS-Ansätzen interpoliert werden. Es ergibt sich der modifizierte HENCKY-Verzerrungstensor

$$\mathbf{H} \xrightarrow{\text{ANS}} \mathbf{H}^a, \quad (3.76)$$

der schließlich über die Beziehung

$$\mathbf{F}^a = \mathbf{F} \cdot \mathbf{U}^{-1} \cdot \exp(\mathbf{H}^a), \quad \text{mit} \quad \mathbf{R} = \mathbf{F} \cdot \mathbf{U}^{-1} \quad (3.77)$$

wieder auf den modifizierten Deformationsgradienten \mathbf{F}^a führt.

Ein erheblicher Nachteil beider Methoden, ist die Notwendigkeit der Eigenvektoren des rechten CAUCHY-GREEN-Tensors, welche für die Implementierung nach dem in Anhang A.4 skizzierten Schema berechnet wurden. Neben dem immensen numerischen Aufwand für die Unterscheidung der verschiedenen Fälle für verschiedene/ gleiche Eigenwerte, stellte sich das Verfahren in der hier mit symbolischer Programmierung umgesetzten Form als numerisch wenig robust heraus. In praktischen Anwendungen – insbesondere bei hohen Geschwindigkeiten und Nichtlinearitäten – konnten also die Ergebnisse des herkömmlichen ANS-Elementes nicht befriedigend reproduziert werden. Da das oben beschriebene Vorgehen im Schrifttum bereits erfolgreich angewendet wurde, liegen wohl numerische Probleme insbesondere bei der Aufstellung der Eigenvektoren vor. Händisch umgesetzte Routinen lassen sich nur schwer mit dem im folgenden Kapitel beschriebenen Programmierkonzept kombinieren, ohne die Vorteile hinsichtlich der Effizienz einzubüßen. Eine effiziente Routine zur Bestimmung der Eigenvektoren mit Hilfe symbolischer Programmierung, liegt allerdings noch nicht vor. Es ist jedoch durch die erforderlichen Fallunterscheidungen neben dem höheren numerischen Aufwand der Materialformulierung auch ein deutlicher Anstieg der Simulationszeiten auf Grund der Implementierung der erforderlichen Spektralzerlegungen

zu erwarten. Wünschenswert wäre eine Umsetzung der ANS-Methode, die sich direkt auf den Deformationsgradienten anwenden lässt, also

$$\mathbf{F} \xrightarrow{\text{ANS}} \mathbf{F}^a, \quad (3.78)$$

eine solche Modifikation konnte jedoch nicht allgemein formuliert werden. Die Möglichkeit der direkten Bestimmung von \mathbf{F}^a über eine erweiterte Verschiebungsinterpolation, also ohne Modifikation der Verzerrungen, wird im nächsten Abschnitt diskutiert.

3.4.3. Die Methode der erweiterten Verschiebungen

Allgemeines

Bei der *Methode der angenommenen Verzerrungen* (ANS) werden die Versteifungseffekte durch Modifikationen an den einzelnen Verzerrungsverläufen eliminiert, was zu den im vorigen Abschnitt besprochenen Problemen bei der effizienten Umsetzung für Plastizität mit großen Deformationen führt. Eine Möglichkeit, die sich insbesondere mit Hilfe der Anwendung symbolischer Programmierung ergibt, besteht in der Modifikation der Verschiebungsinterpolation, sodass sich aus dieser gerade die ANS-Verzerrungsverläufe ergeben. Im folgenden Abschnitt wird ein solches Vorgehen für die linearen Komponenten der Querschubverzerrungen $E_{\xi\zeta}$ und $E_{\eta\zeta}$ einer 8-Knoten Solid-Shell-Formulierung hergeleitet und dessen Anwendbarkeit auf große Deformationen diskutiert. Auf eine ähnliche Strategie, die im Schrifttum mit 'linked interpolation' bezeichnet wird, sei hier aus Gründen der Vollständigkeit hingewiesen [122, 132, 137].

Zur übersichtlicheren Darstellung wird im Laufe der Herleitung die Abkürzung ($\text{---} \parallel \text{---}$) verwendet, welches als eine Wiederholung des in der darüber liegenden Zeile stehenden Ausdrucks zu verstehen ist.

Verzerrungsverläufe mit und ohne ANS

Zunächst wird exemplarisch für eine allgemeine Elementgeometrie

$$\mathbf{x}l = \begin{pmatrix} \bar{\mathbf{X}} \\ \bar{\mathbf{Y}} \\ \bar{\mathbf{Z}} \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 \\ z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \end{pmatrix}, \quad (3.79)$$

welche einem beliebigen Verschiebungszustand

$$\mathbf{u}l = \begin{pmatrix} \bar{\mathbf{u}} \\ \bar{\mathbf{v}} \\ \bar{\mathbf{w}} \end{pmatrix} = \begin{pmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 \\ v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 \\ w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 & w_8 \end{pmatrix} \quad (3.80)$$

ausgesetzt ist, die Querschubverzerrung $E_{\xi\zeta}$ aufgestellt. Diese wird anschließend wie in Abschnitt 3.4.2 beschrieben ausgewertet und interpoliert, sodass der ursprüngliche Verlauf, der die versteifenden Anteile beinhaltet, mit dem versteifungsfreien ANS-Verlauf verglichen

werden kann. Zur übersichtlichen Darstellung empfiehlt sich die Einführung der Interpolationsvektoren

$$\mathbf{A}_0 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad \mathbf{B}_\xi = \begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \\ -1 \end{pmatrix}, \quad \mathbf{B}_\eta = \begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \\ 1 \end{pmatrix}, \quad \mathbf{B}_\zeta = \begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad (3.81)$$

und

$$\mathbf{C}_{\xi\eta} = \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}, \quad \mathbf{C}_{\xi\zeta} = \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \\ 1 \\ 1 \\ -1 \end{pmatrix}, \quad \mathbf{C}_{\eta\zeta} = \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \\ -1 \\ -1 \\ 1 \\ 1 \end{pmatrix}, \quad \mathbf{D}_{\xi\eta\zeta} = \begin{pmatrix} -1 \\ 1 \\ -1 \\ 1 \\ 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}. \quad (3.82)$$

Die Geometrieinterpolation nach Gleichung (3.33) lautet unter Anwendung dieser Vektoren

$$\begin{aligned} X &= \frac{1}{8} \left(\mathbf{A}_0 + \mathbf{B}_\xi \xi + \mathbf{B}_\eta \eta + \mathbf{B}_\zeta \zeta + \mathbf{C}_{\xi\eta} \xi \eta + \mathbf{C}_{\xi\zeta} \xi \zeta + \mathbf{C}_{\eta\zeta} \eta \zeta + \mathbf{D}_{\xi\eta\zeta} \xi \eta \zeta \right) \cdot \bar{\mathbf{X}} \\ Y &= \frac{1}{8} \left(\begin{array}{ccc} & - & \\ & \parallel & - \\ & & \end{array} \right) \cdot \bar{\mathbf{Y}} \\ Z &= \frac{1}{8} \left(\begin{array}{ccc} & - & \\ & \parallel & - \\ & & \end{array} \right) \cdot \bar{\mathbf{Z}}, \end{aligned} \quad (3.83)$$

für die Verschiebungsinterpolation aus Gleichung (3.34) ergibt sich analog

$$\begin{aligned} u &= \frac{1}{8} \left(\mathbf{A}_0 + \mathbf{B}_\xi \xi + \mathbf{B}_\eta \eta + \mathbf{B}_\zeta \zeta + \mathbf{C}_{\xi\eta} \xi \eta + \mathbf{C}_{\xi\zeta} \xi \zeta + \mathbf{C}_{\eta\zeta} \eta \zeta + \mathbf{D}_{\xi\eta\zeta} \xi \eta \zeta \right) \cdot \bar{\mathbf{u}} \\ v &= \frac{1}{8} \left(\begin{array}{ccc} & - & \\ & \parallel & - \\ & & \end{array} \right) \cdot \bar{\mathbf{v}} \\ w &= \frac{1}{8} \left(\begin{array}{ccc} & - & \\ & \parallel & - \\ & & \end{array} \right) \cdot \bar{\mathbf{w}}. \end{aligned} \quad (3.84)$$

Der lineare Anteil der Querschubverzerrung $E_{\xi\zeta,lin}$ ist nach Gleichung (1.12)

$$E_{\xi\zeta,lin} = \frac{1}{2} \left(\frac{\partial u}{\partial \xi} \frac{\partial X}{\partial \zeta} + \frac{\partial v}{\partial \xi} \frac{\partial Y}{\partial \zeta} + \frac{\partial w}{\partial \xi} \frac{\partial Z}{\partial \zeta} + \frac{\partial u}{\partial \zeta} \frac{\partial X}{\partial \xi} + \frac{\partial v}{\partial \zeta} \frac{\partial Y}{\partial \xi} + \frac{\partial w}{\partial \zeta} \frac{\partial Z}{\partial \xi} \right). \quad (3.85)$$

Einsetzen der Vektoren aus den Gleichungen (3.83) und (3.84) ergibt die Form

$$\begin{aligned}
E_{\xi\zeta,\text{lin}} = \frac{1}{128} & \left[\left(\mathbf{B}_\xi + \mathbf{C}_{\xi\eta}\eta + \mathbf{C}_{\xi\zeta}\zeta + \mathbf{D}_{\xi\eta\zeta}\eta\zeta \right) \cdot \bar{\mathbf{u}} \left(\mathbf{B}_\zeta + \mathbf{C}_{\xi\zeta}\xi + \mathbf{C}_{\eta\zeta}\eta + \mathbf{D}_{\xi\eta\zeta}\xi\zeta \right) \cdot \bar{\mathbf{X}} \right. \\
& + \left(\quad \quad \quad \right) \cdot \bar{\mathbf{v}} \left(\quad \quad \quad \right) \cdot \bar{\mathbf{Y}} \\
& + \left(\quad \quad \quad \right) \cdot \bar{\mathbf{w}} \left(\quad \quad \quad \right) \cdot \bar{\mathbf{Z}} \\
& + \left(\mathbf{B}_\zeta + \mathbf{C}_{\xi\zeta}\xi + \mathbf{C}_{\eta\zeta}\eta + \mathbf{D}_{\xi\eta\zeta}\xi\zeta \right) \cdot \bar{\mathbf{u}} \left(\mathbf{B}_\xi + \mathbf{C}_{\xi\eta}\eta + \mathbf{C}_{\xi\zeta}\zeta + \mathbf{D}_{\xi\eta\zeta}\eta\zeta \right) \cdot \bar{\mathbf{X}} \\
& + \left(\quad \quad \quad \right) \cdot \bar{\mathbf{v}} \left(\quad \quad \quad \right) \cdot \bar{\mathbf{Y}} \\
& \left. + \left(\quad \quad \quad \right) \cdot \bar{\mathbf{w}} \left(\quad \quad \quad \right) \cdot \bar{\mathbf{Z}} \right], \tag{3.86}
\end{aligned}$$

auf welche nun die ANS-Methode angewendet werden kann.

Es erfolgt zunächst eine Auswertung an den Kollokationspunkten aus Abbildung 3.7, wobei für die Koordinate ζ gemäß Gleichung (3.58) kein konkreter Wert eingesetzt wird. Die beiden Werte sind

$$\begin{aligned}
E_{\xi\zeta,\text{lin}}^{(2,1)} = \frac{1}{128} & \left[\left(\mathbf{B}_\xi - \mathbf{C}_{\xi\eta} + \mathbf{C}_{\xi\zeta}\zeta - \mathbf{D}_{\xi\eta\zeta}\zeta \right) \cdot \bar{\mathbf{u}} \left(\mathbf{B}_\zeta - \mathbf{C}_{\eta\zeta} \right) \cdot \bar{\mathbf{X}} \right. \\
& + \left(\quad \quad \quad \right) \cdot \bar{\mathbf{v}} \left(\quad \quad \quad \right) \cdot \bar{\mathbf{Y}} \\
& + \left(\quad \quad \quad \right) \cdot \bar{\mathbf{w}} \left(\quad \quad \quad \right) \cdot \bar{\mathbf{Z}} \\
& + \left(\mathbf{B}_\zeta - \mathbf{C}_{\eta\zeta} \right) \cdot \bar{\mathbf{u}} \left(\mathbf{B}_\xi - \mathbf{C}_{\xi\eta} + \mathbf{C}_{\xi\zeta}\zeta - \mathbf{D}_{\xi\eta\zeta}\zeta \right) \cdot \bar{\mathbf{X}} \\
& + \left(\quad \quad \quad \right) \cdot \bar{\mathbf{v}} \left(\quad \quad \quad \right) \cdot \bar{\mathbf{Y}} \\
& \left. + \left(\quad \quad \quad \right) \cdot \bar{\mathbf{w}} \left(\quad \quad \quad \right) \cdot \bar{\mathbf{Z}} \right] \tag{3.87}
\end{aligned}$$

und

$$\begin{aligned}
E_{\xi\zeta,\text{lin}}^{(2,2)} = \frac{1}{128} & \left[\left(\mathbf{B}_\xi + \mathbf{C}_{\xi\eta} + \mathbf{C}_{\xi\zeta}\zeta + \mathbf{D}_{\xi\eta\zeta}\zeta \right) \cdot \bar{\mathbf{u}} \left(\mathbf{B}_\zeta + \mathbf{C}_{\eta\zeta} \right) \cdot \bar{\mathbf{X}} \right. \\
& + \left(\quad \quad \quad \right) \cdot \bar{\mathbf{v}} \left(\quad \quad \quad \right) \cdot \bar{\mathbf{Y}} \\
& + \left(\quad \quad \quad \right) \cdot \bar{\mathbf{w}} \left(\quad \quad \quad \right) \cdot \bar{\mathbf{Z}} \\
& + \left(\mathbf{B}_\zeta + \mathbf{C}_{\eta\zeta} \right) \cdot \bar{\mathbf{u}} \left(\mathbf{B}_\xi + \mathbf{C}_{\xi\eta} + \mathbf{C}_{\xi\zeta}\zeta + \mathbf{D}_{\xi\eta\zeta}\zeta \right) \cdot \bar{\mathbf{X}} \\
& + \left(\quad \quad \quad \right) \cdot \bar{\mathbf{v}} \left(\quad \quad \quad \right) \cdot \bar{\mathbf{Y}} \\
& \left. + \left(\quad \quad \quad \right) \cdot \bar{\mathbf{w}} \left(\quad \quad \quad \right) \cdot \bar{\mathbf{Z}} \right]. \tag{3.88}
\end{aligned}$$

und

$$w_c^{\eta\zeta,2} = (1 - \xi_{GP}^2) \left[\begin{aligned} & \mathbf{C}_{\xi\zeta} \cdot \bar{\mathbf{X}} (\mathbf{C}_{\xi\eta} + \mathbf{D}_{\xi\eta\zeta} \zeta_{GP}) \cdot \bar{\mathbf{u}} \\ & + \mathbf{C}_{\xi\zeta} \cdot \bar{\mathbf{Y}} (\mathbf{C}_{\xi\eta} + \mathbf{D}_{\xi\eta\zeta} \zeta_{GP}) \cdot \bar{\mathbf{v}} \\ & + \mathbf{C}_{\xi\zeta} \cdot \bar{\mathbf{Z}} (\mathbf{C}_{\xi\eta} + \mathbf{D}_{\xi\eta\zeta} \zeta_{GP}) \cdot \bar{\mathbf{w}} \end{aligned} \right], \quad (3.97)$$

und schließlich auf die Erweiterung

$$w_{\text{add}}^{\eta\zeta} = -\frac{1}{8} \frac{1}{\mathbf{J}_{GP}^{33}} \left[\begin{aligned} & \mathbf{J}_{GP}^{21} \left(\mathbf{C}_{\eta\zeta} \frac{1}{3} P_2(\eta) + \mathbf{D}_{\xi\eta\zeta} \frac{1}{3} P_1(\xi) P_2(\eta) \right) \cdot \bar{\mathbf{u}} \\ & + \mathbf{J}_{GP}^{22} \left(\begin{array}{ccc} - & \parallel & - \end{array} \right) \cdot \bar{\mathbf{v}} \\ & + \mathbf{J}_{GP}^{23} \left(\begin{array}{ccc} - & \parallel & - \end{array} \right) \cdot \bar{\mathbf{w}} - \frac{w_c^{\eta\zeta,1}}{8} \eta \\ & + \nabla \mathbf{u}_{GP}^{21} \left(\begin{array}{ccc} - & \parallel & - \end{array} \right) \cdot \bar{\mathbf{X}} \\ & + \nabla \mathbf{u}_{GP}^{22} \left(\begin{array}{ccc} - & \parallel & - \end{array} \right) \cdot \bar{\mathbf{Y}} \\ & + \nabla \mathbf{u}_{GP}^{23} \left(\begin{array}{ccc} - & \parallel & - \end{array} \right) \cdot \bar{\mathbf{Z}} - \frac{w_c^{\eta\zeta,2}}{8} \eta \end{aligned} \right]. \quad (3.98)$$

Mit der Verschiebung w aus Gleichung (3.84) lautet nun die zur Beseitigung der Querschubversteifung erforderliche Verschiebungsinterpolation

$$w_{\text{add}} = w + w_{\text{add}}^{\xi\zeta} + w_{\text{add}}^{\eta\zeta}. \quad (3.99)$$

Durch die Eigenschaft LEGENDRE-Polynome zweiter Ordnung

$$P_2(\pm 1/\sqrt{3}) = 0, \quad (3.100)$$

verschwinden deren Funktionswerte gerade an den Integrationspunkten der hier verwendeten $2 \times 2 \times 2$ Punkt GAUSS-Integration. Daher hat der Term $w_{\text{add}}^{\xi\zeta}$ am Integrationspunkt keinen Einfluss auf $E_{\eta\zeta}$ und umgekehrt. Durch Einsetzen dieser erweiterten Verschiebungsinterpolation in Gleichung (3.85) ergeben sich so an den Integrationspunkten für eine beliebige Elementgeometrie und einen allgemeinen Verschiebungszustand exakt die Werte der ANS-interpolierten Querschubverzerrungen. Es ist also im symbolischen Programmierungskonzept keine Modifikation der Verzerrungen erforderlich, da sich diese direkt aus den Verschiebungen berechnen lassen.

Allgemeingültigkeit der Methode

Für den Fall einer ebenen Elementgeometrie, also bei konstanter Dicke und Zusammenfallen der z - und der ζ -Richtung, gilt für die Einträge der JACOBI-Matrix

$$\mathbf{J}^{13} = \frac{\partial Z}{\partial \xi} = 0 \quad \text{und} \quad \mathbf{J}^{23} = \frac{\partial Z}{\partial \eta} = 0. \quad (3.101)$$

Für diesen Fall und die Annahme kleiner Deformationen liefert die hergeleitete Vorgehensweise exakt die ANS-interpolierten Querschubverzerrungen, ohne dabei die anderen Verzerrungskomponenten zu beeinflussen. Liegt keine ebene Elementgeometrie vor, so geht die modifizierte Verschiebungsinterpolation w_{add} beispielsweise in die Normalverzerrung $E_{\xi\xi}$ mit

$$E_{\xi\xi} = \frac{\partial u}{\partial \xi} \frac{\partial X}{\partial \xi} + \frac{\partial v}{\partial \xi} \frac{\partial Y}{\partial \xi} + \frac{\partial w}{\partial \xi} \frac{\partial Z}{\partial \xi} \quad (3.102)$$

ein. Welchen Einfluss diese Veränderung des Normalverzerrungsverlaufs auf das Systemverhalten hat, wird versucht, in Kapitel 5 an konkreten Beispielen zu belegen.

Ebenso können die durch die ANS-Methode modifizierten nichtlinearen Anteile der Verzerrungen nicht exakt reproduziert werden. Es ist nach Gleichung (1.12)

$$E_{\xi\zeta, \text{nl}} = \frac{1}{2} \left(\frac{\partial u}{\partial \xi} \frac{\partial u}{\partial \zeta} + \frac{\partial v}{\partial \xi} \frac{\partial v}{\partial \zeta} + \frac{\partial w}{\partial \xi} \frac{\partial w}{\partial \zeta} \right). \quad (3.103)$$

Bei der ANS-Interpolation werden durch das Auswerten der Komponenten an den Kollokationspunkten und die lineare Interpolation in η -Richtung alle in ξ und η nichtlinearen Terme eliminiert. Dies kann mit der oben beschriebenen Modifikation des Verschiebungsverlaufes nicht allgemein erreicht werden. Die Grenzen des Vorgehens bei nichtlinearen Problem wird ebenfalls in Kapitel 5 an einem Prinzipbeispiel diskutiert.

Sollte es möglich sein, in der beschriebenen Vorgehensweise eine Beseitigung der Versteifungseffekte auch bei nichtlinearen Verzerrungen zu erreichen, erübrigt sich die im Rahmen der ANS-Methode diskutierte Problematik bei der Anwendung von Materialgesetzen mit finiter Plastizität. Die Vorgabe einer entsprechend modifizierten Verschiebungsinterpolation erlaubt dann eine direkte Bestimmung des Locking-freien Deformationsgradienten ohne den ‘Umweg’ über die Verzerrungen. Dies führt zu einer deutlich effizienteren Elementformulierung, weshalb dieser vielversprechende Weg insbesondere für explizite Zeitintegration weiterverfolgt werden sollte. Ebenfalls offen ist die Behandlung der Normalverzerrungen in Dickenrichtung, für die eine eigene Interpolation in Gleichung (3.59) angegeben wurde. Der vorliegende Fall liefert unmodifizierte Verzerrungen $E_{\zeta\zeta}$, der entsprechende Versteifungseffekt wird so also nicht behoben. Es muss daher auch noch untersucht werden, inwiefern weitere Modifikationen an der Verschiebungsinterpolation möglich sind.

3.4.4. Die Methode der erweiterten Verzerrungen

Entwicklung

Die *Methode der erweiterten Verzerrungen* (*Enhanced Assumed Strains – EAS*) basiert auf der Einführung zusätzlicher Freiheitsgrade auf Elementebene zur Erweiterung des Verzerrungsfeldes. Ziel der Erweiterung ist die Eliminierung künstlicher Versteifungseffekte auf Elementebene, es sollen also die für einen versteifungsfreien Verzerrungsverlauf fehlenden Anteile additiv eingefügt werden. Die globale Anzahl der Freiheitsgrade soll hierbei nicht verändert werden; die zusätzlich eingeführten Freiheitsgrade existieren nur auf Elementebene und werden bei der Aufstellung der Elementmatrizen – also vor der Assemblierung der Systemmatrizen – durch Kondensation eliminiert.

Die EAS-Methode geht zurück auf die Arbeit von SIMO & RIFAI [117] und wurde in verschiedenen Publikationen weiterentwickelt [113, 114]. Eine Grundlage hierfür wurde bereits durch TAYLOR ET. AL. [123] durch die *Methode der inkompatiblen Moden* formuliert, auf die in Abschnitt 3.4.5 noch genauer eingegangen wird. Nach ausgiebigen Diskussionen über die mathematische Konsistenz der von TAYLOR ET. AL. vorgeschlagenen Strategie (siehe hierzu [117, 119]), konnte schließlich durch SIMO & RIFAI eine variationelle Grundlage für die EAS-Methode formuliert werden.

Variationelle Basis

Die von SIMO & RIFAI [117] vorgestellte und in unzähligen Publikationen aufgegriffene variationelle Motivation der EAS-Methode basiert auf der Erweiterung des auf der Verschiebungsinterpolation basierenden Verzerrungsfeldes \mathbf{E}^u , das hier als kompatibles Verzerrungsfeld bezeichnet wird. Es werden inkompatible Verzerrungen $\tilde{\mathbf{E}}$ addiert, die von zusätzlichen (inkompatiblen) Freiheitsgraden $\boldsymbol{\alpha}$ abhängen:

$$\mathbf{E} = \mathbf{E}^u(\mathbf{u}) + \tilde{\mathbf{E}}(\boldsymbol{\alpha}). \quad (3.104)$$

Wie zuvor bei der Formulierung der ANS-Methode, werden die erweiterten Verzerrungen in das HU-WASHIZU-Dreifeldfunktional (1.51) eingesetzt. Es ergibt sich

$$\begin{aligned} \Pi_{\text{HW}} &= \int_{\mathcal{B}_0} \mathcal{W}(\mathbf{E}) + \mathbf{S} : (\mathbf{E}^u - \mathbf{E}) \, dV - \left[\int_{\mathcal{B}_0} \rho_0 \mathbf{b} \cdot \mathbf{u} \, dV + \int_{\partial \mathcal{B}_\sigma} \bar{\mathbf{t}} \cdot \mathbf{u} \, dA \right] + \int_{\mathcal{B}_0} \rho_0 \ddot{\mathbf{u}} \cdot \mathbf{u} \, dV \\ &= \int_{\mathcal{B}_0} \mathcal{W}(\mathbf{E}^u + \tilde{\mathbf{E}}) + \mathbf{S} : (\mathbf{E}^u - \mathbf{E}^u - \tilde{\mathbf{E}}) \, dV - \left[\int_{\mathcal{B}_0} \rho_0 \mathbf{b} \cdot \mathbf{u} \, dV + \int_{\partial \mathcal{B}_\sigma} \bar{\mathbf{t}} \cdot \mathbf{u} \, dA \right] + \int_{\mathcal{B}_0} \rho_0 \ddot{\mathbf{u}} \cdot \mathbf{u} \, dV \\ &= \int_{\mathcal{B}_0} \mathcal{W}(\mathbf{E}^u + \tilde{\mathbf{E}}) - \mathbf{S} : \tilde{\mathbf{E}} \, dV - \left[\int_{\mathcal{B}_0} \rho_0 \mathbf{b} \cdot \mathbf{u} \, dV + \int_{\partial \mathcal{B}_\sigma} \bar{\mathbf{t}} \cdot \mathbf{u} \, dA \right] + \int_{\mathcal{B}_0} \rho_0 \ddot{\mathbf{u}} \cdot \mathbf{u} \, dV. \end{aligned} \quad (3.105)$$

SIMO & RIFAI formulieren in [117] eine Orthogonalitätsbedingung

$$\int_{\mathcal{B}_0} \mathbf{S} : \tilde{\mathbf{E}} \, dV = 0, \quad (3.106)$$

die unabhängigen Spannungen werden also gerade so gewählt, dass sie mit den erweiterten Verzerrungen keinen Beitrag zur Gesamtenergie liefern. Dadurch werden sie aus dem Funktional eliminiert und das Dreifeld- reduziert sich zu einem modifizierten Zweifeld-Funktional:

$$\Pi_{\text{HW}}^{\text{EAS}} = \underbrace{\int_{\mathcal{B}_0} \mathcal{W}(\mathbf{E}^u + \tilde{\mathbf{E}}) \, dV}_{\Pi_{\text{HW}}^{\text{int}}(\mathbf{u}, \tilde{\mathbf{E}})} - \underbrace{\left[\int_{\mathcal{B}_0} \rho_0 \mathbf{b} \cdot \mathbf{u} \, dV + \int_{\partial \mathcal{B}_\sigma} \bar{\mathbf{t}} \cdot \mathbf{u} \, dA \right]}_{\Pi^{\text{ext}}(\mathbf{u})} + \int_{\mathcal{B}_0} \rho_0 \ddot{\mathbf{u}} \cdot \mathbf{u} \, dV \rightarrow \text{stat.},$$

für $0 \leq t \leq T$ (3.107)

Eine ausführlichere Diskussion zur Bedeutung der Orthogonalitätsbedingung (3.106) ist z. B. bei BISCHOFF [19] angegeben. Um das Bestehen des Patchtests zu gewährleisten, fordern SIMO & RIFAI in [117] die Erfüllung von Gleichung (3.106) zumindest für stückweise konstante Spannungsverläufe, sie reduziert sich also zu

$$\int_{\mathcal{B}_0} \tilde{\mathbf{E}} \, dV = 0. \quad (3.108)$$

Um eine Abhängigkeit der erweiterten Verzerrungen von einer veränderlichen Metrik zu vermeiden, werden die Verzerrungsansätze auf eine konstante Metrik in Elementmitte bezogen. Damit ergibt sich für obige Bedingung

$$\int_{\mathcal{B}_0} \tilde{\mathbf{E}} \, dV = \int_{\mathcal{B}_0} \tilde{E}_{ij} \mathbf{G}^i \otimes \mathbf{G}^j \det \mathbf{J} \, dV = \int_{\mathcal{B}_0} \tilde{E}_{kl}^0 \mathbf{G}_0^k \otimes \mathbf{G}_0^l \det \mathbf{J}_0 \, dV = 0 \quad (3.109)$$

mit der kontravarianten Metrik $\mathbf{G}_0^k \otimes \mathbf{G}_0^l$ und der Determinante der JACOBI Matrix $\det \mathbf{J}_0$ jeweils bezogen auf die Elementmitte. Für die Transformation der erweiterten Verzerrungen in das konvektive Koordinatensystem, in dem auch die kompatiblen Verzerrungen definiert sind, ergibt sich schließlich

$$\tilde{E}_{ij} = \frac{\det \mathbf{J}_0}{\det \mathbf{J}} \tilde{E}_{kl}^0 (\mathbf{G}_0^k \cdot \mathbf{G}_i) (\mathbf{G}_j \cdot \mathbf{G}_0^l). \quad (3.110)$$

Die Komponenten der inkompatiblen Verzerrungen werden auf Elementebene durch Multiplikation einer Ansatzmatrix $\mathbf{N}_{\text{EAS}}(\xi, \eta, \zeta)$ mit den erweiterten Freiheitsgraden $\boldsymbol{\alpha}$ berechnet. Beispiele für die Wahl von \mathbf{N}_{EAS} werden bei der Beschreibung der Elementformulierungen in Abschnitt 4.4 angegeben. Für die Implementierung beispielsweise eines Volumenschalenelementes ist es praktisch, die Verzerrungskomponenten in einem Vektor

$$\boldsymbol{\varepsilon} = [E_{\xi\xi} \quad E_{\eta\eta} \quad 2E_{\xi\eta} \quad E_{\zeta\zeta} \quad 2E_{\eta\zeta} \quad 2E_{\xi\zeta}]^T \quad (3.111)$$

anzuordnen, die Matrix \mathbf{N}_{EAS} besitzt dann 6 Zeilen – bei degenerierten Schalenelementen ohne Normalverzerrung in Dickenrichtung sind es entsprechend 5 – und so viele Spalten, wie EAS-Parameter gewählt werden. Die i -te Spalte dieser Matrix lässt sich beispielsweise für Kontinuums-elemente mit linearen Ansatzfunktionen allgemein schreiben als

$$\mathbf{N}_{\text{EAS},i} = \begin{pmatrix} a_{100} \xi + a_{010} \eta + a_{001} \zeta + a_{110} \xi \eta + a_{101} \xi \zeta + a_{011} \eta \zeta + a_{111} \xi \eta \zeta \\ b_{100} \xi + b_{010} \eta + b_{001} \zeta + b_{110} \xi \eta + b_{101} \xi \zeta + b_{011} \eta \zeta + b_{111} \xi \eta \zeta \\ c_{100} \xi + c_{010} \eta + c_{001} \zeta + c_{110} \xi \eta + c_{101} \xi \zeta + c_{011} \eta \zeta + c_{111} \xi \eta \zeta \\ d_{100} \xi + d_{010} \eta + d_{001} \zeta + d_{110} \xi \eta + d_{101} \xi \zeta + d_{011} \eta \zeta + d_{111} \xi \eta \zeta \\ e_{100} \xi + e_{010} \eta + e_{001} \zeta + e_{110} \xi \eta + e_{101} \xi \zeta + e_{011} \eta \zeta + e_{111} \xi \eta \zeta \\ f_{100} \xi + f_{010} \eta + f_{001} \zeta + f_{110} \xi \eta + f_{101} \xi \zeta + f_{011} \eta \zeta + f_{111} \xi \eta \zeta \end{pmatrix}. \quad (3.112)$$

Die Parameter a_{klm} bis f_{klm} , mit $k, \ell, m = \{0, 1\}$ sind Konstanten durch deren Wahl sich beliebige Ansätze für die erweiterten Verzerrungen formulieren lassen. Da sie für jede Spalte der Matrix \mathbf{N}_{EAS} unterschiedlich sind, müsste jede einzelne Konstante in Gleichung (3.112) den Index i tragen. Auf die Indizierung wurde aus Gründen der Übersichtlichkeit an dieser Stelle verzichtet. Bei Elementen höherer Ansatzordnung können auch höhere Ordnungen

bei den erweiterten Verzerrungen sinnvoll sein, es treten dann also entsprechend auch $a_{k\ell m}$ bis $f_{k\ell m}$, mit $k, \ell, m = \{0, 1, 2, \dots\}$ auf.

Die Anwendung der ersten Variation auf Gleichung (3.107) führt auf

$$\begin{aligned}
 \delta \Pi_{\text{HW}}^{\text{EAS}} &= \\
 &= \int_{\mathcal{B}_0} \delta(\mathbf{E}^u + \tilde{\mathbf{E}}) \frac{\partial \mathcal{W}}{\partial(\mathbf{E}^u + \tilde{\mathbf{E}})} dV - \left[\int_{\mathcal{B}_0} \rho_0 \mathbf{b} \cdot \delta \mathbf{u} dV + \int_{\partial \mathcal{B}_\sigma} \bar{\mathbf{t}} \cdot \delta \mathbf{u} dA \right] + \int_{\mathcal{B}_0} \rho_0 \ddot{\mathbf{u}} \cdot \delta \mathbf{u} dV \\
 &= \delta(\mathbf{E}^u + \tilde{\mathbf{E}}) : \mathbf{S}^E dV - \left[\int_{\mathcal{B}_0} \rho_0 \mathbf{b} \cdot \delta \mathbf{u} dV + \int_{\partial \mathcal{B}_\sigma} \bar{\mathbf{t}} \cdot \delta \mathbf{u} dA \right] + \int_{\mathcal{B}_0} \rho_0 \ddot{\mathbf{u}} \cdot \delta \mathbf{u} dV, \quad (3.113)
 \end{aligned}$$

wobei der Index bei den Spannungen \mathbf{S}^E deren Abhängigkeit von den kompatiblen und den erweiterten Verzerrungen deutlich machen soll.

Bei der Diskretisierung im Rahmen der Methode der finiten Elemente werden die kontinuierlichen Verschiebungen \mathbf{u} wie in den Gleichungen (3.3) und (3.34) angegeben durch Interpolation der diskreten Knotenverschiebungen \mathbf{d} ausgedrückt. Die erweiterten Verzerrungen werden wie bereits erwähnt durch die im Vektor $\boldsymbol{\alpha}$ angeordneten zusätzlichen Freiheitsgrade repräsentiert. Die Variation des Funktionals gemäß Gleichung (1.49) lautet im diskretisierten Fall also allgemein

$$\delta \Pi^{\text{EAS}}(\mathbf{d}, \boldsymbol{\alpha}) = \delta \Pi^{\text{int}}(\mathbf{d}, \boldsymbol{\alpha}) - \delta \Pi^{\text{ext}}(\mathbf{d}) + \mathbf{M} \cdot \ddot{\mathbf{d}} \cdot \delta \mathbf{d}. \quad (3.114)$$

Da sich die Erweiterung der Verzerrungen lediglich auf die inneren Anteile auswirkt, ist eine Kondensation der EAS-Parameter auf Elementebene möglich. Dadurch bleibt die Anzahl der Freiheitsgrade auf Systemebene unverändert, es ändert sich jedoch der erforderliche Rechenaufwand auf Elementebene.

Aus Gleichung (3.114) folgt die Beziehung

$$\delta \Pi^{\text{EAS}}(\mathbf{d}, \boldsymbol{\alpha}) = \delta \mathbf{d}^T \cdot \frac{\partial \Pi^{\text{int}}}{\partial \mathbf{d}} + \delta \boldsymbol{\alpha}^T \cdot \frac{\partial \Pi^{\text{int}}}{\partial \boldsymbol{\alpha}} - \delta \mathbf{d}^T \cdot \frac{\partial \Pi^{\text{ext}}}{\partial \mathbf{d}} + \delta \mathbf{d}^T \cdot \mathbf{M} \cdot \ddot{\mathbf{d}} = 0, \quad (3.115)$$

welche nach dem Fundamentallema der Variationsrechnung für beliebige $\delta \mathbf{d} \neq 0$ und $\delta \boldsymbol{\alpha} \neq 0$ erfüllt sein muss. Innerhalb des jeweiligen Zeitintegrationsverfahrens unterscheiden sich die daraus resultierenden, jeweils auf Elementebene zu erfüllenden Bedingungen. In den folgenden Abschnitten, wird das Vorgehen für implizite und explizite Zeitintegration erläutert.

EAS für implizite Zeitintegration

Im Zeitschritt $n \rightarrow n+1$ eines impliziten Zeitintegrationsalgorithmus, wie dem in Abschnitt 2.3 beschriebenen NEWMARK-Verfahren, werden zur Berechnung der Verschiebungen \mathbf{d}_{n+1} wie in Gleichung (2.25) angegeben, die Knotenkräfte und die tangentielle Steifigkeitsmatrix zum Zeitschritt $n+1$ benötigt. Das dafür erforderliche Funktional

$$\Pi_{n+1}^{\text{EAS}}(\mathbf{d}_{n+1}, \boldsymbol{\alpha}_{n+1}) \quad (3.116)$$

hängt wiederum implizit von den unbekanntem Verschiebungen \mathbf{d}_{n+1} und den unbekanntem erweiterten Verzerrungen $\boldsymbol{\alpha}_{n+1}$ ab. Die Kondensation der EAS-Parameter auf Elementebene erfordert also eine Linearisierung von Gleichung (3.115) bezüglich \mathbf{d}_{n+1} und $\boldsymbol{\alpha}_{n+1}$. Die Anwendung des NEWMARK-Verfahrens erfolgt wie in Abschnitt 2.3, wobei das Gleichungssystem $\mathcal{G}(\mathbf{d}_{n+1}, \boldsymbol{\alpha}_{n+1})$ in Gleichung (2.25) jetzt zusätzlich von den erweiterten Freiheitsgraden abhängt, es ist also nichtlinear in \mathbf{d}_{n+1} und $\boldsymbol{\alpha}_{n+1}$. Die Linearisierung bezüglich beider Größen führt für ein ungedämpftes System ($\mathbf{C}^d = \mathbf{0}$) auf eine gekoppelte Iterationsvorschrift für $\Delta \mathbf{d}_{n+1}^{i+1}$ und $\Delta \boldsymbol{\alpha}_{n+1}^{i+1}$

$$\begin{bmatrix} \frac{1}{\beta \Delta t_n^2} \mathbf{M} + \mathbf{K}_{\text{EAS},n+1}^i & \mathbf{L}_{\text{EAS},n+1}^{\text{T},i} \\ \mathbf{L}_{\text{EAS},n+1}^i & \mathbf{D}_{\text{EAS},n+1}^i \end{bmatrix} \begin{pmatrix} \Delta \mathbf{d}_{n+1}^{i+1} \\ \Delta \boldsymbol{\alpha}_{n+1}^{i+1} \end{pmatrix} = -\mathcal{G}(\mathbf{d}_{n+1}^i, \boldsymbol{\alpha}_{n+1}^i). \quad (3.117)$$

Die eingeführten Matrizen sind

$$\mathbf{K}_{\text{EAS},n+1}^i = \left. \frac{\partial^2 \Pi_{n+1}^{\text{int}}}{\partial \mathbf{d} \partial \mathbf{d}} \right|_{\mathbf{d}_{n+1}^i}, \quad \mathbf{L}_{\text{EAS},n+1}^i = \left. \frac{\partial^2 \Pi_{n+1}^{\text{int}}}{\partial \boldsymbol{\alpha} \partial \mathbf{d}} \right|_{\mathbf{d}_{n+1}^i, \boldsymbol{\alpha}_{n+1}^i}, \quad \mathbf{D}_{\text{EAS},n+1}^i = \left. \frac{\partial^2 \Pi_{n+1}^{\text{int}}}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\alpha}} \right|_{\boldsymbol{\alpha}_{n+1}^i} \quad (3.118)$$

und die in $\mathcal{G}(\mathbf{d}_{n+1}^i, \boldsymbol{\alpha}_{n+1}^i)$ enthaltenen Kraftvektoren lauten

$$\mathbf{f}_{n+1}^{\text{int},i} = \left. \frac{\partial \Pi_{n+1}^{\text{int}}}{\partial \mathbf{d}} \right|_{\mathbf{d}_{n+1}^i}, \quad \mathbf{f}_{n+1}^{\text{ext},i} = \left. \frac{\partial \Pi_{n+1}^{\text{ext}}}{\partial \mathbf{d}} \right|_{\mathbf{d}_{n+1}^i}, \quad \mathbf{p}_{\text{EAS},n+1}^i = \left. \frac{\partial \Pi_{n+1}^{\text{int}}}{\partial \boldsymbol{\alpha}} \right|_{\boldsymbol{\alpha}_{n+1}^i}. \quad (3.119)$$

Die Dimensionen der oben eingeführten Matrizen und Vektoren richtet sich nach der Anzahl der Freiheitsgrade des Elementes n_{st} und der Anzahl der erweiterten Freiheitsgrade n_{as} , also der Zahl der eingeführten EAS-Erweiterungen. Da die EAS-Parameter auf Elementebene für jedes Element gesondert definiert sind, also keine Kompatibilität mit den Parametern der Nachbarlemente aufweisen, lässt sich die zweite Zeile von Gleichung (3.117) nach $\Delta \boldsymbol{\alpha}_{n+1}^{i+1}$ auflösen, wobei im Folgenden auf die Indizierung verzichtet wird:

$$\Delta \boldsymbol{\alpha} = -\mathbf{D}_{\text{EAS}}^{-1} \cdot (\mathbf{L}_{\text{EAS}} \cdot \Delta \mathbf{d} + \mathbf{p}_{\text{EAS}}). \quad (3.120)$$

Eingesetzt in die erste Zeile ergibt sich die tangentielle Elementsteifigkeitsmatrix

$$\mathbf{K} = \mathbf{K}_{\text{EAS}} - \mathbf{L}_{\text{EAS}}^{\text{T}} \cdot \mathbf{D}_{\text{EAS}}^{-1} \cdot \mathbf{L}_{\text{EAS}} \quad (3.121)$$

und der Elementresidualvektor

$$\mathbf{f} = \mathbf{f}^{\text{ext}} - (\mathbf{f}^{\text{int}} - \mathbf{L}_{\text{EAS}}^{\text{T}} \cdot \mathbf{D}_{\text{EAS}} \cdot \mathbf{p}_{\text{EAS}}). \quad (3.122)$$

Innerhalb des Zeitschritts im impliziten Zeitintegrationsverfahrens werden die aktuellen Verschiebungen \mathbf{d}_{n+1} iterativ bestimmt. Für jeden Iterationsschritt, in dem auf Systemebene das Inkrement $\Delta \mathbf{d}$ berechnet wird, kann mit den Gleichungen (3.117) – (3.120) auch das Inkrement der erweiterten Freiheitsgrade $\Delta \boldsymbol{\alpha}$ auf Elementebene ermittelt werden. Es müssen dann die Einträge der Matrizen \mathbf{L}_{EAS} und \mathbf{D}_{EAS} sowie des Vektors \mathbf{p}_{EAS} für jedes Element abgespeichert werden, da im aktuellen Iterationsschritt die Werte des vorigen Schrittes erforderlich sind. Bei bestimmten Methoden zur Invertierung der quadratischen Matrix \mathbf{D}_{EAS} sind eventuell noch weitere Größen als Geschichtsvariablen abzuspeichern. Erst beim Erreichen der Konvergenz sowohl für \mathbf{d} als auch für $\boldsymbol{\alpha}$ wird mit dem nächsten Zeitschritt fortgesetzt. Ausführliche Darstellungen zur Implementierung der EAS-Methode in einen nichtlinearen impliziten Lösungsalgorithmus finden sich u. a. in [45, 50].

EAS für explizite Zeitintegration

Für die vorliegende Arbeit wurden Elementformulierungen hauptsächlich für das in Abschnitt 2.2 beschriebene zentrale Differenzenverfahren umgesetzt. Ein wesentlicher Unterschied bei der Anwendung der EAS-Methode ergibt sich durch die explizite Formulierung des Zeitintegrationsverfahrens. Zur Bestimmung der aktuellen Verschiebungen \mathbf{d}_{n+1} sind also nur dem Zeitschritt n zugeordnete Größen erforderlich. Für das Funktional zur Bestimmung der Kraftvektoren bedeutet dies

$$\Pi_n^{\text{EAS}}(\mathbf{d}_n, \boldsymbol{\alpha}_n). \quad (3.123)$$

Da die Verschiebungen \mathbf{d}_n zu Beginn des Zeitschrittes bereits bekannt sind – vgl. hierzu Abbildung 2.2 – entfallen in der Variation (3.115) alle Terme mit $\delta \mathbf{d}$ und es bleibt die auf Elementebene zu erfüllende Nebenbedingung

$$\delta \Pi_n^{\text{EAS}} = \delta \boldsymbol{\alpha}^T \frac{\partial \Pi^{\text{int}}(\mathbf{d}_n, \boldsymbol{\alpha}_n)}{\partial \boldsymbol{\alpha}} = 0. \quad (3.124)$$

Für beliebige $\delta \boldsymbol{\alpha} \neq \mathbf{0}$ ergibt sich so ein in $\boldsymbol{\alpha}$ nichtlineares Gleichungssystem der Dimension n_{as} . Die Linearisierung führt auf

$$\mathbf{D}_{\text{EAS},n}^i \Delta \boldsymbol{\alpha}_n^{i+1} = -\mathbf{p}_{\text{EAS},n}^i \quad (3.125)$$

zur iterativen Bestimmung von $\boldsymbol{\alpha}_n$ auf Elementebene. Wegen der üblicherweise sehr kleinen Zeitschritte bei der expliziten Zeitintegration, gegeben durch das Kriterium (2.13), konnte bei den implementierten Elementformulierungen in vielen Anwendungen – insbesondere bei Problemen mit moderaten Geschwindigkeiten – bereits nach einem Iterationsschritt Konvergenz festgestellt werden. Es wurde also die explizite Gleichung

$$\boldsymbol{\alpha}_n = \boldsymbol{\alpha}_{n-1} - \mathbf{D}_{\text{EAS}}^{-1} \cdot \mathbf{p}_{\text{EAS}} \quad (3.126)$$

umgesetzt, die nur eine einzige Invertierung der Matrix \mathbf{D}_{EAS} je Element und Zeitschritt erfordert. Die Simulation hochdynamischer Prozesse, bei denen auch innerhalb eines sehr kurzen Zeitschrittes größere Änderungen der Verschiebungen auftreten, erfordert allerdings mehr als einen Iterationsschritt. Nähere Untersuchungen sind in Kapitel 5 an einem numerischen Beispiel angegeben. Als Geschichtsvariablen müssen lediglich die aktuellen Werte der erweiterten Freiheitsgrade $\boldsymbol{\alpha}$ abgespeichert werden, da diese für den nächsten Zeitschritt wieder erforderlich sind. Trotz des geringen Speicherbedarfs erfordert die Methode bei n_{as} EAS-Erweiterungen zumindest einmal die Invertierung einer Matrix der Dimension $n_{\text{as}} \times n_{\text{as}}$ für jedes Element in jedem Zeitschritt; hier wird in der Praxis ein Gleichungslösungsalgorithmus verwendet, verschiedene Möglichkeiten werden beispielsweise in [108] diskutiert. Bei einer entsprechenden Zahl zusätzlicher Freiheitsgrade kann dies die numerische Effizienz der Elementformulierung erheblich beeinflussen, sodass EAS für explizite Zeitintegration nur eingeschränkt sinnvoll ist. In der vorliegenden Arbeit wurden überwiegend gezielt bestimmte Versteifungseffekte mit EAS behandelt, für die sehr wenige Erweiterungen ausreichend sind. Jedoch findet man im Schrifttum für implizite Anwendungen durchaus Elementformulierungen etwa mit 30 Parametern (vgl. z. B. [8, 63]), welche sich für die Nutzung in expliziten Verfahren – insbesondere auf Grund der sehr kleinen Zeitschritte – weniger eignen. Entsprechende Untersuchungen zur Effizienz sind in Kapitel 5 angegeben.

Ein Konzept zur Reduktion des Rechenaufwandes auf Elementebene durch die erforderliche Matrixinvertierung ist die Diagonalisierung der Matrix \mathbf{D}_{EAS} aus Gleichung (3.118), da dann lediglich deren Diagonalelemente invertiert werden müssen, also nur wenige skalare Operationen erforderlich sind. Eine Möglichkeit der Diagonalisierung ist die Anwendung der zeilenweisen Summation, wie sie in Abschnitt 3.3.3 für die Massenmatrix beschrieben ist. Es ist dann

$$\left[\mathbf{D}_{\text{EAS}}^d \right]_{ij} = \begin{cases} \sum_{j=1}^{\text{nst}} [\mathbf{D}_{\text{EAS}}]_{ij}, & \text{für } i = j \\ 0, & \text{für } i \neq j \end{cases} . \quad (3.127)$$

Eine weniger aufwändige Vorgehensweise ergibt sich durch Vernachlässigung der Nebendiagonalterme, also

$$\left[\mathbf{D}_{\text{EAS}}^d \right]_{ij} = \begin{cases} [\mathbf{D}_{\text{EAS}}]_{ij}, & \text{für } i = j \\ 0, & \text{für } i \neq j \end{cases} . \quad (3.128)$$

Dies ist jedoch nur zulässig, wenn die Matrix \mathbf{D}_{EAS} dominant diagonal besetzt ist, was für einige untersuchte Beispiele bestätigt werden konnte. Die Diagonalisierung von \mathbf{D}_{EAS} führt zwar zu einer Reduktion der erforderlichen Rechenoperationen auf Elementebene, jedoch sind immer noch aufwändige Initialisierungsschritte zum lokalen Auskondensieren der zusätzlichen Freiheitsgrade α_i erforderlich. Wie am Beispiel in Abschnitt 5 gezeigt wird, sind die Einsparungen hinsichtlich der Gesamtrechnzeit im Vergleich zu den Standard EAS-Elementen vergleichsweise gering, weshalb die Methode nicht weiter verfolgt wurde.

Anwendung für Plastizität mit großen Dehnungen und großen Rotationen

Wie zuvor bei ANS ist die EAS-Methode zunächst für GREEN-LAGRANGE-Verzerrungen in natürlichen Koordinaten formuliert. Jedoch wird von SIMO & ARMERO in [113] eine Möglichkeit angegeben, die Methode direkt auf den Deformationsgradienten anzuwenden, der für die Plastizitätsformulierung mit endlichen Deformationen erforderlich ist (siehe auch [47, 114]). Die Vorgehensweise, die zusammengefasst u. a. auch der Arbeit von FREISCHLÄGER [45] entnommen werden kann, basiert auf der additiven Zusammensetzung des aus der Kinematik bestimmten Deformationsgradienten \mathbf{F}^u aus Gleichung (1.3) in einen konstanten $\mathbf{F}^{u,c}$ und einen veränderlichen Anteil $\mathbf{F}^{u,v}$. Dieser wird durch Addition eines Tensors $\tilde{\mathbf{F}}$, der die Verzerrungsansätze enthält und somit ebenfalls veränderlich ist, erweitert. Es ergibt sich

$$\mathbf{F} = \mathbf{F}^u + \tilde{\mathbf{F}} = \underbrace{\mathbf{F}^{u,c}}_{\text{konstant}} + \underbrace{\mathbf{F}^{u,v} + \tilde{\mathbf{F}}}_{\text{veränderlich}} . \quad (3.129)$$

Zur Bestimmung des erweiterten Deformationsgradienten werden zunächst die Erweiterungen, die den Einträgen aus $\mathbf{N}_{\text{EAS}} \cdot \boldsymbol{\alpha}$ entsprechen, in einer Matrix

$$\mathbf{F}_{\text{EAS},0} = \sum_{i=1}^{\text{nas}} \mathbf{N}_{\text{EAS},i} \alpha_i \quad (3.130)$$

angeordnet. Es ist hierbei $\mathbf{N}_{\text{EAS},i}$ eine 3×3 Matrix, welche die Ansatzfunktion für den i -ten erweiterten Freiheitsgrad α_i enthält. $\mathbf{F}_{\text{EAS},0}$ bezieht sich auf die Elementmitte und muss

nun noch in das konvektive Koordinatensystem transformiert in dem auch die kompatiblen Verzerrungen definiert sind

$$\mathbf{F}_{\text{EAS}} = \frac{\det \mathbf{J}_0}{\det \mathbf{J}} \mathbf{J}_0^{-1} \cdot \mathbf{F}_{\text{EAS},0} \cdot \mathbf{J}_0^{-T}. \quad (3.131)$$

Es folgt die Multiplikation mit dem konstanten Anteil des Deformationsgradienten

$$\tilde{\mathbf{F}} = \mathbf{F}^{u,c} \cdot \mathbf{F}_{\text{EAS}} = \frac{\det \mathbf{J}_0}{\det \mathbf{J}} \mathbf{F}^{u,c} \cdot \mathbf{J}_0^{-1} \cdot \mathbf{F}_{\text{EAS},0} \cdot \mathbf{J}_0^{-T}, \quad (3.132)$$

wodurch die Unabhängigkeit der Erweiterungen von Starrkörperrotationen gewährleistet wird. Nun liegt mit Gleichung (3.129) der erweiterte Deformationsgradient vor, der dann der in Abbildung 1.3 skizzierten Plastizitätsroutine übergeben werden kann.

3.4.5. Die Methode der inkompatiblen Moden

Allgemeines

Ein wesentlicher Nachteil der EAS-Methode stellt wie bereits im vorigen Abschnitt erörtert die erforderliche Invertierung der Matrix \mathbf{D}_{EAS} dar. Selbst unter der Annahme aus Gleichung (3.126), nach der bei kleinen Zeitschritten ein Iterationsschritt zur Lösung ausreichend ist, muss die Invertierung von \mathbf{D}_{EAS} zu jedem Zeitschritt für jedes Element durchgeführt werden. Dies führt zu einem signifikanten Anstieg der Operationen auf Elementebene je nach Anzahl der EAS-Erweiterungen, die sich selbst bei einer Diagonalisierung von \mathbf{D}_{EAS} noch deutlich bemerkbar macht. Im Kontext eines expliziten Zeitintegrationsverfahren, bei dem üblicherweise sehr kleine Zeitschritte auftreten und der numerische Aufwand durch die Operationen auf Elementebene dominiert ist, machen sich diese zusätzlichen Rechenschritte insbesondere bemerkbar.

Eine Verallgemeinerung der *Methode der inkompatiblen Moden*, die wie eingangs in Abschnitt 3.4.4 erwähnt von WILSON ET. AL. [126] und TAYLOR ET. AL. [123] die Grundlage der EAS-Methode darstellt, wird von BISCHOFF & ROMERO in [24] angegeben. Sie erlaubt eine analoge Formulierung beliebiger EAS-Elemente durch die Einführung inkompatibler Verschiebungsfreiheitsgrade, welche auf Systemebene bei entsprechender Anpassung des Zeitintegrationsalgorithmus wie die kompatiblen Freiheitsgrade behandelt werden können. Eine lokale Kondensation, die bei der EAS-Methode die gleiche Anzahl unbekannter Größen auf Systemebene wie für ein Verschiebungselement gewährleistet, kann, muss aber hier nicht durchgeführt werden. Die daraus resultierende größere Anzahl globaler Gleichungen ist für den expliziten Zeitintegrationsalgorithmus verglichen mit der lokalen Gleichungslösung, die für die EAS-Parameter anfällt, jedoch deutlich günstiger. Dies wird in Abschnitt 5 an entsprechenden Prinzipbeispielen verdeutlicht. Im Folgenden wird die in [24] für Scheibenelemente mit linearen Ansätzen hergeleitete Verallgemeinerung auf Solid-Shell-Elemente mit 8 Knoten erweitert.

Inkompatible Freiheitsgrade

Das Ziel der Verallgemeinerung bei BISCHOFF & ROMERO [24] ist die Einführung und Interpolation der inkompatiblen Verschiebungsfreiheitsgrade $\tilde{\mathbf{d}}$, die gerade auf die erweiterten

Verzerrungen $\tilde{\mathbf{E}}(\tilde{\mathbf{d}}) = \tilde{\mathbf{E}}(\boldsymbol{\alpha})$ aus Gleichung (3.104) führen sollen. Hierzu werden LEGENDRE-Polynome vorgeschlagen, die bereits in Abschnitt 3.4.3 in Gleichung (3.92) eingeführt wurden. Die Ansatzfunktionen für den i -ten inkompatiblen Freiheitsgrad lauten allgemein

$$\mathbf{N}_{\text{IM},i} = \begin{pmatrix} N_{\text{IM},i}^{\xi} \\ N_{\text{IM},i}^{\eta} \\ N_{\text{IM},i}^{\zeta} \end{pmatrix} \quad (3.133)$$

mit

$$\begin{aligned} N_{\text{IM},i}^{\xi} = & \frac{1}{3} a_{100} P_2(\xi) + a_{010} P_1(\xi) P_1(\eta) + a_{001} P_1(\xi) P_1(\zeta) + \frac{1}{3} a_{110} P_2(\xi) P_1(\eta) + \\ & \frac{1}{3} a_{101} P_2(\xi) P_1(\zeta) + a_{011} P_1(\xi) P_1(\eta) P_1(\zeta) + \frac{1}{3} a_{111} P_2(\xi) P_1(\eta) P_1(\zeta) + \\ & \frac{1}{3} (c_{010} - b_{100}) P_2(\eta) + \frac{1}{3} (f_{001} - d_{100}) P_2(\zeta) + \frac{1}{2} (c_{001} + f_{010} - e_{100}) P_1(\eta) P_1(\zeta) + \\ & \frac{1}{6} c_{110} P_1(\xi) P_2(\eta) + \frac{1}{6} f_{101} P_1(\xi) P_2(\zeta) + \frac{1}{3} (c_{011} - b_{101}) P_2(\eta) P_1(\zeta) + \\ & \frac{1}{3} (f_{011} - d_{110}) P_1(\eta) P_2(\zeta) + \frac{1}{6} c_{111} P_1(\xi) P_2(\eta) P_1(\zeta) + \frac{1}{6} f_{111} P_1(\xi) P_1(\eta) P_2(\zeta), \end{aligned} \quad (3.134)$$

$$\begin{aligned} N_{\text{IM},i}^{\eta} = & b_{100} P_1(\xi) P_1(\eta) + \frac{1}{3} b_{010} P_2(\eta) + b_{001} P_1(\eta) P_1(\zeta) + \frac{1}{3} b_{110} P_1(\xi) P_2(\eta) + \\ & b_{101} P_1(\xi) P_1(\eta) P_1(\zeta) + \frac{1}{3} b_{011} P_2(\eta) P_1(\zeta) + \frac{1}{3} b_{111} P_1(\xi) P_2(\eta) P_1(\zeta) + \\ & \frac{1}{3} (c_{100} - a_{010}) P_2(\xi) + \frac{1}{3} (e_{001} - d_{010}) P_2(\zeta) + \frac{1}{2} (c_{001} + e_{100} - f_{010}) P_1(\xi) P_1(\zeta) + \\ & \frac{1}{6} c_{110} P_2(\xi) P_1(\eta) + \frac{1}{6} e_{011} P_1(\eta) P_2(\zeta) + \frac{1}{3} (c_{101} - a_{011}) P_2(\xi) P_1(\zeta) + \\ & \frac{1}{3} (e_{101} - d_{110}) P_1(\xi) P_2(\zeta) + \frac{1}{6} c_{111} P_2(\xi) P_1(\eta) P_1(\zeta) + \frac{1}{6} e_{111} P_1(\xi) P_1(\eta) P_2(\zeta) \end{aligned} \quad (3.135)$$

und

$$\begin{aligned} N_{\text{IM},i}^{\zeta} = & d_{100} P_1(\xi) P_1(\zeta) + d_{010} P_1(\eta) P_1(\zeta) + \frac{1}{3} d_{001} P_2(\zeta) + d_{110} P_1(\xi) P_1(\eta) P_1(\zeta) + \\ & \frac{1}{3} d_{101} P_1(\xi) P_2(\zeta) + \frac{1}{3} d_{011} P_1(\eta) P_2(\zeta) + \frac{1}{3} d_{111} P_1(\xi) P_1(\eta) P_2(\zeta) + \\ & \frac{1}{3} (e_{010} - b_{001}) P_2(\eta) + \frac{1}{3} (f_{100} - a_{001}) P_2(\xi) + \frac{1}{2} (e_{100} + f_{010} - c_{001}) P_1(\xi) P_1(\eta) + \\ & \frac{1}{6} e_{011} P_2(\eta) P_1(\zeta) + \frac{1}{6} f_{101} P_2(\xi) P_1(\zeta) + \frac{1}{3} (e_{110} - b_{101}) P_1(\xi) P_2(\eta) + \\ & \frac{1}{3} (f_{110} - a_{011}) P_2(\xi) P_1(\eta) + \frac{1}{6} e_{111} P_1(\xi) P_2(\eta) P_1(\zeta) + \frac{1}{6} f_{111} P_2(\xi) P_1(\eta) P_1(\zeta). \end{aligned} \quad (3.136)$$

Wie zuvor in Gleichung (3.112), wird hier aus Gründen der Übersichtlichkeit auf eine Indizierung der Konstanten $a_{k\ell m}$ bis $f_{k\ell m}$, mit $k, \ell, m = \{0, 1\}$, die für jede Spalte der Matrix \mathbf{N}_{EAS} unterschiedlich sind, verzichtet. Die erweiterten Verschiebungen, welche zu den kompatiblen aus Gleichung (3.34) addiert werden, ergeben sich schließlich durch Interpolation der inkompatiblen Freiheitsgrade zu

$$\mathbf{u}_{\text{IM}} = \mathbf{N}_{\text{IM}} \cdot \tilde{\mathbf{d}} \quad \Rightarrow \quad \mathbf{u}_{\text{enh}} = \mathbf{u} + \mathbf{u}_{\text{IM}}. \quad (3.137)$$

Es liegt nun ein erweitertes Verschiebungsfeld \mathbf{u}_{enh} vor, welches bereits die EAS-Erweiterungen enthält, also automatisch die erweiterten Verzerrungen aus Gleichung (3.104) liefert. Die Verwendung der LEGENDRE-Polynome aus Gleichung (3.92) ist hierbei erforderlich, da sich durch deren Eigenschaft aus Gleichung (3.100) Terme 2. Ordnung $P_2(x)$ in den Verzerrungen bei einer $2 \times 2 \times 2$ -Punkt GAUSS-Integration nicht auf die innere Energie auswirken. Die Ansatzfunktionen \mathbf{N}_{IM} liefern also nicht die korrekten erweiterten Verzerrungsverläufe der EAS-Methode, führen aber durch die verwendeten LEGENDRE-Polynome in Kombination mit einer entsprechenden Integrationsvorschrift auf die gewünschte Verzerrungsenergie und somit auf die gleichen Verschiebungen wie bei EAS.

Zur Veranschaulichung werden nun die hergeleiteten Beziehungen exemplarisch für eine aus dem Schriftum bekannte Formulierung ausgewertet. Das auf dem inkompatiblen Verschiebungselement von TAYLOR ET. AL. [123] basierende Volumenelement mit 9 EAS-Erweiterungen, das beispielsweise bei ANDELFINGER & RAMM [8] vorgestellt und in weiteren Veröffentlichungen mit dem Namen H1E9 bzw. Q1E9 bezeichnet wird, besitzt die \mathbf{N}_{EAS} -Matrix

$$\mathbf{N}_{\text{EAS}} = \begin{pmatrix} \xi & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \eta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \xi & \eta & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \zeta & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \eta & \zeta & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \xi & \zeta \end{pmatrix}. \quad (3.138)$$

Hier ist beispielsweise für die dritte Spalte der Faktor $c_{100} = 1$, alle anderen Größen $a_{k\ell m}$ bis $f_{k\ell m}$ ergeben sich zu Null. Die Auswertung der Gleichungen (3.134), (3.135) und (3.136) für alle Spalten liefert schließlich die Ansatzfunktionen der 9 inkompatiblen Freiheitsgrade

$$\mathbf{N}_{\text{IM}} = \begin{pmatrix} \frac{1}{3} P_2(\xi) & 0 & 0 & \frac{1}{3} P_2(\eta) & 0 & 0 & 0 & 0 & \frac{1}{3} P_2(\zeta) \\ 0 & \frac{1}{3} P_2(\eta) & \frac{1}{3} P_2(\xi) & 0 & 0 & 0 & \frac{1}{3} P_2(\zeta) & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{3} P_2(\zeta) & \frac{1}{3} P_2(\eta) & 0 & \frac{1}{3} P_2(\xi) & 0 \end{pmatrix}, \quad (3.139)$$

mit denen dann die Verschiebungsinterpolation nach Gleichung (3.137) durchgeführt werden kann. Für dieses Beispiel lassen sich die erweiterten Verzerrungen sogar exakt reproduzieren. Es ist also

$$\tilde{\mathbf{E}}_{\text{EAS}}^0 = \tilde{\mathbf{E}}_{\text{IM}}^0 \quad (3.140)$$

mit den Komponenten von $\tilde{\mathbf{E}}_{\text{EAS}}^0$ gemäß Gleichung (3.111) aus

$$\boldsymbol{\epsilon}_{\text{EAS}} = \mathbf{N}_{\text{EAS}} \cdot \boldsymbol{\alpha} \quad (3.141)$$

und

$$[\tilde{\mathbf{E}}_{\text{IM}}^0] = \frac{1}{2} (\Delta \mathbf{u}_{\text{IM}} + \Delta \mathbf{u}_{\text{IM}}^T), \quad \text{mit} \quad \Delta \mathbf{u}_{\text{IM}} = \left(\frac{\partial \mathbf{u}_{\text{IM}}}{\partial \boldsymbol{\xi}} \right)^T \quad (3.142)$$

und \mathbf{u}_{IM} aus Gleichung (3.137). Durch die Eigenschaft der verwendeten LEGENDRE-Polynome aus Gleichung (3.100) gilt für beliebige Erweiterungen \mathbf{N}_{IM} bei einer numerischen Integration mit $2 \times 2 \times 2$ GAUSS-Punkten immer zumindest

$$\tilde{\mathbf{E}}_{\text{EAS}}^0 \Big|_{\xi=\xi_{\text{GP}}, \eta=\eta_{\text{GP}}, \zeta=\zeta_{\text{GP}}} = \tilde{\mathbf{E}}_{\text{IM}}^0 \Big|_{\xi=\xi_{\text{GP}}, \eta=\eta_{\text{GP}}, \zeta=\zeta_{\text{GP}}}. \quad (3.143)$$

Die so berechneten erweiterten Verzerrungen können also genau wie bei den EAS-Formulierungen zur Auswertung der Materialformulierung verwendet werden und führen zu den gleichen inneren Knotenkräfte und somit auch zur entsprechenden Reduktion von Versteifungseffekten.

Die beschriebene Vorgehensweise basiert auf der Reproduktion der erweiterten Verzerrungen durch entsprechende Verschiebungsansätze. Inwiefern eine Anwendung für Plastizität mit großen Deformationen – also eine direkte Bestimmung des in Abschnitt 3.4.4 verwendeten erweiterten Deformationsgradienten aus den Verschiebungen \mathbf{u}_{enh} möglich ist, muss noch untersucht werden.

Beschreibung der Masse der inkompatiblen Freiheitsgrade

Die Behandlung der inkompatiblen Freiheitsgrade auf Systemebene, insbesondere innerhalb des zentralen Differenzenverfahrens, erfordert eine Erweiterung der diagonalisierten Massenmatrix aus Abschnitt 3.3. Jedem zusätzlichen Freiheitsgrad muss also eine Masse zugeordnet sein, da Gleichung (2.5) für alle Freiheitsgrade – kompatible und inkompatible – ausgewertet wird. Analog zu den Ausführungen in Abschnitt 3.3 bietet sich hier die Erweiterung der Matrix der Ansatzfunktionen \mathbf{N} aus Gleichung (3.38) an, die zusätzlichen Ansätze \mathbf{N}_{IM} werden also an die kompatiblen Ansätze angehängt. Analog zu Gleichung (3.38) ergibt sich so für die konsistente Elementmassenmatrix

$$\mathbf{M}_c^e = \rho \int_V \begin{bmatrix} \mathbf{N}_u & \mathbf{N}_{\text{IM}} \end{bmatrix}^T \cdot \begin{bmatrix} \mathbf{N}_u & \mathbf{N}_{\text{IM}} \end{bmatrix} dV = \rho \int_V \begin{bmatrix} \mathbf{N}_u^T \cdot \mathbf{N}_u & \mathbf{N}_u^T \cdot \mathbf{N}_{\text{IM}} \\ \mathbf{N}_{\text{IM}}^T \cdot \mathbf{N}_u & \mathbf{N}_{\text{IM}}^T \cdot \mathbf{N}_{\text{IM}} \end{bmatrix} dV, \quad (3.144)$$

wobei durch \mathbf{N}_u hier die Standard-Verschiebungsansätze repräsentiert werden. Die Diagonalisierung erfolgt für die kompatiblen Freiheitsgrade wie in Abschnitt 3.3.3 beschrieben beispielsweise durch zeilenweise Summation der über $\mathbf{N}_u^T \cdot \mathbf{N}_u$ bestimmten Untermatrix. Bei allen in der vorliegenden Arbeit untersuchten IM-Formulierungen, wie beispielsweise der durch Gleichung (3.139) beschriebenen, beinhaltet \mathbf{N}_{IM} ohnehin Polynome quadratischer Ordnung, bei Multiplikation mit \mathbf{N}_u ergeben sich also kubische Terme im Integranden. Bei rechtwinkligen Elementgeometrien, also wenn $\det \mathbf{J} = \text{const.}$ ist, ergibt die Integration über das Elementvolumen den Wert Null. Bei der durch $\mathbf{N}_{\text{IM}}^T \cdot \mathbf{N}_{\text{IM}}$ berechneten Untermatrix, führt die Integration über das Elementvolumen bei rechtwinkliger Elementgeometrie direkt auf eine Diagonalmatrix. Von der rechtwinkligen Form abweichende Elemente können auf der Nebendiagonalen auch von Null verschiedene Werte aufweisen. Für die im folgenden Abschnitt noch genauer beschriebenen implementierten Elementformulierungen, wird die Untermatrix $\mathbf{N}_u^T \cdot \mathbf{N}_{\text{IM}}$ bzw. deren Transponierte unabhängig von der Elementgeometrie nicht beachtet. Ebenso werden für die Massenrepräsentation der zusätzlichen Freiheitsgrade ausschließlich die Diagonalterme von $\mathbf{N}_{\text{IM}}^T \cdot \mathbf{N}_{\text{IM}}$ verwendet und mit einem Skalierungsfaktor α_m multipliziert. Das genaue Vorgehen sowie der Einfluss dieses Parameters α_m wird noch anhand eines Beispiels diskutiert.

Es ist darauf zu achten, dass einerseits eine $2 \times 2 \times 2$ GAUSS-Integration auf Grund der höheren Polynomordnung hier nicht ausreichend ist. Zudem würde sich wegen der Eigenschaft der LEGENDRE-Polynome aus Gleichung (3.100) für die Masse aller inkompatiblen Freiheitsgrade Null ergeben, die diagonalisierte Massenmatrix wäre dann nicht mehr invertierbar. Die Wahl einer höheren Integrationsordnung mit $3 \times 3 \times 3$ oder mehr GAUSS-Punkten führt

auf von Null verschiedene Massen, inwiefern die daraus resultierenden diagonalen bzw. diagonalisierten Massenmatrizen zur dynamischen Beschreibung von Strukturproblemen mit finiten Elementen geeignet sind, soll im Folgenden anhand einer Eigenwertuntersuchung exemplarisch diskutiert werden.

Verglichen werden die Eigenwerte, bzw. die Eigenfrequenzen eines quaderförmigen Einzelelementes ohne Lagerung, mit den in Abbildung 3.9 angegebenen Geometrie- und Materialeigenschaften. Untersucht werden fünf verschiedene Elementformulierungen:

- DISP – reine Verschiebungsformulierung ohne Modifikationen
- EAS9 – 9 EAS-Parameter, \mathbf{N}_{EAS} aus Gleichung (3.138)
- IM9 – 9 inkompatible Freiheitsgrade, \mathbf{N}_{IM} aus Gleichung (3.139)
 - . . . -01 – Masse der inkompatiblen Freiheitsgrade unskaliert
 - . . . -08 – Masse der inkompatiblen Freiheitsgrade mit $\alpha_m = 8.00$ skaliert
 - . . . -32 – Masse der inkompatiblen Freiheitsgrade mit $\alpha_m = 32.00$ skaliert

Die Lösung des speziellen Eigenwertproblems

$$(\mathbf{K}_0 - \lambda \mathbf{M}_d) \cdot \boldsymbol{\phi} = 0, \quad (3.145)$$

mit der jeweiligen linearen Steifigkeitsmatrix \mathbf{K}_0 aus Gleichung (2.14) und der zugehörigen diagonalisierten Massenmatrix \mathbf{M}_d erfolgte mit MATHEMATICA. Die Eigenwerte λ_i , also die Quadrate der Eigenkreisfrequenzen ω_i , sind in Tabelle 3.1 angegeben.

Beim Vergleich der reinen Verschiebungsformulierung mit dem EAS9-Element wird deutlich, dass die Eigenfrequenzen ganz bestimmter Eigenformen durch die Erweiterung der Verzerrungen verändert werden. Im vorliegenden Fall handelt sich hierbei um Biegeeigenformen, da die EAS-Formulierung insbesondere den Effekt der Biegeversteifung behandelt [45]. Die Reduktion der Biegeeigenfrequenzen zeigt also die Verminderung der Biegesteifigkeit und ist somit erwünscht. Die Elementformulierung IM0-01 mit 9 inkompatiblen Freiheitsgraden und sehr kleiner zugeordneter Masse zeigt numerisch fast identische Eigenwerte, wie das EAS9-Element. Die gewählte Berechnung mit der diagonalen Massenmatrix ist also zulässig, die neun zusätzlichen Eigenfrequenzen sind deutlich höher als die der kompatiblen Freiheitsgrade.

Problematisch im Zusammenhang mit expliziter Zeitintegration ist der Einfluss dieser zusätzlichen Eigenfrequenzen auf den kritischen Zeitschritt, in dessen Berechnung über

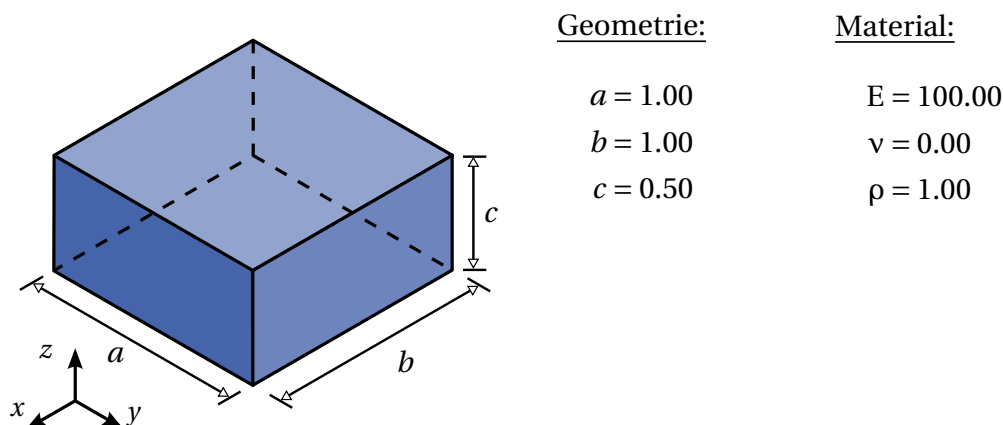


Abbildung 3.9.: Quaderförmiges Einzelelement zum Vergleich der Eigenfrequenzen

Eigenwert	DISP	EAS9	IM9-01	IM9-08	IM9-32
$\lambda_1 - \lambda_6$	0.00	0.00	0.00	0.00	0.00
λ_7	35.74	35.73	35.74	35.74	35.74
λ_8	155.55	133.33	132.58	127.15	106.78
λ_9	155.55	133.33	132.58	127.15	106.78
λ_{10}	200.00	133.33	132.59	127.54	111.85
λ_{11}	200.00	133.33	132.59	127.54	111.85
λ_{12}	222.21	155.55	155.55	155.55	155.55
λ_{13}	266.66	155.55	155.55	155.55	155.55
λ_{14}	400.00	222.21	222.21	222.21	222.21
λ_{15}	400.00	266.66	266.66	266.66	266.66
λ_{16}	400.00	400.00	400.00	400.00	400.00
λ_{17}	400.00	400.00	400.00	400.00	400.00
λ_{18}	400.00	400.00	400.00	400.00	400.00
λ_{19}	497.61	497.61	497.61	497.61	468.20
λ_{20}	600.00	533.34	532.58	526.91	468.20
λ_{21}	600.00	533.34	532.58	526.91	497.61
λ_{22}	1 000.00	1 000.00	1 000.00	1 000.00	499.99
λ_{23}	1 000.00	1 000.00	1 000.00	1 000.00	499.99
λ_{24}	1 600.01	1 600.01	1 600.01	1 572.80	750.00
λ_{25}	–	–	12 067.20	1 572.80	750.00
λ_{26}	–	–	12 067.20	1 600.01	1 000.00
λ_{27}	–	–	24 000.50	3 000.03	1 000.00
λ_{28}	–	–	24 000.50	3 000.03	1 599.98
λ_{29}	–	–	48 067.60	6 073.04	1 599.98
λ_{30}	–	–	48 067.60	6 073.04	1 600.00
λ_{31}	–	–	48 267.60	6 272.41	1 788.12
λ_{32}	–	–	48 267.60	6 272.41	1 788.12
λ_{33}	–	–	384 007.00	48 000.30	11 999.90

	⇒ unveränderter Eigenwert
	⇒ veränderter Eigenwert
	⇒ zusätzlicher Eigenwert aus inkompatiblen Moden

Tabelle 3.1.: Vergleich der Eigenwerte (Quadrate der Eigenfrequenzen) eines quaderförmigen Elements – reine Verschiebungsformulierung (DISP), 9 EAS-Parameter (EAS9) und 9 inkompatible Freiheitsgrade mit $\alpha_m = 1.00$ (IM9-01), $\alpha_m = 8.00$ (IM9-08), bzw. $\alpha_m = 32.00$ (IM9-32)

Gleichung (2.13) der Wert ω_{\max} eingeht. Im konkreten Fall mit den Werten aus Tabelle 3.1 ergibt sich

$$\Delta t_{\text{krit}}^{\text{EAS9}} = \frac{2}{\max \omega} = \frac{2}{\sqrt{1600}} = 5.00 \cdot 10^{-2} \quad \Leftrightarrow \quad \Delta t_{\text{krit}}^{\text{IM9-01}} = \frac{2}{\max \omega} = \frac{2}{\sqrt{384007}} = 3.23 \cdot 10^{-3}, \quad (3.146)$$

also ein etwa um den Faktor 15 kleiner Zeitschritt. Zur Simulation eines Strukturproblems mit IM9-01-Elementen sind also erheblich mehr Zeitschritte erforderlich, als bei der Verwendung von EAS9-Elementen. Um mit größeren Zeitschritten rechnen zu können, bietet sich – wie bereits in Abschnitt 2.2.6 für herkömmliche Elemente angesprochen – eine Skalierung speziell der den inkompatiblen Freiheitsgraden zugeordneten Masse an. Die diagonalisierte Massenmatrix einer Elementformulierung mit n_{st0} kompatiblen und n_{ad} inkompatiblen Freiheitsgraden ergibt sich dann zu

$$[\mathbf{M}_d^e]_{ii} = \begin{cases} [\mathbf{M}_d^e]_{ii}, & \text{für } i = 0 \dots n_{st0} \\ \alpha_m [\mathbf{M}_c^e]_{ii}, & \text{für } i = n_{st0} + 1 \dots n_{st0} + n_{ad} \end{cases}. \quad (3.147)$$

Um den Einfluss des Skalierungsfaktors α_m auf die dynamischen Eigenschaften zu bewerten, sind in Tabelle 3.1 für die IM9-Formulierung die Eigenwerte für zwei verschiedene Skalierungen angegeben. Es zeigt sich, dass durch die Erhöhung der Masse die den inkompatiblen Freiheitsgraden zugeordneten Eigenfrequenzen reduziert werden können. Als Folge ergibt sich die gewünschte Erhöhung des kritischen Zeitschrittes. Von der Massenskalierung ebenfalls betroffen sind jedoch die Biegeeigenfrequenzen, die im Vergleich zur DISP-Formulierung bereits im unskalierten Fall verkleinert wurden; die übrigen Eigenfrequenzen bleiben von der Erhöhung der Masse unbeeinflusst. Inwiefern sich diese Änderung der Eigenfrequenzen im dynamischen Abbildungsverhalten der Elementformulierung bemerkbar machen, wird in Kapitel 5 an Prinzipbeispielen diskutiert.

Eine Veranschaulichung des Einflusses der inkompatiblen Moden auf die verschiedenen Verzerrungskomponenten erreicht man durch gezieltes Aufbringen von Biegeformen, wobei alle 24 kompatiblen Freiheitsgrade des in Abbildung 3.9 skizzierten Elementes vorgegeben werden. Die Antwort der inkompatiblen Freiheitsgrade wird dann für jeden Deformationszustand ausgewertet und ist in Tabelle 3.2 dargestellt. Die skizzierten Formen wurden dabei durch einen linearen Anstieg der Verschiebungen über 10s aufgebracht, die anschließend für weitere 10s konstant gehalten wurden. Zu jedem Deformationszustand sind jeweils nur die beteiligten inkompatiblen Verschiebungen angegeben, alle anderen ergeben sich zu Null. Betrachtet man beispielsweise die erste Zeile der Tabelle, so erwartet man bei der Elementdeformation, die sich bei Biegung um die y -, bzw. η -Achse einstellt, eine durch die Querschubverzerrung $E_{\xi\xi}$ hervorgerufene Versteifung. Folglich wird der achte inkompatible Freiheitsgrad angeregt, der gemäß Gleichung (3.138) einer Erweiterung dieser Verzerrungskomponente mit $\alpha_8 \xi$ entspricht. Wird die Querkontraktionszahl auf einen von Null verschiedenen Wert gesetzt, für den dann auch volumetrische Versteifung zu erwarten ist, so kann man zusätzlich eine Beteiligung des fünften inkompatiblen Freiheitsgrades beobachten. Dieser ist der Erweiterung der Normalverzerrung $E_{\zeta\zeta}$ mit $\alpha_5 \zeta$ zuzuordnen. Für alle anderen Deformationsformen, lassen sich entsprechende Beteiligungen der inkompatiblen Moden beobachten.

Eine nähere Untersuchung der Eigenformen verdeutlicht, dass es sich bei den inkompatiblen Moden um Relativmoden bezüglich der herkömmlichen Deformationsformen handelt.

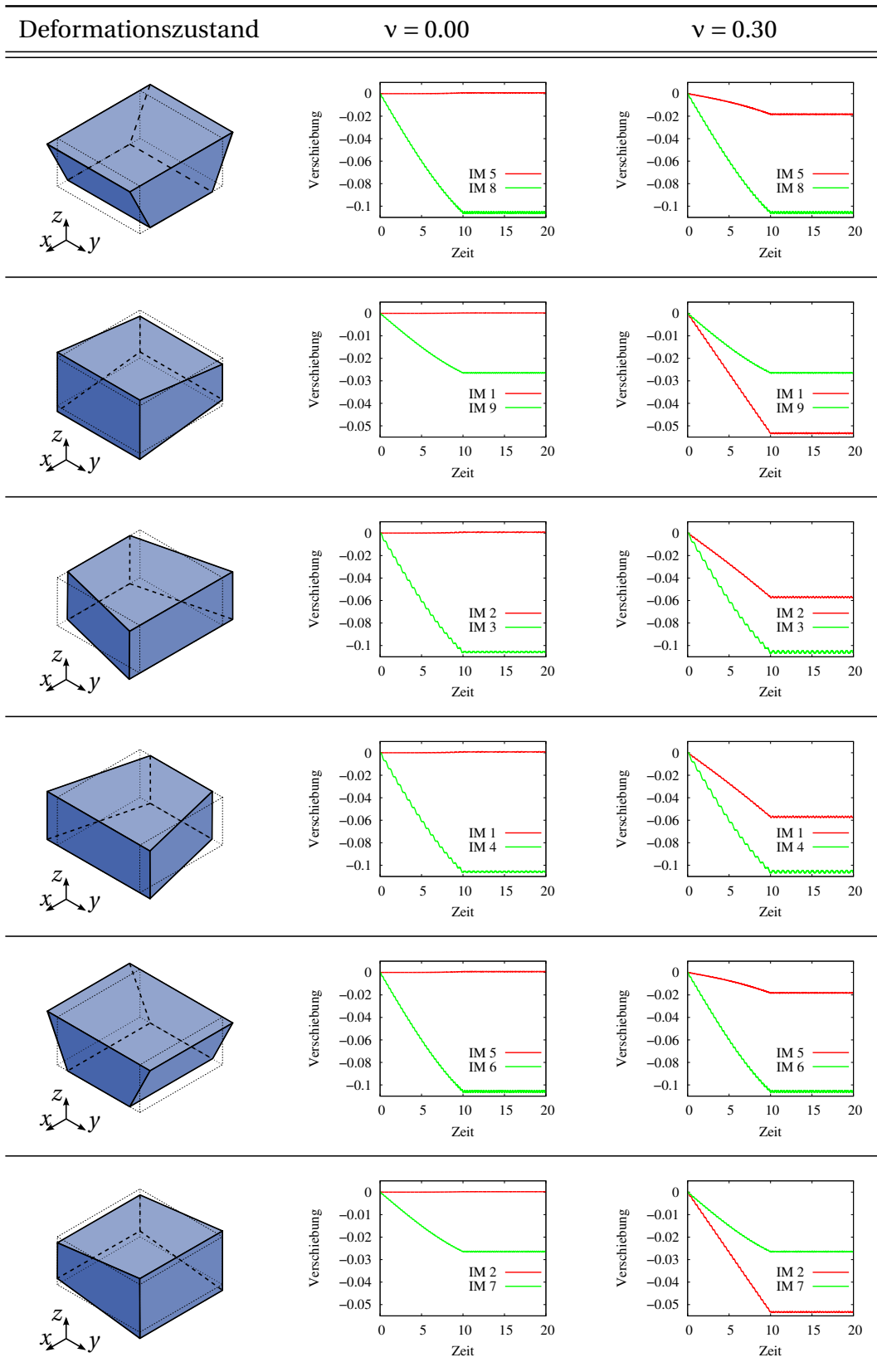


Tabelle 3.2.: Betrachtung der jeweiligen beteiligten inkompatiblen Moden für verschiedene Deformationszustände und unterschiedliche Querdehnzahlen

Damit wird die Analogie zu den erweiterten Verzerrungen nochmals deutlicher. Im Rahmen der vorliegenden Arbeit wurden bereits erste Untersuchungen durchgeführt, die in einem weiterführenden Forschungsprojekt vertieft werden (siehe z. B. [109]).

Bewertung der Methode hinsichtlich Effizienz

Die geringere Anzahl an Rechenoperationen bei Elementen mit inkompatiblen Freiheitsgraden im Vergleich zu äquivalenten EAS-Formulierungen führt auf eine erhebliche Reduktion des Rechenaufwands bei vergleichbarer Zeitschrittweite. Dies wird ebenfalls an Beispielen in Kapitel 5 belegt. Es muss lediglich der innere Knotenkraftvektor gemäß Gleichung (1.59)

$$\mathbf{f}^{\text{int}} = \frac{\partial \Pi_{\text{int},h}}{\partial \mathbf{d}_{\text{IM}}} \quad (3.148)$$

bestimmt werden, wobei in \mathbf{d}_{IM} die kompatiblen und die inkompatiblen Freiheitsgrade zusammengefasst sind. Auf Systemebene werden die kompatiblen Freiheitsgrade wie bisher assembliert, der gesamte Knotenkraftvektor wird dann durch die inkompatiblen Einträge erweitert. Die Programmschleifen zur Berechnung der globalen kinematischen Größen $\ddot{\mathbf{d}}$, $\dot{\mathbf{d}}$ und \mathbf{d} aus Abschnitt 2.2, müssen jetzt über alle Freiheitsgrade laufen, was jedoch dem wesentlich größeren Aufwand der lokalen Matrixinvertierung bei den EAS-Elementen gegenübersteht.

3.5. Kontakt zwischen verformbaren und starren Strukturen (STAS)

Eine wesentliche Schwierigkeit bei Kontaktfragestellungen ist die korrekte Beschreibung der Geometrie der in Kontakt tretenden Strukturteile. Hierzu existieren aus den letzten Jahrzehnten unzählige Publikationen, viele Algorithmen sind auch in den kommerziellen Programmen verfügbar. Für die vorliegende Arbeit wurden exemplarisch Elementformulierungen implementiert, die den Kontakt zwischen deformierbaren und starren Bauteilen beschreiben, da auch auf dem Gebiet der Kontaktmechanik wesentliche Vorteile durch die Nutzung symbolischer Programmierung zu erwarten sind. Die starren Strukturen wurden hierbei durch verschiedene, analytisch beschreibbare Formen repräsentiert (*'Surface-To-Analytical-Surface'* – STAS). Die einzelnen Elementformulierungen sowie Details zu deren Implementierung werden in Abschnitt 4.4.4 vorgestellt, die theoretischen Grundlagen stammen aus einer Forschungsgruppe am Institut für Mechanik und sind im wesentlichen in der Habilitationsschrift von KONYUKHOV [64] zusammengestellt. Weitere Referenzen zu den einzelnen Veröffentlichungen können dem genannten Buch entnommen werden.

Ein mögliches Kontaktszenario ist in Abbildung 3.10 skizziert. Die mit *'slave'* bezeichnete Fläche ist ein Teil der mit finiten Elementen diskretisierten, deformierbaren Struktur, die *'master'*-Fläche kann beispielsweise eine starre Ebene, ein Zylinder oder eine Kugel sein. Bei analytischer Beschreibung des geometrischen Körpers lässt sich aus

$$\mathbf{r}_S = \mathbf{r}_C + \boldsymbol{\rho}_M + p \mathbf{n}_M \quad (3.149)$$

mit den in Abbildung 3.10 gegebenen Größen die Eindringung (Penetration) p der *slave*-Fläche in die *master*-Fläche berechnen. Es folgt eine Abfrage

$$p \begin{cases} > 0 & \text{kein Kontakt} \\ < 0 & \text{Kontakt} \end{cases} . \quad (3.150)$$

Bei negativer Penetration – also bei Kontakt – wird beim hier betrachteten *Penalty*-Verfahren ein Knotenkraftvektor berechnet

$$\mathbf{f}_{\text{Kontakt}}^{\text{int}} = \int_{\mathcal{B}} \varepsilon_N p \mathbf{N}^T \cdot \mathbf{n}_S \, dV. \quad (3.151)$$

Der *Penalty*-Parameter ε_N entspricht der Steifigkeit einer Feder, die bei Auftreten von Kontakt aktiviert wird. In \mathbf{N} sind die Ansatzfunktionen des Kontaktelementes wie in Gleichung (3.35) angeordnet. Entsprechend der in den Abschnitten 3.1 und 3.2 vorgestellten Schalen- und Volumenschalenelemente wurden Kontaktelemente mit 4, 8 und 9 Knoten unter Verwendung der Ansatzfunktionen aus den Gleichungen (3.9) – (3.11) implementiert.

Die Knotenkräfte werden numerisch integriert, die Abfrage aus Gleichung (3.150) erfolgt an jedem Integrationspunkt. Diese Projektion der Integrationspunkte der ‘*slave*’- auf die ‘*master*’-Fläche wurde z. B. von ZAVARISE & WRIGGERS [133] vorgeschlagen, siehe hierzu auch HARNAU ET. AL. [51]. Um bei Kontaktflächen mit sehr kleinen Krümmungsradien eine zuverlässige Erfassung des Kontaktereignisses zu erreichen, wurde die Möglichkeit der Unterteilung einzelner Elemente in Segmente implementiert, in die wiederum eine bestimmte Anzahl Integrationspunkte gelegt werden kann. Neben der GAUSS-Integration wurde auch eine LOBATTO-Integration implementiert, bei der auch auf den Elementkanten Integrationspunkte liegen, da sich dies für Kontaktereignisse im Bereich von Kanten als günstiger erwiesen hat.

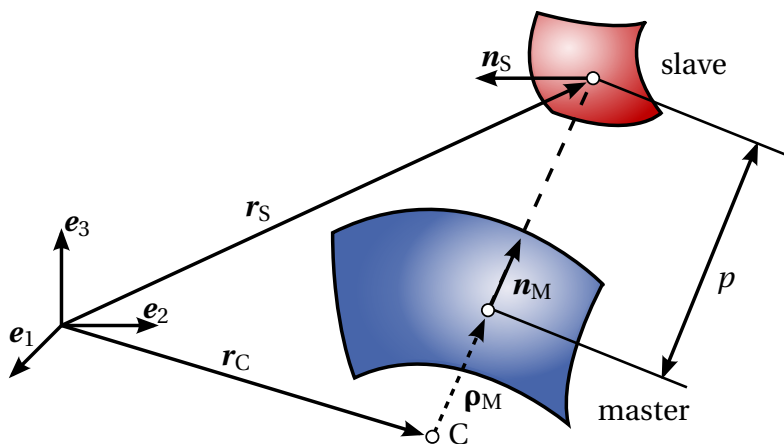


Abbildung 3.10.: Geometrische Beschreibung des Kontakts zwischen starren und deformierbaren Strukturen (STAS)

4. Symbolische Programmierung

Dieses Kapitel beschäftigt sich mit der effizienten Umsetzung der zuvor beschriebenen Elementformulierungen in das auf FEAP basierende und am Institut für Mechanik weiterentwickelte FE-Programm FEAP-MEKA [112]. Die ursprünglich für implizite Statik und Dynamik konzipierte Software wurde um das in Abschnitt 2.2 beschriebene explizite zentrale Differenzenverfahren erweitert und die Elementroutinen werden insbesondere auf die Anforderungen dieses Zeitintegrationsverfahrens angepasst. Das vorgestellte Implementierungskonzept, welches im Rahmen des DFG-Projektes SCHW307/20-1 [90] entwickelt wurde, wird hierbei durch die Anwendung der symbolischen Programmierumgebung ACEGEN [73] unterstützt.

Es erfolgt zunächst eine Definition des Begriffes ‘effiziente Programmierung’ für den Bereich der Elemententwicklung, wobei hier zwei verschiedene Betrachtungsweisen unterschieden werden. Nach einer kurzen Einführung in die Software ACEGEN wird das entwickelte Programmierkonzept anhand der wesentlichen Schritte, die für die Umsetzung einer Elementroutine erforderlich sind, vorgestellt. Hierbei werden die verwendeten ACEGEN-Kommandos – eine auf MATHEMATICA basierende Meta-Sprache zur automatischen Erzeugung optimierter ProgrammROUTINEN z. B. in FORTRAN, C, C++, etc. – beschrieben, die für alle umgesetzten Elementformulierungen identisch sind. Abschließend wird speziell auf die Eigenarten hinsichtlich der Programmierung der einzelnen implementierten Elementformulierungen eingegangen. Die ACEGEN-Kommandos, die insbesondere die verwendeten Modifikationen und Erweiterungen von Verzerrungs- bzw. Verschiebungsverläufen betreffen, werden an entsprechender Stelle erläutert.

Bei der Beschreibung der Implementierung werden MATHEMATICA-Befehle durchgehend in Schreibmaschinenschrift gesetzt, ACEGEN-Kommandos sind zusätzlich blau dargestellt. Da MATHEMATICA-Befehle üblicherweise mit ‘;’ abgeschlossen werden, wird in abgesetzten Formeln aus Gründen der besseren Lesbarkeit auf eine korrekte Zeichensetzung verzichtet.

4.1. Definition: ‘Effiziente Programmierung’

Ein wesentlicher Bestandteil des wissenschaftlichen Arbeitens im Bereich der Methode der finiten Elemente besteht in der programmtechnischen Umsetzung von Algorithmen und deren Implementierung in eine Software. Der Begriff der Effizienz oder Wirtschaftlichkeit ist in diesem Zusammenhang am besten durch die Definition

$$\text{Wirtschaftlichkeit} = \frac{\text{Ertrag}}{\text{Aufwand}} \quad (4.1)$$

beschrieben [94]. Vergleicht man zwei verschiedene Implementierungen des selben Algorithmus, so ist der Ertrag – also das Ergebnis das durch eine entsprechende Berechnung erzielt wird – gleich. Die Effizienz wird somit durch den Aufwand bestimmt, also der Zeit, die zur

Bestimmung des Ergebnisses erforderlich ist. Dieser kann in diesem Zusammenhang auf zwei Arten verstanden werden:

Aufwand für die Programmierung:

Der Programmiervorgang selbst besteht darin, die zuvor theoretisch hergeleiteten Zusammenhänge in einem Algorithmus zusammenzufassen und diesen in einer geeigneten Programmiersprache umzusetzen. Hierbei müssen die erforderlichen Beziehungen zunächst in eine für die gewählte Programmiersprache passende Form gebracht werden. Komplexe Umformungen der Gleichungen sowie ggf. die Bestimmung von Ableitungen erfolgen in der Praxis häufig unter Verwendung von Computeralgebraprogrammen wie MAPLE [86], MATHEMATICA [127] oder MATLAB [87]. Die aufbereiteten Gleichungen müssen schließlich in Programmcode geschrieben werden. Eine effiziente Umsetzung besteht also darin, in möglichst kurzer Zeit ein fehlerfreies Programm zu erstellen. Programmierfehler können hierbei sowohl bei der Umsetzung der theoretischen Beziehungen – also z. B. bei der Berechnung von Ableitungen – entstehen, als auch bei der Programmierung selbst im Sinne von ‘*Tippfehlern*’ beim Schreiben des Quelltextes. Für eine Steigerung der Effizienz müssen solche Fehler – deren Identifikation meist mühsam und zeitaufwändig ist – möglichst ausgeschlossen werden.

Rechenaufwand:

Der zweite Aspekt einer effizienten Programmierung ist die Effizienz des erzeugten Programmes selbst, also der erforderliche Zeitaufwand, um mit der implementierten Routine bei einem Programmablauf Ergebnisse zu erhalten. Bei gleichen Voraussetzungen hinsichtlich der verwendeten Hard- und Software (Prozessor, Arbeitsspeicher, Compiler, etc.) hängt die benötigte Rechenzeit unterschiedlicher Routinen im Wesentlichen von der Anzahl der durchgeführten Rechenoperationen ab. Abhängig von der gewählten Programmiersprache wirkt sich eine ‘*komfortable*’ Programmierung und eine gute Lesbarkeit des Quelltextes häufig negativ auf die Effizienz des Programmes aus. So kann beispielsweise die Verwendung mehrfach indizierter Felder, die für die Programmierung von Tensoroperationen oft sehr angenehm ist, zu einem deutlichen Anstieg der Rechenzeit führen. Eine Steigerung der Programmeffizienz durch Vermeidung redundanter Rechenoperationen, Ausnutzung von Symmetrien bei Matrizen oder Verwendung ausschließlich eindimensionaler Felder führt zwangsläufig zu Einbußen bei der Übersichtlichkeit der Programmroutine. Eine entsprechende Wahl von Optionen bei der Übersetzung durch einen Compiler alleine kann nur partiell zu effizienteren Routinen führen, effiziente und übersichtlich strukturierte Programmierung schließen sich demnach oftmals aus.

Im wissenschaftlichen – also nicht-kommerziellen – Bereich werden zu Gunsten der Nachvollziehbarkeit der programmierten Algorithmen häufig Defizite bei der numerischen Effizienz in Kauf genommen. Die Effizienz im Sinne der zweiten Definition spielt also zunächst eine untergeordnete Rolle. Eine nachträgliche ‘*Optimierung*’ bestehender Routinen ist zeitaufwändig und führt oft wieder zu Programmierfehlern, also einer Reduktion der Effizienz gemäß der ersten Definition. Wünschenswert ist also – insbesondere im Bereich der Forschung – eine Möglichkeit der übersichtlich strukturierten Programmierung, ohne

dabei auf optimale Programmeffizienz verzichten zu müssen, da meist nur dann ein Testen der entwickelten Methoden an realistischen Beispielen möglich ist. Hier bietet sich die Anwendung einer symbolischen Programmierumgebung an, welche die Aufbereitung der Algorithmen und deren Umsetzung in effizienten Programmcode zu einem einzigen Schritt zusammenfasst und somit auch eine potentielle Fehlerquelle völlig eliminiert. Eine solches Werkzeug sowie dessen Anwendung auf die Implementierung von Finite-Elemente-Routinen wird in den folgenden Abschnitten vorgestellt.

4.2. Einführung in ACEGEN

Im Rahmen der vorliegenden Arbeit wurde für die Implementierung der Elementroutinen das symbolische Programmierwerkzeug ACEGEN [73] verwendet, ein sogenanntes Plug-In zum Computeralgebraprogramm MATHEMATICA [127, 128]. Dieses wurde von KORELC in seiner Dissertation [65] erstmals vorgestellt und wird im Rahmen seiner Arbeitsgruppe ständig weiterentwickelt sowie für verschiedene Anwendungen im Bereich der computergestützten Mechanik eingesetzt [66, 68, 69, 73, 77]. Die Software nutzt die symbolischen Eigenschaften von MATHEMATICA um z. B. Ableitungen zu bestimmen oder Vektor- und Matrixoperationen durchzuführen, die Ergebnisse werden dann in automatisch optimierten Programmcode in z. B. C oder FORTRAN ausgegeben. Der Befehlssatz orientiert sich stark an der MATHEMATICA-Syntax und ist in einem Benutzerhandbuch dokumentiert [71], das permanent weiter ausgebaut und um anschauliche Beispiele erweitert wird. Insbesondere bei der Entwicklung von Element- oder Materialformulierungen, bei denen Ableitungen komplexer Funktionen eine zentrale Rolle spielen und die meist eine klare Schnittstelle innerhalb eines FE-Programms besitzen, kann die Nutzung eines solchen Werkzeugs viel Zeit sparen. Selbst viele kommerzielle FE-Pakete bieten üblicherweise die Möglichkeit, benutzerdefinierte Routinen aufzurufen, die gut über ein symbolisches Programmierkonzept erzeugt werden können, da ihnen meist eine standardisierte Variablenübergabe zu Grunde liegt. Der Schritt von der symbolischen Darstellung der theoretischen Formulierung zum Quellcode wird vollständig vom Programmierwerkzeug übernommen. Damit verlagert sich die Verwirklichung der Lösung auf eine der Software ACEGEN angepasste symbolische Ebene, eine besondere Aufbereitung der Algorithmen für die gewählte Programmiersprache ist nicht mehr erforderlich.

Durch die Bereitstellung vorgefertigter Module – eine Art Vorlagenbibliothek – können vollständige Elementroutinen für verschiedene FE-Codes wie FEAP, ABAQUS oder auch das ebenfalls von KORELC entwickelte MATHEMATICA-basierte Paket ACEFEM [70, 73] in einer generalisierten Programmiersprache erstellt werden. Durch die Wahl eines anderen Moduls lässt sich beispielsweise eine Elementformulierung, die in ACEFEM getestet wurde, ohne Änderungen an den Programmschritten vorzunehmen, in ABAQUS implementieren. Die in der vorliegenden Arbeit verwendete Version von FEAP basiert auf einer Fassung von 1994 [112] und beinhaltet weitreichende Modifikationen. Insbesondere die Elementschnittstelle weicht von der einer aktuellen FEAP-Version ab, weshalb sich die Erstellung einer eigenen Schnittstelle zwischen ACEGEN und FEAP als praktikabel herausstellte, die angebotenen Module also nicht verwendet wurden. Ist eine solche Schnittstelle einmal formuliert, lassen sich Varianten – beispielsweise verschiedene Elementformulierungen – sehr intuitiv und fehlerfrei realisieren.

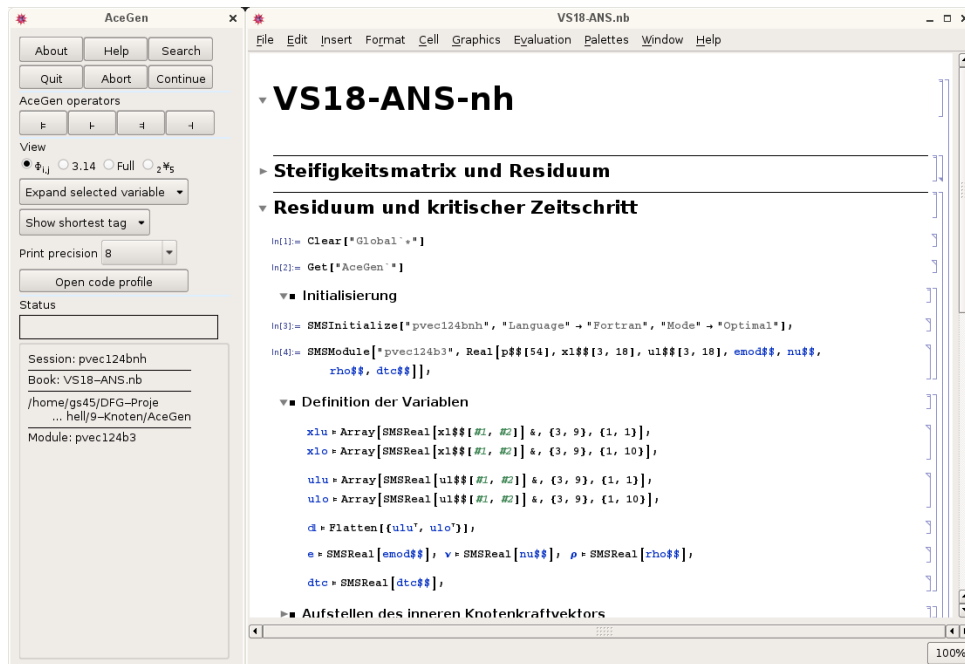


Abbildung 4.1.: Screenshot – MATHEMATICA mit ACEGEN-Steuerfenster

Abbildung 4.1 zeigt ein Bildschirmfoto von MATHEMATICA mit gestartetem ACEGEN-Modul. Es sind zwei Fenster geöffnet, rechts befindet sich die standardmäßige MATHEMATICA-Oberfläche, links das Steuerungsfeld für ACEGEN, welches Informationen über das aktuelle Projekt anzeigt, die Eingabe spezieller Symbole sowie das Starten der Hilfe erlaubt. Für nähere Informationen zu Installation, Start und Bedienung der Software, die für die Betriebssysteme WINDOWS, LINUX und MAC OS erhältlich ist, sei auf das Benutzerhandbuch [71] verwiesen. Ein sehr einfaches Prinzipbeispiel – die Implementierung mehrerer hintereinander geschalteter Matrixoperationen – ist in Anhang B.1 dieser Arbeit angegeben. Es zeigt die erforderlichen Initialisierungsschritte sowie die Verwendung einfacher symbolischer Operationen und deren Übersetzung in effizienten Programmcode. In ähnlicher Weise wird die beschriebene Prozedur auch für die in der Arbeit vorgestellten Elementformulierungen verwendet, auf was im Folgenden noch genauer eingegangen wird. Komplexere Anwendungsbeispiele können ebenfalls der aktuellen Fassung des Handbuchs [71] entnommen werden.

Die Anwendung symbolischer Programmierung mit ACEGEN erlaubt eine direkte Eingabe der in den vorigen Kapiteln beschriebenen kontinuumsmechanischen Beziehungen. Eine aufwändige Aufbereitung und Anpassung der theoretischen Zusammenhänge für die gewählte Programmiersprache ist nicht erforderlich, da die von ACEGEN verwendete symbolische Meta-Sprache ‘*sehr nah*’ an den theoretisch hergeleiteten Zusammenhängen ist. Varianten – etwa Änderungen an den Interpolationsvorschriften oder Modifikationen am Stoffgesetz – können mit vergleichsweise geringem Aufwand realisiert und Fehler bei der Programmierung vermieden werden. Dies führt zu einem wesentlich geringeren Zeitaufwand für die Programmierung selbst. Darüber hinaus sind die generierten Programmerroutinen durch die bei der Codeerzeugung simultan ablaufende automatische Optimierung bezüglich der durchzuführenden Rechenoperationen sehr effizient und eine weitere händische Verbesserung der Implementierung ist nicht erforderlich. Die Anwendung von ACEGEN führt also zu

einer Reduktion des Aufwandes und somit zu einer Steigerung der Effizienz gemäß beider in Abschnitt 4.1 formulierter Definitionen.

4.3. Elementroutinen für explizite Zeitintegration

Durch die in Abschnitt 3.3 ausführlich beschriebene Verwendung diagonalisierter Massenmatrizen sind auf globaler Ebene beim zentralen Differenzenverfahren ausschließlich Vektoroperationen durchzuführen. Im Gegensatz zu impliziten Algorithmen, wie dem in Abschnitt 2.3 beschriebenen NEWMARK-Verfahren, entfällt die numerisch aufwändige Lösung eines nichtlinearen Gleichungssystems, es sind also weniger Operationen pro Zeitschritt durchzuführen. Die Gesamtzeit, die für die Simulation eines einzelnen Zeitschritts erforderlich ist, wird bei expliziter Zeitintegration also durch die Rechenschritte auf Elementebene dominiert. Folglich ist gerade hier eine hinsichtlich der Rechenzeit optimierte Implementierung der Elementroutinen von besonderem Interesse; bei impliziten Algorithmen hingegen spielt die Effizienz der Elementformulierungen eine im Vergleich zu den Operationen auf Systemebene untergeordnete Rolle. Der Programmieraufwand hingegen, also die Effizienz im Sinne der ersten Definition aus Abschnitt 4.1, ist bei Entwicklungen für explizite und implizite Verfahren von gleicher Bedeutung, weshalb sich die Verwendbarkeit von ACEGEN keinesfalls auf die in der vorliegenden Arbeit beschriebenen Anwendungen beschränkt.

Für das gewählte Programmierkonzept stellte sich die Zusammenfassung der kinematischen Annahmen und des Stoffgesetzes in einer einzigen Programmroutine als praktisch heraus. In kommerziellen Anwendungen ist aus Gründen der Übersichtlichkeit und der Vermeidung von Redundanzen im Programmcode eine Trennung der Routinen üblich, da die verschiedenen Elementformulierungen mit den unterschiedlichen Materialmodellen kombinierbar sein müssen. Diese Modularität wird im vorliegenden Fall auf der symbolischen Ebene gewährleistet. Durch Austauschen einzelner Teile der ACEGEN-Routine lassen sich ebenfalls verschiedene kinematische Annahmen und Materialformulierungen kombinieren, es muss dann jeweils eine neue Elementroutine generiert und in das Hauptprogramm eingebunden werden, was bei der Verwendung einer quelloffenen Software im akademischen Bereich kein Problem darstellt.

Im Folgenden wird die Implementierung der Operationen auf Elementebene mit ACEGEN beschrieben und die wesentlichen Unterschiede zu einer händischen Programmierung aufgezeigt. Die verwendeten ACEGEN-Kommandos sind in den einzelnen Formeln **blau** dargestellt. Ausführliche Informationen zu den einzelnen SMS-Befehlen ('*Symbolic Mechanics System*' [65]) sowie die Bedeutung der verschiedenen Zuweisungsoperatoren \mathbb{F} , \mathbb{T} , $\mathbb{=}$ und $\mathbb{+}$ können dem Handbuch [71] sowie dem in Anhang B.2 skizzierten Prinzipbeispiel zur Implementierung einer Elementroutine entnommen werden.

4.3.1. Steifigkeitsmatrix und Residuum

Allgemeines

Von zentraler Bedeutung für die Effizienz eines expliziten Zeitintegrationsverfahrens ist die Berechnung des Vektors der inneren Knotenkräfte, welcher zur Bestimmung der Beschleunigungen in Gleichung (2.3) benötigt wird. Es erfolgt zunächst die Aufstellung der

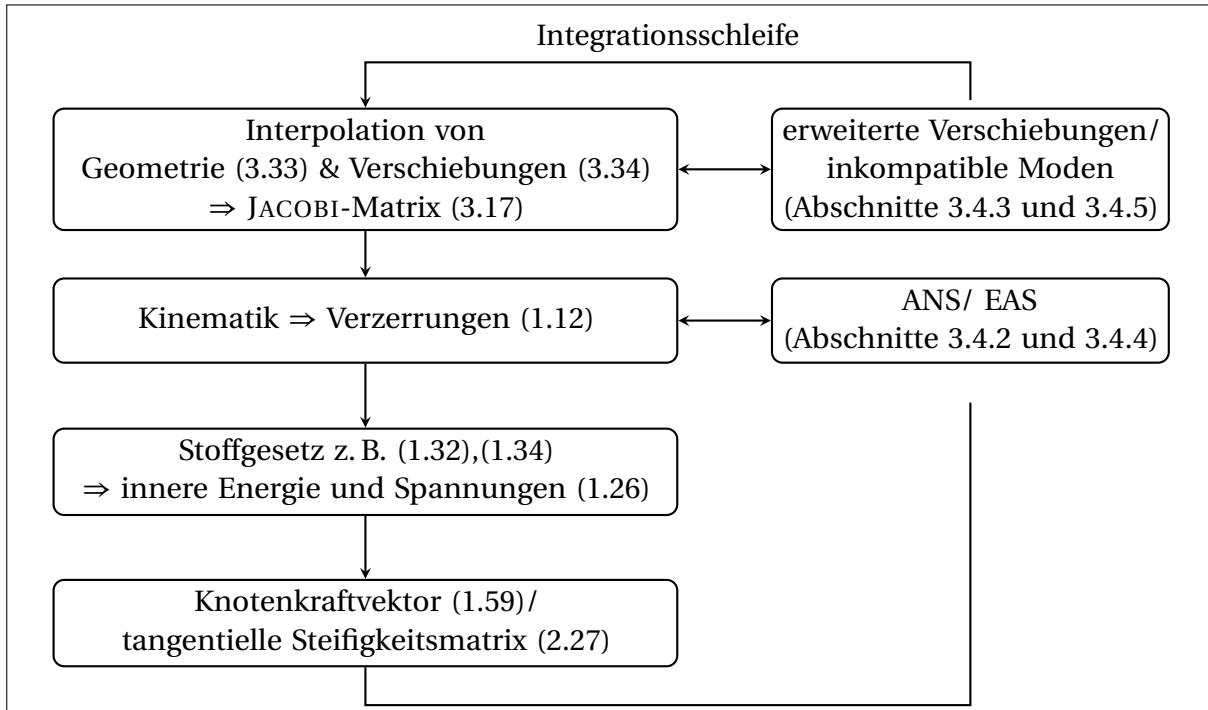


Abbildung 4.2.: Schematischer Programmablauf zur Aufstellung des inneren Knotenkraftvektors bzw. der tangentiellen Steifigkeitsmatrix auf Elementebene mit Verweis auf die jeweiligen Gleichungen und Abschnitte

einzelnen Kraftvektoren auf Elementebene, welche dann in der globalen Routine zum gesamten inneren Knotenkraftvektor \mathbf{f}^{int} assembliert werden. Bei händischer Programmierung einer Elementroutine werden üblicherweise die einzelnen Schritte der in Abbildung 4.2 schematisierten Vorgehensweise in Unterroutinen umgesetzt, die an entsprechender Stelle der Hauptroutine aufgerufen werden. Die Programmierung mit ACEGEN erlaubt eine symbolische Umsetzung der Einzelschritte in MATHEMATICA und liefert schließlich eine einzige Programmroutine, in der alle erforderlichen Operationen in möglichst kompakter Form durchgeführt werden. Redundante Operationen werden vermieden, in dem häufig verwendete Größen automatisch als Hilfsvariablen abgespeichert werden. Der wesentliche Unterschied zwischen Elementroutinen für explizite bzw. implizite Zeitintegration besteht in den erforderlichen Elementmatrizen. Für implizite Verfahren wird zusätzlich zum Element-residualvektor $\mathbf{f}_e^{\text{int}}$ die tangentielle Steifigkeitsmatrix $\mathbf{K}_e^{\text{tang}}$ aus Gleichung (2.27) benötigt. Deren Aufstellung innerhalb der symbolischen Programmierumgebung erfordert lediglich wenige zusätzliche Programmzeilen und wird in Anhang B.2.3 beschrieben, sodass mit der erläuterten Vorgehensweise Elementroutinen sowohl für explizite als auch für implizite Verfahren umgesetzt werden können. Im Anhang wird auch die Schleife für die erforderliche numerische Integration beschrieben, auf die im folgenden Abschnitt auf Gründen der Übersichtlichkeit verzichtet wird.

Interpolation von Geometrie und Verschiebungen

Die zur Geometrie- und Verschiebungsinterpolation verwendeten Ansatzfunktionen können in ACEGEN symbolisch als Funktionen der natürlichen Koordinaten ξ , η und ζ definiert

werden. Eine separate Eingabe der Ableitungen, die etwa zur Berechnung der Verzerrungen erforderlich sind, ist hier nicht erforderlich, da symbolische Ableitungen an entsprechender Stelle direkt gebildet werden. Im automatisch generierten Programmcode werden diese dann an den Integrationspunkten ausgewertet und, wenn sie häufiger benötigt werden, als Hilfsvariable abgespeichert, um Rechenoperationen einzusparen. Die für Schalen- und Volumenschalenformulierungen für die Schalenebene verwendeten linearen Ansatzfunktionen aus Gleichung (3.9) werden beispielsweise über

$$\begin{aligned} \xi_i &= \{-1, 1, 1, -1\}; & \eta_i &= \{-1, -1, 1, 1\}; \\ \text{shp} &\models \text{Table}[1/4 (1 + \xi \xi_i[[i]]) (1 + \eta \eta_i[[i]]), \{i, 1, 4\}]; \end{aligned} \quad (4.2)$$

einggegeben. Für degenerierte Schalenelemente aus Abschnitt 3.1 lautet die Interpolation der Geometrie und der Verschiebungen nach den Gleichungen (3.1) und (3.3) dann

$$\begin{aligned} \mathbb{X} &\models \text{shp} \cdot \mathbf{x1}^T + 1/2 \zeta \mathbf{t} \text{shp} \cdot \mathbf{vn}^T; \\ \mathbf{u1} &\models \text{shp} \cdot \mathbf{u1d}^T + 1/2 \zeta \mathbf{t} \text{shp} \cdot (\mathbf{dki} - \mathbf{vn}^T); \end{aligned} \quad (4.3)$$

wobei $\mathbf{x1}$ die geometrische Lage und $\mathbf{u1d}$ die diskreten Verschiebungen der einzelnen Knoten enthalten. In \mathbf{vn} sind die Schalendirektoren der Knoten in der Referenzkonfiguration abgespeichert, die der Elementroutine übergeben werden. Die Werte für die Momentan-konfiguration werden über eine eigene Funktion nach der in Abbildung 3.3 skizzierten Vorgehensweise berechnet und in \mathbf{dki} abgelegt. Für die Volumenschalenformulierungen ergibt sich nach den Gleichungen (3.33) und (3.34) entsprechend

$$\begin{aligned} \mathbb{X} &\models 1/2 \text{shp} \cdot ((1 + \zeta) \mathbf{x1o}^T + (1 - \zeta) \mathbf{x1u}^T); \\ \mathbf{u1} &\models 1/2 \text{shp} \cdot ((1 + \zeta) \mathbf{u1o}^T + (1 - \zeta) \mathbf{u1u}^T); \end{aligned} \quad (4.4)$$

wobei hier die Felder $\mathbf{x1o}$, $\mathbf{x1u}$ den Ort und $\mathbf{u1o}$, $\mathbf{u1u}$ die Verschiebung der Knoten der Schalenober- und -unterseite enthalten. Hier wird bereits deutlich, dass beispielsweise eine Erhöhung der Ansatzordnung auf quadratische Polynome lediglich eine Änderung in Gleichung (4.2) erfordert, alle folgenden Schritte sind von der Wahl der Interpolationsfunktionen unabhängig. Die in den Abschnitten 3.4.3 und 3.4.5 vorgestellten Strategien zur Behandlung von Versteifungseffekten, die sich auf die Modifikation, bzw. Erweiterung des Verschiebungsansatzes beziehen, können hier direkt auf die interpolierten Verschiebungen angewendet werden. Diese Vorteile werden bei der Beschreibung der einzelnen Elementformulierungen in Abschnitt 4.4 noch genauer erläutert.

Die JACOBI-Matrix aus Gleichung (3.17), deren Determinante und Inverse und somit die Komponenten der konvektiven Basisvektoren lassen sich durch die symbolischen Operationen

$$\begin{aligned} \mathbb{J} &\models \text{SMSD}[\mathbb{X}, \{\xi, \eta, \zeta\}]^T; \\ \det \mathbb{J} &\models \text{SMSDet}[\mathbb{J}]; \\ \mathbb{J}i &\models \text{SMSInverse}[\mathbb{J}]; \end{aligned} \quad (4.5)$$

berechnen. Bei einer händischen Programmierung werden hier die abgeleiteten Ansatzfunktionen benötigt, die vorher berechnet und entsprechend abgespeichert werden müssen. In der von ACEGEN verwendeten Meta-Sprache ergibt sich eine überaus kompakte Schreibweise, die den theoretischen Beziehungen sehr ähnlich ist. Die Übersetzung in die tatsächliche Programmiersprache wird schließlich von der Software unter Verwendung möglichst weniger Rechenoperationen automatisch durchgeführt.

Kinematik \Rightarrow Verzerrungen

Die Komponenten des in Gleichung (1.12) eingeführten GREEN-LAGRANGE-Verzerrungstensors können in konvektiven Koordinaten bestimmt werden

$$\begin{aligned}\Delta u & \mathbb{F} \text{SMSD}[u, \{\xi, \eta, \zeta\}]^T; \\ \mathbb{E} & \mathbb{F} 1/2 (\mathbb{J} \cdot \Delta u^T + \Delta u \cdot \mathbb{J}^T + \Delta u \cdot \Delta u^T); \end{aligned} \quad (4.6)$$

was eine direkte Anwendung der in den Abschnitten 3.4.2 und 3.4.4 beschriebenen Methoden zur Behandlung von Versteifungseffekten ermöglicht. Verzerrungsinterpolationen oder zusätzliche Ansätze im Rahmen der EAS-Methode können symbolisch umgesetzt werden. Die genaue Vorgehensweise ist bei der Beschreibung der einzelnen Elementformulierungen in Abschnitt 4.4 angegeben. Eine Möglichkeit der direkten Bestimmung der kartesischen Komponenten des Verzerrungstensors ist in Anhang B (siehe hierzu Gleichungen (B.39) und (B.40)) beschrieben.

Stoffgesetz \Rightarrow innere Energie und Spannungen

Die endgültigen Verzerrungsverläufe einschließlich aller Modifikationen werden anschließend an die Materialformulierung übergeben. Im Fall der in Abschnitt 1.3.1 beschriebenen Hyperelastizität, geschieht dies durch Aufstellen einer Verzerrungsenergiefunktion in Abhängigkeit der aktuellen Verzerrungen. Beispielsweise für das in Gleichung (1.34) angegebene NEO-HOOKE-Materialgesetz werden zunächst die Komponenten des GREEN-LAGRANGE-Verzerrungstensors in kartesische Koordinaten transformiert und so nach den Gleichungen (1.6) und (1.7) die kartesischen Komponenten des rechten CAUCHY-GREEN-Tensors bestimmt

$$\begin{aligned}\mathbb{E}_k & \mathbb{F} \mathbb{J}_i \cdot \mathbb{E} \cdot \mathbb{J}_i^T; \\ \mathbb{C}_k & \mathbb{F} 2 \mathbb{E}_k + \text{IdentityMatrix}[3]; \end{aligned} \quad (4.7)$$

Die Verzerrungsenergie ergibt sich schließlich zu

$$\begin{aligned}\mathcal{W} & \mathbb{F} \mu/2 (\text{Tr}[\mathbb{C}_k] - 3) - \mu \text{Log}[\text{SMSqrt}[\text{SMSDet}[\mathbb{C}_k]]] + \\ & \lambda/2 (\text{Log}[\text{SMSqrt}[\text{SMSDet}[\mathbb{C}_k]]])^2; \end{aligned} \quad (4.8)$$

Für die degenerierten Schalenformulierungen muss an dieser Stelle noch z. B. die für den ebenen Spannungszustand erforderliche Bedingung $S_{\zeta\zeta} = 0$ umgesetzt werden. Dieser Schritt entfällt bei den Solid-Shell-Elementen, die einen vollständigen 3D-Spannungszustand abbilden können. Für den linearen Zusammenhang zwischen Verzerrungen und Spannungen des ST. VENANT-KIRCHHOFF-Materialgesetzes aus Gleichung (1.32) kann dies über

$$\begin{aligned}\$k & = \text{SMSD}[\mathcal{W}, \mathbb{E}_k, \text{"IgnoreNumbers"} \rightarrow \text{True}, \text{"Symmetric"} \rightarrow \text{True}]; \\ \text{plstr} & = \text{Solve}[\$k[[3, 3]] == 0, \mathbb{E}_k[[3, 3]]][1]; \\ \mathcal{W} & \mathbb{F} \text{SMSReplaceAll}[\mathcal{W}, \text{plstr}]; \end{aligned} \quad (4.9)$$

erfolgen, komplexere Materialformulierungen erfordern die Umsetzung beispielsweise eines NEWTON-Verfahrens zur Erfüllung dieser Bedingung. Es ist zu beachten, dass Spannungen und Verzerrungen hier in lokalen kartesischen Koordinaten mit der z -Achse in Dickenrichtung der Schale zu formulieren sind.

Materialgesetze mit Plastizität (vgl. Abschnitt 1.3.2) lassen sich ebenfalls symbolisch umsetzen. Die in den Abbildungen 1.2 und 1.3 skizzierten Algorithmen erfordern jeweils die Aufstellung und iterative Lösung eines nichtlinearen Gleichungssystems ($\Psi = \mathbf{0}$) und führen nach Erreichen der Konvergenz auf elastische und plastische Verzerrungsanteile. Die zur Formulierung des NEWTON-Verfahrens erforderlichen Ableitungen, bzw. die der Potentialfunktionen können direkt symbolisch eingegeben werden, eine händische Berechnung vorab entfällt also. Änderungen an der Materialformulierung, etwa die Verwendung eines anderen Verfestigungsgesetzes, lassen sich so sehr einfach umsetzen. Mit den konvergierten elastischen Verzerrungen lässt sich schließlich bei bekanntem elastischen Potential eine Verzerrungsenergie wie in Gleichung (4.8) berechnen, die plastischen Verzerrungsanteile müssen als Geschichtsgrößen für jedes Element abgespeichert werden. In Anhang B.3 ist als Beispiel die Implementierung einer Materialformulierung nach Abbildung 1.3 angegeben, bei der die einzelnen Schritte erläutert werden. Eine ausführliche Herleitung und Aufbereitung der Gleichungen zur Plastizität für die symbolische Umsetzung, die als Grundlage für die implementierten Modelle diente, kann der Publikation [72] entnommen werden.

Knotenkraftvektor bzw. Elementsteifigkeitsmatrix

Die der Elementroutine übergebenen aktuellen Werte der Knotenfreiheitsgrade werden zu Beginn in einem eindimensionalen Feld d abgespeichert. Dies erfolgt für Schalenelemente über

$$d \Leftarrow \text{Flatten}[\{ul^T\}]; \quad (4.10)$$

und Volumenschalen entsprechend durch

$$d \Leftarrow \text{Flatten}[\{ulu^T, ulo^T\}]; \quad (4.11)$$

Eine genaue Beschreibung der Übergabe und Initialisierung von Variablen an die aktuelle ACEGEN-Routine wird in Anhang B.2.3 angegeben. Mit Gleichung (1.59) lässt sich der Vektor der inneren Knotenkräfte durch Ableitung nach den Knotenfreiheitsgraden d bestimmen:

```
SMSDo[
  p  $\Leftarrow$  SMSD[ $\mathcal{W}$ , d, i] * detJ * wi;
  SMSEXPOT[-p, p[[i], "AddIn" -> True];
  , {i, 1, Length[d]}]; \quad (4.12)
```

Details zu den einzelnen Kommandos sowie zur Aufstellung der tangentiellen Steifigkeitsmatrix nach Gleichung (2.27) können ebenfalls Anhang B.2.3 entnommen werden. Die Bestimmung der EAS-Matrizen aus den Gleichungen (3.118) und (3.119) erfolgt durch entsprechende Ableitungen nach den erweiterten Freiheitsgraden, bei der Umsetzung inkompatibler Freiheitsgrade nach Abschnitt 3.4.5 muss der Vektor d entsprechend erweitert werden. Genaue Erläuterungen zur Implementierung der jeweiligen Methoden erfolgen bei der Beschreibung der entsprechenden Elementformulierungen in Abschnitt 4.4.

4.3.2. Massenmatrix und Diagonalisierung

Im Gegensatz zur Integration des Knotenkraftvektors hat die Aufstellung und Diagonalisierung der Massenmatrix praktisch keinen Einfluss auf die Effizienz einer Elementformulierung, da sie nur einmal zu Beginn einer Simulation durchgeführt werden müssen. Aus

demselben Grund spielt auch die in Anhang C diskutierte Wahl der Integrationsordnung für die später erforderliche Rechenzeit eine untergeordnete Rolle. Dennoch lässt sich die Programmroutine zur Berechnung der konsistenten Massenmatrix auch sehr komfortabel mit ACEGEN erstellen, da erforderliche Matrixoperationen nicht händisch programmiert werden müssen.

Innerhalb der numerischen Integrationschleife werden wie in Gleichung (4.2) die Ansatzfunktionen aufgestellt, die dann gemäß Gleichung (3.35) in die Matrix \mathbb{N} einsortiert werden. Für ein Volumenschalenelement erfolgt dies z. B. über

$$\begin{aligned} \zeta_u &\equiv 1/2 (1 - \zeta); & \zeta_o &\equiv 1/2 (1 + \zeta); \\ \mathbb{N} &\equiv \{ \{ \zeta_u \text{ shp}[1], 0, 0, \dots, \zeta_u \text{ shp}[4], 0, 0, \zeta_o \text{ shp}[1], 0, 0, \dots, \zeta_o \text{ shp}[4], 0, 0 \} \\ &\quad \{ 0, \zeta_u \text{ shp}[1], 0, \dots, 0, \zeta_u \text{ shp}[4], 0, 0, \zeta_o \text{ shp}[1], 0, \dots, 0, \zeta_o \text{ shp}[4], 0 \} \\ &\quad \{ 0, 0, \zeta_u \text{ shp}[1], \dots, 0, 0, \zeta_u \text{ shp}[4], 0, 0, \zeta_o \text{ shp}[1], \dots, 0, 0, \zeta_o \text{ shp}[4] \} \}; \end{aligned} \quad (4.13)$$

Mit der Dichte ρ ergibt sich die Massenmatrix nach Gleichung (3.38) dann zu

$$\begin{aligned} \mathbb{M} &\equiv \rho \mathbb{N}^T \cdot \mathbb{N} \text{ det } \mathbb{J} \text{ wi}; \\ \text{SMSExport} &[\mathbb{M}, \text{s} \$ \$, \text{"AddIn"} \rightarrow \text{True}]; \end{aligned} \quad (4.14)$$

Wie zuvor bei den Operationen zur Aufstellung des Knotenkraftvektors und der Steifigkeitsmatrix sind diese Schritte in eine Schleife für die numerische Integration eingebettet, wobei w_i der Wichtungsfaktor für den aktuellen Integrationspunkt ist. Bei GAUSS-Integration wird so die konsistente Massenmatrix aufgestellt. Wählt man eine Integrationsregel bei der die Integrationspunkte mit den Elementknoten zusammenfallen, so ergibt sich wie bereits in Abschnitt 3.3.3 diskutiert automatisch eine diagonale Massenmatrix. Die weiteren in Abschnitt 3.3.3 vorgestellten Diagonalisierungsalgorithmen setzen zunächst die konsistente Massenmatrix voraus. Wegen des vergleichsweise geringen Programmieraufwandes wurde die ACEGEN-Routine mit GAUSS-Integration aufgerufen und die zur Diagonalisierung erforderlichen Schritte händisch programmiert.

4.3.3. Spannungsrückrechnung

Die Rückrechnung und Aufbereitung von Spannungen aus dem aktuellen Verschiebungszustand wird als separate Programmroutine auf Elementebene realisiert, da sie nicht wie die Aufstellung von Knotenkraftvektor, bzw. Steifigkeitsmatrix in jedem Zeitschritt durchgeführt werden muss. Die aktuellen Größen der Spannungen und Verzerrungen werden an bestimmten Punkten ausgewählter Elemente berechnet und in einer Formatierung ausgegeben, die schließlich vom verwendeten Post-Processor interpretiert werden kann. Je nach dem, wie häufig eine Spannungsausgabe im Post-Processing gewünscht ist, kann sich auch hier eine effiziente Umsetzung positiv auf die Simulationszeit auswirken, jedoch ist in praktischen Anwendungen eher eine Dominanz der Routinen zur Aufstellung der inneren Knotenkräfte zu erwarten.

Alternativ zur in Gleichung (4.6) beschriebenen Berechnung der GREEN-LAGRANGE-Verzerrungen lassen sich mit den interpolierten Orts- und Verschiebungsvektoren \mathbb{X} und \mathbb{u} aus Gleichung (4.3) bzw. (4.4) auch direkt die kartesischen Komponenten des Deformations-

gradienten angeben:

$$\begin{aligned} \mathbb{X}f &\vdash \text{SMSFreeze}[\mathbb{X}]; \\ \mathbb{F}k &\vdash \text{IdentityMatrix}[3] + \text{SMSD}[u, \mathbb{X}f, \text{"Dependency"} \rightarrow \{\xi, \eta, \zeta\}, \mathbb{X}f, \mathbb{J}i]; \end{aligned} \quad (4.15)$$

Die genaue Bedeutung der Option "Dependency" $\rightarrow \dots$ ist in Anhang B.2.3 in Gleichung (B.40) beschrieben. Es ergeben sich die kartesischen Komponenten des rechten CAUCHY-GREEN-Verzerrungstensors und des GREEN-LAGRANGE-Verzerrungstensors zu

$$\begin{aligned} \mathbb{C}k &\vdash \mathbb{F}k^T \cdot \mathbb{F}k; \\ \mathbb{E}k &\vdash 1/2 (\mathbb{C}k - \text{IdentityMatrix}[3]); \end{aligned} \quad (4.16)$$

Auf diese Weise können auch beliebige andere Verzerrungsmaße berechnet werden. Die Ausgabe erfolgt durch

$$\begin{aligned} \varepsilon &\vdash \{\mathbb{E}k[1,1], \mathbb{E}k[2,2], \mathbb{E}k[3,3], 2 \mathbb{E}k[1,2], 2 \mathbb{E}k[2,3], 2 \mathbb{E}k[1,3]\}; \\ \text{SMSExport} &[\varepsilon, \text{eps}\$\$]; \end{aligned} \quad (4.17)$$

wobei das Feld `eps` nach einer Simulation dem Post-Processor übergeben und von diesem interpretiert und ggf. grafisch aufbereitet wird. Durch Ableiten der inneren Energie, die wie in Gleichung (4.8) durch Einsetzen der Verzerrungen in die Verzerrungsenergiefunktion berechnet wird, ergeben sich die Komponenten des 2. PIOLA-KIRCHHOFF-Spannungstensors aus Gleichung (1.26)

$$\mathbb{S}k \vdash 2 \text{SMSD}[\mathbb{W}, \mathbb{C}k, \text{"IgnoreNumbers"} \rightarrow \text{True}, \text{"Symmetric"} \rightarrow \text{True}]; \quad (4.18)$$

Ein sinnvolles Spannungsmaß zur Interpretation der Ergebnisse stellt der in Gleichung (1.26) eingeführte CAUCHY Spannungstensor dar, da durch ihn wahre Spannungen, also Kräfte bezogen auf momentane Querschnitte, ausgedrückt werden. Dessen kartesische Komponenten werden durch

$$\begin{aligned} \sigma k &\vdash \mathbb{F}k \cdot \mathbb{S}k \cdot \mathbb{F}k^T / \text{SMSDet}[\mathbb{F}k]; \\ \sigma &\vdash \{\sigma k[1,1], \sigma k[2,2], \sigma k[3,3], \sigma k[1,2], \sigma k[2,3], \sigma k[1,3]\}; \\ \text{SMSExport} &[\sigma, \text{sig}\$\$]; \end{aligned} \quad (4.19)$$

berechnet und ausgegeben um wie die Verzerrungen vom Post-Processor interpretiert zu werden. Die Verwendung alternativer Spannungs- und Verzerrungsmaße, sowie die Ausgabe in anderen – z. B. lokalen – Koordinatensystemen lässt sich auf der symbolischen Ebene ebenfalls sehr komfortabel umsetzen.

4.4. Implementierte Elementformulierungen

Die oben zusammengestellten symbolischen Implementierungsschritte sind prinzipiell für alle für die Arbeit umgesetzten Elementformulierungen gleich. Die in Abschnitt 3.4 eingeführten Strategien zur Vermeidung künstlicher Versteifungseffekte greifen an bestimmten Stellen in den Programmablauf ein, um beispielsweise Verzerrungsverläufe zu modifizieren. Die dafür erforderlichen lokalen Änderungen, die sich direkt in ACEGEN umsetzen lassen, sind auch in Abbildung 4.2 angedeutet. Die notwendigen Schritte zur Umsetzung der verschiedenen Formulierungen innerhalb des symbolischen Programmierkonzeptes sind in den nächsten Abschnitten im Einzelnen beschrieben.

4.4.1. Nomenklatur

Die in der Arbeit verwendeten Bezeichnungen für die einzelnen implementierten Elementformulierungen setzen sich aus verschiedenen Teilen zusammen, welche die Eigenschaften des jeweiligen Elementes beschreiben sollen. Die Namen beginnen zunächst mit einem 4-stelligen Code, der die Elementfamilie und die Anzahl der Knoten, also die verwendete Ansatzordnung angibt. Hierbei steht DS für degenerierte Schalen- und VS für Volumenschalenelemente. Beispielsweise DS09 bezeichnet somit eine degenerierte Schalenformulierung mit 9 Knoten, also quadratischen LAGRANGE-Ansatzfunktionen. Durch Bindestriche abgetrennt folgen nun die verschiedenen Verzerrungs- oder Verschiebungsmodifikationen, die zur Behandlung von Versteifungseffekten implementiert sind. Hier ist

- ...-DISP – keine Modifikationen \equiv reine Verschiebungsformulierung
- ...-ANS – angenommene Verzerrungen nach Abschnitt 3.4.2
- ...-EAS_{xx} – erweiterte Verzerrungen nach Abschnitt 3.4.4
 \Rightarrow 'xx' bezeichnet die Anzahl der Erweiterungen
- ...-XDIS – erweiterte Verschiebungen nach Abschnitt 3.4.3
- ...-IM_{xx} – inkompatible Moden nach Abschnitt 3.4.5
 \Rightarrow 'xx' bezeichnet die Anzahl der inkompatiblen Freiheitsgrade

Der Zusatz -DISP wird nur bei Elementen ohne jegliche Modifikation verwendet, sind verschiedene Methoden in einer Elementformulierung implementiert, so werden die Bezeichnungen hintereinander gestellt. Eine Volumenschale mit linearen LAGRANGE-Ansätzen, ANS und 3 EAS-Parametern heißt also beispielsweise VS08-ANS-EAS03.

Im Folgenden sind die implementierten Elementformulierungen zusammengestellt, die in Kapitel 5 an numerischen Beispielen getestet werden. Besonderheiten bei der Implementierung des Vektors der inneren Knotenkräfte – also Modifikationen, die vom Standardvorgehen aus Abschnitt 4.3.1 abweichen – werden angegeben. Durch die Modularität des symbolischen Programmierkonzeptes können weitere Varianten und Kombinationen der beschriebenen Formulierungen problemlos erstellt werden, die jedoch nicht Bestandteil der vorliegenden Arbeit sind.

4.4.2. Degenerierte Schalenelemente

DS04-DISP

Die reine Verschiebungsformulierung des degenerierten 4-Knoten-Schalenelementes mit linearer Interpolation von Geometrie und Verschiebungen wird ausschließlich durch die in Abschnitt 4.3.1 beschriebenen Schritte umgesetzt. In Kombination mit der gewählten vollständigen Integrationsregel mit $2 \times 2 \times 2$ GAUSS-Punkten treten bekanntermaßen Versteifungseffekte auf, deren Ursache bereits in Abschnitt 3.4 diskutiert wurde. Die Umsetzung der Formulierung ohne Erweiterungen und Modifikationen erfolgte, um die Recheneffizienz der verschiedenen Methoden zur Beseitigung von Versteifungseffekten bewerten zu können, und sie diene als Grundlage für die weiteren Implementierungen.

DS04-ANS

Die Elementformulierung DS04-ANS entspricht dem bei BATHE & DVORKIN [13] vorgestellten MITC4-Element und enthält die in den Gleichungen (3.57) und (3.58) angegebenen Modifikationen der Querschubverzerrungen $E_{\eta\zeta}$ und $E_{\xi\zeta}$. Die Einbindung der Verzerrungsinterpolation in das prinzipielle Programmierkonzept aus Abbildung 4.2 ist in Abbildung 4.3 skizziert. Zunächst erfolgt die Aufstellung der Interpolationsfunktionen und der Kollokationspunkte

$$\begin{aligned} i23 &= \{1/2 (1 - \xi), 1/2 (1 + \xi)\}; & i13 &= \{1/2 (1 - \eta), 1/2 (1 + \eta)\}; \\ k23 &= \{-1, 1\}; & k13 &= \{-1, 1\}; \end{aligned} \quad (4.20)$$

und schließlich die Auswertung und Interpolation der ANS-Verzerrungen

$$\begin{aligned} \mathbb{E}[2, 3] &\Leftarrow \text{Sum}[i23[i] \text{ SMSReplaceAll}[\mathbb{E}[2, 3], \{\xi \rightarrow k23[i], \eta \rightarrow 0\}], \{i, 1, 2\}]; \\ \mathbb{E}[3, 2] &\Leftarrow \mathbb{E}[2, 3]; \\ \mathbb{E}[1, 3] &\Leftarrow \text{Sum}[i13[i] \text{ SMSReplaceAll}[\mathbb{E}[2, 3], \{\xi \rightarrow 0, \eta \rightarrow k13[i]\}], \{i, 1, 2\}]; \\ \mathbb{E}[3, 1] &\Leftarrow \mathbb{E}[1, 3]; \end{aligned} \quad (4.21)$$

Im Gegensatz zu einer händischen Umsetzung der ANS-Methode, bei der die Interpolation auch für die Ableitungen der Verzerrungen durchgeführt werden muss, kann bei der Programmierung mit ACEGEN mit den symbolisch vorliegenden interpolierten Verzerrungsverläufen gearbeitet werden, da alle erforderlichen Ableitungen bei Bedarf automatisch bestimmt werden. Es wird deutlich, dass die hergeleiteten Beziehungen (Gleichungen (3.57) und (3.57)) ohne eine Aufbereitung für die Programmierung direkt umgesetzt werden können, die theoretischen Zusammenhänge und die Implementierung also sehr nah beieinander liegen.

DS04-EAS_{xx}

Als Beispiel für die Anwendung der EAS-Methode für degenerierte 4-Knoten-Schalenelemente wurde die von SIMO & RIFAI in [117] vorgestellte Formulierung mit 4 EAS-Parametern zur

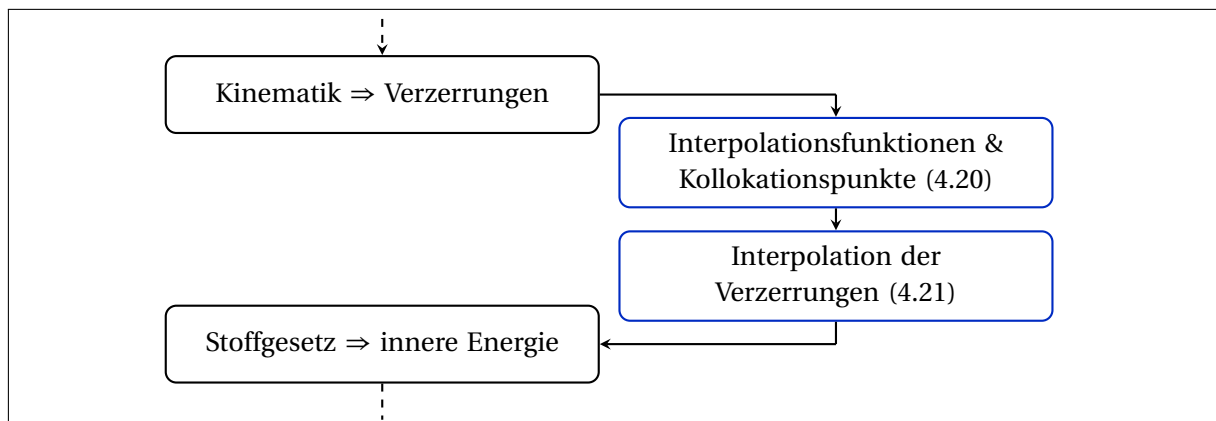


Abbildung 4.3.: Schema für ACEGEN-unterstützte Implementierung der *Methode der angenommenen Verzerrungen* (ANS) – Detaildarstellung aus Abbildung 4.2

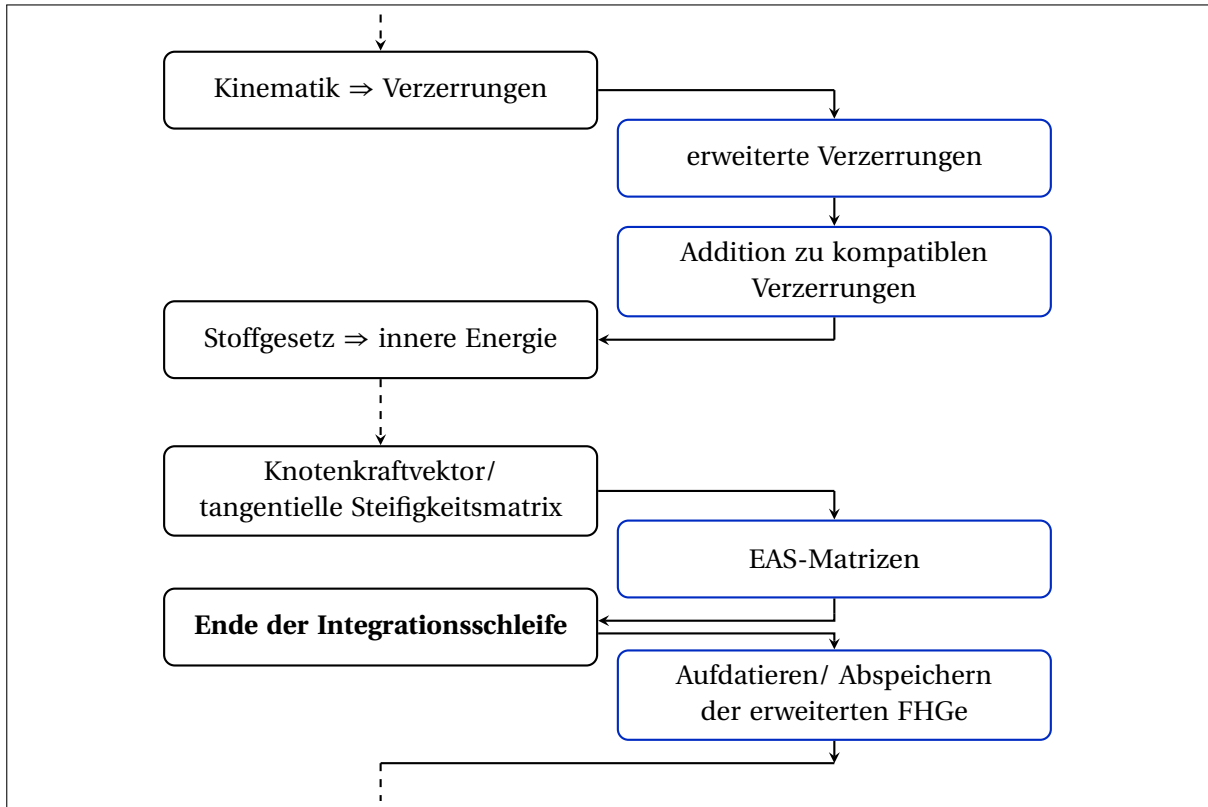


Abbildung 4.4.: Schema für die ACEGEN-unterstützte Implementierung der *Methode der Erweiterten Verzerrungen* (EAS) – Detaildarstellung aus Abbildung 4.2

Erweiterung der Normal- und Schubverzerrungen in der Schalenebene umgesetzt. Die erweiterten Verzerrungsansätze für das DS04-EAS04-Element lauten

$$\mathbf{N}_{\text{EAS}} = \begin{pmatrix} \xi & 0 & 0 & 0 \\ 0 & \eta & 0 & 0 \\ 0 & 0 & \xi & \eta \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (4.22)$$

Eine schematisierte Darstellung der erforderlichen Implementierungsschritte ist in Abbildung 4.4 skizziert. Nach der Bestimmung der kompatiblen Verzerrungen gemäß Gleichung (4.6) erfolgt zunächst die Aufstellung der erweiterten Verzerrungen:

$$\mathbf{N}_{\text{EAS}} = \begin{pmatrix} \xi & 0 & 0 & 0 \\ 0 & \eta & 0 & 0 \\ 0 & 0 & \xi & \eta \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; \quad \boldsymbol{\varepsilon} = \mathbf{N}_{\text{EAS}} \cdot \boldsymbol{\alpha};$$

$$\mathbf{E}_{\text{EAS}} = 1/2 \begin{pmatrix} 2\varepsilon_{[1,1]} & \varepsilon_{[1,2]} & \varepsilon_{[1,3]} \\ \varepsilon_{[1,2]} & 2\varepsilon_{[2,2]} & \varepsilon_{[2,3]} \\ \varepsilon_{[1,3]} & \varepsilon_{[2,3]} & 0 \end{pmatrix}; \quad (4.23)$$

wobei im eindimensionalen Feld $\boldsymbol{\alpha}$ die der Elementroutine übergebenen aktuellen Werte der erweiterten Freiheitsgrade angeordnet sind. Diese müssen nun nach Gleichung (3.110)

in das konvektive Elementkoordinatensystem transformiert werden. Eine Transformationsmatrix lässt sich mit der herkömmlichen (Gleichung (4.5)) und der am Elementmittelpunkt ausgewerteten JACOBI-Matrix bestimmen:

$$\begin{aligned} \mathbb{J}0 &\equiv \text{SMSReplaceAll}[\mathbb{J}, \{\xi \rightarrow 0, \eta \rightarrow 0, \zeta \rightarrow 0\}]; \\ \det\mathbb{J}0 &\equiv \text{SMSDet}[\mathbb{J}0]; \\ \mathbb{J}i0 &\equiv \text{SMSInverse}[\mathbb{J}0]; \end{aligned} \quad (4.24)$$

$$\mathbb{T} \equiv \mathbb{J} \cdot \mathbb{J}i0; \quad (4.25)$$

Die EAS-Verzerrungen werden transformiert und zu den kompatiblen Verzerrungen addiert

$$\mathbb{E} \equiv \mathbb{E} + \frac{\det\mathbb{J}0}{\det\mathbb{J}} \mathbb{T} \cdot \mathbb{E}EAS \cdot \mathbb{T}^T; \quad (4.26)$$

und es ergibt sich schließlich der erweiterte Verzerrungsverlauf, der dann zur Berechnung der inneren Energie (Gleichungen (4.7) und (4.8)) verwendet wird.

Zusätzlich zur Aufstellung des Vektors der inneren Knotenkräfte nach Gleichung (4.12) müssen nun die in den Gleichungen (3.118) und (3.119) eingeführten Matrizen und Vektoren durch Ableitung der inneren Energie nach den erweiterten Freiheitsgraden aufgestellt werden. Für eine Elementformulierung für explizite Zeitintegration werden gemäß Abschnitt 3.4.4 \mathbf{D}_{EAS} und \mathbf{p}_{EAS} durch

```
SMSDo [
  pas ≡ SMSD[ $\mathcal{W}$ ,  $\alpha$ , i] * detJ * wi;
  SMSEXPORt [pas, pas$$[i], "AddIn" -> True];
  SMSDo [
    das ≡ SMSD[pas,  $\alpha$ , j];
    SMSEXPORt [das, das$$[i, j], "AddIn" -> True];
    , {j, 1, Length[ $\alpha$ ]}];
  , {i, 1, Length[ $\alpha$ ]}];
```

(4.27)

bestimmt. Da für implizite Verfahren nach Abschnitt 3.4.4 zusätzlich die Matrix \mathbf{L}_{EAS} erforderlich ist, empfiehlt es sich hier, das Feld \mathbf{d} um die zusätzlichen Freiheitsgrade zu erweitern und sämtliche Matrizen über

```
SMSDo [
   $\Psi$  ≡ SMSD[ $\mathcal{W}$ ,  $\mathbf{d}$ , i] * detJ * wi;
  SMSEXPORt [- $\Psi$ , p$$[i], "AddIn" -> True];
  SMSDo [
    Kij ≡ SMSD[ $\Psi$ ,  $\mathbf{d}$ , j];
    SMSEXPORt [Kij, s$$[i, j], "AddIn" -> True];
    , {j, i, Length[ $\mathbf{d}$ ]}];
  , {i, 1, Length[ $\mathbf{d}$ ]}];
```

(4.28)

durch eine Schleife über alle – kompatible und erweiterte – Freiheitsgrade zu bestimmen. Die entspricht exakt der Vorgehensweise zur Aufstellung von Knotenkraftvektor und Steifigkeitsmatrix für reine Verschiebungselemente. Zur Einsparung von Programmschritten

startet hier die innere Schleife statt bei 1 bei i , sodass das Feld s außerhalb der Routine noch symmetrisch gemacht, also an der Hauptdiagonalen gespiegelt werden muss. Für eine Elementformulierung mit $nst0$ kompatiblen Freiheitsgraden und nas EAS-Erweiterungen beinhalten die zwei Felder p und s nun

$$p = \left(\begin{array}{c} \mathbf{f}^{\text{int}} [nst0] \\ \mathbf{p}_{\text{EAS}} [nas] \end{array} \right) \quad \text{und} \quad s = \left(\begin{array}{c|c} \mathbf{K}_{\text{EAS}} [nst0 \times nst0] & \mathbf{L}_{\text{EAS}}^T [nst0 \times nas] \\ \hline \mathbf{L}_{\text{EAS}} [nas \times nst0] & \mathbf{D}_{\text{EAS}} [nas \times nas] \end{array} \right), \quad (4.29)$$

die Werte in eckigen Klammern geben die Dimension der jeweiligen Untermatrix an.

Bei der Programmierung verschiedener EAS-Elemente zeigte sich, dass eine Implementierung der in den Abschnitten 3.4.4 und 3.4.4 beschriebenen Kondensation der erweiterten Freiheitsgrade mit ACEGEN nicht von Vorteil ist. Dieser Schritt wurde bei allen für die Arbeit umgesetzten Formulierungen mit der händischen Umsetzung eines GAUSS-Eliminationsverfahrens durchgeführt. Der Routine wurden die zuvor berechneten Matrizen und Vektoren übergeben und – da insbesondere Elemente für explizite Analysen umgesetzt wurden – nach Gleichung (3.125), die aktuellen Werte der erweiterten Freiheitsgrade berechnet. Wie bereits in Abschnitt 3.4.4 diskutiert, reichte in den praktischen Anwendungen auf Grund der kleinen Zeitschritte meist eine Iteration zur Bestimmung des Vektors α_n aus, der schließlich für jedes Element für den nächsten Zeitschritt abgespeichert wird.

Eine dem inkompatiblen Scheibenelement von WILSON, ET. AL. [126] entsprechende Formulierung DS04-EAS07 mit 7 EAS-Erweiterungen erhält man mit der Ansatzmatrix

$$\mathbf{N}_{\text{EAS}} = \begin{pmatrix} \xi & 0 & 0 & 0 & \xi\eta & 0 & 0 \\ 0 & \eta & 0 & 0 & 0 & \xi\eta & 0 \\ 0 & 0 & \xi & \eta & 0 & 0 & \xi\eta \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (4.30)$$

die jedoch wie bereits in verschiedenen Arbeiten gezeigt wurde, nicht zu wesentlichen Verbesserungen der Abbildungseigenschaften führt [8, 54, 117]. Das verwendete Implementierungskonzept erlaubt auch hier eine sehr einfache Umsetzung zusätzlicher Erweiterungen, da in den beschriebenen Schritten lediglich die Matrix \mathbf{N}_{EAS} sowie die Dimensionen einiger Felder angepasst werden müssen. Da in beiden Elementformulierungen lediglich Normal- und Membranschubverzerrungen erweitert werden, ist eine Kombination mit der ANS-Methode problemlos möglich. Durch Interpolation und Erweiterung der Verzerrungen gemäß der beschriebenen Schritte ergibt sich beispielsweise die Elementformulierung DS04-ANS-EAS04.

DS09-DISP

Eine degenerierte Schalenformulierung mit 9 Elementknoten erhält man durch die Interpolation von Geometrie und Verschiebungen mit quadratischen LAGRANGE-Ansätzen. Die

Ansatzfunktionen aus Gleichung (4.2) werden dann über

$$\begin{aligned}
 \xi_i &= \{-1, 1, 1, -1, 0, 1, 0, -1, 0\}; & \eta_i &= \{-1, -1, 1, 1, -1, 0, 1, 0, 0\}; \\
 \text{shp} &\models \text{Table}[\\
 &\quad (1/2 \xi \xi_i[[i]] (1 + \xi \xi_i[[i]]) + (1 - \xi^2) (1 - \xi_i[[i]]^2)) \\
 &\quad (1/2 \eta \eta_i[[i]] (1 + \eta \eta_i[[i]]) + (1 - \eta^2) (1 - \eta_i[[i]]^2)) \\
 &\quad , \{i, 1, 9\}];
 \end{aligned} \tag{4.31}$$

eingetragen, alle weiteren Programmschritte verlaufen völlig analog. Durch die höhere Interpolationsordnung können geometrische Versteifungseffekte deutlich reduziert werden, es tritt allerdings insbesondere bei gekrümmten Strukturen signifikant das sog. *Membran-Locking* auf, wie u. a. in [50, 54, 78] ausführlich diskutiert wird.

DS09-ANS

Analog zur DS04-ANS-Formulierung führt die Anwendung der ANS-Methode aus Abschnitt 3.4.2 auf das ursprünglich von BUCALEM & BATHE [29] vorgeschlagene MITC9-Element. Die Umsetzung von Auswertung und Interpolation der einzelnen Verzerrungskomponenten nach den Gleichungen (3.60) bis (3.65) erfolgt völlig analog wie bei DS04-ANS beschrieben. Es werden zusätzlich zum Querschub auch die Membranverzerrungen interpoliert. Beispielsweise für die Normalverzerrung $E_{\xi\xi}$ lautet die Eingabe der Interpolationsfunktionen und Kollokationspunkte

$$\begin{aligned}
 i11\xi &= \{1/2 (1 - \text{Sqrt}[3] \xi), 1/2 (1 + \text{Sqrt}[3] \xi)\}; \\
 i11\eta &= \{1/2 \text{Sqrt}[5/3] (\text{Sqrt}[5/3] \eta - 1), 1 - 5/3 \eta^2, 1/2 \text{Sqrt}[5/3] (\text{Sqrt}[5/3] \eta + 1)\}; \\
 k11\xi &= \{-\text{Sqrt}[1/3], \text{Sqrt}[1/3]\}; \\
 k11\eta &= \{-\text{Sqrt}[3/5], 0, \text{Sqrt}[3/5]\};
 \end{aligned} \tag{4.32}$$

und schließlich die Interpolationsvorschrift

$$\begin{aligned}
 \mathbb{E}[[1, 1]] &\models \\
 &\quad \text{Sum}[\\
 &\quad \quad \text{Sum}[\\
 &\quad \quad \quad i11\xi[[j]] i11\eta[[i]] \text{SMSReplaceAll}[\mathbb{E}[[1, 1]], \{\xi \rightarrow k11\xi[[j]], \eta \rightarrow k11\eta[[i]]\}] \\
 &\quad \quad \quad , \{j, 1, 2\}; \\
 &\quad \quad , \{i, 1, 3\};
 \end{aligned} \tag{4.33}$$

Im ACEGEN-Handbuch wird vor einer häufigen Anwendung des `SMSReplaceAll`-Kommandos gewarnt, da dadurch große Mengen Programmcode erzeugt werden. Tatsächlich unterscheiden sich die Elementroutinen mit ANS hinsichtlich ihrer Länge deutlich von der reinen Verschiebungsformulierung. Es ist denkbar, dass durch geschickte Anwendung der ACEGEN-Kommandos noch einige Programmschritte eingespart werden können, dennoch ist die ANS-Methode numerisch recht aufwändig, was sich insbesondere innerhalb eines expliziten Zeitintegrationsverfahrens bemerkbar macht.

Eine Anwendung der EAS-Methode auf degenerierte Schalenelemente mit quadratischen Ansatzfunktionen erfolgte im Rahmen der vorliegenden Arbeit nicht. Die zur Implementierung erforderlichen Schritte sind allerdings völlig analog zur beim DS04-EAS_{xx}-Element beschriebenen Vorgehensweise.

4.4.3. Solid-Shell-Elemente

VS08-DISP

Ebenso wie die degenerierte Schale mit linearen Ansätzen DS04-DISP, weist die entsprechende Volumenschale bei vollständiger Integration mit $2 \times 2 \times 2$ GAUSS-Punkten verschiedene Versteifungseffekte auf und eignet sich wenig für die praktische Anwendung. Die Implementierung dieser Formulierung erfolge sowohl durch die in Abschnitt 4.3.1 beschriebenen Schritte mit ACEGEN, als auch durch händische Programmierung der einzelnen Unterrouninen. Zum einen wird das VS08-DISP Element in den Beispielen in Kapitel 5 zur Diskussion der Wirksamkeit der verschiedenen Verfahren zur Beseitigung von Versteifungseffekten verwendet, zum anderen erlaubt der Vergleich mit der händischen Implementierung eine Bewertung der Effizienz der automatisch generierten Programmrouninen.

VS08-ANS

Neben der bereits bei der degenerierten Schalenformulierung DS04-ANS beschriebenen Interpolation der Querschubverzerrungen, enthält das VS08-ANS Element zusätzlich die Modifikation der Normalverzerrungen in Dickenrichtung $E_{\zeta\zeta}$. Diese werden nach Gleichung (3.59) an den auf die Mittelebene projizierten Elementknoten ausgewertet und mit den linearen Ansatzfunktionen interpoliert. Für die symbolische Implementierung bedeutet dies

$$\begin{aligned} \mathbb{E}[[3, 3]] &= \\ \text{Sum} [& \\ \text{shp}[[i]] \text{SMSReplaceAll} [\mathbb{E}[[3, 3]], \{\xi \rightarrow \xi_i[[i]], \eta \rightarrow \eta_i[[i]], \zeta \rightarrow 0\} & \\ , \{i, 1, 4\}; & \end{aligned} \quad (4.34)$$

mit den Ansatzfunktionen `shp` und der Lage der Elementknoten ξ_i und η_i aus Gleichung (4.2). Wie zuvor beschrieben werden die modifizierten Verzerrungsverläufe der jeweiligen Materialformulierung übergeben, wobei eine Anwendung dieser Methode für Plastizität mit großen Deformationen problematisch ist. Wie bereits in Abschnitt 3.4.2 erläutert, wird der modifizierte Deformationsgradient benötigt, dessen Rückrechnung aus den modifizierten Verzerrungen sehr aufwändig ist, da Eigenwerte und Eigenvektoren der modifizierten und unmodifizierten Deformationstensoren erforderlich sind. Die Programmschritte zur Bestimmung der Eigenwerte erhält man über die Funktion `SMSEigenvalues`, die Aufstellung der Eigenwerte erfolgte über den in Anhang A.4 skizzierten Algorithmus. Beide Prozeduren sind sehr aufwändig und führen zu einer großen Menge Programmcode mit verschachtelten If-Operationen, um die einzelnen Fälle für gleiche und verschiedene Eigenwerte abzufragen. Die resultierende Elementformulierung ist wenig effizient, hier sind weitere Überlegungen hinsichtlich der Umsetzung der Methode erforderlich.

VS08-XDIS

Ein Beispiel für eine Implementierung bei der die Vorteile des symbolischen Programmierkonzeptes besonders deutlich werden, stellt die Umsetzung der in Abschnitt 3.4.3 diskutierten Methode zur Reproduktion der ANS-Verzerrungsverläufe durch Modifikation der Verschiebungsinterpolation dar. Es können direkt Veränderungen am Verschiebungsverlauf

vorgenommen werden, alle weiteren Schritte werden dann automatisch mit der modifizierten Interpolation durchgeführt. Eine händische Umsetzung der Methode wäre sehr aufwändig, da vorab die Auswirkungen der veränderten Verschiebungsapproximation auf die abgeleiteten Größen (Verzerrungen und Spannungen) diskutiert und umgesetzt werden müssten.

Für die Umsetzung der erweiterten Verschiebungen $w_{\text{add}}^{\xi\zeta}$ und $w_{\text{add}}^{\eta\zeta}$ aus den Gleichungen (3.95) und (3.98) werden die am aktuellen Integrationspunkt ausgewerteten Größen der JACOBI-Matrix \mathbb{J} und des lokalen Verschiebungsgradienten $\Delta \mathbf{u}$ aus den Gleichungen (3.17) und (3.26) benötigt. Beide Matrizen sind im symbolischen Kontext im Allgemeinen veränderlich in ξ , η und ζ . Um im Verlauf der Prozedur eine Ableitung nach den natürlichen Koordinaten zu verhindern, werden die nach den Gleichungen (4.5) und (4.6) aufgestellten Matrizen durch

$$\begin{aligned} \mathbb{J}\mathbf{f} &\equiv \text{SMSReplaceAll}[\mathbb{J}, \{\xi \rightarrow \xi\mathbf{f}, \eta \rightarrow \eta\mathbf{f}, \zeta \rightarrow \zeta\mathbf{f}\}]; \\ \Delta\mathbf{u}\mathbf{f} &\equiv \text{SMSReplaceAll}[\Delta\mathbf{u}, \{\xi \rightarrow \xi\mathbf{f}, \eta \rightarrow \eta\mathbf{f}, \zeta \rightarrow \zeta\mathbf{f}\}]; \end{aligned} \quad (4.35)$$

ausgewertet, um als konstante Werte am aktuellen Integrationspunkt behandelt zu werden. Die Variablen $\xi\mathbf{f}$, $\eta\mathbf{f}$ und $\zeta\mathbf{f}$ werden hierbei zwar genau wie ξ , η und ζ bei der automatischen Codeerzeugung durch die Koordinaten der Integrationspunkte ersetzt, jedoch erfolgt nach den neu eingeführten Variablen im weiteren Verlauf keine symbolische Ableitung. Die erforderlichen LEGENDRE-Polynome aus Gleichung (3.92) werden allgemein über eine selbst definierte MATHEMATICA-Funktion

$$\text{pi}[\mathbf{x}_-, \mathbf{k}_-] := 1/(2^{\mathbf{k}} \mathbf{k}!) \quad \text{D}[(\mathbf{x}^2 - 1)^{\mathbf{k}}, \{\mathbf{x}, \mathbf{k}\}]; \quad (4.36)$$

realisiert, sodass beispielsweise das Polynom 2. Grades $P_2(\xi)$ über den Befehl `pi[2, xi]` abgerufen werden kann. Mit den eingeführten Größen lassen sich die erweiterten Verschiebungen nach den Gleichungen (3.95) und (3.98) eingeben und in den Variablen `wadd13` und `wadd23` abspeichern. Die Erweiterung der Verschiebungsinterpolation erfolgt dann durch

$$\mathbf{u}[\mathbf{3}] \rightarrow \mathbf{u}[\mathbf{3}] + w_{\text{add13}} + w_{\text{add23}}; \quad (4.37)$$

und die Verzerrungen sowie alle weiteren Größen werden schließlich mit dieser modifizierten Verschiebungsinterpolation ermittelt.

Ausgehend vom VS08-XDIS-Element lassen sich durch die Verwendung der symbolischen Programmierumgebung sehr einfach Varianten umsetzen, da Anpassungen an der Verschiebungsinterpolation direkt vorgenommen werden können. Dies ermöglicht ein effizientes Testen verschiedener Elementformulierungen durch Änderungen im Programmcode, die direkt der theoretischen Herleitung entsprechen. Bei händischer Programmierung dagegen würde eine kleine Änderung in der Formulierung zu umfangreichen Modifikationen an verschiedenen Stellen des Programmcodes führen.

VS08-EAS_{xx}

Die Umsetzung der in Abschnitt 3.4.4 beschriebenen *Methode der erweiterten Verzerrungen* für die Volumenschalenelemente erfolge analog zur bei den DS04-EAS_{xx}-Elementen angegebenen Vorgehensweise. In Kombination mit der ANS-Methode wurde EAS eingesetzt, um den Einfluss der volumetrischen Dickenversteifung – ein materieller Locking-Effekt, der durch

ANS nicht behoben werden kann – zu behandeln. Es wurden zwei Elementformulierungen VS08-ANS-EAS01 und VS08-ANS-EAS03 implementiert, bei denen die Normalverzerrungen in Dickenrichtung mit Hilfe der hier sechszeiligen Ansatzmatrizen

$$\mathbf{N}_{\text{EAS}}^{01} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \zeta \\ 0 \\ 0 \end{pmatrix}, \quad \text{bzw.} \quad \mathbf{N}_{\text{EAS}}^{03} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \zeta & \xi\zeta & \eta\zeta \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (4.38)$$

erweitert wurden. Zusätzliche Erweiterungen von $E_{\zeta\zeta}$ können durch entsprechende Anpassung der Matrix \mathbf{N}_{EAS} in der symbolischen Programmierumgebung realisiert werden, führten aber nicht zu erkennbaren Verbesserungen der Abbildungseigenschaften. Alle erforderlichen Implementierungsschritte können der Beschreibung bei den DS04-EAS_{xx}-Formulierungen entnommen werden.

Ein Volumenelement, bei welchem ausschließlich durch die Einführung von EAS-Parametern eine Beseitigung der Biegeversteifung zumindest für Quader und Parallelepipede erreicht wird [45], wurde von TAYLOR, ET. AL. [123] vorgeschlagen und bereits als Beispiel in Abschnitt 3.4.5 eingeführt. Die Erweiterungen der aus dem Schrifttum als Q1E9 bekannten Elementformulierung wurden für das VS08-EAS09 in der Matrix

$$\mathbf{N}_{\text{EAS}}^{09} = \begin{pmatrix} \xi & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \eta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \xi & \eta & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \zeta & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \eta & \zeta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \xi & \zeta & 0 \end{pmatrix}. \quad (4.39)$$

angeordnet. Zusätzliche Erweiterungen der Normalverzerrungen sollen das Verhalten hinsichtlich volumetrischer Versteifungseffekte verbessern (vgl. [45]) und führen auf die 12-Parameter Formulierung VS08-EAS12 mit der Erweiterungsmatrix

$$\mathbf{N}_{\text{EAS}}^{12} = \begin{pmatrix} \xi & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \eta\zeta & 0 & 0 \\ 0 & \eta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \xi\zeta & 0 \\ 0 & 0 & \xi & \eta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \zeta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \xi\eta \\ 0 & 0 & 0 & 0 & 0 & \eta & \zeta & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \xi & \zeta & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (4.40)$$

In Abschnitt 3.4.4 wurde für die Anwendung von Materialformulierungen mit finiter Plastizität die Notwendigkeit eines erweiterten Deformationsgradienten $\tilde{\mathbf{F}}$ diskutiert. Die in Gleichung (3.132) beschriebene Vorgehensweise erfordert die Anordnung der EAS-Erweiterungen in einer Matrix $\mathbf{F}_{\text{EAS},0}$. Einer der aus Gleichung (4.39) entsprechenden Formulierungen mit 9 Erweiterungen ergibt sich über

$$\mathbf{F}_{\text{EAS}} = \begin{pmatrix} \alpha[1]\xi & \alpha[4]\eta & \alpha[7]\zeta \\ \alpha[2]\xi & \alpha[5]\eta & \alpha[8]\zeta \\ \alpha[3]\xi & \alpha[6]\eta & \alpha[9]\zeta \end{pmatrix}; \quad (4.41)$$

Von SIMO, ET. AL. [114] wurde eine Formulierung mit 12 Parametern vorgestellt, deren Erweiterungsmatrix über

$$\begin{aligned} \mathbb{I} & \mathbb{F} \text{ IdentityMatrix}[3]; \\ \mathbb{F}_{\text{EAS}} & \mathbb{F} \begin{pmatrix} \alpha[1] \xi & \alpha[4] \eta & \alpha[7] \zeta \\ \alpha[2] \xi & \alpha[5] \eta & \alpha[8] \zeta \\ \alpha[3] \xi & \alpha[6] \eta & \alpha[9] \zeta \end{pmatrix} + \alpha[10] \xi \eta \mathbb{I} + \alpha[11] \xi \zeta \mathbb{I} + \alpha[12] \eta \zeta \mathbb{I}; \end{aligned} \quad (4.42)$$

implementiert werden kann. Der die EAS-Erweiterungen beinhaltende Deformationsgradient wird schließlich analog zu Gleichung (4.26) durch

$$\mathbb{F} \mathbb{F} + \frac{\det \mathbb{J}_0}{\det \mathbb{J}} \mathbb{F}_0 \cdot (\mathbb{J}_i^0 \cdot \mathbb{F}_{\text{EAS}} \cdot \mathbb{J}_i^0)^T; \quad (4.43)$$

bestimmt und kann dann einer beliebigen Materialformulierung übergeben werden. Eine alternative Formulierung die auf einem Vorschlag von WRIGGERS & KORELC [75, 131] unter dem Namen CG9 basiert und zu einer Steigerung der Robustheit führen soll, wurde unter der Bezeichnung VS08-EAS09T implementiert. Es wird hier die Transponierte der Erweiterungsmatrix $\mathbb{F}_{\text{EAS},0}$ des VS08-EAS09 verwendet, also

$$\mathbb{F}_{\text{EAS},0}^{09T} = (\mathbb{F}_{\text{EAS},0}^{09})^T. \quad (4.44)$$

Die Umsetzung der reinen EAS-Elemente erfolgte insbesondere zum Vergleich mit den VS08-IM_{xx}-Elementen, die im folgenden Abschnitt beschrieben werden. Die Einführung weiterer EAS-Parameter ist prinzipiell völlig analog möglich, macht aber insbesondere im Kontext expliziter Zeitintegrationsverfahren wie bereits diskutiert nur wenig Sinn, da der Aufwand für die Invertierung der Matrix \mathbf{D}_{EAS} erheblich ansteigt.

VS08-IM_{xx}

Die in Abschnitt 3.4.5 beschriebene *Methode der inkompatiblen Moden* basiert auf der Erweiterung der Verschiebungsinterpolation durch Einführung inkompatibler Freiheitsgrade mit dem Ziel, die erweiterten Verzerrungsverläufe der EAS-Elemente zu reproduzieren. Der wesentliche Unterschied besteht im lokalen Auskondensieren der erweiterten Freiheitsgrade, die bei der EAS-Methode einen erheblichen Einfluss auf die numerische Effizienz der Elementformulierung hat. Bei den IM-Elementen werden die inkompatiblen Freiheitsgrade wie kompatible behandelt und in die globalen Gleichungen einbezogen. Dies erfordert wie bereits in Abschnitt 3.4.5 diskutiert die Formulierung einer Massenmatrix für die erweiterten Freiheitsgrade.

Der Elementroutine zur Aufstellung des inneren Knotenkraftvektors werden sowohl aktuellen Werte der kompatiblen als auch die der erweiterten Freiheitsgrade übergeben, letztere werden analog zu den EAS-Elementen in einem Vektor α abgelegt. Es folgt eine Erweiterung des Feldes aller Freiheitsgrade \mathfrak{d} aus Gleichung (4.11) durch die inkompatiblen Freiheitsgrade gemäß

$$\mathfrak{d} \mathbb{F} \text{ Flatten}[\{\text{ulu}^T, \text{ulo}^T, \alpha\}]; \quad (4.45)$$

Geometrie und Verschiebungen werden zunächst wie bei der VS08-DISP-Formulierung interpoliert, die Ansätze für die Interpolation der inkompatiblen Freiheitsgrade werden

vorab über die Gleichungen (3.134), (3.135) und (3.136) berechnet und in einer Matrix \mathbb{N}_{add} nach Gleichung (3.139) abgespeichert. Die eigentliche Interpolation erfolgt dann durch

$$\mathbb{u}_{\text{add}} \mathbb{F} \mathbb{N}_{\text{add}} \cdot \alpha; \quad (4.46)$$

und führt dann über

$$\begin{aligned} \Delta \mathbb{u}_{\text{add}} &\mathbb{F} \text{SMSD}[\mathbb{u}_{\text{add}}, \{\xi, \eta, \zeta\}]^T; \\ \mathbb{E}_{\text{IM}} &\mathbb{F} 1/2 (\Delta \mathbb{u}_{\text{add}}^T + \Delta \mathbb{u}_{\text{add}}); \end{aligned} \quad (4.47)$$

auf die der zugeordneten EAS-Formulierung entsprechenden erweiterten Verzerrungen. Diese können nun wie in Gleichung (4.26) in die lokalen Elementkoordinaten transformiert und zu den kompatiblen Verzerrungen addiert werden. Am Ende der Routine wird wie in Gleichung (4.12) der Vektor der inneren Kräfte aufgestellt, mit dem Unterschied, dass die Freiheitsgrade d zuvor in Gleichung (4.45) um die inkompatiblen Freiheitsgrade erweitert wurden. Die Dimension von f^{int} beispielsweise für das VS08-IM09-Element beträgt wegen 24 kompatibler und 9 inkompatibler Freiheitsgrade also 33.

VS18-DISP

Die Verwendung der quadratischen LAGRANGE-Ansätze aus Gleichung (4.31) zur Interpolation von Geometrie und Verschiebungen führt auf die reine Verschiebungsformulierung des Volumenschalenelementes mit 18 Elementknoten. Wie bei der degenerierten Schalenformulierung DS09-DISP können auch hier im Vergleich zum linearen VS08-DISP-Element geometrische Versteifungseffekte deutlich reduziert werden. Die numerische Integration erfolgte mit $3 \times 3 \times 2$ GAUSS-Punkten, was zu einem höheren numerischen Aufwand für die Aufstellung der inneren Knotenkräfte führt. Ebenso wie bei der degenerierten Schalenformulierung treten zusätzliche Membranversteifungen auf, die durch entsprechende Modifikation der Verzerrungsverläufe mit der ANS-Methode behandelt werden können.

VS18-ANS

Die VS18-ANS-Formulierung stellt das Pendant zum DS09-ANS-Element dar, bei dem die Querschub- und Membranverzerrungen mit der ANS-Methode behandelt werden. Analog zum Vorgehen bei VS08-ANS werden zusätzlich die Normalverzerrungen in Dickenrichtung nach Gleichung (3.66) behandelt. Die entsprechende Eingabe in ACEGEN erfolgt über

$$\begin{aligned} \mathbb{E}[[3, 3]] &\mathbb{F} \\ \text{Sum}[& \\ \text{shp}[[i]] &\text{SMSReplaceAll}[\mathbb{E}[[3, 3]], \{\xi \rightarrow \xi i[[i]], \eta \rightarrow \eta i[[i]], \zeta \rightarrow 0\} \\ & , \{i, 1, 9\}; \end{aligned} \quad (4.48)$$

mit den quadratischen LAGRANGE-Ansätze aus Gleichung (4.31).

VS18-EASxx

Im Zusammenhang mit quadratischen Solid-Shell-Elementen wurde die *Methode der erweiterten Verzerrungen* im Rahmen der Arbeit lediglich zur Behandlung der volumetrischen

Dickenversteifung durch Erweiterung der Verzerrungskomponente $E_{\zeta\zeta}$ eingesetzt. Die zugehörige Ansatzmatrix für die acht erweiterten Freiheitsgrade lautet

$$\mathbf{N}_{\text{EAS}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \zeta & \xi\zeta & \eta\zeta & \xi\eta\zeta & \xi^2\zeta & \eta^2\zeta & \xi^2\eta\zeta & \xi\eta\zeta^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (4.49)$$

die Implementierung erfolgte völlig analog zur bei den VS08-EAS_{xx}-Elementen beschriebenen Vorgehensweise. Zusätzliche EAS-Erweiterungen können prinzipiell ebenso formuliert und implementiert werden, die Untersuchungen in dieser Arbeit beschränken sich jedoch auf das VS18-EAS08-Element.

VS16-DISP

Bei der Anwendung quadratischer *Serendipity*-Funktionen aus Gleichung (3.11) für die Interpolation von Geometrie und Verschiebungen ergibt sich eine 16-Knoten-Formulierung, die hier der Vollständigkeit halber erwähnt sei. Es wurde ausschließlich die reine Verschiebungsformulierung umgesetzt, also keine Modifikationen der Verschiebungs- oder Verzerrungsverläufe vorgenommen. Durch das Fehlen des Mittelknotens besitzt diese Formulierung 6 Freiheitsgrade weniger als das VS18-DISP-Element, weshalb eine geringere Rechenzeit bei vergleichbaren Anwendungen zu erwarten ist. Darüber hinaus kann bei *Serendipity*-Elementen in praktischen Anwendungen eine reduzierte Integrationsregel – also 2×2 statt 3×3 GAUSS-Punkte in der Schalenebene – ohne Stabilisierung angewendet werden. Durch die Unterintegration treten lediglich 2 zusätzliche kinematische Eigenformen auf, die jedoch innerhalb einer Diskretisierung mit mehreren Elementen nicht *'kommunizierbar'*, also bei praktischen Anwendungen irrelevant sind (siehe z. B. [58]).

4.4.4. Kontaktelemente

Die für die vorliegende Arbeit implementierten STAS-Kontaktelemente basieren wie bereits in Abschnitt 3.5 angegeben auf der Arbeit von KONYUKHOV [64] und wurden für die symbolische Programmierung mit ACEGEN aufbereitet. Für die Beschreibung der einzelnen Formulierungen – die sich lediglich durch die Art der analytisch definierten Kontaktfläche unterscheiden – wird auf die bereits erwähnte Unterteilung in Segmente bei der numerischen Integration nicht detailliert eingegangen. In Kombination mit verschiedenen Integrationsordnungen und -regeln erlaubt sie die Verwendung von praktisch beliebig vielen Integrationspunkten pro Element, was besonders bei stark gekrümmten Beispielen und grober Diskretisierung von Bedeutung ist. Für die dafür erforderliche Transformation der Koordinaten der Integrationspunkte wird auf die Ausführungen in [64] verwiesen.

Für die Implementierung wird innerhalb der numerischen Integrationsschleife zunächst eine Interpolation der Geometrie der *'slave'*-Fläche in der Momentankonfiguration vorgenommen. Mit der unverformten Geometrie \mathbf{x}_1 und den aktuellen Knotenverschiebungen \mathbf{u}_1 ,

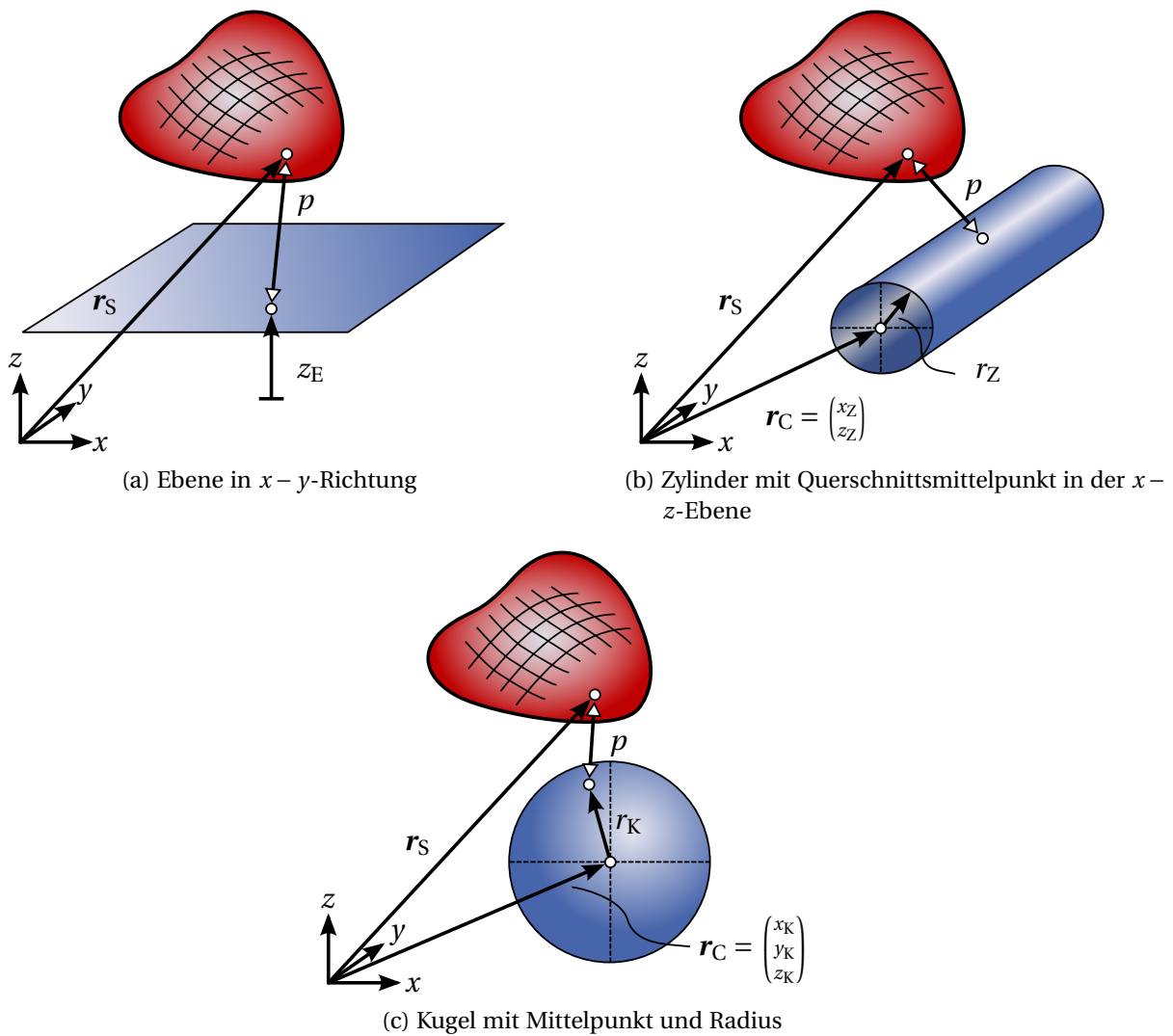


Abbildung 4.5.: Implementierte analytische Kontaktkörper

die der Elementroutine übergeben werden, lautet diese

$$\begin{aligned} \mathbf{x}_d &\Leftarrow \mathbf{x}_1 + \mathbf{u}_1; \\ \mathbf{x}_{s1} &\Leftarrow \mathbf{shp} \cdot \mathbf{x}_d^T; \end{aligned} \quad (4.50)$$

wobei \mathbf{shp} hier die Ansatzfunktionen an den transformierten Integrationspunkten beinhaltet. In \mathbf{x}_{s1} sind also die Komponenten des Ortsvektors der 'slave'-Fläche in der Momentankonfiguration \mathbf{r}_S abgelegt. Es folgt die Bestimmung der Penetration p , die durch die jeweilige analytisch beschriebene 'master'-Fläche charakterisiert ist. Für die Arbeit wurden drei verschiedene Elementtypen implementiert, deren geometrische Beschreibung in Abbildung 4.5 schematisch skizziert ist:

- Ebene z. B. in $x - y$ -Richtung, definiert durch z -Koordinate z_e

$$\mathbf{ptr} \Leftarrow \mathbf{x}_{s1}[[3]] - z_e; \quad (4.51)$$

- Zylinder z. B. mit Querschnittsmittelpunkt in der $x - z$ -Ebene definiert durch xz und zz sowie Radius rz

$$\text{ptr} \models \text{SMSSqrt}[(\text{xs1}[[1]] - \text{xz})^2 + (\text{xs1}[[3]] - \text{zz})^2] - \text{rz}; \quad (4.52)$$

- Kugel mit Mittelpunktskoordinaten xk , yk und zk sowie Radius rk

$$\text{ptr} \models \text{SMSSqrt}[(\text{xs1}[[1]] - \text{xk})^2 + (\text{xs1}[[2]] - \text{yk})^2 + (\text{xs1}[[3]] - \text{zk})^2] - \text{rk}; \quad (4.53)$$

Mit der Abfrage

$$\text{SMSIf}[\text{ptr} < 0]; \quad (4.54)$$

wird geprüft ob die ‘*slave*’- in die ‘*master*’-Fläche eindringt. Ist dies nicht der Fall, ist die Elementroutine beendet und es wird kein Beitrag zu den inneren Knotenkräften berechnet. Bei Kontakt wird der Beitrag zum inneren Knotenkraftvektor nach Gleichung (3.151) ermittelt, wobei sich der Normaleneinheitsvektor auf die ‘*slave*’-Fläche \mathbf{n}_S aus dem Kreuzprodukt der zwei Tangentenvektoren an die verformte Fläche durch

$$\begin{aligned} \text{rvec} &\models \text{shp.xd}^T; \\ \text{ro} &\models \text{SMSD}[\text{rvec}, \{\xi, \eta\}]^T; \\ \text{rxr} &\models \text{ro}[[1]] \times \text{ro}[[2]]; \\ \text{norm} &\models \text{SMSSqrt}[\text{rxr}[[1]]^2 + \text{rxr}[[2]]^2 + \text{rxr}[[3]]^2]; \\ \text{ns} &\models \text{rxr}/\text{norm}; \end{aligned} \quad (4.55)$$

ergibt. Die numerische Integration des Beitrages zum inneren Knotenkraftvektor erfolgt bei einer $\text{mdiv} \times \text{ndiv}$ Unterteilung des Kontaktelementes über

$$\begin{aligned} \text{res} &\models -\text{ptr} \varepsilon \mathbb{N} \cdot \text{ns} \, \mathbf{w}_\xi \, \mathbf{w}_\eta \, \text{norm} / (\text{mdiv} \, \text{ndiv}); \\ \text{SMSExport} &[\text{res}, \text{p}\$, "AddIn" \rightarrow \text{True}]; \end{aligned} \quad (4.56)$$

mit dem Penalty-Parameter $\varepsilon \mathbb{N}$, der Matrix der Ansatzfunktionen \mathbb{N} und den Wichtungsfaktoren \mathbf{w}_ξ und \mathbf{w}_η . Die Penetrationsabfrage wird schließlich mit

$$\text{SMSEndIf}[]; \quad (4.57)$$

abgeschlossen. Für eine detaillierte Herleitung dieser Integrationsvorschrift sowie weiterer Kontaktformulierungen wird auf [64] verwiesen.

4.4.5. Tabellarische Zusammenstellung der verwendeten Elementformulierungen

Zur übersichtlichen Darstellung werden die in der Arbeit verwendeten Schalen- und Volumenschalenformulierungen im Folgenden mit einer kurzen Beschreibung in zwei Tabellen aufgelistet:

degenerierte Schalenelemente

Bezeichnung	Kurzbeschreibung
DS04-DISP	<ul style="list-style-type: none"> • 4-Knoten-Schale mit linearen LAGRANGE-Ansätzen • reine Verschiebungsformulierung ohne Modifikationen
DS04-ANS	<ul style="list-style-type: none"> • 4-Knoten-Schale mit linearen LAGRANGE-Ansätzen • ANS für $E_{\xi\xi}$ und $E_{\eta\xi}$
DS04-EAS04	<ul style="list-style-type: none"> • 4-Knoten-Schale mit linearen LAGRANGE-Ansätzen • 4-Parameter EAS nach Gleichung (4.22)
DS04-EAS07	<ul style="list-style-type: none"> • 4-Knoten-Schale mit linearen LAGRANGE-Ansätzen • 7-Parameter EAS nach Gleichung (4.30)
DS04-ANS-EAS04	<ul style="list-style-type: none"> • 4-Knoten-Schale mit linearen LAGRANGE-Ansätzen • ANS für $E_{\xi\xi}$ und $E_{\eta\xi}$ • 4-Parameter EAS nach Gleichung (4.22)
DS09-DISP	<ul style="list-style-type: none"> • 9-Knoten-Schale mit quadratischen LAGRANGE-Ansätzen • reine Verschiebungsformulierung ohne Modifikationen
DS09-ANS	<ul style="list-style-type: none"> • 9-Knoten-Schale mit quadratischen LAGRANGE-Ansätzen • ANS für $E_{\xi\xi}$, $E_{\eta\eta}$, $E_{\xi\eta}$, $E_{\xi\xi}$ und $E_{\eta\xi}$

Solid-Shell-Elemente

Bezeichnung	Kurzbeschreibung
VS08-DISP	<ul style="list-style-type: none"> • 8-Knoten-Solid-Shell mit linearen LAGRANGE-Ansätzen • reine Verschiebungsformulierung ohne Modifikationen
VS08-ANS	<ul style="list-style-type: none"> • 8-Knoten-Solid-Shell mit linearen LAGRANGE-Ansätzen • ANS für $E_{\xi\xi}$, $E_{\eta\xi}$ und $E_{\zeta\xi}$
VS08-XDIS	<ul style="list-style-type: none"> • 8-Knoten-Solid-Shell mit linearen LAGRANGE-Ansätzen • modifizierte Verschiebungen nach Gleichung (3.95) und (3.98)
VS08-ANS-EAS01	<ul style="list-style-type: none"> • 8-Knoten-Solid-Shell mit linearen LAGRANGE-Ansätzen • ANS für $E_{\xi\xi}$, $E_{\eta\xi}$ und $E_{\zeta\xi}$ • 1-Parameter EAS nach Gleichung (4.38)
VS08-ANS-EAS03	<ul style="list-style-type: none"> • 8-Knoten-Solid-Shell mit linearen LAGRANGE-Ansätzen • ANS für $E_{\xi\xi}$, $E_{\eta\xi}$ und $E_{\zeta\xi}$ • 3-Parameter EAS nach Gleichung (4.38)
VS08-EAS09	<ul style="list-style-type: none"> • 8-Knoten-Solid-Shell mit linearen LAGRANGE-Ansätzen • 9-Parameter EAS nach Gleichung (4.39) bzw. (4.41)
VS08-EAS09T	<ul style="list-style-type: none"> • 8-Knoten-Solid-Shell mit linearen LAGRANGE-Ansätzen • 9-Parameter EAS nach Gleichung (4.44)
VS08-EAS12	<ul style="list-style-type: none"> • 8-Knoten-Solid-Shell mit linearen LAGRANGE-Ansätzen • 12-Parameter EAS nach Gleichung (4.40) bzw. (4.42)

Bezeichnung	Kurzbeschreibung
VS08-IM09	<ul style="list-style-type: none"> • 8-Knoten-Solid-Shell mit linearen LAGRANGE-Ansätzen • 9 inkompatible Moden \equiv analog zu VS08-EAS09
VS08-IM12	<ul style="list-style-type: none"> • 8-Knoten-Solid-Shell mit linearen LAGRANGE-Ansätzen • 12 inkompatible Moden \equiv analog zu VS08-EAS12
VS08-ANS-IM04	<ul style="list-style-type: none"> • 8-Knoten-Solid-Shell mit linearen LAGRANGE-Ansätzen • ANS für $E_{\xi\xi}$, $E_{\eta\xi}$ und $E_{\zeta\xi}$ • 4 inkompatible Moden \equiv analog zu VS08-ANS-EAS04
VS18-DISP	<ul style="list-style-type: none"> • 18-Knoten-Solid-Shell mit quadratischen LAGRANGE-Ansätzen • reine Verschiebungsformulierung ohne Modifikationen
VS18-ANS	<ul style="list-style-type: none"> • 18-Knoten-Solid-Shell mit quadratischen LAGRANGE-Ansätzen • ANS für $E_{\xi\xi}$, $E_{\eta\eta}$, $E_{\xi\eta}$, $E_{\zeta\xi}$, $E_{\xi\zeta}$ und $E_{\eta\zeta}$
VS18-ANS-EAS08	<ul style="list-style-type: none"> • 18-Knoten-Solid-Shell mit quadratischen LAGRANGE-Ansätzen • ANS für $E_{\xi\xi}$, $E_{\eta\eta}$, $E_{\xi\eta}$, $E_{\zeta\xi}$, $E_{\xi\zeta}$ und $E_{\eta\zeta}$ • 8-Parameter EAS nach Gleichung (4.49)
VS16-DISP	<ul style="list-style-type: none"> • 16-Knoten-Solid-Shell mit quadratischen Serendipity-Ansätzen • reine Verschiebungsformulierung ohne Modifikationen
VS16-DISPr	<ul style="list-style-type: none"> • 16-Knoten-Solid-Shell mit quadratischen Serendipity-Ansätzen • vollständig reduzierte Integration mit $2 \times 2 \times 2$ GAUSS-Punkten

5. Numerische Beispiele

In diesem letzten Abschnitt der Arbeit werden die zuvor entwickelten und beschriebenen Elementformulierungen an numerischen Beispielen bewertet. Diese sind zum einen gewählt, um die Einflüsse bestimmter Versteifungseffekte zu zeigen. Zum anderen spielt die Effizienz der einzelnen Methoden im Sinne der zweiten Definition aus Kapitel 4 eine zentrale Rolle, sodass die jeweils erforderliche Rechenzeiten an verschiedenen Beispielen verglichen werden. Es wurde darauf geachtet, innerhalb eines Beispiels stets identische Bedingungen hinsichtlich der Computerhardware zu gewährleisten, sodass die angegebenen Zeiten einen objektiven Vergleich zwischen den verschiedenen Implementierungen erlauben. Sämtliche Simulationen erfolgten auf Standard-PCs; auf eine aufwändigere Bewertung der Methoden – etwa durch ‘Zählen’ der umgesetzten Programmschritte – wurde im Rahmen der Arbeit verzichtet, da die Simulationszeiten, die über das UNIX-Kommando ‘time’ ausgegeben wurden, einen sehr anschaulichen Vergleich ermöglichten.

5.1. Kragarm unter Einzellast

5.1.1. Problembeschreibung und Diskretisierung

Betrachtet wird der in Abbildung 5.1 skizzierte Kragarm der Länge $\ell = 200$ mm, der Breite $t = 20$ mm und der Dicke $b = 2$ mm. Am freien Ende ist eine Einzellast aufgebracht, die energetisch äquivalent über alle Elementknoten bei $x = \ell$ verteilt ist. Die Kraft $F(t)$ wird über eine Lastkurve aufgebracht; je nach deren Verlauf ergibt sich eine quasi-statische oder eine dynamische Belastung. Für lineare Analysen wird ein ST. VENANT-KIRCHHOFF-Material angewendet, Simulationen mit großen Deformationen erfolgen mit einer NEO-HOOKE-Formulierung. Für beide Fälle ist der E-Modul zu $E = 100 \cdot 10^3$ N/mm² und die Rohdichte zu $\rho = 10 \cdot 10^3$ kg/m³ gewählt. Soweit nicht anders angegeben, wird die Querkontraktionszahl mit $\nu = 0.00$ angenommen.

Die Diskretisierung der Struktur erfolgte zunächst mit $20 \times 2 \times 2$ linearen Volumenelementen (8 Knoten), wie in Abbildung 5.2 dargestellt. Um den Einfluss unregelmäßiger Elementgeometrien zu bewerten, führen vier Parameter d_1 , d_2 , d_3 und d_4 zu einer Abweichung der Elemente von der Quaderform. Dabei beeinflussen d_1 , d_2 und d_3 die Elementgeometrie in der Schalenebene, wobei immer $d_2 = (d_1 + d_3)/2$, also eine konstante Direktorrichtung der übereinanderliegenden Elemente gewählt wurde; d_4 verändert die Elementform in Dickenrichtung. Alle Parameter wirken sich nur auf die “inneren” Knoten aus, die Berandung der Struktur bleibt also in allen betrachteten Beispielen erhalten. Die durch die irregulären Netze entstehenden Phänomene sind vergleichbar mit Effekten, die bei der Simulation beliebig gekrümmter Strukturen auftreten, lassen sich jedoch hier besser differenzieren und bewerten. Im Fall kleiner Deformationen lässt sich beispielsweise mit der

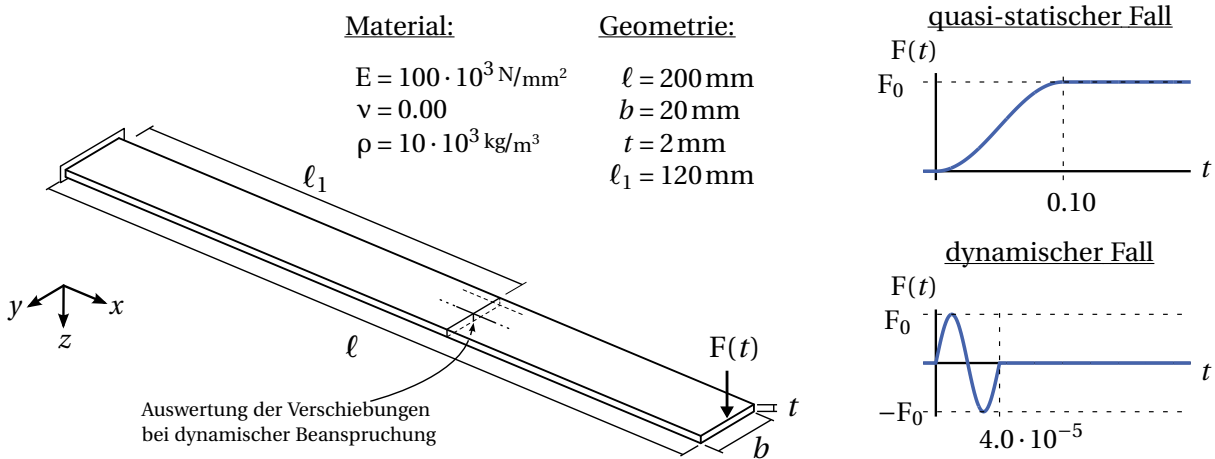


Abbildung 5.1.: Kragarm unter Einzellast – zeitlicher Verlauf der Belastung für quasi-statischen und dynamischen Fall

linearen Balkentheorie leicht eine analytische Lösung angeben. Abweichungen von der in Abbildung 5.2 skizzierten Diskretisierung werden an entsprechender Stelle erläutert.

5.1.2. Bewertung der Effizienz

Um den numerischen Aufwand verschiedener, mit dem in Kapitel 4 eingeführten Programmierkonzept implementierter, Elementformulierungen zu vergleichen, wird die oben beschriebene Struktur zunächst für $1.00 \cdot 10^{-3} \text{ s}$ unter Verwendung der quasi-statischen Lastkurve aus Abbildung 5.1 simuliert. Die Zeitschrittweite wird fest auf $\Delta t = 1.00 \cdot 10^{-8} \text{ s}$ gesetzt, was in jedem Fall kleiner als Δt_{krit} ist. Ein Vergleich der jeweils erforderlichen CPU-Zeit für 10000 Zeitschritte mit verschiedenen Elementformulierungen auf einer identischen Rechenumgebung gibt somit die Effizienz der einzelnen Elementroutinen im Sinne der zweiten in Kapitel 4 formulierten Definition an. Die im Folgenden angegebenen CPU-Zeiten sind Mittelwerte aus sechs Simulationen auf dem selben Rechner, wobei der jeweils kleinste und größte Wert nicht berücksichtigt wurden.

Um die Effizienz des in Kapitel 4 eingeführten Programmierkonzeptes selbst zu bewerten, wurde die reine Verschiebungsformulierung der 8-Knoten-Volumenschale VS08-DISP sowohl mit ACEGEN, als auch auf herkömmlichem Weg, also durch Unterteilung der Programmschritte in Unterroutrinen und manuelle Eingabe des Quellcodes umgesetzt. Eine genaue Beschreibung der erforderlichen Elementmatrizen, die bei der symbolischen Programmierung nicht explizit aufgestellt werden müssen, kann [50, 54] entnommen werden. Bei der manuellen Implementierungen wurde kein besonderer Wert auf eine effiziente Umsetzung gelegt, da dies auch bei Verwendung von ACEGEN nicht erforderlich ist. Durch die Option "Mode" -> "Optimal", erfolgt eine entsprechende Optimierung der Programmschritte automatisch. Zum Vergleich ist in der folgenden Tabelle zusätzlich der Zeitbedarf bei Verwendung einer ACEGEN-Routine jeweils mit "Mode" -> "Plain" (ohne jegliche Optimierung)

Ein Vergleich verschiedener, mit ACEGEN umgesetzter, Solid-Shell-Varianten mit 8 Knoten ergibt folgende Tabelle, wobei die CPU-Zeit der reinen Verschiebungsformulierung VS08-DISP als Referenzwert festgelegt wurde:

VS08-DISP	37 s	≅ 100 %
VS08-ANS	45 s	≅ 122 %
VS08-XDIS	65 s	≅ 176 %
VS08-ANS-EAS03	91 s	≅ 246 %
VS08-ANS-IM03	58 s	≅ 157 %
VS08-EAS09	199 s	≅ 538 %
VS08-IM09	62 s	≅ 168 %

Tabelle 5.2.: Effizienzvergleich – lineare Solid-Shell-Elemente (8 Knoten)

Auffällig ist der erheblich höhere Rechenaufwand bei Anwendung der EAS-Methode, der sich, wie bereits in Abschnitt 3.4.4 diskutiert, auf die Kondensation der lokalen Unbekannten auf Elementebene zurückführen lässt. Es sei angemerkt, dass bei der vorliegenden quasi-statischen Anwendung stets eine einzige Iteration zur Kondensation der EAS-Freiheitsgrade auf Elementebene ausreichend war. In dynamischen Anwendungen, also bei größeren Änderungen der Verschiebungen innerhalb eines Zeitschrittes, ist mehr als ein Iterationsschritt erforderlich, was in einem späteren Beispiel noch verdeutlicht wird. Der numerische Aufwand für die EAS-Elemente steigt dann weiter an, da die Operationen auf Elementebene pro Zeitschritt mehr als einmal durchgeführt werden müssen. Bei den IM-Elementen wurde die den inkompatiblen Moden zugeordnete Masse so gewählt, dass die gleiche Zeitschrittweite verwendet werden konnte. Erwartungsgemäß sind diese dann durch die Behandlung der zusätzlichen Freiheitsgrade als globale Unbekannte erheblich effizienter als die entsprechenden EAS-Formulierungen. Inwiefern eine solche Massenskalierung Einfluss auf die Korrektheit der Ergebnisse hat, wird an weiteren Beispielen untersucht. Die erforderliche CPU-Zeit muss dann an konkreten Beispielen unter Berücksichtigung der tatsächlichen Zeitschrittweite verglichen werden. Die Simulation mit VS08-XDIS als Alternative zu VS08-ANS benötigt etwas mehr Rechenzeit, was auf die aufwändigen Erweiterung der Verschiebungsinterpolation zurückzuführen sind. Offensichtlich ist die Auswertung und Interpolation der einzelnen Verzerrungskomponenten im hier betrachteten Fall numerisch günstiger. Die XDIS-Methode sollte sich auszahlen, sofern eine Erweiterung auf große Deformationen möglich ist, da dann direkt der für die Plastizitätsformulierung erforderliche, modifizierte Deformationsgradient angegeben werden kann.

Für die Solid-Shell-Elemente mit 18 Knoten wurde die Struktur mit $10 \times 1 \times 2$ Elementen diskretisiert, also die gleiche Anzahl Freiheitsgraden wie bei den 8-Knoten-Elementen verwendet. Auch hier wurden 10000 Zeitschritte mit $\Delta t = 1.00 \cdot 10^{-8} \text{ s} < \Delta t_{\text{krit}}$ simuliert und die erforderlichen CPU-Zeiten auf die der reinen Verschiebungsformulierung VS18-DISP bezogen:

VS18-DISP	42 s	≅ 100 %
VS18-ANS	275 s	≅ 655 %
VS18-ANS-EAS08	332 s	≅ 790 %

Tabelle 5.3.: Effizienzvergleich – quadratische Solid-Shell-Elemente (18 Knoten)

Im Vergleich zu den Solid-Shell-Elementen mit linearen Ansatzfunktionen fällt der we-

sentlich deutlichere Anstieg des numerischen Aufwandes bei Verwendung der ANS-Methode auf, da wie in Abschnitt 3.4.2 beschrieben fünf Verzerrungskomponenten behandelt werden und Auswertung sowie Interpolation entsprechend aufwändiger sind. Der direkte Vergleich der Formulierungen VS08-DISP und VS18-DISP zeigt einen kleinen Anstieg der Rechenzeiten bei der Verwendung quadratischer Geometrie- und Verschiebungsinterpolation bei gleicher Anzahl Freiheitsgrade und identischer Zeitschrittweite. Ein sinnvoller Vergleich beider Elementtypen ist jedoch nur an konkreten Beispielen unter Betrachtung der Ergebnisqualität möglich und wird im Folgenden noch geführt.

5.1.3. Lineare quasi-statische Analysen

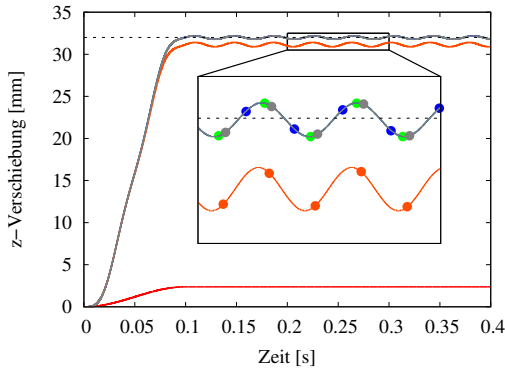
Wird die beschriebene Struktur quasi-statisch belastet, sodass die aufgebrachte Kraft lediglich kleine Deformationen verursacht, so kann im Rahmen der linearen Balkentheorie eine analytische Lösung angegeben werden. Mit der in Abbildung 5.1 angegebenen Lastkurve für quasi-statische Analysen, und einer maximalen Kraft von $F_0 = 16\text{ N}$, führt diese auf

$$w_{\text{stat}} = \frac{F \ell^3}{3EI} = 32\text{ mm} \quad \text{mit} \quad I = \frac{b h^3}{12} = \frac{40}{3}\text{ mm}^4. \quad (5.1)$$

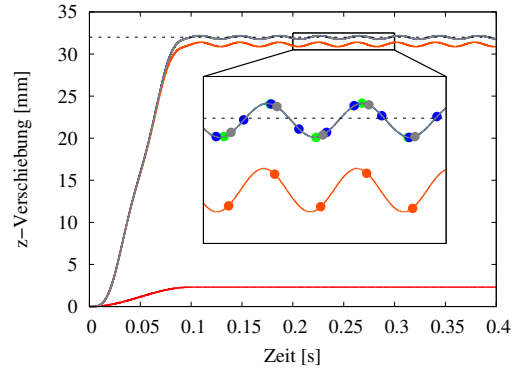
Für eine dynamische Analyse wird also entsprechend eine Schwingung um diese statische Lösung w_{stat} erwartet.

Abbildung 5.3 zeigt die grafische Auswertung der Ergebnisse für reguläre und irreguläre Diskretisierungen bei verschiedenen Elementformulierungen. Da durchgehend $\nu = 0.00$ gewählt wurde, kann hier der Einfluss geometrischer Versteifungseffekte bewertet werden. Die Simulationen wurden mit dem reinen Verschiebungselement VS08-DISP, mit dem ANS-Element VS08-ANS aus Abschnitt 3.4.2 sowie der Formulierung mit erweitertem Verschiebungsansatz VS08-XDIS aus Abschnitt 3.4.3 durchgeführt. Da letztere nur für kleine Deformationen hergeleitet wurde und nur bestimmte angenommene Verzerrungsverläufe beinhaltet, ist hier mit VS08-ANS-lin eine Elementformulierung für lineare Verzerrungen gemeint, bei der lediglich die Querschubverzerrungen nach den Gleichungen (3.52) und (3.53) modifiziert sind. Zum Vergleich sind die Ergebnisse mit VS08-ANS-lin33 angegeben, bei der zusätzlich die Normalverzerrungen in Dickenrichtung gemäß Gleichung (3.59) verändert wurden. Die Gültigkeit der Annahme kleiner Deformationen wurde durch VS08-ANS-nl bestätigt, einer ANS-Formulierung mit modifizierten Querschubverzerrungen und vollständig nichtlinearer Verzerrungsannahme. Bei der Wahl eines kleineren Wertes für F_0 , würden die in den Teilbildern der Abbildung 5.3 erkennbaren Unterschiede zwischen VS08-ANS-lin und VS08-ANS-nl entsprechende noch kleiner ausfallen.

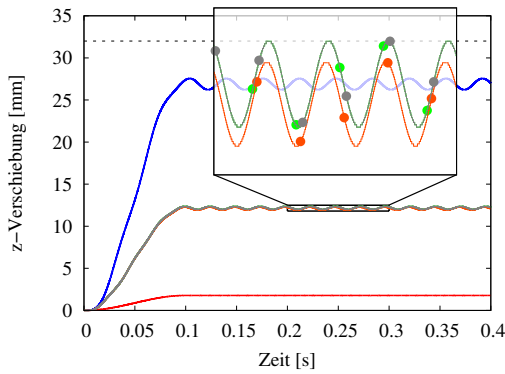
Die Teilbilder (a) und (b) der Abbildung 5.3 zeigen die erwarteten deutlichen Versteifungseffekte der reinen Verschiebungsformulierung, welche durch die ANS- und XDIS-Elemente vollständig beseitigt werden können. Sowohl bei regulärer Elementgeometrie, als auch bei lediglich in Schalenebene irregulären Elementen, also parallelen Schalendirektoren, führen beide Formulierungen erwartungsgemäß zu identischen Ergebnissen. Bei einer Schiefstellung der Direktoren, was bei einer regulären Diskretisierung einer gekrümmten Struktur automatisch auftritt, kommt bekanntermaßen zur Querschubversteifung ein weiterer Effekt hinzu, der im Schrifttum als ‘*Curvature-Thickness Locking*’ bezeichnet wird [50, 78]. Dessen Behandlung erfordert zusätzlich die Modifikation der Normalverzerrungen in Dickenrichtung, wie Abbildung 5.3 (c) zu entnehmen ist. Da hier eine Schiefstellung der Direktoren in



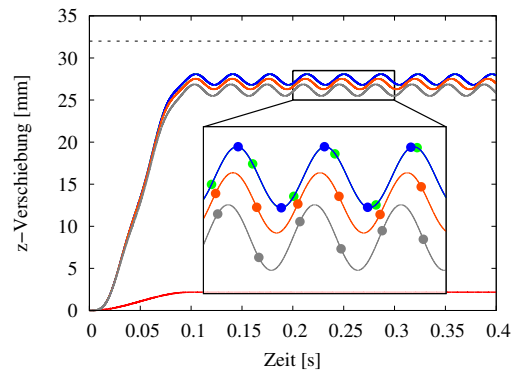
(a) $d_1 = 0.00, d_2 = 0.00, d_3 = 0.00, d_4 = 0.00$



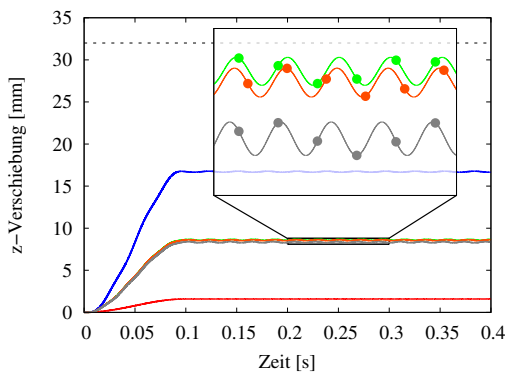
(b) $d_1 = 2.00, d_2 = 2.00, d_3 = 2.00, d_4 = 0.00$



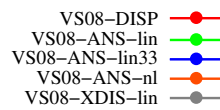
(c) $d_1 = 1.00, d_2 = 0.00, d_3 = -1.00, d_4 = 0.00$



(d) $d_1 = 2.00, d_2 = 2.00, d_3 = 2.00, d_4 = 0.40$



(e) $d_1 = 1.00, d_2 = 0.00, d_3 = -1.00, d_4 = 0.40$



(f) Legende

Abbildung 5.3.: Kragarm unter quasi-statisch aufgebrachtter Einzellast – vertikale Verschiebung des Kragarmendes bei kleinen Deformationen und Variation der Elementgeometrie

beiden Richtungen der Schalenebene erzwungen wird, kann die analytische Lösung jedoch auch von der VS08-ANS-lin33 Formulierung nicht mehr exakt reproduziert werden. Da alle Elemente bei dieser Diskretisierung noch eine konstante Dicke besitzen, sind die Ergebnisse für VS08-ANS und VS08-XDIS identisch. Wird durch die Wahl von $d_4 \neq 0.00$ auch eine veränderliche Elementdicke herbeigeführt, zeigen sich Unterschiede, deren Ursache bereits in Abschnitt 3.4.3 diskutiert wurde.

In diesem Beispiel wird der Einfluss der Elementgeometrie auf das Auftreten von Versteifungseffekten deutlich. Selbst Elementformulierungen, mit denen sich für regelmäßige Netze die analytische Lösung exakt reproduzieren lässt, können die auftretenden Versteifungseffekte nicht vollständig beseitigen. Bestimmte Effekte lassen sich oft nicht vermeiden – ‘*Curvature-Thickness Locking*’ beispielsweise tritt bei regelmäßiger Vernetzung einer gekrümmten Struktur automatisch auf – jedoch wird die Notwendigkeit einer ‘*sorgfältigen*’ Vernetzung deutlich, da sich so bereits gewisse Versteifungseffekte reduzieren oder ganz vermeiden lassen.

Das Beispiel zeigt außerdem die Möglichkeit der Reproduktion der ANS-Methode durch Modifikationen an der Verschiebungsinterpolation. Durch eine Erweiterung auf die Normalverzerrungen in Dickenrichtung, ließen sich auch Versteifungseffekte bei gekrümmten Strukturen beseitigen. Eine Untersuchung des Verhaltens bei großen Deformationen folgt im nächsten Abschnitt.

5.1.4. Nichtlineare quasi-statische Analysen

Zur Bewertung des Verhaltens verschiedener Elementformulierungen bei großen Deformationen, wurde wie im vorigen Beispiel die in Abbildung 5.1 angegebene Lastkurve für quasi-statische Analysen gewählt, jedoch eine maximale Kraft von $F_0 = 64\text{ N}$ aufgebracht.

Einfluss der Elementform auf ANS, EAS und IM-Elemente

Abbildung 5.4 (a) zeigt die grafische Auswertung der vertikalen Verschiebung des Kragarmendes bei Verwendung rechtwinkliger Elemente. Die drei untersuchten Elementformulierungen VS08-ANS, VS08-EAS09 und VS08-IM09 zeigen identisches Verhalten im Vergleich zur erwartungsgemäß deutlich versteifenden VS08-DISP Formulierung. Bei Verwendung der EAS-Methode reichte jeweils eine Iteration zur Kondensation der erweiterten Verzerrungsfreiheitsgrade aus. Die den erweiterten Verschiebungsfreiheitsgraden beim IM-Element zugeordnete Masse wurde mit dem Faktor $\alpha_m = 40.0$ skaliert, was die Verwendung einer unveränderten Zeitschrittweite erlaubte. Bei der quasi-statischen Anwendung hat diese Skalierung also offensichtlich keinen Einfluss auf das Abbildungsverhalten. Interessant ist für dieses Beispiel nochmal der Vergleich der erforderlichen CPU-Zeiten, also eine Bewertung

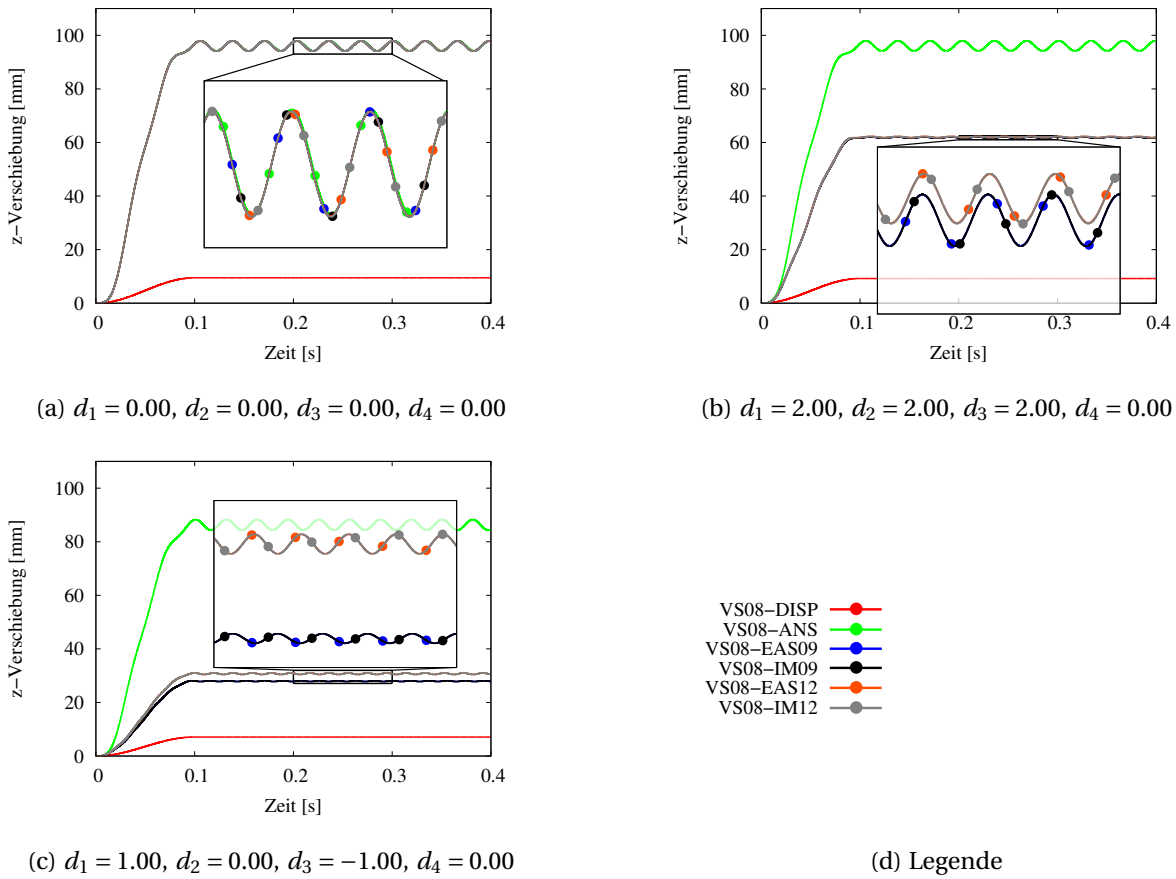


Abbildung 5.4.: Kragarm unter quasi-statisch aufgebrachtter Einzellast – vertikale Verschiebung des Kragarmendes bei großen Deformationen und Variation der Elementgeometrie – Vergleich zwischen ANS-, EAS- und IM-Formulierungen

des numerischen Aufwandes der einzelnen Methoden. Bezogen auf VS08-DISP sind diese:

VS08-DISP	638 s	≅ 100 %
VS08-ANS	768 s	≅ 120 %
VS08-EAS09	2827 s	≅ 443 %
VS08-IM09	902 s	≅ 141 %
VS08-EAS12	3685 s	≅ 578 %
VS08-IM12	971 s	≅ 152 %

Tabelle 5.4.: Effizienzvergleich – verschiedene Strategien zur Reduktion geometrischer Versteifungseffekte

Der numerische Aufwand für die VS08-EAS09 Formulierung mit lokaler Kondensation der zusätzlichen Unbekannten ist also etwa um Faktor 3 höher als der für das IM-Element, bei dem die inkompatiblen Verschiebungsfreiheitsgrade wie globale Unbekannte behandelt werden. Die in vielzähligen Arbeiten vorgeschlagenen Elementformulierungen mit sehr vielen EAS-Parametern scheinen also im Rahmen des zentralen Differenzenverfahrens nur wenig sinnvoll zu sein, da der Aufwand für die Kondensation entsprechend der Zahl der Erweiterungen ansteigt. Für das vorliegende Beispiel ist anzumerken, dass eine Formulierung mit

deutlich weniger Erweiterungen zu gleichen Ergebnissen führen würde. Wie in Tabelle 3.2 diskutiert, wird bei der gegebenen Belastungssituation mit $\nu = 0.00$ lediglich ein einziger inkompatibler Freiheitsgrad bzw. EAS-Parameter angeregt, auf die anderen könnte also verzichtet werden. Ein Kriterium zur problemabhängigen Beurteilung, welche Erweiterungen erforderlich sind, würde die numerische Effizienz der EAS- und IM-Methoden entsprechend erheblich steigern. Bei entsprechender Reduktion der Erweiterungen ist durchaus ein geringerer numerischer Aufwand als für die ANS-Formulierung zu erwarten.

Die Untersuchung des gleichen Beispiels mit von der rechtwinkligen Form abweichenden Elementgeometrien zeigt einen deutlichen Einfluss der Elementform auf die Abbildungsqualität bei den EAS- und IM-Elementen. Die Abbildungen 5.4 (b) und (c) zeigen zwei Beispiele, bei denen das ANS-Element keinen, bzw. nur wenig Abweichung von der Lösung mit regulären Elementen aufweist. Da dies für die VS08-EAS09 Formulierung bekannt ist, überrascht es auch für das VS08-IM09 Element nicht, da die aus den eingeführten Erweiterungen resultierenden Verzerrungen identisch sind. Die Empfindlichkeit gegenüber irregulären Elementformen sinkt bei der Verwendung von 12 Erweiterungen nur unwesentlich, bei gleichzeitig deutlichem Anstieg des numerischen Aufwandes.

Vergleich unterschiedlicher Ansatzordnungen

Ein Vergleich der Solid-Shell-Elemente mit linearer Interpolation von Geometrie und Verschiebungen in Schalenebene und Dickenrichtung mit den 16- bzw. 18-Knoten-Formulierungen, ist in Abbildung 5.5 angegeben. Die 16-Knoten-Elemente verwenden quadratische Serendipity-Polynome für die Interpolation in Schalenebene, bei den 18-Knoten-Formulierungen kommen quadratische LAGRANGE-Ansätze zum Einsatz. In allen Fällen wird in Dickenrichtung linear interpoliert. Es zeigt sich deutlich, dass durch die höhere Polynomordnung bereits ein erheblicher Teil der Elementversteifung eliminiert werden kann, die quadratischen DISP-Formulierungen also schon erheblich bessere Ergebnisse liefern, als das VS08-DISP Element. Weitere Verbesserungen der Abbildungseigenschaften und somit identische Ergebnisse wie mit VS08-ANS, können sowohl durch die Anwendung der ANS-Methode bei den quadratischen LAGRANGE-Elementen, als auch durch vollständig reduzierte Integration bei den Serendipity-Formulierungen erreicht werden. Wieder ist ein Vergleich der benötigten Rechenzeiten interessant:








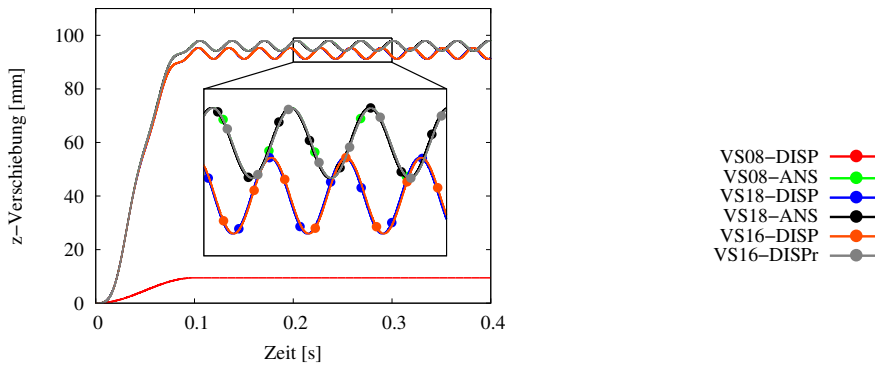
VS08-DISP		638 s	$\hat{=}$ 100 %
VS08-ANS		768 s	$\hat{=}$ 120 %
VS18-DISP		1 083 s	$\hat{=}$ 170 %
VS18-ANS	 / ... / 	6 984 s	$\hat{=}$ 1 095 %
VS16-DISP		966 s	$\hat{=}$ 151 %
VS16-DISPr		570 s	$\hat{=}$ 89 %

Tabelle 5.5.: Effizienzvergleich – lineare und quadratische Interpolation von Geometrie und Verschiebungen in Schalenebene

Die Tabelle zeigt erneut den erheblichen Anstieg des numerischen Aufwandes bei Verwendung der ANS-Methode für das quadratische Solid-Shell-Element, dessen Ursache bereits in Abschnitt 5.1.2 diskutiert wurde. Bei vollständiger Unterintegration und somit Beseitigung der Querschubversteifung, benötigt die 16-Knoten-Formulierung sogar weniger Rechenzeit, als das VS08-DISP Element. Für einen realistischen Vergleich der Wirtschaftlichkeit der

(a) $d_1 = 0.00$, $d_2 = 0.00$, $d_3 = 0.00$, $d_4 = 0.00$

(b) Legende

Abbildung 5.5.: Kragarm unter quasi-statisch aufgebrachtter Einzellast – vertikale Verschiebung des Kragarmendes bei großen Deformationen – Vergleich zwischen 8-, 16- und 18-Knoten-Formulierungen

verschiedenen Ansatzordnungen müssten hier jedoch zusätzlich verschiedene Diskretisierungen untersucht werden, um die Ergebnisqualität in die Bewertung mit einzubeziehen. Dies wird im weiteren Verlauf an einem dynamischen Beispiel noch kurz diskutiert.

XDIS für große Deformationen

Bei der Herleitung VS08-XDIS Formulierung wurde versucht, die linearen Anteile der durch ANS modifizierten Querschubverzerrungen durch entsprechende Anpassung der Verschiebungsinterpolation zu reproduzieren, ohne dabei andere Verzerrungskomponenten zu beeinflussen. In Abbildung 5.6 sind die Ergebnisse des Kragarmes unter quasi-statisch aufgebrachtter Einzellast für verschiedene Elementvarianten mit ANS und erweitertem Verschiebungsansatz verglichen. Die additive Aufspaltung des GREEN-LAGRANGE-Verzerrungstensors aus Gleichung (1.12) in lineare und nichtlineare Anteile, erlaubt eine separate Anwendung der ANS-Interpolationen. So werden bei VS08-ANS-l nur die linearen und bei VS08-ANS-nl nur die nichtlinearen Anteile modifiziert, wobei die jeweils anderen unverändert in den Gesamtverzerrungsverlauf eingehen. Eine Aufstellung des vollständigen Verzerrungstensors – also der linearen und nichtlinearen Anteile – mit der in Abschnitt 3.4.3 hergeleiteten Verschiebungsinterpolation führt auf die in Abbildung 5.6 mit VS08-XDIS-nl bezeichnete Elementformulierung. Werden die linearen Verzerrungsanteile mit den erweiterten Verschiebungen und die nichtlinearen mit dem herkömmlichen Verlauf bestimmt, so ergibt sich VS08-XDIS-l.

Die Untersuchung zeigt, dass sich sowohl die linearen als auch die nichtlinearen Anteile der Verzerrungen versteifende Anteile beinhalten, die offensichtlich durch Anwendung der ANS-Methode auf die gesamten Verläufe beseitigt werden können. Eine Auswertung und Interpolation lediglich der linearen Anteile reicht also nicht aus, um die gesamten Versteifungen zu beseitigen. Es ist zu erwarten, dass der Einfluss der in den nichtlinearen Anteilen enthaltenen Versteifungen bei stärker nichtlinearen Analysen weiter ansteigt. Eine direkte Berechnung der Verzerrungen aus den erweiterten Verschiebungen ist offensichtlich nicht möglich, die Formulierung VS08-XDIS-nl weist deutliche Abweichungen von der Lösung des ANS-Elementes auf. Die Erweiterung der Verschiebungen führt also zu einer falschen

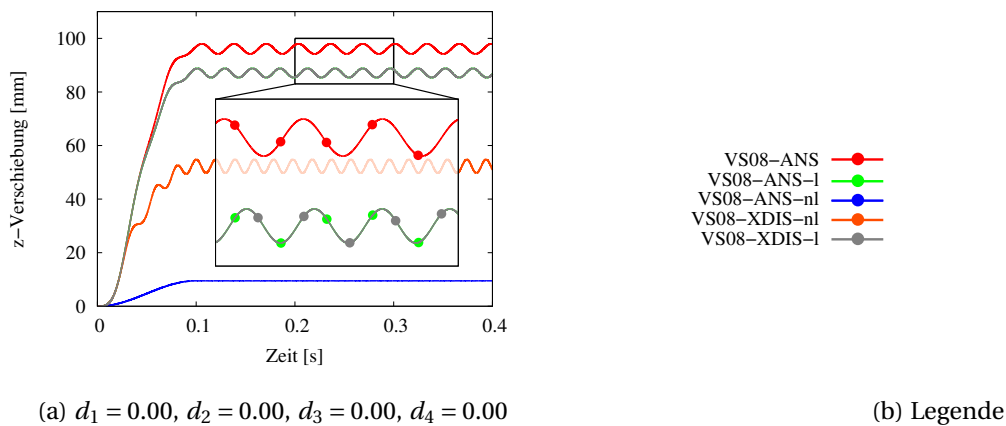


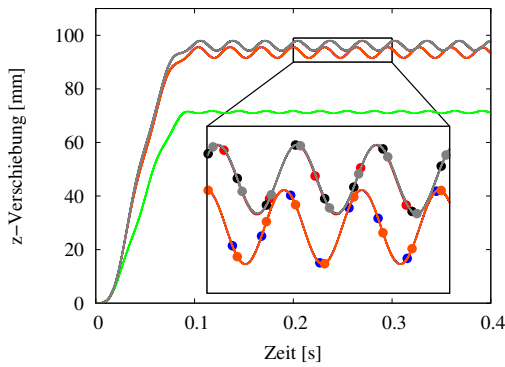
Abbildung 5.6.: Kragarm unter quasi-statisch aufgebrachtener Einzellast – vertikale Verschiebung des Kragarmendes bei großen Deformationen – Vergleich zwischen VS08-ANS und VS08-XDIS Formulierungen

Darstellung der nichtlinearen Verzerrungsanteile. Die Formulierung einer entsprechenden Verschiebungsinterpolation wäre jedoch für die Anwendung beispielsweise für Plastizität wünschenswert, da sich dann direkt der versteifungsfreie Deformationsgradient berechnen ließe. Die Ergebnisse mit den Formulierungen VS08-ANS-I und VS08-XDIS-I sind erwartungsgemäß identisch, da jeweils die gleichen Annahmen für die linearen Verzerrungen zu Grunde liegen und die unmodifizierten nichtlinearen Anteile addiert werden.

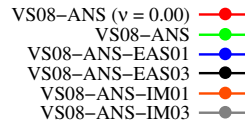
Verhalten bei nahezu inkompressiblem Material

Bei einer statisch bestimmten Lagerung der Struktur sollte sich die Verschiebung des Kragarmendes unabhängig von der Querkontraktionszahl ν einstellen. Um eine vollständig zwängungsfreie Lagerung zu ermöglichen, wurde das System mit nur $10 \times 1 \times 1$ linearen Solid-Shell-Elementen diskretisiert und quasi-statisch mit $F_0 = 64 \text{ N}$ belastet. Die Ergebnisse sind in Abbildung 5.7 dargestellt.

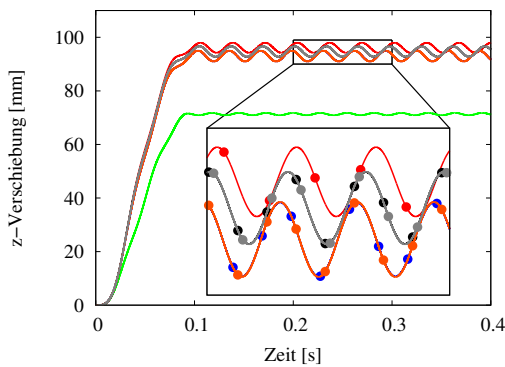
Bei der Verwendung der VS08-ANS Formulierung, bei der wie zuvor diskutiert geometrische Versteifungseffekte auszuschließen sind, zeigt sich eine deutliche Abweichung der Lösung bei nahezu inkompressiblem Material ($\nu = 0.499$) von der mit $\nu = 0.00$. Dies ist auf einen materiellen Versteifungseffekt zurückzuführen, der im Schrifttum mit 'POISSON-Dickenlocking' bezeichnet wird. Dieser lässt sich sowohl durch Anwendung der EAS-Methode, als auch durch Einführen entsprechender inkompatibler Moden behandeln. Durch die Erweiterung des Verzerrungsfeldes, können die bei $\nu \rightarrow 0.50$ auftretenden parasitären Spannungen eliminiert werden. Im vorliegenden Beispiel wurde, wie in den vorigen quasi-statischen Analysen, stets ein einziger Iterationsschritt zur Kondensation der lokalen EAS-Parameter verwendet. Die Masse der inkompatiblen Freiheitsgrade bei den IM-Elementen wurde wieder so angepasst, dass mit einem unveränderten Zeitschritt gerechnet werden konnte. Während die Elemente VS08-ANS-EAS01 und VS08-ANS-IM01 bereits deutlich weicher reagieren, tritt der Versteifungseffekt bei VS08-ANS-EAS03 bzw. VS08-ANS-IM03 praktisch nicht mehr auf. Die reinen EAS-Elemente VS08-EAS09 und VS08-IM12 können die volumetrische Versteifung erwartungsgemäß nicht vollständig beseitigen, da hier insbesondere die Normalverzerrungen in Dickenrichtung, für welche bei der gegebenen Belastung die verstei-



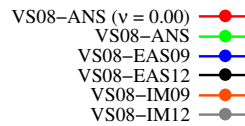
(a) ANS-EAS und ANS-IM-Elemente



(b) Legende zu (a)



(c) reine EAS- und IM-Elemente



(d) Legende zu (c)

Abbildung 5.7.: Kragarm unter quasi-statisch aufgebrachtter Einzellast – vertikale Verschiebung des Kragarmendes bei großen Deformationen $\nu = 0.499$ – Einfluss von EAS und IM auf volumetrisches Locking

fenden Anteile zu erwarten sind, nicht in der gleichen Form, wie bei der VS08-ANS-EAS03 Formulierung erweitert werden. Hier sind zusätzliche Erweiterungen erforderlich, wodurch der numerische Aufwand zur Kondensation der lokalen Unbekannten weiter ansteigt. In allen betrachteten Fällen lassen sich die Ergebnisse der EAS-Elemente durch Einführung entsprechender inkompatibler Moden reproduzieren. Für dieses Beispiel ergeben sich die Simulationszeiten zu

VS08-ANS	677 s	≅ 100 %
VS08-ANS-EAS01	832 s	≅ 123 %
VS08-ANS-EAS03	1 255 s	≅ 185 %
VS08-ANS-IM01	768 s	≅ 113 %
VS08-ANS-IM03	801 s	≅ 118 %
VS08-EAS09	2 340 s	≅ 346 %
VS08-EAS12	2 747 s	≅ 406 %
VS08-IM09	783 s	≅ 116 %
VS08-IM12	807 s	≅ 119 %

Tabelle 5.6.: Effizienzvergleich – EAS- und IM-Elemente bei $\nu = 0.499$

Erwartungsgemäß sind die IM-Formulierungen im quasi-statischen Fall deutlich effizien-

ter als die entsprechenden EAS-Elemente, da infolge entsprechender Wahl der Masse, eine Anpassung des Zeitschrittes nicht erforderlich ist. Außerdem erscheint im Kontext expliziter Zeitintegration die ausschließliche Verwendung von EAS zur Beseitigung von Locking unwirtschaftlich. Eine selektive Erweiterung der Verzerrungen in Kombination mit ANS führt zu deutlich effizienteren Elementformulierungen. Eine Bewertung des numerischen Aufwandes der IM- im Vergleich zu den EAS-Elementen in dynamischen Anwendungen wird im folgenden Abschnitt angegeben, da hier ein Einfluss sowohl der verwendeten Iterationsschritte zur Kondensation der EAS-Parameter, als auch der Massenskalierung der inkompatiblen Freiheitsgrade zu erwarten ist.

Durch die zusätzlichen Knoten der Elementformulierungen mit quadratischer Ansatzordnung, lässt sich die hier verwendete vollständig zwängungsfreie Lagerung nicht umsetzen. Der Einfluss der volumetrischen Versteifung lässt sich daher nicht ganz so anschaulich darstellen, tritt jedoch in gleichem Maße auch bei den VS16 und VS18 Formulierungen auf. Eine Beseitigung der volumetrischen Versteifung bei VS18-ANS-EAS08 im Vergleich zu VS18-ANS ist in gleicher Art zu erwarten, siehe hierzu u. a. [50].

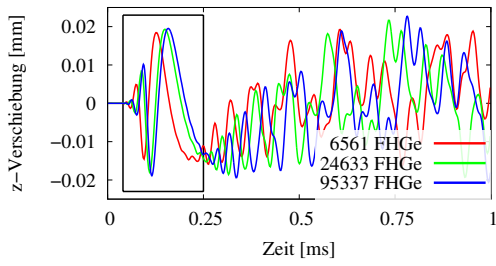
5.1.5. Dynamische Analysen

Ein wesentliches Anwendungsgebiet expliziter Zeitintegrationsverfahren ist die Simulation kurzzeitdynamischer Prozesse. Um die Eigenschaften verschiedener Elementformulierungen bei Auftreten höherer Frequenzen zu diskutieren, wird im Folgenden die in Abbildung 5.1 angegebene Lastkurve für dynamische Beanspruchung mit der Kraft $F_0 = 640\text{N}$ angewendet. Dabei ist der verwendete Zeitschritt stets so klein, dass der sinusförmige zeitliche Verlauf der Belastung bei jeder Simulation durch mindestens 500 Zeitschritte abgebildet wird. Bei einer derart kurzen Einwirkungszeit der Kraft von 0.04ms ist eine Systemantwort nicht ausschließlich in der unteren Eigenfrequenz zu erwarten. Eine Abbildung dieses dynamischen Verhaltens mit finiten Elementen erfordert erfahrungsgemäß eine recht hohe Netzfeinheit, weshalb zunächst eine Konvergenzuntersuchung durchgeführt wurde.

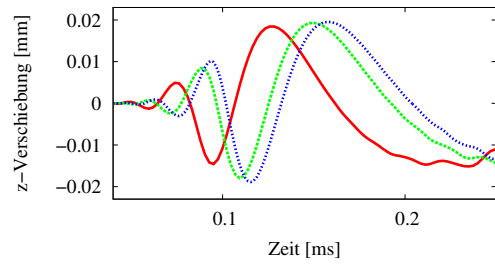
Die Abbildungen 5.8 und 5.9 zeigen die grafische Auswertung der Vertikalverschiebung des in Abbildung 5.1 angegebenen Punktes 120mm von der Einspannung des Kragarmes entfernt. Verwendet wurden hier lineare und quadratische Solid-Shell-Elemente jeweils ohne Modifikation der Verzerrungen und mit ANS. Die Simulationen wurden zunächst mit $\nu = 0.00$ durchgeführt, um materielle Versteifungseffekte auszuschließen. Die Diskretisierung erfolgte durchgehend mit 2 Elementen über die Dicke, in Schalenebene wurde bei der Verfeinerung stets die Elementkantenlänge in beiden Richtungen halbiert. Daraus ergeben sich die folgenden Netze:

- $80 \times 8 \times 2$ 8-Knoten-Elemente, bzw. $40 \times 4 \times 2$ 18-Knoten-Elemente $\hat{=}$ 6 561 FHGe
- $160 \times 16 \times 2$ 8-Knoten-Elemente, bzw. $80 \times 8 \times 2$ 18-Knoten-Elemente $\hat{=}$ 24 633 FHGe
- $320 \times 32 \times 2$ 8-Knoten-Elemente, bzw. $160 \times 16 \times 2$ 18-Knoten-Elemente $\hat{=}$ 95 337 FHGe

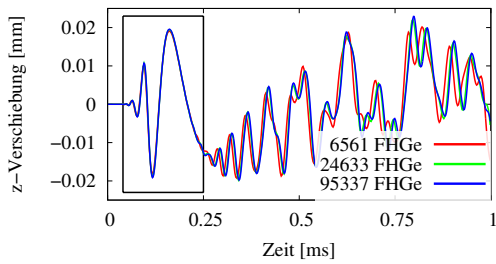
Es zeigt sich deutlich, dass sowohl die DISP-Formulierungen als auch die Elemente mit ANS bei ausreichender Netzfeinheit gegen die gleiche Lösung konvergieren, der Einfluss der geometrischen Versteifung also erwartungsgemäß bei feinen Netzen abnimmt. Bei den quasi-statischen Analysen konnte durch die Verwendung von ANS bereits bei einer recht groben Diskretisierung Konvergenz gegen die richtige Lösung festgestellt werden; die Abbildung dynamischer Effekte erfordert jedoch in jedem Fall feinere Netze, sodass insbesondere bei



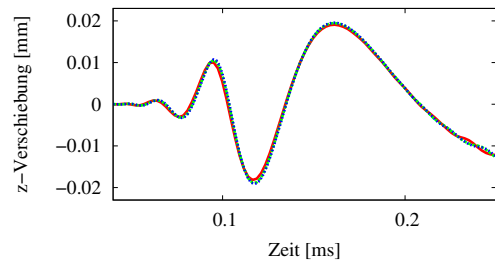
(a) VS08-DISP



(b) VS08-DISP – Ausschnitt

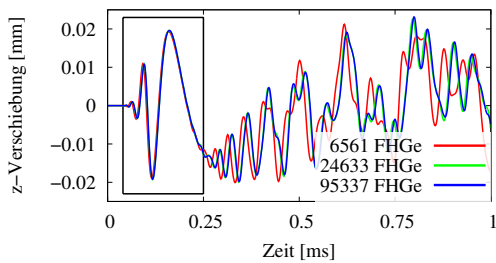


(c) VS08-ANS

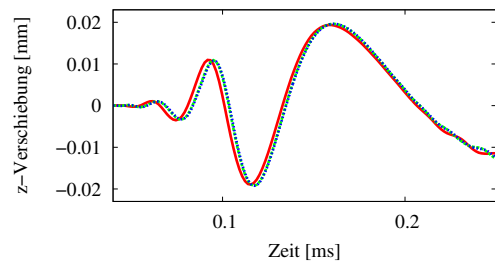


(d) VS08-ANS – Ausschnitt

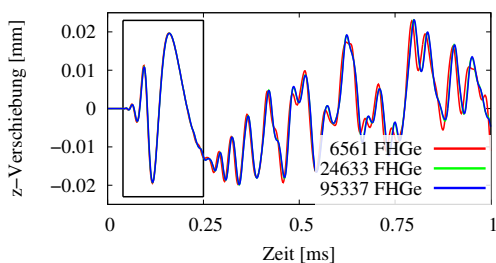
Abbildung 5.8.: Kragarm unter dynamisch aufgebrachtener Einzellast – Netzkonvergenzuntersuchung für lineare Solid-Shell-Elemente



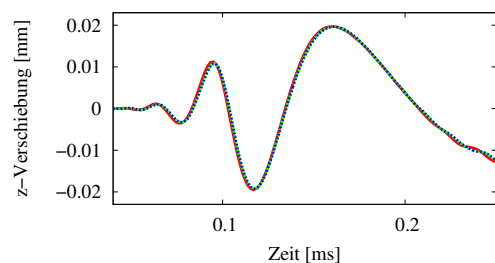
(a) VS18-DISP



(b) VS18-DISP – Ausschnitt



(c) VS18-ANS



(d) VS18-ANS – Ausschnitt

Abbildung 5.9.: Kragarm unter dynamisch aufgebrachtener Einzellast – Netzkonvergenzuntersuchung für quadratische Solid-Shell-Elemente

den Elementen mit quadratischer Ansatzordnung geometrische Versteifungseffekte praktisch keine Rolle mehr spielen. Dieser Effekt wird in Abbildung 5.10 beim direkten Vergleich der Ergebnisse für die verschiedenen Elementformulierungen mit 24 633, bzw. 95 337 Freiheitsgraden nochmal deutlicher. Während für die VS08-DISP Formulierung offensichtlich eine weitere Verfeinerung erforderlich ist, um das Ergebnis des VS08-ANS Elementes zu reproduzieren, ist bereits bei 24 633 Freiheitsgraden kaum ein Unterschied zwischen VS18-DISP und VS18-ANS zu erkennen. Es sei angemerkt, dass hier lediglich rechtwinklige Elementgeometrien betrachtet wurden, der Einfluss der Elementform bei dynamischer Beanspruchung also nicht untersucht wurde.

Im Gegensatz zu den geometrischen Versteifungseffekten, lässt sich materielles Locking, bei dem das zu steife Verhalten durch einen Materialparameter gesteuert wird, nicht durch Netzverfeinerung beheben. Bei endlicher Querkontraktionszahl muss also der oben beschriebene volumetrische Locking-Effekt auch bei kurzzeit-dynamischen Problemen mit sehr feiner Diskretisierung behandelt werden. Bei der Diskretisierung mit 24 633 Freiheitsgraden ist wie zuvor diskutiert bei Verwendung der VS08-ANS Formulierung keine geometrische Versteifung mehr zu erwarten, Unterschiede in den Verschiebungsantworten sind also ausschließlich auf materielle Versteifung zurückzuführen. Im Folgenden werden daher ausschließlich Simulationen mit $\nu = 0.499$ und dieser Diskretisierung verglichen.

Abbildung 5.11 zeigt den Vergleich zwischen VS08-ANS und VS08-ANS-EAS03 mit unterschiedlicher Anzahl Iterationsschritte zur Kondensation der lokalen Unbekannten auf Elementebene. Wie in Abschnitt 3.4.4 beschrieben, wurde ein Abbruchkriterium implementiert und über die Eingabedatei die maximale Anzahl durchzuführender Iterationsschritte angegeben. In Abbildung 5.11 sind die Ergebnisse für einen Schritt und maximal drei Schritte angegeben, bei näherer Betrachtung stellte sich heraus, dass im vorliegenden Beispiel stets nach maximal zwei Schritten Konvergenz erreicht wurde, mehr Iterationen also keine Veränderung der Ergebnisse mehr bewirkten. Es zeigt sich, dass die globale Antwort, also die Repräsentation der tieferen Frequenzen, bereits bei einem Iterationsschritt erreicht werden kann, die Abbildung höherer Frequenzen jedoch Konvergenz bei der Kondensation der lokalen Parameter erfordert. Es ist zu erwarten, dass bei hochfrequenten Anwendungen ein noch deutlicherer Einfluss der Iterationen zu erkennen ist, dann also noch mehr Schritte je Element zur Beseitigung der Versteifungseffekte erforderlich sind. Dies wirkt sich natürlich direkt auf die Effizienz der Elementformulierung aus, da bestimmte Operationen auf Elementebene dann zu jedem Zeitschritt mehrfach ausgeführt werden müssen.

Die Ergebnisse der EAS-Formulierung lassen sich wie im quasi-statischen Fall durch die Einführung entsprechender inkompatibler Moden reproduzieren. In den Abbildungen 5.11 (c) und (d) sind die Ergebnisse für verschiedene Skalierungen der den zusätzlichen Freiheitsgraden zugeordneten Masse dargestellt. Offensichtlich hat dieser Skalierungsfaktor bei der hier sehr feinen Diskretisierung – die verwendeten Elemente haben praktisch Würfelform – keinen signifikanten Einfluss. Wählt man $\alpha_m = 40.0$, so kann bereits der gleiche Zeitschritt wie bei der VS08-ANS-EAS03 Formulierung verwendet werden. Zum Vergleich erfordert $\alpha_m = 1.0$ eine derzeit manuelle Reduktion durch entsprechende Wahl von $\alpha_{\Delta t}$ auf ca. 15% des ursprünglichen Wertes. Zur Darstellung des Einflusses der Massenskalierung, ist noch eine Simulation mit einem sehr hohen Skalierungsfaktor $\alpha_m = 20000$ angegeben, bei der leichte Abweichungen zu erkennen sind. Bei Anwendungen mit stärkerer Beteiligung höherer Frequenzen oder Elementformen mit ungünstigeren Seitenverhältnissen, wird sich der Einfluss der Massenskalierung deutlicher bemerkbar machen, bzw. schon bei

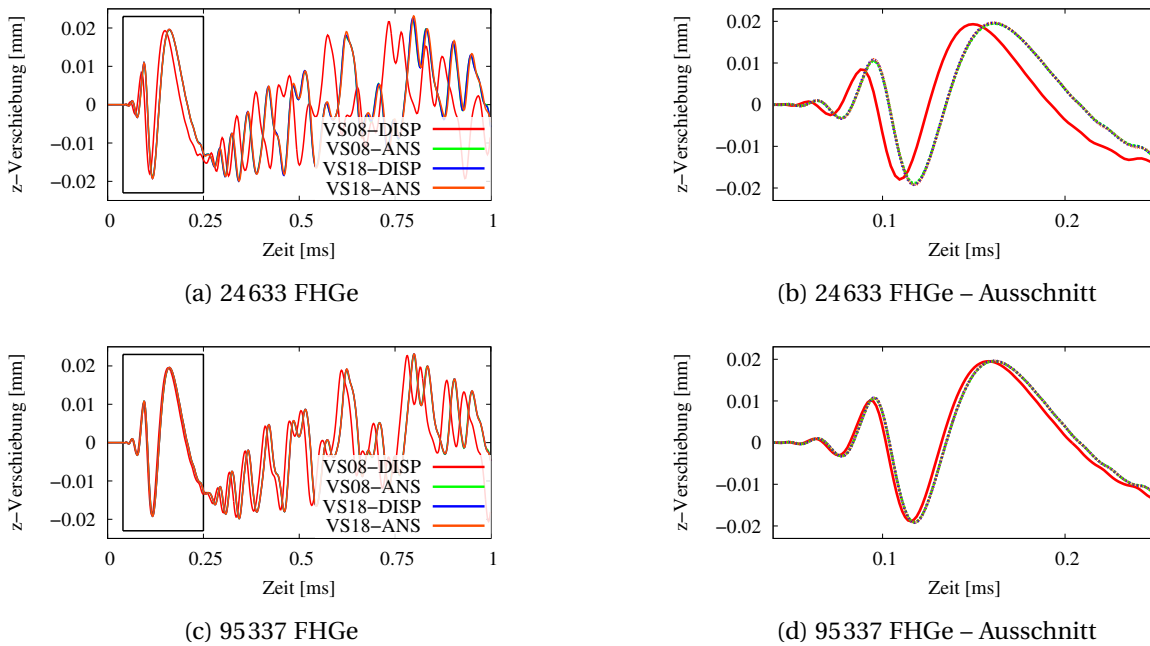


Abbildung 5.10.: Kragarm unter dynamisch aufgebrachtter Einzellast – Diskretisierungen mit 24633 und 95337 Freiheitsgraden

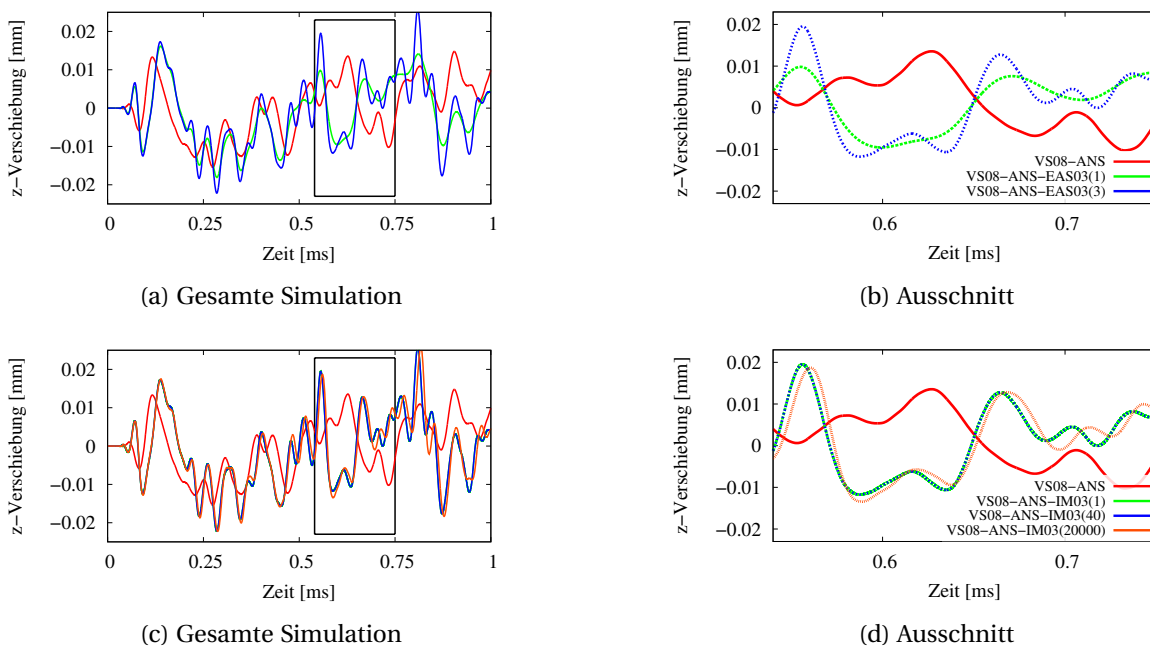


Abbildung 5.11.: Kragarm unter dynamisch aufgebrachtter Einzellast mit $\nu = 0.499$ – VS08-ANS, VS08-ANS-EAS03 und mit unterschiedlicher Anzahl Iterationsschritten sowie VS08-ANS-IM03 mit unterschiedlicher Massenskalierung α_m

kleineren Werten für α_m auftreten. Dann sollte die bereits angesprochenen Möglichkeiten einer selektiven Massenskalierung diskutiert werden, bei der die Masse der einzelnen zusätzlichen Freiheitsgrade individuell angegeben wird, um einzelne hohe Eigenfrequenzen gezielt abzusenken. Auch das Ersetzen eines bestimmten inkompatiblen Modes durch einen einzigen EAS-Parameter scheint vielversprechend, da der Aufwand für die Kondensation dann überschaubar ist.

Betrachtet man die Simulationszeiten für das vorliegende Beispiel mit $\nu = 0.499$ ergibt sich:

VS08-ANS	1 554 s	$\cong 100\%$
VS08-ANS-EAS03 (1)	3 074 s	$\cong 198\%$
VS08-ANS-EAS03 (3)	5 479 s	$\cong 353\%$
VS08-ANS-IM03 (1.00)	... / 13 899 s	$\cong 894\%$
VS08-ANS-IM03 (>40.00)	1 914 s	$\cong 123\%$

Tabelle 5.7.: Effizienzvergleich – EAS- und IM-Elemente bei $\nu = 0.499$ – Zahl in Klammern: maximale Anzahl Iterationsschritte (EAS), bzw. Skalierungsfaktor für inkompatible Masse (IM)

Erwartungsgemäß dauert die Simulation mit der VS08-ANS-IM03 Formulierung mit $\alpha_m = 1.00$ mit Abstand am längsten, was auf die erforderliche Reduzierung der Zeitschrittweite und damit die erheblich größere Anzahl an Zeitschritten zurückzuführen ist. Bei $\alpha_m = 40.00$, also bei gleicher Zeitschrittweite wie für die anderen Elementformulierungen, ist das VS08-ANS-IM03 wesentlich effizienter als das entsprechende EAS-Element, selbst bei nur einem einzigen Iterationsschritt. Offensichtlich ist also die Methode der inkompatiblen Moden insbesondere bei hochdynamischen Anwendungen günstiger als EAS, da hier die erforderliche feine Diskretisierung eine entsprechende Massenskalierung ohne signifikanten Einfluss auf das Systemverhalten erlaubt. Der Vergleich zwischen VS08-ANS-EAS03 mit maximal drei Iterationsschritten und VS08-ANS-IM03 mit $\alpha_m = 40.0$ ist nochmal in Abbildung 5.12 angegeben; beide Elementformulierungen führen auf identische Systemantworten. Dies muss noch an weiteren Beispielen getestet werden, besonders bei ungünstigeren Seitenverhältnissen der Elemente. Weiter sind eine automatische Bestimmung der Zeitschrittweite sowie eine Bewertung wann und in welchem Maße die den inkompatiblen Moden zugeordnete Masse skaliert werden kann erforderlich. Bei der EAS-Formulierung erfordert die Überprüfung der

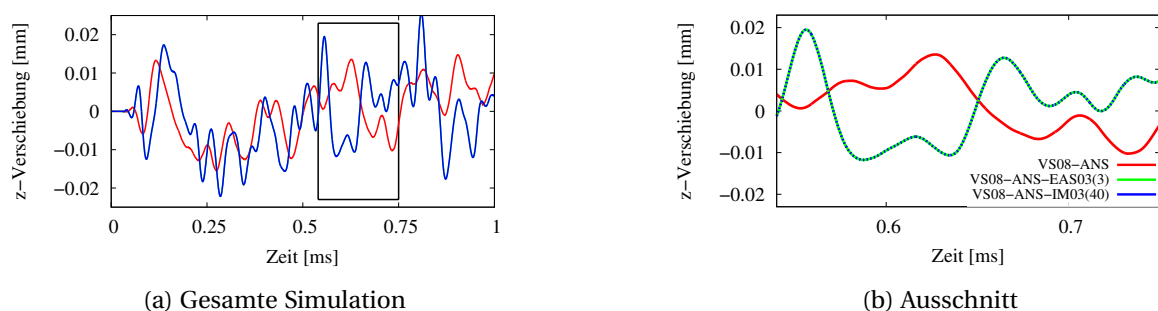


Abbildung 5.12.: Kragarm unter dynamisch aufgebrachtener Einzellast mit $\nu = 0.499$ – VS08-ANS, VS08-ANS-EAS03 mit drei Iterationsschritten und VS08-ANS-IM03 mit $\alpha_m = 40.0$

Konvergenz bei der Kondensation der Parameter zusätzliche Programmschritte, sodass hier möglicherweise die Angabe einer festen Anzahl Iterationsschritte günstiger sein kann.

5.2. Halbkugelschale unter dynamischer Biegebeanspruchung

5.2.1. Problembeschreibung

Untersucht wird die in Abbildung 5.13 dargestellte Halbkugelschale mit Loch, welche durch eine vorgegebene Vertikalverschiebung der Knoten entlang des Lochs beansprucht wird. Der Öffnungswinkel des Lochs beträgt 20° gemessen von der z -Achse, Radius und Dicke sowie die gewählten Materialparameter, sind Abbildung 5.13 zu entnehmen. Die Verschiebung wird über eine Zeit von 0.04 ms in Form einer Sinuswelle mit $w_{\max} = 0.25$ mm aufgebracht. Am unteren Rand der Halbkugel sind alle Knoten in vertikaler Richtung gehalten sowie die Verdrehungen behindert; in der Ebene können sich die Knoten frei verschieben. Symmetriestandbedingungen erlauben die Diskretisierung lediglich eines Viertels der Struktur, was in Abbildung 5.13 angedeutet ist. Für verschiedene Elementformulierungen aus den Familien der degenerierten Schalen und der Volumenschalen, wird die Verschiebung in x -Richtung des in der Abbildung mit P bezeichneten Punktes untersucht. Details zur Diskretisierung und den verwendeten Formulierungen werden im Folgenden angegeben.

5.2.2. Simulation mit degenerierten Schalenelementen

Für die implementierten degenerierten Schalenelemente mit linearen und quadratischen LAGRANGE-Ansätzen, erfolgte eine Simulation der beschriebenen Struktur mit folgenden Diskretisierungen für den in Abbildung 5.13 markierten Bereich der Struktur:

- 50×50 4-Knoten-Elemente, bzw. 25×25 9-Knoten-Elemente $\hat{=}$ 13 005 FHGe
- 100×100 4-Knoten-Elemente, bzw. 50×50 9-Knoten-Elemente $\hat{=}$ 51 005 FHGe
- 200×200 4-Knoten-Elemente, bzw. 100×100 9-Knoten-Elemente $\hat{=}$ 202 005 FHGe

Hierbei sind jeweils die gesamte Anzahl Freiheitsgrade angegeben, also auch die durch Randbedingungen gehaltenen. Für die Knoten entlang des Lochs wird die Verschiebung in z -Richtung gemäß des in Abbildung 5.13 angegebenen zeitlichen Verlaufs vorgegeben; zusätzlich sind jeweils beide Rotationsfreiheitsgrade festgehalten. Für die Knoten des unteren Endes der Struktur sind die Verschiebungen in z -Richtung sowie beide Rotationen behindert.

In Abbildung 5.14 sind die Verschiebungen des Punktes P in x -Richtung über die Zeit für verschiedene degenerierte Schalenformulierungen mit linearen und quadratischen Ansätzen aufgetragen. Sowohl die reinen Verschiebungsformulierungen (DISP), als auch die ANS-Elemente zeigen eine Konvergenz der Systemantwort bei entsprechend feiner Diskretisierung. Bei den 4-Knoten und 9-Knoten-Formulierungen mit ANS ändert sich durch die letzte Netzverfeinerung das Ergebnis nur noch geringfügig. Während für die DS09-DISP Formulierung ein ähnliches Verhalten zu erkennen ist – geometrische Versteifungseffekte also durch die Netzfeinheit schon weitestgehend beseitigt zu sein scheinen – erfordert das lineare

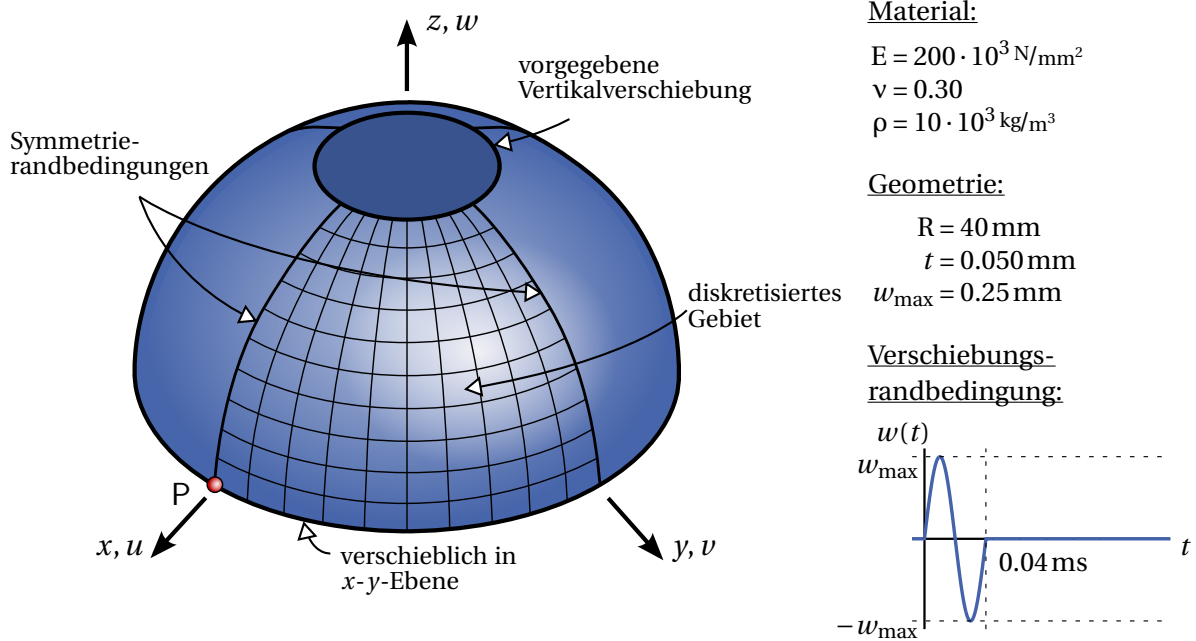


Abbildung 5.13.: Halbkugelschale unter Biegebeanspruchung – Geometrie, Materialeigenschaften und Belastung

DS04-DISP Element offensichtlich eine noch feinere Diskretisierung. Unklar ist die Interpretation der Oszillationen, die bei der feinsten Diskretisierung mit DS04-DISP im letzten Drittel des zeitlichen Verlaufes auftreten, sowie bei DS09-DISP ganz am Ende des untersuchten Zeitbereichs zu erkennen sind. Eine weitere Simulation mit kleinerem Zeitschritt führte zu keiner Veränderung. In praktischen Anwendungen würde ein solcher – offensichtlich durch die Kombination aus Diskretisierung und Elementtyp erzeugter – Effekt, durch Einführung einer moderaten Dämpfung korrigiert, worauf hier verzichtet wurde.

In Abbildung 5.15 sind die Ergebnisse der Elemente mit linearer und quadratischer Ansatzordnung für verschiedene Netzfeinheiten direkt verglichen. Hier zeigt sich nochmal, dass im Fall der reinen Verschiebungsformulierung mit dem DS09-DISP Element Konvergenz erreicht wird, während die Ergebnisse für die entsprechende 4-Knoten-Formulierung noch sehr stark abweichen. Bei den 9-Knoten-Elementen spielen die Modifikationen der Verzerrungen durch ANS bei entsprechender Netzfeinheit also praktisch keine Rolle mehr, da kaum noch geometrische Versteifungseffekte auftreten. Die Krümmung der Struktur kann im Falle sehr feiner Netze offensichtlich auch mit linearen Elementen hinreichend gut abgebildet werden. So werden bei quadratischer Interpolation ebenso viele Freiheitsgrade für eine konvergierte Lösung benötigt, wie bei linearer Ansatzordnung. Ein Vergleich der

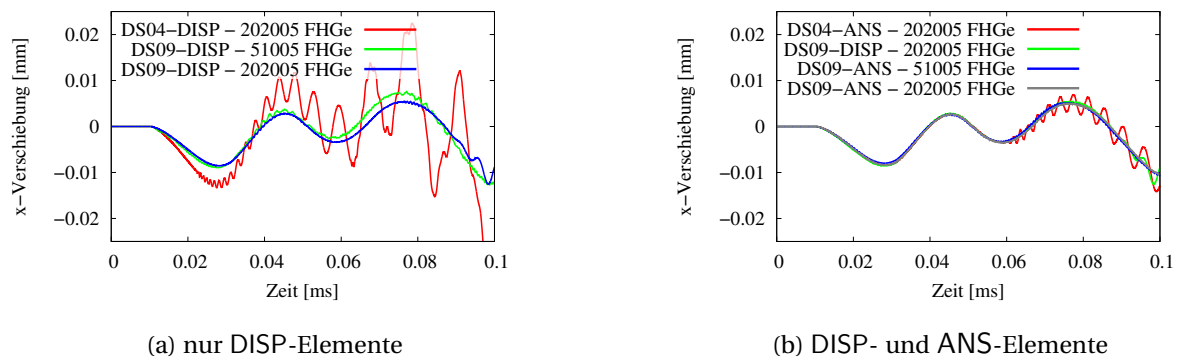


Abbildung 5.15.: Halbkugelschale mit degenerierten Schalenelementen – Vergleich der Systemantworten bei P für 4-Knoten und 9-Knoten-Elemente

fende Versteifungseffekte zu verbessern. Es wurden zwei Elementformulierungen mit 4 und 7 Erweiterungen implementiert, deren Ergebnisse in Abbildung 5.16 dargestellt sind. Wie direkt zu erkennen ist, führt die Anwendung der EAS-Methode zu unwesentlicher Veränderung der Ergebnisse, die Membranverzerrungen haben hier also offensichtlich keinen wesentlichen Einfluss auf die Versteifungseffekte. Um den Einfluss der Erweiterungen zu verdeutlichen, wurden zusätzlich Simulationen ohne ANS durchgeführt (Abbildung 5.16 (b) und (d)). Ein geringer Einfluss der Membranversteifung ist lediglich bei der groben Diskretisierung zu erkennen. Da es sich wie bei der Querschubversteifung um einen geometrischen Versteifungseffekt handelt, verschwindet dessen Einfluss bei feineren Netzen, sodass EAS hier nicht erforderlich ist. Zwischen DS04-EAS04 und DS04-EAS07 lässt sich praktisch kein Unterschied feststellen. Dies wurde in der Vergangenheit bereits im Schrifttum diskutiert, siehe z. B. [54]. Eine Gegenüberstellung des numerischen Aufwandes für die degenerierte Schalenformulierung mit und ohne EAS liefert beispielsweise für die Diskretisierung mit 202005 Freiheitsgraden:

DS04-DISP	3775 s	≅ 100 %
DS04-EAS04	14305 s	≅ 379 %
DS04-EAS07	18044 s	≅ 478 %

Tabelle 5.9.: Halbkugelschale mit degenerierten Schalenelementen – Numerischer Aufwand der Simulationen mit linearen Ansatzfunktionen und unterschiedlichen EAS-Erweiterungen

Insbesondere bei der hier vorliegenden, hochdynamischen Beanspruchung, steigt der numerische Aufwand bei Verwendung von EAS erheblich. Denkbar wäre, wie bereits im vorigen Beispiel für die Volumenschalen diskutiert, die Notwendigkeit der unterschiedlichen EAS-Parameter in regelmäßigen Intervallen während einer Simulation zu überprüfen und nur die für die vorliegende Belastungssituation erforderlichen Erweiterungen zu aktivieren. Neben der Frage nach einem entsprechenden Kriterium, die im Rahmen dieser Arbeit nicht beantwortet werden soll, erfordert eine solche Strategie die Implementierung aller möglichen, die EAS-Erweiterungen betreffenden Kombinationen. Dies lässt sich mit Hilfe des in Abschnitt 4 vorgestellte Programmierkonzeptes sehr einfach realisieren, da jeweils nur minimale Änderungen an der Programmierung mit ACEGEN erforderlich sind.

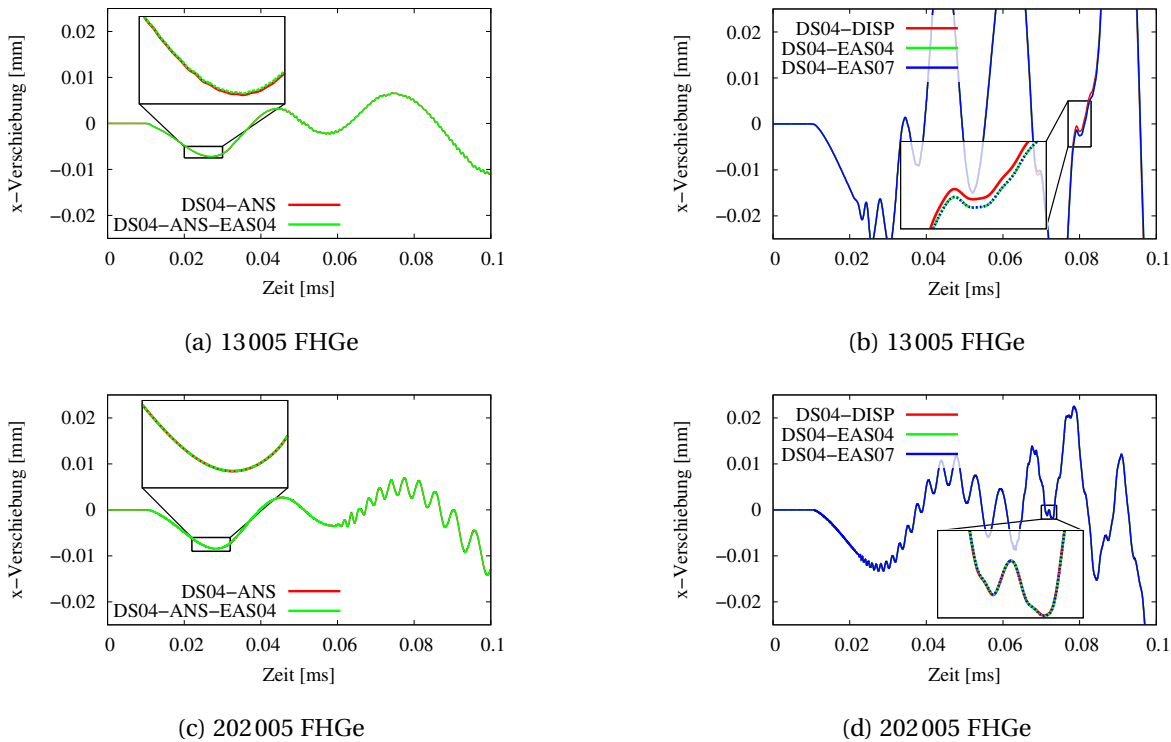


Abbildung 5.16.: Halbkugelschale mit degenerierten Schalenelementen – Vergleich der Systemantworten bei P für unterschiedliche EAS-Erweiterungen der Membranverzerrungen

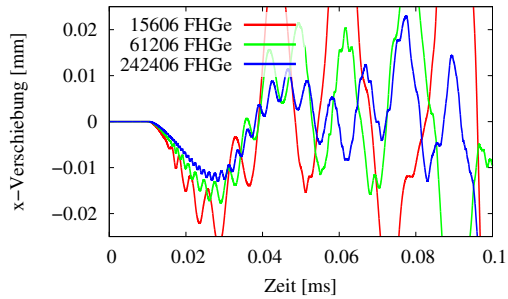
5.2.3. Simulation mit Volumenschalenelementen

In Analogie zu den degenerierten Schalenelementen, erfolgte die Diskretisierung mit den 8-Knoten und 18-Knoten-Volumenschalen jeweils mit einem einzigen Element über die Dicke. Daraus ergeben sich die untersuchten Netze:

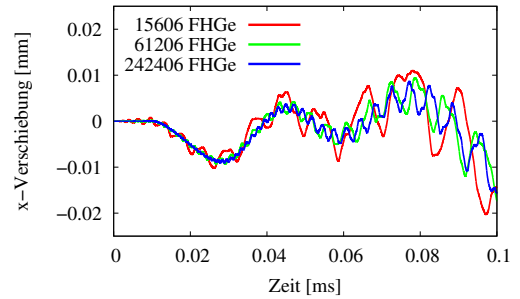
- 50×50 8-Knoten-Elemente, bzw. 25×25 18-Knoten-Elemente $\hat{=}$ 15 606 FHGe
- 100×100 8-Knoten-Elemente, bzw. 50×50 18-Knoten-Elemente $\hat{=}$ 61 206 FHGe
- 200×200 8-Knoten-Elemente, bzw. 100×100 18-Knoten-Elemente $\hat{=}$ 242 406 FHGe

Der in Abbildung 5.13 angegebene Verschiebungsverlauf wird auf alle Knoten entlang des Lochs aufgebracht. Dadurch ergibt sich auch eine entsprechende Behinderung der Rotationen. Ebenso sind alle Knoten am unteren Rand in z -Richtung gehalten, wodurch die Rotationen dort ebenso behindert sind.

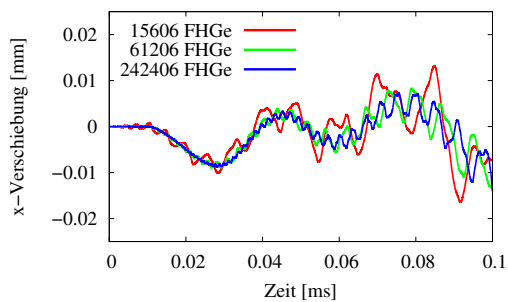
In Abbildung 5.17 werden die Verschiebung des Punktes P in x -Richtung für verschiedene Netzfeinheiten und Elementformulierungen verglichen. Um die gleiche geometrische Lage wie bei den Modellen mit degenerierten Schalenelementen zu erreichen, wurden die Verschiebungen der beiden Knoten über die Dicke für die Ausgabe gemittelt. Neben der Beseitigung geometrischer Versteifungseffekte – insbesondere Biegeversteifungen – mit ANS, sind bei den VS08-ANS-EAS03, bzw. VS08-ANS-IM03 Formulierungen die Normalverzerrungen in Dickenrichtung erweitert, um die bei $\nu \neq 0$ auftretende volumetrische Versteifung zu behandeln. Dabei wurde beim VS08-ANS-IM03 Element eine Massenskalierung $\alpha_m = 400.0$ gewählt und der Zeitschritt gegenüber der EAS-Formulierung um den Faktor 5 reduziert.



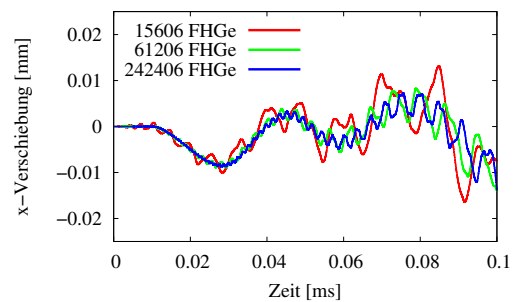
(a) VS08-DISP



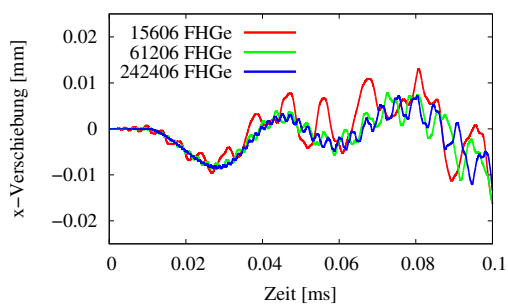
(b) VS08-ANS



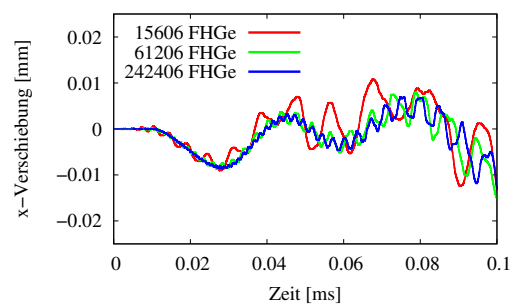
(c) VS08-ANS-EAS03



(d) VS08-ANS-IM03



(e) VS08-EAS09



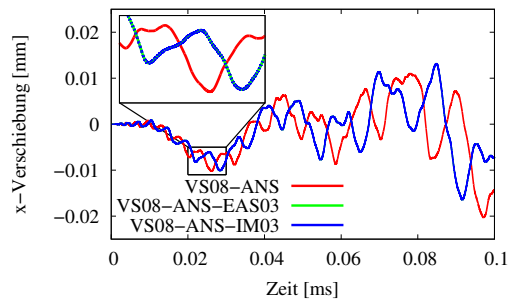
(f) VS08-IM09

Abbildung 5.17.: Halbkugelschale mit Volumenschalenelementen – Vergleich der Systemantworten bei P für verschiedenen 8-Knoten-Formulierungen und unterschiedliche Netzfeinheiten

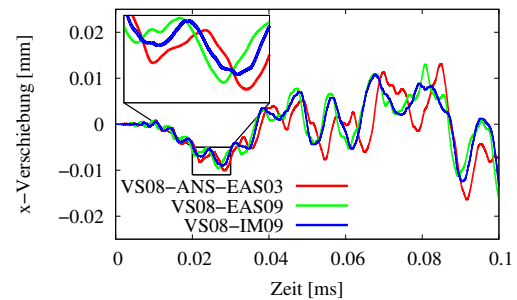
Weiter erfolgten Simulationen mit den VS08-EAS09 und VS08-IM09 Formulierungen mit den gleichen Werten für Massenskalierung und Zeitschritt. Abgesehen von der sehr steifen reinen Verschiebungsformulierung, bei der weitere Netzverfeinerungen erforderlich sind, weisen alle untersuchten Elementtypen ein klares Konvergenzverhalten auf. In allen Fällen sind die Unterschiede zwischen den Diskretisierungen mit 61 206 und 242 406 Freiheitsgraden sehr gering.

Eine genauere Betrachtung des Einflusses der EAS-, bzw. IM-Erweiterungen auf die volumetrische Versteifung erfolgt in Abbildung 5.18. Auf der linken Seite sind die Ergebnisse für verschiedene Diskretisierungen der VS08-ANS Formulierung ohne Modifikation der Normalverzerrung in Dickenrichtung, mit EAS bzw. mit entsprechenden inkompatiblen Moden verglichen. Da der volumetrische Versteifungseffekt nicht durch Netzverfeinerung beseitigt werden kann, stellt sich auch beim feinsten Netz noch ein Unterschied ein. Die Ergebnisse mit EAS sind identisch mit denen der IM-Formulierung, die gewählte Skalierung der Masse führt also nicht zu signifikanten Änderungen am dynamischen Verhalten. Das rechte Teilbild zeigt die Resultate für die VS08-EAS09 und VS08-IM09 Elemente. Bei grober Diskretisierung sind deutliche Abweichungen zu den Ergebnissen mit der VS08-ANS-EAS03 Formulierung zu erkennen, die auf den Einfluss der nicht-rechtwinkligen Elementgeometrien bei der gekrümmten Struktur zurückzuführen sind. Dieser Effekt verschwindet bei Netzverfeinerung, da dann die Elemente näher an der Quaderform sind. Ebenso wird eine Abweichung zwischen den VS08-EAS09 und VS08-IM09 Elementen deutlich, die auf die gewählte Massenskalierung zurückzuführen ist. Offensichtlich hat diese bei grober Diskretisierung einen höheren Einfluss als bei feinen Netzen, da auch dieser Unterschied bei Netzverfeinerung verschwindet. Interessant sind die bei identischer Massenskalierung auftretenden, erheblichen Unterschiede zwischen den VS08-EAS09 und VS08-IM09 Elementen, während sich die VS08-ANS-EAS03 und VS08-ANS-IM03 Elementen praktisch gleich verhalten. Eine genauere Untersuchung des Einflusses der Massenskalierung auf die Abbildungseigenschaften bei verschiedenen IM-Elementen sollte in Erweiterung der vorliegenden Untersuchungen noch durchgeführt werden und wird für die Formulierung entsprechender Kriterien zur Wahl der Skalierung und des zugehörigen Zeitschrittes hilfreich sein.

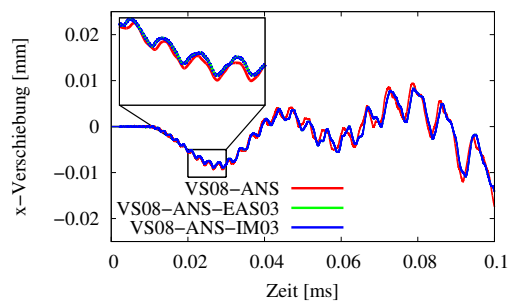
In Abbildung 5.19 werden zwei Elementformulierungen VS08-ANS-IM03-2 und VS08-IM09-2 eingeführt, bei denen die den inkompatiblen Freiheitsgraden zugeordnete Masse mit $\alpha_m = 6000$ skaliert wurde. Für die vorliegende Geometrie erlaubte dies die Verwendung der gleichen Zeitschrittweite wie bei den Elementen ohne inkompatible Moden. Es zeigt sich wie zuvor ein deutlicher Einfluss bei der VS08-IM09 Formulierung, wohingegen die Masse der die Normalverzerrungen in Dickenrichtung beeinflussenden inkompatiblen Freiheitsgrade beim VS08-ANS-IM03 Element offensichtlich keinen signifikanten Einfluss auf das Abbildungsverhalten hat. Da es sich hier um ein biegedominantes Problem handelt, beeinflussen also offensichtlich diejenigen Massen das dynamische Verhalten der Struktur, die den Schubverzerrungen zugeordnet sind. Daher erscheint eine Kombination aus ANS und inkompatiblen Moden für die Normalverzerrungen in Dickenrichtung hier optimal, da die Masse so skaliert werden kann, dass keine Anpassung der Zeitschrittweite erforderlich ist. Erkennbar in Abbildung 5.19 ist außerdem die Abnahme des Einflusses der Massenskalierung für zunehmende Netzfeinheit, sodass bei sehr feinen Netzen auch beim VS08-IM09 Element keine Anpassung des Zeitschrittes erforderlich ist. Eine Alternative stellt die Kombination der Erweiterungen aus EAS und inkompatiblen Moden – die VS08-IM08-EAS01 Formulierung – dar, bei der lediglich die Erweiterung der Normalverzerrung in Dickenrichtung, für die in Abschnitt 3.4.5



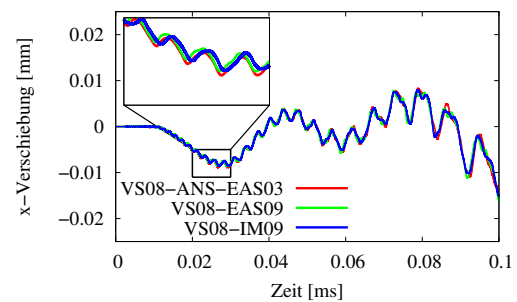
(a) 15606 FHGe



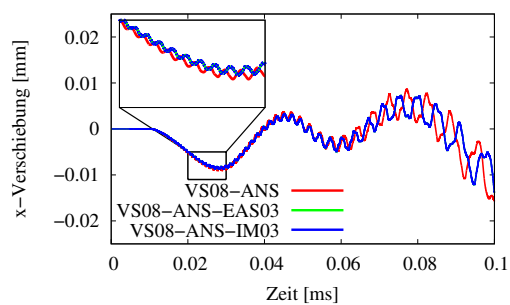
(b) 15606 FHGe



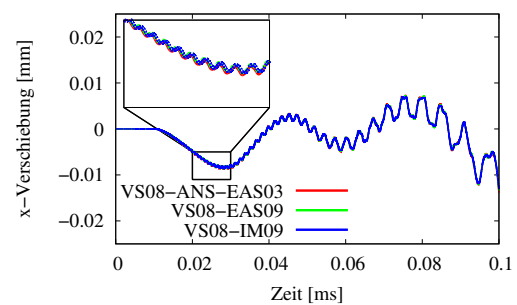
(c) 61206 FHGe



(d) 61206 FHGe



(e) 242406 FHGe



(f) 242406 FHGe

Abbildung 5.18.: Halbkugelschale mit Volumenschalenelementen – Vergleich der Systemantworten bei P für verschiedenen EAS- und IM-Formulierungen mit und ohne ANS und unterschiedliche Netzfeinheiten

die höchste Eigenfrequenz identifiziert wurde, durch einen einzigen EAS-Parameter ersetzt wurde. Erwartungsgemäß ergeben sich die gleichen Ergebnisse wie beim VS08-IM09 Element, jedoch kann auch bei der kleineren Massenskalierung von $\alpha_m = 400.0$ der unveränderte Zeitschritt verwendet werden.

Von besonderem Interesse sind die erforderlichen Rechenzeiten, die sich beispielsweise für die Diskretisierung mit 242 406 Freiheitsgraden wie folgt ergeben:












VS08-ANS	 5 054 s	$\cong 100\%$
VS08-ANS-EAS03	 16 538 s	$\cong 327\%$
VS08-ANS-IM03	 / ... /  30 426 s	$\cong 602\%$
VS08-ANS-IM03-2	 6 051 s	$\cong 120\%$
VS08-EAS09	 / ... /  33 670 s	$\cong 666\%$
VS08-IM09	 / ... /  30 233 s	$\cong 598\%$
VS08-IM09-2	 6 231 s	$\cong 123\%$
VS08-IM08-EAS01	 11 995 s	$\cong 237\%$

Tabelle 5.10.: Halbkugelschale mit Volumenschalenelementen – Numerischer Aufwand der Simulationen mit EAS- und IM-Erweiterungen, Massenskalierung mit $\alpha_{m,1} = 400$ bei IM0 \times und $\alpha_{m,2} = 6000$ bei IM0 \times -2

Hier lässt sich erkennen, dass die Methode der inkompatiblen Moden innerhalb eines expliziten Zeitintegrationsverfahrens im Hinblick auf die Effizienz eine gute Alternative zur bewährten EAS-Methode darstellen kann. Der erforderliche numerische Aufwand, hängt jedoch signifikant von der Wahl der den zusätzlichen Freiheitsgraden zugeordneten Massen ab, welche sich wiederum erheblich auf das dynamische Verhalten der Elemente auswirken kann. Eine genaue Untersuchung zur Wahl der Masse und der zugehörigen Zeitschrittweite wurde im Rahmen dieser Arbeit noch nicht durchgeführt. Erste Untersuchungen der Eigenformen unter Berücksichtigung der inkompatiblen Moden führten bereits zu interessanten Ergebnissen [88]. Hier besteht jedoch weiterer Forschungsbedarf, um in Abhängigkeit der jeweiligen Diskretisierung und Belastungssituation möglichst allgemeine Kriterien für die zulässige Massenskalierung formulieren zu können. Die Elementformulierung VS08-IM08-EAS01 zeigt im obigen Beispiel die Möglichkeit der Kombination von EAS- und IM-Erweiterungen, um einen zu starken Einfluss auf den kritischen Zeitschritt zu vermeiden. Der vergleichsweise geringe numerische Aufwand für einen einzigen EAS-Parameter erlaubt hier eine fünfmal größeren Zeitschritt und führt entsprechend zu einem wesentlich effizienterem Element im Vergleich zum VS08-IM09 Element. In Abschnitt 3.4.5 wurde die Möglichkeit einer selektiven Massenskalierung angesprochen, deren weitere Untersuchung ebenfalls Gegenstand der aktuellen Untersuchungen ist. Die Masse für die einzelnen zusätzlichen Freiheitsgrade wird hierbei unabhängig voneinander gewählt, um gerade eine gewünschte Eigenfrequenz nicht zu über- und somit eine bestimmte kritische Zeitschrittweite nicht zu unterschreiten. So kann der Einfluss auf das dynamische Elementverhalten so gering wie möglich gehalten werden.

Als Vorschlag zur Steigerung der Effizienz bei den EAS-Elementen, wurde in Abschnitt 3.4.4 die Diagonalisierung der für die Kondensation zu invertierenden Matrix \mathbf{D}_{EAS} angeführt. Abbildung 5.20 zeigt die Ergebnisse für die Elementformulierungen mit 9 und mit 12 EAS-Erweiterungen am Beispiel der Diskretisierung mit 242 406 Freiheitsgraden. Bei den Elementformulierungen mit dem Zusatz (1), wurde lediglich ein Iterationsschritt für die Kondensation gewählt. Zunächst ist zu erkennen, dass die drei zusätzlichen EAS-Erweiterungen bei

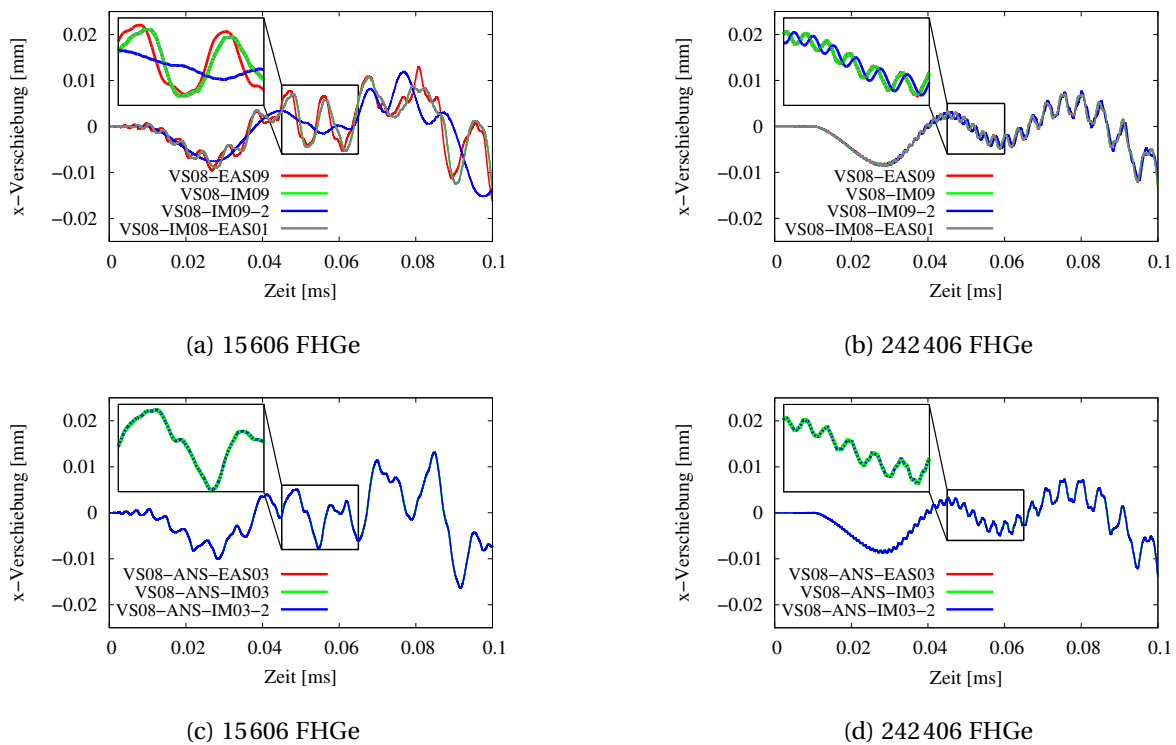


Abbildung 5.19.: Halbkugelschale mit Volumenschalenelementen – Vergleich der Systemantworten bei P für IM-Formulierungen mit verschiedenen Massenskalierungen und unterschiedliche Netzfeinheiten

der VS08-EAS12 Formulierung keine Änderung im Systemverhalten hervorrufen, was auf die moderate Querdehnzahl von $\nu = 0.30$ zurückzuführen ist. Eine zusätzliche Erweiterung der Normalverzerrungen wirkt sich also nur bei nahezu inkompressiblem Materialverhalten und entsprechend deutlicher volumetrischer Versteifung aus. Die Diagonalisierung der Matrix \mathbf{D}_{EAS} , scheint in beiden Fällen zulässig, da sich nahezu exakt dieselben Ergebnisse einstellen. Ein Vergleich der Rechenzeiten beispielsweise für die VS08-EAS09 Formulierung zeigt:

VS08-EAS09	33670s	≅ 100%
VS08-EAS09d	31918s	≅ 95%
VS08-EAS09 (1)	18008s	≅ 53%
VS08-EAS09d (1)	16509s	≅ 49%

Tabelle 5.11.: Halbkugelschale mit Volumenschalenelementen – Numerischer Aufwand der Simulationen mit verschiedenen EAS-Elementen mit/ ohne Diagonalisierung der \mathbf{D}_{EAS} -Matrix sowie mit/ ohne Iteration der EAS-Parameter

Zwar ist der Aufwand für die lokale Kondensation geringer, jedoch erfordert die Diagonalisierung der Matrix selbst wieder zusätzliche Operationen, was insgesamt zu einer vergleichsweise geringen Einsparung der Rechenzeit führt. Überraschenderweise konnte bei den größeren Netzen sogar ein Anstieg der Rechenzeit bei der VS08-EAS09d Formulierung beobachtet werden, was auf eine Erhöhung der erforderlichen Iterationsschritte schließen lässt, was noch genauer untersucht werden sollte. Abbildung 5.20 zeigt weiter deutlich den Einfluss der Zahl der Iterationsschritte auf den Aufwand für die Kondensation der EAS-

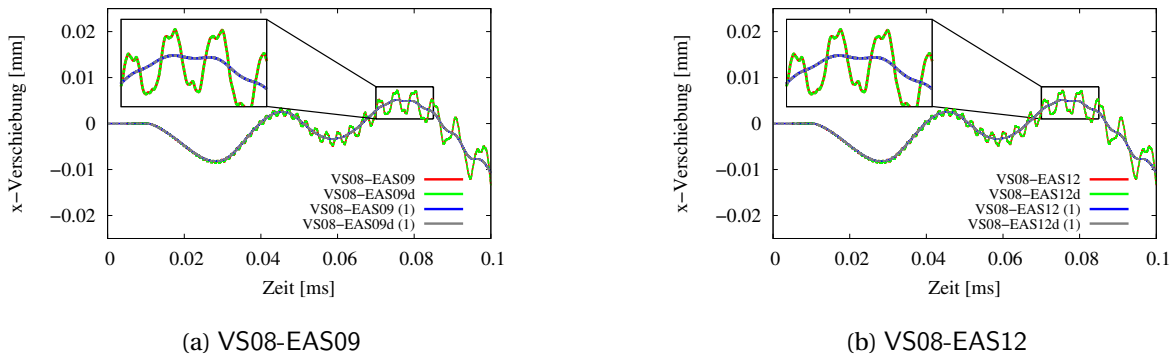


Abbildung 5.20.: Halbkugelschale mit Volumenschalenelementen – Vergleich der Systemantworten bei P bei diagonalisierter Matrix \mathbf{D}_{EAS} sowie einem festen Iterationsschritt für unterschiedliche EAS-Erweiterungen

Parameter. Offensichtlich lässt sich das globale Systemverhalten bereits mit einem einzigen Schritt abbilden, wobei allerdings höhere Frequenzen ‘verloren’ gehen.

Die Diskretisierung der Struktur mit Solid-Shell-Elementen mit quadratischer Interpolation von Geometrie und Verschiebung in der Schalenebene führte zu vergleichbaren Ergebnissen wie bei den degenerierten Schalenelementen, wie den Abbildungen 5.21 (a) und (b) zu entnehmen ist. Wegen des geringeren Einflusses geometrischer Versteifungseffekte weist bereits die reine Verschiebungsformulierung VS18-DISP ein deutlich erkennbares Konvergenzverhalten auf, sodass die numerisch vergleichsweise aufwändige Verwendung der ANS-Methode praktisch keine Verbesserung der Ergebnisqualität mehr zur Folge hat. Bei der feinsten Vernetzung mit 242 406 Freiheitsgraden liefern beide Formulierungen gleiche Ergebnisse, wobei das VS18-ANS Element diese auch schon bei größerem Netz abbilden kann, da hier geometrische Versteifungseffekte noch einen deutlicheren Einfluss haben.

Der Vergleich zwischen linearen und quadratischen Solid-Shell-Elementen ist in den Abbildungen 5.21 (c) und (d) dargestellt und verdeutlicht den Effekt der geometrischen Versteifungseffekte nochmals. Während die reine Verschiebungsformulierung mit 8 Knoten noch weitere Verfeinerungen der Vernetzung erfordert, können mit quadratischen Ansätzen auch ohne numerisch aufwändige Verzerrungsmodifikationen konvergierte Ergebnisse erzielt werden. Die Elementformulierungen mit ANS verhalten sich nahezu gleich. Insbesondere bei feiner Diskretisierung verschwindet der Einfluss der gekrümmten Struktur, da die sehr kleinen Elemente annähernd eben sind – daher unterscheiden sich die Ergebnisse kaum.

5.2.4. Vergleich beider Elementfamilien

Abbildung 5.22 enthält für die Halbkugelschale mit Loch den Vergleich der Ergebnisse der Berechnungen mit degenerierten Schalen- und mit Solid-Shell-Elementen jeweils für die feinste Diskretisierung. Es lässt sich erkennen, dass das globale Strukturverhalten von beiden Elementfamilien abgebildet werden kann. Die Ergebnisse unterscheiden sich hinsichtlich lokaler Effekte und höherer Frequenzen, was im Wesentlichen auf die geometrische Auflösung der Dickenrichtung bei den Volumenschalen zurückzuführen ist, welche bei den degenerierten Schalenelementen fehlt. Insbesondere bei der hier vorliegenden, sehr dünnen Struktur,

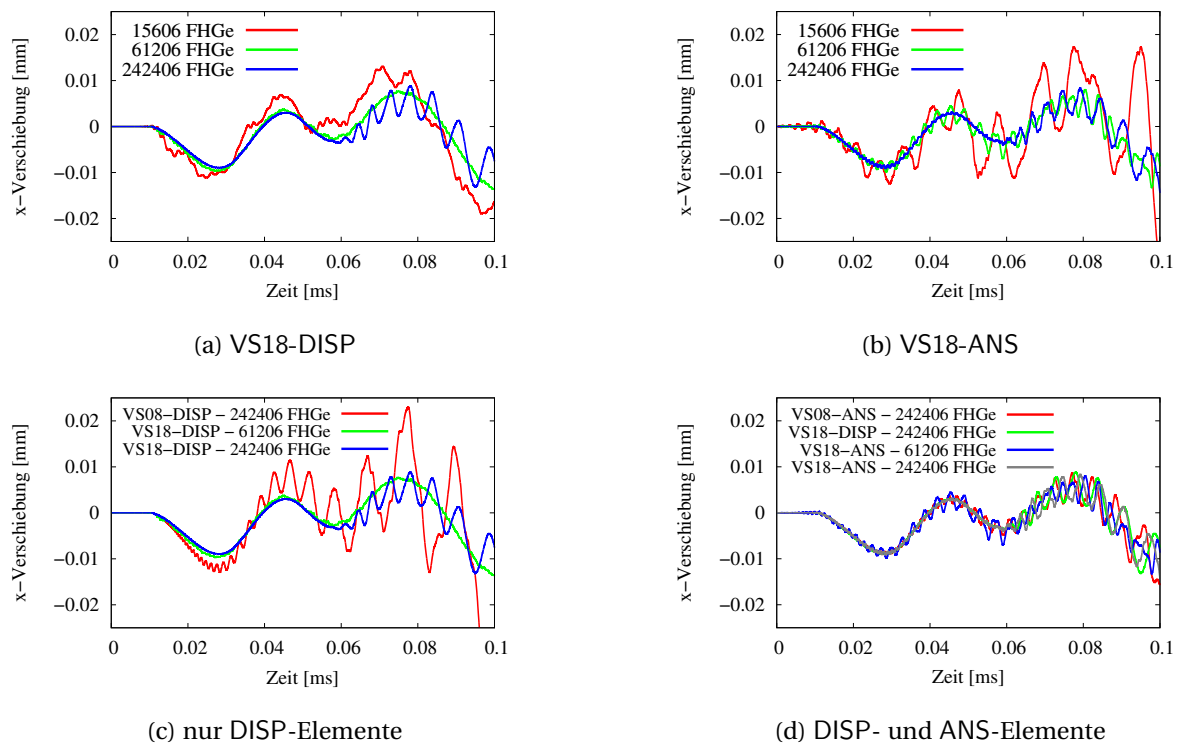


Abbildung 5.21.: Halbkugelschale mit Volumenschalenelementen – Vergleich der Systemantworten bei P für verschiedene quadratische Solid-Shell-Formulierungen sowie Vergleich zwischen linearen und quadratischen Solid-Shell-Elementen bei unterschiedlichen Netzfeinheiten

schlägt sich dies in einer wesentlich kleineren Zeitschrittweite bei den Solid-Shell-Elementen nieder, welche etwa bei der Verwendung mehrerer Elemente in Dickenrichtung noch weiter reduziert werden würde. Bei den Elementen mit linearen Ansätzen beispielsweise war der Zeitschritt für die degenerierten Schalenelemente selbst bei der feinsten Diskretisierung etwa doppelt so groß wie für die Volumenschalen. Für die Rechenzeiten bedeutet dies:

VS08-DISP	4329 s	≅ 100 %
VS08-ANS	5054 s	≅ 117 %
DS04-DISP	3775 s	≅ 87 %
DS04-ANS	4373 s	≅ 101 %

Tabelle 5.12.: Halbkugelschale mit Loch – Numerischer Aufwand für die Berechnungen mit degenerierten Schalen- und mit Solid-Shell-Elemente

Es fällt auf, dass sich die Rechenzeiten mit degenerierten Schalenelementen trotz größerem Zeitschritt und weniger Freiheitsgraden nur vergleichsweise gering von denen mit Solid-Shell-Elementen unterscheiden. Dies ist auf den größeren numerischen Aufwand auf Elementebene auf Grund des Degenerationskonzeptes zurückzuführen, der bereits in Abschnitt 4.4.2 diskutiert wurde. Bei sehr feinen Diskretisierungen, bei denen die Elemente annähernd Würfelform annehmen, relativiert sich die unterschiedliche Zeitschrittweite, sodass Volumenschalenelemente hinsichtlich Effizienz den degenerierte Schalenformulierungen überlegen sein können. Im Hinblick auf kommerzielle FE-Anwendungen, muss ein

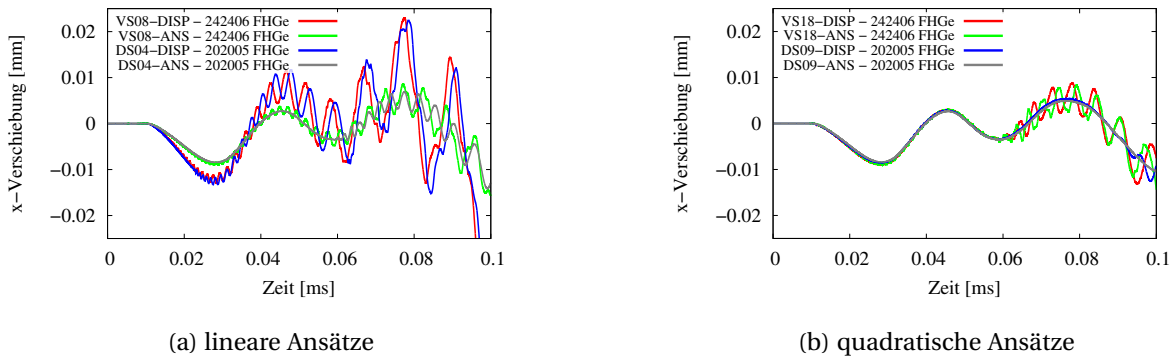


Abbildung 5.22.: Halbkugelschale mit Loch – Vergleich der Systemantworten bei degenerierten Schalen- und Solid-Shell-Elementen

solcher Effizienzvergleich kritisch betrachtet werden. Da die hier angebotenen Schalenelemente schon seit vielen Jahren verwendet und verbessert werden, sind diese sicherlich hinsichtlich Effizienz den hier vorgestellten Implementierungen noch überlegen.

Je nach Problemstellung und Diskretisierung ist also zu entscheiden, welche Elementfamilie besser geeignet ist. Obgleich im Bereich der degenerierten Schalenelemente auch komplexere Formulierungen mit entsprechenden Annahmen für die Dickenrichtung vorliegen, stellen die Volumenschalen insbesondere bei Fragestellungen, bei denen Dickeneffekte von Interesse sind, eine echte Alternative dar. Etwa bei der Simulation geschichteter Bauteile – im Bereich der Komposit-Werkstoffe – kann der Einsatz von Volumenschalenelementen zur Beurteilung von Dickenänderungen oder Delaminationen sehr sinnvoll sein.

5.3. Vollkugelschale mit Loch

5.3.1. Problembeschreibung und Diskretisierung

Ein typisches Anwendungsfeld für explizite Zeitintegration stellt die Simulation von Impaktproblemen dar. Als Beispiel wird die in Abbildung 5.23 dargestellte Vollkugelschale mit Loch untersucht. Nachdem sich die ungelagerte Struktur ursprünglich in Ruhe befindet, wird sie von einer sich mit der konstanten Geschwindigkeit von 80 m/s bewegenden, starren Ebene impaktiert. Im Abstand von 18 mm befindet sich eine weitere starre Ebene, die unbeweglich ist. Die obere Ebene stoppt ihre Bewegung nach 0.25 ms , sodass der Abstand beider Flächen konstant 82.1 mm beträgt. Die gesamte Simulationszeit beträgt 1.0 ms . Es treten insgesamt drei Kontakt Ereignisse im betrachteten Zeitraum auf: zuerst mit der verschieblichen Fläche, dann mit der unbeweglichen und schließlich wieder mit der ersten Fläche. Für die Diskretisierung wird die Symmetrie der Struktur bezüglich der $x-z$ -Ebene ausgenutzt. Die Hälfte der Kugel wird mit linearen bzw. quadratischen Solid-Shell-Elementen vernetzt; zusätzlich werden Kontaktelemente der entsprechenden Ansatzordnung auf die Kugeloberfläche gelegt, die in Abschnitt 3.5 beschrieben sind. Hierbei werden die äußeren Elementknoten der Strukturdiskretisierung verwendet. Als *Penalty*-Steifigkeit wird der Elastizitätsmodul der Struktur, also $\epsilon_N = 10 \cdot 10^3\text{ N/mm}^2$, gewählt. Die untersuchten Vernetzungen sind:

- $40 \times 10 \times 2$ 8-Knoten-Elemente, bzw. $20 \times 5 \times 2$ 18-Knoten-Elemente $\hat{=} 3960$ FHGe

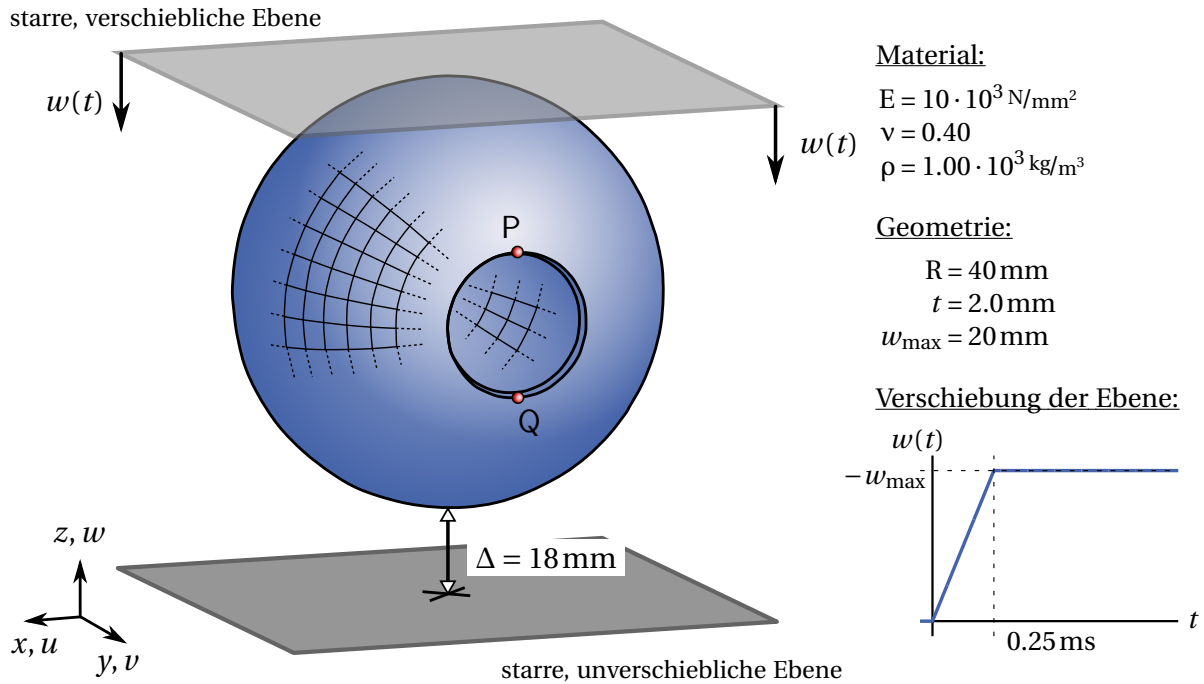


Abbildung 5.23.: Vollkugelschale mit Loch – Geometrie, Materialeigenschaften und Belastung

- $80 \times 20 \times 2$ 8-Knoten-Elemente, bzw. $40 \times 10 \times 2$ 18-Knoten-Elemente $\hat{=}$ 15 120 FHGe
- $160 \times 40 \times 2$ 8-Knoten-Elemente, bzw. $80 \times 20 \times 2$ 18-Knoten-Elemente $\hat{=}$ 59 040 FHGe

Es werden also immer zwei Elemente über die Dicke verwendet und lediglich die Netzfeinheit in Membranrichtung verändert. Um ein an Gummi angelehntes Materialverhalten zu erreichen, kommt das hyperelastische NEO-HOOKE-Materialgesetz mit einer Querkontraktionszahl von $\nu = 0.40$ zum Einsatz. So wird ein deutlicher Einfluss der volumetrischen Dickenversteifung erwartet, der sich unabhängig von der Netzfeinheit einstellen wird.

5.3.2. Diskussion der Ergebnisse bei elastischem Materialverhalten

Die Abbildungen 5.24, 5.25 und 5.26 zeigen das Verhalten der untersuchten Struktur für die verschiedenen Diskretisierungen und Elementformulierungen. Dabei sind jeweils fünf Zeitpunkte der Simulation visualisiert, wobei mit Hilfe der Farbe die resultierenden Verschiebungen dargestellt werden.

Es zeigt sich sehr deutlich der erwartete Einfluss der volumetrischen Versteifung und dessen Unabhängigkeit von der Netzfeinheit. Durch die Verfeinerung lassen sich geometrische Versteifungseffekte beheben, doch alle Elementformulierungen ohne EAS-, bzw. IM-Erweiterungen, weisen auch bei der feinsten Vernetzung ein deutlich zu steifes Verhalten auf. Auch die höhere Interpolationsordnung bei den VS18-Elementen kann ohne EAS-Erweiterung nicht zu einem weicherem Verhalten der Struktur führen.

In Abbildung 5.27 ist die Relativverschiebung zwischen den in Abbildung 5.23 markierten Punkten P und Q für die feinste Diskretisierung über die Zeit aufgetragen. Bei beiden Ansatzordnungen lässt die etwas niedrigere Grundfrequenz bei den EAS-Elementen auf

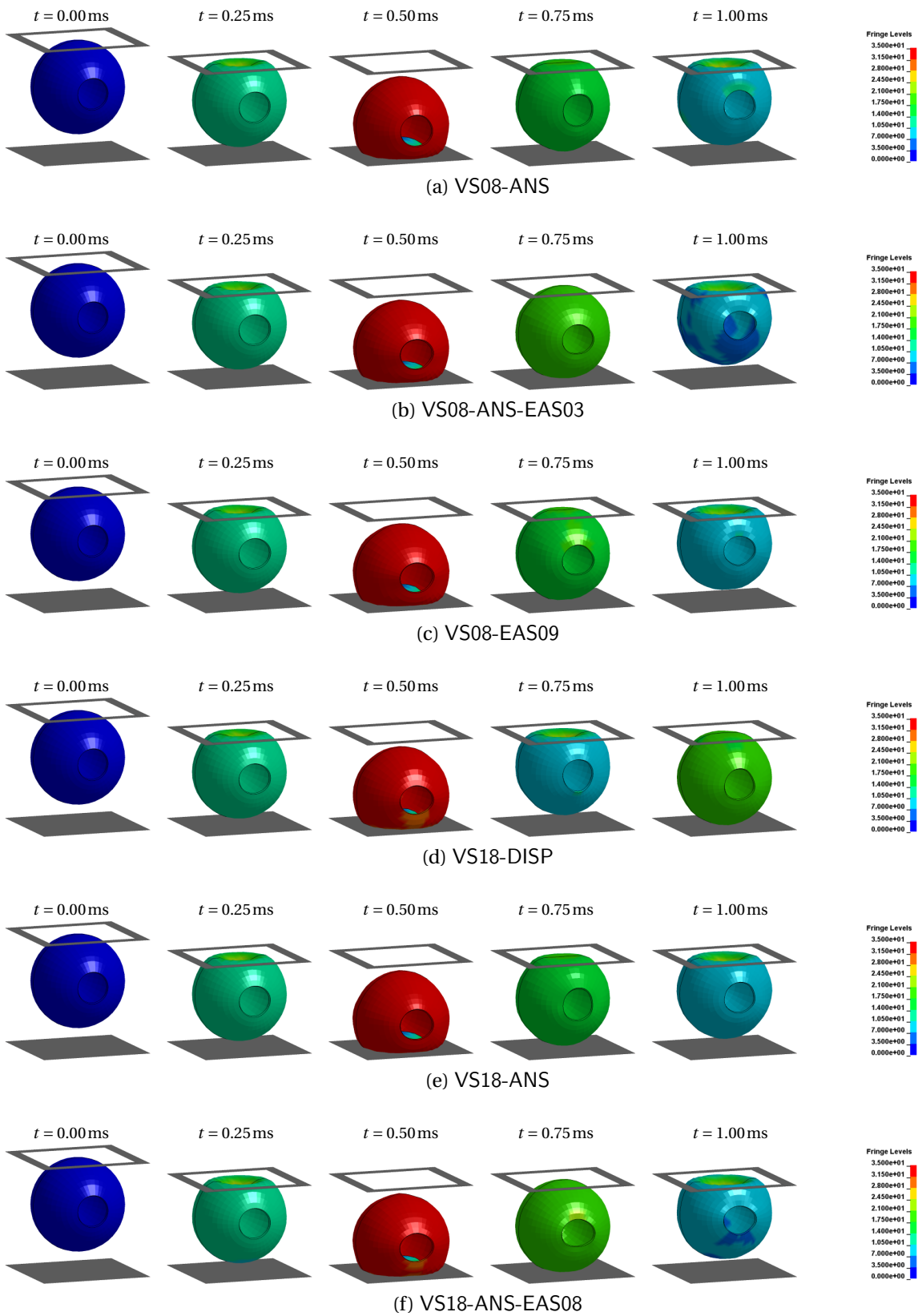


Abbildung 5.24.: Vollkugelschale mit Loch – 3960 FHGe – resultierende Verschiebungen

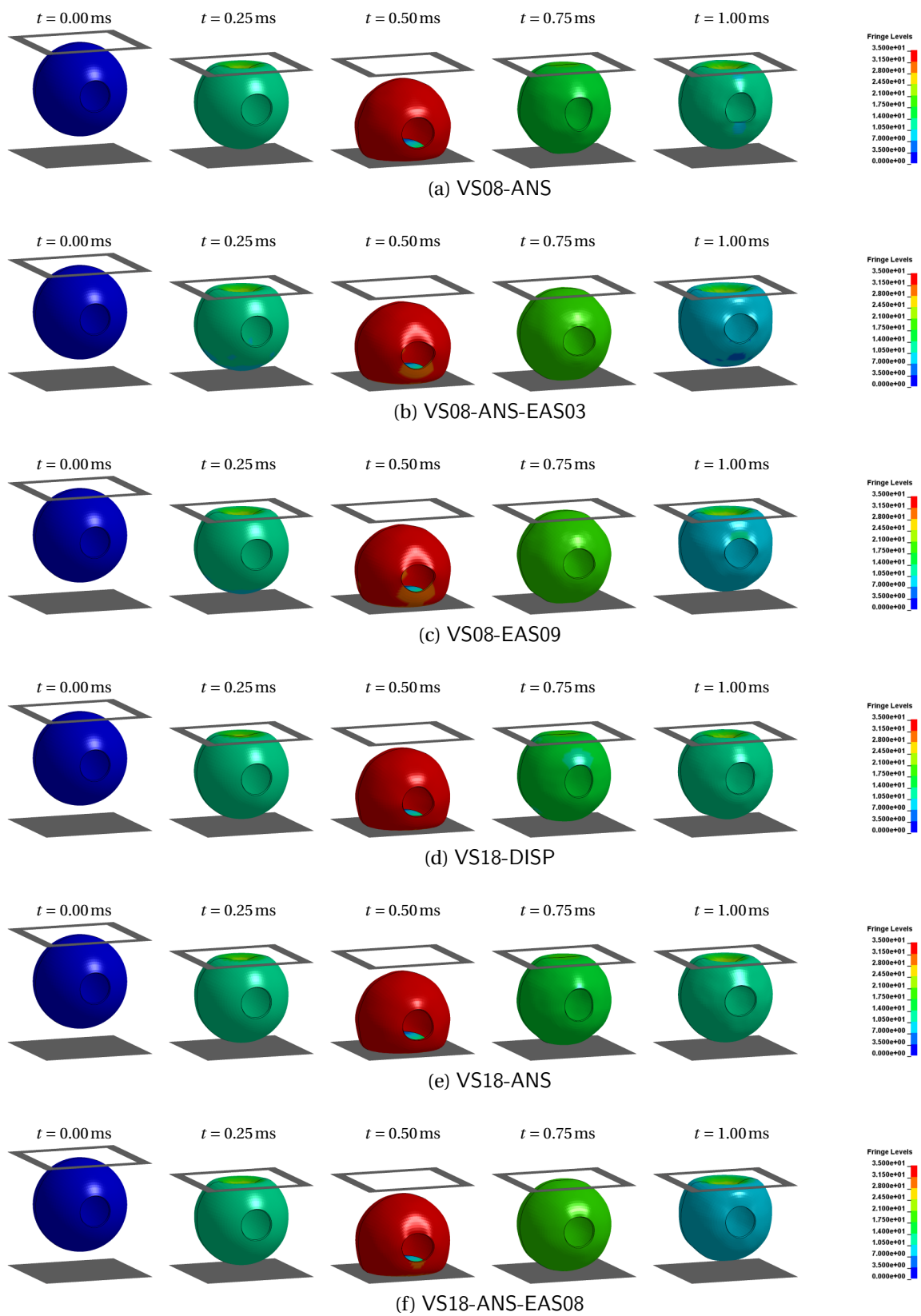


Abbildung 5.25.: Vollkugelschale mit Loch – 15 120 FHGe – resultierende Verschiebungen

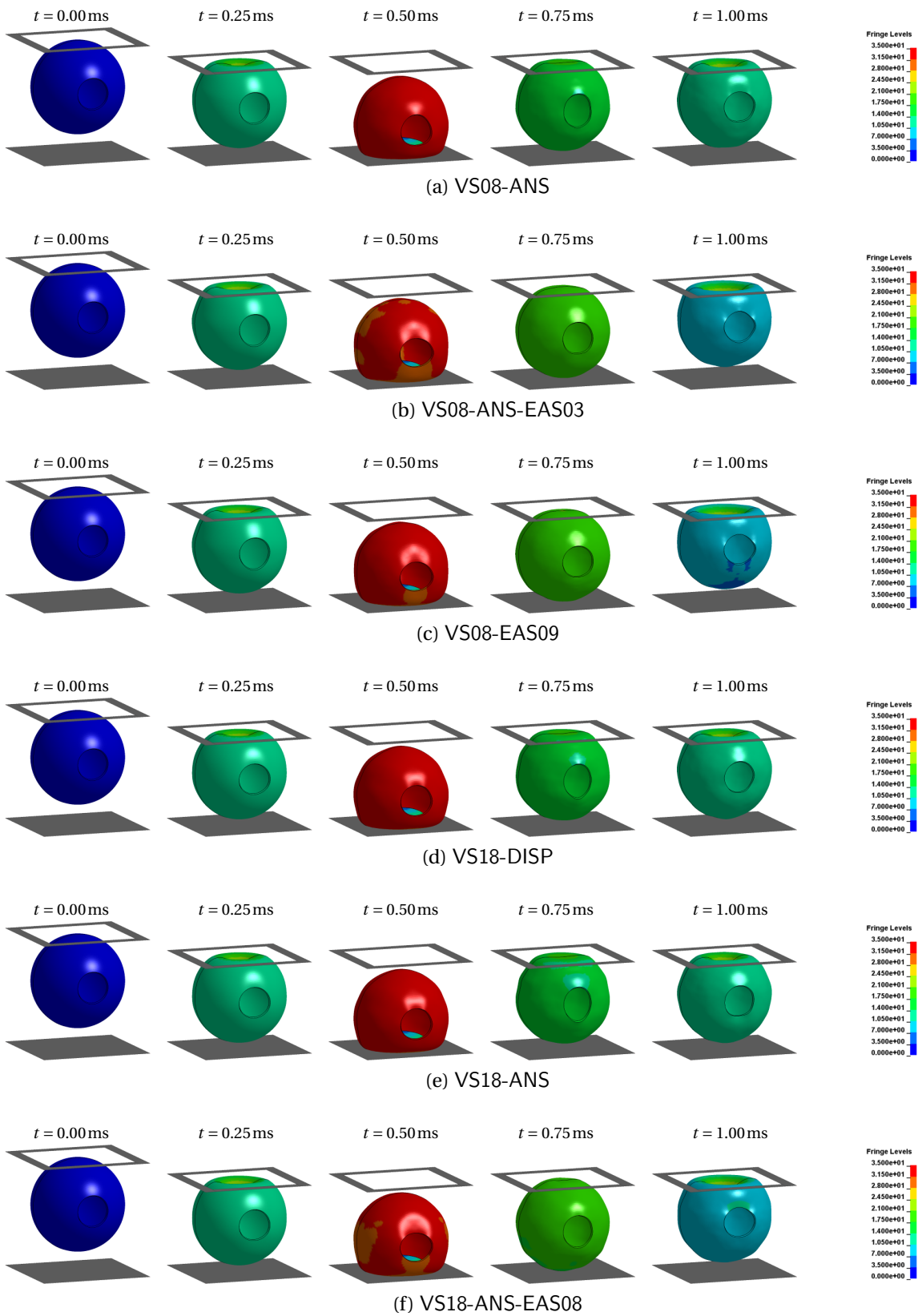


Abbildung 5.26.: Vollkugelschale mit Loch – 59040 FHGe – resultierende Verschiebungen

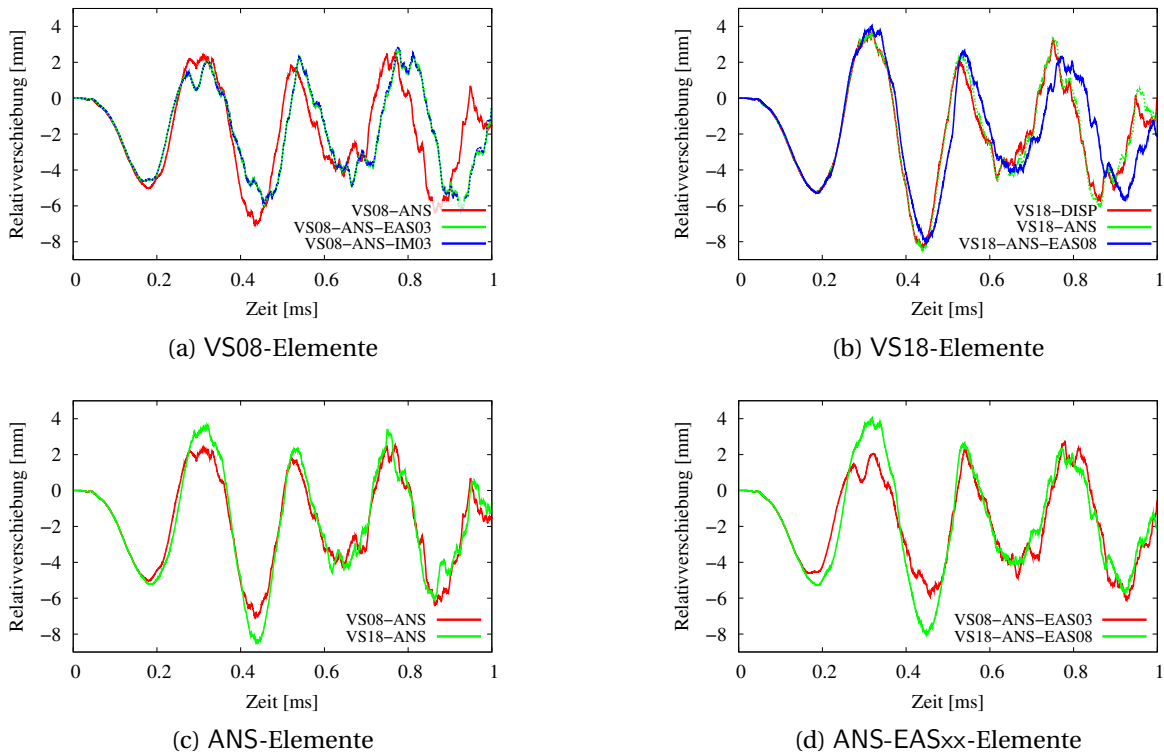


Abbildung 5.27.: Vollkugelschale mit Loch – Relativverschiebung der Punkte P und Q in z-Richtung für verschiedene Volumenschalenelemente

die Reduzierung der volumetrischen Versteifung schließen. Wie schon in den Beispielen zuvor tritt bei der quadratischen Interpolation für die untersuchte Netzfeinheit praktisch keine geometrische Versteifung auf, sodass die ANS-Modifikationen kaum eine Änderung des Systemverhaltens verursachen. Die Teilbilder (c) und (d) der Abbildung 5.27 zeigen, dass die Grundfrequenz mit linearen und quadratischen Ansätzen sowohl ohne als auch mit EAS jeweils gleich abgebildet werden. Unterschiede ergeben sich in der Amplitude und den höheren Frequenzen. Diese sind im globalen Verhalten aus den Abbildungen 5.24 – 5.26 jedoch kaum zu erkennen.

Es sei angemerkt, dass für die linearen Elemente wie bei den vorigen Beispielen die EAS-Erweiterungen durch entsprechende inkompatible Verschiebungsfreiheitsgrade ersetzt werden können. Die Systemantwort für eine Massenskalierung von $\alpha_m = 100.0$, bei der die gleiche Zeitschrittweite wie beim EAS-Element verwendet werden kann, ist in Abbildung 5.27 (a) angegeben und unterscheidet sich praktisch nicht von der VS08-ANS-EAS03 Formulierung. Wie zuvor stellt die IM-Methode innerhalb des expliziten Zeitintegrationsverfahrens eine interessante Alternative zu EAS dar, was insbesondere im Vergleich der erforderlichen Rechenzeiten deutlich wird:

VS08-ANS	923 s	≅ 100 %
VS08-ANS-EAS03	3521 s	≅ 381 %
VS08-ANS-IM03	1060 s	≅ 115 %

Tabelle 5.13.: Vollkugelschale mit Loch – Numerischer Aufwand der Simulation mit verschiedenen linearen Solid-Shell-Formulierungen (8 Knoten)

5.3.3. Diskussion der Ergebnisse mit Plastizität

Neben dem hyperelastischen NEO-HOOKE-Materialmodell, wurde für einige Elementtypen die in Abschnitt 1.3.2 beschriebene Formulierung für Plastizität mit finiten Deformationen umgesetzt. Für das oben beschriebene Beispiel wurde gemäß Gleichung (1.37) exemplarisch ein Verfestigungsgesetz

$$\sigma_y(\gamma^p) = \sigma_y^0 + c_1 \gamma^p + c_2 \left(1 - e^{-c_3 \gamma^p}\right)$$

mit

$$\sigma_y^0 = 200 \text{ N/mm}^2, \quad c_1 = 10.0 \text{ N/mm}^2, \quad c_2 = 1.00 \text{ N/mm}^2 \quad \text{und} \quad c_3 = 0.10$$

gewählt. Diese Materialparameter sind fiktiv gewählt und sollen somit kein reales Material repräsentieren. Neben den reinen Verschiebungsformulierungen VS08-DISP und VS18-DISP, wurde die Materialformulierung, wie in Abschnitt 4.4 beschrieben, auf die EAS-Formulierungen VS08-EAS09 und VS08-EAS09T angewendet, bei denen direkt ein erweiterter Deformationsgradient angegeben wird. Die Anwendung auf die ANS-Elemente mit Hilfe des in Abschnitt 3.4.2 beschriebenen Vorgehens zur Bestimmung des modifizierten Deformationsgradienten, konnte hier auf Grund der bereits erwähnten numerischen Probleme nicht angewendet werden. Selbst bei elastischem Materialverhalten konnten die Ergebnisse der VS08-ANS Formulierung, bei diesem von Nichtlinearitäten geprägten Beispiel, nicht reproduziert werden, da die Simulationen bereits nach wenigen Zeitschritten abbrechen. Eine robuste und insbesondere effiziente symbolische Implementierung zur Berechnung von Eigenvektoren, die für die Umsetzung des in Abschnitt 3.4.2 skizzierten Algorithmus erforderlich ist, liegt, auch nach Rücksprache mit dem ACEGEN-Entwickler, bislang nicht vor. Es sind weiterführende Überlegungen hinsichtlich der effizienten Anwendung der ANS-Methode erforderlich. Als eine Möglichkeit wurde in Abschnitt 3.4.3 die Modifikation der Verschiebungsinterpolation zur Reproduktion der ANS-Erweiterungen genannt, welche jedoch zunächst nur für kleine Deformationen formuliert werden konnte, weshalb auch hier die Umsetzung für Plastizität nicht durchgeführt wurde.

Zusätzlich wurde die Implementierung für die Serendipity-Elemente mit 16 Elementknoten VS16-DISP und VS16-DISPr durchgeführt. Die Diskretisierung erfolgte analog zur 18-Knoten-Formulierung, es ergeben sich jedoch wegen der fehlenden Mittelknoten entsprechend weniger Freiheitsgrade:

- $20 \times 5 \times 2$ 16-Knoten-Elemente $\hat{=}$ 3060 FHGe
- $40 \times 10 \times 2$ 16-Knoten-Elemente $\hat{=}$ 11520 FHGe
- $80 \times 20 \times 2$ 16-Knoten-Elemente $\hat{=}$ 44640 FHGe

Für die Diagonalisierung der Massenmatrix, kam bei den 16-Knoten-Elementen die in Abschnitt 3.3.3 beschriebene ‘*Special Lumping*’ Methode zum Einsatz, da zeilenweises Summieren, wie bereits erwähnt, zu negativen Diagonaltermen führte.

Eine grafische Aufbereitung der Ergebnisse für die verschiedenen Vernetzungen ist den Abbildungen 5.28 – 5.30 zu entnehmen. Es wurde lediglich die halbe Kugel visualisiert, um das Verhalten im Inneren der Struktur zeigen zu können. Als Ergebnisgröße für die Farbplots ist die akkumulierte plastische Dehnung als Maß für den Grad der plastischen Deformation dargestellt. Es lässt sich deutlich erkennen, dass sich die Ergebnisse der Elementformulierungen mit 9 EAS-Parametern bei Netzverfeinerung nur unwesentlich verändern. Lediglich an

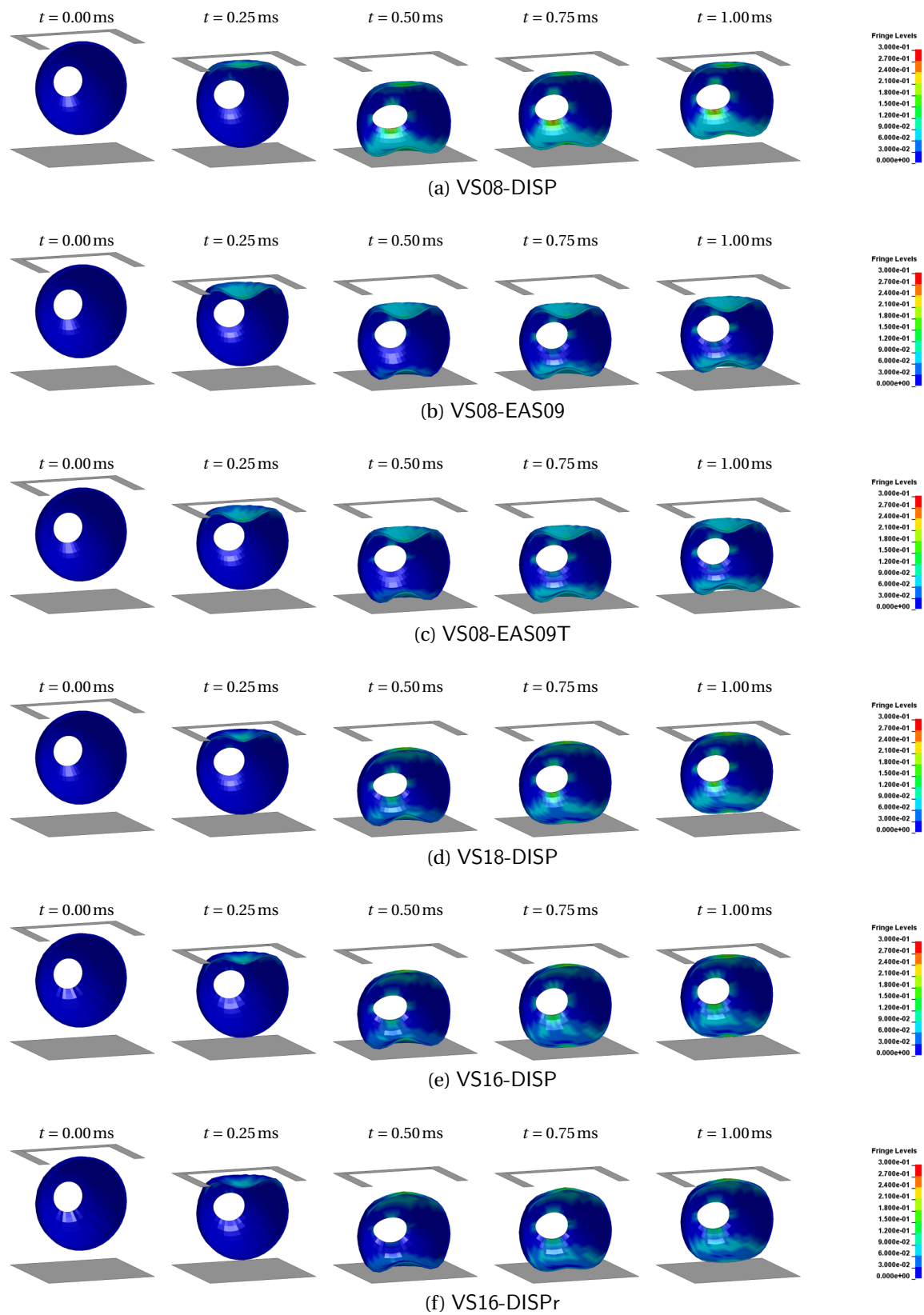


Abbildung 5.28.: Vollkugelschale mit Loch – Materialverhalten mit Plastizität bei grobe Ver-
netzung – akkumulierte plastische Dehnungen

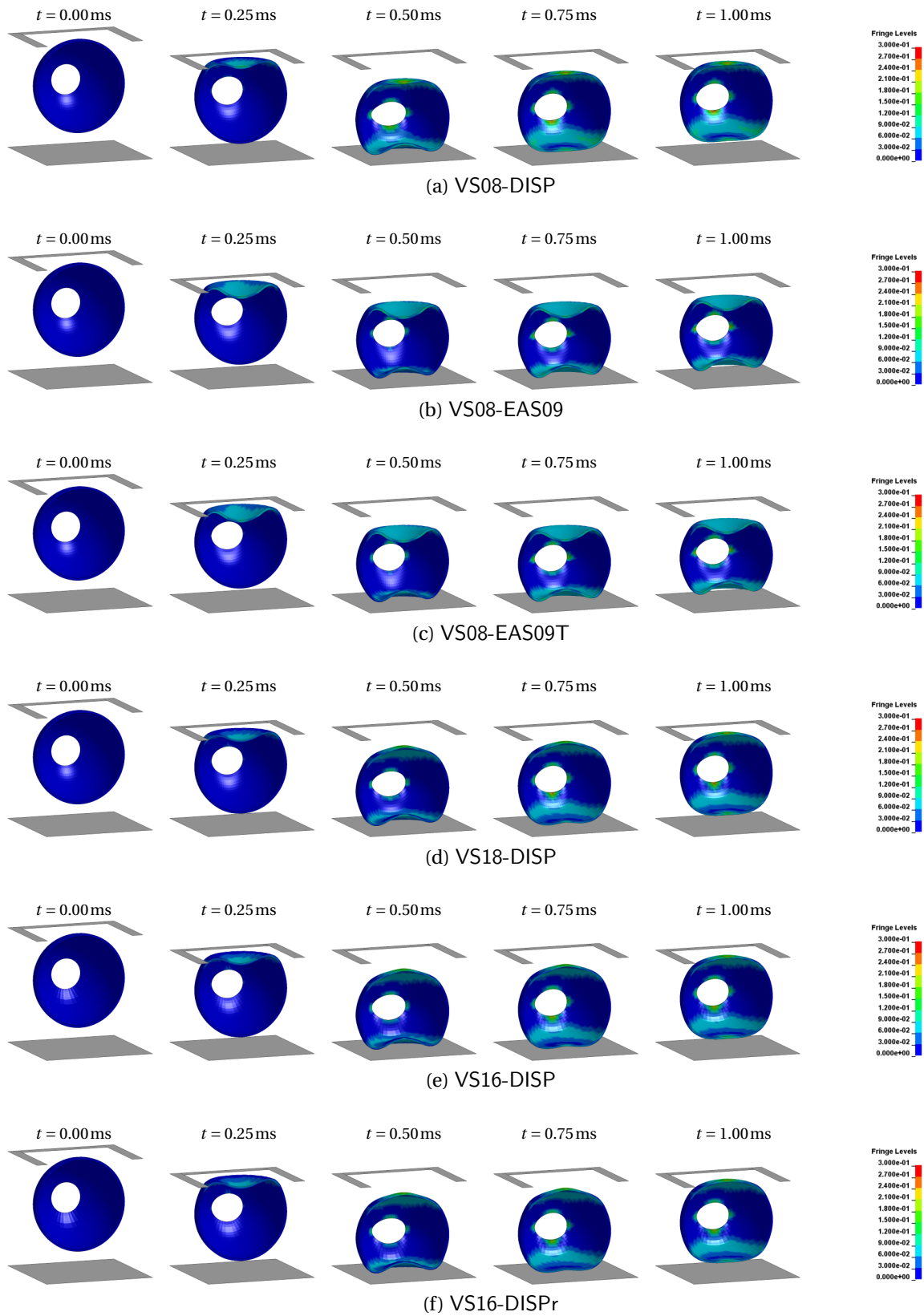


Abbildung 5.29.: Vollkugelschale mit Loch – Materialverhalten mit Plastizität bei mittlerer Vernetzung – akkumulierte plastische Dehnungen

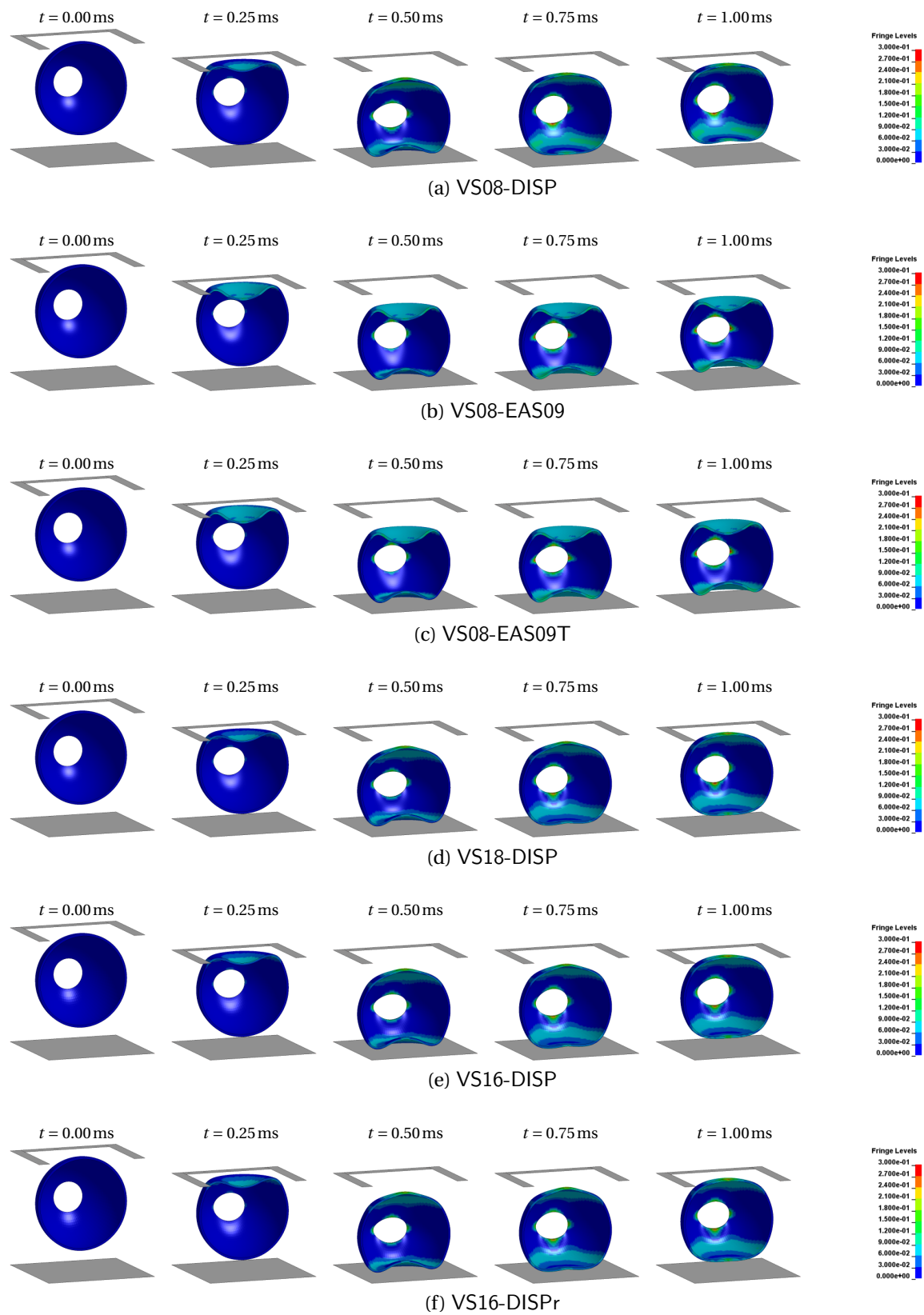


Abbildung 5.30.: Vollkugelschale mit Loch – Materialverhalten mit Plastizität bei feiner Ver-
netzung – akkumulierte plastische Dehnungen

mulierung auf. Bei genauerer Betrachtung zeigt sich bei einigen wenigen Elementen und nur in manchen Zeitschritten eine sehr große Zahl von Iterationsschritten für die Kondensation der EAS-Parameter, insbesondere in Bereichen mit starker Plastizität. Eine grafische Darstellung der erforderlichen Iterationen bei elastischem, bzw. elasto-plastischem Materialverhalten, ist exemplarisch für zwei Zeitschritte in Abbildung 5.32 angegeben. Offensichtlich tritt der volumetrische Versteifungseffekt durch die plastische Inkompressibilität der Materialformulierung hier besonders deutlich auf. Die Wahl des Abbruchkriteriums für die Kondensation, muss noch genauer untersucht werden; möglicherweise ist eine weniger konservative Schranke für die Änderung der EAS-Parameter ausreichend. Es zeigt sich erneut die Notwendigkeit einer alternativen Möglichkeit zur Behandlung der volumetrischen Versteifung, also beispielsweise eine Erweiterung der Methode der inkompatiblen Moden, die eine direkte Bestimmung des Deformationsgradienten erlaubt. Die quadratische Geometrie- und Verschiebungsinterpolation ist erwartungsgemäß etwas zeitaufwändiger im Vergleich zur linearen Formulierung. Bei gleicher Anzahl an Freiheitsgraden sind zwar insgesamt weniger Integrationspunkte vorhanden, jedoch wirkt sich die Verwendung quadratischer Ansätze scheinbar deutlicher auf die Rechenzeiten aus. Die Anwendung reduzierter Integration in Schalenebene, bei der die Anzahl der Integrationspunkte – und folglich als Nebeneffekt auch der Speicheraufwand für die lokalen Geschichtsvariablen – erheblich reduziert wird, führt zu einer sehr effizienten Formulierung. Die Verwendung quadratischer Interpolationsfunktionen in Kombination mit einer gezielten Behandlung der volumetrischen Versteifung – etwa durch einige wenige EAS-Erweiterungen – scheint also eine sinnvolle Alternative zu den linearen Elementen zu sein.

Zur Bewertung des durch die Materialformulierung verursachten numerischen Aufwandes, werden im Folgenden die Simulationszeiten mit elastischem und mit plastischem Materialverhalten verglichen. Für die jeweils feinste Diskretisierung sind diese:










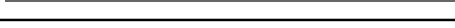
VS08-DISP	elast.	 2276s	≅ 100 %
	plast.	 3795s	≅ 167 %
VS08-EAS09	elast.	 22859s	≅ 100 %
	plast.	 30691s	≅ 134 %
VS18-DISP	elast.	 3879s	≅ 100 %
	plast.	 5524s	≅ 142 %
VS16-DISP	elast.	 3400s	≅ 100 %
	plast.	 4943s	≅ 148 %
VS16-DISPr	elast.	 2158s	≅ 100 %
	plast.	 3078s	≅ 143 %

Tabelle 5.15.: Vollkugelschale mit Loch – Numerischer Aufwand der Simulation mit elastischem und plastischem Materialverhalten

Es zeigt sich bei allen Elementformulierung ein ähnliches Verhältnis der Rechenzeiten; der numerische Aufwand der Materialroutine kann hier also recht gut abgelesen werden. Überraschenderweise fällt der Unterschied bei der VS08-EAS09 Formulierung mit und ohne Plastizität weniger deutlich aus als erwartet. Der in Abbildung 5.32 dargestellte Anstieg der erforderlichen Iterationsschritte in einigen wenigen Elementen und Zeitschritten, scheint sich also nicht signifikant auszuwirken. Hier kann allerdings auch die ungenaue Zeiterfas-

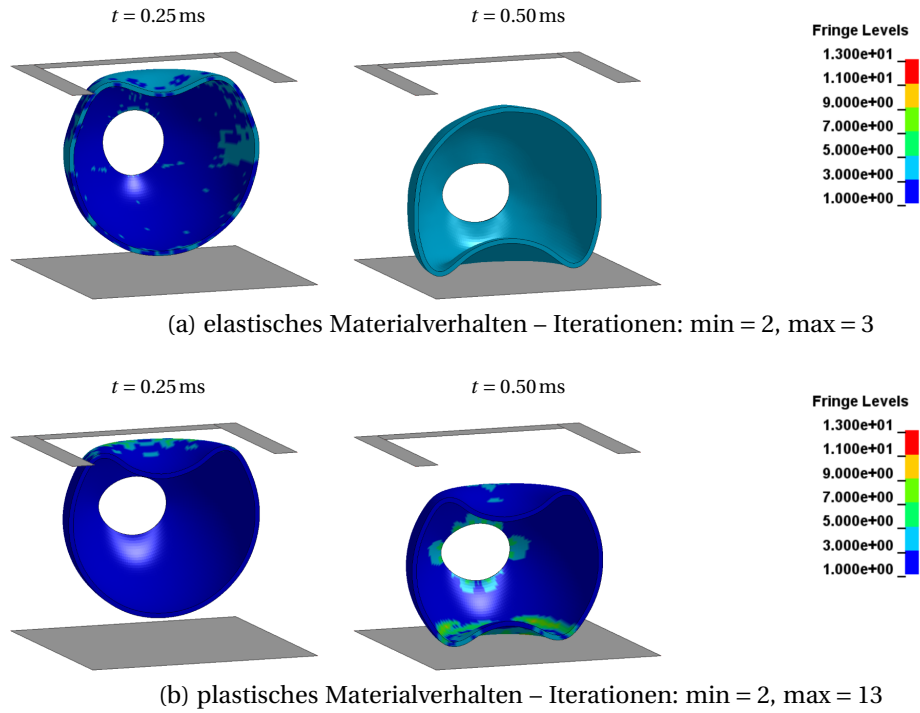


Abbildung 5.32.: Vollkugelschale mit Loch – Farbliche Darstellung der Anzahl erforderlicher Iterationsschritte zur Kondensation der EAS-Parameter bei elastischem und elasto-plastischem Materialverhalten

sung eine Rolle spielen, da die Simulationen auf PCs durchgeführt wurden, weitere parallel laufende störende Prozesse – insbesondere bei sehr langen Simulationen – also nicht auszuschließen sind.

Zusammenfassung und Ausblick

Die vorliegende Arbeit beschäftigt sich mit der Anwendung des symbolischen Programmierwerkzeugs ACEGEN für die Entwicklung und Umsetzung von Schalen- und Volumenschalenelementen im Zusammenhang mit expliziter Zeitintegration. Ziel des vorgestellten Konzeptes ist die Steigerung der *Effizienz*, deren Definition sich im Wesentlichen durch zwei Punkte beschreiben lässt:

1. Aufwand für die Programmierung
2. Rechenaufwand

Der erste Punkt bezieht sich auf die programmiertechnische Umsetzung einer theoretisch hergeleiteten Methode einschließlich deren algorithmischer Aufbereitung und Implementierung in einer Programmiersprache. Dies kann durch die Anwendung von ACEGEN erheblich unterstützt werden, da dessen symbolische Eigenschaften eine Umsetzung erlauben, die den Schreibweisen der theoretischen Zusammenhänge sehr ähnlich ist. So müssen Matrix- und Tensoroperationen nicht in Programmschleifen übersetzt und Ableitungen nicht vorab ausgewertet und deren Ergebnisse implementiert werden. Dies erlaubt zum Einen eine sehr übersichtliche Programmierung, da diese kaum von den theoretisch hergeleiteten Formeln abweichen muss. Fehler, die bei der Aufbereitung des Formelapparates für die jeweilige Programmiersprache entstehen, können bei entsprechender Anwendung der ACEGEN-Syntax damit praktisch ausgeschlossen werden. Ebenso können triviale Tippfehler vermieden werden, deren Suche häufig sehr zeitaufwändig ist. Zum Anderen werden die automatisch erzeugten Programmroutinen durch ACEGEN hinsichtlich der durchgeführten Rechenoperationen optimiert, was einen sehr kompakten und effizienten Programmcode zur Folge hat.

Die Anwendung des Programmierkonzeptes wird innerhalb der Arbeit an verschiedenen Beispielen beschrieben. Der Schwerpunkt liegt hierbei auf der Elementtechnologie, also der Umsetzung von Methoden zur Beseitigung ungewollter Versteifungseffekte, welche teilweise im Schrifttum bereits für verschiedene – überwiegend implizite – Anwendungen getestet wurden. Die Konzepte basieren in der Regel auf einer Modifikation von Verschiebungsannahmen oder Verzerrungs-, bzw. Spannungsverläufen, die sich dann auf alle weiteren Rechenschritte innerhalb der Elementroutine auswirken. Hier erlaubt die symbolische Programmierung eine schnelle und fehlerfreie Umsetzung von Varianten, da etwa eine Auswertung und Interpolation von Verzerrungen symbolisch erfolgen kann und die sich daraus ergebenden Elementmatrizen automatisch ermittelt werden. Als weitere Anwendung ist die Umsetzung von Materialmodellen genannt, welche am Beispiel einer Plastizitätsformulierung gezeigt wird. Hier macht sich insbesondere der Vorteil der automatischen Differentiation bemerkbar, welcher beispielsweise bei der lokalen Lösung nichtlinearer Gleichungen zum Einsatz kommt.

Die Effizienzsteigerung der umgesetzten Methoden selbst – also der zweite Punkt der genannten Definition – bezieht sich auf die erforderliche Zeit, die benötigt wird, um mit

einer Implementierung Ergebnisse zu erhalten. Dieser Aspekt macht sich insbesondere im Zusammenhang mit dem verwendeten expliziten Zeitintegrationsverfahren bemerkbar, welches durch kleine Zeitschritte und numerisch günstige globale Operationen charakterisiert ist. Hier haben die Operationen auf Elementebene einen wesentlichen Anteil an der Gesamtrechnzeit, weshalb deren effiziente Umsetzung von besonderem Interesse ist. Dies kann an einem Prinzipbeispiel, an dem händisch programmierter mit ACEGEN Code verglichen wird, eindrucksvoll gezeigt werden. Selbstverständlich ist eine Verbesserung der händischen Umsetzung, etwa durch Vermeidung redundanter Rechenoperationen möglich, diese steigert jedoch die Zeit, die für die Programmierung erforderlich ist und wirkt sich somit wieder auf den ersten Punkt der genannten Definition aus. Durch die automatisierte Codeerzeugung sind keine weiteren Arbeitsschritte des Programmierers zur Effizienzsteigerung erforderlich. Im Bereich der Forschung und Entwicklung numerischer Modelle, gerade wenn eine entwickelte Methode möglichst schnell an Beispielen getestet werden soll, ist dies von besonderem Interesse.

Im Rahmen der Arbeit werden neben bekannten Methoden zur Beseitigung von Versteifungseffekten Alternativen diskutiert, deren Umsetzung durch die Anwendung von ACEGEN unterstützt werden. Insbesondere innerhalb des expliziten Zeitintegrationsverfahrens kann die Methode der erweiterten Verzerrungen (EAS) wegen der erforderlichen lokalen Kondensation zusätzlicher Unbekannter den numerischen Aufwand der Elementformulierung erheblich steigern. Hier wird die Methode der inkompatiblen Moden als vielversprechende Alternative vorgeschlagen, bei der die Erweiterung der Verzerrungsverläufe durch Einführung zusätzlicher Verschiebungsfreiheitsgrade erreicht wird. Wegen der Behandlung der zusätzlichen Freiheitsgrade als globale Unbekannte können so die effizienten Beziehungen des zentralen Differenzenverfahrens genutzt werden. Als Alternative zur Methode der angenommenen Verzerrungen (ANS) wird ebenfalls eine entsprechende Modifikation der Verschiebungsinterpolation diskutiert. Hier können – zumindest für kleine Deformationen – die angenommenen Verzerrungsverläufe der ANS-Methode reproduziert werden. Änderungen an der Verschiebungsinterpolation bedeuten üblicherweise bei händischer Umsetzung einen sehr hohen Programmieraufwand, da bei Modifikationen sämtliche Elementmatrizen neu bestimmt und implementiert werden müssen. Hier zahlte sich die symbolische Programmierung insbesondere aus.

Weitere Untersuchungen im Bereich der Elementtechnologie – insbesondere unter Anwendung der symbolischen Programmierung – können durch die vorliegende Arbeit motiviert werden. Die Anwendung der Methode der inkompatiblen Moden stellt offensichtlich im Zusammenhang mit expliziter Zeitintegration eine attraktive Alternative zu den bekannten EAS-Formulierungen dar. Weitere Überlegungen hinsichtlich der Verwendbarkeit mit Materialformulierungen, die Plastizität mit großen Deformationen einschließen, sind allerdings erforderlich, da dann ein erweiterter Deformationsgradient angegeben werden muss. Auch die Repräsentation der den zusätzlichen Freiheitsgraden zugeordneten Massen, deren Einfluss schon an verschiedenen Beispielen diskutiert wurde, sollte noch weiter untersucht werden. Eine effiziente Anwendung der ANS-Methode mit allgemeinen Materialformulierungen führte im Zusammenhang mit der hier verwendeten Zeitintegration bislang zu unbefriedigenden Ergebnissen, da die Rückrechnung des veränderten Deformationsgradienten für ein explizites Verfahren numerisch aufwändig und sich im vorliegenden Fall als wenig robust erwies. Hier sollte der Übergang auf eine inkrementelle Formulierung günstiger sein, wie sie in kommerziellen Anwendungen expliziter Zeitintegrationsverfahren

üblich ist. Alternativ würde eine Erweiterung der XDIS-Elemente auf große Deformationen – sofern sich diese allgemein formulieren lässt – insbesondere im Zusammenhang mit ACEGEN einen Zugang zu einem versteifungsfreien Deformationsgradienten darstellen. Dieser ließe sich durch symbolische Ableitung direkt aus einer entsprechend modifizierten Verschiebungsinterpolation bestimmen.

Untersuchungen im Bereich der Materialmodelle sind in der vorliegenden Arbeit exemplarisch dargestellt, um die Vorteile und Möglichkeiten der symbolischen Programmierung aufzuzeigen. Der Bereich der Formulierung von Materialmodellen, insbesondere unter Betrachtung verschiedener Skalen, stellt derzeit einen wesentlichen Forschungsaspekt auf dem Gebiet der computergestützten Mechanik dar. Auch hier sollte sich eine intensive Anwendung symbolischer Programmierung zur Umsetzung entwickelter Modelle langfristig bezahlt machen. Ebenso ist die Weiterführung der Überlegungen auf dem Gebiet der Kontaktmechanik unter Verwendung von ACEGEN vielversprechend.

Zusammenfassend scheint die konsequente Anwendung symbolischer Programmierung ein sinnvoller Weg in Kombination mit Forschungsarbeiten auf dem Gebiet der computergestützten Mechanik zu sein. Zunächst erfordert dies zwar eine gewisse Einarbeitungszeit, um durch korrekte Anwendung der ACEGEN-Syntax die daraus entstehenden Vorteile gegenüber händischer Programmierung optimal nutzen zu können. Letztlich sollte sich jedoch die anfangs ungewohnte Form der Programmierung auszahlen, da beispielsweise Modifikationen und Erweiterungen bestehender Formulierungen dann mit sehr geringem Aufwand und praktisch fehlerfrei realisiert werden können. Somit verkürzt sich sowohl die Zeit, bis ein ‘*Testen*’ der Methode möglich ist, als auch – durch die Optimierung der automatisch erzeugten Programmerroutinen – die Durchführung der Testbeispiele selbst. Die Effizienz von Forschungsprozessen kann mit symbolischer Programmierung hinsichtlich beider oben genannter Punkte deutlich gesteigert werden.

Schrifttum

- [1] AHMAD, S., B.M. IRONS und O.C. ZIENKIEWICZ: *Analysis of thick and thin shell structures by curved finite elements*. International Journal for Numerical Methods in Engineering, 2(3):419–451, 1970.
- [2] ALMEIDA AREIAS, P.M., J.M.A. CÉSAR DE SÁ, C.A. CONCEIÇÃO ANTÓNIO und A.A. FERNANDES: *Analysis of 3D problems using a new enhanced strain hexahedral element*. International Journal for Numerical Methods in Engineering, 58(11):1637–1682, 2003.
- [3] ALMEIDA AREIAS, P.M., J.H. SONG und T. BELYTSCHKO: *A finite-strain quadrilateral shell element based on discrete Kirchhoff-Love constraints*. International Journal for Numerical Methods in Engineering, 64(9):1166–1206, 2005.
- [4] ALVES DE SOUSA, R.J., R.P.R. CARDOSO, R.A. FONTES VALENTE, J.-W. YOON, R.M. NATAL JORGE und J.J. GRACIO: *A new one-point quadrature enhanced assumed strain (EAS) solid-shell element with multiple integration points along thickness: Part I - geometrically linear applications*. International Journal for Numerical Methods in Engineering, 62(7):952–977, 2005.
- [5] ALVES DE SOUSA, R.J., R.M. NATAL JORGE, R.A. FONTES VALENTE und J.M.A. CESAR DE SÁ: *A new volumetric and shear locking-free 3D enhanced strain element*. Engineering Computations, 20:896–925, 2003.
- [6] ALVES DE SOUSA, R.J., J.-W. YOON, R.P.R. CARDOSO, R.A. FONTES VALENTE und J.J. GRACIO: *On the use of a reduced enhanced solid-shell (RESS) element for sheet forming simulations*. International Journal of Plasticity, 23(3):490–515, 2007.
- [7] ANDELFINGER, U.: *Untersuchungen zur Zuverlässigkeit hybrid-gemischter Finiter Elemente für Flächentragwerke*. Dissertation, Institut für Baustatik, Universität Stuttgart, 1991.
- [8] ANDELFINGER, U. und E. RAMM: *EAS-elements for two-dimensional, three-dimensional, plate and shell structures and their equivalence to HR-elements*. International Journal for Numerical Methods in Engineering, 36(8):1311–1337, 1993.
- [9] ANDERSON JR., C.E.: *An Overview of the Theory of Hydrocodes*. International Journal of Impact Engineering, 5:33 – 59, 1987.
- [10] ANSYS, INC.: ANSYS, 2012. <http://www.ansys.com>.
- [11] ARGYRIS, J.: *An Excursion into Large Rotations*. Computer Methods in Applied Mechanics and Engineering, 32(1-3):85–105, 1982.
- [12] BATHE, K.-J.: *Finite-Elemente-Methoden*. Springer, 2. Auflage, 2002.
- [13] BATHE, K.-J. und E.N. DVORKIN: *A formulation of general shell elements – the use of mixed interpolation of tensorial components*. International Journal for Numerical Methods in Engineering, 22(3):697–722, 1986.

- [14] BELYTSCHKO, T. und L.P. BINDEMAN: *Assumed Strain Stabilization of the Eight Node Hexahedral Element*. Computer Methods in Applied Mechanics and Engineering, 105(2):225–260, 1993.
- [15] BELYTSCHKO, T., W.K. LIU und B. MORAN: *Nonlinear Finite Elements for Continua and Structures*. Wiley, 2000.
- [16] BELYTSCHKO, T. und C.S. TSAY: *Explicit Algorithms for Nonlinear Dynamics of Shells*. Computer Methods in Applied Mechanics and Engineering, 43:251–276, 1984.
- [17] BENSON, D.J.: *Stable time step estimation for multi-material Eulerian hydrocodes*. Computer Methods in Applied Mechanics and Engineering, 167(1-2):191–205, 1998.
- [18] BETSCH, P. und E. STEIN: *An assumed strain approach avoiding artificial thickness straining for an non-linear 4-node shell element*. Communications in Numerical Methods in Engineering, 11:899–909, 1995.
- [19] BISCHOFF, M.: *Theorie und Numerik einer dreidimensionalen Schalenformulierung*. Dissertation, Institut für Baustatik, Universität Stuttgart, 1999.
- [20] BISCHOFF, M. und K.-U. BLETZINGER: *Stabilized DSG Plate and Shell Elements*. In: WALL, W.A., K.-U. BLETZINGER und K. SCHWEIZERHOF (Herausgeber): *Trends in Computational Structural Mechanics*, Seiten 253 – 263, 2001.
- [21] BISCHOFF, M. und K.-U. BLETZINGER: *Improving stability and accuracy of Reissner–Mindlin plate finite elements via algebraic subgrid scale stabilization*. Computer Methods in Applied Mechanics and Engineering, 193(15–16):1517–1528, 2004.
- [22] BISCHOFF, M., F. KOSCHNICK und K.-U. BLETZINGER: *Stabilized DSG Elements – A New Paradigm in Finite Element Technology*. In: *Proceedings of the 4th European LS-DYNA Users Conference*, Seiten H–I–49–H–I–62, 2003.
- [23] BISCHOFF, M. und E. RAMM: *Shear deformable shell elements for large strains and rotations*. International Journal for Numerical Methods in Engineering, 40(23):4427–4449, 1997.
- [24] BISCHOFF, M. und I. ROMERO: *A generalization of the method of incompatible modes*. International Journal for Numerical Methods in Engineering, 69(9):1851–1868, 2007.
- [25] BISCHOFF, M., W.A. WALL, K.-U. BLETZINGER und E. RAMM: *Models and Finite Elements for Thin-Walled Structures*. Encyclopedia of Computational Mechanics, 2:59 – 137, 2004.
- [26] BLETZINGER, K.-U., M. BISCHOFF und E. RAMM: *A unified approach for shear-locking-free triangular and rectangular shell finite elements*. Computers & Structures, 75:321–334, 2000.
- [27] BONET, J. und R.D. WOOD: *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, 2008.
- [28] BRAESS, D.: *Finite Elemente: Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*. Springer, 1992.
- [29] BUCALEM, M.L. und K.-J. BATHE: *Higher-order MITC general shell elements*. International Journal for Numerical Methods in Engineering, 36(21):3729–3754, 1993.
- [30] BÜCHTER, N., E. RAMM und D. ROEHL: *Three-dimensional extension of non-linear shell formulation based on the enhanced assumed strain concept*. International Journal for Numerical Methods in Engineering, 37(15):2551–2568, 1994.

- [31] CARDOSO, R.P.R. und J.-W. YOON: *One point quadrature shell elements for sheet metal forming analysis*. Archives of Computational Methods in Engineering, 12(1):3–66, 2005.
- [32] CARDOSO, R.P.R., J.-W. YOON, J.J. GRÁCIO, F. BARLAT und J.M.A. CÉSAR DE SÁ: *Development of a one point quadrature shell element for nonlinear applications with contact and anisotropy*. Computer Methods in Applied Mechanics and Engineering, 191(45):5177–5206, 2002.
- [33] CARDOSO, R.P.R., J.W. YOON, M. MAHARDIKA, S. CHOUDHRY, R.J. ALVES DE SOUSA und R.A. FONTES VALENTE: *Enhanced assumed strain (EAS) and assumed natural strain (ANS) methods for one-point quadrature solid-shell elements*. International Journal for Numerical Methods in Engineering, 75(2):156–187, 2008.
- [34] CLOUGH, R.W. und J. PENZIEN: *Dynamics of structures*. McGraw-Hill, 2. Auflage, 1993.
- [35] COOK, R.D., D.S. MALKUS und M.E. PLESHA: *Concepts and Applications of Finite Element Analysis*. Wiley, 1989.
- [36] COURANT, R., K.O. FRIEDRICHS und H. LEWY: *Über die partiellen Differenzgleichungen der mathematischen Physik*. Mathematische Annalen, 100:32–74, 1928.
- [37] DE SOUZA NETO, E.A., D. PERIĆ und D.R.J. OWEN: *Computational methods for plasticity: theory and applications*. Wiley, 2008.
- [38] DOLL, S.: *Zur numerischen Behandlung großer elasto-viskoplastischer Deformationen bei isochor-volumetrisch enkoppeltem Stoffverhalten*. Dissertation, Institut für Mechanik, Universität Karlsruhe (TH), 1998.
- [39] DOLL, S., K. SCHWEIZERHOF, R. HAUPTMANN und C. FREISCHLÄGER: *On volumetric locking of low-order solid and solid-shell elements for finite elastoviscoplastic deformations and selective reduced integration*. Engineering Computations, 17(7):874–902, 2000.
- [40] DVORKIN, E.N., D. PANTUSO und E.A. REPETTO: *A formulation of the MITC4 shell element for finite strain elasto-plastic analysis*. Computer Methods in Applied Mechanics and Engineering, 125(1–4):17–40, 1995.
- [41] EBERLEIN, R.: *Finite-Elemente-Konzepte für Schalen mit großen elastischen und plastischen Deformationen*. Dissertation, Institut für Mechanik, Technische Hochschule Darmstadt, 1997.
- [42] ESI GROUP: *PAM-CRASH*, 2012. <http://www.esi-group.com>.
- [43] FELIPPA, C.A.: *Introduction to Finite Element Methods*. Herbst 2004. Aerospace Engineering Sciences Department of the University of Colorado at Boulder, 1986.
- [44] FONTES VALENTE, R.A., R.J. ALVES DE SOUSA und R.M. NATAL JORGE: *An enhanced strain 3D element for large deformation elastoplastic thin-shell applications*. Computational Mechanics, 34:38–52, 2004.
- [45] FREISCHLÄGER, CH.: *Konzepte zur Formulierung versteifungsfreier Volumenelemente*. Dissertation, Institut für Mechanik, Universität Karlsruhe (TH), 2000.
- [46] GEBHARDT, H.: *Finite Element Konzepte für schubelastische Schalen mit endlichen Drehungen*. Dissertation, Institut für Baustatik, Universität Karlsruhe (TH), 1990.
- [47] GLASER, S. und F. ARMERO: *On the formulation of enhanced strain finite elements in finite deformations*. Engineering Computations, 14(7):759–791, 1997.

- [48] GRUTTMANN, F.: *Theorie und Numerik dünnwandiger Faserverbundstrukturen*. Habilitation, Universität Hannover, 1995.
- [49] HALLQUIST, J.O.: *LS-DYNA Theory Manual*. Livermore Software Technology Corporation, March 2006.
- [50] HARNAU, M.: *Finite Volumen-Schalenelemente für große Deformationen und Kontakt*. Dissertation, Institut für Mechanik, Universität Karlsruhe (TH), 2004.
- [51] HARNAU, M., A. KONYUKHOV und K. SCHWEIZERHOF: *Algorithmic aspects in large deformation contact analysis using 'Solid-Shell' elements*. Computers & Structures, 83(21–22):1804–1823, 2005.
- [52] HARNAU, M. und K. SCHWEIZERHOF: *About linear and quadratic 'Solid-Shell' elements at large deformations*. Computers & Structures, 80(9):805–817, 2002.
- [53] HAUPT, P.: *Continuum Mechanics and Theory of Materials*. Springer, 2000.
- [54] HAUPTMANN, R.: *Strukturangepasste geometrisch nichtlineare Finite Elemente für Flächentragwerke*. Dissertation, Institut für Mechanik, Universität Karlsruhe (TH), 1997.
- [55] HAUPTMANN, R. und K. SCHWEIZERHOF: *A systematic development of 'solid-shell' element formulations for linear and non-linear analyses employing only displacement degrees of freedom*. International Journal for Numerical Methods in Engineering, 42(1):49–69, 1998.
- [56] HAUPTMANN, R., K. SCHWEIZERHOF und S. DOLL: *Extension of the 'solid-shell' concept for application to large elastic and large elastoplastic deformations*. International Journal for Numerical Methods in Engineering, 49(9):1121–1141, 2000.
- [57] HOLZAPFEL, G.A.: *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*. Wiley, 2000.
- [58] HUGHES, T.J.R.: *The Finite Element Method*. Dover Publications, 1. Auflage, 2000.
- [59] HUGHES, T.J.R., W.K. LIU und I. LEVIT: *Nonlinear Dynamic Finite Element Analysis of Shells*. In: WUNDERLICH, W., E. STEIN und K.-J. BATHE (Herausgeber): *Nonlinear Finite Element Analysis in Structural Mechanics, Proceedings of the Europe-U.S. Workshop*, Seiten 151–168. Springer, 1980.
- [60] HUGHES, T.J.R. und T.E. TEZDUYAR: *Finite Elements Based Upon Mindlin Plate Theory with Particular Reference to the Four-Node Bilinear Isoparametric Element*. Journal of Applied Mechanics, 48(3):587–595, 1981.
- [61] KIM, K.D., G.Z. LIU und S.C. HAN: *A resultant 8-node solid-shell element for geometrically nonlinear analysis*. Computational Mechanics, 35(5):315–331, 2005.
- [62] KLINKEL, S., F. GRUTTMANN und W. WAGNER: *A robust non-linear solid shell element based on a mixed variational formulation*. Computer Methods in Applied Mechanics and Engineering, 195(1):179–201, 2006.
- [63] KLINKEL, S. und W. WAGNER: *A Geometrical Non-Linear Brick Element Based on the EAS-Method*. International Journal for Numerical Methods in Engineering, 40(24):4529–4545, 1997.
- [64] KONYUKHOV, A.: *Geometrically Exact Theory of Contact Interactions*. Habilitation, Institut für Mechanik, Karlsruher Institut für Technologie (KIT), 2010.
- [65] KORELC, J.: *Symbolic approach in computational mechanics and its application to the enhanced strain method*. Dissertation, Institut für Mechanik, TH Darmstadt, 1996.

- [66] KORELC, J.: *Automatic generation of finite-element code by simultaneous optimization of expressions*. Theoretical Computer Science, 187(1–2):231–248, 1997.
- [67] KORELC, J.: *Automatic derivation of sensitivity terms for mixed Galerkin-collocation problems*. Zeitschrift für Angewandte Mathematik und Mechanik, 80:533–534, 2000.
- [68] KORELC, J.: *Multi-language approach in automatic generation of numerical procedures*. Zeitschrift für Angewandte Mathematik und Mechanik, 81:723–724, 2001.
- [69] KORELC, J.: *Multi-language and multi-environment generation of nonlinear finite element codes*. Engineering with Computers, 18(4):312–327, 2002.
- [70] KORELC, J.: *AceFEM*. University of Ljubljana – Faculty of Civil and Geodetic Engineering, 2009.
- [71] KORELC, J.: *AceGen*. University of Ljubljana – Faculty of Civil and Geodetic Engineering, 2009.
- [72] KORELC, J.: *Automation of primal and sensitivity analysis of transient coupled problems*. Computational Mechanics, 44(5):631–649, 2009.
- [73] KORELC, J.: *AceGen & AceFEM*, 2012. <http://www.fgg.uni-lj.si/Symech/>.
- [74] KORELC, J., U. SOLINC und P. WRIGGERS: *An Improved EAS Brick Element for Finite Deformation*. Computational Mechanics, 46(4):641–659, 2010.
- [75] KORELC, J. und P. WRIGGERS: *Consistent gradient formulation for a stable enhanced strain method for large deformations*. Engineering Computations, 13(1):103–123, 1996.
- [76] KORELC, J. und P. WRIGGERS: *An efficient 3D enhanced strain element with Taylor expansion of the shape functions*. Computational Mechanics, 19(2):30–40, 1996.
- [77] KORELC, J. und P. WRIGGERS: *Computer algebra and automatic differentiation in derivation of finite element code*. Zeitschrift für Angewandte Mathematik und Mechanik, 79:811–812, 1999.
- [78] KOSCHNICK, F.: *Geometrische Lockingeffekte bei Finiten Elementen und ein allgemeines Konzept zu ihrer Vermeidung*. Dissertation, Lehrstuhl für Statik, Technische Universität München, 2004.
- [79] KOSCHNICK, F., M. BISCHOFF, N. CAMPRUBÍ und K.-U. BLETZINGER: *The discrete strain gap method and membrane locking*. Computer Methods in Applied Mechanics and Engineering, 194(21–24):2444–2463, 2005.
- [80] LARCHER, M.: *Numerische Simulation des Betonverhaltens unter Stoßwellen mit Hilfe des Elementfreien Galerkin-Verfahrens*. Dissertation, Institut für Massivbau und Baustofftechnologie, Universität Karlsruhe (TH), 2007.
- [81] LEE, E.H.: *Elastic-Plastic Deformation at Finite Strains*. Journal of Applied Mechanics, 36:1–6, 1969.
- [82] LEE, E.H. und D.T. LIU: *Finite-Strain Elastic-Plastic Theory with Application to Plane-Wave Analysis*. Journal of Applied Physics, 38(1):19–27, 1967.
- [83] LSTC – LIVERMORE SOFTWARE TECHNOLOGY CORPORATION: *LS-DYNA*, 2012. <http://www.lstc.com>.
- [84] LUBLINER, J.: *Plasticity Theory*. Macmillan Publications, 1990.
- [85] MAGNUS, K., K. POPP und W. SEXTRO: *Schwingungen: eine Einführung in die physikalischen Grundlagen und die theoretische Behandlung von Schwingungsproblemen*. Vieweg + Teubner, 8. Auflage, 2008.

- [86] MAPLESOFT: *Maple*, 2012. <http://www.maplesoft.com/products/maple/>.
- [87] MATHWORKS: *MATLAB*, 2012. <http://www.mathworks.de/products/matlab/>.
- [88] MATTERN, S., CH. SCHMIED und K. SCHWEIZERHOF: *Highly Efficient Solid-Shell Finite Elements for Explicit Dynamic Analysis using Symbolic Programming*. 2012. in preparation.
- [89] MATTERN, S. und K. SCHWEIZERHOF: *On an efficient implementation of 'Solid-Shell' finite elements with quadratic shape functions for explicit time integration*. In: PIETRASZKIEWICZ, W. und I. KREJA (Herausgeber): *Proceedings – Shell Structures, Theory and Applications*, Band 2, Seiten 265–268. CRC Press, 2009.
- [90] MATTERN, S. und K. SCHWEIZERHOF: *Entwicklung hoch effizienter Schalelemente mit quadratischer Ansatzordnung in Schalenebene für transiente Analysen – Aufbau einer Systematik zur programmunterstützten Entwicklung von Schalelementen*. Abschlussbericht zum DFG-Forschungsvorhaben SCHW307/20-1, Deutsche Forschungsgemeinschaft, 2010.
- [91] MATTERN, S. und K. SCHWEIZERHOF: *Highly Efficient Implementation of ANS/ EAS 'Solid-Shell' Finite Elements for Explicit Time Integration*. *Proceedings of Applied Mathematics and Mechanics*, 10:191–192, 2010.
- [92] MATTERN, S. und K. SCHWEIZERHOF: *Software Supported Implementation of Efficient Solid-Shell Finite Elements*. In: TOPPING, B.H.V., J.M. ADAM, F.J. PALLARÉS, R. BRU und M.L. ROMERO (Herausgeber): *Proceedings of the Tenth International Conference on Computational Structures Technology*. Civil-Comp Press, 2010. paper 365.
- [93] MATTERN, S. und K. SCHWEIZERHOF: *Highly Efficient Solid-Shell Finite Elements for Dynamic Applications – an Implementation Concept using Symbolic Programming*. In: ROECK, G. DE, G. DEGRANDE, G. LOMBAERT und G. MÜLLER (Herausgeber): *8th International Conference on Structural Dynamics (EURODYN2011)*, 2011.
- [94] MAY, H. und U. MAY (Herausgeber): *Lexikon der ökonomischen Bildung*. Oldenburg Verlag, 5. aktualisierte und erweiterte Auflage, 2004.
- [95] MIEHE, CH.: *A Theoretical and Computational Model for Isotropic Elastoplastic Stress Analysis in Shells at Large Strains*. *Computer Methods in Applied Mechanics and Engineering*, 155(3–4):193–234, 1998.
- [96] NEUMANN, J. VON und R.D. RICHTMYER: *A Method for the Numerical Calculation of Hydrodynamic Shocks*. *Journal of Applied Physics*, 21:232, 1950.
- [97] NEWMARK, N.M.: *A method of computation for structural mechanics*. *Proceedings of the American Society of Civil Engineers, Journal of the Engineering Mechanics Division*, 85:67–94, 1959.
- [98] OLOVSSON, L. und K. SIMONSSON: *Iterative solution technique in selective mass scaling*. *Communications in Numerical Methods in Engineering*, 22(1):77–82, 2006.
- [99] OLOVSSON, L., K. SIMONSSON und M. UNOSSON: *Selective mass scaling for explicit finite element analyses*. *International Journal for Numerical Methods in Engineering*, 63(10):1436–1445, 2005.
- [100] OLOVSSON, L., M. UNOSSON und K. SIMONSSON: *Selective mass scaling for thin walled structures modeled with tri-linear solid elements*. *Computational Mechanics*, 34(2):134–136, 2004.

- [101] PAPADRAKAKIS, M.: *A Method for the Automated Evaluation of the Dynamic Relaxation Parameters*. Computer Methods In Applied Mechanics and Engineering, 25:35 – 48, 1981.
- [102] PARISCH, H.: *A continuum-based shell theory for non-linear applications*. International Journal for Numerical Methods in Engineering, 38(11):1855–1883, 1995.
- [103] PARISCH, H.: *Festkörper-Kontinuumsmechanik: Von den Grundgleichungen zur Lösung mit Finiten Elementen*. Teubner, 2003.
- [104] PUSO, M.A.: *A highly efficient enhanced assumed strain physically stabilized hexahedral element*. International Journal for Numerical Methods in Engineering, 49(8):1029–1064, 2000.
- [105] REESE, S.: *On a consistent hourglass stabilization technique to treat large inelastic deformations and thermo-mechanical coupling in plane strain problems*. International Journal for Numerical Methods in Engineering, 57(8):1095–1127, 2003.
- [106] REESE, S.: *A large deformation solid-shell concept based on reduced integration with hourglass stabilization*. International Journal for Numerical Methods in Engineering, 69(8):1671–1716, 2007.
- [107] RICHTMYER, R.D. und K.W. MORTON: *Difference Methods for Initial-Value Problems*. Interscience Publishers, 1967.
- [108] ROTTNER, TH.: *Sensitivitätsanalyse stabiler Gleichgewichtslagen dünnwandiger Strukturen – Entwicklung effizienter Lösungsverfahren für Parallelrechner*. Dissertation, Institut für Mechanik, Universität Karlsruhe (TH), 2000.
- [109] SCHMIED, CH., S. MATTERN und K. SCHWEIZERHOF: *Comparison of the numerical efficiency for variations of solid-shell finite elements using symbolic programming in explicit time integration*. In: *6th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS)*, 2012.
- [110] SCHWARZE, M. und S. REESE: *A reduced integration solid-shell finite element based on the EAS and the ANS concept - Geometrically linear problems*. International Journal for Numerical Methods in Engineering, 80(10):1322–1355, 2009.
- [111] SCHWARZE, M. und S. REESE: *A reduced integration solid-shell finite element based on the EAS and the ANS concept—Large deformation problems*. International Journal for Numerical Methods in Engineering, 85(3):289–329, 2011.
- [112] SCHWEIZERHOF, K. und MITARBEITER: *FEAP-MeKa*. Karlsruher Institut für Technologie, basierend auf: R. Taylor, “FEAP – A Finite Element Analysis Program”, University of California, Berkeley, 1991.
- [113] SIMO, J.C. und F. ARMERO: *Geometrically non-linear enhanced strain mixed methods and the method of incompatible modes*. International Journal for Numerical Methods in Engineering, 33(7):1413–1449, 1992.
- [114] SIMO, J.C., F. ARMERO und R.L. TAYLOR: *Improved versions of assumed enhanced strain tri-linear elements for 3D finite deformation problems*. Computer Methods in Applied Mechanics and Engineering, 110:359–386, 1993.
- [115] SIMO, J.C. und T.J.R. HUGHES: *On variational foundations of assumed strain methods*. Journal of Applied Mechanics, 53:51 – 54, 1986.
- [116] SIMO, J.C. und T.J.R. HUGHES: *Computational Inelasticity*. Springer, 2. Auflage, 2000.

- [117] SIMO, J.C. und M.S. RIFAI: *A class of mixed assumed strain methods and the method of incompatible modes*. International Journal for Numerical Methods in Engineering, 29(8):1595–1638, 1990.
- [118] SIMULIA: *Abaqus/Standard & Abaqus/Explicit*, 2012. <http://www.simulia.com>.
- [119] STRANG, G. und G.J. FIX: *An Analysis of the Finite Element Method*. Prentice-Hall, 1973.
- [120] SZE, K.Y. und L.Q. YAO: *A hybrid stress ANS solid-shell element and its generalization for smart structure modelling. Part I – solid-shell element formulation*. International Journal for Numerical Methods in Engineering, 48(4):545 – 564, 2000.
- [121] TAN, X.G. und L. VU-QUOC: *Efficient and accurate multilayer solid-shell element: Non-linear materials at finite strain*. International Journal for Numerical Methods in Engineering, 63(15):2124 – 2170, 2005.
- [122] TAYLOR, R.L. und F. AURICCHIO: *Linked Interpolation for Reissner-Mindlin Plate Elements: Part II – A Simple Triangle*. International Journal For Numerical Methods In Engineering, 36(18):3057–3066, 1993.
- [123] TAYLOR, R.L., P.J. BERESFORD und E.L. WILSON: *A non-conforming Element for Stress Analysis*. International Journal for Numerical Methods in Engineering, 10(6):1211–1219, 1976.
- [124] VERLET, L.: *“Experiments” on classical Fluids I. Thermomechanical Properties of Lennard-Jones Molecules*. Physical Review, 159:98–103, 1967.
- [125] WEMPNER, G., D. TALASLIDIS und C.-M. HWANG: *A Simple and Efficient Approximation of Shells via Finite Quadrilateral Elements*. Journal of Applied Mechanics, 49:115–120, 1982.
- [126] WILSON, E.L., R.L. TAYLOR, W.P. DOHERTY und J. GHABOUSSI: *Incompatible Displacement Models*. In: FENVES, S.J., N. PERRONE, A.R. ROBINSON und W.C. SCHNOBRICH (Herausgeber): *Numerical and Computer Models in Structural Mechanics*, Kapitel Finite Elements Fundamentals, Seiten 43–57. Academic Press, 1973.
- [127] WOLFRAM: *Mathematica*, 2012. <http://www.wolfram.com/mathematica>.
- [128] WOLFRAM, S.: *The Mathematica Book*. Wolfram Research, 5 Auflage, 2004.
- [129] WOOD, W. L.: *Practical Time-Stepping Schemes*. Oxford Applied Mathematics and Computing Science Series. Clarendon Press, Oxford, 1990.
- [130] WRIGGERS, P.: *Nichtlineare Finite-Element-Methoden*. Springer, Berlin, 2001.
- [131] WRIGGERS, P. und J. KORELC: *On enhanced strain methods for small and finite deformations of solids*. Computational Mechanics, 18(6):413–428, 1996.
- [132] XU, Z., O.C. ZIENKIEWICZ und L.F. ZENG: *Linked Interpolation for Reissner-Mindlin Plate Elements: Part III – An Alternative Quadrilateral*. International Journal For Numerical Methods In Engineering, 37(9):1437–1443, 1994.
- [133] ZAVARISE, G. und P. WRIGGERS: *A segment-to-segment contact strategy*. Mathematical and Computer Modelling, 28(4–8):497–515, 1998.
- [134] ZIENKIEWICZ, O. C. und R. L. TAYLOR: *The Finite Element Method for Solid and Structural Mechanics*, Band 2. Butterworth-Heinemann, 2006.
- [135] ZIENKIEWICZ, O. C., R. L. TAYLOR und P. NITHIARASU: *The Finite Element Method for Fluid Dynamics*, Band 3. Butterworth-Heinemann, 2006.

-
- [136] ZIENKIEWICZ, O. C., R. L. TAYLOR und J. Z. ZHU: *Finite Element Method: its Basis and Fundamentals*, Band 1. Butterworth-Heinemann, 2006.
- [137] ZIENKIEWICZ, O.C., Z. XU, L.F. ZENG, A. SAMUELSSON und N.-E. WIBERG: *Linked Interpolation for Reissner-Mindlin Plate Elements: Part I – A Simple Quadrilateral*. International Journal For Numerical Methods In Engineering, 36(18):3043–3056, 1993.

A. Elemente der Tensoralgebra

Die folgenden Abschnitte stellen eine Übersicht der in der vorliegenden Arbeit verwendeten Rechenregeln für Vektoren und Tensoren dar und sollen keinesfalls eine vollständige Beschreibung der Tensoralgebra liefern, da ausführliche Sammlungen der für die Kontinuumsmechanik erforderlichen Zusammenhänge bereits in den wesentlichen Fachbüchern zu diesem Thema enthalten sind (siehe z. B. [27, 53, 57, 103]). An einigen Stellen werden die für die symbolische Programmierung verwendeten Befehle erläutert. Der für die Arbeit gewählten Konvention folgend, sind MATHEMATICA-Befehle in Schreibmaschinenschrift und ACEGEN-Kommandos zusätzlich in blau dargestellt.

A.1. Grundlegende Rechenoperationen

Gegeben seien zwei Vektoren

$$\mathbf{a} = a_i \mathbf{G}^i = a^i \mathbf{G}_i \quad \text{und} \quad \mathbf{b} = b_i \mathbf{G}^i = b^i \mathbf{G}_i \quad (\text{A.1})$$

und zwei zweistufige Tensoren

$$\mathbf{A} = A_{ij} \mathbf{G}^i \otimes \mathbf{G}^j = A^{ij} \mathbf{G}_i \otimes \mathbf{G}_j \quad \text{und} \quad \mathbf{B} = B_{ij} \mathbf{G}^i \otimes \mathbf{G}^j = B^{ij} \mathbf{G}_i \otimes \mathbf{G}_j \quad (\text{A.2})$$

die sich z. B. in einer ko- und einer kontravarianten Basis darstellen lassen. Die folgenden grundlegenden Rechenregeln sind definiert:

- Skalarprodukt zweier Vektoren → einfache Verjüngung

$$\mathbf{a} \cdot \mathbf{b} = a_i b^i \quad (\text{A.3})$$

- Skalarprodukt zweier Tensoren → zweifache Verjüngung

$$\mathbf{A} : \mathbf{B} = A_{ij} B^{ij} \quad (\text{A.4})$$

- einfache Verjüngung eines Tensors zweiter Stufe durch einen Vektor

$$\mathbf{A} \cdot \mathbf{b} = A^{ij} b_j \mathbf{G}_i \quad (\text{A.5})$$

- dyadisches Produkt zweier Vektoren

$$\mathbf{C} = \mathbf{a} \otimes \mathbf{b}, \quad \text{mit} \quad \mathbf{C} \cdot \mathbf{c} = \mathbf{a}(\mathbf{b} \cdot \mathbf{c}) \quad (\text{A.6})$$

Die einfache Verjüngung lässt sich in MATHEMATICA durch den Befehl `Dot[a, b]` bestimmen, der auch durch

$$\mathbf{c} = \mathbf{a} . \mathbf{b} \quad (\text{A.7})$$

abgekürzt werden kann. Die zweifache Verjüngung kann durch `Tr[A^T.B]` über die Spur (Trace) eingegeben werden. Das dyadische Produkt erhält man durch `Outer[Times, a, b]`.

A.2. Ableitungsregeln für Tensoren

Bei der Bildung des Gradienten eines Tensors wird dessen Stufe um 1 erhöht. Für den Vektor $\mathbf{u}(\mathbf{X})$ als Tensor 1. Stufe ergibt sich:

$$\text{grad } \mathbf{u}(\mathbf{X}) = \frac{\partial u^i}{\partial x^j} \mathbf{G}_i \otimes \mathbf{G}^j. \quad (\text{A.8})$$

Die Divergenzbildung erniedrigt die Tensorstufe um 1:

$$\text{div } \boldsymbol{\sigma}(\mathbf{X}) = \frac{\partial \sigma^{ij}}{\partial x^i} \mathbf{G}_j. \quad (\text{A.9})$$

Die beiden Operationen lassen sich in MATHEMATICA durch

$$\text{grad } \mathbf{u} = \text{D}[\mathbf{u}, \{\mathbb{X}\}] \quad \text{bzw.} \quad \text{div } \boldsymbol{\sigma} = \text{Inner}[\text{D}, \boldsymbol{\sigma}, \mathbb{X}, \text{Plus}] \quad (\text{A.10})$$

umsetzen, in ACEGEN steht der entsprechende Befehl [SMSD](#) zur Verfügung. Für eine genauere Beschreibung der vorhandenen Optionen – einige werden in der Arbeit angewendet und an entsprechender Stelle erläutert – sei auf das Benutzerhandbuch [71] und die darin angegebenen Beispiele verwiesen.

A.3. Invarianten und Eigenwerte eines Tensors

Die Invarianten eines zweistufigen Tensors lauten

$$\begin{aligned} \text{I}_{\mathbf{A}} &= \text{tr } \mathbf{A} = A_i^i \\ \text{II}_{\mathbf{A}} &= \frac{1}{2} [(\text{tr } \mathbf{A})^2 - \text{tr } (\mathbf{A}^2)] = \frac{1}{2} \left[(A_i^i)^2 - A_m^i A_i^m \right] \\ \text{III}_{\mathbf{A}} &= \det \mathbf{A} = \frac{1}{6} [(\text{tr } \mathbf{A})^3 - 3 \text{tr } \mathbf{A} \text{tr } (\mathbf{A}^2) + 2 \text{tr } (\mathbf{A}^3)]. \end{aligned} \quad (\text{A.11})$$

Zur Bestimmung der Eigenwerte λ_i ist die Lösung des charakteristischen Polynoms

$$\det (\mathbf{A} - \lambda \mathbf{1}) = \lambda^3 - \text{I}_{\mathbf{A}} \lambda^2 + \text{II}_{\mathbf{A}} \lambda - \text{III}_{\mathbf{A}} = 0 \quad (\text{A.12})$$

erforderlich. Der symbolische Befehl in ACEGEN zur Aufstellung der Invarianten lautet [SMSInvariantsI](#)[\mathbf{A}], die Eigenwerte können durch [SMSEigenvalues](#)[\mathbf{A}] ermittelt werden. Der erzeugte Programmcode beinhaltet If-Abfragen zur Ermittlung der Lösungen der kubischen Bestimmungsgleichung.

A.4. Eigenrichtungen eines symmetrischen reellwertigen Tensors

Die hier beschriebene Vorgehensweise zur Bestimmung der Eigenrichtungen eines symmetrischen reellwertigen Tensors ist dem Anhang zur Tensorrechnung von WRIGGERS [130] entnommen.

Es sei \mathbf{A} ein symmetrischer reellwertiger Tensor zweiter Stufe mit den Eigenwerten λ_1 , λ_2 und λ_3 und den Invarianten nach Gleichung (A.11). Zur Bestimmung der dyadischen Produkte der Eigenvektoren $\varphi_i \otimes \varphi_i$ sind drei Fälle zu unterscheiden.

1. Alle Eigenwerte sind ungleich: $\lambda_1 \neq \lambda_2 \neq \lambda_3$

$$\boldsymbol{\varphi}_i \otimes \boldsymbol{\varphi}_i = \frac{\lambda_i}{2\lambda_i^3 - \text{I}_{\mathbf{A}}\lambda_i^2 + \text{III}_{\mathbf{A}}} \left(\mathbf{A} \cdot \mathbf{A} - (\text{I}_{\mathbf{A}} - \lambda_i) \mathbf{A} + \frac{\text{III}_{\mathbf{A}}}{\lambda_i} \mathbf{1} \right) \quad (\text{A.13})$$

2. Zwei Eigenwerte sind gleich: $\lambda_1 \neq \lambda_2 = \lambda_3$

$$\begin{aligned} \boldsymbol{\varphi}_1 \otimes \boldsymbol{\varphi}_1 & \text{ aus Gleichung (A.13) mit } i = 1 \\ \boldsymbol{\varphi}_j \otimes \boldsymbol{\varphi}_j & = \mathbf{1} - \boldsymbol{\varphi}_1 \otimes \boldsymbol{\varphi}_1 \end{aligned} \quad (\text{A.14})$$

3. Alle Eigenwerte sind gleich: $\lambda_1 = \lambda_2 = \lambda_3$

$$\boldsymbol{\varphi}_i \otimes \boldsymbol{\varphi}_i = \mathbf{1} \quad (\text{A.15})$$

Zur Bestimmung der Eigenrichtungen liegt kein symbolischer ACEGEN-Befehl vor, der dem MATHEMATICA-Kommando `Eigensystem` entspricht. Aus diesem Grund wurde die beschriebene Vorgehensweise in einer Funktion umgesetzt, durch deren Aufruf die drei dyadischen Produkte $\boldsymbol{\varphi}_i \otimes \boldsymbol{\varphi}_i$ zurückgegeben werden. Der erzeugte Programmcode ist recht umfangreich, da die Eigenwerte berechnet und die oben angegebenen Fälle durch verschachtelte If-Anfragen abgeprüft werden müssen.

B. Programmierung mit ACEGEN

Ziel dieses Anhangs ist die anschauliche Beschreibung des im Rahmen der vorliegenden Arbeit entwickelten Implementierungskonzeptes für Finite-Elemente-Routinen innerhalb eines expliziten Zeitintegrationsverfahrens. Die Implementierung erfolgte mit der Programmierumgebung ACEGEN, die in Kapitel 4 im Hauptteil beschrieben ist. Es werden die erforderlichen Implementierungsschritte skizziert und die verwendeten ACEGEN-Kommandos erläutert. Dabei sind MATHEMATICA-Befehle durchgehend in Schreibmaschinenschrift gesetzt, ACEGEN-Kommandos sind zusätzlich blau dargestellt. Eine vollständige Beschreibung des Befehlsatzes, sowie weitere Details zur Programmierung und zu einzelnen Befehlen können dem Handbuch [71], bzw. den darin enthaltenen Beispielen entnommen werden.

Zunächst wird an einem Prinzipbeispiel die grundsätzliche Funktionalität von ACEGEN gezeigt, in dem einfache Matrix-Operationen symbolisch umgesetzt und in Programmcode übersetzt werden. Am einfachen Beispiel eines 4-Knoten-Scheibenelementes werden die wesentlichen Unterschiede zwischen händischem Programmieren und der Implementierung mit ACEGEN verdeutlicht. Abschließend wird die Umsetzung einer Materialformulierung mit Plastizität für finite Deformationen beschrieben, wie sie teilweise für die in Abschnitt 3.2 eingeführten Solid-Shell-Elemente umgesetzt wurde. An diesem Beispiel lassen sich noch einige programmiertechnische Besonderheiten zeigen.

B.1. Prinzipbeispiel – Matrizenoperation

Mit der Hilfe von ACEGEN soll eine Routine zur Berechnung von

$$\mathbf{D} = \det \mathbf{A} (\mathbf{A} \cdot \mathbf{B}^T \cdot \mathbf{C}^{-1}) \quad (\text{B.1})$$

aus den beliebigen Matrizen $\mathbf{A}^{3 \times 3}$, $\mathbf{B}^{3 \times 3}$ und $\mathbf{C}^{3 \times 3}$ implementiert werden. Die Umsetzung mit symbolischer Programmierung erfordert im Einzelnen folgende Schritte:

- **Initialisierung**

Die Sitzung wird eingerichtet, die gewünschte Programmiersprache und der Grad der Optimierung gewählt, sowie der Name der Routine und die zu übergebenden Felder spezifiziert.

```
SMSInitialize["mmult", "Language" -> "Fortran", "Mode" -> "Optimal"];  
SMSModule["mmult", Real [a$$[3, 3], b$$[3, 3], c$$[3, 3], d$$[3, 3]]];
```

 (B.2)

Durch die verwendete Option "Mode" -> "Optimal" wird hierbei ein möglichst kompakter Quellcode erzeugt. Die Generierung der dauert im Vergleich zu beispielsweise "Mode" -> "Prototype" länger, was sich jedoch erst bei aufwändigeren Programmen bemerkbar macht.

- **Zuweisung von Variablen**

Im nächsten Schritt werden die der Routine übergebenen Werte auf MATHEMATICA-Ebene eingeführt, um für symbolische Operationen verwendet werden zu können.

```
A = Array[SMSReal[a$$[#1,#2]] &, {3, 3}];
B = Array[SMSReal[b$$[#1,#2]] &, {3, 3}];
C = Array[SMSReal[c$$[#1,#2]] &, {3, 3}];
```

(B.3)

- **Ausführen der Rechenoperation**

Nun kann die Operation selbst mit den angelegten Feldern symbolisch durchgeführt werden. Die verwendeten Befehle für die Determinante und die Inverse sind dabei an die MATHEMATICA-Syntax angelehnt.

```
D = SMSDet[A] (A.BT.SMSInverse[C]);
```

(B.4)

- **Ausgabe**

Es erfolgt die Zuweisung des Ergebnisses zur globalen Variablen und die Erstellung des Programmcodes in FORTRAN

```
SMSExport[D, d$$];
SMSWrite[];
```

(B.5)

In Abbildung B.1 sind jeweils Auszüge aus dem erzeugten FORTRAN-Quelltext angegeben, der sich durch die Wahl "Mode" -> "Optimal", bzw. "Mode" -> "Prototype" bei der Initialisierung in Gleichung (B.2) ergibt. Bei der Variablendeklaration wird neben den übergebenen Feldern a, b, c und dem Ausgabefeld d ein eindimensionales Feld v angelegt, auf welchem sämtliche Rechenoperationen durchgeführt werden. Insbesondere bei "Mode" -> "Prototype" wird deutlich, dass zunächst die dem Programm übergebenen Größen in v abgespeichert werden und dann ausschließlich mit diesen Werten gearbeitet wird. Das Ergebnis wird schließlich aus v in d umgespeichert, um so aus der Routine exportiert werden zu können. Die Größe des internen Feldes ist standardmäßig auf eine angemessen hohe Zahl als Voreinstellung (5000) gesetzt und kann mit der Option "VectorLength" in MATHEMATICA gewählt werden. Werden mehr als 5000 interne Variablen benötigt, ist eine Anpassung erforderlich, bei kleinen Routinen kann die Wahl eines kleineren Wertes sinnvoll sein. Auffällig ist in Abbildung B.1 bei der Variante mit "Mode" -> "Optimal" der Beginn der Operationen auf dem lokalen Feld bei v(30), was auf die Optimierung der Operationen zurückzuführen ist. Diese Optimierung wird am Ende der Code-Erzeugung automatisch durchgeführt; Übergabe- und Rechenoperationen werden zusammengefasst und redundante Schritte eliminiert, wodurch die Generierung der Routine insbesondere bei größeren Programmen deutlich länger dauern kann. Bei "Mode" -> "Prototype" – hier wird laut Handbuch eine 'moderate Optimierung' durchgeführt – werden die genannten Schritte Übergabe, Rechenoperationen und Export nacheinander durchgeführt, was deutlich mehr Programmzeilen in der generierten Routine zur Folge hat. Die lokalen Variablen werden dann durchgängig beginnend mit v(1) beschrieben. Als weitere Optionen ohne Optimierung stehen "Mode" -> "Plain" und "Mode" -> "Debug" zur Verfügung, für genauere Informationen sei auf die Dokumentation im Handbuch [71] verwiesen. Die erzeugte Routine kann danach direkt in ein Hauptprogramm eingebunden, aufgerufen und mit einem Standard-Compiler übersetzt werden. Die Wahl einer anderen Programmiersprache (z. B. C) kann durch Setzen der entsprechenden Option im Initialisierungsschritt (B.2) vorgenommen werden.

"Mode" -> "Optimal"

```

SUBROUTINE mmult(a,b,c,d)
  IMPLICIT NONE
  DOUBLE PRECISION v(5001),a(3,3),b(3,3),c(3,3),d(3,3)
  v(30)=a(1,3)*(-(a(2,2)*a(3,1))+a(2,1)*a(3,2))+a(1,2)*(a(2,3)*a
&(3,1)-a(2,1)*a(3,3))+a(1,1)*(-(a(2,3)*a(3,2))+a(2,2)*a(3,3))
  v(54)=a(1,3)*b(1,1)+a(2,3)*b(2,1)+a(3,3)*b(3,1)
  v(44)=a(1,2)*b(1,1)+a(2,2)*b(2,1)+a(3,2)*b(3,1)
  v(33)=a(1,1)*b(1,1)+a(2,1)*b(2,1)+a(3,1)*b(3,1)
  v(53)=a(1,3)*b(1,2)+a(2,3)*b(2,2)+a(3,3)*b(3,2)

  [...]

  d(2,2)=(v(40)*v(41)+v(42)*v(43)+v(44)*v(45))*v(58)
  d(2,3)=(v(40)*v(47)+v(42)*v(48)+v(44)*v(49))*v(58)
  d(3,1)=(v(36)*v(52)+v(37)*v(53)+v(38)*v(54))*v(58)
  d(3,2)=(v(41)*v(52)+v(43)*v(53)+v(45)*v(54))*v(58)
  d(3,3)=(v(47)*v(52)+v(48)*v(53)+v(49)*v(54))*v(58)
END

```

"Mode" -> "Prototype"

```

SUBROUTINE mmult(a,b,c,d)
  IMPLICIT NONE
  DOUBLE PRECISION v(5001),a(3,3),b(3,3),c(3,3),d(3,3)
  v(1)=a(1,1)
  v(2)=a(1,2)
  v(3)=a(1,3)
  v(4)=a(2,1)
  v(5)=a(2,2)
  v(6)=a(2,3)
  v(7)=a(3,1)
  v(8)=a(3,2)
  v(9)=a(3,3)
  v(30)=v(3)*(-(v(5)*v(7))+v(4)*v(8))-v(2)*(-(v(6)*v(7))+v(4)*v(9)
&)+v(1)*(-(v(6)*v(8))+v(5)*v(9))
  v(10)=b(1,1)
  v(11)=b(1,2)

  [...]

  v(50)=v(30)*(v(28)*v(40)*v(47)+v(28)*v(42)*v(48)+v(28)*v(44)*v
&(49))
  v(51)=v(30)*(v(28)*v(36)*v(52)+v(28)*v(37)*v(53)+v(28)*v(38)*v
&(54))
  v(55)=v(30)*(v(28)*v(41)*v(52)+v(28)*v(43)*v(53)+v(28)*v(45)*v
&(54))
  v(56)=v(30)*(v(28)*v(47)*v(52)+v(28)*v(48)*v(53)+v(28)*v(49)*v
&(54))
  d(1,1)=v(29)
  d(1,2)=v(34)
  d(1,3)=v(35)
  d(2,1)=v(39)
  d(2,2)=v(46)
  d(2,3)=v(50)
  d(3,1)=v(51)
  d(3,2)=v(55)
  d(3,3)=v(56)
END

```

Abbildung B.1.: Auszug aus den ACEGEN-generierten Routinen zur Auswertung von Gleichung (B.1) bei unterschiedlicher Wahl der Optimierungs-Option

B.2. Prinzipbeispiel zum Implementierungskonzept

Im Folgenden wird das Implementierungskonzept für Elementroutinen unter Verwendung des symbolischen Programmierwerkzeugs ACEGEN beschrieben, wie es in der vorliegenden Arbeit insbesondere für die Nutzung innerhalb expliziter Zeitintegrationsverfahren entwickelt wurde. Aus Gründen der Darstellung wird das Implementierungskonzept am einfachen Prinzipbeispiel eines 4-Knoten-Scheibenelementes erläutert, da hier das Vorgehen und die erforderlichen Elementmatrizen und Vektoren sehr übersichtlich sind. Nach einer kurzen theoretischen Einführung der Elementformulierung wird eine mögliche händische Umsetzung besprochen, wie sie üblicherweise durch eine Strukturierung der erforderlichen Programmschritte in Unterroutrinen erreicht wird. Anschließend wird die Strategie zur softwaregestützten Programmierung vorgestellt, indem die zuvor beschriebenen Schritte in symbolischer Form mit ACEGEN umgesetzt werden. Das entwickelte Konzept wird im Rahmen der Arbeit für komplexere Formulierungen aus dem Bereich der Schalen- und Volumenschalenelemente angewendet, welche in konvektiven Koordinaten formuliert sind. Zur besseren Übertragbarkeit der einzelnen Implementierungsschritte wird auch für die Scheibenformulierung eine Schreibweise in konvektiven Koordinaten gewählt, obwohl dies für eine 2D-Elementformulierung nicht erforderlich ist.

B.2.1. 4-Knoten-Scheibenelement

Die zur prinzipiellen Darstellung verwendete Elementformulierung ist ein isoparametrisches 4-Knoten-Element mit zwei Verschiebungsfreiheitsgraden pro Knoten. Mit den bilinearen LAGRANGE-Ansätzen

$$N_i(\xi, \eta) = \frac{1}{4}(1 + \xi_i \xi)(1 - \eta_i \eta), \quad i = 1 \dots 4 \quad (\text{B.6})$$

ergibt sich die Interpolation der Geometrie und der Verschiebungen zu

$$\mathbf{X} = \sum_{i=1}^4 N_i \mathbf{X}_i \quad \text{bzw.} \quad \mathbf{u} = \sum_{i=1}^4 N_i \mathbf{u}_i, \quad (\text{B.7})$$

wobei $\mathbf{X}_i = [X_i \quad Y_i]^T$ den Ortsvektor und $\mathbf{u}_i = [u_i \quad v_i]^T$ den Verschiebungsvektor des Knotens i darstellen. Ordnet man die Knotenverschiebungen in einem Vektor

$$\mathbf{d} = [u_1 \quad v_1 \quad \dots \quad u_4 \quad v_4]^T \quad (\text{B.8})$$

an, so ergibt sich der innere Kraftvektor durch einfache Ableitung des Funktionals der inneren Energie nach diesem Vektor:

$$\mathbf{f}^{\text{int}} = \Pi_{,\mathbf{d}}^{\text{int}} = \frac{\partial}{\partial \mathbf{d}} \int_V \mathcal{W}(\mathbf{E}) \, dV. \quad (\text{B.9})$$

Dies gilt für den Fall eines hyperelastischen Materialgesetz, welches sich durch eine Verzerungsenergiefunktion $\mathcal{W}(\mathbf{E})$ beschreiben lässt. Für den einfachen Fall des linear-elastischen ST. VENANT-KIRCHHOFF-Material gilt

$$\mathcal{W}(\mathbf{E}) = \frac{1}{2} \lambda (\text{tr } \mathbf{E})^2 + \mu \mathbf{E} : \mathbf{E}, \quad (\text{B.10})$$

mit den Lamé-Konstanten

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad \text{und} \quad \mu = \frac{E}{2(1+\nu)}. \quad (\text{B.11})$$

Hier ist \mathbf{E} der GREEN-LAGRANGE-Verzerrungstensor aus Gleichung (1.12) an einer kontravarianten Basis mit den kovarianten Komponenten

$$E_{ij} = \frac{1}{2} \left(\mathbf{G}_i \cdot \frac{\partial \mathbf{u}}{\partial \xi_j} + \frac{\partial \mathbf{u}}{\partial \xi_i} \cdot \mathbf{G}_j + \frac{\partial \mathbf{u}}{\partial \xi_i} \cdot \frac{\partial \mathbf{u}}{\partial \xi_j} \right), \quad \text{mit } i, j = 1 \dots 2. \quad (\text{B.12})$$

und

$$E_{33} = \frac{\nu}{\nu-1} (E_{11} + E_{22}) \quad (\text{B.13})$$

zur Erfüllung der Annahme eines ebenen Spannungszustandes. Wie bereits eingangs erwähnt ist eine Formulierung in einer konvektiven Metrik mit den kovarianten Basisvektoren \mathbf{G}_i , die als Spalten in der Jacobi-Matrix für die isoparametrische Abbildung

$$\mathbf{J} = \begin{pmatrix} \frac{\partial X}{\partial \xi} & \frac{\partial Y}{\partial \xi} \\ \frac{\partial X}{\partial \eta} & \frac{\partial Y}{\partial \eta} \end{pmatrix} = \begin{pmatrix} - & \mathbf{G}_1 & - \\ - & \mathbf{G}_2 & - \end{pmatrix} \quad (\text{B.14})$$

angeordnet sind, für die Scheibenformulierung nicht erforderlich, jedoch für den Übergang auf Schalenformulierungen hilfreich, da dann die gleiche Schreibweise verwendet werden kann. Die Spannungen lassen sich als Ableitung von Gleichung (B.10) nach dem Verzerrungstensor bestimmen. Es ergibt sich der zum GREEN-LAGRANGE-Verzerrungstensor energetisch konjugierte 2. POILA-KIRCHHOFF Spannungstensor zu

$$\mathbf{S}(\mathbf{E}) = \lambda \operatorname{tr} \mathbf{E} \mathbf{I} + 2\mu \mathbf{E}. \quad (\text{B.15})$$

B.2.2. Händische Implementierung

Die Elementformulierung wurde zunächst ‘*händisch*’, d. h. durch manuelle Eingabe von Quelltext implementiert. Die Vorgehensweise zur Implementierung der Routine zur Aufstellung der internen Knotenkräfte in FEAP-MEKA [112] ist ausführlich in Abbildung B.2 angegeben. Auf einige Schritte wird im Folgenden detailliert eingegangen. Die in runden Klammern angegebenen Zahlen beziehen sich auf Abbildung B.2.

- **Aufstellen der Ansatzfunktionen (1)**

Die Ansatzfunktionen werden gemäß Gleichung (B.6) aufgestellt; deren Ableitungen müssen vorher gebildet, implementiert und vorgehalten werden.

- **Jacobi-Matrix (2)**

Die Jacobi-Matrix aus Gleichung (B.14) wird mit den abgeleiteten Ansatzfunktionen aufgestellt:

$$\mathbf{J} = \begin{pmatrix} \sum \frac{\partial N_i}{\partial \xi_i} X_i & \sum \frac{\partial N_i}{\partial \xi_i} Y_i \\ \sum \frac{\partial N_i}{\partial \eta_i} X_i & \sum \frac{\partial N_i}{\partial \eta_i} Y_i \end{pmatrix}, \quad i = 1 \dots 4 \quad (\text{B.16})$$

Die Determinante und die Inverse werden entsprechend gebildet:

$$\det \mathbf{J} = J_{11} J_{22} - J_{12} J_{21}, \quad \mathbf{J}^{-1} = \frac{1}{\det \mathbf{J}} \begin{pmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{pmatrix}. \quad (\text{B.17})$$

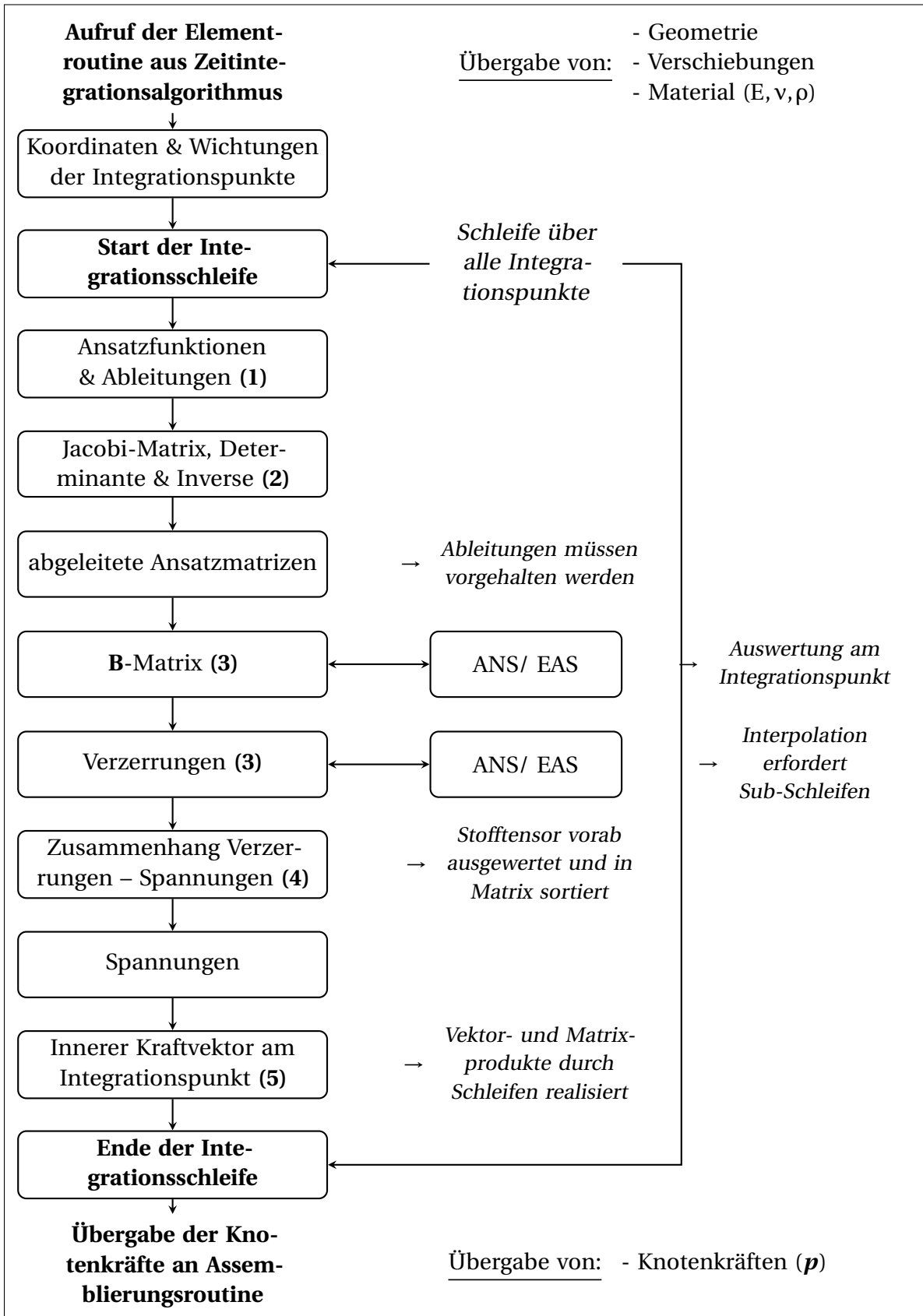


Abbildung B.2.: Schema für händische Implementierung der Elementroutine (Referenzierungen in Klammern beziehen sich auf Abschnitt B.2.2)

- **Kinematik → B-Matrix & Verzerrungen (3)**

Die händische Implementierung erfordert die separate Aufstellung der Verzerrungen und der sogenannten **B**-Matrix – also der Ableitung nach den Knotenfreiheitsgraden $\boldsymbol{\varepsilon}, \boldsymbol{d}$. Um in der Formulierung in konvektiven Koordinaten zu bleiben, ergibt sich für die nichtlinearen Verzerrungen aus Gleichung (B.12)

$$\boldsymbol{\varepsilon} = \begin{pmatrix} E_{11} \\ E_{22} \\ 2E_{12} \end{pmatrix} = \begin{pmatrix} \mathbf{G}_1^T \mathbf{D}_{,\xi} \boldsymbol{d} & + \frac{1}{2} \boldsymbol{d}^T \mathbf{D}_{,\xi}^T \mathbf{D}_{,\xi} \boldsymbol{d} \\ \mathbf{G}_2^T \mathbf{D}_{,\eta} \boldsymbol{d} & + \frac{1}{2} \boldsymbol{d}^T \mathbf{D}_{,\eta}^T \mathbf{D}_{,\eta} \boldsymbol{d} \\ \mathbf{G}_1^T \mathbf{D}_{,\eta} \boldsymbol{d} + \mathbf{G}_2^T \mathbf{D}_{,\xi} \boldsymbol{d} & + \boldsymbol{d}^T \mathbf{D}_{,\xi}^T \mathbf{D}_{,\eta} \boldsymbol{d} \end{pmatrix} \quad (\text{B.18})$$

und für die **B**-Matrix oder Verzerrungs- Verschiebungsmatrix

$$\mathbf{B} = \boldsymbol{\varepsilon}, \boldsymbol{d} = \begin{pmatrix} \mathbf{G}_1^T \mathbf{D}_{,\xi} & + \boldsymbol{d}^T \mathbf{D}_{,\xi}^T \mathbf{D}_{,\xi} \\ \mathbf{G}_2^T \mathbf{D}_{,\eta} & + \boldsymbol{d}^T \mathbf{D}_{,\eta}^T \mathbf{D}_{,\eta} \\ \mathbf{G}_1^T \mathbf{D}_{,\eta} + \mathbf{G}_2^T \mathbf{D}_{,\xi} & + \boldsymbol{d}^T \mathbf{D}_{,\xi}^T \mathbf{D}_{,\eta} + \boldsymbol{d}^T \mathbf{D}_{,\eta}^T \mathbf{D}_{,\xi} \end{pmatrix}. \quad (\text{B.19})$$

In den Matrizen $\mathbf{D}_{,\xi}$ und $\mathbf{D}_{,\eta}$ sind die Ableitungen der Ansatzfunktionen in der Form angeordnet, sodass die Beziehungen

$$\frac{\partial \boldsymbol{u}}{\partial \xi} = \mathbf{D}_{,\xi} \boldsymbol{d}, \quad \text{bzw.} \quad \frac{\partial \boldsymbol{u}}{\partial \eta} = \mathbf{D}_{,\eta} \boldsymbol{d} \quad (\text{B.20})$$

gelten. Die Aufstellung der Verzerrungen am Integrationspunkt wird durch Schleifen realisiert. Eine Interpolation der Verzerrungen, wie sie z. B. für die *Methode der angenommenen Verzerrungen (ANS)* erforderlich ist, erfolgt dann über die Auswertung der Verzerrungen an sog. Kollokationspunkten, was über Unterschleifen implementiert wird. Bei einer Interpolation der Verzerrungen muss demnach auch die **B**-Matrix entsprechend neu aufgestellt werden, was sehr viele Rechenoperationen erfordert.

- **Stoffgesetz & Spannungen (4)**

Der Zusammenhang zwischen Verzerrungen und Spannungen aus Gleichung (B.10) wird vorab ausgewertet und in eine Materialmatrix sortiert. Hierzu werden die Komponenten des Verzerrungs- und des Spannungstensor in Vektoren sortiert. Es ergibt sich für das verwendete einfache Materialgesetz die bekannte Beziehung

$$\boldsymbol{\sigma} = \begin{pmatrix} S^{11} \\ S^{22} \\ S^{12} \end{pmatrix} = \begin{pmatrix} C^{11} & C^{12} & C^{13} \\ C^{12} & C^{22} & C^{23} \\ C^{13} & C^{23} & C^{33} \end{pmatrix} \begin{pmatrix} E_{11} \\ E_{22} \\ 2E_{12} \end{pmatrix} = \mathbf{C} \boldsymbol{\varepsilon}. \quad (\text{B.21})$$

mit der in die konvektive Metrik transformierten, symmetrischen Materialmatrix

$$\mathbf{C} = \lambda \begin{pmatrix} G^{11} G^{11} & G^{11} G^{22} & G^{11} G^{12} \\ G^{11} G^{22} & G^{22} G^{22} & G^{22} G^{33} \\ G^{12} G^{11} & G^{12} G^{22} & G^{12} G^{12} \end{pmatrix} + \mu \begin{pmatrix} 2G^{11} G^{11} & 2G^{12} G^{12} & 2G^{11} G^{12} \\ 2G^{12} G^{12} & 2G^{22} G^{22} & 2G^{12} G^{22} \\ 2G^{11} G^{12} & 2G^{12} G^{22} & G^{12} G^{12} + G^{22} G^{22} \end{pmatrix}. \quad (\text{B.22})$$

Problematisch bei der händischen Implementierung von Materialgesetzen ist die erforderliche Vorbereitung der kontinuumsmechanischen Materialformulierung für die

Programmierung. Bei komplexeren Materialmodellen kann dieser Prozess zeitaufwändig und auch fehleranfällig sein. Eine Änderung in der Materialformulierung, hier also beispielsweise die Verwendung einer ebenfalls hyperelastischen NEO-HOOKE-Formulierung, erfordert eine vollständig neue Programmierung und entsprechende Vorüberlegungen über einzelnen Einträge und deren Sortierung in der Materialmatrix. Zusätzliche Programmschritte sind erforderlich, da die Materialmatrix dann nicht mehr konstant, sondern vom aktuellen Verschiebungszustand abhängig ist.

- **Vektor der inneren Kräfte am Integrationspunkt (5)**

Der Knotenkraftvektor wird schließlich am Integrationspunkt aufgestellt, mit der entsprechenden Wichtung multipliziert und innerhalb der Integrationsschleife aufsummiert. Die Auswertung der Vektor- und Matrixprodukte in

$$f_{GP}^{\text{int}} = \det \mathbf{J} w_i \mathbf{B}^T \boldsymbol{\sigma} = \det \mathbf{J} w_i \mathbf{B}^T \mathbf{C} \boldsymbol{\epsilon} \quad (\text{B.23})$$

wird durch entsprechende Schleifen implementiert.

Da einzelne Programmteile des händisch umgesetzten Programms häufiger ausgeführt werden müssen, bietet sich hier eine Gliederung in Unterroutinen an. Um Programmzeilen zu sparen und eine übersichtliche Implementierung zu erreichen, werden üblicherweise die in Abbildung B.2 aufgeführten Schritte sukzessive aufgerufen. Modifikationen an der Elementformulierung erfordern dann jedoch Änderungen an verschiedenen Teilen der Programmierung.

B.2.3. Implementierung mit ACEGEN

Bei der Implementierung mit ACEGEN ist grundsätzlich zu beachten, möglichst viele Rechenschritte in wenigen Unterroutinen zusammenzufassen, um die Vorteile der symbolischen Programmierung nutzen zu können. Im Folgenden wird die ACEGEN-gestützte Programmierung der Routine zur Berechnung der inneren Knotenkräfte für die in Abschnitt B.2.1 eingeführte Scheibenformulierung vorgestellt. Hierbei wird die in Abschnitt B.2.2 entwickelte Programmstruktur zur händischen Implementierung weitestgehend beibehalten, jedoch werden alle erforderlichen Schritte in einer einzigen Routine durchgeführt, die aus den symbolischen Operationen automatisch generiert wird. So ist die Initialisierung, bei der die übergebenen Größen für die symbolische Programmierung aufbereitet werden, nur einmal durchzuführen. Die einzelnen Schritte mit der zugehörigen Syntax werden erläutert und entsprechende Besonderheiten diskutiert. Die Zahlen in Klammern beziehen sich auf die in Abbildung B.3 referenzierten Stellen.

- **Initialisierung und Definition von Variablen (1)**

Die Programmierung mit ACEGEN erfordert, wie bereits in Abschnitt B.1 zunächst einige Schritte zur Initialisierung. Diese werden in MATHEMATICA zu Beginn der Eingabedatei – des sog. *Notebooks* – durchgeführt und sind im Wesentlichen für alle Implementierungen gleich. Das aktuelle Projekt und die zu erstellende Subroutine wird über das Kommando

```
SMSInitialize["pvec012", "Language" -> "Fortran", "Mode" -> "Optimal"];
(B.24)
```

initialisiert. Der Dateiname, unter dem die Routine gespeichert wird, sowie die Programmiersprache und der gewünschte Grad der Optimierung werden hier festgelegt.

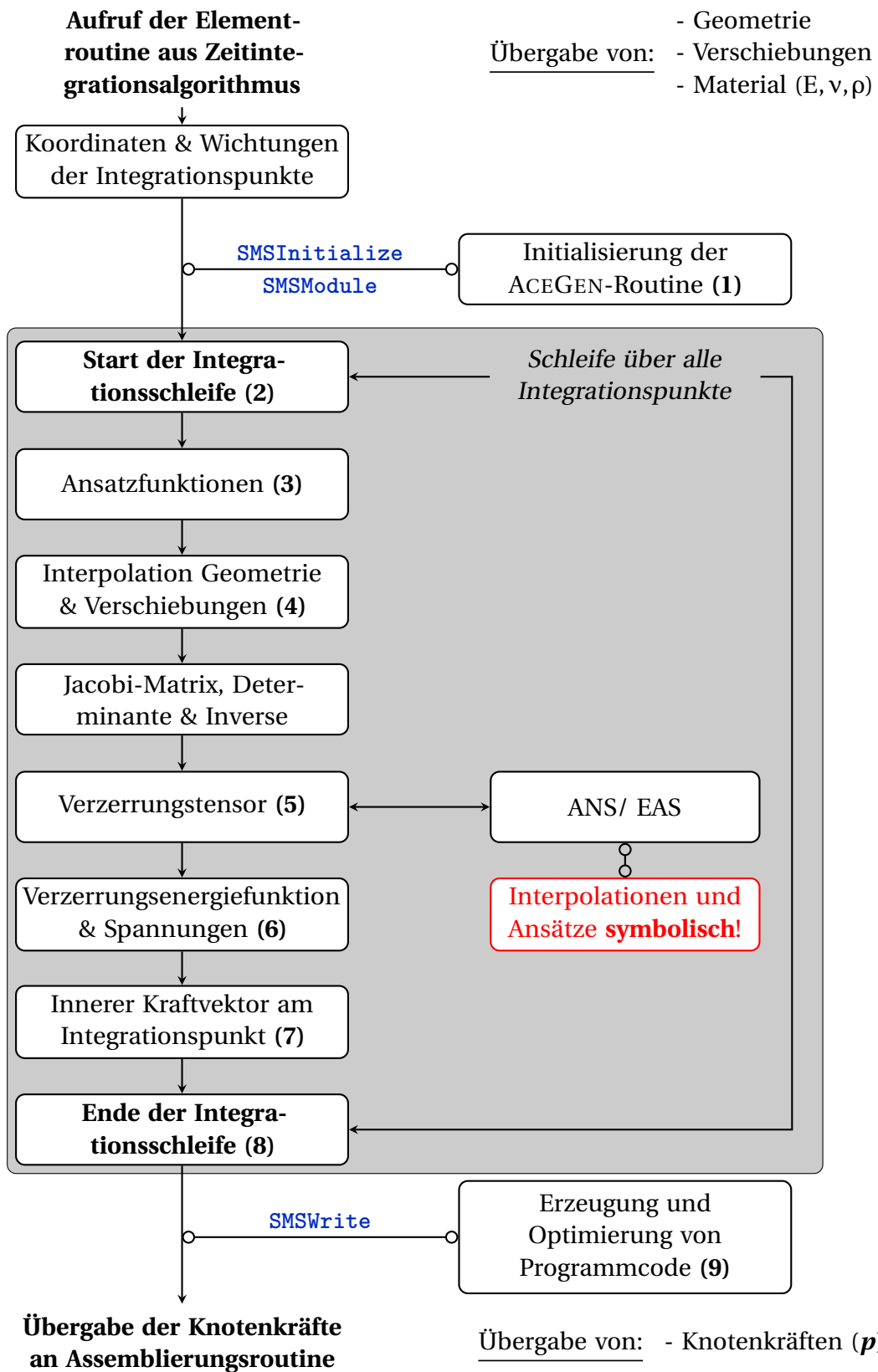


Abbildung B.3.: Schema für ACEGEN-unterstützte Implementierung der Elementroutine – grau hinterlegt: in ACEGEN umgesetzte Routine (Referenzierungen in Klammern beziehen sich auf Abschnitt B.2.3)

Die Anzahl der für die Routine erforderlichen lokalen Variablen ist zunächst auf 5000 festgesetzt, kann jedoch über die Option "VectorLength" gesteuert werden. Über den Befehl

```
SMSModule["pvec001", Real[p$$[8], x1$$[2, 4], u1$$[2, 4], emod$$,
nu$$, thk$$]];
```

 (B.25)

wird der Name der Routine selbst, sowie die zu übergebenden Variablen und Felder definiert. Im vorliegenden Beispiel wird das Feld für den Knotenkraftvektor übergeben, die Elementinformationen für Geometrie, aktuelle Verschiebungen und Material.

Allen Variablen, die der Routine übergeben werden, muss ein lokaler Variablenname zugewiesen werden, über den sie innerhalb der symbolischen Umgebung angesprochen werden können. Hierzu stehen statt dem in MATHEMATICA üblichen Zuweisungszeichen (=) vier verschiedene Operatoren (\equiv , \vdash , \equiv , \dashv) für Einfach- und Mehrfachzuweisungen zur Verfügung. Der Typ der Variablen wird über die Befehle `SMSReal`, `SMSInteger`, etc. gesteuert. Werden Variablen nur innerhalb der Routine benötigt, um z. B. nach ihnen abzuleiten, so müssen diese über das Kommando `SMSFictive` eingeführt werden. Weitere Hilfsvariablen müssen, wie bei Computeralgebra-Programmen üblich, nicht explizit deklariert werden. Die Zuweisung der Geometrie zu einer lokalen Variable erfolgt z. B. über

```
x1  $\equiv$  Array[SMSReal[x1$$[#1, #2]] &, {2, 4}];
```

 (B.26)

Die Initialisierungsschritte sind in Abbildung B.3 unter Punkt (I) zusammengefasst, da sie streng genommen nicht Teil der tatsächlichen Routine sind.

Die Lage der Punkte und Wichtungen für die numerische Integration werden direkt auf der symbolischen Ebene über

```
 $\xi$ GP  $\equiv$  SMSArray[{-1/Sqrt[3], 1/Sqrt[3], 1/Sqrt[3], -1/Sqrt[3]}];
 $\eta$ GP  $\equiv$  SMSArray[{-1/Sqrt[3], -1/Sqrt[3], 1/Sqrt[3], 1/Sqrt[3]}];
wGP  $\equiv$  SMSArray[{1, 1, 1, 1}];
nGP  $\equiv$  4;
```

 (B.27)

implementiert. Auf die Einträge eines über `SMSArray` definierten Feldes kann innerhalb der Integrationschleife über den Befehl `SMSPart` zugegriffen werden. Bei Elementformulierungen mit variabler Integrationsordnung, bietet sich auch eine Übergabe dieser Informationen über `SMSModule` an. Dies kann z. B. bei Schalenelementen mit beliebiger Anzahl an Integrationspunkten über die Schalendicke praktikabel sein, soll jedoch hier nicht weiter vertieft werden.

- **Start der Integrationschleife (2)**

Nachdem die erforderlichen Felder angelegt sind und einige lokale Variablen (z. B. die LAMÉ-Konstanten aus Gleichung (B.11)) berechnet und lokalen Variablen zugewiesen wurden, wird die Integrationschleife mit

```
SMSDo[iip, 1, nGP];
```

 (B.28)

eingeleitet. Ein Schleifenzähler, der Beginn und das Ende der Schleife müssen festgelegt werden. Der Befehl wird später in der gewählten Programmiersprache direkt in eine Schleife umgesetzt.

- **Aufstellen der Ansatzfunktionen (3)**

Innerhalb der Schleife müssen den lokalen Koordinaten ξ und η zunächst die Werte des aktuellen Integrationspunktes zugewiesen werden. Dies erfolgt über

$$\begin{aligned} \xi &\vdash \text{SMSFreeze}[\text{SMSPart}[\xi_{\text{GP}}, \text{iip}]]; \\ \eta &\vdash \text{SMSFreeze}[\text{SMSPart}[\eta_{\text{GP}}, \text{iip}]]; \\ w_i &\vdash \text{SMSFreeze}[\text{SMSPart}[w_{\text{GP}}, \text{iip}]]; \end{aligned} \quad (\text{B.29})$$

Hier werden die zuvor über `SMSArray` definierten Felder für den aktuellen Wert des Schleifenzählers `iip` ausgelesen. Die Ansatzfunktionen werden wie in Gleichung (B.6) angegeben aufgestellt, wobei ξ und η jetzt symbolische Variablen sind, nach denen z. B. differenziert werden kann. Die Aufstellung erfolgt über den `Table`-Befehl in MATHEMATICA, wobei dieser später im Programmcode in Einzeloperationen aufgelöst wird.

$$\text{shp} \equiv \text{Table}[1/4 (1 + \xi \xi_i[\text{i}]) (1 + \eta \eta_i[\text{i}]), \{i, 1, 4\}]; \quad (\text{B.30})$$

Im Gegensatz zur händischen Programmierung, müssen die Ableitungen der Ansatzfunktionen nicht direkt definiert werden. Die Programmierumgebung erlaubt bei Bedarf symbolische Ableitungen nach ξ oder η .

- **Interpolation (4)**

Die Interpolation von Geometrie und Verschiebungen wie in Gleichung (B.7) angegeben erfolgt symbolisch durch Matrixmultiplikation

$$\mathbb{X} \equiv \text{shp}.\text{x1}; \quad (\text{B.31})$$

$$u_i \equiv \text{shp}.\text{u1}; \quad (\text{B.32})$$

Die JACOBI-Matrix wird durch symbolische Ableitung des Ortsvektors \mathbf{X} nach dem Vektor der lokalen Koordinaten $[\xi \ \eta]^T$ mit dem Befehl `SMSD` ermittelt:

$$\mathbb{J} \equiv \text{SMSD}[\mathbb{X}, \{\xi, \eta\}]^T; \quad (\text{B.33})$$

Deren Determinante und Inverse werden über

$$\text{det}\mathbb{J} \equiv \text{SMSDet}[\mathbb{J}]; \quad (\text{B.34})$$

$$\mathbb{J}i \equiv \text{SMSInverse}[\mathbb{J}]; \quad (\text{B.35})$$

implementiert.

- **Verzerrungstensor (5)**

Die kovarianten Komponenten des in Gleichung (B.12) angegebene GREEN-LAGRANGE-Verzerrungstensors werden durch symbolische Ableitung der Verschiebungen nach den lokalen Koordinaten gebildet, wobei erneut der Befehl `SMSD` angewendet wird. Die Basisvektoren sind wie in Gleichung (B.14) angegeben zeilenweise in der JACOBI-Matrix angeordnet. Ausgehend von der händischen Programmierung bei der die in der Tensorrechnung übliche Indexschreibweise in entsprechende Programmschleifen umgesetzt wird, wurde zunächst auch in MATHEMATICA die `Do`-Funktion eingesetzt.

Die Eingabe

$$\begin{aligned}
 & \Xi = \{\xi, \eta\}; \\
 & \text{Do}[\\
 & \quad \text{Do}[\\
 & \quad \quad \mathbb{E}[[i, j]] \models \\
 & \quad \quad \quad 1/2 (\mathbb{J}[[i]] \cdot \text{SMSD}[\mathfrak{u}, \Xi[[j]]] + \text{SMSD}[\mathfrak{u}, \Xi[[i]]] \cdot \mathbb{J}[[i]] + \\
 & \quad \quad \quad \quad \text{SMSD}[\mathfrak{u}, \Xi[[i]]] \cdot \text{SMSD}[\mathfrak{u}, \Xi[[j]]) \\
 & \quad \quad \quad , \{j, 1, 2\}]; \\
 & \quad \quad , \{i, 1, 2\}]; \\
 & \quad] \\
 &]
 \end{aligned} \tag{B.36}$$

bedeutet für ACEGEN ein mehrfaches Aufrufen von `SMSD`, was sich negativ auf die Effizienz der Codeerzeugung auswirkt. Es wird daher empfohlen, die Ableitungen des Verschiebungsvektors nach den lokalen Koordinaten vorab zu erzeugen

$$\Delta \mathfrak{u} \models \text{SMSD}[\mathfrak{u}, \{\xi, \eta\}]^T; \tag{B.37}$$

und schließlich die Verzerrungskomponenten in Matrixform durch symbolische Operationen gemäß Gleichung (3.25) zu bestimmen:

$$\mathbb{E} \models 1/2 (\mathbb{J} \cdot \Delta \mathfrak{u}^T + \Delta \mathfrak{u} \cdot \mathbb{J}^T + \Delta \mathfrak{u} \cdot \Delta \mathfrak{u}^T); \tag{B.38}$$

\mathbb{E} beinhaltet jetzt die in einer 2×2 -Matrix angeordneten kovarianten Komponenten des GREEN-LAGRANGE-Verzerrungstensors. Eine Möglichkeit, die direkt auf die kartesischen Komponenten führt, ergibt sich durch die Definition innerer Ableitungen mit `SMSD` über die "Dependency" Option. Hierbei werden zunächst die kartesischen Komponenten des Verschiebungsgradienten bestimmt

$$\begin{aligned}
 & \text{SMSFreeze}[\mathbb{X}f, \mathbb{X}]; \\
 & \Delta \mathfrak{u} \models \text{SMSD}[\mathfrak{u}, \mathbb{X}f, \text{"Dependency"} \rightarrow \{\{\xi, \eta\}, \mathbb{X}f, \mathbb{J}i^T\}]^T;
 \end{aligned} \tag{B.39}$$

Durch den Befehl `SMSFreeze` werden die Einträge des Ortsvektors \mathbb{X} explizit eindeutigen Hilfsvariablen zugewiesen. Die Eindeutigkeit einer Variable ist erforderlich, um eine korrekte Behandlung einer Ableitung nach dieser durch die automatisch von ACEGEN durchgeführte Optimierung zu gewährleisten. Schließlich ergeben sich die kartesischen Komponenten des GREEN-LAGRANGE-Verzerrungstensors über

$$\mathbb{E}k \models 1/2 (\Delta \mathfrak{u} + \Delta \mathfrak{u}^T + \Delta \mathfrak{u} \cdot \Delta \mathfrak{u}^T); \tag{B.40}$$

welche dann bei Bedarf – etwa für die Anwendung der ANS-Methode – über

$$\mathbb{E} \models \mathbb{J} \cdot \mathbb{E}k \cdot \mathbb{J}^T; \tag{B.41}$$

in die konvektive Basis transformiert werden können.

Die Komponenten des Verzerrungstensor liegen jetzt in symbolischer Schreibweise vor. Modifikationen an den Verzerrungen – z. B. durch die ANS-Methode – lassen sich nun sehr "komfortabel" umsetzen, da die Interpolationsvorschriften direkt für die

Verzerrungen eingegeben werden können. Das jeweilige Vorgehen und die verwendeten ACEGEN-Befehle, sind im Hauptteil dieser Arbeit in Kapitel 4 beschrieben. Nach der Eingabe aller erforderlichen Verzerrungsmodifikationen erfolgt schließlich die Transformation in kartesische Koordinaten über

$$\mathbb{E}_k \mathbb{F} \mathbb{J}_i \cdot \mathbb{E} \cdot \mathbb{J}_i^T; \quad (\text{B.42})$$

da dies die Implementierung des Materialgesetzes erleichtert.

Der ebene Spannungszustand erfordert nun noch die separate Bestimmung der Normalverzerrung in Dickenrichtung, welche wie in Gleichung (B.13) über die Querkontraktionszahl mit E_{11} und E_{22} gekoppelt ist. Diese Besonderheit entfällt bei Kontinuums-elementen, bei denen sich der komplette 3D-Verzerrungstensor über die in Gleichung (B.38) beschriebene Eingabe bestimmen lässt.

- **Verzerrungsenergiefunktion & Spannungen (6)**

Mit den in kartesische Koordinaten transformierten Verzerrungen \mathbb{E}_k kann nun die Verzerrungsenergiefunktion direkt wie in Gleichung (B.10) angegeben implementiert werden. Es ergibt sich

$$\mathcal{W} \mathbb{F} 1/2 \lambda \text{Tr} [\mathbb{E}_k]^2 + \mu \text{Tr} [\mathbb{E}_k \cdot \mathbb{E}_k]; \quad (\text{B.43})$$

Eine Berechnung der Spannungen ist ebenso symbolisch möglich und ergibt sich durch Ableiten der Verzerrungsenergiefunktion nach den Verzerrungen. Da jedoch die Knotenkräfte direkt durch Ableitung der Energie nach den Knotenfreiheitsgraden berechnet werden, ist die Aufstellung der Spannungen innerhalb dieser Routine nicht erforderlich und erfolgt erst im Rahmen der Spannungsrückrechnung.

Die symbolische Eingabe der Verzerrungsenergiefunktion zur Beschreibung des hyperelastischen Materialverhaltens erlaubt einen sehr komfortablen Wechsel, z. B. zu einer NEO-HOOKE-Formulierung. Insbesondere bei Volumenschalenelementen, bei denen die kontinuumsmechanische Materialformulierung ohne Modifikationen angewendet werden kann, muss hier lediglich die Verzerrungsenergiefunktion geändert werden. Im Gegensatz zur händischen Implementierung ist ein Übergang auf den Stofftensor und ein entsprechendes Umsortieren in eine Matrixform nicht erforderlich.

- **Vektor der inneren Kräfte (7)**

Im Gegensatz zur händischen Implementierung müssen die einzelnen Matrizen zur Aufstellung des Vektors der inneren Kräfte nicht aufgestellt und ausgewertet werden, da eine symbolische Differentiation der inneren Energie nach den Knotenfreiheitsgraden, wie in Gleichung (B.9) möglich ist. Zunächst wird also wie unter Punkt (6) beschrieben die innere Energie am Integrationspunkt nach Gleichung (B.43) aufgestellt und schließlich zur Bestimmung des Vektors der inneren Knotenkräfte nach dem Vektor der Knotenfreiheitsgrade \mathfrak{d} abgeleitet. Anfangs wurde durch Differentiation nach dem gesamten Vektor der Freiheitsgrade der vollständige Kraftvektor bestimmt:

$$\mathfrak{f}_{\text{int}} \mathbb{F} \text{SMSD}[\mathcal{W}, \mathfrak{d}] * \text{det}\mathbb{J} * \mathfrak{w}_i * \mathfrak{t}; \quad (\text{B.44})$$

Da dieser Schritt innerhalb der Integrationsschleife implementiert ist, werden bei der Codegenerierung die Koordinaten der Integrationspunkte eingesetzt. Die Ausgabe in das globale Feld \mathfrak{p} erfolgt mit

$$\text{SMSExport}[-\mathfrak{f}_{\text{int}}, \mathfrak{p} \$, \$, \text{"AddIn"} \rightarrow \text{True}]; \quad (\text{B.45})$$

wobei die Option "AddIn" \rightarrow True eine Summation über alle Integrationspunkte bewirkt.

Auf Empfehlung von Prof. J. Korelc, dem Entwickler von ACEGEN [73], wurde die Differentiation und die Ausgabe in eine Schleife über alle Freiheitsgrade eingebettet, also eine sukzessive Ableitung der inneren Energie nach den einzelnen Freiheitsgraden gewählt. Für den inneren Knotenkraftvektor lautet die Umsetzung dann

```
SMSDo [
  fint  $\models$  SMSD[ $\mathcal{W}$ , d, i] * detJ * wi * t;
  SMSEXPORt [-fint, p$$[i], "AddIn"  $\rightarrow$  True];
, {i, 1, Length[d]}}];
```

 (B.46)

Soll zusätzlich die Elementsteifigkeitsmatrix bestimmt und über das Feld s ausgegeben werden, ergibt sich

```
SMSDo [
  fint  $\models$  SMSD[ $\mathcal{W}$ , d, i] * detJ * wi * t;
  SMSEXPORt [-fint, p$$[i], "AddIn"  $\rightarrow$  True];
  SMSDo [
    Kij  $\models$  SMSD[fint, d, j];
    SMSEXPORt [Kij, s$$[i, j], "AddIn"  $\rightarrow$  True];
  , {j, i, Length[d]}}];
, {i, 1, Length[d]}}];
```

 (B.47)

Da die innere Schleife nicht bei 1, sondern bei i beginnt, wird nur das obere Dreieck der symmetrischen Steifigkeitsmatrix ausgegeben, sie muss also später noch an der Diagonalen gespiegelt werden. Dieses Vorgehen führt im Vergleich zu Gleichung (B.44) sowohl zu deutlich weniger Programmcode, als auch zu einer erheblichen Beschleunigung der Codeerzeugung. Insbesondere bei der Aufstellung der Elementsteifigkeitsmatrix machen sich diese Effekte deutlich bemerkbar. So nimmt etwa die Übersetzung durch den Compiler weniger Zeit in Anspruch. Für die entwickelten Schalen- und Volumenschalenelemente wurde jeweils eine Routine zur ausschließlichen Bestimmung des Residuums und eine zur Aufstellung von Steifigkeitsmatrix und Residuum programmiert, da diese Trennung für die Nutzung innerhalb eines expliziten Zeitintegrationsverfahrens von Vorteil ist.

- **Ende der Integrationsschleife (8)**

Das Ende der Integrationsschleife wird durch den Befehl

```
SMSEndDo [];
```

 (B.48)

bestimmt.

- **Ausgabe in Programmcode (9)**

Der Programmcode in der zuvor bestimmten Sprache wird über den Befehl

```
SMSWrite ["LocalAuxiliaryVariables"  $\rightarrow$  True];
```

 (B.49)

erzeugt. Mit der Option "LocalAuxiliaryVariables" verhindert man die Übergabe des Feldes der Hilfsvariablen v über den Kopf der erzeugten Programmroutine. Die Hilfsvariablen werden so nur auf lokaler Ebene verwendet. Die generierte Routine kann nun direkt in das Gesamtprogramm eingebaut werden. Wie schon bei Schritt (1) gehört dieser Befehl nicht unmittelbar zum Programmablauf und ist in Abbildung B.3 daher außerhalb des eigentlichen Programms unter Punkt (9) angegeben.

B.3. Implementierung einer Materialformulierung mit Plastizität

In diesem Abschnitt wird die symbolische Umsetzung mit ACEGEN für eine Materialformulierung mit Plastizität für finite Deformationen beschrieben, wie sie im Hauptteil der Arbeit in Abschnitt 1.3.2 beschrieben und in Abbildung 1.3 skizziert ist. Die Implementierung entstand mit Unterstützung durch den Entwickler von ACEGEN, die wesentliche Zusammenhänge der gewählten Formulierung sind in dessen Publikation [72] zusammengestellt. Im Folgenden werden die wesentlichen Schritte zur Programmierung mit ACEGEN für ein Solid-Shell-Element mit linearer Geometrie- und Verschiebungsinterpolation angegeben, da an diesem Beispiel noch einige interessante Aspekte der symbolischen Programmierung gezeigt werden können. Wie im vorigen Abschnitt ist die Materialformulierung in die Programmroutine zur Aufstellung der inneren Knotenkräfte, bzw. der Elementsteifigkeitsmatrix integriert, die grundlegende Programmstruktur entspricht also der zuvor entwickelten.

Die gewählte Materialformulierung erfordert 7 Geschichtsvariablen für jeden Integrationspunkt, bei numerischer Integration mit $2 \times 2 \times 2$ GAUSS-Punkten wird der Elementroutine über den Befehl `SMSModule` also neben weiterer Materialparameter beispielsweise zur Beschreibung der Verfestigungsregel zusätzlich ein Feld `hg` der Dimension 56 übergeben. Innerhalb der Schleife für die numerische Integration wird für den Verschiebungszustand der Deformationsgradient in kartesischen Koordinaten bestimmt

```
SMSFreeze[Xf, X];
Δu1 = SMSD[u1, Xf, "Dependency" -> {{ξ, η, ζ}, Xf, Ji^T}]^T;
SMSFreeze[F, IdentityMatrix[3] + Δu1, "KeepStructure" -> True];
```

(B.50)

sowie für den aktuellen Integrationspunkt das Geschichtsfeld ausgelesen

```
hgn = SMSRealList[Array["hgn", 7], hg$$[#] &, "Length" -> ngp, "Index" -> iip];
```

(B.51)

Die Inverse des plastischen rechten CAUCHY-GREEN-Deformationstensors sowie die akkumulierte plastische Verzerrung ergeben sich zu

$$C_{pi} = \text{IdentityMatrix}[3] + \begin{pmatrix} \text{hgn}[1] & \text{hgn}[3] & \text{hgn}[6] \\ \text{hgn}[3] & \text{hgn}[2] & \text{hgn}[5] \\ \text{hgn}[6] & \text{hgn}[5] & \text{hgn}[4] \end{pmatrix};$$

$$\gamma = \text{hgn}[7];$$
(B.52)

Dieses Vorgehen ist von Vorteil, da die Geschichtsvariablen mit dem Wert 0 initialisiert sind, der Tensor C^{p-1} im elastischen Fall jedoch der Einheitstensor ist. Zur Beurteilung ob

Plastizität vorliegt, werden die Trial-Spannungen wie folgt berechnet:

```

SMSFreeze[be, F.Cpi.FT, "Ignore" -> (# == 0 &), "Symmetric" -> True];

Ibe ≡ Tr[be];   IIIbe ≡ SMSDet[be];

W = μ/2 (Ibe + 3) - μ Log[SMSSqrt[IIIbe]] + λ/2 (Log[SMSSqrt[IIIbe]])2

SMSFreeze[τ, 2 be.SMSD[W, be, "IgnoreNumbers" -> True, "Symmetric" -> True]
, "KeepStructure" -> True];

s = τ - 1/3 IdentityMatrix[3] Tr[τ];

σy ≡ σy0 + kh γ + σyInf (1 - Exp[-Δh γ]);

σVM = SMSSqrt[3/2 Total[s s, 2]];

```

(B.53)

Die Größen σ_{y0} , kh , σ_{yInf} und Δh sind Materialkonstanten, die der Elementroutine übergeben werden. Auf eine genauere Beschreibung der einzelnen verwendeten Optionen wird hier verzichtet und es sei auf das Benutzerhandbuch [70] verwiesen. Es folgt die Überprüfung, ob "Plastizität" vorliegt, durch Vergleich der berechneten Vergleichsspannung σ_{VM} mit der aktuellen Fließspannung σ_y

```

Φ = σVM - σy;
noPlasti ≡ SMSLogical[Φ < 10-8];

```

(B.54)

Die eigentliche Abfrage beginnt mit dem Fall 'keine Plastizität', bei dem die Geschichtsvariablen unverändert wieder zurückgeschrieben werden:

```

SMSIf[noPlasti];
hg = hgn;
SMSExport[hgn, SMSRealList["hgn"[_], "Export"]];

```

(B.55)

Im anderen Fall, also wenn "Plastizität" vorliegt, wird das in Abbildung 1.3 angegeben Gleichungssystem zur Erfüllung der Fließbedingung aufgestellt und mit Einbindung eines NEWTON-Verfahrens die plastischen Größen C^{p-1} und γ für den Fall $\Psi = \mathbf{0}$ ermittelt. Zunächst werden die Geschichtsvariablen umgespeichert und die Do-Schleife initialisiert

```

SMSElse[];
hgj = hgn;
SMSDo[ii, 1, 30, 1, hgj];

```

(B.56)

Es folgt die Aufstellung der Verzerrungsenergie und die Bestimmung der Vergleichsspannung im aktuellen Iterationsschritt, also mit den Größen aus hgj . Unter Verwendung der Bezeichnungen C_{pj} und γ_j , entspricht dies genau dem Vorgehen nach Gleichung (B.53) und wird daher nicht erneut angegeben. Anschließend wird das Gleichungssystem in der

Form $\Psi = \mathbf{0}$ aufgestellt:

```

Φ = σVM - σγ;
dΦdτ = SMSD[Φ, τ, "IgnoreNumbers" -> True, "Symmetric" -> True];
M = -2 (γj - γ) dΦdτ;
Z = F.Cpj - SMSMatrixExp[M, "Module" -> False].F.Cpi;
Ψ = Join[Extract[Z, {{1, 1}, {2, 2}, {1, 2}, {3, 3}, {2, 3}, {1, 3}}], {Φ = Φ}];

```

(B.57)

Unter Verwendung der symbolischen Ableitung, kann nun der NEWTON-Schritt für das Gleichungssystem durchgeführt werden:

```

A = SMSD[Ψ, hgj];
LU = SMSLUFactor[A];
Δh = SMSLUSolve[LU, -Ψ];
hgj = hgj + Δh;

```

(B.58)

Es wird überprüft, ob die Änderung Δh im aktuellen Iterationsschritt hinreichend klein ist, um ein Abbruchkriterium zu erfüllen, und die Do-Schleife wird beendet:

```

incr = SMSSqrt[Δh.Δh/Length[Δh]];
SMSIf [incr < 10-8];
  SMSBreak [];
SMSEndIf [];
SMSEndDo [hgj];

```

(B.59)

Die konvergierten plastischen Verzerrungen werden wieder als Geschichtsvariablen für den nächsten Zeitschritt abgespeichert und die If-Abfrage für das Auftreten von Plastizität beendet:

```

hg = hg - SMSFreeze [hgj, "Dependency" -> True];
SMSExport [hgj, SMSRealList ["hgn"[_], "Export"]];
SMSEndIf [hg];

```

(B.60)

Im letzten Schritt wird erneut die Verzerrungsenergie – jetzt mit den Werten aus hg – wie in Gleichung (B.53) aufgestellt. Die Aufstellung der inneren Knotenkräfte erfolgt schließlich analog zu Gleichung (B.46) durch

```

SMSDo [
  fint = SMSD[W, d, i, "Constant" -> hg] * detJ * wi;
  SMSExport [-fint, p$$[i], "AddIn" -> True];
, {i, 1, Length[d]};

```

(B.61)

wobei durch die Option "Constant" -> hg eine Ableitung nach den in den plastischen Verzerrungen enthaltenen Freiheitsgraden verhindert wird. Die konvergierten Größen werden also am Integrationspunkt als konstant angenommen.

Die oben beschriebene Implementierung des Materialmodells zeigt beispielsweise, dass die Umsetzung eines NEWTON-Verfahrens mit Hilfe der symbolischen Eigenschaften von ACEGEN in wenigen Programmzeilen implementiert werden kann. Die dafür erforderlichen Ableitungen teilweise recht komplexer Funktionen müssen hierfür nicht explizit aufgestellt werden, wodurch sich Änderungen an der Formulierung sehr schnell und fehlerfrei realisieren lassen. Das Vorgehen lässt sich prinzipiell auch auf komplexere Materialformulierungen anwenden.

C. Beispiel zur Diagonalisierung der Elementmassenmatrix

Im Folgenden werden anhand eines 1-Element Beispiels die in Abschnitt 3.3 vorgestellten Strategien zur Aufstellung einer diagonalisierten Elementmassenmatrix diskutiert. Wie bereits angesprochen unterscheiden sich die Resultate für Elemente mit linearer Ansatzordnung nicht, weshalb hier die quadratische Solid-Shell-Formulierung mit 18 Knoten verwendet wird. Da das für die Massenmatrix zu integrierende Polynom höherer Ordnung ist als etwa bei der Steifigkeitsmatrix, wird auch die Integrationsordnung, d. h. die erforderliche Zahl der Integrationspunkte für eine numerische Integration diskutiert.

C.1. Konsistente Massenmatrix

Die meisten der in Abschnitt 3.3.3 besprochenen Diagonalisierungsstrategien für die Elementmassenmatrix basieren auf der in Gleichung (3.38) angegebenen konsistenten Form. Durch Abbildung auf das Einheitselement und Integration über die natürlichen Koordinaten folgt aus Gleichung (3.38)

$$\mathbf{M}_c^e = \rho \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} \mathbf{N}^T \mathbf{N} \det \mathbf{J} d\xi d\eta d\zeta. \quad (\text{C.1})$$

Im Gegensatz zur Elementsteifigkeitsmatrix berechnet sich der Integrand aus den Ansatzfunktionen direkt. Er besitzt also eine höhere Polynomordnung, was im Allgemeinen eine höhere Integrationsordnung erfordert, um exakte Integration zu gewährleisten. Da sich die Massenmatrix im Laufe der in dieser Arbeit behandelten strukturmechanischen Untersuchungen als konstant angenommen wird, d. h. nur eine Berechnung der einzelnen Terme zu Beginn der Simulation erforderlich ist, spielen Effizienz Aspekte bei der Wahl der Integrationsordnung keine große Rolle. Wie viele Integrationspunkte für die korrekte Repräsentation der Masse erforderlich sind, soll ebenfalls am folgenden Beispiel diskutiert werden.

Es werden zwei verschiedene 18-Knoten-Einzelemente untersucht, welche in Abbildung C.1 dargestellt sind. Die Geometrie- und Verschiebungsinterpolation erfolgt bei dieser Elementformulierung mit quadratischen LAGRANGE-Polynomen in der ξ - η -Ebene und linearen LAGRANGE-Polynomen in ζ -Richtung. Die zugehörigen Koordinaten der Elementknoten sind in Tabelle C.1 angegeben.

Um eine anschauliche Darstellung der Matrizen der Dimension 54 zu erreichen, wird die Funktion `MatrixPlot` des Computeralgebraprogramms MATHEMATICA verwendet. Die Einträge der Matrizen werden durch Graustufen dargestellt, wobei *weiß* für die kleinsten und *schwarz* für die größten Einträge steht.



(a) regelmäßige Elementgeometrie

(b) unregelmäßige Elementgeometrie

Abbildung C.1.: Betrachtete Elementgeometrien mit den in Tabelle C.1 gegebenen Knotenkoordinaten

Knotennummer	regelmäßige Geometrie			unregelmäßige Geometrie		
	x	y	z	x	y	z
1	0.000	0.000	0.000	0.000	0.000	0.000
2	2.000	0.000	0.000	2.000	0.000	0.000
3	2.000	2.000	0.000	3.000	2.500	0.000
4	0.000	2.000	0.000	-0.250	2.500	0.000
5	1.000	0.000	0.000	1.000	0.000	0.000
6	2.000	1.000	0.000	2.500	1.250	0.000
7	1.000	2.000	0.000	1.000	2.000	0.000
8	0.000	1.000	0.000	-0.125	1.250	0.000
9	1.000	1.000	0.000	1.000	1.000	0.000
10	0.000	0.000	1.000	0.000	0.000	1.000
11	2.000	0.000	1.000	2.000	0.000	1.000
12	2.000	2.000	1.000	2.500	2.250	1.000
13	0.000	2.000	1.000	-0.500	3.000	1.000
14	1.000	0.000	1.000	1.000	0.000	1.000
15	2.000	1.000	1.000	2.250	1.125	1.000
16	1.000	2.000	1.000	1.000	2.000	0.500
17	0.000	1.000	1.000	-0.250	1.500	1.000
18	1.000	1.000	1.000	1.000	1.000	0.750

Tabelle C.1.: Knotenkoordinaten der beiden betrachteten Elementgeometrien

C.2. 1-Element Beispiel – regelmäßige Geometrie

Zunächst erfolgt die Berechnung der verschiedenen Massenmatrizen für den einfachen Fall einer regelmäßigen Elementgeometrie, wie sie bei der Vernetzung regelmäßig berandeter Strukturen üblicherweise vorkommt. Da die Ansatzfunktionen für jedes Element gleich sind, ist der bestimmende Anteil für die Polynomordnung des Integranden aus Gleichung (C.1) und somit für die Integrationsordnung die Determinante der JACOBI-Matrix. Die in Tabelle C.1 gegebenen Knotenkoordinaten führen für das Element zu einer konstanten Determinante

$$\mathbf{X} = \begin{pmatrix} 1 + \xi \\ 1 + \eta \\ \frac{1+\zeta}{2} \end{pmatrix} \Rightarrow \det \mathbf{J} = \det \frac{\partial \mathbf{X}}{\partial \xi} = \frac{1}{2}. \quad (\text{C.2})$$

Demzufolge besitzt der Integrand aus Gleichung (C.1) die Ordnung 4 in ξ und η , bzw. 2 in ζ . Eine exakte Integration erfordert also $3 \times 3 \times 2$ GAUSS-Punkte, da

$$\left. \begin{array}{l} 2 \cdot n_{\xi} - 1 \\ 2 \cdot n_{\eta} - 1 \end{array} \right\} = 2 \cdot 3 - 1 = 5 > 4, \quad \text{bzw.} \quad 2 \cdot n_{\zeta} - 1 = 2 \cdot 2 - 1 = 3 > 2 \quad (\text{C.3})$$

Um den Unterschied zwischen analytisch und numerisch integrierten Massenmatrizen zu bewerten, wird die Norm

$$\frac{\|\mathbf{A}\| - \|\mathbf{B}\|}{\|\mathbf{A}\|}, \quad \text{mit} \quad \|\mathbf{A}\| = \|\mathbf{A}\|_2 = \sqrt{\lambda_{\max}(\mathbf{A}^T \cdot \mathbf{A})} \quad (\text{C.4})$$

eingeführt, deren Auswertung für verschiedene Integrationsordnungen in Abbildung C.2 grafisch aufbereitet ist. Es wird in Gleichung (C.4) für \mathbf{A} die analytisch integrierte und für \mathbf{B} eine numerisch integrierte konsistente Massenmatrix eingesetzt. Bei $2 \times 2 \times 2$ Integrationspunkten ist noch eine deutliche Abweichung zu erkennen, zwischen den Ergebnissen mit $3 \times 3 \times 2$ und $4 \times 4 \times 2$ Integrationspunkten besteht jedoch erwartungsgemäß kein Unterschied mehr. In allen drei Fällen wird die Gesamtmasse korrekt berechnet, es gilt also für jede der gewählten Integrationsordnungen

$$\sum_{i=1}^{54} \sum_{j=1}^{54} [\mathbf{M}_c^e]_{ij} = 3 m^e \quad (\text{C.5})$$

Beim Vergleich der verschiedenen diagonalisierten Massenmatrizen, werden die in Abschnitt 3.3.3 vorgestellten Algorithmen angewendet. Bis auf die ‘nodal quadrature’ Methode basieren alle Strategien auf der konsistenten Massenmatrix, die hier durch eine $3 \times 3 \times 2$ GAUSS-Integration aufgestellt wird. Bei ‘nodal quadrature’ ist die Integrationsordnung durch die Lage der Knotenpunkte bestimmt, es kann also nur mit $3 \times 3 \times 2$ GAUSS-LOBATTO Punkten

‘analytisch’ – ‘ $2 \times 2 \times 2$ ’	182.37 · 10 ⁻³
‘analytisch’ – ‘ $3 \times 3 \times 2$ ’	0.00 · 10 ⁻³
‘analytisch’ – ‘ $4 \times 4 \times 2$ ’	0.00 · 10 ⁻³

Abbildung C.2.: Grafische Darstellung der in Gleichung (C.4) eingeführten Matrixnorm für analytisch und numerisch integrierte konsistente Massenmatrizen

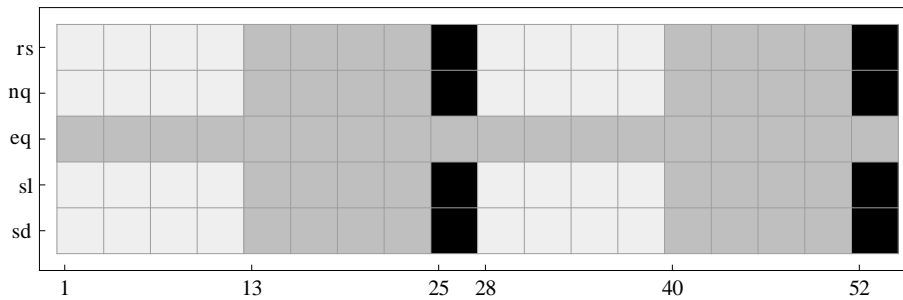


Abbildung C.3.: Darstellung der verschiedenen diagonalisierten Massenmatrizen – regelmäßige Elementgeometrie

integriert werden, um eine Diagonalmatrix zu erhalten. Diese Integrationsvorschrift besitzt die Genauigkeit $2 \cdot n - 3$, liefert also nur für Polynome der Ordnung 3 ein exaktes Ergebnis.

Abbildung C.3 zeigt eine Darstellung der verschiedenen Diagonalmatrizen, die jeweils als Zeilenvektor angeordnet sind. Ein Kästchen repräsentiert hierbei die Einträge eines Knotens, da sich für die drei Verschiebungsfreiheitsgrade jeweils die gleichen Wert ergeben. Die verwendeten Abkürzungen am Anfang jeder Zeile stehen für den jeweiligen in Abschnitt 3.3.3 beschriebenen Diagonalisierungsalgorithmus. Diese sind im Einzelnen:

'rs'	– 'row sum'	– zeilenweise Summation
'nq'	– 'nodal quadrature'	– Integration an den Elementknoten
'eq'	– 'equal lumping'	– Gleichmäßige Verteilung
'sl'	– 'special lumping'	– proportionale Skalierung der Diagonalelemente
'sd'	– 'scaled diagonals'	– gewichtete Skalierung der Diagonalelemente

In Abbildung C.3 ist sehr deutlich zu erkennen, dass – bis auf die gleichmäßige Verteilung der Elementmasse ('eq') – alle Diagonalisierungsstrategien zu identischen Massenmatrizen führen. Die Wahl der Methode zur Berechnung einer diagonalisierten Massenmatrix, spielt also bei regelmäßigen Elementgeometrien keine Rolle. Die Strategie, die Elementmasse gleichmäßig auf die Knoten zu verteilen führt bei linearen Ansätzen ebenfalls auf die richtige Massenmatrix, bei höherer Ansatzordnung liefert diese Methode ein abweichendes Ergebnis. Im vorliegenden Beispiel wurden die Skalierungsfaktoren α_k für die Methode 'scaled diagonals' zu

$$\alpha_1 = \alpha_2 = \alpha_3 = \frac{m \cdot \text{ndf}}{\sum_{i=1}^{\text{nst}} [\mathbf{M}_c^e]_{ii}} \quad (\text{C.6})$$

gewählt, sodass sich automatisch die gleiche Matrix wie bei 'special lumping' ergibt. Im Rahmen der in Abschnitt 3.3.3 genannten Grenzen, können hier beliebige diagonalisierte Massenmatrizen erstellt werden, es lassen sich also durchaus Abweichungen von den anderen Algorithmen erzielen.

C.3. 1-Element Beispiel – unregelmäßige Geometrie

Als zweites Beispiel wurde eine unregelmäßige Elementgeometrie gewählt, wie sie etwa bei der Vernetzung komplexer Strukturen mit gekrümmten Rändern auftreten kann. Abbildung C.1b zeigt die Elementgeometrie, wobei in der gewählten Darstellung Geraden für die Verbindung der Knoten verwendet wurden. Tatsächlich ist die Geometrie wie bereits erwähnt durch quadratische Polynome approximiert, wodurch sich in diesem Fall gekrümmte Elementkanten ergeben. Im Unterschied zum ersten Beispiel mit regelmäßiger Geometrie liegt hier ein höherer Polynomgrad des Integranden aus Gleichung (C.1), bedingt durch die Determinante der JACOBI-Matrix vor. Es ist

$$\mathbf{X} = \frac{1}{32} \begin{pmatrix} 32 + 41\xi - \zeta\xi + 9\eta\xi - \zeta\eta\xi + 3\xi^2 - 3\zeta\xi^2 + 3\eta\xi^2 - 3\zeta\eta\xi^2 \\ 32 + 32\eta - 3\xi - 3\zeta\xi - 3\eta\xi - 3\zeta\eta\xi + 9\xi^2 + \zeta\xi^2 + 9\eta\xi^2 + \zeta\eta\xi^2 \\ 12 + 12\zeta - 4\eta - 4\zeta\eta + 4\xi^2 + 4\zeta\xi^2 + 4\eta\xi^2 + 4\zeta\eta\xi^2 \end{pmatrix}, \quad (\text{C.7})$$

und damit

$$\det \mathbf{J} = \det \frac{\partial \mathbf{X}}{\partial \xi} \in \mathcal{O}(\xi^4, \eta^2, \zeta^2). \quad (\text{C.8})$$

Andere unregelmäßige Geometrien würden auch zu einer höheren Ordnung in η führen, weshalb sinnvollerweise die gleiche Anzahl Integrationspunkte für die beiden Ebenenrichtungen verwendet wird. Eine exakte numerische Integration des Integranden für diese Elementgeometrie erfordert also für dieses Beispiel $5 \times 5 \times 3$ Integrationspunkte, da dann

$$\left. \begin{array}{l} 2 \cdot n_\xi - 1 \\ 2 \cdot n_\eta - 1 \end{array} \right\} = 2 \cdot 5 - 1 = 9 > 8, \quad \text{bzw.} \quad 2 \cdot n_\zeta - 1 = 2 \cdot 3 - 1 = 5 > 4 \quad (\text{C.9})$$

gilt. Bei ganz allgemeiner Geometrievorgabe kann die Ordnung von $\det \mathbf{J}$ in ξ und η sogar 5 betragen, was noch einen weiteren Integrationspunkt in jede Richtung erfordern würde.

Abbildung C.4 zeigt erneut eine grafische Darstellung der Ergebnisse der in Gleichung (C.4) eingeführten Matrixnorm für analytische Integration und numerische Integration mit unterschiedlichen Integrationsordnungen. Es ist zu erkennen, dass bereits eine Integration mit $3 \times 3 \times 3$ Integrationspunkten zu einem Ergebnis führt, welches sehr nahe an der analytischen Integration ist. Bei $4 \times 4 \times 3$, bzw. $5 \times 5 \times 3$ Integrationspunkten ist praktisch kein Unterschied mehr zu erkennen, eine sehr hohe Integrationsordnung scheint also auch hier nicht erforderlich zu sein. Eine Integration mit $2 \times 2 \times 2$ Integrationspunkten ist erwartungsgemäß wie bei der regulären Geometrie (vgl. Abbildung C.2) nicht ausreichend.

Interessant ist hier die Betrachtung der resultierenden Masse, also die Auswertung von Gleichung (C.5) für die verschiedenen Integrationsordnungen. Während bei $2 \times 2 \times 2$ Integrationspunkten die Gesamtmasse um 2.64‰ unterschätzt wird, liefert bereits die nächsthöhere





'analytisch' – '2 × 2 × 2'		119.70 · 10 ⁻³
'analytisch' – '3 × 3 × 3'		10.07 · 10 ⁻³
'analytisch' – '4 × 4 × 3'		0.24 · 10 ⁻³
'analytisch' – '5 × 5 × 3'		0.00 · 10 ⁻³

Abbildung C.4.: Grafische Darstellung der in Gleichung (C.4) eingeführten Matrixnorm für analytisch und numerisch integrierte konsistente Massenmatrizen

Ordnung $3 \times 3 \times 3$ keinen Unterschied im Vergleich zur analytischen Integration, die Gesamtmasse wird also korrekt berechnet. Dies wirkt sich insbesondere auf die Qualität der diagonalisierten Massenmatrizen aus, da sich bei guter Repräsentation der Gesamtmasse unabhängig von der Verteilung innerhalb der konsistenten Massenmatrix, immer eine gleiche Form der Diagonalmatrix ergibt. Es sei angemerkt, dass auch eine 2-Punkt Integration in Dickenrichtung, also $3 \times 3 \times 2$ Integrationspunkte zur korrekten Gesamtmasse führt. Es reicht also die gleiche Integrationsordnung aus, wie sie zur Integration der Steifigkeitsmatrix, bzw. des Elementresidualvektors angewendet wird.

Wie im vorigen Beispiel für die regelmäßige Geometrie, sind in Abbildung C.5 die diagonalisierten Massenmatrizen aus den unterschiedlichen Diagonalisierungsstrategien abgebildet. Die Grundlage für die auf der konsistenten Massenmatrix basierenden Strategien ist wie zuvor diskutiert die durch Integration an $3 \times 3 \times 2$ GAUSS-Punkten ermittelte. Wie schon im Abschnitt zuvor angesprochen besteht bei der ‘nodal quadrature’ Methode keine Möglichkeit der Erhöhung der Integrationsordnung, sodass auch hier $3 \times 3 \times 2$ GAUSS-LOBATTO-Punkte verwendet werden.

Wie im Beispiel mit regelmäßiger Elementgeometrie wurden die Skalierungsfaktoren α_k für die Methode ‘scaled diagonals’ (sd) so gewählt, dass sich die resultierende Matrix nicht von der durch ‘special lumping’ (sl) berechneten unterscheidet. Die Unterschiede zwischen den Methoden ‘row sum’ (rs), ‘nodal quadrature’ (nq) und ‘special lumping’ (sl) lassen sich am besten durch die Quotienten der einzelnen Einträge verdeutlichen, welche in Abbildung C.6 angegeben sind. Es ist erkennbar, dass zeilenweise Summation und proportionale, bzw. gewichtete Skalierung der Diagonalelemente auf sehr ähnliche Massenverteilungen führen, was in der ersten Zeile von Abbildung C.6 durch nahe bei 1 liegende Quotienten repräsentiert wird. Integration an den Elementknoten hingegen ergibt eine deutlich abweichenden Massenmatrix, die durch Grautöne dargestellten Quotienten liegen im Bereich 0.5 – 1.5. Welche dieser Massenmatrizen besser für eine dynamisch Simulation geeignet ist, ist an geeigneten numerischen Beispielen zu untersuchen, sinnvoll ist auch die Betrachtung der Eigenfrequenzen und Eigenformen sowie deren Beeinflussung durch die verschiedenen Massenmatrizen. Die Methode der gleichmäßigen Verteilung der Elementmasse sei hier nur der Vollständigkeit halber angegeben, da sie wie schon im vorigen Beispiel angesprochen für höhere als lineare Ansatzordnungen ungeeignet ist, die sich ergebenden Massenmatrix also sehr stark von denen der anderen Vorgehensweisen abweicht.

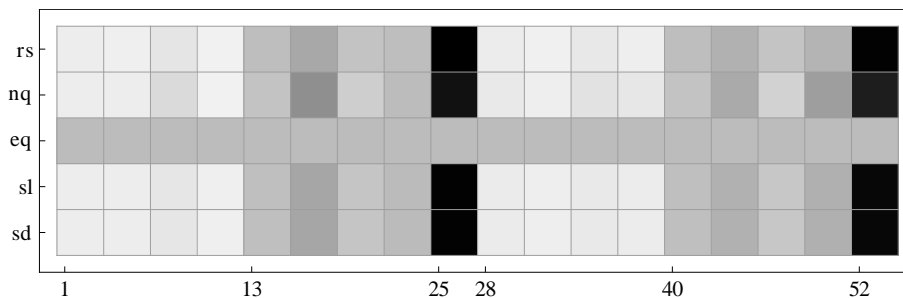


Abbildung C.5.: Darstellung der verschiedenen diagonalisierten Massenmatrizen – unregelmäßige Elementgeometrie

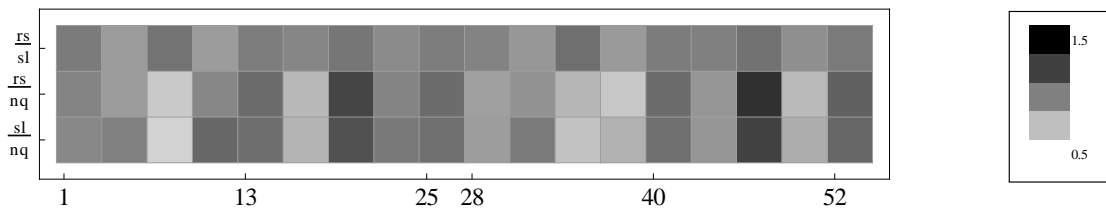


Abbildung C.6.: Darstellung der Verhältnisse der Massenmatrizen zueinander

C.4. Folgerung für die Integration der Massenmatrix

Die Untersuchung zeigt, dass eine besonders genaue Integration der Massenmatrix nicht erforderlich ist, da sich selbst bei stark unregelmäßiger Elementgeometrie kaum Änderungen bei der diagonalisierten Massenmatrix ergeben. Es genügt also, für die Massenmatrix die gleiche Integrationsordnung zu verwenden wie für die Steifigkeitsmatrix, auch wenn der Polynomgrad der zu integrierenden Funktion deutlich höher ist. Nichtsdestotrotz hat eine höhere Integrationsordnung auf die Effizienz der Elementformulierung praktisch keinen Einfluss, da die Massenmatrix nur zu Beginn einer Simulation einmal berechnet wird. Interessant wäre noch eine Untersuchung hinsichtlich der Auswirkungen der konsistenten Massenmatrix, wenn diese z. B. innerhalb eines NEWMARK-Verfahrens verwendet wird. Hier wirkt sich die unterschiedliche Verteilung der Gesamtmasse innerhalb der konsistenten Massenmatrix, also der unterschiedliche Einfluss auf die einzelnen Elementknoten möglicherweise deutlicher aus, als bei der Verwendung diagonalisierter Matrizen. Bekanntermaßen lässt sich das dynamische Verhalten am Besten über eine Eigenwertanalyse bewerten, wie sie auch in Abschnitt 3.4.5 durchgeführt wurde.

Lebenslauf

Name	Steffen Mattern
Geburtsdatum	23. Oktober 1979
Geburtsort	Neustadt a. d. Weinstraße, Deutschland
Familienstand	verheiratet mit Susanne Mattern, geb. Schmalenberg, eine Tochter Sophia Anna Mattern
Schulbesuch	1986 – 1990 Hans-Geiger Grundschule Neustadt a. d. Weinstraße 1990 – 1999 Leibniz Gymnasium Neustadt a. d. Weinstraße Abschluss mit allgemeiner Hochschulreife
Wehrersatzdienst	1999 – 2000 Zivildienst in einer Tagesförderstätte für geistig behinderte, erwachsene Menschen (Lebenshilfe e. V., Neustadt)
Studium	2000 – 2005 Bauingenieurwesen an der Universität Karlsruhe (TH) Vertiefungsrichtung: Konstruktiver Ingenieurbau Abschluss als Diplom-Ingenieur (Februar 2005)
Beruf	2005 – 2012 akademischer Mitarbeiter am Institut für Mechanik am Karlsruher Institut für Technologie (ehemals Universität Karlsruhe) seit 2012 Berechnungsingenieur bei DYNAmore GmbH in Stuttgart-Vaihingen