

Lehrstuhl für Bauinformatik
Fakultät für Bauingenieur- und Vermessungswesen
Technische Universität München

**Berechnung der elastischen Rückfederung von Tiefziehbauteilen
mit der p -Version der Finite-Elemente-Methode**

Alexander Muthler

Vollständiger Abdruck der von der Fakultät für Bauingenieur- und Vermessungswesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. K.-U. Bletzinger

Prüfer der Dissertation:

1. Univ.-Prof. Dr. rer. nat. E. Rank
2. Univ.-Prof. Dr.-Ing. St. M. Holzer,
Universität der Bundeswehr München

Die Dissertation wurde am 07.04.2005 bei der Technischen Universität München eingereicht und durch die Fakultät für Bauingenieur- und Vermessungswesen am 22.06.2005 angenommen.

Für Julita.

Zusammenfassung

Die Blechumformung ist ein wichtiges Verfahren zur wirtschaftlichen Produktion von Blechteilen z. B. für den Automobilbau. Neben der Riss- und Faltenbildung stellt dabei die elastische Rückfederung, welche bei der Entnahme der Ziehteile aus der Presse aufgrund der Relaxation der elastischen Spannungen auftritt, eine Problematik im Entwicklungsprozess der Ziehanlage dar. Die Finite-Elemente-Methode hat sich in den letzten Jahren zu einem unverzichtbaren Werkzeug bei der virtuellen Entwicklung dieser Ziehanlagen entwickelt. Durch Berechnung der elastischen Rückfederung können mit entsprechenden Kompensationsmethoden die unerwünschten Effekte eliminiert werden.

Numerische Beispiele haben allerdings gezeigt, dass diese Berechnung mit expliziten Programmen oft zu ungenauen Ergebnissen und bei einer zuverlässigeren impliziten Berechnung zu einem hohen Bedarf an Rechenzeit und Speicherplatz führt. Weiterhin sind die Annahmen der Schalentheorie nicht immer in allen Bereichen des Bauteils erfüllt.

Basierend auf einer Volumendiskretisierung hoher Ordnung wird in dieser Arbeit eine effiziente und zuverlässige Berechnung der elastischen Rückfederung vorgeschlagen. Durch Formulierung mit Hexaederelementen können die Polynomgrade für jede der drei lokalen Elementkoordinaten und für jede Komponente des Verschiebungsfeldes unterschiedlich gewählt werden. Der Modellfehler, der jeder Schalentheorie anhaftet, wird bei dieser Methode durch einen besser beherrschbaren Diskretisierungsfehler ersetzt. Da bei der p -Version der FEM der Elementdurchmesser unverändert bleibt, wird die Blending-Funktionen-Methode zur genauen Beschreibung der Geometrie verwendet. Numerische Beispiele demonstrieren die Leistungsfähigkeit des Verfahrens.

Abstract

Metal forming is an economical method to produce sheet metals e. g. for the automotive industry. One important feature of forming metals is its spring back behavior, which means that the workpiece tends to relax the residual stresses from the plastic deformation stage by partly reversing its acquired shape. The finite element analysis of sheet metal forming processes is highly developed and frequently applied in deep drawing computations. By computing the elastic spring back, undesired effects can be eliminated by compensation methods.

Numerical examples have shown that the computation of elastic spring back based on explicit finite element codes may yield unreliable results and the more reliable analysis with implicit codes is highly demanding in terms of computer resources. Moreover, it is questionable if the assumptions of the underlying shell-theory are fulfilled in the whole computational domain.

To overcome these problems a new approach which allows to compute efficient and reliable approximations of the elastic spring back is presented in this thesis. It is based on a strictly three-dimensional high order solid finite element formulation for curved thin-walled structures. A hexahedral element is applied, allowing for an anisotropic Ansatz of the displacement field, where the polynomial degree of each separate component can be chosen individually and may also be varied in the three local directions of the element. The model error, inherent in each shell-theory turns into a discretization error, which can be readily controlled. Curved boundaries are taken care of by applying the blending function method. Several numerical examples demonstrate the efficiency and reliability of the proposed approach.

Vorwort

Die vorliegende Arbeit entstand in den Jahren 2002 bis 2005 im Rahmen eines von der BMW Group gewährten Industriestipendiums während meiner Tätigkeit am Lehrstuhl für Bauinformatik an der Technischen Universität München.

An dieser Stelle möchte ich mich bei all denen bedanken, die zum Gelingen dieser Arbeit beigetragen haben. Mein besonderer Dank gilt Herrn Professor Dr. Ernst Rank für die sehr gute Betreuung und großzügige Förderung dieser Arbeit. Seine Vorschläge und Anregungen haben die Richtung meiner Arbeit maßgeblich beeinflusst.

Herrn Professor Dr. Stefan M. Holzer danke ich für das Interesse an dieser Arbeit und für die Übernahme des zweiten Gutachtens.

Den beiden „Vätern“ des Projektes, Dr. Alexander Düster und Dr. Wolfram Volk, möchte ich für das Engagement und die zahlreichen Vorarbeiten danken, durch welche dieses Projekt ins Leben gerufen wurde. Während der letzten drei Jahre standen sie mir mit Rat und Tat zur Seite und haben mich durch viele Fachgespräche maßgeblich unterstützt.

Bei der BMW Group bedanke ich mich für das mir durch das Promotionsstipendium entgegengebrachte Vertrauen, ohne das diese Arbeit nicht zustande gekommen wäre. Insbesondere möchte ich mich hier bei Ingo Heinle und Dr. Marcus Wagner für das Interesse und die gute Zusammenarbeit sowie bei allen Mitarbeitern der Abteilung TK-411 für das freundliche Arbeitsklima recht herzlich bedanken.

Den Kolleginnen und Kollegen des Lehrstuhls für Bauinformatik danke ich für die offene und herzliche Arbeitsatmosphäre. Besonders bedanken möchte ich mich bei meinem Zimmerkollegen Richard Romberg für die zahlreichen fachlichen Diskussionen und die gute gemeinsame Zeit im Büro.

Meinen Eltern, Inge und Wolfgang Muthler, danke ich für ihre stetige Unterstützung welche sie mir zu jeder Zeit entgegengebracht haben.

Ganz besonders bedanken möchte ich mich bei meiner Lebensgefährtin Julita, die mir durch ihr Verständnis und viel Liebe den nötigen Rückhalt gegeben hat.

Inhaltsverzeichnis

Inhaltsverzeichnis	i
1 Einleitung	1
1.1 Motivation	1
1.2 Inhaltsübersicht	2
2 Grundlagen der Umformtechnik	5
2.1 Grundbegriffe des Tiefziehens	6
2.1.1 Materialbeschreibung	7
2.2 Grundlagen der elastischen Rückfederung	10
2.2.1 Definition	10
2.2.2 Kompensation der elastischen Rückfederung	10
2.2.3 Modellierung	13
3 Die p-Version der Finite-Elemente-Methode	15
3.1 Einführung	15
3.2 Dreidimensionale lineare Elastizitätsprobleme	16
3.2.1 Gleichgewicht	16
3.2.2 Kinematik	17
3.2.3 Stoffgesetz	17
3.2.4 Randbedingungen	18
3.3 Das Prinzip der virtuellen Arbeit	18
3.4 Die Finite-Elemente-Approximation	20
3.4.1 Abbildungsfunktionen	22
3.5 Die p -Version der Finite-Elemente-Methode	23
3.5.1 Hierarchische Ansatzfunktionen für eindimensionale Elemente	24
3.5.2 Hierarchische Ansatzfunktionen für Hexaeder-Elemente	25
3.6 Geometrisch nichtlineare Probleme	28
3.6.1 Greenscher Verzerrungstensor	29
3.6.2 Linearisierung der virtuellen Arbeit	30
3.6.3 Inkrementelle Lösung	32
3.6.4 Das Newton-Raphson-Verfahren	33
3.6.5 Inkrementelles Newton-Raphson-Verfahren	34
3.6.6 Modifiziertes Newton-Raphson-Verfahren	34
3.6.7 Erhöhung der Effizienz für statisch bestimmte Probleme	35

4	Grundlagen der geometrischen Modellierung	37
4.1	Topologie des Boundary-Representation-Modells	37
4.1.1	Der <i>vef</i> -Graph	38
4.1.2	Euler-Operatoren	38
4.1.3	Graphenstruktur in einem objektorientierten Geometriemodellierer	39
4.2	Geometrische Attributierung	42
4.2.1	Freiformkurven	43
4.2.2	Freiformflächen	48
4.3	Interpolation und Approximation	49
4.3.1	Interpolation von Kurven	49
4.3.2	Interpolation von Flächen	50
4.3.3	Approximation	50
4.3.4	Approximation von Kurven	51
4.3.5	Approximation von Flächen	52
4.3.6	Approximation mit spezifischer Genauigkeit	52
4.4	Boolesche Operationen	54
4.4.1	Aufbau des Verschneidungsgraphen	54
4.4.2	Verschneidung Fläche-Fläche	54
4.4.3	Randklassifizierung	56
4.4.4	Regularisierte und nicht-regularisierte Boolesche Operationen	56
4.5	Die Blending-Funktionen-Methode	56
4.5.1	Zweidimensionale Blending-Funktionen-Methode	57
4.5.2	Dreidimensionale Blending-Funktionen-Methode	58
4.6	Beschreibung der Geometrie für die Blending-Funktionen-Methode	60
4.6.1	Quasi-regionale Abbildung	60
4.6.2	Abbildung von kontinuierlicher auf transfinite Form	61
5	Kopplung der Simulationsverfahren	67
5.1	Motivation	67
5.2	Volumenrückführung	68
5.2.1	Extrusion des Schalennetzes	68
5.2.2	Einteilung der Regionen	69
5.2.3	Zuordnung Schalenelemente - Patch	71
5.2.4	Flächenrückführung	71
5.2.5	Projektion der Berandung	74
5.2.6	Erzeugung des Volumenkörpers	75
5.3	Automatische Netzgenerierung für die <i>p</i> -Version der Finite-Elemente-Methode	76
5.3.1	Gebietsteilungstechnik	77
5.3.2	Konvertierung des Dreiecksnetzes in ein Vierecksnetz	77
5.3.3	Adaptive Netzgenerierung	78
5.3.4	Netzgenerierung auf Parameterflächen	81
5.3.5	Datenstruktur	84
5.3.6	Konvertierung des Vierecksnetzes in ein gekrümmtes Hexaedernetz	85
5.4	Datentransfer	86
5.4.1	Koordinatentransformation	87
5.4.2	Integrationspunkte	88

5.4.3	Aufbau eines Space-Tree	88
5.4.4	Effiziente Punktsuche	90
5.4.5	Horizontale Interpolation	91
5.4.6	Vertikale Interpolation	93
5.5	Effiziente Integrationsmethoden	94
5.6	Gleichungslöser	95
5.6.1	Direkte Gleichungslöser	96
5.6.2	Iterative Gleichungslöser	96
5.6.3	Anmerkungen zur Wahl des Gleichungslösers	97
5.7	Ansatz zur Berechnung der erforderlichen Pressenkraft	98
5.7.1	Motivation	98
5.7.2	Erweiterungen des Modells	100
6	Anwendungs-Beispiele	103
6.1	U-Profil	104
6.1.1	Geometrisch nichtlineare Berechnung der Rückfederung	104
6.2	S-Rail	108
6.2.1	Konvergenz der p -Version	108
6.2.2	Anisotrope Ansatzräume	111
6.2.3	p -Adaptivität	112
6.2.4	Geometrische Nichtlinearität	115
6.3	Wasserstoff-Tank	117
6.3.1	Genauigkeit	117
6.3.2	Konvergenz der p -Version	118
6.4	Längsträger	120
6.4.1	Ansatzräume	120
6.4.2	Geometrische Nichtlinearität	124
7	Zusammenfassung	127
	Literaturverzeichnis	129

Kapitel 1

Einleitung

1.1 Motivation

Als eine wichtige Variante des Umformens stellt das Tiefziehen einen Prozess dar, bei dem die ursprüngliche Form eines festen Körpers in eine neue Form unter Beibehaltung der Masse übergeführt wird. Wesentlich für die Qualität und vor allem die Kosten der Bauteile ist die Entwicklung präziser Umformwerkzeuge. Beim Umformen kommt es dabei zu mehr oder weniger von Prozessparametern abhängigen, unerwünschten Effekten wie Riss- und Faltenbildung sowie zur elastischen Rückfederung. Die elastische Rückfederung entsteht beim Wegfahren der Werkzeuge durch Relaxation der im Bauteil vorliegenden elastischen Spannungen.

Während die Umformsimulation von Blechteilen ein weit entwickeltes und in der Praxis zuverlässig eingesetztes Verfahren ist, stellt die daran anschließende Berechnung der elastischen Rückfederung nach wie vor ein nicht zufriedenstellend gelöstes Problem dar. Die Bestimmung der elastischen Rückfederung ist jedoch von besonderem Interesse, da sie Aufschluss über die Geometrie der einzusetzenden Umformwerkzeuge gibt. Numerische Experimente haben gezeigt, dass die Berechnung der elastischen Rückfederung mit expliziten Finite-Elemente-Programmen oft keine zuverlässigen Ergebnisse liefert. Eine Berechnung mit impliziten Programmen führt zwar zu besseren Ergebnissen, diese Simulationen sind jedoch aufgrund des in der Praxis üblicherweise verwendeten Diskretisierungsansatzes (h -Version) und der gewählten numerischen Umsetzung der geometrischen Nichtlinearität sehr zeitintensiv. Ferner ist nicht immer sichergestellt, dass die der Elementformulierung zugrunde liegenden vereinfachenden Annahmen der Schalentheorie in allen Bereichen des Bauteils, wie z. B. in engen Radien hinreichend genau erfüllt sind.

Im Rahmen des dieser Arbeit zugrunde liegenden Forschungsprojektes wurde die Berechnung der elastischen Rückfederung mithilfe einer strikt dreidimensionalen Diskretisierung hoher Ordnung (p -Version) durchgeführt. Der Modellfehler, der jeder Schalentheorie und somit auch den entsprechenden Diskretisierungen anhaftet, kann bei der gewählten Vorgehensweise mit dreidimensionalen FE-Ansätzen hoher Ordnung durch die Erhöhung des Polynomgrades in Dickenrichtung leicht kontrolliert werden. Somit entfallen sämtliche modellbedingte Beschränkungen, die bei der Verwendung von klassischen Schalenelementen entstehen. Geometrische Nichtlinearitäten können dabei leicht berücksichtigt werden, da eine rein kontinu-

umsmechanische Formulierung vorliegt und somit die problematische Behandlung von Rotationsfreiheitsgraden entfällt. Mit dem dreidimensionalen Ansatz ist ferner eine einheitliche Diskretisierung von dünn- und dickwandigen Strukturen möglich, ohne dabei Übergangselemente verwenden zu müssen. Die Geometrie gekrümmter Bleche wird unter Verwendung der Blending-Funktionen-Methode beschrieben. Diese Technik erlaubt eine sehr genaue Beschreibung der Geometrie, unabhängig von der Anzahl der verwendeten Finiten Elemente. Die genaue Auswertung der Geometrie ist dabei auch für die weitere Verwendung der Ergebnisse der Rückfederung leicht möglich.

1.2 Inhaltsübersicht

Das Anliegen von **Kapitel 2** ist zunächst eine zusammenfassende Darstellung der Umformtechnik. Dabei wird im Speziellen ein Überblick über die Grundlagen des Tiefziehens, die Rechenverfahren und die numerische Modellbeschreibung gegeben. Weiter wird das Problem der elastischen Rückfederung erläutert. Abschließend werden aktuelle Möglichkeiten zur Kompensation der elastischen Rückfederung mit 'virtuellen Methoden' vorgestellt.

In **Kapitel 3** wird eine Einführung in die p -Version der Finite-Elemente-Methode für dreidimensionale Kontinuumsprobleme gegeben. Dabei wird ausgehend von den mathematischen und mechanischen Grundlagen das Prinzip der virtuellen Arbeit erläutert, mit dem die zugrunde liegende Differentialgleichung gelöst werden kann. Weiterhin werden die hierarchischen Ansatzfunktionen und anisotrope Ansatzräume vorgestellt, welche für die p -Version der Finite-Elemente-Methode verwendet werden. Schließlich wird noch auf eine effiziente Methode zur Berechnung der geometrisch nichtlinearen Rückfederung eingegangen.

Kapitel 4 erläutert die Grundlagen des geometrischen Modellierens. Dabei werden zunächst die Topologieobjekte und die Graphenstruktur in einem Geometriemodellierer dargestellt. Die Beschreibung von Freiformkurven und -flächen erfolgt im Abschnitt über 'Geometrische Attributierung'. Für den im Kopplungsprozess zentralen Punkt der Flächenrückführung werden Interpolations- und Approximationsverfahren sowie die für die Netzgenerierung wichtigen Booleschen Operatoren ausführlich beschrieben. Im Gegensatz zur h -Version werden bei der p -Version große Elemente verwendet, welche eine genaue Beschreibung der Geometrie erfordern. Für die p -Version wird dafür abschließend die Blending-Funktionen-Methode in Kombination mit der Quasi-regionalen Abbildung erläutert.

In **Kapitel 5** wird die Aufbereitung der Rechenergebnisse aus der Tiefziehberechnung der auf der Schalentheorie basierenden Finite-Elemente-Programme für die p -Version der FEM vorgestellt. Dabei wird zunächst die für die Neuvernetzung notwendige Flächenrückführung erläutert. Der Diskretisierung der Geometrie mithilfe eines Netzgenerators folgt die ausführliche Beschreibung der Blending-Funktionen-Methode und der Vorgehensweise zur Erfassung der exakten Geometrie der Hexaeder-Elemente mithilfe eines Geometriemodellierers. Weiter folgt eine Beschreibung der Datenstrukturen und Algorithmen welche für die Abbildung und Transformation der Spannungsverteilung der Tiefziehsimulationsergebnisse auf die Diskretisierung für die p -Version notwendig sind. Abschließend folgt ein Ansatz zur Korrektur des 2D-Spannungszustands für Berechnungen der beim Tiefziehen erforderlichen Pressenkraft.

Beispiele für die Simulation der elastischen Rückfederung mit der p -Version der Finite-Elemente-Methode werden in **Kapitel 6** gegeben. Dabei wird ein akademisches Beispiel zur Verifikation der Implementierung der geometrischen Nichtlinearität, das Benchmarkbauteil der NUMISHEET 1996 (S-Rail) sowie die Berechnungen von zwei aktuellen Industriebauteilen, welche die Praxistauglichkeit und Effizienz von Elementen höherer Ordnung belegen sollen, präsentiert.

Abschließend erfolgt in **Kapitel 7** eine Zusammenfassung der Arbeit und ein kurzer Ausblick auf anstehende Weiterentwicklungen zur numerischen Simulationen von Problemstellungen in der Umformtechnik mithilfe der p -Version der Finite-Elemente-Methode.

Kapitel 2

Grundlagen der Umformtechnik

Das Tiefziehen mit starren Werkzeugen stellt ein wichtiges Verfahren der Blechumformung zur wirtschaftlichen Herstellung von Blechteilen z. B. für den Automobil-, Anlagen- und Behälterbau dar. Wesentlich für die Qualität und Kosten der Ziehteile ist die Entwicklung präziser Umformwerkzeuge. Bezüglich der Wirtschaftlichkeit wird dabei eine niedrige Ausschussrate und ein geringer Verschleiß der Werkzeuge angestrebt. Die Planung und Konstruktion einer Ziehanlage stellt einen aufwändigen Entwicklungsprozess dar. Früher wurden Werkzeuge aufgrund von Erfahrungswerten entworfen und mithilfe von Versuchen optimiert. Die Anzahl der Iterationsschleifen und damit die Höhe der Kosten sowie die Qualität hing von der Erfahrung des Planers und des Werkzeugbauers ab.

Eine Problematik bei der Blechumformung ist die elastische Rückfederung von Bauteilen, welche bei der Entnahme des Ziehteils aus der Presse auftritt. Optimierungen bezüglich der elastischen Rückfederung basieren meist auf dem Prinzip, Bauteile gezielt zu überbiegen, so dass diese nach der Rückfederung die gewünschte Form annehmen. Allerdings liegen hierfür nur Erfahrungswerte für die früher verwendeten konventionellen Tiefziehstähle vor. Aufgrund der Gewichtsreduktion der Blechteile kommen heute häufig Werkstoffe wie Aluminium und hochfeste Stähle zum Einsatz, für welche nicht genügend solcher Erfahrungswerte existieren. Zudem entstehen bei diesen Materialien wegen des geringeren E-Moduls bei Aluminium bzw. der höheren Fließspannung bei hochfestem Stahl größere Verformungen infolge der Rückfederung [8, 33, 36].

Die Simulation der Umformung mit modernen Computersystemen hat daher das Ziel, die Werkzeuggeometrie bereits während der Planungsphase ohne zeitaufwändige Versuche zu optimieren. Durch die Simulation können bereits vor Herstellung der Werkzeuge Aussagen über die Riss- und Faltenbildung sowie die elastische Rückfederung getroffen werden und durch geeignete Prozessparameter Maßnahmen zu deren Verhinderung bzw. Kompensation ergriffen werden. Längerfristig wird dabei angestrebt, die gesamte Werkzeugentwicklung mit allen technisch relevanten Effekten auf den Rechner zu verlagern und somit eine geschlossene, virtuelle Prozesskette zu ermöglichen.

2.1 Grundbegriffe des Tiefziehens

Bei der Herstellung von Ziehteilen wird in der Regel eine so genannte einfach wirkende Presse verwendet. Dabei befindet sich der Stempel statisch im Pressenunterteil, Matrize und Blechhalter sind in Ziehrichtung beweglich angeordnet. Der Blechhalter ist über mehrere Hydraulikzylinder in Ziehrichtung gelagert. Dadurch kann der Blechhalter nur durch das Aufbringen einer definierten Kraft (Blechhalterkraft) in Bewegung gesetzt werden. Beim Blechhalterschließen fährt die Matrize auf den Blechhalter (siehe Abbildung 2.1).

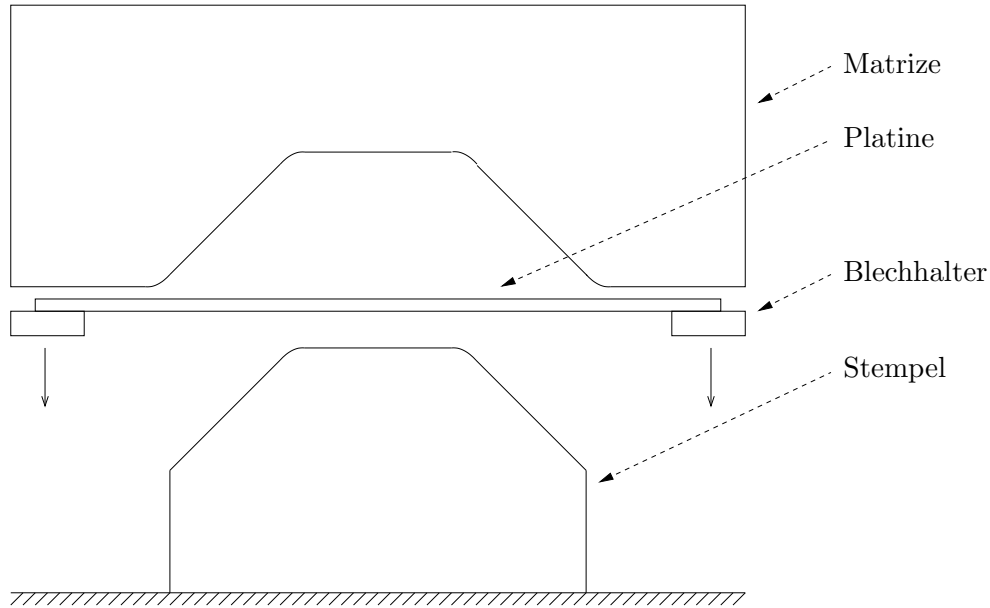


Abbildung 2.1: Vorgang Blechhalterschließen

Beim anschließenden Ziehvorgang zieht die Matrize das Blech über den Stempel (siehe Abbildung 2.2).

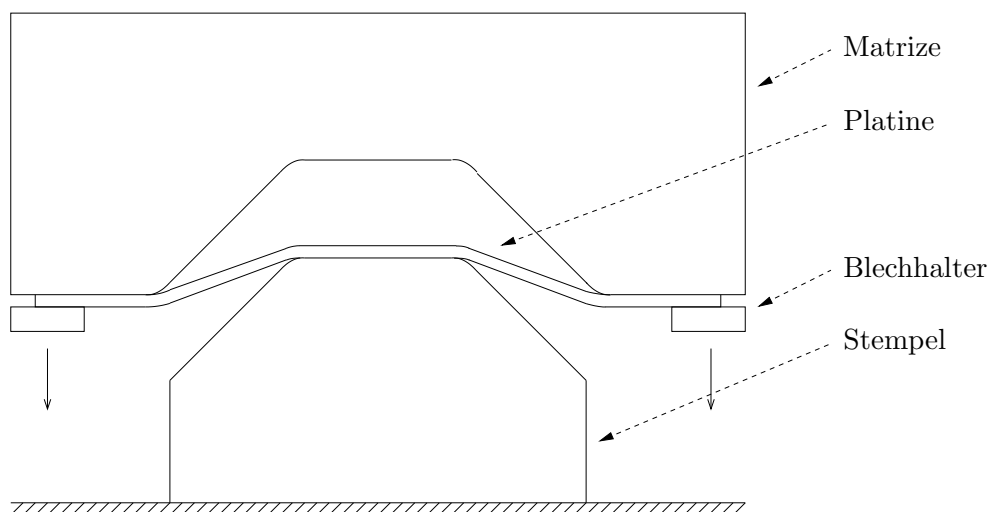


Abbildung 2.2: Ziehvorgang

Die in Abbildung 2.3 dargestellte Sicke übernimmt die Steuerung des Materialflusses indem sie Rückhaltekräfte aufbaut. Über die Sickenhöhe bzw. ihren Radius lässt sich die Rückhaltekraft verändern.

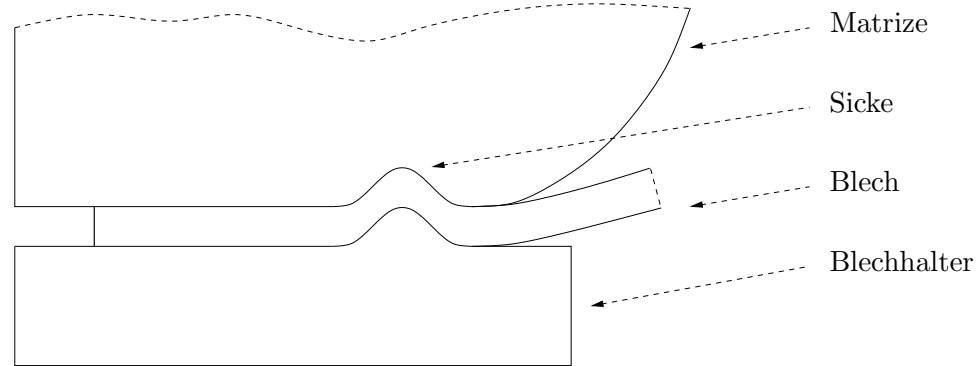


Abbildung 2.3: Sicke auf Blechhalter

Ein weiterer wichtiger Prozessparameter ist die Blechhalterkraft. Sie bezeichnet die Kraft, mit der der Blechhalter gegen die Matrize gedrückt wird. Über die Blechhalterkraft kann die Faltenbildung unter dem Blechhalter beeinflusst werden.

Um einen erhöhten Verschleiß zu vermeiden, muss im geschlossenen Zustand der Presse ein Spalt zwischen Matrize und Stempel für das Werkstück vorhanden sein.

2.1.1 Materialbeschreibung

Damit beim Umformen plastisches Fließen im Werkstück eintritt, muss die tatsächlich auftretende Spannung eine charakteristische Größe erreichen. In der Umformtechnik ist es daher üblich, die wirkende Kraft F auf den tatsächlichen Querschnitt bzw. dessen Fläche S zu beziehen. Die Fließspannung k_f ist definiert als

$$k_f = \frac{F}{S}. \quad (2.1)$$

In der Festigkeitslehre ist es üblich, die Verzerrung ε für den einachsigen Zugversuch als Längenänderung $L_1 - L_0$ im Verhältnis zur Ausgangslänge L_0 zu definieren:

$$\varepsilon = \int_{L_0}^{L_1} \frac{dL}{L_0} = \frac{L_1 - L_0}{L_0} \quad (2.2)$$

Da in der Umformtechnik in der Regel große plastische Formänderungen vorliegen, ist die in Gleichung (2.2) definierte Ingenieurs-Dehnung nicht zur Beschreibung der Verzerrung geeignet. Statt dessen wird häufig der *Umformgrad* φ , bei dem die Längenänderung auf die augenblickliche Länge L bezogen ist, verwendet:

$$\varphi = \int_{L_0}^{L_1} \frac{dL}{L} = \ln \frac{L_1}{L_0} \quad (2.3)$$

Die *Fließkurve*

$$k_f(\varphi) \tag{2.4}$$

stellt den Bereich der im allgemeinen nicht-linearen Beziehung zwischen Umformgrad φ und der Spannung dar, welche für einen einachsigen Spannungszustand notwendig ist damit Fließen eintritt bzw. aufrecht erhalten wird. Bei mehrachsigen Spannungszuständen hängt der Eintritt des Fließens nicht nur von einer Spannungskomponente, sondern vom gesamten Spannungstensor ab. Der Übergang von ein- auf mehrachsige Spannungszustände ist durch Fließbedingungen, z. B. von Tresca und von Mises möglich, bei denen eine Vergleichsspannung σ_v berechnet und über ihren Vergleichsumformgrad φ_v mit der Fließspannung k_f verglichen wird [46]. $\dot{\varphi}_v$ definiert dabei die Vergleichsumformgeschwindigkeit.

$$\begin{aligned} \text{von Mises:} \quad \dot{\varphi}_v &= \sqrt{\frac{2}{3} (\dot{\varphi}_1^2 + \dot{\varphi}_2^2 + \dot{\varphi}_3^2)} \\ \varphi_v &= \int_{t_0}^{t_1} \dot{\varphi}_v dt \end{aligned} \tag{2.5}$$

$$\begin{aligned} \text{Tresca:} \quad \dot{\varphi}_v &= \max(\dot{\varphi}_1, \dot{\varphi}_2, \dot{\varphi}_3) \\ \varphi_v &= \max(\varphi_1, \varphi_2, \varphi_3) \end{aligned} \tag{2.6}$$

Unter dem *Formänderungsvermögen* eines Werkstoffes wird der beim Bruch erreichbare Vergleichsumformgrad φ_B verstanden. Die Bestimmung des Formänderungsvermögens ist im allgemeinen schwierig, da der beim Bruch erreichte Umformgrad von verschiedenen Einflüssen abhängt. Selbst beim einachsigen Spannungszustand ist der ermittelte Umformgrad abhängig von der Art des Spannungszustands, der Umformgeschwindigkeit sowie der Temperatur.

Bei höheren Temperaturen finden im Werkstoff Erholungs- und Rekristallisationsvorgänge infolge thermischer Aktivierung statt [46]. Ob Rekristallierung oder Kristallerholung der maßgebliche Vorgang ist, muss von Fall zu Fall entschieden werden. Beide Vorgänge erfolgen mit einer endlichen, temperaturabhängigen Geschwindigkeit. Der Verlauf der Fließkurve wird beeinflusst durch die Verfestigung infolge der Zunahme der Versetzungsdichte bei der Umformung sowie durch die Abnahme der Versetzungsdichte infolge von Kristallerholung bzw. Rekristallierung. Daher sind die Fließkurven außer von der Temperatur T und dem Umformgrad φ stark von der Umformgeschwindigkeit $\dot{\varphi}$ abhängig. Kann sich ein völliger Temperatenausgleich einstellen, so lässt sich daraus die isotherme Fließkurve ermitteln. Im Falle einer unendlich hohen Umformgeschwindigkeit wird die adiabatische Fließkurve erhalten, da die Umformwärme nicht abgeleitet werden kann. Grundsätzlich gilt, dass das Formänderungsvermögen mit wachsender Temperatur zu- und mit wachsender Umformgeschwindigkeit abnimmt (siehe Abbildung 2.4). Allgemein lassen sich nach [56] Kalt- und Warmfließkurve formulieren zu

$$k_f(\varphi_v, \dot{\varphi}_v) = C \varphi_v^n \dot{\varphi}_v^m. \tag{2.7}$$

Der Verfestigungsexponent n ist ein Maß für die Kaltverfestigung und hängt vom Gefügezustand ab. Ansteigende Werte für die Festigkeitseigenschaften entsprechen einem kleineren n -Wert und vermindern die Umformbarkeit eines Werkstoffs. Durch einen größeren n -Wert wird die Neigung des Werkstoffs zu örtlichem Einschnüren beim Streckziehen herabgesetzt, was in einer höheren Gleichmaßdehnung resultiert. Wenn sich die Fließkurve nicht mit einem konstanten n -Wert beschreiben lässt, wird n häufig als Funktion

$$n(\varphi) \tag{2.8}$$

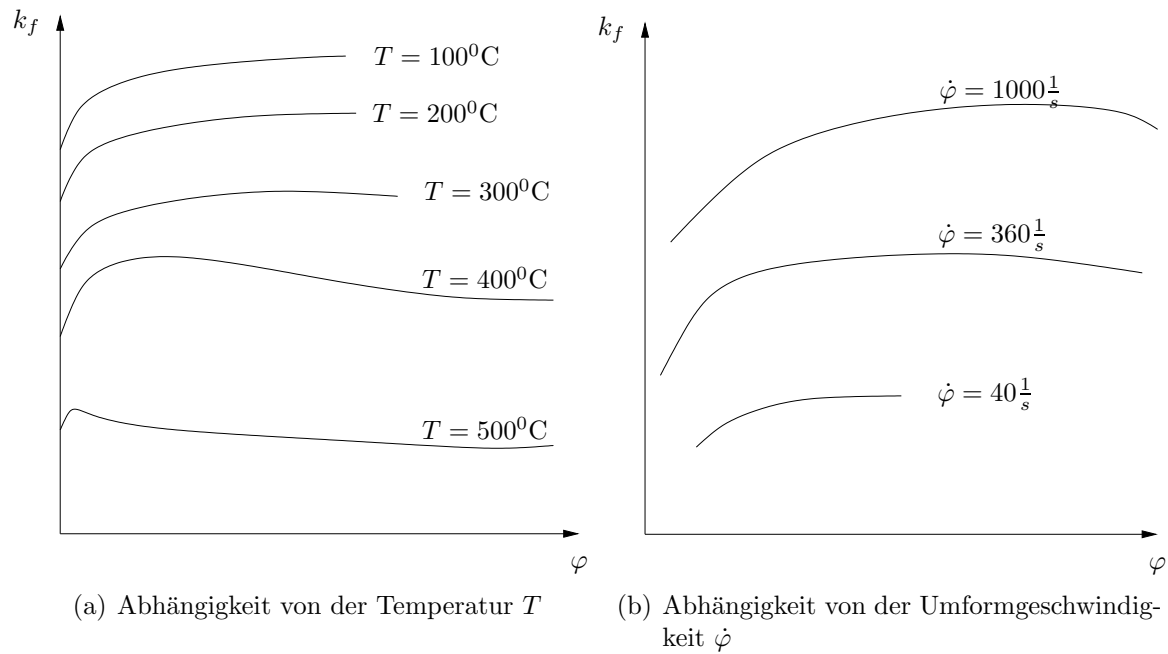


Abbildung 2.4: Fließkurven

definiert. Der Exponent m ist von Temperatur und Material abhängig. Werte für C , m und n sind in [46] tabelliert.

Die Beziehung für *Kaltfließkurven* ergibt sich aus Gleichung (2.7) zu

$$k_f(\varphi) = C\varphi_v^n \quad (2.9)$$

wenn $m = 0$ gesetzt wird, da bei Raumtemperatur der Einfluss der Umformgeschwindigkeit auf die Fließspannung vernachlässigt werden kann.

Bei *Warmfließkurven* ist in der Regel der Einfluss der Umformgeschwindigkeit viel höher als der des Umformgrades. Daher gilt

$$n \ll m \quad (2.10)$$

und Gleichung (2.7) geht näherungsweise über in die Potenzfunktion

$$k_f \approx k_{f1} \left(\frac{\dot{\varphi}_v}{\dot{\varphi}_{v1}} \right)^m, \quad (2.11)$$

wobei k_{f1} die Fließspannung bei der Umformgeschwindigkeit $\dot{\varphi}_{v1}$ ist.

Zugspannungen führen eher zu einem Bruch als Druckspannungen. Wird der Spannungszustand so beeinflusst, dass die auftretenden Spannungen im Druckgebiet liegen, so erhöht sich das Formänderungsvermögen. Die mittlere Hauptspannung

$$\sigma_m = \frac{\sigma_1 + \sigma_2 + \sigma_3}{3} \quad (2.12)$$

beschreibt, wie weit der Spannungszustand im Druckgebiet liegt. Sie ist somit ein Maß für das Formänderungsvermögen.

Der bei einem bestimmten Umformverfahren erreichbare Vergleichsumformgrad wird als *Grenzformänderung* φ_G bezeichnet. Dieser wird zusätzlich zu den genannten Einflüssen vom Fertigungsverfahren und den in der Kontaktzone vorliegenden Gegebenheiten durch

- Krafteinleitungsvermögen,
- unzulässige Deformation (Knicken, Stauchen), sowie
- Einschränkung durch Werkzeugeigenschaften

beeinflusst. Dabei kann die Grenzformänderung höchstens gleich dem Formänderungsvermögen des Werkstoffs sein:

$$\varphi_G \leq \varphi_B \quad (2.13)$$

Elastizität hängt mit zwischenatomaren Kräften zusammen, daher ist der E-Modul im allgemeinen richtungsabhängig. Liegt ein anisotroper Werkstoff vor, ist zusätzlich die Kenntnis des Fließorts erforderlich. Fließortkurven sind der geometrische Ort für alle Kombinationen der Spannungskomponenten, bei denen plastisches Fließen eintritt. Die plastische Anisotropie der Walzbleche wird durch so genannte r -Werte beschrieben, welche die Fließspannungsverhältnisse in Längs- (r_0) und Breitenrichtung (r_{90}) charakterisieren.

2.2 Grundlagen der elastischen Rückfederung

2.2.1 Definition

Nach dem Umformen stehen die Werkzeugkräfte bei geschlossener Presse mit den verbliebenen elastischen Spannungen im Bauteil im Gleichgewicht. Während des Auseinanderfahrens der Werkzeuge kommt es zur elastischen Rückfederung und damit zu einer Veränderung der Geometrie des Bauteils. Die nach dem Umformen im Werkstück verbleibenden elastischen Spannungen stehen mit den Kontaktkräften zwischen Werkzeug und Bauteil im Gleichgewicht. Die Rückfederung beschreibt demnach die Verschiebungen, die durch die Relaxation der Spannungen beim Auseinanderfahren der Werkzeuge hervorgerufen werden. Bei der Rückfederung handelt es sich um einen unvermeidbaren physikalischen Effekt [49], welcher durch Prozessparameter mehr oder weniger kompensiert werden kann. Biegespannungen entstehen bei der Umformung hauptsächlich in den Bauteilradien und bewirken bei der Rückfederung von Werkstücken große Geometrieänderungen, da sie im Gegensatz zu Normalspannungen eine Veränderung der Krümmung im Bauteil bewirken.

2.2.2 Kompensation der elastischen Rückfederung

Unter der Kompensation der elastischen Rückfederung wird die Modifizierung der Ausgangswerkzeuggeometrie verstanden, um die Geometrieabweichungen zwischen Zielgeometrie und dem rückgefederten Bauteil zu vermindern.

Die Ansätze zur Kompensation der elastischen Rückfederung lassen sich in zwei Verfahrensweisen gruppieren. Die erste Gruppe sind Methoden, welche die Blechhalterkräfte variieren, um direkt Einfluss auf das Verhalten der Rückfederung zu nehmen. Die zweite Verfahrensweise hat nicht das Ziel, das Verhalten der Rückfederung zu beeinflussen, sondern vielmehr durch geeignete Verfahren die Zielgeometrie nach Umformung und elastischer Rückfederung zu erhalten. Dies kann zum einen durch *Verschiebungsanpassungs-* [36, 79, 80, 82] oder durch *Spring-Forward-*Methoden [3, 42] erreicht werden.

Die Verschiebungsanpassung ist ein iterativer Prozess, dessen Ziel eine optimierte Werkzeuggeometrie zur Herstellung der Bauteilgeometrie ist. Nachfolgend ist das Verfahren für den Iterationsschritt i dargestellt. G_i^{mod} ist dabei die Oberfläche nach der Rückfederung (f_{SB}) von G_i :

$$G_i^{\text{mod}} = f_{\text{SB}}(G_i) \quad (2.14)$$

Im folgenden werden die Oberflächen G_1 und G_2 durch ihre Abbildungen f_1 und f_2 bezüglich ihrer gemeinsamen Parametrisierung (r, s) in den euklidischen Raum \mathbb{R}^3 beschrieben:

$$G_1 = f_1(r, s) = (x_1, y_1, z_1) \quad (2.15)$$

$$G_2 = f_2(r, s) = (x_2, y_2, z_2) \quad (2.16)$$

Für die Differenz

$$G_1 - G_2 = f_1(r, s) - f_2(r, s) =: g_{1,2}(r, s) \quad (2.17)$$

gilt:

$$g_{1,2}(r, s) \equiv 0 \quad \forall r, s \Rightarrow G_1 = G_2 \quad (2.18)$$

Die Norm

$$\|g\| = \iint_{r s} |g_{1,2}(r, s)| dr ds =: d(G_1, G_2) \quad (2.19)$$

beschreibt demnach das Volumen zwischen den Oberflächen G_1 und G_2 .

Die verbesserte, kompensierende Werkzeuggeometrie ergibt sich somit zu

$$G_{i+1} = G_i + \alpha (G_i^{\text{mod}} - G^{\text{ref}}) \quad \alpha \leq -1 \quad (2.20)$$

wobei G^{ref} die Zielgeometrie und α den Kompensationsfaktor darstellt. Der Kompensationsfaktor wird dabei gewöhnlich zu $\alpha < -1$ gesetzt. Ein in der Praxis häufig verwendeter Wert ist dabei $\alpha = -1.3$. Aus diesem Wert ist ersichtlich, dass bei der Kompensationsmethode der Rückfederungseffekt verstärkt wird. Dies ist auch nicht weiter verwunderlich, zumal das zur Kompensation erforderliche Überbiegen eine größere Verformung bewirkt. Die Iteration wird schließlich solange fortgesetzt bis

$$d(G^{\text{ref}}, G_i^{\text{mod}}) < \varepsilon. \quad (2.21)$$

In der h -Version der Finite-Elemente-Methode liegen die Verschiebungen gewöhnlich an den Elementknoten der Platine vor. Um die kompensierende Werkzeuggeometrie zu ermitteln,

müssen die Verschiebungswerte entsprechend auf die Geometrie der Werkzeugoberfläche interpoliert werden. Eine verbesserte Vorgehensweise ist dabei die Verwendung einer geglätteten, kontinuierlichen Verschiebungsfunktion [82]. Durch die glattere Beschreibung von Geometrie und Verschiebung ist dies bei der p -Version der Finite-Elemente-Methode leichter möglich.

Bei der *Spring-Forward*-Methode besteht der grundlegende Ansatz in der Invertierung des Spannungszustandes. Die optimierte Werkzeuggeometrie wird aus der Verformung der mit einem invertierten Spannungszustand berechneten elastischen Rückfederung bestimmt. Die Spring-Forward-Methode ist in Abbildung 2.5 schematisch dargestellt. Zunächst wird das Bauteil im ersten Iterationsschritt mit einer der Zielgeometrie entsprechenden Werkzeuggeometrie umgeformt. In Abbildung 2.5(a) ist die Form des Bauteils nach der elastischen Rückfederung als gestrichelte Linie u_{SB} dargestellt. Bei Anwendung der Spring-Forward-Methode werden die Biegespannungen von σ invertiert und es ergibt sich die elastische Rückfederung des in Abbildung 2.5(b) dargestellten Verschiebungsverlaufs u_{SF} . Dabei stellt u_{SF} in Abbildung 2.5(c) den Ausgangspunkt der Werkzeuggeometrie für den nächsten Iterationsschritt dar, so dass schließlich im kompensierten Ergebnis die Form des rückgefederten Bauteils u_{SB} der Zielgeometrie entspricht. Üblicherweise sind dazu einige Iterationsschritte notwendig. Im folgenden wird die

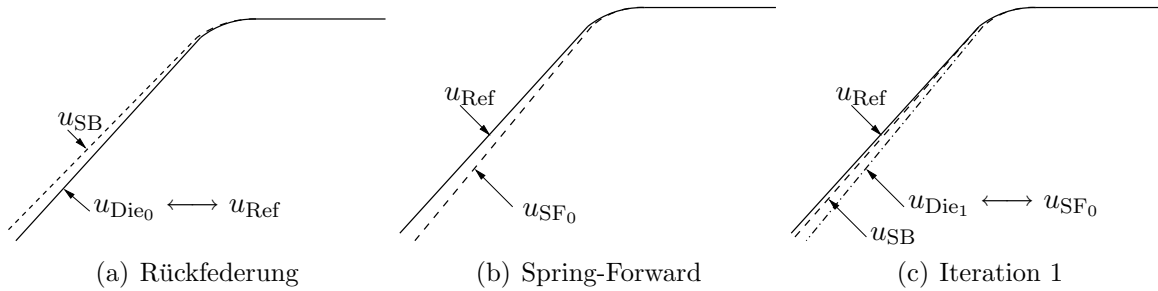


Abbildung 2.5: Spring-Forward-Methode

in [3] modifizierte Spring-Forward-Methode beschrieben, welche zu einer höheren Effizienz des Verfahrens führt. Dazu wird der skalare Spring-Forward-Faktor α definiert, mit welchem die elastischen Spannungen nach der Umformsimulation skaliert werden. Für $\alpha = 0$ ergibt sich die Werkzeugform als Initialgeometrie u_{Soll} , $\alpha = -1$ liefert die Werkzeugform, welche der rückgefederten Geometrie mit invertierten Spannungskomponenten entspricht. Abbildung 2.6 stellt schematisch für den eindimensionalen Fall das Verfahren dar. In Abbildung 2.6(a) repräsentiert die dargestellte Kurve den nichtlinearen Zusammenhang zwischen der Rückfederung u und den Resteigenspannungen σ . Der Punkt $u_{Soll} = 0$ stellt den Nullpunkt der Verschiebung, also die Geometrie nach dem Umformen im geschlossenen Zustand der Presse dar. Weiterhin wird u für $\alpha = 0$ und $\alpha = -1$ berechnet. Die diskreten Beziehungen zwischen α und der Verschiebung u (siehe Abbildung 2.6(b)) können über ein Lagrange-Polynom als Kurve interpoliert werden:

$$u = p(\alpha) \quad (2.22)$$

Der für die Kompensation erforderliche Wert α ermittelt sich durch

$$p(\alpha) \stackrel{!}{=} 0. \quad (2.23)$$

Die Iteration

$$G_i = f_{\text{mSF}}(G_{i-1}, \alpha_i) \quad (2.24)$$

wird schließlich so lange durchgeführt, bis

$$d(G^{\text{ref}}, f_{\text{SB}}(G_i)) < \varepsilon. \quad (2.25)$$

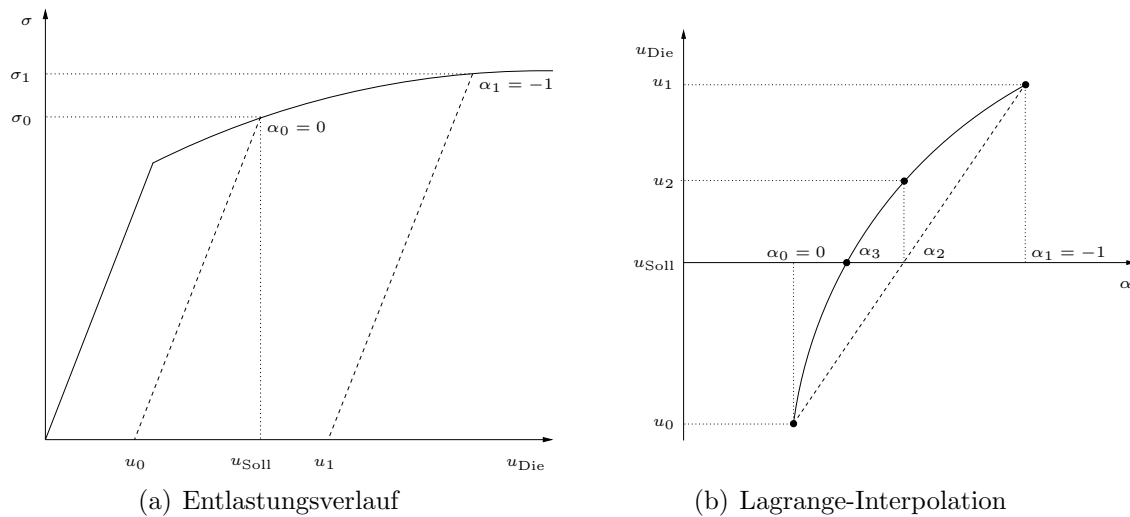


Abbildung 2.6: Modifizierte Spring-Forward-Methode nach [3]

Bei der ursprünglichen Spring-Forward-Methode muss für jeden Iterationsschritt ein neuer Spannungszustand σ berechnet werden, um schließlich mit dem konstanten Spring-Forward-Faktor $\alpha = -1$ die Werkzeuggeometrie für den nächsten Iterationsschritt zu bestimmen. Der Vorteil der modifizierten Methode liegt nun darin, dass der Spannungszustand während der Iteration konstant bleibt und lediglich die skalare Größe α iterativ bestimmt wird, um daraus die kompensierende Werkzeuggeometrie näherungsweise zu ermitteln. Durch diese Vorgehensweise reduziert sich die Anzahl der äußeren Iterationen zur Bestimmung des tatsächlichen Spannungszustands gegenüber der ursprünglichen Spring-Forward-Methode.

2.2.3 Modellierung

Bei der Berechnung der elastischen Rückfederung wird von der Annahme ausgegangen, dass die Formänderung während des Entlastungsvorgangs vollkommen elastisch geschieht. Daher kann näherungsweise ein lineares Materialgesetz verwendet werden. Kontaktdefinitionen sind aufgrund der statischen Gleichgewichtsberechnung nicht erforderlich. Die Berechnung der elastischen Rückfederung erfolgt in der Regel mit impliziten Verfahren. Der durch die Umformsimulation ermittelte elastische Spannungszustand im geschlossenen Zustand der Presse stellt die Krafrandbedingung für die Simulationen der elastischen Rückfederung dar. Die Qualität der Rückfederungsberechnung hängt damit von der Genauigkeit ab, mit der die Spannungen im Bauteil durch die Umformung bestimmt wurden. Die Umformsimulation muss mit

einer ausreichend großen Anzahl von Integrationspunkten über die Schalendicke erfolgen, damit die Biegespannungen hinreichend genau abgebildet werden. Für die Untersuchung der elastischen Rückfederung muss die umgeformte Platine statisch bestimmt gelagert werden. Statisch unterbestimmte Lagerungen führen zu kinematischen Systemen, bei denen sich kein Gleichgewicht einstellen kann. Eine statisch überbestimmte Lagerung würde das Ergebnis der elastischen Rückfederung beeinflussen, da sich die Platine in diesem Fall nicht mehr ungehindert verformen kann. Statisch bestimmte Lagerungen führen in Abhängigkeit des Ortes der Verschiebungsrandbedingung zu unterschiedlichen Starrkörperrotationen am Bauteil. Um die Ergebnisse mehrerer mit unterschiedlich definierten Dirichlet-Randbedingungen berechneten Simulationen vergleichen zu können, muss das Bauteil deshalb in eine definierte Referenzlage rotiert werden.

Kapitel 3

Die p -Version der Finite-Elemente-Methode

3.1 Einführung

Die Finite-Elemente-Methode ist eine der am häufigsten eingesetzten Verfahren zur numerischen Lösung partieller Differentialgleichungen für Probleme im Maschinenbau, im Bauingenieurwesen und in vielen anderen Ingenieurwissenschaften. Ihre Anwendung findet sie z. B. in der Berechnung von Festkörpern, Strukturen, Fluiden sowie für Probleme der Wärmeleitung [20]. Dabei wird das reale Problem unter gewissen Annahmen auf ein physikalisches System idealisiert. Dieses physikalische Modell wird durch ein mathematisches Modell beschrieben, welches zu einem System partieller Differentialgleichungen führt. Die Finite-Elemente-Methode ist ein Verfahren zur numerischen Lösung dieser Differentialgleichungen. Die Genauigkeit dieser Lösung ist dabei beschränkt durch vereinfachende Modellannahmen („Modellfehler“) und durch den *Diskretisierungsfehler* der Finiten-Element-Methode. Beide Fehler, sowohl Modell- als auch Diskretisierungsfehler müssen kontrolliert werden, um zuverlässige Berechnungsergebnisse zu erhalten.

Der Diskretisierungsfehler kann bei der h -Version durch lokale oder globale Netzverfeinerung kontrolliert werden. Die Grundidee der p -Version der Finite-Elemente-Methode besteht darin, für ein fixes Netz den Polynomgrad der Ansatzfunktionen schrittweise zu erhöhen, um eine konvergente Lösung zu erhalten. Es wurde gezeigt, dass die p -Version für linear elliptische Probleme mit glatten Lösungen eine exponentielle Konvergenz für den Fehler in der Energienorm besitzt. Sogar für Probleme mit Singularitäten kann durch eine entsprechende lokale *a priori* Netzverfeinerungen eine prä-asymptotisch exponentielle Konvergenz erreicht werden [76], wohingegen die h -Version nur eine algebraische Konvergenz aufweist.

Der Modellfehler entsteht durch die Einführung verschiedener physikalisch motivierter vereinfachender Annahmen wie der Vernachlässigung zeit-abhängiger Effekte oder der Annahme vereinfachender konstitutiver Gesetze. Eine weitere Quelle für Modellfehler ist die Dimensionsreduzierung bei Verwendung von Platten- bzw. Schalenmodellen für die Berechnung dünnwandiger Strukturen. Im Mittelpunkt jeder Dimensionsreduzierung steht die Annahme einer bestimmten polynomialen Verschiebung in Dickenrichtung der Platte bzw. Schale. Die

Reissner-Mindlin-Theorie für Platten bzw. die Naghdi-Theorie für Schalen nimmt z. B. eine lineare Verdrehung des Querschnitts bezüglich der Mittelebene an. In Bereichen, welche durch diese Theorien nicht abgebildet werden können, treten signifikante Fehler in der Lösung auf.

Im folgenden Kapitel wird zunächst das zu lösende allgemeine Elastizitätsproblem aufgestellt. In Kapitel 3.3 wird das Prinzip der virtuellen Verschiebungen, welches die Grundlage der Finite-Elemente-Methode bildet, erörtert. Schließlich wird in Kapitel 3.4 die Assemblierung und die Lösung des Gleichungssystems vorgestellt. Der darauf folgende Abschnitt beschreibt die grundsätzlichen Unterschiede der p -Version zur h -Version sowie die unterschiedlichen Ansatzräume. Abschließend folgt noch ein Abschnitt über geometrisch nichtlineare Probleme mit einer effizienten Lösungsstrategie für die elastische Rückfederung.

3.2 Dreidimensionale lineare Elastizitätsprobleme

3.2.1 Gleichgewicht

Die Gleichgewichtsbedingungen für ein dreidimensionales Elementarteilchen (siehe Abbildung 3.1) lassen sich in Form eines Differentialgleichungssystems beschreiben als

$$\begin{aligned} \frac{\partial \sigma_x^{(u)}}{\partial x} + \frac{\partial \tau_{xy}^{(u)}}{\partial y} + \frac{\partial \tau_{xz}^{(u)}}{\partial z} + f_x &= 0 \\ \frac{\partial \tau_{xy}^{(u)}}{\partial x} + \frac{\partial \sigma_y^{(u)}}{\partial y} + \frac{\partial \tau_{yz}^{(u)}}{\partial z} + f_y &= 0 \\ \frac{\partial \tau_{xz}^{(u)}}{\partial x} + \frac{\partial \tau_{yz}^{(u)}}{\partial y} + \frac{\partial \sigma_z^{(u)}}{\partial z} + f_z &= 0 \end{aligned} \quad (3.1)$$

wobei f_x, f_y, f_z die Volumenkräfte darstellen.

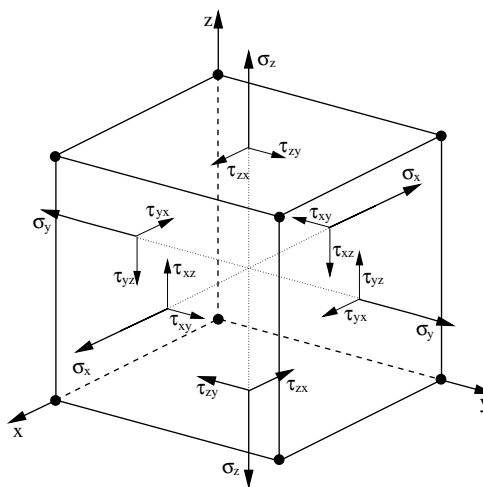


Abbildung 3.1: Komponenten des Spannungszustands am Elementarteilchen

3.2.2 Kinematik

Die Verschiebungen \mathbf{u} des Körpers aus der unverformten Konfiguration werden durch den Verschiebungsvektor

$$\mathbf{u} = \left[u_x(x, y, z) \quad u_y(x, y, z) \quad u_z(x, y, z) \right]^T \quad (3.2)$$

erfasst. Die Beziehungen zwischen Verzerrungen und Verschiebungen werden durch einen linearen Zusammenhang zu den Ableitungen der Verschiebungen

$$\begin{aligned} \varepsilon_x^{(u)} &= \frac{\partial u_x}{\partial x}, & \varepsilon_y^{(u)} &= \frac{\partial u_y}{\partial y}, & \varepsilon_z^{(u)} &= \frac{\partial u_z}{\partial z}, \\ \gamma_{xy}^{(u)} &= \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}, & \gamma_{yz}^{(u)} &= \frac{\partial u_y}{\partial z} + \frac{\partial u_z}{\partial y}, & \gamma_{zx}^{(u)} &= \frac{\partial u_z}{\partial x} + \frac{\partial u_x}{\partial z} \end{aligned} \quad (3.3)$$

dargestellt und können zum Verzerrungsvektor $\boldsymbol{\varepsilon}$ zusammengefasst werden.

$$\boldsymbol{\varepsilon}^{(u)} = \left[\varepsilon_x^{(u)} \quad \varepsilon_y^{(u)} \quad \varepsilon_z^{(u)} \quad \gamma_{xy}^{(u)} \quad \gamma_{yz}^{(u)} \quad \gamma_{zx}^{(u)} \right]^T = \mathbf{D}\mathbf{u} \quad (3.4)$$

\mathbf{D} bezeichnet dabei die 'Differentialoperatormatrix':

$$\mathbf{D} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} \quad (3.5)$$

3.2.3 Stoffgesetz

Für dreidimensionale isotrope, linear elastische Materialien berechnen sich die Spannungen

$$\boldsymbol{\sigma}^{(u)} = \left[\sigma_x^{(u)} \quad \sigma_y^{(u)} \quad \sigma_z^{(u)} \quad \tau_{xy}^{(u)} \quad \tau_{yz}^{(u)} \quad \tau_{zx}^{(u)} \right]^T \quad (3.6)$$

mit der Beziehung

$$\boldsymbol{\sigma}^{(u)} = \mathbf{C} (\boldsymbol{\varepsilon}^{(u)} - \boldsymbol{\varepsilon}^{(0)}) + \boldsymbol{\sigma}^{(0)}. \quad (3.7)$$

Dabei werden in $\boldsymbol{\varepsilon}^{(0)}$ vorgegebene Anfangsverzerrungen, welche oft thermische Verzerrungen darstellen, und in $\boldsymbol{\sigma}^{(0)}$ aufgeprägte Anfangsspannungen erfasst. \mathbf{C} definiert die konstitutive Beziehung, welche sich für ein isotropes, linear elastisches Material mithilfe des Hookeschen Gesetzes zu

$$\mathbf{C} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} (1-\nu) & \nu & \nu & 0 & 0 & 0 \\ \nu & (1-\nu) & \nu & 0 & 0 & 0 \\ \nu & \nu & (1-\nu) & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \quad (3.8)$$

ergibt, wobei E den Elastizitätsmodul und ν die Poissonzahl bezeichnet.

3.2.4 Randbedingungen

Neumann-Randbedingungen Γ_T (Kraftrandbedingungen) und Dirichlet-Randbedingungen Γ_u (Verschiebungsrandbedingungen) werden entlang des Randes $\partial\Omega$ des Gebietes Ω definiert. Die Randbedingungen werden mithilfe des Normalenvektors \mathbf{n} und den Tangentenvektoren \mathbf{t}_1 und \mathbf{t}_2 in zum Rand normalen und tangentialen Komponenten angegeben, für welche folgende Beziehung gilt:

$$\mathbf{t}_1 \times \mathbf{t}_2 = \mathbf{n} \quad (3.9)$$

wobei

$$\mathbf{n} = [n_x \ n_y \ n_z]^T, \quad \mathbf{t}_1 = [t_{1x} \ t_{1y} \ t_{1z}]^T, \quad \mathbf{t}_2 = [t_{2x} \ t_{2y} \ t_{2z}]^T. \quad (3.10)$$

Aufgrund des Kräftegleichgewichts am Rand ergibt sich:

$$\begin{bmatrix} T_n^{(u)} \\ T_{t_1}^{(u)} \\ T_{t_2}^{(u)} \end{bmatrix} = \begin{bmatrix} n_x & n_y & n_z \\ t_{1x} & t_{1y} & t_{1z} \\ t_{2x} & t_{2y} & t_{2z} \end{bmatrix} \begin{bmatrix} \sigma_x^{(u)} & \tau_{xy}^{(u)} & \tau_{xz}^{(u)} \\ \tau_{xy}^{(u)} & \sigma_y^{(u)} & \tau_{yz}^{(u)} \\ \tau_{xz}^{(u)} & \tau_{yz}^{(u)} & \sigma_z^{(u)} \end{bmatrix} \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \quad (3.11)$$

$$\begin{bmatrix} u_n^{(u)} \\ u_{t_1}^{(u)} \\ u_{t_2}^{(u)} \end{bmatrix} = \begin{bmatrix} n_x & n_y & n_z \\ t_{1x} & t_{1y} & t_{1z} \\ t_{2x} & t_{2y} & t_{2z} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \quad (3.12)$$

An jedem Randpunkt muss in tangentialer Richtung entweder eine Verschiebungs- oder Kraft-randbedingung gegeben sein. Ebenso muss in Normalenrichtung eine Verschiebungs- oder Kraftrandbedingung gestellt sein.

$$\begin{aligned} \Gamma &= \Gamma_T^n \cup \Gamma_u^n, & \Gamma_T^n \cap \Gamma_u^n &= \emptyset \\ \Gamma &= \Gamma_T^{t_1} \cup \Gamma_u^{t_1}, & \Gamma_T^{t_1} \cap \Gamma_u^{t_1} &= \emptyset \\ \Gamma &= \Gamma_T^{t_2} \cup \Gamma_u^{t_2}, & \Gamma_T^{t_2} \cap \Gamma_u^{t_2} &= \emptyset \end{aligned} \quad (3.13)$$

3.3 Das Prinzip der virtuellen Arbeit

Die schwache Formulierung kann über das Prinzip der virtuellen Arbeit, welches als klassische Variationsaufgabe verstanden werden kann, hergeleitet werden. Dabei wird eine Testfunktion

$$\mathbf{v} = \begin{bmatrix} v_x(x, y, z) \\ v_y(x, y, z) \\ v_z(x, y, z) \end{bmatrix} \quad (3.14)$$

definiert, auf deren Anforderungen später näher eingegangen wird. Aus den Gleichgewichtsbedingungen in Gleichungen (3.1) – (3.2) ergibt sich durch das Skalarprodukt mit der Testfunktion \mathbf{v} und Integration über das Gebiet Ω das Prinzip der virtuellen Arbeit.

$$\begin{aligned} \iiint_{\Omega} \left[\left(\frac{\partial \sigma_x^{(u)}}{\partial x} + \frac{\partial \tau_{xy}^{(u)}}{\partial y} + \frac{\partial \tau_{xz}^{(u)}}{\partial z} \right) v_x + \left(\frac{\partial \tau_{xy}^{(u)}}{\partial x} + \frac{\partial \sigma_y^{(u)}}{\partial y} + \frac{\partial \tau_{yz}^{(u)}}{\partial z} \right) v_y + \right. \\ \left. + \left(\frac{\partial \tau_{xz}^{(u)}}{\partial x} + \frac{\partial \tau_{yz}^{(u)}}{\partial y} + \frac{\partial \sigma_z^{(u)}}{\partial z} \right) v_z \right] dx dy dz + \iiint_{\Omega} (f_x v_x + f_y v_y + f_z v_z) dx dy dz = 0 \end{aligned} \quad (3.15)$$

Aus der partiellen Integration in drei Dimensionen mithilfe des Greenschen Satzes folgt der Ausdruck:

$$\begin{aligned} \iiint_{\Omega} (\sigma_x^{(u)} \varepsilon_x^{(v)} + \tau_{xy}^{(u)} \gamma_{xy}^{(v)} + \tau_{xz}^{(u)} \gamma_{xz}^{(v)} + \sigma_y^{(u)} \varepsilon_y^{(v)} + \tau_{yz}^{(u)} \gamma_{yz}^{(v)} + \sigma_z^{(u)} \varepsilon_z^{(v)}) dx dy dz + \\ + \iint_{\Gamma} (T_n^{(u)} v_n + T_{t1}^{(u)} v_{t1} + T_{t2}^{(u)} v_{t2}) d\Gamma + \iiint_{\Omega} (f_x v_x + f_y v_y + f_z v_z) dx dy dz = 0 \end{aligned} \quad (3.16)$$

Wenn sich der Körper im Gleichgewicht befindet, muss für jede beliebige, kompatible, kleine, dem Körper im Gleichgewichtszustand erteilte virtuelle Verschiebung die innere virtuelle Arbeit gleich der äußeren virtuellen Arbeit entsprechen. Mit Gleichung (3.4) und (3.7) ergibt sich:

$$\boldsymbol{\varepsilon}^{(v)T} \boldsymbol{\sigma}^{(u)} = (\mathbf{D}\mathbf{v})^T \mathbf{C} \mathbf{D}\mathbf{u} - (\mathbf{D}\mathbf{v})^T \mathbf{C} \boldsymbol{\varepsilon}^{(0)} - (\mathbf{D}\mathbf{v})^T \boldsymbol{\sigma}^{(0)} \quad (3.17)$$

Aus Gleichung (3.16) folgt

$$\begin{aligned} \iiint_{\Omega} (\mathbf{D}\mathbf{v})^T \mathbf{C} \mathbf{D}\mathbf{u} dx dy dz &= \iiint_{\Omega} (f_x v_x + f_y v_y + f_z v_z) dx dy dz + \\ &+ \iint_{\Gamma} (\mathbf{D}\mathbf{v})^T (\mathbf{C} \boldsymbol{\varepsilon}^{(0)} + \boldsymbol{\sigma}^{(0)}) dx dy dz + \\ &+ \iint_{\Gamma} (T_n^{(u)} v_n + T_{t1}^{(u)} v_{t1} + T_{t2}^{(u)} v_{t2}) d\Gamma \end{aligned} \quad (3.18)$$

wobei

- \mathbf{v} virtuelle Verschiebungen,
- $\boldsymbol{\varepsilon}^{(v)}$ virtuelle Verzerrungen und
- $\boldsymbol{\sigma}^{(v)}$ virtuelle Spannungen

bezeichnen. Die linke Seite von Gleichung (3.18) stellt das Energieskalarprodukt dar:

$$\mathcal{B}(\mathbf{u}, \mathbf{v}) := \iiint_{\Omega} (\mathbf{D}\mathbf{v})^T \mathbf{C} \mathbf{D}\mathbf{u} dx dy dz \quad (3.19)$$

$\mathcal{B}(\mathbf{u}, \mathbf{v})$ ist eine symmetrische Bilinearform ($\mathcal{B}(\mathbf{u}, \mathbf{v}) = \mathcal{B}(\mathbf{v}, \mathbf{u})$). Die rechte Seite entspricht dem Lastfunktional und stellt eine Linearform \mathcal{F} dar.

$$\begin{aligned} \mathcal{F}(\mathbf{v}) &:= \iiint_{\Omega} (Xv_x + Yv_y + Zv_z) \, dx \, dy \, dz + \\ &\quad + \iiint_{\Omega} (\mathbf{D}\mathbf{v})^T (\mathbf{C}\boldsymbol{\varepsilon}^{(0)} + \boldsymbol{\sigma}^{(0)}) \, dx \, dy \, dz + \\ &\quad + \iint_{d\Gamma} (T_n v_n + T_{t1} v_{t1} + T_{t2} v_{t2}) \, d\Gamma \end{aligned} \quad (3.20)$$

Die Verzerrungsenergie ist definiert als

$$\mathcal{U}(\mathbf{u}) := \frac{1}{2} \mathcal{B}(\mathbf{u}, \mathbf{u}) \quad (3.21)$$

und kann als Energienorm \mathbf{u} im Gebiet Ω ausgedrückt werden als

$$\|\mathbf{u}\|_{E(\Omega)} := \sqrt{\mathcal{U}(\mathbf{u})}. \quad (3.22)$$

Das Prinzip der virtuellen Arbeit kann wie folgt beschrieben werden:

Finde die Funktion \mathbf{u}_{ex} mit endlicher Verzerrungsenergie, welche die Randbedingungen und

$$\mathcal{B}(\mathbf{u}_{\text{ex}}, \mathbf{v}) = \mathcal{F}(\mathbf{v}) \quad (3.23)$$

für alle Funktionen \mathbf{v} erfüllt. \mathbf{v} ist eine beliebige Funktion mit endlicher Verzerrungsenergie. \mathbf{u}_{ex} ist die schwache Lösung des ebenen Elastizitätsproblems und minimiert die potentielle Energie

$$\Pi(\mathbf{u}_{\text{ex}}) = \min_{\mathbf{u} \in \Omega} \Pi(\mathbf{u}) \quad (3.24)$$

in Bezug auf alle zulässigen Verschiebungsfunktionen

$$\Pi(\mathbf{u}) = \mathcal{U}(\mathbf{u}) - \mathcal{F}(\mathbf{u}). \quad (3.25)$$

3.4 Die Finite-Elemente-Approximation

In den meisten Fällen ist es nicht möglich, eine exakte Lösung für Gleichung (3.23) zu ermitteln. Generell kann für \mathbf{u}_{ex} nur eine Näherungslösung gefunden werden. Dafür wird das Gebiet Ω in eine endliche Anzahl von Teilgebieten Ω_e geteilt und ein Satz von Basisfunktionen auf Ω so definiert, dass jede der Basisfunktionen ungleich Null über dem jeweiligen Element oder direkt benachbarten Elementen ist. Diese Basisfunktionen werden aus Polynomen konstruiert und sind auf Standardelementen definiert. Diese Elemente werden dann über Abbildungsfunktionen auf ihre wahre Geometrie abgebildet. Die Verschiebungsfunktionen sind von der Form:

$$\mathbf{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n a_i N_i(x, y, z) \\ \sum_{i=1}^n a_{n+i} N_i(x, y, z) \\ \sum_{i=1}^n a_{2n+i} N_i(x, y, z) \end{bmatrix} \quad (3.26)$$

In Matrix-Schreibweise:

$$\mathbf{u} = \mathbf{N}\mathbf{a} \quad (3.27)$$

wobei

$$\mathbf{N} := \begin{bmatrix} N_1 & N_2 & \cdots & N_n & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & N_1 & N_2 & \cdots & N_n & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & N_1 & N_2 & \cdots & N_n \end{bmatrix} \quad (3.28)$$

die Basis-Funktionen und

$$\mathbf{a} := [a_1 \ a_2 \ \cdots \ a_n \ a_{n+1} \ \cdots \ a_{3n}]^T \quad (3.29)$$

die Freiheitsgrade sind. Mit den Definitionen

$$\begin{aligned} \mathbf{N}_i &:= \begin{bmatrix} N_i(x, y, z) \\ 0 \\ 0 \end{bmatrix}; & i = 1, 2, \dots, n \\ \mathbf{N}_i &:= \begin{bmatrix} 0 \\ N_i(x, y, z) \\ 0 \end{bmatrix}; & i = n+1, n+2, \dots, 2n \\ \mathbf{N}_i &:= \begin{bmatrix} 0 \\ 0 \\ N_i(x, y, z) \end{bmatrix}; & i = 2n+1, 2n+2, \dots, 3n \end{aligned} \quad (3.30)$$

kann das Prinzip der virtuellen Arbeit (3.23) zu

$$\mathcal{B} \left(\sum_{i=1}^{3n} a_i \mathbf{N}_i, \mathbf{N}_j \right) = \mathcal{F}(\mathbf{N}_j) \quad (3.31)$$

für alle virtuellen Funktionen \mathbf{N}_j formuliert werden. $\mathbf{u} = \mathbf{N}\mathbf{a}$ muss dabei die geometrischen Randbedingungen erfüllen.

Die linke Seite von Gleichung (3.31) kann umgeschrieben werden zu

$$\begin{aligned} \mathcal{B} \left(\sum_{i=1}^{3n} a_i \mathbf{N}_i, \mathbf{N}_j \right) &= \sum_{i=1}^{3n} a_i \mathcal{B}(\mathbf{N}_i, \mathbf{N}_j) \\ &= \sum_{i=1}^{3n} a_i \iiint_{\Omega} (\mathbf{D}\mathbf{N}_i)^T \mathbf{C}\mathbf{D}\mathbf{N}_j \, dx \, dy \, dz \\ &= \sum_{i=1}^{3n} a_i \sum_{e=1}^r \iiint_{\Omega_e} (\mathbf{D}\mathbf{N}_i^e)^T \mathbf{C}\mathbf{D}\mathbf{N}_j^e \, dx \, dy \, dz \end{aligned} \quad (3.32)$$

wobei i, j die Freiheitsgrade der Finite-Element Diskretisierung und e die Anzahl der Elemente sind. Die globale Steifigkeitsmatrix ist dabei definiert als

$$\mathbf{K} := (k_{ij})_{i,j=1,\dots,3n} := \left(\iiint_{\Omega} (\mathbf{D}\mathbf{N}_i)^T \mathbf{C}\mathbf{D}\mathbf{N}_j \, dx \, dy \, dz \right)_{i,j=1,\dots,3n} \quad (3.33)$$

und die Element-Steifigkeitsmatrix als

$$\mathbf{K}^e := (k_{ij}^e)_{i,j \in I_e} := \left(\iiint_{\Omega_e} (\mathbf{DN}_i^e)^T \mathbf{CDN}_j^e dx dy dz \right)_{i,j \in I_e}. \quad (3.34)$$

I_e ist die Menge aller Funktionen, welche ungleich Null innerhalb des Elements e sind. \mathbf{N}^e stellt dabei die Matrix der Elementansatzfunktionen im Element e dar. Mit Gleichung (3.20) kann \mathbf{N}_j^e in das Funktional \mathcal{F} eingesetzt werden und der Lastvektor ergibt sich zu

$$\begin{aligned} \mathbf{f}^e := (f_j^e)_{j \in I_e} &= \left(\iiint_{\Omega_e} \mathbf{N}_j^{eT} \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} dx dy dz + \iiint_{\Omega_e} (\mathbf{DN}_j^e)^T (\mathbf{C}\boldsymbol{\varepsilon}^{(0)} + \boldsymbol{\sigma}^{(0)}) dx dy dz \right. \\ &+ \left. \iint_{\Gamma_e} \mathbf{N}_j^{eT} \begin{pmatrix} T_n \\ T_{t1} \\ T_{t2} \end{pmatrix} d\Gamma \right)_{j \in I_e} \\ &= \mathbf{f}_v^e + \mathbf{f}_\varepsilon^e + \mathbf{f}_\Gamma^e. \end{aligned} \quad (3.35)$$

3.4.1 Abbildungsfunktionen

Die in Kapitel 3.4 erwähnten Ansatzfunktionen sind auf dem Standardelement definiert. Um die Ansatzfunktionen vom Standardelement auf ein beliebiges Element zu transformieren wird eine Abbildungsfunktion benötigt. Im folgenden wird die Zuordnung zwischen kartesischen und Einheitskoordinaten $[-1, +1] \times [-1, +1] \times [-1, +1]$ auf dem Standardelement erläutert.

$$\begin{aligned} x &= Q_x(\xi, \eta, \zeta) \\ y &= Q_y(\xi, \eta, \zeta) \\ z &= Q_z(\xi, \eta, \zeta) \end{aligned} \quad (3.36)$$

sind Abbildungsfunktionen mit der Umkehrabbildung

$$\begin{aligned} \xi &= Q_\xi(x, y, z) \\ \eta &= Q_\eta(x, y, z) \\ \zeta &= Q_\zeta(x, y, z). \end{aligned} \quad (3.37)$$

Mit dieser Transformationsbeziehung ergibt sich für die Ansatzfunktionen aus Gleichung (3.26):

$$N(x, y, z) = N_{st}(Q_\xi(x, y, z), Q_\eta(x, y, z), Q_\zeta(x, y, z)) \quad (3.38)$$

Um die Steifigkeitsmatrix eines Elements zu ermitteln, muss zunächst die Transformationsvorschrift für die Verzerrungs-Verschiebungs-Beziehung aufgestellt werden. Die Element-Verzerrungen werden in Form von Ableitungen der Element-Verschiebungen nach den lokalen Koordinaten erhalten. Da die Ansatzfunktionen im natürlichen Koordinatensystem definiert sind, müssen die globalen Ableitungen nach x, y, z mit den lokalen Ableitungen nach ξ, η, ζ verknüpft werden. Die ersten Ableitungen ermitteln sich in Matrix-Schreibweise für ξ, η, ζ zu:

$$\begin{bmatrix} \frac{\partial N_{ist}}{\partial \xi} \\ \frac{\partial N_{ist}}{\partial \eta} \\ \frac{\partial N_{ist}}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} \quad (3.39)$$

Die Funktionen \mathbf{N}_i (3.30) sowie die *Jacobi-Matrix* \mathbf{J} in Gleichung (3.39) können direkt berechnet werden, da die Beziehungen für x, y , and z mit den lokalen Koordinaten ξ, η und ζ explizit gegeben ist. Die Ableitung bezüglich der globalen Koordinaten wird durch die Invertierung von \mathbf{J} bestimmt.

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial N_{ist}}{\partial \xi} \\ \frac{\partial N_{ist}}{\partial \eta} \\ \frac{\partial N_{ist}}{\partial \zeta} \end{bmatrix} \quad (3.40)$$

Aus Gleichung (3.40) folgt, dass die Inverse der Jacobi-Matrix existieren muss, um eine eindeutige Zuordnung zwischen lokalen und globalen Koordinaten zu gewährleisten. Das bedeutet, dass \mathbf{J} nicht singulär sein darf:

$$\det \mathbf{J} \neq 0 \quad \forall (\xi, \eta, \zeta) \quad (3.41)$$

Singularitäten können z. B. bei sich selbst überdeckenden Elementen auftreten. Die Transformation für Variablen und Integrationsbereiche ergibt sich somit zu

$$dx \, dy \, dz = \det \mathbf{J} \, d\xi \, d\eta \, d\zeta \quad (3.42)$$

und die Steifigkeitsmatrix (3.34) zu

$$\mathbf{K}^e := (k_{ij}^e)_{i,j \in I_e} := \left(\int_{-1}^1 \int_{-1}^1 \int_{-1}^1 (\mathbf{D}\mathbf{N}_i^e)^T \mathbf{C} \mathbf{D}\mathbf{N}_j^e \det \mathbf{J} \, d\xi \, d\eta \, d\zeta \right)_{i,j \in I_e} . \quad (3.43)$$

3.5 Die p -Version der Finite-Elemente-Methode

Die Reduktion des Diskretisierungsfehlers kann bei der h -Version durch eine lokale oder globale Netzverfeinerung erreicht werden. Die Grundidee bei der p -Version besteht darin, das Finite-Element-Netz unverändert zu belassen und den Polynomgrad der Ansatzfunktionen zu erhöhen um Konvergenz für die Finite-Element-Lösung zu erhalten. Für linear elliptische Probleme wurde bereits oft gezeigt, dass die p -Version zu sehr effizienten Approximationen mit einer exponentiellen Konvergenz im Fehler in der Energienorm führt und damit der klassischen h -Version überlegen ist [5, 37, 38, 39, 59, 60, 61, 62]. Sogar im Falle von Singularitäten in der exakten Lösung kann eine prä-asymptotisch exponentielle Konvergenz durch eine lokale Netzverfeinerung an den Orten der Singularitäten (p -Version auf geometrisch verfeinerten Netzen) oder durch Einführung von Randschicht-Elementen erreicht werden.

3.5.1 Hierarchische Ansatzfunktionen für eindimensionale Elemente

Orthogonale Polynome spielen eine wesentliche Rolle bei der p -Version der Finite-Elemente-Methode. Die hierarchischen Ansatzfunktionen basieren auf orthogonalen Legendre-Polynomen. Der grundlegende Unterschied zu den üblichen nicht-hierarchischen Ansatzfunktionen, welche üblicherweise aus Lagrange-Polynomen konstruiert sind besteht darin, dass bei hierarchischen Ansatzfunktionen alle Funktionen niedrigerer Ordnung in denen der höheren Ordnung enthalten sind. In Tabelle 3.1 sind nicht-hierarchische Ansatzfunktionen den hierarchischen Ansatzfunktionen für den eindimensionalen Fall gegenübergestellt:

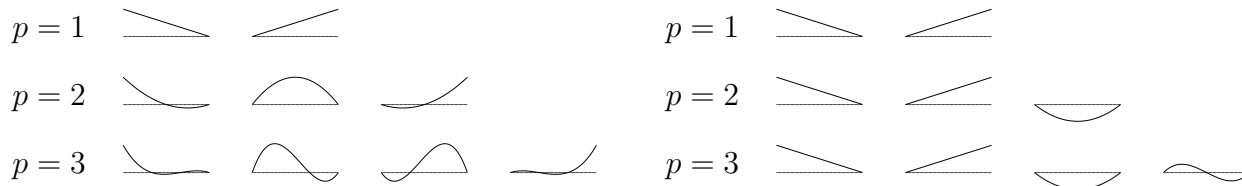


Tabelle 3.1: Eindimensionale nicht-hierarchisch (links) und hierarchisch (rechts) aufgebaute Ansatzfunktionen für $p = 1, 2, 3$

Eindimensionale hierarchische Ansatzfunktionen können nach [77] wie folgt definiert werden:

$$N_1(\xi) = 1/2(1 - \xi) \quad (3.44)$$

$$N_2(\xi) = 1/2(1 + \xi) \quad (3.45)$$

$$N_i(\xi) = \phi_{i-1}(\xi), \quad i = 3, 4, \dots, p + 1 \quad (3.46)$$

mit

$$\phi_j(\xi) = \sqrt{\frac{2j-1}{2}} \int_{-1}^{\xi} L_{j-1}(x) dx = \frac{1}{\sqrt{4j-2}} (L_j(\xi) - L_{j-2}(\xi)), \quad j = 2, 3, \dots \quad (3.47)$$

wobei L_j die Legendre-Polynome sind:

$$L_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n, \quad x \in (-1, 1), \quad n = 0, 1, 2, \dots \quad (3.48)$$

Die linearen Ansatzfunktionen aus den Gleichungen (3.44) und (3.45) sind die so genannten *Knotenmoden*. Da

$$N_i(-1) = N_i(1) = 0, \quad i = 3, 4, \dots \quad (3.49)$$

werden die Funktionen $N_i(\xi)$, $i = 3, 4, \dots$ *innere Moden* genannt. Die Orthogonalitätseigenschaft der Legendre-Polynome führt zu

$$\int_{-1}^1 \frac{dN_i}{d\xi} \frac{dN_j}{d\xi} d\xi = \delta_{ij}, \quad i \geq 3 \wedge i \geq 1 \quad \vee \quad i \geq 1 \wedge i \geq 3. \quad (3.50)$$

In [20, 58] wird gezeigt, dass die Konditionszahl der Element-Steifigkeitsmatrix mit hierarchischen Ansatzfunktionen signifikant günstiger ist als die auf nicht-hierarchischen Ansatzfunktionen basierende.

3.5.2 Hierarchische Ansatzfunktionen für Hexaeder-Elemente

Die in dieser Arbeit verwendete Implementierung der p -Version für dreidimensionale Probleme basiert auf einer Hexaeder-Formulierung mit hierarchischen Ansatzfunktionen [77]. Verglichen mit Tetraeder- bzw. Pentaeder-Elementen weisen Hexaeder-Elemente (siehe Abbildung 3.2) entscheidende Vorteile auf.

- Hexaeder-Elementformulierungen führen zu einer höheren Genauigkeit.
- Hexaeder-Elemente sind sehr gut geeignet für dünne Strukturen, wie sie bei Tiefziehteilen üblich sind. Es ist möglich den Polynomgrad in Dickenrichtung unabhängig vom Polynomgrad in Elementebene zu wählen. Dies führt zu effizienten Diskretisierungen.
- Die numerische Integration kann leicht mithilfe des Gaußschen Quadraturverfahrens durchgeführt werden.

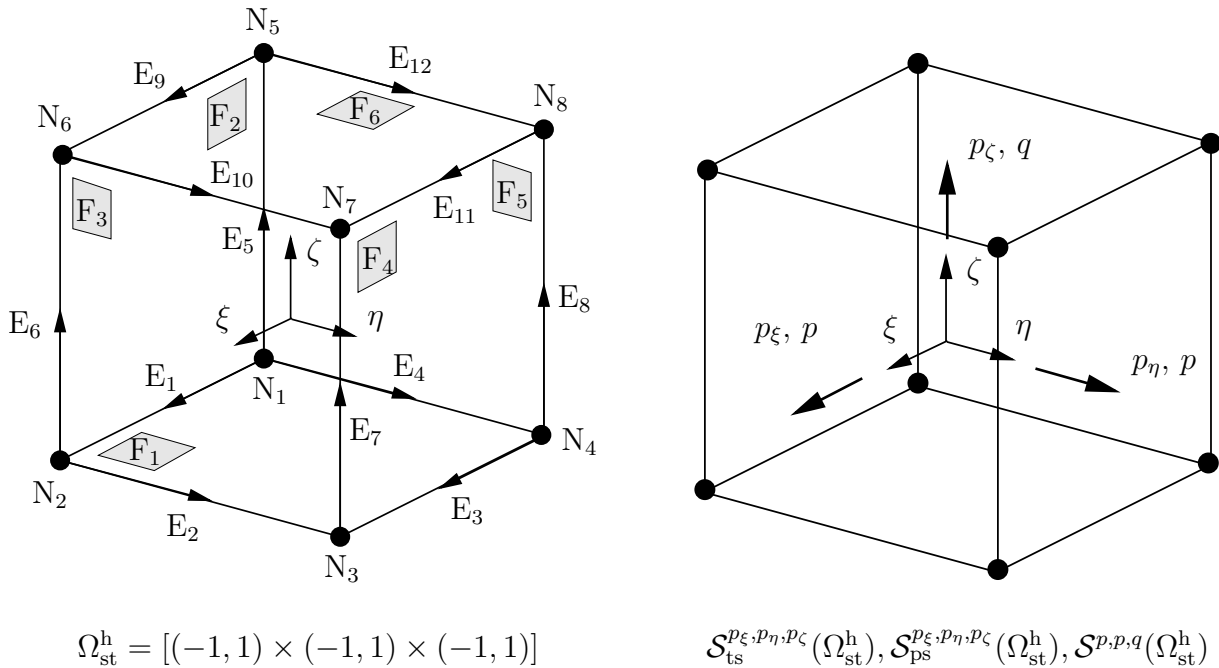


Abbildung 3.2: Standard-Hexaeder-Element Ω_{st}^h : Definition von Knoten, Kanten, Flächen und Polynomgrad

Dreidimensionale Ansatzfunktionen lassen sich in vier Gruppen einteilen:

1. **Knoten-Moden:** Die Knoten-Moden

$$N_{1,1,1}^{N_i}(\xi, \eta, \zeta) = \frac{1}{8} (1 + \xi_i \xi)(1 + \eta_i \eta)(1 + \zeta_i \zeta), \quad i = 1, \dots, 8 \quad (3.51)$$

sind die üblichen trilinearen Ansatzfunktion. (ξ_i, η_i, ζ_i) sind die lokalen Koordinaten des i -ten Knotens.

2. **Kanten-Moden:** Die Kanten-Moden werden jeweils linear zu den gegenüberliegenden Kanten auf den beiden benachbarten Flächen ausgeblendet und verschwinden an den sie berandenden Randknoten. Für die lokale Kante E_1 im Element (siehe Abbildung 3.2) ergibt sich z. B. die zugehörige Ansatzfunktion zu

$$N_{i,1,1}^{E_1}(\xi, \eta, \zeta) = \frac{1}{4}(1 - \eta)(1 - \zeta)\phi_i(\xi). \quad (3.52)$$

3. **Flächen-Moden:** Flächen-Moden werden linear zu ihrer im Element gegenüberliegenden Fläche ausgeblendet und verschwinden an ihren Rändern. Für die lokale Fläche F_1 im Element ergibt sich z. B. die zugehörige Ansatzfunktion zu

$$N_{i,j,1}^{F_1}(\xi, \eta, \zeta) = \frac{1}{2}(1 - \zeta)\phi_i(\xi)\phi_j(\eta). \quad (3.53)$$

4. **Innere Moden:** Die inneren Moden

$$N_{i,j,k}^{\text{int}}(\xi, \eta, \zeta) = \phi_i(\xi)\phi_j(\eta)\phi_k(\zeta) \quad (3.54)$$

sind rein lokal und verschwinden an den Rändern des Elementes. Sie sind daher nicht mit benachbarten Elementen gekoppelt und können aus dem globalen Gleichungssystem auf Elementebene herauskondensiert werden.

Innere Moden können die Effizienz einer Finite-Elemente-Berechnung erhöhen. Im zu lösenden Gleichungssystem

$$\begin{bmatrix} \mathbf{K}_{ee} & \mathbf{K}_{ei} \\ \mathbf{K}_{ie} & \mathbf{K}_{ii} \end{bmatrix} \begin{bmatrix} \mathbf{u}_e \\ \mathbf{u}_i \end{bmatrix} = \begin{bmatrix} \mathbf{f}_e \\ \mathbf{f}_i \end{bmatrix}, \quad (3.55)$$

steht der Index e für die externen Moden und i für die inneren Moden. Durch Kondensation der inneren Moden aus der Systemsteifigkeitsmatrix ergeben sich die Verschiebungen für die inneren Moden zu

$$\mathbf{u}_i = \mathbf{K}_{ii}^{-1}(\mathbf{f}_i - \mathbf{K}_{ie}\mathbf{u}_e) \quad (3.56)$$

und das Gleichungssystem (3.55) reduziert sich zu:

$$(\mathbf{K}_{ee} - \mathbf{K}_{ei}\mathbf{K}_{ii}^{-1}\mathbf{K}_{ie})\mathbf{u}_e = (\mathbf{f}_e - \mathbf{K}_{ei}\mathbf{K}_{ii}^{-1}\mathbf{f}_i) \quad (3.57)$$

$$\tilde{\mathbf{K}}_e\mathbf{u}_e = \tilde{\mathbf{f}}_e \quad (3.58)$$

Dadurch steigt zwar zum einen die Rechenzeit auf Elementebene an, auf der anderen Seite reduziert sich die Rechenzeit beim Lösen des Gleichungssystems aufgrund einer geringeren Größe und der kleineren Konditionszahl der Gesamtsteifigkeitsmatrix.

Es wurden die drei verschiedenen Ansatzräume *trunk space* $\mathcal{S}_{\text{ts}}^{p_\xi, p_\eta, p_\zeta}(\Omega_{\text{st}}^{\text{h}})$, *tensor product space* $\mathcal{S}_{\text{ps}}^{p_\xi, p_\eta, p_\zeta}(\Omega_{\text{st}}^{\text{h}})$ und *anisotropic tensor product space* $\mathcal{S}^{p, p, q}(\Omega_{\text{st}}^{\text{h}})$ implementiert. Eine detaillierte Beschreibung der einzelnen Ansatzräume wird in [20, 77, 78] gegeben.

Der Unterschied zwischen *trunk space* und *tensor product space* bezieht sich auf Flächen- und innere Moden. Beim *trunk space* werden für Flächen- und innere Moden nur Monome berücksichtigt, deren Summe der Exponenten kleiner gleich dem maximalen Polynomgrad in Richtung der lokalen Koordinaten ist. Der *tensor product space* enthält alle Monome und wird daher auch als voller Ansatzraum bezeichnet. Die Flächen-Moden der beiden Ansatzräume ergeben sich für die lokale Fläche 1 zu:

trunk space	tensor product space
$i = 2, \dots, p_\xi - 2$	$i = 2, \dots, p_\xi$
$j = 2, \dots, p_\eta - 2$	$j = 2, \dots, p_\eta$
$i + j = 4, \dots, \max\{p_\xi, p_\eta\}$	

Für die inneren Moden $N_{i,j,k}^{\text{int}}(\xi, \eta, \zeta) = \phi_i(\xi)\phi_j(\eta)\phi_k(\zeta)$ ergibt sich analog:

trunk space	tensor product space
$i = 2, \dots, p_\xi - 4$	$i = 2, \dots, p_\xi$
$j = 2, \dots, p_\eta - 4$	$j = 2, \dots, p_\eta$
$k = 2, \dots, p_\zeta - 4$	$k = 2, \dots, p_\zeta$
$i + j + k = 6, \dots, \max\{p_\xi, p_\eta, p_\zeta\}$	

Der Polynomgrad für die Ansatzräume $\mathcal{S}_{\text{ts}}^{p_\xi, p_\eta, p_\zeta}(\Omega_{\text{st}}^{\text{h}})$ und $\mathcal{S}_{\text{ps}}^{p_\xi, p_\eta, p_\zeta}(\Omega_{\text{st}}^{\text{h}})$ kann für jede lokale Richtung und für jede primäre Variable $\mathbf{u} = [u_x, u_y, u_z]^T$ separat gewählt werden (siehe Abbildung 3.2).

Der in [77] vorgestellte Ansatzraum $\mathcal{S}^{p, p, q}(\Omega_{\text{st}}^{\text{h}})$ stellt einen anisotropen Satz von Ansatzfunktionen speziell für dünnwandige Strukturen wie sie bei der Blechumformung vorliegen, dar. Der Polynomgrad p ist dabei mit den lokalen Koordinatenrichtungen ξ und η verknüpft, wobei die Ansatzfunktionen der lokalen Flächen 1 und 6 denen des *trunk space* $\mathcal{S}_{\text{ts}}^{p_\xi, p_\eta, p_\zeta}(\Omega_{\text{st}}^{\text{h}})$ mit $p = p_\xi = p_\eta$ entsprechen. q definiert den Polynomgrad aller Ansatzfunktionen in lokale ζ -Richtung. Die lokalen Flächen 2,3,4,5 besitzen dabei die Ansatzfunktionen des vollen *tensor product space* $\mathcal{S}_{\text{ps}}^{p_\xi, p_\eta, p_\zeta}(\Omega_{\text{st}}^{\text{h}})$.

Bei anisotropen Ansatzräumen für dünnwandige Strukturen ist es wichtig die lokale Orientierung der Elemente zu beachten. Durch die Verwendung von Hexaederelementen ist dies im Gegensatz zu Tetraederelementen leicht sicher zu stellen. Abbildung 3.3 zeigt die Orientierung eines Hexaederelements für dünnwandige Strukturen. Dabei korrespondiert die lokale ζ -Koordinate mit der Dickenrichtung der Struktur. Durch eine einheitliche Orientierung aller Elemente einer Diskretisierung kann somit ein unterschiedlicher Polynomgrad für die Ansätze

in der Ebene und in Dickenrichtung verwendet werden.

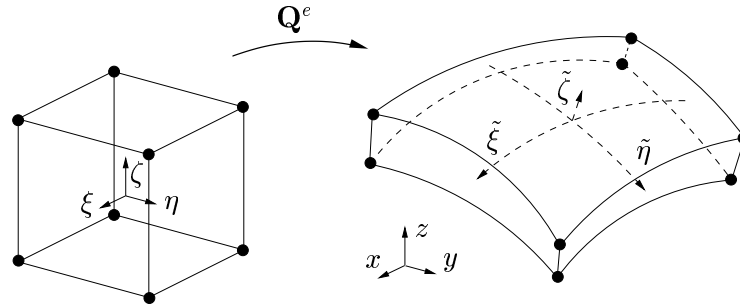


Abbildung 3.3: Orientierung für dünnwandige Strukturen

3.6 Geometrisch nichtlineare Probleme

Oft ist es in der Strukturmechanik ausreichend, kleine Deformationen und Verzerrungen zu betrachten. In diesen Fällen wird eine lineare Theorie verwendet, solange ein elastisches Materialverhalten vorausgesetzt werden kann. Nichtlineare Probleme führen im allgemeinen zu einem System nichtlinearer partieller Differentialgleichungen. Mithilfe der Diskretisierung mit Finiten Elementen reduziert sich dieses System für zeitunabhängige Probleme auf ein nichtlineares algebraisches Gleichungssystem

$$\Psi(\mathbf{a}) = \mathbf{K}_T(\mathbf{a})\mathbf{a} - \mathbf{f} = \mathbf{0} \quad (3.59)$$

bei denen große Verschiebungen auftreten, obwohl die Verzerrungen klein sind (St. Venant-Kirchhoff). Dabei ist die Tangentensteifigkeit $\mathbf{K}_T(\mathbf{a})$ definiert als

$$\mathbf{K}_T(\mathbf{a}) = \int_{\Omega} \mathbf{B}^T(\mathbf{a})\boldsymbol{\sigma} d\Omega \quad (3.60)$$

wobei

$$\mathbf{B}(\mathbf{a}) = \mathbf{B}_l + \mathbf{B}_{nl}(\mathbf{a}) \quad (3.61)$$

mit dem linearen Anteil

$$\mathbf{B}_l = \mathbf{D}\mathbf{N}_i \quad i \in I_e. \quad (3.62)$$

Gleichung (3.59) drückt allgemein eine nichtlineare Beziehung zwischen Kraft und Verschiebung aus. Während lineare Systeme direkt gelöst werden können, müssen für nichtlineare Systeme iterative Lösungsverfahren zur Anwendung kommen. Neben Reduktionsverfahren, welche zunächst auf vereinfachte nichtlineare Gleichungssysteme führen, sind Minimierungs- und Linearisierungsverfahren gebräuchliche Methoden. Ein in der Praxis häufig eingesetztes Verfahren zur Bestimmung der Verschiebung \mathbf{a} bei einer gegebenen Belastung \mathbf{f} ist das Newton-Raphson-Verfahren [16, 17].

3.6.1 Greenscher Verzerrungstensor

In Abbildung 3.4 beschreibt $d\mathbf{X}$ ein Linienelement in der Referenzkonfiguration und $d\mathbf{x}$ ein Linienelement der Momentankonfiguration. Die Verzerrung wird durch den Green-Lagrangeschen Verzerrungstensor \mathbf{E} als Änderung des Skalarprodukts der Linienelemente von Referenz- und Momentankonfiguration beschrieben:

$$d\mathbf{x}^2 - d\mathbf{X}^2 = 2d\mathbf{X}^T \mathbf{E} d\mathbf{X} \quad (3.63)$$

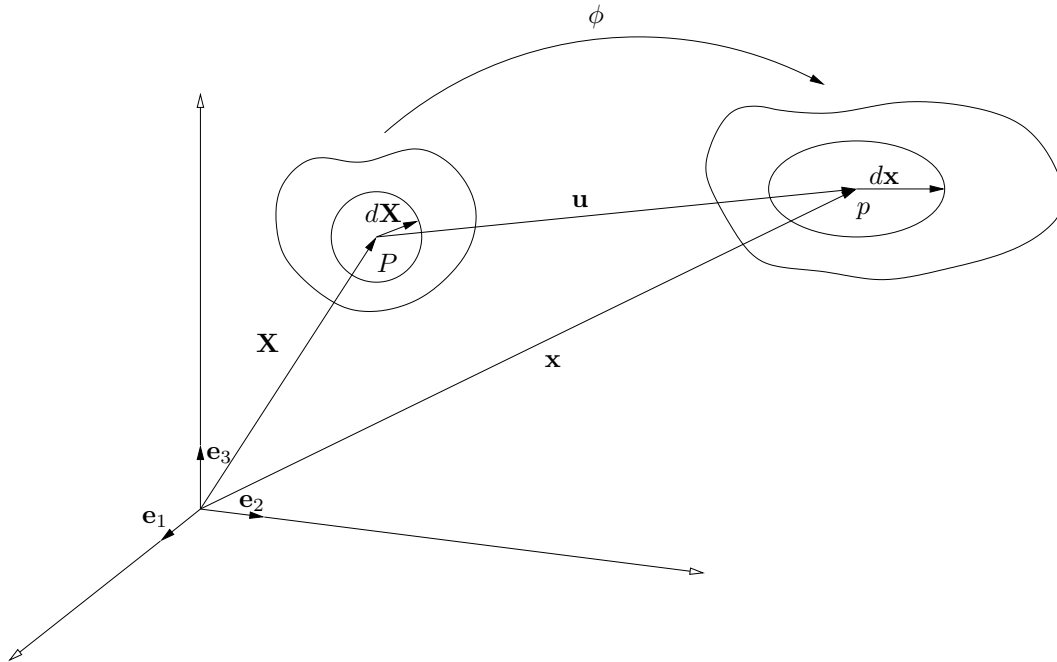


Abbildung 3.4: Referenz- und Momentankonfiguration

Nach der Deformation geht \mathbf{X} über in \mathbf{x} .

$$\mathbf{x} = \mathbf{X} + \mathbf{u} \quad (3.64)$$

Dabei bezeichnet \mathbf{u} den Verschiebungsvektor. Um den Deformationsprozess zu beschreiben, wird \mathbf{F} als Deformationsgradient, welcher die Tangentenvektoren der Ausgangs- auf die Momentankonfiguration abbildet, eingeführt. Dieser Tensor ermittelt sich über das totale Differential:

$$d\mathbf{x} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} d\mathbf{X} = \frac{\partial (\mathbf{X} + \mathbf{u})}{\partial \mathbf{X}} d\mathbf{X} \quad (3.65)$$

Diese Beziehung kann geschrieben werden als

$$d\mathbf{x} = \mathbf{F} d\mathbf{X} = [\mathbf{I} + \text{Grad } \mathbf{u}] d\mathbf{X} \quad (3.66)$$

wobei $\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}$ den Deformationsgradienten und $\text{Grad } \mathbf{u} = \frac{\partial \mathbf{u}}{\partial \mathbf{X}}$ den Verschiebungsgradienten darstellt. Gleichung (3.66) ermöglicht somit den *push forward*, also die Transformation differentieller Größen des Ausgangszustands auf die Momentankonfiguration. Mit der Inversen des

Deformationsgradienten

$$d\mathbf{X} = \mathbf{F}^{-1}d\mathbf{x} = (\mathbf{I} - \text{grad } \mathbf{u})d\mathbf{x} \quad \text{grad } \mathbf{u} = \frac{\partial \mathbf{u}}{\partial \mathbf{x}} \quad (3.67)$$

lässt sich der so genannte *pull back* durchführen, bei dem die Größen der Momentankonfiguration in die Ausgangskonfiguration übergeführt werden. In Gleichung (3.63) kann nun mithilfe des Deformationsgradienten aus Gleichung (3.66) die Momentankonfiguration über die Referenzkonfiguration ausgedrückt werden:

$$d\mathbf{X} \cdot \mathbf{F}^T \mathbf{F} d\mathbf{X} - d\mathbf{X} \cdot d\mathbf{X} = 2d\mathbf{X}^T \mathbf{E} d\mathbf{X} \quad (3.68)$$

$$d\mathbf{X} \cdot (\mathbf{F}^T \mathbf{F} - \mathbf{I})d\mathbf{X} = 2d\mathbf{X}^T \mathbf{E} d\mathbf{X} \quad (3.69)$$

Der Greensche Verzerrungstensor kann nun bezüglich der Referenzkonfiguration aus Gleichung (3.69) geschrieben werden als

$$\mathbf{E} = \frac{1}{2} [\mathbf{F}^T \mathbf{F} - \mathbf{I}] = \frac{1}{2} [\text{Grad } \mathbf{u} + (\text{Grad } \mathbf{u})^T] + \frac{1}{2} (\text{Grad } \mathbf{u})^T \text{Grad } \mathbf{u}. \quad (3.70)$$

Dabei beschreibt $\frac{1}{2}(\text{Grad } \mathbf{u} + (\text{Grad } \mathbf{u})^T)$ den linearen und $\frac{1}{2}(\text{Grad } \mathbf{u})^T \text{Grad } \mathbf{u}$ den nichtlinearen Anteil der Verzerrung \mathbf{E} :

$$\mathbf{E} = \mathbf{E}_l + \mathbf{E}_{nl} \quad (3.71)$$

Der erste Term in Gleichung (3.70) enthält die linearen Verschiebungsableitungen.

3.6.2 Linearisierung der virtuellen Arbeit

Im folgenden soll der Tensor zweiter Stufe \mathbf{E} aus Gleichung (3.71) mithilfe der Voigt-Notation in eine rechnernahe Formulierung überführt werden:

$$\begin{aligned} \boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_l + \boldsymbol{\varepsilon}_{nl} &= \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{bmatrix} = \begin{bmatrix} \frac{\partial u_x}{\partial X} \\ \frac{\partial u_y}{\partial Y} \\ \frac{\partial u_z}{\partial Z} \\ \frac{\partial u_x}{\partial Y} + \frac{\partial u_y}{\partial X} \\ \frac{\partial u_y}{\partial Z} + \frac{\partial u_z}{\partial Y} \\ \frac{\partial u_x}{\partial Z} + \frac{\partial u_z}{\partial X} \end{bmatrix} + \begin{bmatrix} \frac{1}{2} \left(\left(\frac{\partial u_x}{\partial X} \right)^2 + \left(\frac{\partial u_y}{\partial X} \right)^2 + \left(\frac{\partial u_z}{\partial X} \right)^2 \right) \\ \frac{1}{2} \left(\left(\frac{\partial u_x}{\partial Y} \right)^2 + \left(\frac{\partial u_y}{\partial Y} \right)^2 + \left(\frac{\partial u_z}{\partial Y} \right)^2 \right) \\ \frac{1}{2} \left(\left(\frac{\partial u_x}{\partial Z} \right)^2 + \left(\frac{\partial u_y}{\partial Z} \right)^2 + \left(\frac{\partial u_z}{\partial Z} \right)^2 \right) \\ \left(\frac{\partial u_x}{\partial X} \right) \left(\frac{\partial u_x}{\partial Y} \right) + \left(\frac{\partial u_y}{\partial X} \right) \left(\frac{\partial u_y}{\partial Y} \right) + \left(\frac{\partial u_z}{\partial X} \right) \left(\frac{\partial u_z}{\partial Y} \right) \\ \left(\frac{\partial u_x}{\partial Y} \right) \left(\frac{\partial u_x}{\partial Z} \right) + \left(\frac{\partial u_y}{\partial Y} \right) \left(\frac{\partial u_y}{\partial Z} \right) + \left(\frac{\partial u_z}{\partial Y} \right) \left(\frac{\partial u_z}{\partial Z} \right) \\ \left(\frac{\partial u_x}{\partial X} \right) \left(\frac{\partial u_x}{\partial Z} \right) + \left(\frac{\partial u_y}{\partial X} \right) \left(\frac{\partial u_y}{\partial Z} \right) + \left(\frac{\partial u_z}{\partial X} \right) \left(\frac{\partial u_z}{\partial Z} \right) \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial u_x}{\partial X} \\ \frac{\partial u_y}{\partial Y} \\ \frac{\partial u_z}{\partial Z} \\ \frac{\partial u_x}{\partial Y} + \frac{\partial u_y}{\partial X} \\ \frac{\partial u_y}{\partial Z} + \frac{\partial u_z}{\partial Y} \\ \frac{\partial u_x}{\partial Z} + \frac{\partial u_z}{\partial X} \end{bmatrix} + \frac{1}{2} \underbrace{\begin{bmatrix} \frac{\partial u_x}{\partial X} & 0 & 0 & \frac{\partial u_y}{\partial X} & 0 & 0 & \frac{\partial u_z}{\partial X} & 0 & 0 \\ 0 & \frac{\partial u_x}{\partial Y} & 0 & 0 & \frac{\partial u_y}{\partial Y} & 0 & 0 & \frac{\partial u_z}{\partial Y} & 0 \\ 0 & 0 & \frac{\partial u_x}{\partial Z} & 0 & 0 & \frac{\partial u_y}{\partial Z} & 0 & 0 & \frac{\partial u_z}{\partial Z} \\ \frac{\partial u_x}{\partial Y} & \frac{\partial u_x}{\partial X} & 0 & \frac{\partial u_y}{\partial Y} & \frac{\partial u_y}{\partial X} & 0 & \frac{\partial u_z}{\partial Y} & \frac{\partial u_z}{\partial X} & 0 \\ 0 & \frac{\partial u_x}{\partial Z} & \frac{\partial u_x}{\partial Y} & 0 & \frac{\partial u_y}{\partial Z} & \frac{\partial u_y}{\partial Y} & 0 & \frac{\partial u_z}{\partial Z} & \frac{\partial u_z}{\partial Y} \\ \frac{\partial u_x}{\partial Z} & 0 & \frac{\partial u_x}{\partial X} & \frac{\partial u_y}{\partial Z} & 0 & \frac{\partial u_y}{\partial X} & \frac{\partial u_z}{\partial Z} & 0 & \frac{\partial u_z}{\partial X} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \frac{\partial u_x}{\partial X} \\ \frac{\partial u_x}{\partial Y} \\ \frac{\partial u_x}{\partial Z} \\ \frac{\partial u_y}{\partial X} \\ \frac{\partial u_y}{\partial Y} \\ \frac{\partial u_y}{\partial Z} \\ \frac{\partial u_z}{\partial X} \\ \frac{\partial u_z}{\partial Y} \\ \frac{\partial u_z}{\partial Z} \end{bmatrix}}_{\boldsymbol{\theta}} \quad (3.72) \end{aligned}$$

Dabei lässt sich der nichtlineare Anteil $\boldsymbol{\varepsilon}_{nl}$ als Produkt

$$\boldsymbol{\varepsilon}_{nl} = \frac{1}{2} \mathbf{A} \boldsymbol{\theta} \quad (3.73)$$

formulieren. Weiter kann $d\boldsymbol{\varepsilon}_{nl}$ linearisiert werden zu

$$d\boldsymbol{\varepsilon}_{nl} = \frac{1}{2} d\mathbf{A} \boldsymbol{\theta} + \frac{1}{2} \mathbf{A} d\boldsymbol{\theta}. \quad (3.74)$$

Es ist leicht zu zeigen, dass

$$d\mathbf{A} \boldsymbol{\theta} = \mathbf{A} d\boldsymbol{\theta} \quad (3.75)$$

gilt. Damit vereinfacht sich Gleichung (3.74) zu

$$d\boldsymbol{\varepsilon}_{nl} = \mathbf{A} d\boldsymbol{\theta}. \quad (3.76)$$

In diskreter Form wird $\boldsymbol{\theta}$ als Produkt der Ableitungen der Ansatzfunktionen \mathbf{N} und der Freiheitsgrade \mathbf{a} formuliert

$$\boldsymbol{\theta} = \mathbf{G} \mathbf{a} \quad (3.77)$$

wobei

$$\mathbf{G} = \begin{bmatrix} \bar{\mathbf{G}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{G}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \bar{\mathbf{G}} \end{bmatrix}, \quad \bar{\mathbf{G}} = \begin{bmatrix} \frac{\partial N_1}{\partial X} & \frac{\partial N_2}{\partial X} & \cdots & \frac{\partial N_n}{\partial X} \\ \frac{\partial N_1}{\partial Y} & \frac{\partial N_2}{\partial Y} & \cdots & \frac{\partial N_n}{\partial Y} \\ \frac{\partial N_1}{\partial Z} & \frac{\partial N_2}{\partial Z} & \cdots & \frac{\partial N_n}{\partial Z} \end{bmatrix}. \quad (3.78)$$

Mit Gleichung (3.77) folgt aus Gleichung (3.76)

$$d\boldsymbol{\varepsilon}_{nl} = \mathbf{A} \mathbf{G} d\mathbf{a} \quad (3.79)$$

sowie für den nichtlinearen Anteil \mathbf{B}_{nl} von \mathbf{B}

$$\mathbf{B}_{nl} = \mathbf{A} \mathbf{G}. \quad (3.80)$$

Durch Differentiation von $\boldsymbol{\Psi}$ nach \mathbf{a} folgt aus Gleichung (3.59)

$$\frac{d\boldsymbol{\Psi}}{d\mathbf{a}} = \int_{\Omega} d\mathbf{B}^T \boldsymbol{\sigma} d\Omega + \int_{\Omega} \mathbf{B}^T d\boldsymbol{\sigma} d\Omega = \mathbf{K}_T \quad (3.81)$$

wobei

$$d\boldsymbol{\sigma} = \mathbf{C} d\boldsymbol{\varepsilon} = \mathbf{C} \mathbf{B} d\mathbf{a}. \quad (3.82)$$

Aus Gleichung (3.61) folgt aus der Differentiation nach \mathbf{a} :

$$d\mathbf{B} = d\mathbf{B}_{nl} \quad (3.83)$$

Gleichung (3.81) vereinfacht sich somit zu

$$\frac{d\Psi}{d\mathbf{a}} = \int_{\Omega} d\mathbf{B}_{nl}^T \boldsymbol{\sigma} d\Omega + \int_{\Omega} \mathbf{B}^T \mathbf{C} \mathbf{B} d\Omega. \quad (3.84)$$

Der erste Term von Gleichung (3.84) wird als *geometrische* Steifigkeit bezeichnet und ist definiert als \mathbf{K}_{σ} :

$$\mathbf{K}_{\sigma} d\mathbf{a} := \int_{\Omega} d\mathbf{B}_{nl}^T \boldsymbol{\sigma} d\Omega \quad (3.85)$$

Mit Gleichung (3.80) ergibt sich für Gleichung (3.84) $\mathbf{K}_{\sigma} d\mathbf{a}$ zu

$$\mathbf{K}_{\sigma} d\mathbf{a} = \int_{\Omega} \mathbf{G}^T d\mathbf{A}^T \boldsymbol{\sigma} d\Omega. \quad (3.86)$$

Durch Definition des Spannungstensors zweiter Stufe $\hat{\boldsymbol{\sigma}}$

$$\hat{\boldsymbol{\sigma}} = \begin{bmatrix} \bar{\boldsymbol{\sigma}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \bar{\boldsymbol{\sigma}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \bar{\boldsymbol{\sigma}} \end{bmatrix}, \quad \bar{\boldsymbol{\sigma}} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ & \sigma_{yy} & \sigma_{yz} \\ \text{sym.} & & \sigma_{zz} \end{bmatrix} \quad (3.87)$$

ergibt sich Gleichung (3.86) zu

$$\mathbf{K}_{\sigma} d\mathbf{a} = \int_{\Omega} \mathbf{G}^T \hat{\boldsymbol{\sigma}} \mathbf{G} da d\Omega \quad (3.88)$$

und damit

$$\mathbf{K}_{\sigma} = \int_{\Omega} \mathbf{G}^T \hat{\boldsymbol{\sigma}} \mathbf{G} d\Omega. \quad (3.89)$$

Die Linearisierung von Gleichung (3.81) führt zur materiellen (rechter Term in Gleichung (3.84)) und geometrischen Steifigkeit in Gleichung (3.89).

Anmerkung:

In den folgenden Abschnitten verwendete tiefgestellte Indizes kennzeichnen inkrementelle Prediktor-Schritte, hochgestellte kennzeichnen Korrektor-Schritte.

3.6.3 Inkrementelle Lösung

Das inkrementelle oder *Eulersche* Lösungsverfahren entspricht der wiederholten Anwendung von

$$\Delta \mathbf{a} = \left(\frac{d\mathbf{f}}{d\mathbf{a}} \right)^{-1} \Delta \mathbf{f} = \mathbf{K}_T^{-1} \Delta \mathbf{f} \quad (3.90)$$

Im ersten Schritt wird die Anfangssteifigkeit \mathbf{K}_{T_0} und damit

$$\mathbf{a}_1 = \Delta \mathbf{a}_0 = (\mathbf{K}_{T_0})^{-1} \Delta \mathbf{f} \quad (3.91)$$

berechnet, wobei $\Delta \mathbf{f}$ das aufgebrachte Lastinkrement ist. Ab dem zweiten Schritt wird $\mathbf{K}_{T_{n-1}}$ bezogen auf \mathbf{a}_{n-1} berechnet um

$$\Delta \mathbf{a}_{n-1} = (\mathbf{K}_{T_{n-1}})^{-1} (\mathbf{a}_{n-1}) \Delta \mathbf{f} \quad (3.92)$$

zu bestimmen, so dass sich

$$\mathbf{a}_n = \mathbf{a}_{n-1} + \Delta \mathbf{a}_{n-1}. \quad (3.93)$$

ergibt. Wie aus Abbildung 3.5 ersichtlich ist, weicht die erhaltene Lösung von der wahren Gleichgewichtskurve ab.

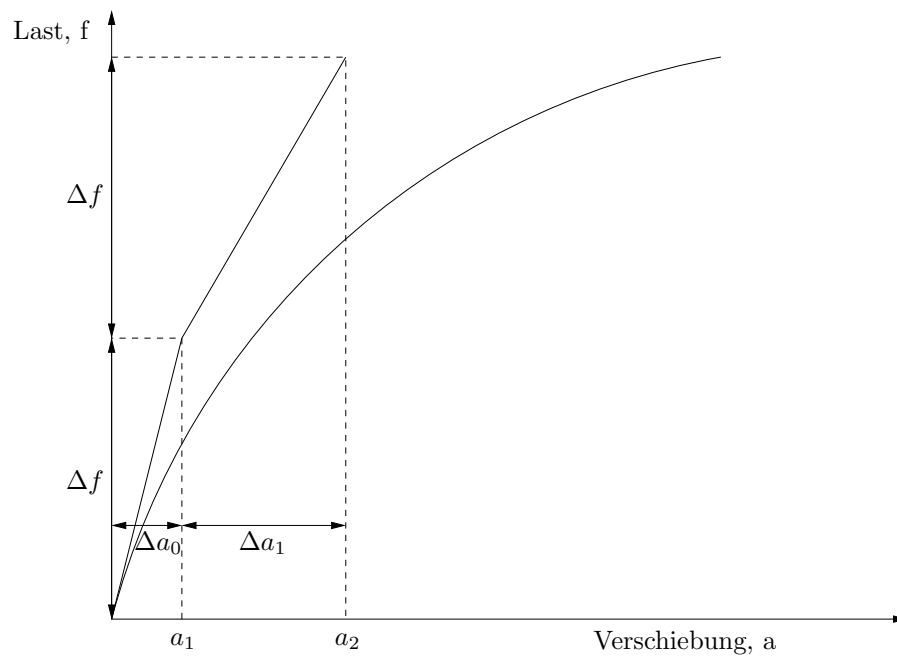


Abbildung 3.5: Inkrementelles Lösungsverfahren

3.6.4 Das Newton-Raphson-Verfahren

Die Berechnung von \mathbf{a}^0 liefert eine erste Näherung, für welche $\Psi(\mathbf{a}^0) \neq \mathbf{0}$ gilt. Eine verbesserte Lösung lässt sich mithilfe einer abgebrochenen Taylorreihen-Entwicklung erreichen.

$$\Psi(\mathbf{a}^{i+1}) := \Psi(\mathbf{a}^i) + \left(\frac{d\Psi}{d\mathbf{a}}\right)^i \Delta \mathbf{a}^i = 0 \quad (3.94)$$

mit

$$\mathbf{a}^{i+1} = \mathbf{a}^i + \Delta \mathbf{a}^i. \quad (3.95)$$

In diesem Ausdruck bezeichnet

$$\left(\frac{d\Psi}{d\mathbf{a}}\right)^i =: \mathbf{K}_T^i \quad (3.96)$$

die vollen Tangentensteifigkeitsmatrix. Eine verbesserte Lösung kann durch Berechnung von

$$\Delta \mathbf{a}^i = -(\mathbf{K}_T^i)^{-1} \Psi(\mathbf{a}^i) \quad (3.97)$$

erreicht werden.

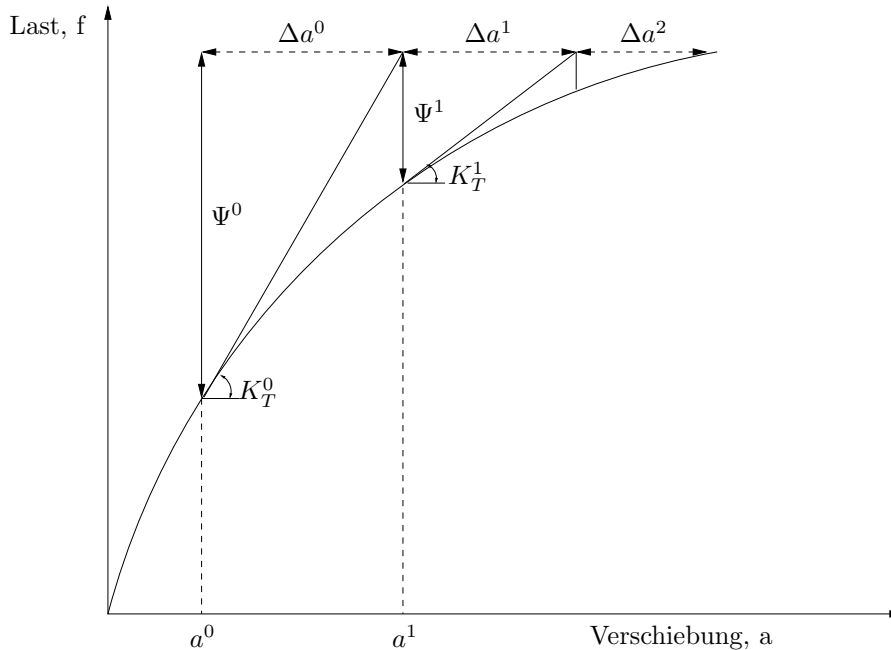


Abbildung 3.6: Das Newton-Raphson Verfahren

3.6.5 Inkrementelles Newton-Raphson-Verfahren

Das iterative Verfahren führt zu einer Lösung für einen einzelnen Lastschritt. Für die Ermittlung des gesamten Belastungs-Verschiebungs-Verlauf hat es sich bewährt, inkrementelle mit iterativen Verfahren zu koppeln.

Die inkrementelle Lösung wird dann als *Prediktor*-Schritt bezeichnet, welcher den Anfangswert \mathbf{a}^0 für den iterativen Prozess liefert. Das Newton-Raphson-Verfahren wird wiederholt in *Korrektor*-Schritten angewendet, bis $|\Psi(\mathbf{a}^i)| \leq \varepsilon$, wobei ε eine vorher festgelegte Genauigkeit ist. Üblicherweise reichen hierzu wenige Schritte aus, da das volle Newton-Raphson ein quadratisches Konvergenzverhalten zeigt. Abbildung 3.7 zeigt die allgemeine Kombination von inkrementellen Prediktor-Schritten mit dem Newton-Raphson Korrektor-Verfahren.

3.6.6 Modifiziertes Newton-Raphson-Verfahren

Ein Nachteil des vollen Newton-Raphson-Verfahrens ist, dass für jeden Korrektor-Schritt eine neue Tangentensteifigkeit berechnet werden und damit ein vollständig neues Gleichungssystem

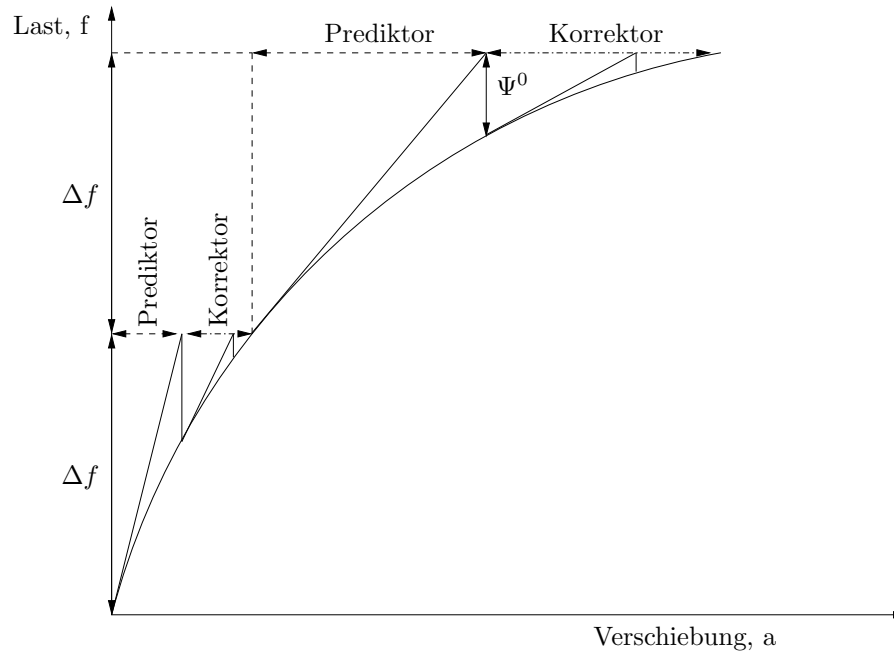


Abbildung 3.7: Kombination inkrementeller Prediktor-Schritte mit Newton-Raphson Iterationen

gelöst werden muss. Um dies zu vermeiden, kann das Verfahren durch Erweiterung um die Annahme

$$\mathbf{K}_{T_n}^i = \mathbf{K}_{T_n}^0 \quad (3.98)$$

modifiziert werden. Mit dieser Vereinbarung geht Gleichung (3.97) über in

$$\Delta \mathbf{a}^i = - (\mathbf{K}_T^0)^{-1} \Psi^i. \quad (3.99)$$

Im Gegensatz zum vollen Newton-Raphson-Verfahren ist ersichtlich, dass zum einen der Rechenaufwand in jedem Iterationsschritt sinkt, zum anderen das Konvergenzverhalten ein nur noch lineares Verhalten aufzeigt. Für viele Anwendungen wie z. B. bei geometrisch schwach nichtlinearen Problemen hat sich das modifizierte Verfahren als günstige Alternative erwiesen.

3.6.7 Erhöhung der Effizienz für statisch bestimmte Probleme

Die Implementierung der Tangenten-Stifigkeitsmatrix \mathbf{K}_T ermöglicht die quadratische Konvergenz des Newton-Raphson-Verfahrens. Jedoch kann es bei großen Lastinkrementen im Newton-Raphson-Verfahren zu negativen Einträgen in der Tangenten-Stifigkeitsmatrix \mathbf{K}_T kommen, was zu einem Problem bei der Lösung des Gesamtgleichungssystems führt. Die Ursache hierfür liegt darin, dass die Summe der Komponenten aus σ^i und der negativen Initialspannung σ_{n+1}^0 in Gleichung (3.87) negative Werte annehmen kann. Um große Lastinkremente und Robustheit des Newton-Raphson-Verfahrens zu ermöglichen, wird in DÜSTER [23] empfohlen, den geometrischen Teil der Tangenten-Stifigkeitsmatrix erst ab dem zweiten Iterationsschritt eines jeden Lastinkrements n zu berechnen. Durch diese Vorgehensweise wird

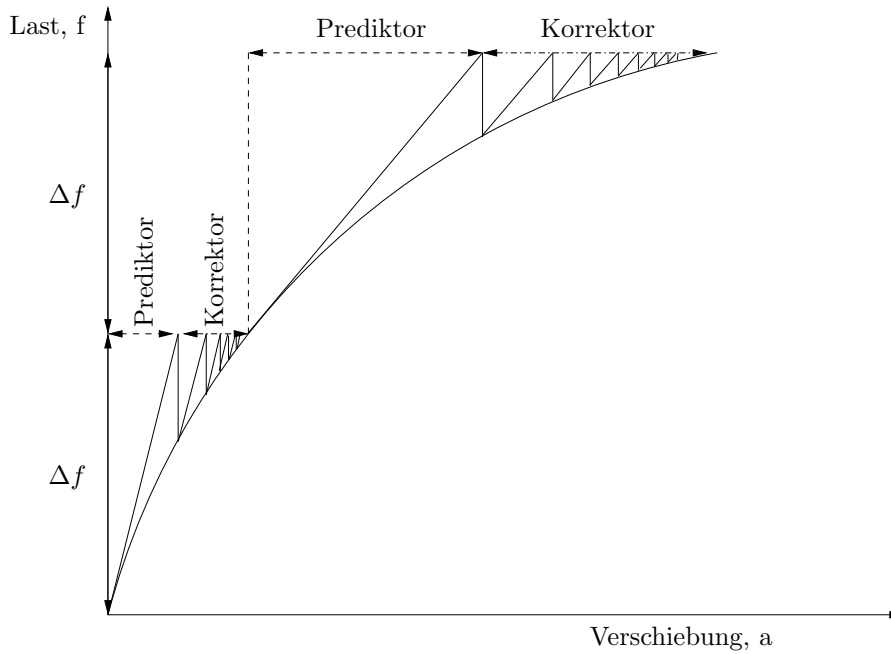


Abbildung 3.8: Kombination inkrementeller Prediktor-Schritte mit modifizierten Newton-Raphson Iterationen

die Stabilität erhöht und eine sehr große Lastschrittweite ermöglicht. Numerische Untersuchungen haben gezeigt, dass die geometrisch nichtlineare Berechnung der elastischen Rückfederung mit einer modifizierten Tangentensteifigkeit

$$\mathbf{K}_T^i = \int_{\Omega} \mathbf{B}^{iT} \mathbf{C} \mathbf{B}^i d\Omega \quad (3.100)$$

bei der die geometrische Steifigkeit vernachlässigt wird, ein sehr effizientes Verfahren darstellt. Da ein Großteil der Rechenzeit bei Berechnungen mit Finiten Elementen hoher Ordnung für die Integration der Elementmatrizen verwendet wird, kann durch die Vernachlässigung von \mathbf{K}_σ die Anzahl der Matrix-Multiplikationen signifikant reduziert werden.

Ausführliche Beispiele zur Konvergenz bei geometrisch nichtlinearen Berechnungen sowie der Reduktion der Iterationen bei Verwendung der modifizierten Steifigkeitsmatrix werden in den Kapiteln 6.1.1, 6.2.4 und 6.4.2 vorgestellt.

Kapitel 4

Grundlagen der geometrischen Modellierung

Die Erzeugung von Beschreibungen geometrischer Modelle sowie deren Verarbeitung hat bereits Generationen von Wissenschaftlern beschäftigt. Insbesondere die computergestützte Modellierung geometrischer Körper hat in den letzten zwanzig Jahren einen bedeutenden Aufschwung erlebt. Dieses Kapitel möchte die für das Verständnis von der in Kapitel 5 dargestellten Konvertierung notwendigen geometrischen Grundlagen vermitteln und darüber hinaus einen Überblick über den aktuellen Stand der geometrischen Modellierung geben. Der in dieser Arbeit verwendete Geometriemodellierer basiert auf dem *Boundary-Representation-Modell*, kurz *B-Rep* genannt [10]. Dabei ist das B-Rep nicht die einzige Möglichkeit geometrische Modelle zu beschreiben. Tabelle 4.1 zeigt eine kurze Übersicht für unterschiedliche Modelle.

4.1 Topologie des Boundary-Representation-Modells

Die topologische Struktur eines Körpers wird beim B-Rep-Modell durch die Objekte Punkt, Kante, Fläche und ihre gegenseitigen Beziehungen beschrieben. Die Topologie ist dabei grundlegender als die Geometrie, welche die Informationen über Lage der Punkte, Kurvenbeschreibung der Kante und Oberflächenbeschreibung der Flächen beinhaltet, da zwei Körper die gleiche Topologie, aber unterschiedliche Geometrie besitzen können (siehe Abbildung 4.1). Mit Ausnahme der Punkte wird durch die klassische analytische Darstellung von Kurven bzw.

Modell	Beschreibung
Wire frame	Knoten und Kanten, keine Flächen bzw. Oberflächen
Set-theoretic	nur Oberflächen (d. h. Halbräume), keine Knoten und Kanten, implizite Randbeschreibung
B-Rep	Knoten, Kanten, Flächen, explizite Randbeschreibung
Polyeder	Nur ebene Geometrie, triangulierte Flächen
Oktalbaum	Diskrete Darstellung durch endliche Anzahl an Oktanten
Oberfläche	keine Volumenmodellierung möglich
R-Rep	Durch Funktion definierte Punktmenge

Tabelle 4.1: Modelle zur Beschreibung geometrischer Objekte [15]

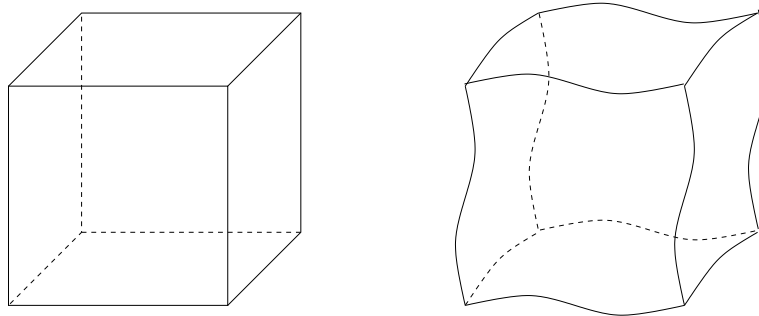


Abbildung 4.1: Zwei topologisch gleiche Körper mit unterschiedlicher Geometrie

Oberflächen eine unberandete Geometrie beschrieben. Die Grundidee des B-Rep-Modells besteht darin, die Berandung der geometrischen Grundobjekte eines Körpers durch hierarchisch niedrigere Objekte *explizit* zu definieren:

- Eine Kante wird durch zwei Punkte, welche auf der Kurve der Kante liegen, berandet.
- Eine Fläche wird durch einen geschlossenen Kantenzug, welcher auf der Oberfläche der Fläche liegt, berandet.

Durch diese Festlegung wird die Geometrie einer Kante bzw. Fläche durch eine Untermenge ihrer Kurven- bzw. Oberflächenbeschreibung definiert. Die Topologie ist somit durch Nachbarschaftsbeziehungen definiert. Bei diesen Nachbarschaftsbeziehungen teilen sich Objekte gemeinsame Berandungsobjekte.

4.1.1 Der *vef*-Graph

Die topologische Struktur wird in [10] mithilfe des *vef*-Graphen $G = (V, E, F; R)$ beschrieben. Dabei definiert $V := v_1, \dots, v_{n_V}$ die Menge der Eckpunkte, $E := e_1, \dots, e_{n_E}$ die Menge der Kanten und $F := f_1, \dots, f_{n_F}$ die Menge der Flächen. Die Knoten in G werden durch Elemente aus V, E und F , die Kanten durch die Adjazenzrelationen R festgelegt. Dabei ist es unnötig, alle möglichen Relationen explizit zu speichern, da sich dabei hochgradige Redundanzen ergeben würde. Bestimmte Relationen lassen sich durch andere ausdrücken und somit den Speicherplatzverbrauch reduzieren. Es stellt sich daher die Frage, welche Relationen zu speichern sind und welche Relationen schnell und günstig aus den gespeicherten errechnet werden können. Das Ziel ist dabei, ein ausgewogenes Verhältnis zwischen Speicherplatzverbrauch und Rechenzeit zu finden.

4.1.2 Euler-Operatoren

Nicht jeder beliebige *vef*-Graph kann einen starren Körper beschreiben. Es müssen daher Kriterien für den Graphen gefunden werden, damit dieser einem starren Körper entspricht. Die *Formel von Euler* (1752) bildet ein solches Kriterium für konvexe Polyeder. Danach muss jeder *vef*-Graph die Gleichung

$$n_V - n_E + n_F = 2 \tag{4.1}$$

erfüllen. H. Poincaré hat 1893 die Formel von Euler für Körper mit Löchern und Hohlräumen verallgemeinert:

$$n_V - n_E + n_F = 2(n_S - n_H) + n_R \quad (4.2)$$

wobei

n_S die Anzahl der Zusammenhangskomponenten der Oberfläche,

n_H die Anzahl der Löcher durch den Körper und

n_R die Anzahl der Löcher durch Flächen bezeichnet.

Eine linear unabhängige Basis von Operatoren stellt dabei sicher, dass das Ergebnis einer Modifikation stets die Formel von Euler erfüllt. Über diese *Euler-Operatoren* kann der Benutzer Modelloperationen durchführen, ohne dabei direkt auf den *vef*-Graph zuzugreifen. Eine umfassende Darstellung von Euler-Operatoren ist in [50] gegeben.

4.1.3 Graphenstruktur in einem objektorientierten Geometriemodellierer

Im *vef*-Graph wird der Zusammenhang der räumlichen Objekte von den Objekten selbst getrennt gehalten. Dadurch erfolgt eine Trennung von Topologie und Geometrie. Bei modernen Geometriemodellierern wird der Zusammenhang zwischen Topologie und Geometrie durch Membervariablen und Zugriffsfunktionen des jeweiligen topologischen Objekts sichergestellt. Obwohl theoretisch die drei topologischen Objekttypen Knoten, Kanten und Flächen hinreichend zur Beschreibung von Volumenkörpern sind, verwenden Geometriemodellierer eine Reihe anderer Objekte, welche den Zugriff erleichtern, die Geschwindigkeit von Modelloperationen erhöhen und die Erzeugung nicht-mannigfaltiger Objekte erlauben. Am Beispiel des Geometriemodellierers *ACIS* werden diese im folgenden vorgestellt. Die dort verwendete Datenstruktur (siehe Abbildung 4.2) hat einen hierarchischen Aufbau.

Als Wurzelement und damit als in der Hierarchie am höchsten stehendes Objekt gibt es den *Body*. Ein *Body* stellt dabei ein abstraktes Objekt dar, welches nicht notwendigerweise einen Körper im mathematischen Sinne beschreibt. Jeder *Body* besitzt mindestens einen *Lump*.

Ein *Lump* kann verstanden werden als ein ein-, zwei-, oder dreidimensionaler Körper welcher disjunkt zu allen anderen *Lumps* des gemeinsamen *Body* ist. Ein *Lump* stellt somit einen abgeschlossenen Körper als Teil des *Body* dar. Jeder *Lump* hat wiederum mindestens eine *Shell*.

Eine *Shell* stellt eine Flächenberandung des *Lump* dar. Dabei kann ein *Lump* mehrere Flächenberandungen besitzen, z. B. innere und äußere Oberflächen-Berandungen bei Hohlkörpern. Ist der Bereich innerhalb der von den *Faces* berandeten *Shell* ein Körper, beschreibt sie eine *peripheral Shell*, umschließt die *Shell* einen Hohlraum, beschreibt sie eine *void Shell*. *Shells* wiederum zeigen auf eine endliche Anzahl von verbundenen *Faces* oder *Wires* welche über nicht-mannigfaltige Knoten verbunden sind. *Faces* sind über gemeinsame Knoten oder Kanten miteinander verbunden, *Wires* können an Endknoten mit *Faces* verbunden sein. Falls eine oder mehrere *Edges* nur mit einer *Face* verbunden sind, werden sie als *freie Edges* und die dazugehörige *Shell* als *offen* bezeichnet. Treffen sich mehr als zwei Flächen an einer Kante, handelt es sich um eine *nicht-mannigfaltige* Kante. Sind Objekte in einem *Body* nicht-mannigfaltig,

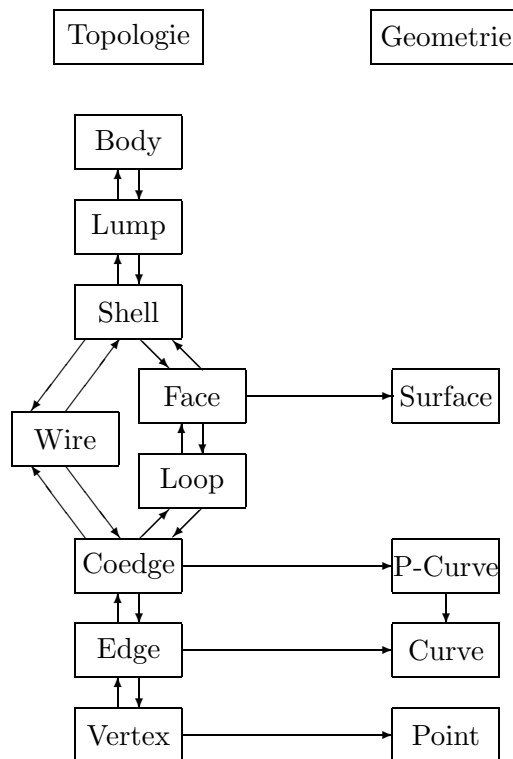


Abbildung 4.2: Vereinfachte topologische und geometrische Hierarchie des Geometriemodellierers ACIS [15]

so sind auch die hierarchisch höheren Objekte nicht-mannigfaltig.

Eine *Face* ist nur auf einer Teilmenge einer *Surface* definiert. *Surfaces* definieren die geometrische Beschreibung der *Face*. Im Sinne des B-Rep-Modells wird eine *Face* durch eine oder mehrere *Loops* berandet. Jede *Loop* legt dabei fest, welcher Teil der *Surface* im Inneren und welcher im Äußeren der *Face* liegt. Eine *Face* kann *ein-* oder *zweiseitig* sein. Eine einseitige Fläche beschreibt einen Halbraum und kann somit teilweise oder vollständig einen Volumenkörper darstellen. Zweiseitige *Faces* werden weiter als *internal* oder *external* unterschieden. Zeigen die Normalenvektoren auf jeder Seite nach außen beschreibt die *Face* eine unendlich dünne zweidimensionale Struktur, zeigen sie nach innen handelt es um eine innere Teilungsfläche eines Volumenkörpers. Da sich mehrere *Faces* eine geometrische Beschreibung (*Surface*) teilen können, müssen Normalenvektor von *Face* und *Surface* nicht notwendigerweise übereinstimmen (siehe Abbildung 4.3).

Eine *Loop* beschreibt die Berandung einer *Face*. Sie besteht aus einer endlichen Anzahl von *Coedges* welche durch eine doppelt verkettete Liste, die geschlossen oder offen sein kann, verknüpft sind. Ist die *Loop* offen, ist sie unvollständig.

Ein *Wire* ist eine verbundene Anzahl von *Edges*, welche nicht zu *Faces* gehören und kein Volumen beranden. *Wires* können abstrakte Objekte wie Profile, Konstruktionslinien oder Schnittlinien beschreiben.

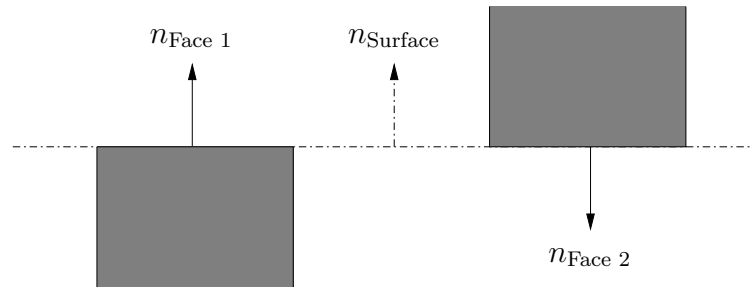


Abbildung 4.3: Orientierung von Faces mit gemeinsamer Surface

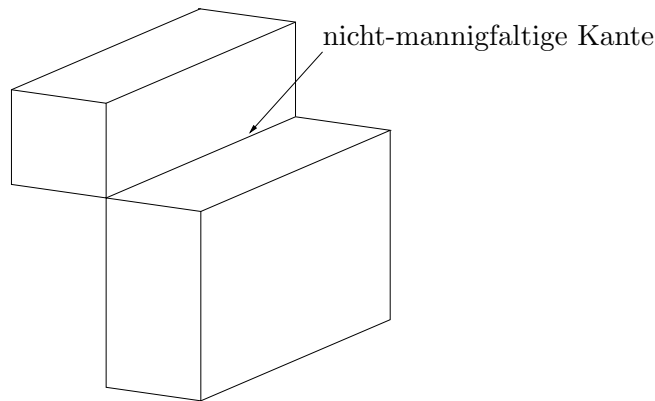


Abbildung 4.4: Körper mit nicht-mannigfaltiger Kante

Eine *Coedge* beschreibt eine Edge bezüglich der zur Coedge gehörenden Face. Der Grund für die Differenzierung in Coedges und Edges hat seine Wurzeln in mathematischen Theoremen, welchen das B-Rep-Modell unterliegt und welche Bedingungen für die Gültigkeit und Integrität der Modelle definieren. Das *Gesetz von Möbius* besagt, dass Kanten, welche zu zwei benachbarten Flächen gehören eine gegenläufige Orientierung besitzen müssen. Damit haben Edges nicht nur die Funktion der Berandung einer Face, sondern auch eine topologische Funktion der Nachbarschaftsassoziaton benachbarter Faces. Um diese topologische Struktur zu erfassen gibt es z. B. die winged-edge Datenstruktur [10] bzw. die split-edge Datenstruktur [15]. Um die Anforderungen an Orientierung und Adjazenz bezüglich mannigfaltiger als auch nicht-mannigfaltiger Objekte zu erfüllen, verwendet ACIS eine abgeleitete Variante dieser beiden Strukturen. Das Hauptziel der Coedge/Edge-Struktur ist die Trennung der Orientierung einer Kante (Coedge) und ihrer topologischen Aufgabe der Nachbarschaftsbeziehung (Edge). Diese Trennung ermöglicht die Modellierung mannigfaltiger Objekte, welche mehr als zwei sich eine Kante teilende Flächen besitzen (siehe Abbildung 4.4). Coedges sind bezüglich ihrer Loop rechtsdrehend orientiert und besitzen einen Zeiger auf die Partner-Coedge der benachbarten Face. Im Falle nicht-mannigfaltiger Objekte werden die Partner-Coedges in einer geschlossenen Liste gehalten. Dabei ist die Reihenfolge der Listenobjekte wichtig, da diese die radiale Ordnung der benachbarten Faces im Gegenuhrzeigersinn beschreibt (siehe Abbildung 4.5). Coedges werden in der Literatur auch als Half-Edges bezeichnet [50, 71].

Edges beinhalten die topologischen und geometrischen Informationen einer Kante. Topologisch wird eine Edge durch zwei Knoten (Vertices) berandet und ist dadurch auf einer Teilmenge

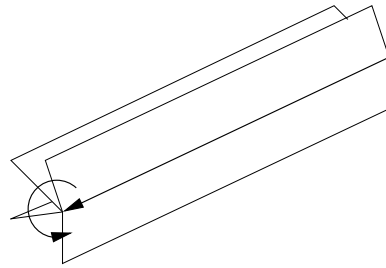


Abbildung 4.5: Radiale Ordnung benachbarter Faces einer nicht-mannigfaltigen Edge

der sie beschreibenden geometrischen Curve definiert. Die Orientierung einer Edge kann in Richtung oder Gegenrichtung der Curve definiert sein. Falls Start- und Endknoten identisch sind ist die Edge geschlossen.

Vertices sind die niedrigste Gruppe in der hierarchischen Struktur und besitzen die geometrische Information ihrer Koordinaten.

4.2 Geometrische Attributierung

Im vorigen Abschnitt erfolgte die Beschreibung der topologischen Struktur eines Körpers über das Boundary-Representation-Modell. Nun folgt die geometrische Attributierung der einzelnen topologischen Objekte. Dabei besitzen nur die Objekte Vertex, Edge und Face eine geometrische Information. Der Geometriemodellierer ACIS z. B. verwendet drei verschiedene Möglichkeiten der geometrischen Attributierung:

1. Explizit:

$$x = f(y, z), y = f(x, z), z = f(x, y) \quad (4.3)$$

2. Implizit:

$$f(x, y, z) = 0 \quad (4.4)$$

3. Parametrisch:

$$\begin{aligned} x = f_x(t), y = f_y(t), z = f_z(t) & \quad \text{für Kurven} \\ x = f_x(u, v), y = f_y(u, v), z = f_z(u, v) & \quad \text{für Flächen} \end{aligned} \quad (4.5)$$

Analytische Kurven und Oberflächen haben jeweils eine implizite und eine parametrische Beschreibung ihrer Geometrie, während z. B. Freiformkurven bzw. -flächen nur eine parametrische Beschreibung besitzen. Bei Operationen zwischen Objekten welche jeweils sowohl durch eine implizite als auch eine parametrische Beschreibung dargestellt werden können, verwendet ACIS Algorithmen für analytische Objekte. Aus Gleichung (4.5) ist die Abbildung der Parameterbereiche von Kurven und Flächen in den euklidischen Raum \mathbb{R}^3 ersichtlich. Für bestimmte

Operationen, wie z. B. Verschneidungen ist es erforderlich, Kurven in der Parametrisierung der Face zu beschreiben. Bi-parametrische *P-Curves*

$$c(t) \rightarrow f(u, v) \quad (4.6)$$

beschreiben eine solche Abbildung. Da Edges und damit auch ihre geometrische Beschreibung zu mehreren Faces mit unterschiedlicher Parametrisierung gehören können, werden P-Curves mit den jeweiligen Coedges assoziiert.

4.2.1 Freiformkurven

Liegt die Kurvenbeschreibung nicht in analytischer Form vor, gibt es verschiedene Verfahren eine gekrümmte Kurve durch eine endliche Anzahl an Punkten zu beschreiben. Es wird dabei in *Interpolations-* und *Approximationsverfahren* unterschieden.

4.2.1.1 Bézierkurven

Dreidimensionale Bézierkurven sind definiert durch die parametrisch eindimensionale Form

$$\mathbf{X}(t) := \sum_{i=0}^n \mathbf{b}_i \cdot B_i^n(t) \quad t \in [0, 1] \quad \mathbf{b}_i \in \mathbb{R}^3. \quad (4.7)$$

wobei die n geometrischen Koeffizienten $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$ als *Bézierpunkte* bezeichnet werden und mit ihrer konvexen Hülle das Kontrollpolygon der Kurve aufspannen. Die Ansatzfunktionen $B_i^n(t)$ stellen dabei die *Bernsteinpolynome* dar. Sie sind definiert als

$$B_i^n(t) := \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} \quad t \in [0, 1], i = 0, \dots, n \quad (4.8)$$

mit der Eigenschaft

$$\sum_{i=0}^n B_i^n(t) = 1 \quad \forall t \in [0, 1]. \quad (4.9)$$

Das Bernsteinpolynom aus Gleichung (4.8) lässt sich auch rekursiv ermitteln. Diese elegante Darstellungsweise ermöglicht eine numerisch günstige Berechnung:

$$B_i^n(t) = (1-t)B_i^{n-1}(t) + tB_{i-1}^{n-1}(t) \quad (4.10)$$

Die Kurve $\mathbf{X}(t)$ besitzt unter anderem folgende Eigenschaften:

1. Bei n Bézierpunkten ergibt sich der Polynomgrad der Kurve zu $p = n - 1$, wodurch ein hoher numerischer Aufwand bei einer großen Anzahl von Stützstellen innerhalb einer Bézierkurve erforderlich ist.
2. Im Gegensatz zu den inneren Stützstellen liegen die Start- und Endpunkte $\mathbf{X}(0) = \mathbf{b}_0$ bzw. $\mathbf{X}(1) = \mathbf{b}_1$ auf der Kurve.
3. $\dot{\mathbf{X}}(0)$ verläuft in Richtung \mathbf{b}_1 , $\dot{\mathbf{X}}(1)$ in Richtung \mathbf{b}_{n-1} .

4. Da $B_i^n(t) \neq 0 \forall t \in [0, 1]$ ist keine lokale Kontrollierbarkeit der Kurvenform möglich und eine Veränderung der Koordinaten eines Kontrollpunktes verändert die Geometrie der gesamten Kurve.

Im folgenden wird nur noch auf die in dieser Arbeit verwendeten rationalen bzw. nicht-rationalen B-Splines eingegangen. Für eine umfassende Beschreibung der Bézierkurven wird auf [24] verwiesen.

4.2.1.2 B-Spline-Kurven

Die in Kapitel 4.2.1.1 erläuterten Bézier-Kurven verhalten sich bei der Beschreibung komplexer Geometrien ineffizient, da aufgrund ihrer globalen Definition der Polynomgrad mit der Anzahl an Stützstellen linear anwächst. B-Spline-Kurven können als eine stückweise, an den Rändern mit einer vom Polynomgrad abhängigen Stetigkeit, verknüpfte Anzahl von „Bézier-artigen“ Kurven angesehen werden. Aufgrund der stückweisen Definition lässt sich bei Variation der Kontrollpunkte der Einfluss auf das jeweilige Kurvenstück beschränken. Durch diese Beschreibung wird ferner sichergestellt, dass der Polynomgrad und somit die Ordnung der Kurve unabhängig von der Anzahl der Stützstellen definiert werden kann. Abbildung 4.6 zeigt das Kontrollpolygon einer B-Spline-Kurve. Wie im Falle der Bézierkurve liegt die B-Spline-Kurve innerhalb der konvexen Hülle ihres Kontrollpolygons.

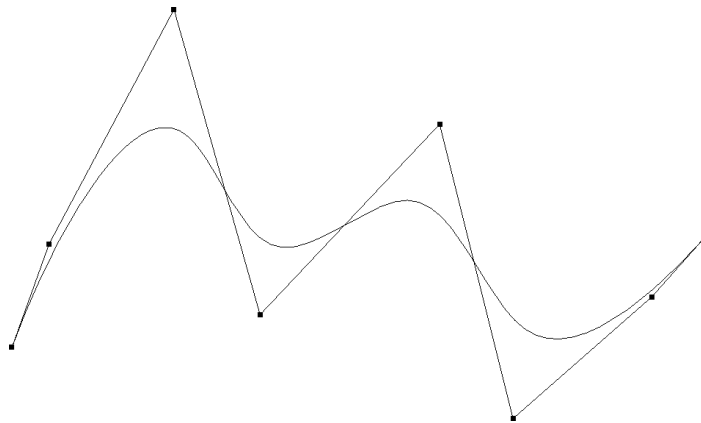


Abbildung 4.6: B-Spline-Kurve mit acht-knotigem Kontrollpolygon

Stetigkeit

Die Glattheit, mit der zwei Kurven miteinander verbunden werden, wird durch die *Stetigkeit* beschrieben. Dabei wird unterschieden zwischen *geometrischer* (G) und *parametrischer* (C) Stetigkeit. Verschiedene Ordnungen der Stetigkeit werden durch die Funktionsräume G^p bzw. C^p der stetigen, p -mal differenzierbaren Funktionen repräsentiert. Die physikalische Bedeutung der geometrischen Stetigkeit ist für $p = 1, 2, 3$ in Abbildung 4.7 illustriert.

- G^0 Kurven haben gleichen Start- bzw. Endpunkt $(x_0, y_0) = (x_1, y_1)$.
- G^1 Tangentenvektoren an Start- bzw. Endpunkt der Kurven haben die gleiche Richtung.
- G^2 Kurven haben gleiche Krümmung an Start- bzw. Endpunkt.

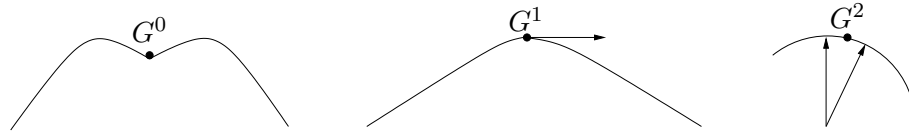
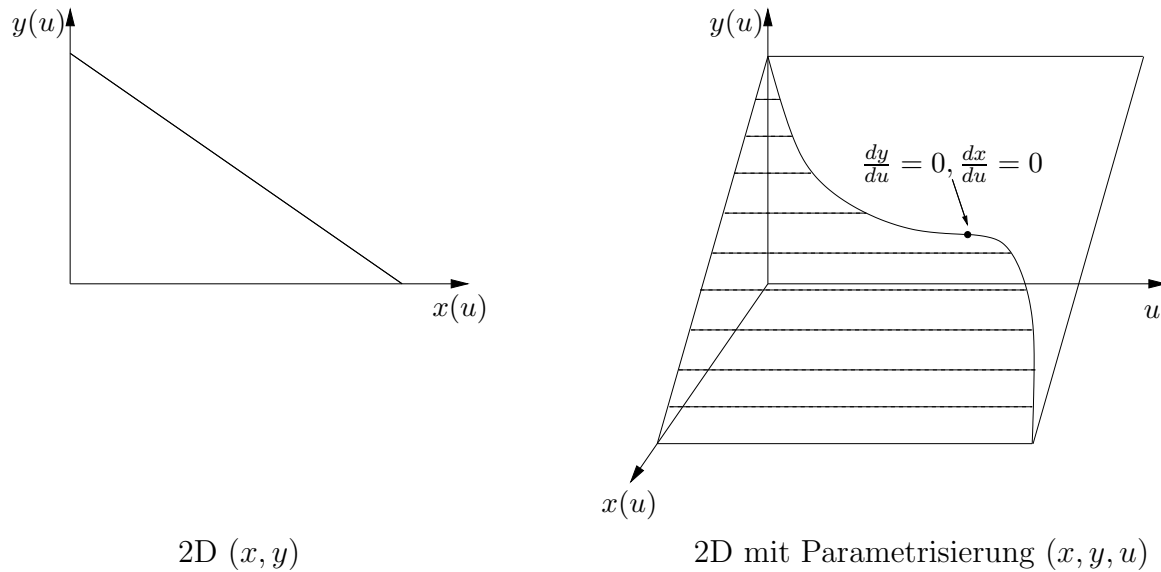


Abbildung 4.7: Geometrische Stetigkeit

Abbildung 4.8: Gerade in (x,y) -Ebene (links) mit zugehöriger nicht-äquidistanter Parametrisierung (rechts) [15]

Im Gegensatz zur geometrischen Stetigkeit wird die parametrisische Stetigkeit durch ihre Stetigkeit $\frac{d^n x}{du^n}$ bezüglich des Parameters u definiert. Daher ist die C^n -Stetigkeit durch die alleinige Darstellung der Geometrie nicht ersichtlich (siehe Abbildung 4.8).

- C^0 Kurven haben gleichen Start- bzw. Endpunkt für gegebenen Parameter u .
 $(x_0(u), y_0(u)) = (x_1(u), y_1(u))$
- C^1 Tangentenvektoren an Start- bzw. Endpunkt der Kurven haben die gleiche Richtung bezüglich u . $(x'_0(u), y'_0(u)) = (x'_1(u), y'_1(u))$
- C^2 Kurven haben gleiche Krümmung an Start- bzw. Endpunkt bezüglich u .
 $(x''_0(u), y''_0(u)) = (x''_1(u), y''_1(u))$

Die Tatsache, dass die parametrisische Stetigkeit visuell nicht ersichtlich ist, legt die Vermutung nahe, dass für die praktische Anwendung von Freiformkurven und -flächen nur die geometrische Stetigkeit von Interesse ist. Singularitäten in der Parametrisierung können jedoch weitreichende Auswirkungen auf Modelloperationen haben, wie später in Kapitel 4.6.2 gezeigt werden wird.

In der Konstruktion von Umformwerkzeugen ist die Stetigkeit ein sehr wichtiger Parameter. Unstetigkeiten in der Krümmung der Werkzeugflächen können zu sehr hohen Spannungen an der Unstetigkeitsstelle während des Umformvorgangs und damit zu einem erhöhten Verschleiß des Werkzeugs führen. Daher wird eine G^2 -Stetigkeit der Werkzeugfläche angestrebt.

Generell weisen stückweise definierte B-Spline-Kurven vom Grad p an den Übergangsstellen eine C^{p-1} -Stetigkeit auf. Weiterhin besitzen B-Spline-Kurven folgende Eigenschaften:

1. Für den Polynomgrad p eines Kurvensegments gilt $p = n - 1$, wobei n die Anzahl der Kontrollpunkte ist.
2. Benachbarte Segmente haben gemeinsame Kontrollpunkte. Die Anzahl der gemeinsamen Kontrollpunkte hängt dabei vom Polynomgrad und der geforderten Stetigkeit zwischen zwei Segmentübergängen ab.
3. Die Segmente interpolieren jeweils nur Start- und Endpunkt des Kontrollpolygons.

Anders als bei Bézierkurven, welche vollständig durch einen Satz an Kontrollpunkten beschrieben werden, erfordern B-Spline-Kurven sowohl Polynomgrad als auch die Parameterwerte des Segmentstart- und Segmentendpunkt um die endgültige Form der Kurve bestimmen zu können.

Knotenvektor

Die Start- und Endpunkte jedes stückweise definierten Polynoms werden als *Knotenvektor-Komponenten* bezeichnet und über den Parameter t spezifiziert. Die Menge an Knoten einer Kurve bildet den *Knotenvektor* \mathbf{v} . Die Länge des Knotenvektors ist dabei abhängig von der Anzahl der Kontrollpunkte sowie dem Polynomgrad der Kurve. Die Werte im Knotenvektor müssen aufsteigend sein, dürfen sich aber wiederholen. Die Anzahl der Wiederholungen einer Knotenvektor-Komponente gibt dessen *Vielfachheit* an. Wiederholte Knoten reduzieren die Kontinuität der Kurve, wobei sich die Differenzierbarkeit jeweils um Eins erniedrigt. Während sich an den Start- und Endpunkten dieses Verhalten anbietet um sicherzustellen, dass die resultierende Kurve den ersten und letzten Kontrollpunkt interpoliert, sind wiederholte Einträge im mittleren Knotenvektor aufgrund der resultierenden Diskontinuitäten unerwünscht. Der Knotenvektor

$$\mathbf{v} = \left\{ \underbrace{0, 0, 0}_{\text{Vielfachh. 3}}, 0.2, 0.4, 0.6, 0.8, \underbrace{1.0, 1.0, 1.0}_{\text{Vielfachh. 3}} \right\}$$

stellt z. B. für eine B-Spline-Kurve vom Polynomgrad $p = 3$ sicher, dass Start- und End-Kontrollpunkte der Kurve interpoliert werden. Dieser Knotenvektor wird auch als *uniformer Knotenvektor* bezeichnet, da die Komponenten bis auf die wiederholten Einträge einen äquidistanten Abstand im Parameterbereich aufweisen. Im Gegensatz dazu wird ein Vektor mit nicht-äquidistanten Einträgen als *nicht-uniformer Knotenvektor* bezeichnet.

B-Spline-Funktion

B-Spline-Kurven haben einen ähnlichen Aufbau wie Bézierkurven. Sie werden als Summe der Produkte der Kontrollpunkte mit der zugehörigen Basisfunktion als

$$\mathbf{X}(t) := \sum_{i=0}^n \mathbf{p}_i \cdot N_i^p(t) \tag{4.11}$$

beschrieben, wobei

- n die Anzahl der Kontrollpunkte,
- p den Polynomgrad der Kurve,
- N_i^p die Basisfunktion vom Grad p für den Punkt i ,
- t den Parameter zwischen min. und max. Knotenvektor-Komponente, und
- \mathbf{p}_i den Kontrollpunkt i darstellt.

Ähnlich wie bei Bézierkurven kann die Basisfunktion rekursiv ermittelt werden:

$$N_i^p(t) = \frac{t - t_i}{t_{i+p} - t_i} N_i^{p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} N_{i+1}^{p-1}(t) \quad p \geq 1 \quad (4.12)$$

mit

$$N_i^0(t) = \begin{cases} 1 & t_i \leq t \leq t_{i+1} \\ 0 & \text{sonst} \end{cases} \quad (4.13)$$

wobei t_i der Wert des t -Parameters der Knotenvektor-Komponente i ist. Da der Nenner in Gleichung (4.12) Null werden kann wird hierfür $\frac{0}{0} := 0$ definiert. Es ist ersichtlich, dass die stückweise definierten B-Spline-Polynome $p + 2$ Parameter t_i benötigen, da neben den von p abhängigen Parametern t_{i+p} und t_{i+p+1} ferner die von p unabhängigen Parameter t_i und t_{i+1} benötigt werden. Weiterhin folgt aus Gleichung (4.12), dass die Basisfunktion N_i^p aus $p + 1$ Basisfunktionen $N_i^0, N_{i+1}^0, \dots, N_{i+p}^0$ konstruiert wird und somit nur lokal auf diesen $p + 1$ Teilintervallen von Null verschiedene Werte annimmt. Aus diesem Sachverhalt folgt, dass die Veränderung eines Kontrollpunktes auch nur eine lokale Veränderung des Kurvenverlaufs nach sich zieht. Dieser Einfluss ist umso höher, je höher der Polynomgrad p der B-Spline-Kurve ist. Für die Anzahl n_s der stückweise definierten Polynome gilt

$$n_s = n - p - 1. \quad (4.14)$$

NURBS

Non-Uniform-Rational-B-Splines stellen eine Erweiterung der B-Spline-Formulierung dar. *Rationale* B-Splines sind durch

$$\mathbf{X}(t) := \frac{\sum_{i=0}^n N_i^p(t) w_i \mathbf{p}_i}{\sum_{i=0}^n N_i^p(t) w_i} \quad (4.15)$$

definiert, wobei

- n die Anzahl der Kontrollpunkte,
- \mathbf{p}_i den Kontrollpunkt i ,
- $N_i^p(t)$ die Basisfunktion vom Grad p für den Punkt i , und
- w_i den üblicherweise positiven Wichtungsfaktor für den Punkt i darstellt.

NURBS sind dabei die allgemeinste Darstellung der bisher genannten Funktionen. Da

$$\sum_{i=0}^n N_i^p(t) = 1 \quad (4.16)$$

ist offensichtlich, dass Gleichung (4.15) für einen Wichtungsfaktor $w_i = 1$ den Ausdruck der ganzrationalen B-Spline-Funktion liefert. Eine NURBS-Kurve ohne innere Knoten und $w_i = 1$ führt ferner zu den in Kapitel 4.2.1.1 erläuterten Beziérkurven. Aus Gleichung (4.15) lassen sich mit

$$R_i^p(t) := \frac{N_i^p(t)w_i}{\sum_{j=0}^n N_j^p(t)w_j} \quad (4.17)$$

die rationalen Basisfunktionen definieren. Ein Vorteil der NURB-Splines gegenüber der nicht-rationalen B-Splines liegt in der erweiterten Darstellbarkeit von Kurven. Bei NURBS kann durch Modifikation des Kontrollpunktes \mathbf{p}_i sowie des Gewichtungsfaktors w_i die Form des Kurvensegments im Intervall $t \in [t_i, t_{i+d+1}]$ lokal kontrolliert werden. Dabei wird der Punkt $\mathbf{X}(t)$ bei Erhöhung von w_i näher am Kontrollpunkt \mathbf{p}_i ausgerichtet. Bei variierendem w_i mit $0 < w_i < \infty$ für ein konstantes t liegen alle $\mathbf{X}(t)$ auf einer Geraden durch den Kontrollpunkt \mathbf{p}_i . NURBS können beispielsweise zur Darstellung konischer Kurven mithilfe des Kontrollpolygons $C = (P_0, P_1, P_2)$, des Knotenvektors $V = \{0, 0, 1, 1\}$ und den Wichtungsfaktoren $w_0 = 1, w_1 = \frac{r}{1-r}, w_2 = 1$ Anwendung finden. Durch Variation von r können vier verschiedene Arten von Kurven dargestellt werden:

$$\begin{aligned} r > \frac{1}{2} & \text{ hyperbolische Kurve} \\ r = \frac{1}{2} & \text{ parabolische Kurve} \\ r < \frac{1}{2} & \text{ elliptische Kurve} \\ r = 0 & \text{ Gerade} \end{aligned}$$

4.2.2 Freiformflächen

Ausgehend von den verschiedenen Ansätzen zur Modellierung von Freiformkurven kann nun zur Modellierung von Freiformflächen übergegangen werden. Neben der impliziten und in seltenen Fällen der expliziten Darstellung ist die Parameterdarstellung $\mathbf{X}(u, v) = \{f_x(u, v), y = f_y(u, v), z = f_z(u, v)\}$ die am häufigsten eingesetzte Form zur Flächenbeschreibung. Dabei wird die zweidimensionale Parameterfläche in einen dreidimensionalen Raum abgebildet. Die beschreibenden Polynome können durch das *Tensorprodukt* von Freiformkurven konstruiert werden. Die Basis-Funktionen sind dabei zweidimensionale Funktionen der Parameter u und v , welche durch das Produkt von eindimensionalen Basis-Funktionen bestimmt werden. Dabei werden die geometrischen Kontrollpunkte $\mathbf{p}_{i,j}$, $i = 0, \dots, n$, $j = 0, \dots, m$ in einem topologisch rechteckigen Gitter der Größe $(n + 1) \times (m + 1)$ angeordnet. Für B-Spline-Flächen führt das Tensorprodukt zu der Form

$$\mathbf{X}(u, v) := \sum_{i=0}^n \sum_{j=0}^m \mathbf{p}_{i,j} N_i^p(u) N_j^q(v) \quad (4.18)$$

wobei

$p_{i,j}$ ein rechteckiges Feld von Kontrollpunkten der Größe $(n + 1) \times (m + 1)$,

N_i^p die Basisfunktion in Richtung u vom Grad p und

N_j^q die Basisfunktion in Richtung v vom Grad q darstellt.

Gleichung (4.18) kann in Matrix-Notation formuliert werden als

$$\mathbf{X}(u, v) := [N_i^p(u)]^T [\mathbf{p}_{i,j}] [N_j^q(v)]. \quad (4.19)$$

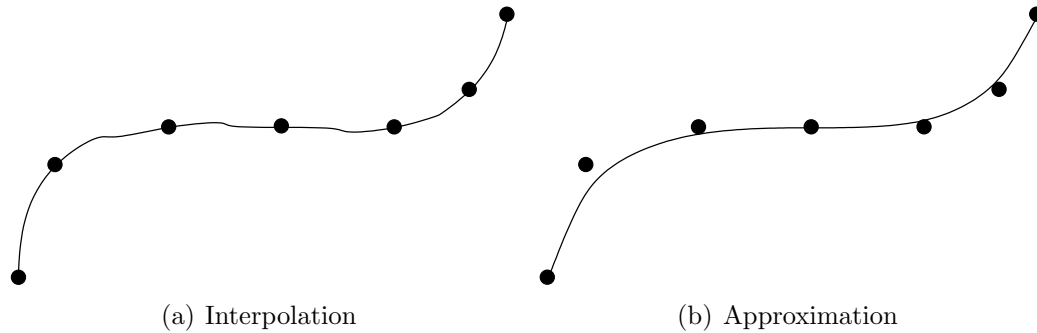


Abbildung 4.9: Kurve durch sieben Punkte (mit Programm Xfig)

4.3 Interpolation und Approximation

In den bisherigen Kapiteln wurden die einzelnen verschiedenen Darstellungsarten von Freiformkurven und -flächen vorgestellt. Anliegen des vorliegenden Kapitels ist die Präsentation von Verfahren zur Abbildung geometrischer Objekte mithilfe dieser mathematischen Beschreibungsformen, im speziellen wird dies hierbei nur für die in dieser Arbeit verwendeten NURBS bzw. B-Splines diskutiert. Die Ermittlung der geometrischen Parameter für die kontinuierliche Beschreibung von Kurven und Flächen aus diskreten Ausgangsdaten untergliedert sich in *Interpolation* und *Approximation*. Bei der Interpolation werden die diskreten Werte exakt beschrieben, d. h. die Kurve bzw. Fläche verläuft durch diese Punkte. Abbildung 4.9(a) zeigt eine Kurve, welche sieben Punkte und die ersten Ableitungen an Start- und Endpunkt interpoliert. Bei der Approximation verläuft die zu konstruierende Kurve oder Fläche nur näherungsweise mit einer optional zu definierenden Genauigkeit durch die jeweiligen Punkte. Approximierende Verfahren haben im allgemeinen einen weicheren Verlauf. Abbildung 4.9(b) zeigt eine approximierende Kurve durch die sieben Punkte.

4.3.1 Interpolation von Kurven

Gegeben sei eine Menge $\mathbf{q}_k, k = 0, \dots, n$ von Punkten welche durch eine B-Spline-Kurve vom Grad p interpoliert werden sollen. Dazu wird jedem \mathbf{q}_k ein Parameterwert \bar{t}_k zugeordnet und ein entsprechender Knotenvektor $\mathbf{t} = \{t_0, \dots, t_m\}$ festgelegt. Das zur Bestimmung der $n + 1$ unbekanntenen Kontrollpunkte \mathbf{p}_i zu lösende lineare Gleichungssystem ergibt sich somit zu

$$\mathbf{q}_k = \mathbf{X}(\bar{t}_k) = \sum_{i=0}^n N_i^p(\bar{t}_k) \mathbf{p}_i. \quad (4.20)$$

Die Wahl der Parametrisierung $\bar{t}_k \in [0, 1]$ beeinflusst dabei die resultierende Form der Kurve und ist entscheidend für die Qualität der Interpolation. Ein häufig verwendetes Verfahren zur Bestimmung der Parametrisierung ist dabei das *Sehnenlängenverfahren*. Dabei ist s die Sehnenlänge

$$s = \sum_{k=1}^n |\mathbf{q}_k - \mathbf{q}_{k-1}|. \quad (4.21)$$

Die Parametrisierung ermittelt sich zu

$$\bar{t}_k = \bar{t}_{k-1} + \frac{|\mathbf{q}_k - \mathbf{q}_{k-1}|}{s} \quad k = 1, \dots, n-1 \quad \bar{t}_0 = 0 \quad \bar{t}_n = 1. \quad (4.22)$$

Die Knotenvektor-Komponenten werden durch Mittelung der Parameter ermittelt:

$$t_j = \begin{cases} 0 & j = 0, \dots, p \\ \frac{1}{p} \sum_{i=j-p}^{j-1} \bar{t}_i & j = p, \dots, n \\ 1 & j = m-p, \dots, m \end{cases} \quad (4.23)$$

Weitere Methoden zur Parametrisierung werden in [57] diskutiert.

4.3.2 Interpolation von Flächen

Gegeben sei ein topologisch uniformes Gitter $(n+1) \times (m+1)$ von Punkten $\mathbf{q}_{k,l}$, $k = 0, \dots, n$ und $l = 0, \dots, m$, welche durch eine B-Spline-Fläche vom Grad p in u -Richtung und q in v -Richtung interpoliert werden sollen. Analog zum Vorgehen in Kapitel 4.3.1 ergibt sich das lineare Gleichungssystem

$$\mathbf{q}_{k,l} = \mathbf{X}(\bar{u}_k, \bar{v}_l) = \sum_{i=0}^n \sum_{j=0}^m N_i^p(\bar{u}_k) N_j^q(\bar{v}_l) \mathbf{p}_{i,j}. \quad (4.24)$$

Die Parametrisierung \bar{u}_k ergibt sich durch Ermittlung von $\bar{u}_0^l, \dots, \bar{u}_n^l$ nach Gleichung (4.22) und Mittelung über alle \bar{u}_k^l , $l = 0, \dots, m$

$$\bar{u}_k = \frac{1}{m+1} \sum_{l=0}^m \bar{u}_k^l \quad k = 0, \dots, n. \quad (4.25)$$

Die Parametrisierung \bar{v}_l erfolgt analog. Die Knotenvektor-Komponenten können gemäß Gleichung (4.23) ermittelt werden. Die Kontrollpunkte $\mathbf{p}_{i,j}$ können schließlich aus den $(n+1) \times (m+1)$ linearen Gleichungen ermittelt werden. Effizienter ist dabei die Ermittlung für Kurven auf $\mathbf{X}(u, v)$ bei denen eine der Parameterkoordinaten konstant gehalten wird. Gleichung (4.24) ergibt sich für $l = \text{const.}$ zu

$$\mathbf{q}_{k,l} = \sum_{i=0}^n N_i^p(\bar{u}_k) \mathbf{r}_{i,l} \quad \text{mit} \quad \mathbf{r}_{i,l} = \sum_{j=0}^m N_j^q(\bar{v}_l) \mathbf{p}_{i,j} \quad (4.26)$$

wobei $\mathbf{r}_{i,l}$ die Kontrollpunkte darstellen. Die Kurven für $k = \text{const.}$ ergeben sich analog, so dass schließlich das Tensorprodukt zur Beschreibung der Fläche gebildet werden kann.

4.3.3 Approximation

Bei der Approximation wird im Gegensatz zur Interpolation nicht verlangt, dass die kontinuierliche Geometriebeschreibung die diskreten Stützstellen enthält. Statt dessen wird eine gewisse Genauigkeit gefordert, welche durch eine Fehlerschranke E definiert wird. Bei der Interpolation besteht eine Abhängigkeit zwischen der Anzahl der Kontrollpunkte, des Polynomgrads und der Anzahl der Interpolationspunkte. Bei der Approximation ist die zur geforderten Genauigkeit erforderliche Anzahl an Kontrollpunkten a priori nicht bekannt, sondern muss über ein iteratives Verfahren ermittelt werden. Dabei werden zwei Vorgehensweisen unterschieden:

Annäherung von unten. Als Startparameter wird eine minimale Anzahl von Kontrollpunkten festgelegt. Nach Approximation der Kurve wird die Abweichung bestimmt und mit der geforderten Genauigkeit verglichen. Die Anzahl der Kontrollpunkte wird dann solange iterativ erhöht bis der Fehler innerhalb der für die geforderte Genauigkeit definierten Schranke E liegt.

Annäherung von oben. Als Startparameter wird eine maximale Anzahl von Kontrollpunkten festgelegt. Auch hier wird der Fehler bezüglich der geforderten Genauigkeit mit der Fehlerschranke E verglichen. Die Anzahl der Kontrollpunkte wird in einer Iteration solange reduziert bis der Fehler des aktuellen Iterationsschrittes die Fehlerschranke überschreitet.

Approximationen sind numerisch komplexer und aufwändiger, da sie iterative Prozesse darstellen und die Ermittlung der Abweichung numerisch aufwändig ist. Zentraler Bestandteil innerhalb eines Iterationsschrittes der Approximation ist dabei ein nichtlineares Optimierungsproblem bezüglich Kontrollpunkten, Parameter, Knotenvektor und Gewichtungsvektor. Mit Least-Squares-Fit-Verfahren kann die Abweichung der approximierten Geometrie zu Stützstellen minimiert werden.

4.3.4 Approximation von Kurven

Im folgenden werden die Gewichte der zu ermittelnden Geometrie zu eins gesetzt, um nichtlineare Probleme zu vermeiden. Die ganzrationale B-Spline-Kurve

$$\mathbf{X}(t) = \sum_{i=0}^n N_i^p(t) \mathbf{p}_i \quad t \in [0, 1] \quad 1 \leq p \leq n \quad (4.27)$$

approximiert m Stützstellen $\mathbf{q}_0, \dots, \mathbf{q}_m$ ($m > n$), wobei

$$\begin{aligned} \mathbf{q}_0 &= \mathbf{X}(0), \\ \mathbf{q}_m &= \mathbf{X}(1) \end{aligned} \quad (4.28)$$

gilt. Im folgenden definiert f eine Skalarfunktion aus der Summe der Fehlerquadrate für die inneren Stützstellen $\mathbf{q}_1, \dots, \mathbf{q}_{m-1}$

$$f = \sum_{k=1}^{m-1} |\mathbf{q}_k - \mathbf{X}(\bar{t}_k)|^2 = \sum_{k=1}^{m-1} \left| \mathbf{r}_k - \sum_{i=1}^{n-1} N_i^p(\bar{t}_k) \mathbf{p}_i \right|^2 \quad (4.29)$$

wobei

$$\mathbf{r}_k = \mathbf{q}_k - N_0^p(\bar{t}_k) \mathbf{q}_0 - N_n^p(\bar{t}_k) \mathbf{q}_m. \quad (4.30)$$

Beim Least-Squares-Fit-Verfahren soll f bezüglich der Kontrollpunkte \mathbf{p}_l minimiert werden:

$$\frac{\partial f}{\partial \mathbf{p}_l} = \sum_{k=1}^{m-1} \left(-2N_l^p(\bar{t}_k) \mathbf{r}_k + 2N_l^p(\bar{t}_k) \sum_{i=1}^{n-1} N_i^p(\bar{t}_k) \mathbf{p}_i \right) \stackrel{!}{=} 0 \quad (4.31)$$

Aus Gleichung (4.31) ergibt sich das Gleichungssystem

$$(\mathbf{N}^T \mathbf{N}) \mathbf{P} = \mathbf{R} \quad (4.32)$$

zur Ermittlung der unbekanntenen Kontrollpunkte \mathbf{P} . Für die Komponenten von \mathbf{N} müssen zunächst noch der unbekanntene Knotenvektor \mathbf{t} und die Parametrisierung $\bar{\mathbf{t}}$ festgelegt werden. $\bar{\mathbf{t}}$ kann dabei nach Gleichung (4.22) ermittelt werden. Um eine positiv-definite Matrix $\mathbf{N}^T \mathbf{N}$ zu erhalten, werden die Komponenten des Knotenvektors nach [18, 57] ermittelt zu

$$t_{p+j} = (1 - \alpha) \bar{t}_{i-1} + \alpha \bar{t}_i \quad j = 1, \dots, n - p \quad (4.33)$$

wobei

$$i = \text{int} \left(\frac{j(m+1)}{n-p+1} \right), \quad \alpha = \frac{j(m+1)}{n-p+1} - i.$$

Eine ausführliche Beschreibung der Approximation von Kurven wird in [57] gegeben.

4.3.5 Approximation von Flächen

Analog zum Vorgehen in Kapitel 4.3.4 kann eine kontinuierliche Fläche für ein rechteckiges Gitter der Größe $(r+1) \times (s+1)$ an diskreten Stützstellen $\mathbf{q}_{k,l}$, $k = 0, \dots, r$, $l = 0, \dots, s$ approximiert werden. Dabei wird für die k -Richtung der Polynomgrad p , für die l -Richtung der Polynomgrad q definiert. Die durch einen Least-Squares-Fit zu ermittelnden Kontrollpunkte werden als rechteckiges Gitter der Größe $(n+1) \times (m+1)$ definiert, wobei die Größe durch das Least-Squares-Fit-Verfahren iterativ ermittelt wird ($n < s$, $m < s$). Approximierte Flächen werden ähnlich wie dem Vorgehen in Kapitel 4.3.2 zunächst als Kurve in k - und dann in l -Richtung approximiert. Dabei enthält die resultierende Fläche die Stützstellen $\mathbf{q}_{0,0}$, $\mathbf{q}_{r,0}$, $\mathbf{q}_{0,s}$, $\mathbf{q}_{r,s}$ exakt, die inneren Stützstellen ergeben sich durch Lösen eines Gleichungssystems analog Gleichung (4.32).

4.3.6 Approximation mit spezifischer Genauigkeit

Zentrale Größe bei der Approximation ist die maximale Abweichung der Kurve von jeder diskreten Stützstelle \mathbf{q}_k . Da dieser Wert einen großen Einfluss auf den Verlauf der Kurve hat, muss er dem Problem und den Modelldimensionen entsprechend gewählt werden. Handelt es sich bei den Punkten z. B. um Werte aus einer optischen Messaufnahme mit numerischem Rauschen und Messungenauigkeiten, ist die Toleranz höher zu wählen als bei der Approximation durch die Knoten eines Finite-Elemente-Netzes. Wichtig ist außerdem die Relation der Toleranz zur Modellgröße. Berechnungen der elastischen Rückfederung im Automobilbau liefern üblicherweise Verschiebungen im Millimeterbereich. Aus diesem Grund wäre es ungeeignet, die zugrunde liegende Geometrie mit einer ebenfalls im Millimeterbereich liegenden maximalen Abweichung zu approximieren. Üblicherweise liefern Approximationen weniger oszillierende Beschreibungen als Interpolationen. Daher ist es wichtig, für die maximale Abweichungen einen Kompromiss zwischen einer möglichst glatten Approximation und hoher Genauigkeit zu finden. In der Literatur finden sich für die maximale Abweichung E im wesentlichen zwei

Definitionen:

$$\begin{aligned} \text{Kurven} & : \max_{0 \leq k \leq m} |\mathbf{q}_k - \mathbf{X}(\bar{t}_k)| \\ \text{Flächen} & : \max_{0 \leq k \leq r, 0 \leq l \leq s} |\mathbf{q}_{k,l} - \mathbf{X}(\bar{u}_k, \bar{v}_l)| \end{aligned} \quad (4.34)$$

oder

$$\begin{aligned} \text{Kurven} & : \max_{0 \leq k \leq m} \left(\min_{0 \leq t \leq 1} |\mathbf{q}_k - \mathbf{X}(t)| \right) \\ \text{Flächen} & : \max_{0 \leq k \leq r, 0 \leq l \leq s} \left(\min_{0 \leq u \leq 1, 0 \leq v \leq 1} |\mathbf{q}_{k,l} - \mathbf{X}(u, v)| \right) \end{aligned} \quad (4.35)$$

Die in Gleichung (4.35) definierte *Maximumsnorm* ist dabei numerisch aufwändiger als Gleichung (4.34), hat aber eine höhere Aussagekraft, da sie die minimale Abweichung eines Stütz-knotens zur Kurve misst. Außerdem führt sie bei Kurven wegen

$$\min_{0 \leq t \leq 1} |\mathbf{q}_k - \mathbf{X}(t)| \leq |\mathbf{q}_k - \mathbf{X}(\bar{t}_k)| \quad (4.36)$$

bzw. bei Flächen wegen

$$\min_{0 \leq u \leq 1, 0 \leq v \leq 1} |\mathbf{q}_{k,l} - \mathbf{X}(u, v)| \leq |\mathbf{q}_{k,l} - \mathbf{X}(\bar{u}_k, \bar{v}_l)| \quad (4.37)$$

zu Beschreibungen mit einer geringeren Anzahl an Kontrollpunkten. Bei der Iteration mit *Annäherung von unten* (siehe Kapitel 4.3.3) für Kurven wird zunächst eine minimale Anzahl an Kontrollpunkten festgelegt. Es wird nun in jedem Iterationsschritt und für jedes durch den Knotenvektor stückweise definierte Polynom geprüft, ob Gleichung (4.35) erfüllt ist. Falls der Fehler für einen Bereich den Wert überschreitet, wird in der Mitte dieses Bereichs ein neuer Knoten eingefügt, wodurch sich die Anzahl der Kontrollpunkte erhöht. Es ist wichtig, dass der Algorithmus auch mit „leeren“ Knotenbereichen umgehen kann, d. h. wenn Knoten hinzugefügt werden und kein Kontrollpunkt im neuen Bereich liegt. Für Flächen wird in [57] eine effiziente Vorgehensweise beschrieben. Hierbei wird die Reihe l_0 und Spalte k_0 mit der Anzahl an Punkten, deren Abweichung größer als die durch E definierte Fehlerschranke ist, lokalisiert. Die Kurven $\mathbf{X}_{\bar{u}_{k_0}}(v)$ und $\mathbf{X}_{\bar{v}_{l_0}}(u)$ werden solange iterativ approximiert, bis sie innerhalb der Fehlerschranke E liegen. Die daraus ermittelten neuen Knotenvektoren \mathbf{u} und \mathbf{v} können dann für die nächste Flächenapproximation verwendet werden.

Die Entfernung von Knotenvektor-Komponenten ist der zentrale Punkt beim zweiten Iterationstyp, der *Annäherung von oben*. Dabei stellt $\mathbf{X}(t)$ eine Kurve vom Polynomgrad p , mit inneren Knoten \mathbf{t}_r und der Vielfachheit s dar. $\tilde{\mathbf{C}}(t)$ bezeichnet dann die Kurve nach dem Entfernen eines inneren Knoten. Im weiteren stellen \mathbf{q}_k die Stützstellen und $\bar{\mathbf{t}}_k$ die ihnen zugeordneten Parameter dar. \mathbf{e}_k sind die mit \mathbf{q}_k assoziierten Fehler, welche im iterativen Prozess mit der Fehlerschranke E verglichen werden. Dabei werden solange mit dem in [57] erläuterten Verfahren Knoten entfernt, bis jede Stützstelle \mathbf{q}_k zur Kurve $\mathbf{X}(u)$ einen Abstand $\mathbf{e}_k < E$ hat. Das Verfahren lässt sich analog für Flächen $\mathbf{X}(u, v)$ formulieren.

4.4 Boolesche Operationen

Boolesche Operationen im Sinne der geometrischen Modellierung beinhalten die Operationen Vereinigung (\cup), Durchschnitt (\cap) und Differenz (\setminus) geometrischer Objekte und gehören zu den essentiellen Komponenten eines Solid-Modellierers. Objekte können dabei ein-, zwei- und dreidimensional sein. Mithilfe beliebig vieler Operationen lassen sich komplexe Objekte aus einfachen Grundkörpern erzeugen. Definitionsgemäß wird eine boolesche Operation zwischen zwei Objekten in vier Schritten durchgeführt:

1. Verschneidung aller Flächen und Kanten der Volumenkörper und Aufbau eines Verschneidungsgraphen, welcher die Datenstruktur der Verschneidung zwischen den beiden Objekten darstellt. Dies wird im folgenden Abschnitt ausführlich dargestellt.
2. Aufprägen des Verschneidungsgraphen auf beide Ausgangsobjekte. Dabei kommt es im allgemeinen zu Teilungen von Kanten und Flächen.
3. Entscheidung, welche Objekte beider Körper in Abhängigkeit der vorliegenden Operationen zu verwerfen bzw. zu behalten sind.
4. Zusammenfügen bzw. Trennen der Körper entlang von Verschneidungskanten und Aufbau der Datenstruktur des neuen Körpers.

Im folgenden sollen die einzelnen Schritte der booleschen Operationen für zwei mannigfaltige Körper A und B erläutert werden. Dabei ist die Reihenfolge der Objekte zu beachten, da die Operation Subtraktion nicht kommutativ ist.

4.4.1 Aufbau des Verschneidungsgraphen

Bei der Ermittlung des Verschneidungsgraphen werden im ersten Schritt Faces der Objekte A und B gegenseitig verschnitten. Aus Effizienzgründen wird dabei zunächst überprüft, ob sich die zugehörigen Bounding-Boxen überlappen. Das Ergebnis der einzelnen Flächenverschneidungen ist in Abbildung 4.10 als Wire-Body dargestellt, welcher in diesem Zusammenhang als Verschneidungsgraph bezeichnet wird. Jede Edge dieses Graphen ist nicht-mannigfaltig und besitzt vier Coedges, wobei jeweils zwei zu Objekt A und B gehören. Die Geometrie der Edges wird durch die Flächen als Schnittkurve beschrieben, wobei für jede Coedge eine P-Curve des jeweiligen Parameterbereichs der Face erzeugt wird. Im übrigen werden bei der Verschneidung nur neue Knoten und Kanten aber keine neuen Oberflächenbeschreibungen erzeugt.

4.4.2 Verschneidung Fläche-Fläche

Für Paare von Flächen, welche in analytischer Beschreibung vorliegen, können analytische Lösungen verwendet werden (z. B. Verschneidung zwischen Kugel und Kegel). Im allgemeinen Fall wird die Beschreibung jedoch in parametrischer Form vorliegen. Verschneidungen vom Typ Kurve-Fläche und Kurve-Kurve liefern bestimmte bzw. überstimmte Gleichungssysteme. Dort auftretende Schwierigkeiten wie mehrfache Schnittmengen können durch Projektionen

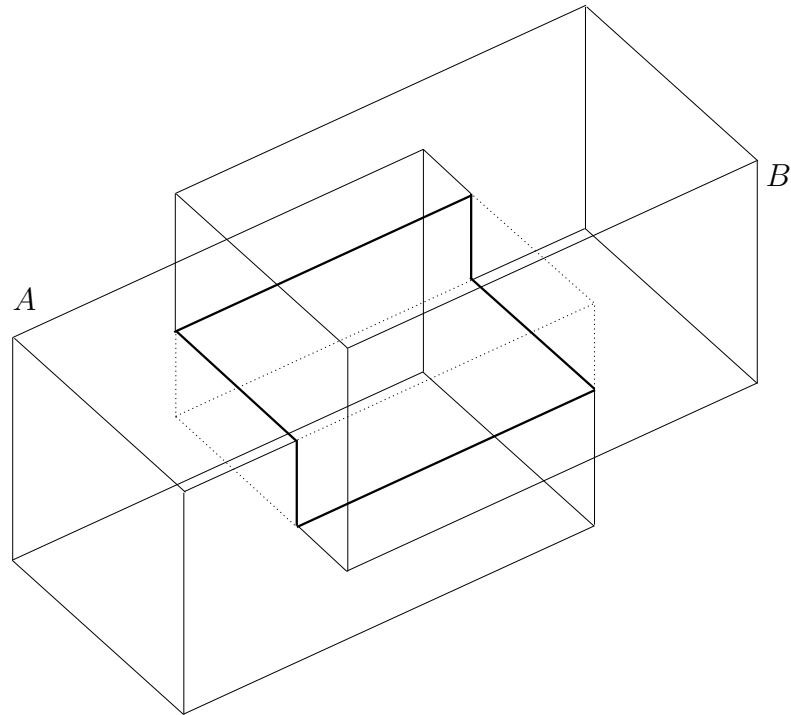


Abbildung 4.10: Verschneidungsgraph zweier Volumenkörper

und entsprechende Startparameter im Newton-Raphson-Verfahren eliminiert werden. Bei Verschneidungen zweier Flächen $s(u, v)$ und $t(w, m)$ liegt jedoch ein unterbestimmtes Gleichungssystem vor:

$$\begin{aligned} s_x(u, v) &= t_x(w, m) \\ s_y(u, v) &= t_y(w, m) \\ s_z(u, v) &= t_z(w, m) \end{aligned} \tag{4.38}$$

Schnittkurven können somit nicht direkt aus dem Gleichungssystem ermittelt werden, da vier Unbekannten nur drei Gleichungen gegenüberstehen. Eine indirekte Methode der Ermittlung der Schnittkurve ist das so genannte *Curve-Tracing* [15]. Das Verfahren lässt sich wie folgt darstellen:

1. Ermittlung eines Schnittpunktes zwischen beiden Flächen als Startpunkt. Dies geschieht durch Verschneidung von Kurven einer Fläche, welche parallel zu einer Koordinatenrichtung sind mit der zu verschneidenden Fläche.
2. Abschätzen eines neuen Punktes auf der Schnittkurve mithilfe der approximierten Tangente bezüglich des aktuellen Punktes.
3. Verbesserung der Koordinaten des neuen Schnittpunktes mithilfe numerischer Verfahren.

Es ist offensichtlich, dass die Stabilität des Verfahrens von der Schrittweite zwischen den Punkten auf der Kurve sowie der Komplexität der zu verschneidenden Flächen abhängt.

4.4.3 Randklassifizierung

Durch den Verschneidungsgraphen werden die jeweiligen Faces der Körper A und B in neue Faces geteilt. Dabei wird nur eine neue Topologie erzeugt, so dass die beiden neuen Faces einer geteilten Face die selbe Geometrie besitzen. Im folgenden bezeichnet $b(A)$ die Berandung von A und $b(B)$ die Berandung von B . Die geteilten Faces werden dementsprechend klassifiziert in A_{iB} für Faces von A welche innerhalb B ($A_{iB} \subseteq B$) und A_{aB} für Faces von A welche außerhalb von B ($A_{aB} \not\subseteq B$) liegen. Die vier Mengen an Flächen A_{iB} , A_{aB} , B_{iA} und B_{aA} stellen dabei die Randklassifizierung von A und B dar.

Die booleschen Operationen können durch die Randklassifizierung von A und B ausgedrückt werden:

$$A \cap B = A_{aB} \oplus B_{aA} \quad (4.39)$$

$$A \cup B = A_{iB} \oplus B_{iA} \quad (4.40)$$

$$A \setminus B = A_{aB} \oplus (B_{iA})^{-1} \quad (4.41)$$

wobei \oplus die Operation des Zusammenfügens und $(B_{iA})^{-1}$ die Menge an Flächen mit bezüglich zu B_{iA} invertierten Normalenvektoren darstellt.

4.4.4 Regularisierte und nicht-regularisierte Boolesche Operationen

Falls A und B zwei Körper im mathematischen Sinne, also 2-mannigfaltig sind, erfüllen sie die folgenden Kriterien.

1. Jede Edge gehört zu genau zwei Faces.
2. Jeder Knoten ist umgeben durch eine einfache Folge von Edges und Faces.
3. Faces dürfen sich außer an gemeinsamen Kanten und Knoten nicht schneiden.

Gewöhnliche boolesche Operatoren stellen nicht sicher, dass der resultierende Körper wieder 2-mannigfaltig ist. Sie werden daher auch als *nicht-regularisiert* bezeichnet. Abbildung 4.11 zeigt eine nicht-regularisierte Verschneidung zwischen zwei mannigfaltigen Körpern, dessen Resultat ein nicht-mannigfaltiger Körper ist. Die durch die Verschneidung entstandene hängende Face verletzt dabei die Kriterien an einen mannigfaltigen Körper. *Regularisierte* boolesche Operatoren \cap^* , \cup^* und \setminus^* stellen durch eine differenziertere Unterscheidung in Berandung und inneres Volumen einen mannigfaltigen Körper als Resultat sicher. Bei regularisierten Operatoren sind alle Randobjekte zu einem Objekt im Inneren benachbart. Die Elimination von hängenden Edges oder Faces wird daher auch als Regularisierung bezeichnet. Bei regularisierten Operatoren werden nur die Objekte des Verschneidungsgraphen betrachtet, bei denen die inneren Volumen beider Körper auf der gleichen Seite wie die gemeinsame Berandung liegen.

4.5 Die Blending-Funktionen-Methode

Wie bereits im Kapitel 3 erläutert, liegt der Unterschied zwischen h - und p -Version in den Ansätzen der Diskretisierung. Beim isoparametrischen Konzept wird das Verschiebungsfeld

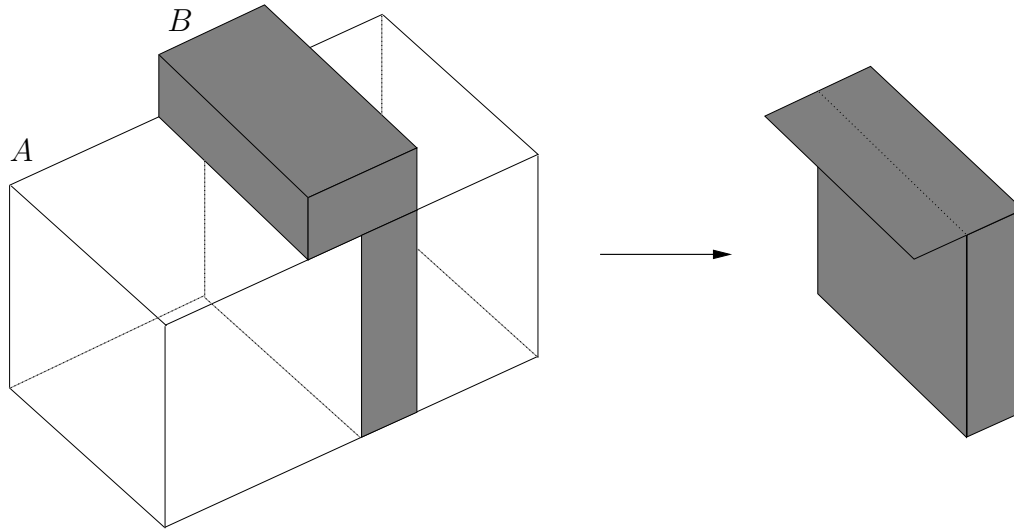


Abbildung 4.11: Nicht-regularisierte Verschneidung

durch die gleichen Funktionen wie die Geometrie beschrieben [83, 85]. Bei der h -Version wird der Elementdurchmesser schrittweise verkleinert um eine konvergente Lösung zu erhalten. Die Approximation der Geometrie ist dabei an die Verfeinerung des Netzes gekoppelt. Die p -Version verwendet einen reicheren Ansatzraum über den Finiten Elementen, so dass hierbei das Finite-Element-Netz in der Regel nicht verfeinert wird. Daher ist es notwendig, die Randgeometrie der relativ großen Elemente unabhängig vom Elementdurchmesser zu beschreiben. Eine Möglichkeit der exakten Beschreibung komplexer Geometrien ist dabei die Blending-Funktionen-Methode von GORDON und HALL [31, 32]. Verglichen mit dem isoparametrischen Konzept der h -Version zeigt HOLZER in [40], dass die Beschreibung der Geometrie mit der Blending-Funktionen-Methode zu einer Verbesserung der Ergebnisqualität führt.

4.5.1 Zweidimensionale Blending-Funktionen-Methode

Im folgenden soll das zweidimensionale Blending für ein Scheibenelement erläutert werden, dessen lokale Kanten $E_i, i = 1, \dots, 4$ mit einer gekrümmten Geometrie attribuiert sind. Dabei wird die Geometrie der Kanten durch eine parametrische Beschreibung

$$\mathbf{E}_i = \begin{bmatrix} E_{ix}(\xi) \\ E_{iy}(\xi) \end{bmatrix} \quad i = 1, 3 \quad \mathbf{E}_j = \begin{bmatrix} E_{jx}(\eta) \\ E_{jy}(\eta) \end{bmatrix} \quad j = 2, 4 \quad \xi, \eta \in [-1, 1] \quad (4.42)$$

definiert.

Die zweidimensionale Blending-Funktionen-Methode bildet nun die Kante E_2 über die Abbildungsfunktion

$$\mathbf{Q}_e(\xi, \eta) = \mathbf{x}(\xi, \eta) = \sum_{i=1}^4 N_i(\xi, \eta) \mathbf{X}_i + \sum_{i=1}^4 \mathbf{e}_i(\xi, \eta), \quad (4.43)$$

vom Parameterraum $\boldsymbol{\xi} = [\xi, \eta]$ in den euklidischen Raum \mathbb{R}^2 ab, wobei der erste Terme in Gleichung (4.43) dabei die üblichen bilinearen Abbildungsfunktionen darstellt, wie sie auch beim isoparametrischen Konzept für einen Polynomgrad $p = 1$ verwendet werden.

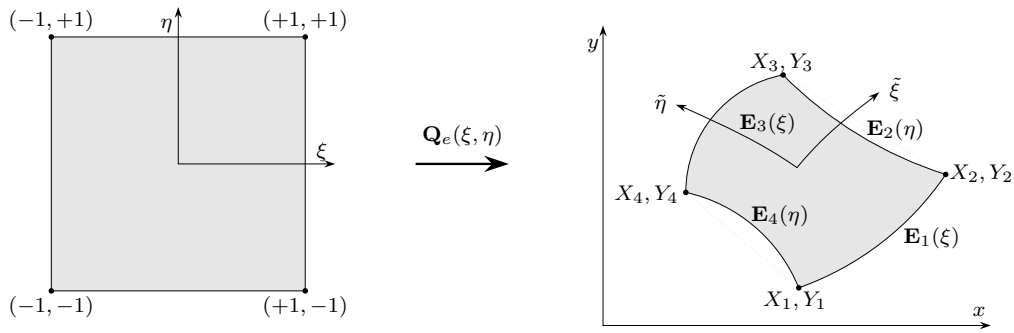


Abbildung 4.12: Die zweidimensionale Blending-Funktionen-Methode

$\mathbf{e}_i(\xi, \eta), i = 1, \dots, 4$ beschreiben die mit dem Blending-Term multiplizierte Differenz der gekrümmten Kurve E_i zur linearen Verbindung der die Kurve berandenden Knoten. Für Kante E_2 z. B. ergibt sich:

$$\begin{aligned} e_{2x}(\xi, \eta) &= \left(E_{2x}(\eta) - \left(\frac{1-\eta}{2} x_2 + \frac{1+\eta}{2} x_3 \right) \right) \frac{1+\xi}{2} \\ e_{2y}(\xi, \eta) &= \left(E_{2y}(\eta) - \left(\frac{1-\eta}{2} y_2 + \frac{1+\eta}{2} y_3 \right) \right) \frac{1+\xi}{2} \end{aligned} \quad (4.44)$$

Der Blending-Term $\frac{1+\xi}{2}$ stellt dabei sicher, dass die Kante E_2 linear zur gegenüberliegenden Kante E_4 ausgeblendet wird. Mithilfe der Definition von $\mathbf{e}_i, i = 1, \dots, 4$ kann Gleichung (4.43) umformuliert werden zu

$$\begin{aligned} \mathbf{Q}_e(\xi, \eta) &= \frac{1}{2} [\mathbf{E}_1(\xi) (1-\eta) + \mathbf{E}_2(\eta) (1+\xi) + \mathbf{E}_3(\xi) (1+\eta) + \mathbf{E}_4(\eta) (1-\xi)] \\ &\quad - \sum_{i=1}^4 N_i(\xi, \eta) \mathbf{X}_i. \end{aligned} \quad (4.45)$$

Wird das Flächen-Blending auf den euklidischen Raum \mathbb{R}^3 erweitert, so entspricht die Darstellung der Geometrie den Coons-Patches [2], bei denen die geometrische Beschreibung der Fläche nur durch die Berandung gegeben ist und daher eine bilineare Interpolation der berandenden Kanten darstellt. Die geometrische Beschreibung der Kanten für die Blending-Funktionen-Methode wird in Kapitel 4.6 vorgestellt.

4.5.2 Dreidimensionale Blending-Funktionen-Methode

Zur geometrischen Beschreibung allgemeiner, gekrümmter Volumenelemente, muss das in Kapitel 4.5.1 vorgestellte Verfahren erweitert werden. Die folgende Beschreibung orientiert sich an [9, 20, 44].

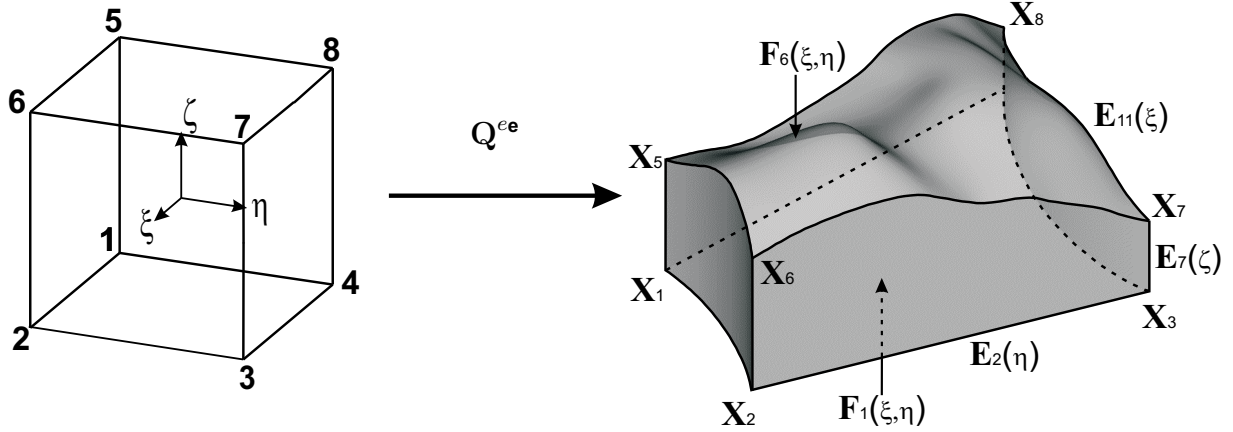


Abbildung 4.13: Die Blending-Funktionen-Methode für Hexaeder-Elemente

Das in Abbildung 4.13 dargestellte Hexaeder-Element kann über seine Randgeometrie, bestehend aus 8 Knoten

$$\mathbf{X}_i = \begin{bmatrix} X_{ix} \\ X_{iy} \\ X_{iz} \end{bmatrix} \quad i = 1, \dots, 8 \quad (4.46)$$

12 Kanten

$$\mathbf{E}_j(t) = \begin{bmatrix} E_{jx}(t) \\ E_{jy}(t) \\ E_{jz}(t) \end{bmatrix} \quad i = 1, \dots, 12 \quad \begin{array}{ll} t = \xi & \text{für } i = 1, 3, 9, 11 \\ t = \eta & \text{für } i = 2, 4, 10, 12 \\ t = \zeta & \text{für } i = 5, 6, 7, 8 \end{array} \quad (4.47)$$

sowie 6 Flächen

$$\mathbf{F}_k(u, v) = \begin{bmatrix} F_{kx}(u, v) \\ F_{ky}(u, v) \\ F_{kz}(u, v) \end{bmatrix} \quad k = 1, \dots, 6 \quad \begin{array}{ll} (u, v) = (\xi, \eta) & \text{für } k = 1, 6 \\ (u, v) = (\xi, \zeta) & \text{für } k = 2, 4 \\ (u, v) = (\eta, \zeta) & \text{für } i = 3, 5 \end{array} \quad (4.48)$$

exakt als B-Rep-Körper beschrieben werden. Die Abbildungsfunktion vom lokalen Elementkoordinatensystem in den euklidischen Raum \mathbb{R}^3 ergibt sich zu

$$\mathbf{Q}^e(\xi, \eta, \zeta) = \sum_{i=1}^8 N_{1,1,1}^{N_i}(\xi, \eta, \zeta) \mathbf{X}_i + \sum_{i=1}^6 \mathbf{f}_i(\xi, \eta, \zeta) - \sum_{i=1}^{12} \mathbf{e}_i(\xi, \eta, \zeta). \quad (4.49)$$

Dabei stellt der erste Term die trilineare Abbildung des isoparametrischen, achtknotigen Hexaeder-Element dar. Der zweite Term beschreibt das Flächen-Blending als Produkt aus Korrekturterm mit der Differenz der gekrümmten Fläche und der bilinearen Regelfläche bezüglich der die Fläche berandenden Knoten. Der dritte Term ist das in Kapitel 4.5.1 formulierte Kanten-Blending, wobei hierbei mit einem zweiten Korrekturterm zu multiplizieren ist, da die Kante zu den drei topologisch parallelen Kanten bilinear auszublenden ist. Da jede Kante durch zwei Flächen beschrieben wird, muss sie einmal in Gleichung (4.49) subtrahiert werden.

Durch Substitution der Flächen- $\mathbf{f}_i(\xi, \eta, \zeta)$, $i = 1, \dots, 6$ und Kanten-Blending-Terme $\mathbf{e}_i(\xi, \eta, \zeta)$, $i = 1, \dots, 12$ folgt aus Gleichung (4.49)

$$\begin{aligned}
\mathbf{Q}^e(\xi, \eta, \zeta) = & \frac{1}{2} \left[(1 - \zeta) \mathbf{F}_1(\xi, \eta) + (1 - \eta) \mathbf{F}_2(\xi, \eta) + (1 + \xi) \mathbf{F}_3(\xi, \eta) + \right. \\
& \left. + (1 + \eta) \mathbf{F}_4(\xi, \eta) + (1 - \xi) \mathbf{F}_5(\xi, \eta) + (1 + \zeta) \mathbf{F}_6(\xi, \eta) \right] - \\
& - \frac{1}{4} \left[(1 - \eta)(1 - \zeta) \mathbf{E}_1(\xi) + (1 + \xi)(1 - \zeta) \mathbf{E}_2(\eta) + (1 + \eta)(1 - \zeta) \mathbf{E}_3(\xi) + \right. \\
& (1 - \xi)(1 - \zeta) \mathbf{E}_4(\eta) + (1 - \xi)(1 - \eta) \mathbf{E}_5(\zeta) + (1 - \xi)(1 - \eta) \mathbf{E}_6(\zeta) + \\
& (1 + \xi)(1 + \eta) \mathbf{E}_7(\zeta) + (1 - \xi)(1 + \eta) \mathbf{E}_8(\zeta) + (1 - \eta)(1 + \zeta) \mathbf{E}_9(\xi) + \\
& \left. (1 + \xi)(1 + \zeta) \mathbf{E}_{10}(\eta) + (1 + \eta)(1 + \zeta) \mathbf{E}_{11}(\xi) + (1 - \xi)(1 + \zeta) \mathbf{E}_{12}(\eta) \right] + \\
& + N_{1,1,1}^{N_1}(\xi, \eta, \zeta) \mathbf{X}_1 + N_{1,1,1}^{N_2}(\xi, \eta, \zeta) \mathbf{X}_2 + N_{1,1,1}^{N_3}(\xi, \eta, \zeta) \mathbf{X}_3 + \\
& + N_{1,1,1}^{N_4}(\xi, \eta, \zeta) \mathbf{X}_4 + N_{1,1,1}^{N_5}(\xi, \eta, \zeta) \mathbf{X}_5 + N_{1,1,1}^{N_6}(\xi, \eta, \zeta) \mathbf{X}_6 + \\
& + N_{1,1,1}^{N_7}(\xi, \eta, \zeta) \mathbf{X}_7 + N_{1,1,1}^{N_8}(\xi, \eta, \zeta) \mathbf{X}_8. \tag{4.50}
\end{aligned}$$

4.6 Beschreibung der Geometrie für die Blending-Funktionen-Methode

In den Kapiteln 4.5.1 - 4.5.2 wurde die Abbildung von Elementen mit beliebiger Geometrie vom lokalen Elementkoordinatensystem in den jeweiligen euklidischen Raum erläutert. Dabei wurde \mathbf{E} und \mathbf{F} als abstrakte, nicht näher erläuterte geometrische Attributierung der jeweiligen Kanten und Flächen eingeführt. Auf diese geometrische Beschreibung soll im vorliegenden Kapitel näher eingegangen werden.

4.6.1 Quasi-regionale Abbildung

In [44] wird zur Beschreibung der Geometrie ein transfinites diskretes Verfahren präsentiert. Dieses ist ähnlich zum isoparametrischen Konzept, da die Abbildung lokal über stückweise, kontinuierliche Polynome mit diskreten Stützstellen erfolgt, unterscheidet sich aber vom isoparametrischen Ansatz dadurch, dass für die Abbildungsfunktionen die Blending-Funktionen-Methode verwendet wird und sich diese von den Verschiebungs-Ansatzfunktionen unterscheiden. Statt der Ansatzfunktionen werden dabei Lagrange-Polynome

$$L_i(t) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(t - t_j)}{(t_i - t_j)} \quad i = 0, \dots, n \tag{4.51}$$

mit der Eigenschaft

$$L_i(t_k) = \delta_{ik} \tag{4.52}$$

verwendet. Die Approximation $P_n(x)$ wird durch eine Linearkombination der Lagrange-Polynome beschrieben:

$$P_n(t) = \sum_{k=0}^n f(t_k) L_k(t) \tag{4.53}$$

Die Approximation der Flächen durch das transfinite diskrete Abbildungsverfahren geschieht über das Tensorprodukt der Lagrange-Polynome in beiden Parameterrichtungen

$$\begin{aligned} P_n(r, s) = P_n(r) \cdot P_n(s) &= \left(\sum_{k=1}^n f(r_k, s) L_k(r) \right) \left(\sum_{j=1}^m f(r, s_j) L_j(s) \right) \\ &= \sum_{k=1}^n \sum_{j=1}^m f(r_k, s_j) L_k(r) L_j(s). \end{aligned} \quad (4.54)$$

Die Bestimmung der optimalen Stützstellen $f(x_k), k = 0, \dots, n$ der Polynominterpolation n -ten Grades ist Gegenstand vieler aktueller Forschungsarbeiten. CHEN und BABUŠKA geben in [13, 14] eine optimale, nicht-äquidistante Stützstellenverteilung für verschiedene Elementtypen an. Ziel der Bestimmung der optimalen Stützstellen ist es, die durch die Interpolation hervorgerufenen Oszillationen zu minimieren. Ein Maß für die Güte ist dabei die Lebesque-Konstante welche einen möglichst geringen Wert annehmen soll. Die Lebesque-Funktion

$$\lambda_n(x, S) = \sum_{i=0}^n |L_{i,k}(x)| \quad (4.55)$$

der Ordnung $n + 1$ über den Stützstellen $x_{n,0}, x_{n,1}, \dots, x_{n,n}$ des Stützstellenschemas S ist definiert durch die Summe der Lagrange-Polynome $L_{i,k}(x)$. Die Lebesque-Konstante der Ordnung $n + 1$ im Intervall $[a, b]$ ist definiert zu

$$\Lambda_n(S) = \max\{\lambda_n(x, S) : x \in [a, b]\}. \quad (4.56)$$

In Abbildung 4.14 sind die Lebesque-Funktionsverläufe für äquidistante und nicht-äquidistante BABUŠKA-CHEN-Stützstellen abgebildet. Daraus wird ersichtlich, dass die Lebesque-Konstante für äquidistante Stützstellen höher und damit die Interpolation ungenauer als bei Vorliegen einer nicht-äquidistanten Stützstellenverteilung nach BABUŠKA und CHEN ist.

Die Bestimmung der optimalen Stützstellenverteilung wird in [13, 14] ausführlich beschrieben. Mit dem hier vorgestellten Verfahren wird eine Möglichkeit zur polynomialen Beschreibung der Flächen \mathbf{F} und Kanten \mathbf{E} mit diskreten Stützstellen für die Abbildung \mathbf{Q}_e mit der Blending-Funktionen-Methode gegeben.

4.6.2 Abbildung von kontinuierlicher auf transfinite Form

Mit dem in Kapitel 4.6.1 erläuterten Verfahren kann die geometrische Beschreibung für die Blending-Funktionen-Methode aufgestellt werden. CAD-Geometrien liegen meist als Beziér-, B-Spline- oder NURBS-Beschreibungen in parametrischer Form vor. Gewöhnlich decken die Flächen und Kurven eines Elements in dieser Darstellung auch nur einen Teilbereich ihres Definitionsbereichs ab. Aus Gleichung (4.49) ist ersichtlich, dass die Stützstellen für die geometrische Beschreibung in der Blending-Funktionen-Methode in lokalen Elementkoordinaten (ξ, η, ζ) des Standard-Hexaeder-Elements gegeben sind. Ziel dieses Kapitels ist es daher eine Abbildungsfunktion zu finden, welche die im Geometriemodellierer für die parametrischen Kurven und Flächen definierten Parameter t bzw. (u, v) auf die lokalen Hexaeder-Elementkoordinaten (ξ, η, ζ) abbildet (siehe Tabelle 4.2).

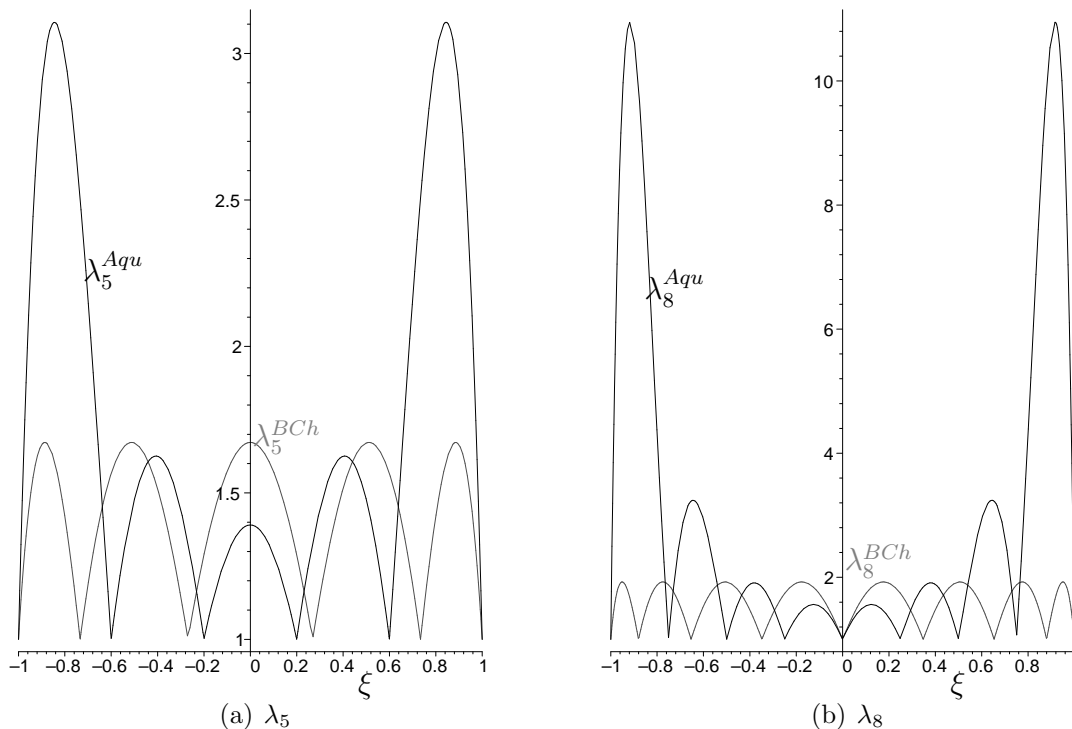


Abbildung 4.14: Lebesgue-Funktion λ_i , $i = 5, 8$ der Polynominterpolation für äquidistante und nicht-äquidistante Babuška-Chen-Stützstellen

	Hexaeder		lokale Element- koordinaten		B-Rep		euklidischer Raum
Kurven	(ξ, η, ζ)	\leftrightarrow	(q)	\leftrightarrow	(t)	\leftrightarrow	(x, y, z)
Flächen	(ξ, η, ζ)	\leftrightarrow	(r, s)	\leftrightarrow	(u, v)	\leftrightarrow	(x, y, z)

Tabelle 4.2: Abbildung der Geometrie FEM - CAD

Lokale Elementkoordinaten beziehen sich auf lokale Flächen und Kanten des Hexaeders und stellen einen Randbereich mit einem bzw. zwei konstanten Parametern aus ξ, η oder ζ dar. Das zugehörige Koordinatensystem orientiert sich an der lokalen Orientierung der Fläche bzw. Kante. Es ergibt sich daraus die Abbildungsvorschrift für Kurven

$$\mathbf{E}_i(\xi, \eta, \zeta) = \begin{bmatrix} E_{ix}(t(q_i(\xi, \eta, \zeta))) \\ E_{iy}(t(q_i(\xi, \eta, \zeta))) \\ E_{iz}(t(q_i(\xi, \eta, \zeta))) \end{bmatrix} \quad i = 1, \dots, 12 \quad (4.57)$$

sowie für Flächen

$$\mathbf{F}_i(\xi, \eta, \zeta) = \begin{bmatrix} E_{ix}(u(r_i(\xi, \eta, \zeta), s_i(\xi, \eta, \zeta)), v(r_i(\xi, \eta, \zeta), s_i(\xi, \eta, \zeta))) \\ E_{iy}(u(r_i(\xi, \eta, \zeta), s_i(\xi, \eta, \zeta)), v(r_i(\xi, \eta, \zeta), s_i(\xi, \eta, \zeta))) \\ E_{iz}(u(r_i(\xi, \eta, \zeta), s_i(\xi, \eta, \zeta)), v(r_i(\xi, \eta, \zeta), s_i(\xi, \eta, \zeta))) \end{bmatrix} \quad i = 1, \dots, 6. \quad (4.58)$$

Dabei sind die Abbildungsfunktionen von Flächen und Kanten gekoppelt, um einen konsistenten B-Rep-Körper zu erhalten. Auf Seite des Geometriemodellierers wird dies über die bi-parametrische Datenstruktur P-Curve (siehe Kapitel 4.2) sichergestellt. Bei der FE-

Datenstruktur geschieht dies über Bereiche mit konstantem Parameter am jeweiligen Rand, da dort Flächen und Kanten auf ihrem gesamten Parameterbereich definiert sind.

4.6.2.1 Kurven

Die diskreten Stützstellen q werden in einem Intervallbereich $q \in [-1; +1]$ ermittelt. Der Parameterbereich einer NURBS-Kurve ist im allgemeinen nicht auf diesen Bereich skaliert, so dass eine lineare Reparametrisierung mit der Vorschrift

$$t(q) = \frac{1}{2}(t_1 - t_0)(q + 1) + t_0 \quad t \in [t_0; t_1] \quad (4.59)$$

durchgeführt werden muss. Bei analytischen Kurvenbeschreibungen führt dieses Vorgehen zu einer längentreuen Parametrisierung. Anders sieht dies bei nicht-uniformen Parametrisierungen wie sie bei NURBS oder B-Splines verwendet werden aus. Hier wird die nicht-längentreue Parametrisierung t gezielt zur Beeinflussung des Kurvenverlaufs eingesetzt (siehe Kapitel 4.2.1.2). Durch die lineare Abbildung in Gleichung (4.59) bleiben diese Verzerrungen im Parameterbereich q der Kurve im Hexaederelements erhalten (siehe Abbildung 4.15).

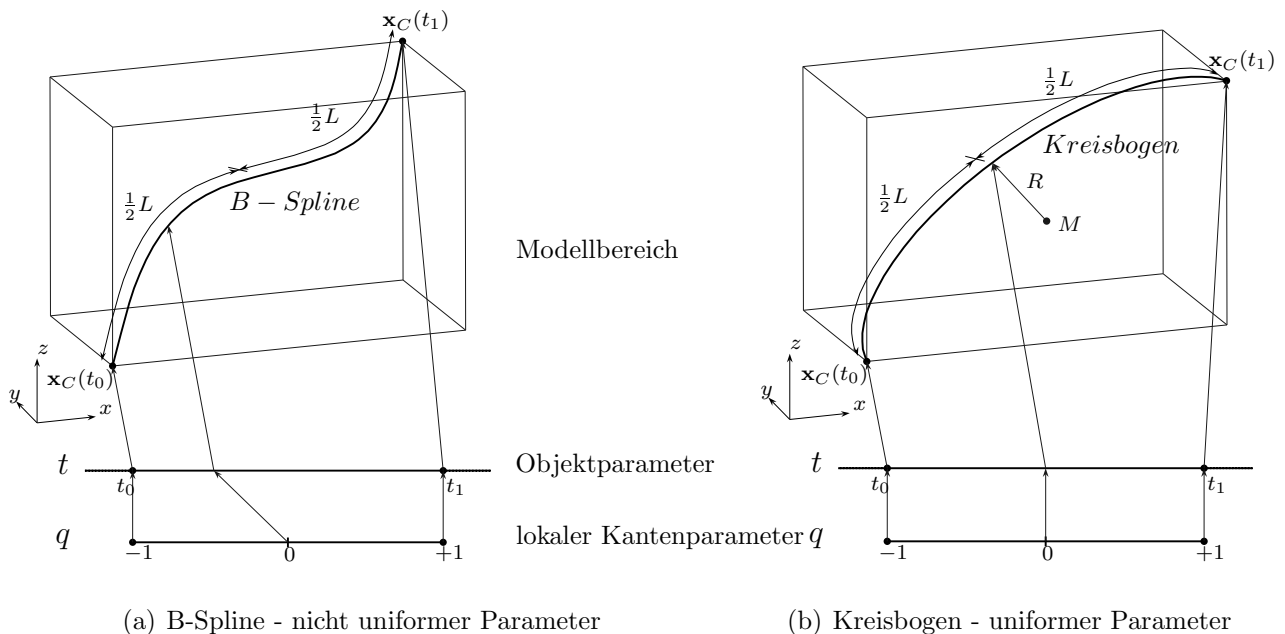


Abbildung 4.15: Parametrisierung der Kurve

Verzerrte Abbildungen im Parameterbereich des Hexaederelements sind gewöhnlich unerwünscht, da sie zu Verlusten in der Genauigkeit führen können. Bereiche mit einer hohen Geschwindigkeit in der Parametrisierung ergeben betragsmäßig kleine Jacobi-Determinanten, welche zu Spannungskonzentrationen bei der FE-Approximation führen können. Desweiteren könnte es zu Singularitäten in der Abbildung kommen. Daher wird im folgenden für Kurven eine Transformationsvorschrift aufgestellt, welche eine uniforme Reparametrisierung $\bar{t}(q)$

ermöglicht:

$$\bar{t}(q) = \frac{1}{2} (1 + q) L \quad q \in [-1; +1] \quad (4.60)$$

Dabei beschreibt L die Bogenlänge von Start- bis Endpunkt der Kurve und ermöglicht eine längentreue Reparametrisierung.

4.6.2.2 Flächen

Flächen sind im Geometriemodellierer allgemein nur auf einem Teilbereich ihrer Parameterbereichs $[u_0, v_0] \times [u_1, v_1]$ definiert. Kanten besitzen daher P-Curves, welche eine bi-parametrische Abbildung zwischen Kanten- und Flächenparametrisierung ermöglichen.

Würde $u = \text{const.}$ und $v = \text{const.}$ an den Rändern gelten, könnte die Abbildung $(u, v) \rightarrow (r, s)$ durch eine bilineare Interpolation geschehen. Da dies jedoch für den allgemeinen Fall nicht vorliegt, ist eine erweiterte Abbildungsvorschrift erforderlich. Ein geeignetes Verfahren stellt die zweidimensionale Blending-Funktionen-Methode dar (siehe Abbildung 4.16).

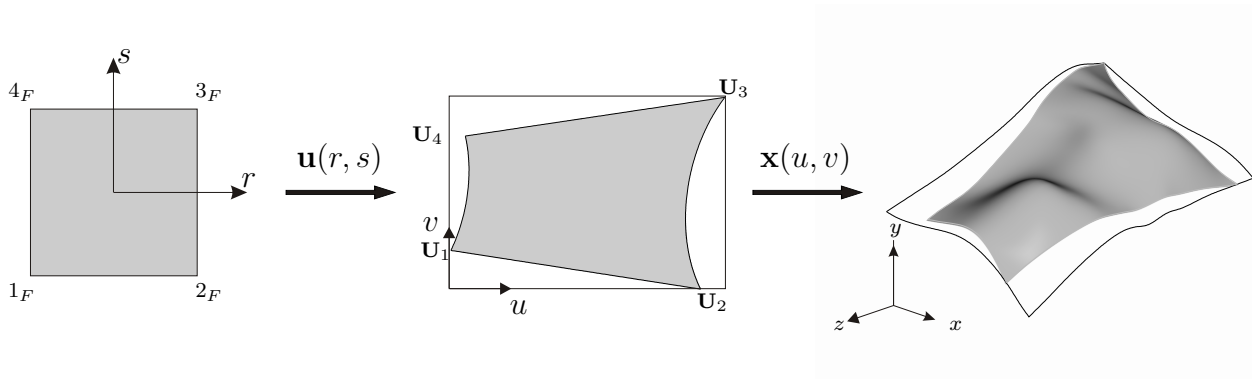


Abbildung 4.16: Abbildungsvorschrift durch Blending-Funktion in 2D

Die Abbildungsvorschrift lautet dabei

$$u(r, s) = \frac{1}{2} [\mathbf{E}_{1u}(r)(1 - s) + \mathbf{E}_{2u}(s)(1 + r) + \mathbf{E}_{3u}(r)(1 + s) + \mathbf{E}_{4u}(s)(1 - r)] - \sum_{i=1}^4 N_i(r, s) \mathbf{U}_{iu} \quad (4.61)$$

$$v(r, s) = \frac{1}{2} [\mathbf{E}_{1v}(r)(1 - s) + \mathbf{E}_{2v}(s)(1 + r) + \mathbf{E}_{3v}(r)(1 + s) + \mathbf{E}_{4v}(s)(1 - r)] - \sum_{i=1}^4 N_i(r, s) \mathbf{U}_{iv} \quad (4.62)$$

wobei

$$\mathbf{E}_i(r, s) = \begin{bmatrix} u(t(r, s)) \\ v(t(r, s)) \end{bmatrix} \quad (4.63)$$

die P-Curves darstellen, welche die inverse Abbildung vom Parameterbereich der Kurve in den Parameterbereich der Fläche beinhalten. Da Fälle denkbar wären, in denen auf die Erzeugung von P-Curves verzichtet wird, wird nachfolgend ein Verfahren präsentiert, das die Abbildung zwischen den Parameterbereichen von Kurve t und Fläche (u, v) über globale Koordinaten ermöglicht. Wie in Kapitel 4.6.2.1 bereits vorgestellt, können zu einem Kurvenparameter t die globalen Koordinaten (x, y, z) ermittelt werden. Abbildung 4.17 zeigt die inverse Abbildung der Flächen von globalen Koordinaten (x, y, z) auf den Parameterbereich (u, v) , welche eine Abbildung vom Parameterbereich der Kanten t auf den Parameterbereich der Flächen (u, v) ermöglicht [9, 21].

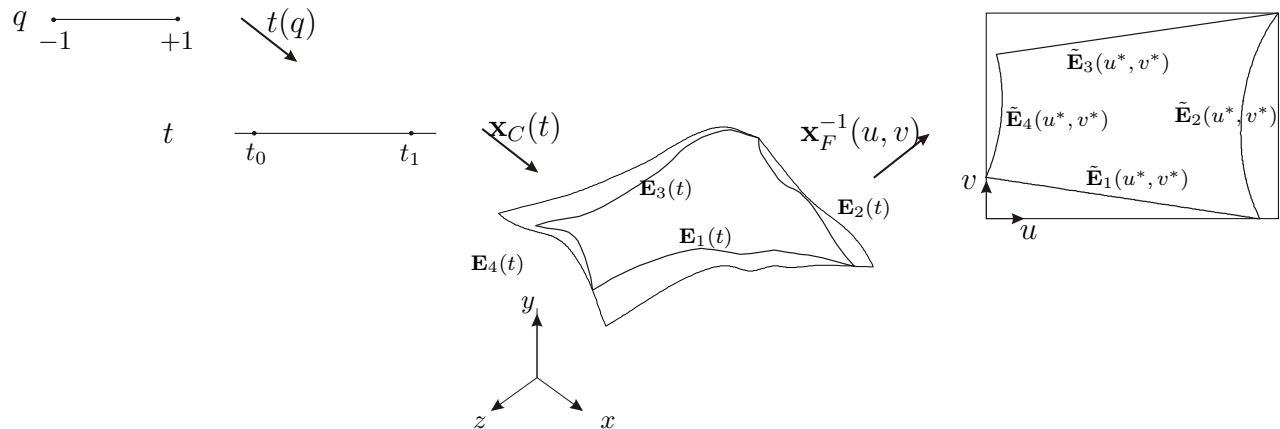


Abbildung 4.17: Abbildung der Kante auf die Fläche

Bei NURBS- und B-Spline-Flächen kann es aufgrund der nicht-uniformen Parametrisierung (u, v) genauso wie bei Kurven zu einer verzerrten Abbildung kommen. Wie in Abbildung 4.18 zu erkennen ist, kann diese Verzerrung sogar dazu führen, dass die Parameterabbildung bei Polstellen außerhalb des Elementes zum liegen kommen können. Eine entzerrende Reparametrisierung liegt durch längentreue Reparametrisierung der Kurven (siehe Kapitel 4.6.2.1) an den Rändern der Fläche bereits vor. Für die Reparametrisierung im Innern der Fläche muss das bisherige Verfahren erweitert werden. Da die Parametrisierungen der beiden Richtungen miteinander gekoppelt sind, wurde für die Reparametrisierung bei Flächen ein iteratives lokales Relaxationsverfahren entwickelt. Zu Beginn jeder Iteration werden alle diskreten Stützstellen auf den Definitionsbereich der betrachteten Fläche normal projiziert. Punkte für einen konstanten u - bzw. v -Parameter werden über eine B-Spline-Kurve miteinander verbunden. Diese B-Spline-Kurve wird dann schließlich auf die B-Spline-Oberfläche projiziert und entsprechend dem in Kapitel 4.6.2.1 beschriebenen Verfahren reparametrisiert. Dieses Verfahren wird iterativ (für alle Kurven mit konstantem u - bzw. v -Parameterwert) durchgeführt, bis die größte Verschiebung einer Stützstelle gegenüber der letzten Iteration unterhalb einer bestimmten Fehlerschranke liegt.

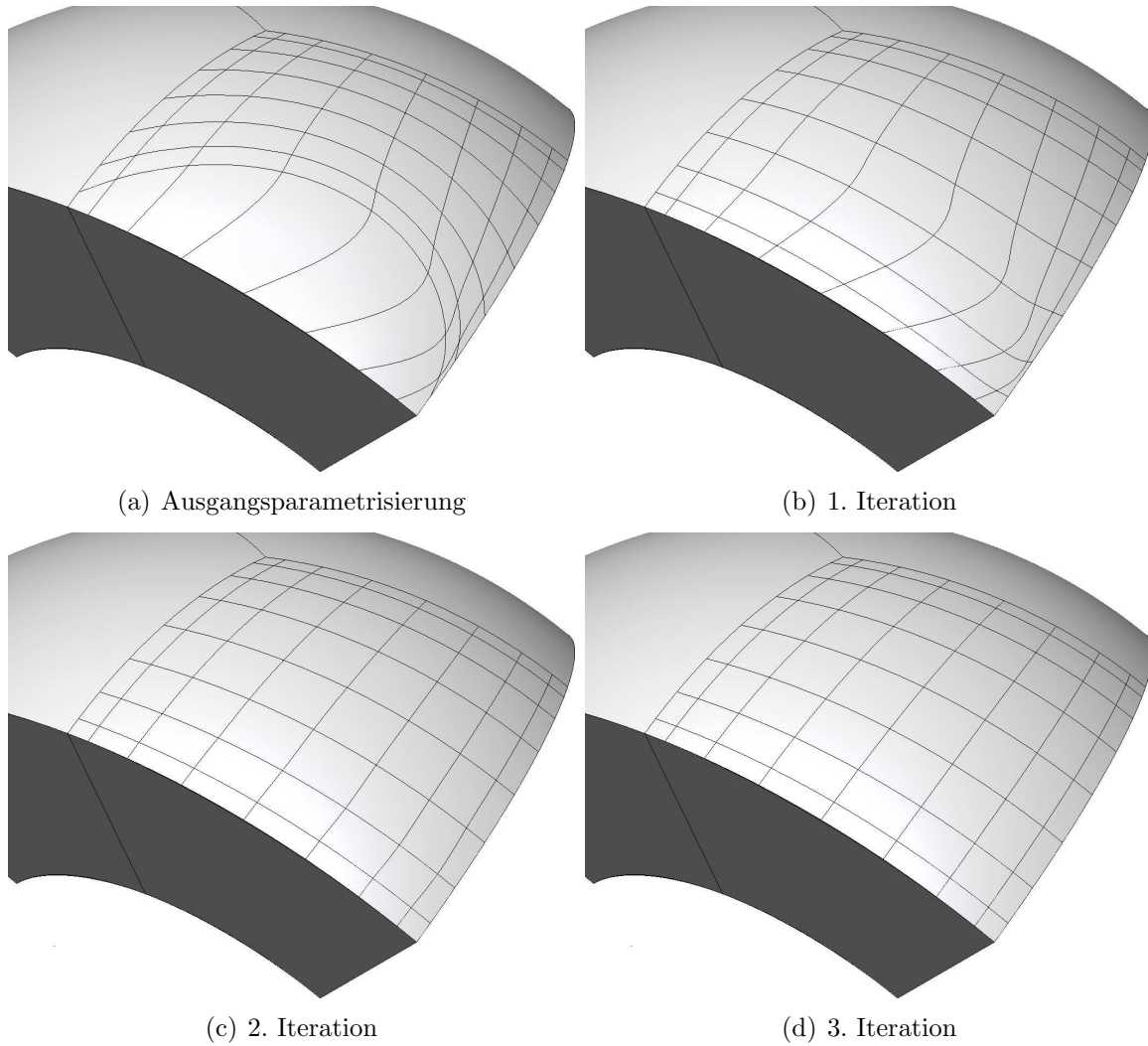


Abbildung 4.18: Relaxierungsverfahren zur Ermittlung der Stützstellen nach Babuška-Chen für eine verzerrte B-Spline-Oberfläche am Beispiel des in [52] verwendeten Kugelsegments

Kapitel 5

Kopplung der Simulationsverfahren

Nachdem in den vorangegangenen Kapiteln eine Beschreibung der im Rahmen dieser Arbeit verwendeten Verfahren und deren Grundlagen erfolgte, soll nun die Kopplung von Umformsimulation und Berechnung der elastischen Rückfederung erläutert werden. Die Kopplung wurde in den Präprozessor *Mesh & Marry* implementiert.

Ausgehend von der Motivation der Konvertierung von der h - zur p -Diskretisierung, wird ausführlich auf die Oberflächenrückführung der rekonstruierten Volumenkörper eingegangen. Weiter erfolgt die Erläuterung der Diskretisierung für die p -Version der Finite-Elemente-Methode mithilfe eines Schalenvernetzers und Projektion des Netzes auf die gekrümmte Oberflächengeometrie. Anschließend werden die im Bauteil vorhandenen Resteigenspannungen auf die rediskretisierte Geometrie transformiert. Hierfür werden die erforderlichen Such- und Interpolationsalgorithmen beschrieben. Spezielle, auf die Problematik der elastischen Rückfederung ausgelegte numerische Verfahren, ermöglichen einen weiteren Effizienzgewinn. Abschließend wird ein Ausblick auf die zur Dimensionierung der Ziehanlage notwendige Berechnung der Pressenkraft gegeben.

5.1 Motivation

Eine Umformsimulation besteht im wesentlichen aus der virtuellen Abbildung des Tiefziehprozesses, gefolgt von der Berechnung der elastischen Rückfederung. Da die Simulation des Tiefziehvorgangs ein bereits weit entwickeltes und zuverlässig eingesetztes Verfahren darstellt, wird im Rahmen dieser Arbeit der Tiefziehprozess mit einem kommerziellen Finite-Elemente-Programm berechnet, welches auf einer expliziten h -Version mit einer Schalenelementformulierung basiert. Die elastische Rückfederung liefert jedoch bei expliziter Berechnung häufig unzuverlässige Ergebnisse, implizite Simulationen sind dagegen sehr teuer in Bezug auf den Speicherplatzbedarf. Weiterhin unterliegen diese Berechnungen den Annahmen der dimensionsreduzierten Modelle, die nicht in allen Bereichen des Bauteils erfüllt werden können. Daher wird im Rahmen dieser Arbeit die elastische Rückfederung mit der p -Version der Finite-Elemente-Methode berechnet (siehe Abbildung 5.1). Die wesentlichen Vorteile dieser Vorgehensweise können wie folgt dargestellt werden:

- Durch den strikt dreidimensionalen Ansatz können Modellfehler, wie sie in dimensionsreduzierten Modellen auftreten, eliminiert werden.

- Aufgrund des physikalischen Prozesses werden keine Spannungssingularitäten in der Lösung erwartet. Durch diese Tatsache kann mit einer reinen p -Version eine exponentielle Konvergenz für den Fehler der Energienorm erreicht werden. Somit lässt sich die Effizienz und die erzielbare Genauigkeit erhöhen.
- Eine Kopplung der Elementgeometrie mit dem CAD-Modell ermöglicht eine Einbettung der Berechnung der elastischen Rückfederung in den gesamten Entwicklungsprozess.

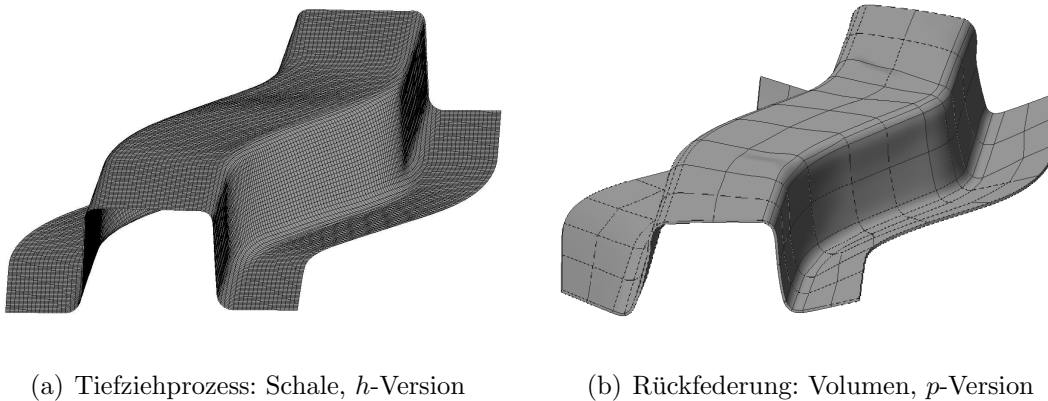


Abbildung 5.1: Umformen als gekoppelter Prozess

5.2 Volumenrückführung

Für die Tiefziehberechnung werden üblicherweise zweidimensionale, bilineare Schalenelemente mit kleinem Elementdurchmesser verwendet. Für die Berechnung der elastischen Rückfederung mit der p -Version der Finiten-Element-Methode ist dagegen eine Hexaeder-Diskretisierung mit großem Elementdurchmesser und exakter Geometriebeschreibung notwendig. Da im allgemeinen die Vergrößerung eines vorhandenen Netzes eine nicht triviale Aufgabe darstellt und außerdem die Extraktion von geometrischen Informationen für die neue Diskretisierung nicht direkt möglich ist, geschieht im folgenden Ansatz die Diskretisierung über eine vollständige Neuvernetzung.

Dabei wird zunächst aus der Schalengeometrie und der Information über die Elementdicke ein Volumenkörper konstruiert, welcher anschließend für die p -Version diskretisiert wird.

5.2.1 Extrusion des Schalennetzes

Da für die dreidimensionale Berechnung mit der p -Version Volumenelemente benötigt werden, muss zunächst aus dem Schalennetz ein Volumennetz generiert werden. Dazu werden in einem ersten Schritt die Normalenvektoren \mathbf{n}_{av} für jeden Knoten des Schalennetzes durch Mittelung der Flächen-gewichteten Normalenvektoren \mathbf{n}_i der am jeweiligen Knoten angrenzenden

Schalenmittelflächen berechnet:

$$\mathbf{n}_{\text{av}} = \sum_{i=0}^{n_f} \frac{A_i}{\sum_{i=0}^{n_f} A_i} \mathbf{n}_i \quad (5.1)$$

Nach Ermittlung der gemittelten Normalenvektoren wird jeder Knoten des Schalennetzes in positive und negative Richtung des Normalenvektors um jeweils die halbe Blechdicke $\frac{t}{2}$ extrudiert (siehe Abbildung 5.2):

$$\mathbf{v}_{\text{top, bottom}} = \mathbf{v}_{\text{mid}} \pm \frac{t}{2} \mathbf{n}_{\text{av}} \quad (5.2)$$

Durch die Extrusion hat das Netz eine Dimensionserhöhung in Dickenrichtung erfahren, besitzt in lokaler Elementebene allerdings noch immer die gleiche Diskretisierung.

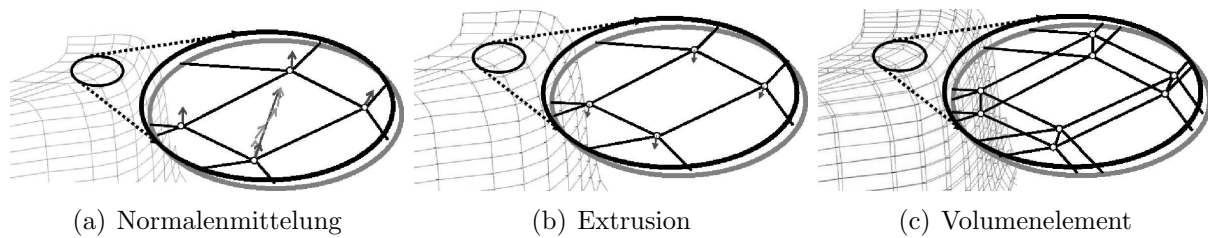


Abbildung 5.2: Extrusion Schalennetz - Hexaedernetz

5.2.2 Einteilung der Regionen

Durch starke Krümmungswechsel innerhalb eines Elements kann es durch das quasi-regionale Mapping (siehe Kapitel 4.6.2) zu hohen Abweichungen von der tatsächlichen Geometrie kommen. Dies kann durch eine Erhöhung des Polynomgrades der Geometriebeschreibung behoben werden. Meist treten diese Problemzonen nur lokal auf. Da eine lokale Erhöhung des Polynomgrades der Geometrie nicht ohne weiteres möglich ist und eine globale Erhöhung nach sich ziehen würde, stellt sich dieses Vorgehen als ineffizient dar. Eine Partitionierung des zu vernetzenden Gebiets in geeignete Regionen verhindert, dass innerhalb eines Elements starke Krümmungs- bzw. Radienwechsel auftreten. Somit ist eine bezüglich der Ordnung der geometrischen Beschreibung effiziente Diskretisierung für die p -Version der Finite-Elemente-Methode möglich.

5.2.2.1 Kantendetektion

Zur Einteilung der Geometrie in Regionen müssen die entsprechenden Bereiche separiert werden. Da für die Trennung dieser Bereiche Krümmungs- und Radienwechsel maßgebliche Kriterien darstellen, ist eine mögliche Vorgehensweise die Verwendung von Kantendetektionsverfahren [12] aus der Bildverarbeitung. Dabei basieren die meisten Verfahren auf Gradienten- oder Laplacemethoden [41]. Überschreitet eine Krümmung einen bestimmten Wert, ist dies ein Kriterium für das Vorliegen einer Kante. Bei Laplacemethoden werden zusätzlich noch die zweiten Ableitungen untersucht. Segmentierungsmethoden sind meist pixelorientiert. Dies

bedeutet, dass allgemeine dreidimensionale Objekte in ihre Parameterebene abgebildet werden und eine der erforderlichen Genauigkeit entsprechende Diskretisierung erfahren müssen. Den diskreten „Pixeln“ werden als „Farbwerte“ die entsprechenden Gradienten zugeordnet. Ein Kantendetektionsverfahren ist z. B. das Sobelverfahren (siehe Abbildung 5.3), welches die euklidische Norm M_A des diskretisierten Gradienten misst.

$$M_A := \sqrt{du(u,v)^2 + dv(u,v)^2} \quad (5.3)$$

wobei

$$du(u,v) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad dv(u,v) = \begin{bmatrix} 1 & 2 & 1 \\ -0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (5.4)$$

die bezüglich der Parameterebene (u,v) diskreten Ableitungen in u - und v -Richtung beschreiben. Da die Kantendetektion in diskreter Form erfolgt, müssen die ermittelten Kantenpunkte

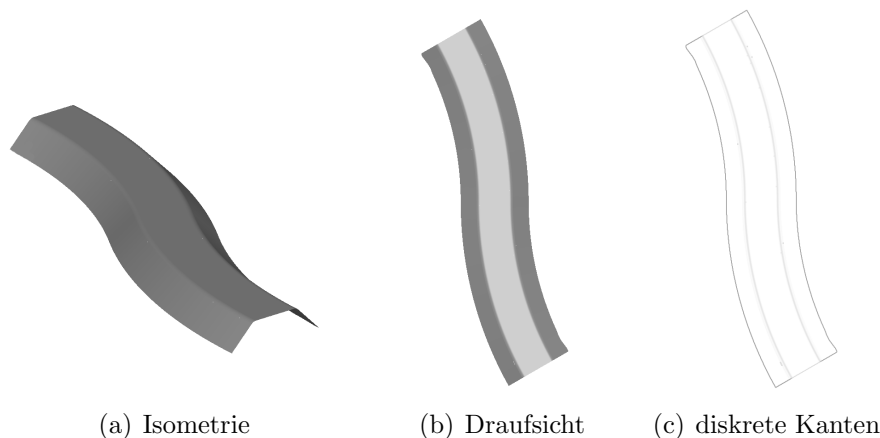


Abbildung 5.3: Kantendetektion am Beispiel des S-Bauteils [36] mithilfe des Sobel-Filters

zu einer Kante zusammengefügt werden. Dies ist im allgemeinen ein nicht-trivialer Prozess. Eine weitere Schwierigkeit besteht in der Generierung der Regionstopologie aus den einzelnen Kantenobjekten. Hier müssen oftmals auch noch Kantenzüge mit zusätzlichen Segmenten verbunden werden damit sich geschlossene Regionsbeschreibungen ergeben. Aufgrund der genannten Schwierigkeiten wurde für die Kantendetektion nach einem anderen Verfahren gesucht.

5.2.2.2 Regionsvererbung der Werkzeugpatches

Im Rückfederungsprozess besitzt zunächst nur die Werkzeugoberfläche eine CAD-Beschreibung, welche sowohl in parametrischer als auch in analytischer Form vorliegen kann. Da das Werkzeug die für die Platine prägende Form ist, besitzt sie in der Regel eine ähnliche Geometrie. Im Konstruktionsprozess werden die meist analytischen bzw. einfachen parametrischen Flächenbeschreibungen abschnittsweise für ihre jeweilige geometrische Grundform definiert. Die Patcheinteilung des Werkzeugs entspricht somit den Anforderungen an die Regionseinteilung des Bauteils für die Diskretisierung für die p -Version der Finite-Elemente-Methode. Allerdings ist die Geometrie des Werkzeugs der rückzuführenden Bauteilgeometrie nur ähnlich, da

sie zum einen um einen Offset verschoben ist (Oberfläche) und es zum anderen zu Klaffungen zwischen Bauteil und Werkzeug kommt. Daher entstand die Idee der Projektion der Patchberandung des Werkzeugs auf die umgeformte Platine [51, 65]. Durch diese Vorgehensweise entfällt die Kantendetektion und die damit verbundenen Schwierigkeiten der Generierung einer konsistenten Topologie und Geometrie.

Die-Stitching

Oft bieten sich die Werkzeugteilflächen als topologische Regionseinteilung für die p -Version an. Allerdings enthalten komplexere Geometrien oft sehr kleine Werkzeug-Patches, welche zu entsprechend kleinen Elementen führen würden. Im Die-Stitching-Schritt werden Patches erkannt, welche entweder sehr klein oder im Vergleich zu ihren Nachbarpatches einen geringen Unterschied ihres Normalenvektors (an charakteristischen Stellen auf der Oberfläche) aufweisen. Diese erkannten Patches werden nun zu ihren benachbarten Flächen hinzugefügt (Stitching [74]). Dadurch wird eine verbesserte Regionseinteilung für die Diskretisierung für die p -Version der Finite-Elemente-Methode ermöglicht.

5.2.3 Zuordnung Schalenelemente - Patch

Im folgenden soll die zu jedem Werkzeug-Patch zugehörige Bauteilgeometrie aus dem Schalennetz bzw. extrudierten Volumennetz erzeugt werden. Zunächst werden dazu jedem Werkzeug-Patch die entsprechenden Schalenelemente zugeordnet. Dazu wird mithilfe eines Oktalbaums (siehe auch Kapitel 5.4.3) ein Schalenelement der Mittelebene identifiziert, welches vom Normalenvektor des Die-Patches (an charakteristischen Stellen) getroffen wird. Ausgehend von diesem Schalenelement wird nun durch ein Ray-Firing-Verfahren [74] in Richtung seines Normalenvektors rekursiv für jedes Nachbarerelement ermittelt, ob es sich auf das Patch projizieren lässt. Falls es im Inneren liegt, wird es zusammen mit den extrudierten Ober- und Unterseiten zur Kandidatenliste hinzugefügt. Die endgültige Kandidatenliste erzeugt in der Regel eine unregelmäßige Berandung (Abbildung 5.4(a)). Nach der Zuordnung wird eine Regularisierung durchgeführt, bei der das Schalenelement-Patch um Elemente erweitert wird, so dass ein gleichmaschiges Netz mit einer gleichen Anzahl von Knoten in zwei Richtungen entsteht. Abschließend werden noch einige Elementreihen zur Verbesserung der Stetigkeit zu Nachbarregionen um die Kandidatenelemente addiert. Dieser Vorgang ist in Abbildung 5.4(b) dargestellt.

In Fällen, in denen eine Regularisierung nicht möglich ist, z. B. bei adaptiven oder gemischten Diskretisierungen, werden vier Eck- bzw. Randknoten interaktiv festgelegt. Mithilfe dieses Verfahrens wird für jedes Werkzeug-Patch jeweils eine für die Erstellung der Parameterflächen für Ober-, Unter- und Mittelfläche notwendige vierseitige Berandung gewährleistet. Innerhalb der Berandungen werden über ein bilineares Blending zunächst Parameterflächen erzeugt, wie dies auch bei Coons-Patches üblich ist (siehe Abbildung 5.5).

5.2.4 Flächenrückführung

Die durch das in Kapitel 5.2.3 beschriebene Verfahren erstellten Parameterflächen beschreiben die zu approximierende Geometrie nur an den Rändern exakt. Um die Beschreibung im Inneren zu erhalten, wird zunächst eine der gewünschten Genauigkeit genügende Auflösung festgelegt.

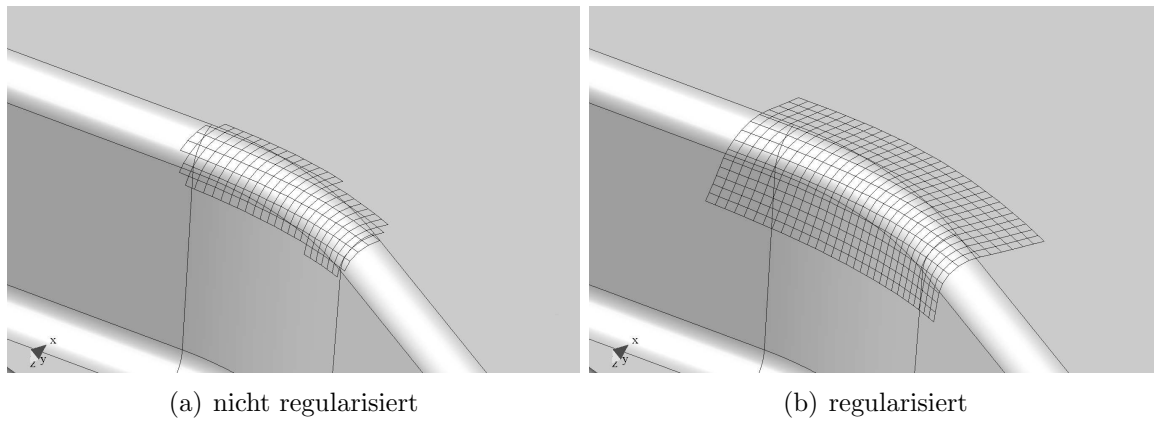


Abbildung 5.4: Zuordnung Schalenelement-Patch zu Werkzeug-Patch



Abbildung 5.5: Angenäherte Flächenbeschreibung durch bilineares Blending der vierseitigen Berandung

Für jedes Parametertupel der Mittelfläche wird eine Gerade, bestehend aus Koordinate des Flächendurchstoßpunktes und Normalenvektor bezüglich der Parameterfläche, festgelegt. Es wird nun mithilfe des Octrees das jeweilige Element des h -Netzes gesucht, welches mit der Gerade eine nicht-leere Schnittmenge bildet und die zugehörigen lokalen Koordinaten ermittelt. Bei Problemzonen (z. B. Schließen von Löchern) bei denen der Schnittpunkt nicht im Elementinneren ($|\xi| > 1 \vee |\eta| > 1$) liegt, wird die bilineare Fläche für das nächstgelegene Element entsprechend extrapoliert und die globale Koordinate ermittelt. Liegt der Schnittpunkt im Elementinneren, kann eine verbesserte Koordinate berechnet werden, indem aus Kenntnis der Nachbarelemente die Tangentenvektoren an den Elementrändern bestimmt und daraus eine kubische Parameterfläche aufgespannt wird (Flächenrückführung auf Elementebene) und die Koordinaten bezüglich dieser Fläche ausgewertet werden (siehe Abbildung 5.6).

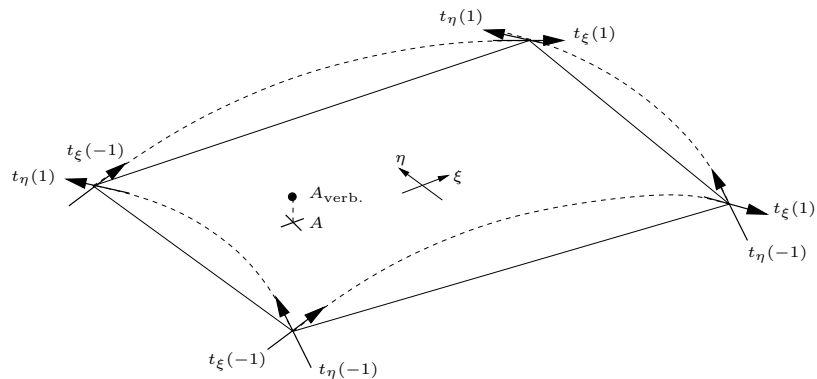


Abbildung 5.6: Verbesserte Koordinaten durch lokale, bikubische Approximation

Schließlich entsteht ein reguläres Netz von Punkten, welches mit dem in Kapitel 4.3.3 beschriebenen Verfahren durch eine parametrische B-Spline-Fläche approximiert werden kann. Dieses Verfahren wird sowohl für die Schalenelementebene als auch für Ober- und Unterfläche der durch Extrusion (siehe Kapitel 5.2.1) gewonnenen Volumenelemente durchgeführt. In Abbildung 5.7 sind diese drei Flächen (Ober-, Unter- und Mittelfläche) dargestellt.



Abbildung 5.7: Rückgeführte Flächenbeschreibung für Ober-, Unter-, und Mittelfläche

5.2.5 Projektion der Berandung

Für die Beschreibung der Region wird jeweils nur ein Teilbereich des rechteckigen Definitionsbereich der mit dem in Kapitel 5.2.4 erläuterten Verfahren erstellten Flächen benötigt. Diese Teilbereiche werden nachfolgend als Regionen bezeichnet. Die Begrenzung der Regionen wird durch die Projektion der zugehörigen Werkzeug-Patch-Berandung auf die rückgeführte Ober-, Unter- und Mittelfläche erzeugt. Abbildung 5.8 zeigt eine Region als Teilmenge der zugrunde liegenden geometrischen Beschreibung.

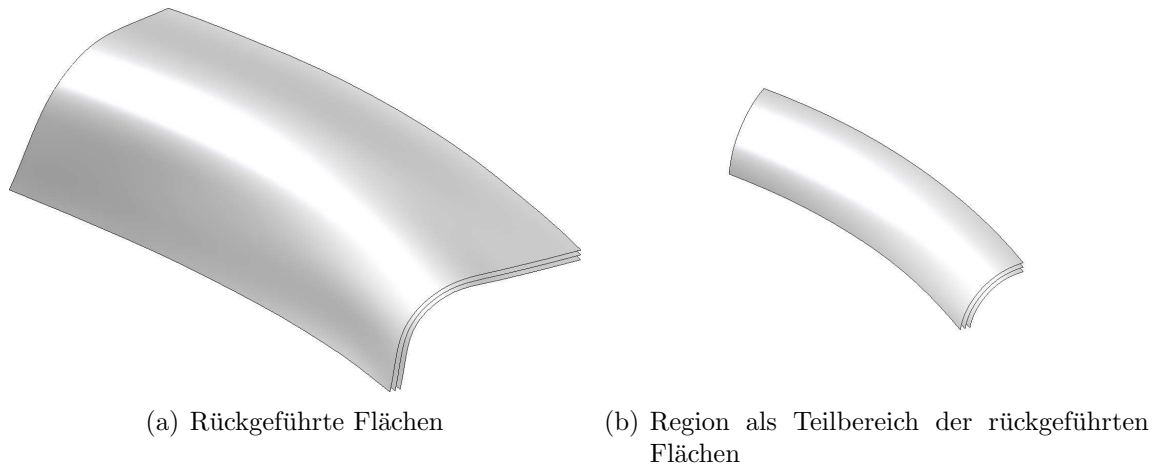


Abbildung 5.8: Projektion der Berandung

Da die Bauteilgeometrie gewöhnlich nur einen Teilbereich des Werkzeugs abdeckt, können nicht alle Kanten in ihrem gesamten Definitionsbereich projiziert werden. Zwangsläufig entstehen dadurch Regionen mit nicht geschlossener Berandung. Nicht geschlossene Regionen werden durch entsprechende Randkurven der Bauteilgeometrie geschlossen (siehe Abbildung 5.9).

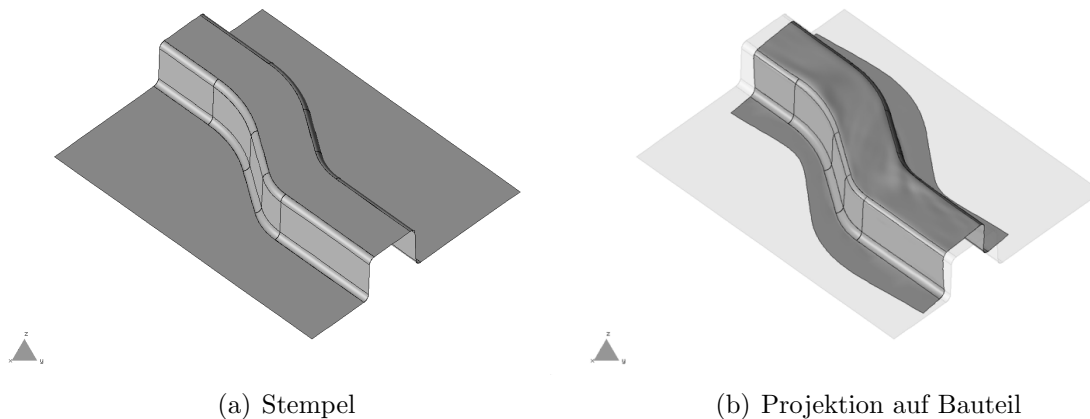


Abbildung 5.9: Regionseinteilung für S-Rail (Benchmarkbauteil der NUMISHEET 1996)

5.2.6 Erzeugung des Volumenkörpers

Im folgenden wird aus den einzelnen Oberflächen ein Volumenkörper generiert. Da jede Region für die einzelnen Flächen durch die Projektion auch eine eigene geometrische Beschreibung der Kanten besitzt, werden zunächst mithilfe des inkrementellen Algorithmus „Stitching“ Flächen so aneinander gefügt, dass sie einen topologisch vollständigen Körper mit sich teilender Randgeometrie bilden (siehe auch [73]). Dabei werden auch kleinere Lücken, welche sich durch den Projektionsprozess ergeben, geschlossen sowie ein G^1 -stetiger Übergang der Flächen erzeugt. Kanten welche nur auf einem Teilbereich mit einer benachbarten Kante zusammenfallen werden gesplittet, um bei der anschließenden Vernetzung hängende Knoten zu vermeiden (siehe Abbildung 5.10).

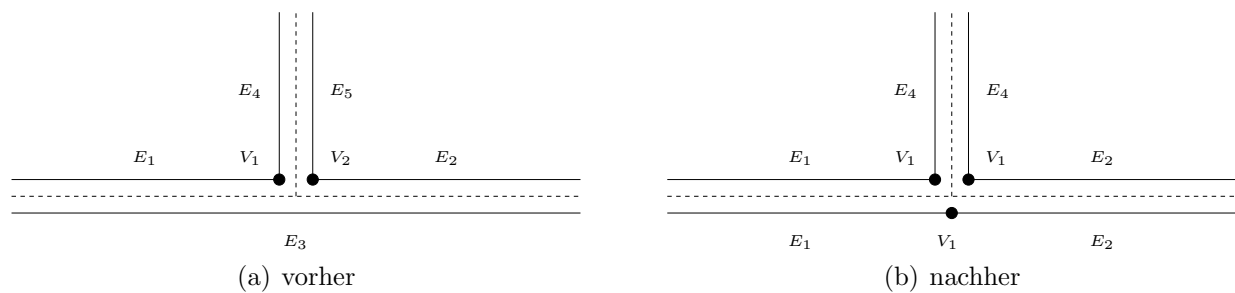


Abbildung 5.10: Stitching von Knoten und Kanten

Da schließlich Volumenkörper diskretisiert und somit ein Hexaeder-Netz generiert werden soll, wird dieser Vorgang für alle drei Flächen (Ober-, Unter-, und Mittelebene) einer Region durchgeführt. Nachdem die Oberflächen topologisch und geometrisch definiert sind, wird der Volumenkörper erzeugt. Da aus dem Schalennetz außer der diskreten Verteilung der Dicke keine weitere Information über die geometrische Ausdehnung in Dickenrichtung vorliegt, muss eine Annahme über die Beschreibung der Seitenflächen getroffen werden. Numerische Untersuchungen haben gezeigt, dass es sinnvoll ist, den Polynomgrad der Finite-Elemente-Lösung über die Dicke für den *anisotropic tensor product space* $\mathcal{S}^{p,p,q}(\Omega_{st}^h)$ zu $q > 2$ und für *anisotropic trunk space* $\mathcal{S}_{ts}^{p_\xi, p_\eta, p_\zeta}(\Omega_{st}^h)$ zu $q > 3$ zu wählen um Locking-Effekte zu vermeiden. Allerdings sollte aus Effizienzgründen $q \leq p$ gewählt werden. Um die exakte Wiedergabe von Starrkörperverschiebungen zu gewährleisten ist es notwendig, dass die Beschreibung der Geometrie einen Unterraum der Beschreibung der Verschiebungen bilden muss [9]. Um diese Forderung nach subparametrischen bzw. isoparametrischen Elementformulierungen zu erfüllen, werden die Seitenflächen als bilinear geblendete Flächen modelliert. Dazu bietet sich das Verfahren des *Skinnings* an, bei dem eine Fläche durch mindestens zwei Profile (Kurven) approximiert wird. Beim *Lofting* wird zusätzlich an Start- und Endprofil ein Tangentenvektor definiert. Abbildung 5.11 zeigt wie aus Ober- und Unterfläche mithilfe des Skinnings ein Volumenkörper erzeugt werden kann. Die Randkanten von Ober- und Unterfläche stellen dabei die Profile für die zu erzeugenden Seitenflächen dar.

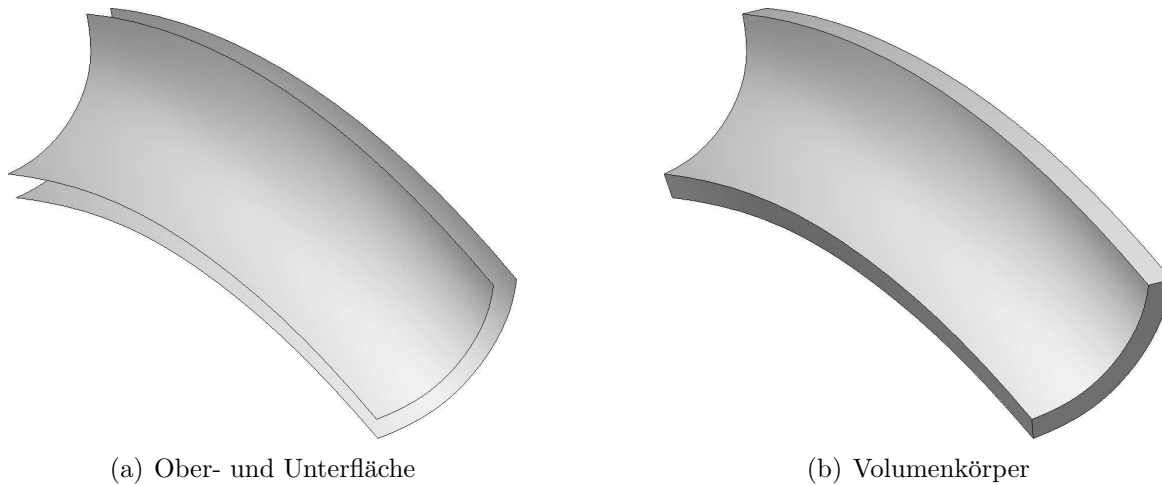


Abbildung 5.11: Erzeugung eines Volumenkörpers durch Skinning

5.3 Automatische Netzgenerierung für die p -Version der Finite-Elemente-Methode

Bei der p -Version der Finite-Elemente-Methode wird die Anzahl der Freiheitsgrade des resultierenden Gleichungssystems durch die Unterteilung des Berechnungsgebietes sowie dem Polynomgrad der verwendeten Ansatzfunktionen beeinflusst. Der vorliegende Abschnitt beschreibt nun die Erzeugung eines für die p -Version geeigneten Finite-Element-Netzes.

Prinzipiell wird zwischen Makronetzgeneratoren und Freivernetzern unterschieden. Erstere sind auf ZIENKIEWICZ [84] zurückzuführen und haben seitdem diverse Erweiterungen erfahren. Makrogeneratoren sind als alleinige Netzgenerierungstechnik für allgemeine Geometrien im Automobilbau aufgrund der Komplexität der zu diskretisierenden Flächen nicht geeignet. Bei Freivernetzern wird unterschieden zwischen verschiedenen Grundtechniken, wie der Delaunay-Triangulierung, welche auf der Idee des Voronoi-Diagramms beruht [27], der Quadtree-Methode, bei der das Berechnungsgebiet in strukturierte Zellen geteilt und anschließend vernetzt wird [43], der Advancing-Front-Methode [48, 55, 54] und der Gebietsteilungstechnik, welche im nachfolgenden Kapitel beschrieben wird.

Bei allen genannten Verfahren handelt es sich um Diskretisierungsmethoden für ebene, zweidimensionale Gebiete, welche schließlich zu Scheibenelementen führen. Die für diese Arbeit notwendige Diskretisierung mit Volumenelementen kann durch eine Extrusion der Schalenelemente zu Hexaederelementen erreicht werden [67]. Da es beim Umformprozess aber üblicherweise zu einer Ausdünnung des Bleches kommt, müssen auch die Hexaederelemente eine variable Dicke besitzen können. In Kapitel 5.3.6 wird ein erweitertes Verfahren beschrieben, das es ermöglicht, die Elementkanten der automatisch generierten Schalenelemente auf die Berandung des rückgeführten, dreidimensionalen Volumenkörpers (siehe Kapitel 5.2.6) zu projizieren und somit Hexaederelemente zu generieren.

5.3.1 Gebietsteilungstechnik

Der in *Mesh & Marry* eingebettete Netzgenerator DO_MESH arbeitet nach der Gebietsteilungstechnik. Die folgende Zusammenfassung der Funktionsweise des Netzgenerators bezieht sich auf [70]. Zunächst wird das Verfahren für ebene, zweidimensionale Gebiete erläutert. Die Vernetzung gekrümmter Flächen wird in Kapitel 5.3.4 dargestellt. Bei einem polygonal berandeten Berechnungsgebiet werden die Randkanten durch ihre Anfangs- und Endknoten definiert. Jeder Knoten besitzt einen so genannten h -Wert, welcher den lokalen Abstand zum nächstgelegenen Knoten angibt. Auf den Randkanten des Berechnungsgebietes werden entsprechend des h -Wertes gegebenenfalls neue Knoten erzeugt (siehe Abbildung 5.12).

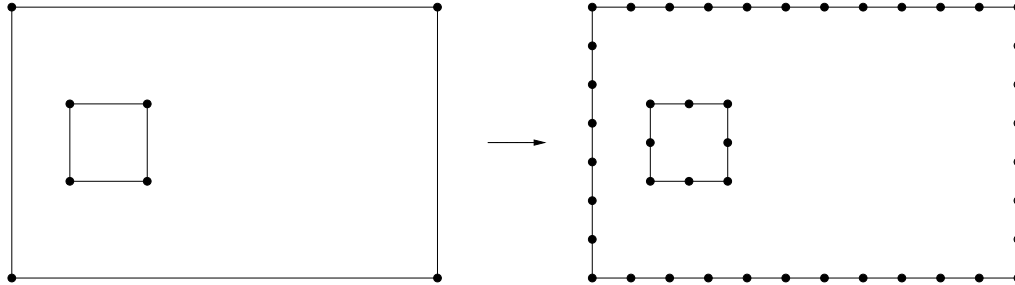


Abbildung 5.12: Teilung der Randkanten entsprechend h -Wert

Zunächst wird überprüft, ob sich durch Verbindung zweier Randknoten ein Dreieck aus dem Gebiet abspalten lässt. Dabei muss darauf geachtet werden, dass der Winkel zwischen Randkante und Schnittgerade in einem bestimmten Bereich liegt und der Abstand eines neuen Knotens auf der Teilungskante und eines bestehenden Randknotens nicht zu klein wird, um möglichst gleichseitige Dreieckselemente zu generieren [6, 66]. Kann ein solches Dreieck nicht erzeugt werden, so wird von einem Randknoten aus ein gerader Schnitt des Gebietes gesucht und das Gebiet in zwei Teilgebiete unterteilt. Dabei müssen die auftretenden Schnittwinkel folgende Gütekriterien erfüllen (siehe Abbildung 5.13):

$$\alpha_1 > \frac{\pi}{4} \quad \alpha_2 > \frac{\pi}{4} \quad \alpha_3 < \frac{\pi}{4} \quad \alpha_4 < \frac{\pi}{4} \quad (5.5)$$

Die Teilung wird rekursiv so lange fortgesetzt, bis ein Gebiet in Teilgebiete zerlegt ist, welche jeweils nur noch aus drei Knoten bestehen (Dreieckselemente).

5.3.2 Konvertierung des Dreiecksnetzes in ein Vierecksnetz

Der für diese Arbeit gewählte Ansatz verwendet eine dreidimensionale p -Version mit Hexaederelementen, welche durch eine „Quasi“-Extrusion von vierknotigen Schalenelementen erzeugt werden. Dazu werden zunächst die nach dem in Kapitel 5.3.1 beschriebenen Verfahren erstellten Dreieckselemente zu Viereckselementen konvertiert. Dies geschieht in drei Schritten:

1. Zusammenfassen von vier Dreieckselementen, welche an einem gemeinsamen Knoten zu einem Viereck zusammentreffen und Unterteilen des Vierecks in vier Viereckselemente gemäß Abbildung 5.14 links.

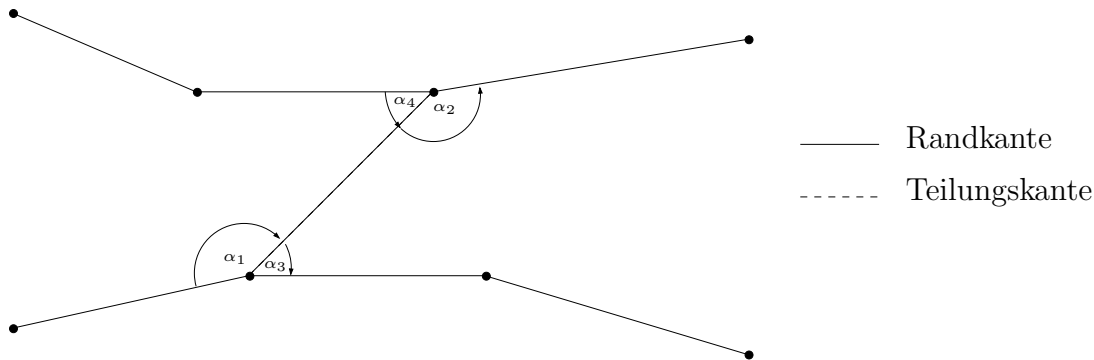


Abbildung 5.13: Auftretende Winkel an einer Teilungskante

2. Zusammenfassen zweier, durch eine gemeinsame Kante benachbarte Dreiecke zu einem Viereck und Zerlegung des Vierecks in vier Viereckselemente gemäß Abbildung 5.14 Mitte.
3. Auflösen der so genannten isolierten Dreiecke. Da nicht alle Dreiecke die Voraussetzung der ersten beiden Schritte erfüllen, müssen diese im letzten Schritt wie in Abbildung 5.14 rechts dargestellt in drei Vierecke unterteilt werden.

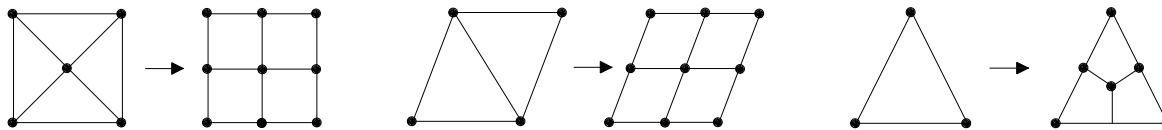


Abbildung 5.14: Schritte der Dreiecksnetzkonvertierung in ein reines Vierecksnetz

Durch die bei der Konvertierung konsequent durchgeführte Unterteilung der Elementkanten wird ferner die Kompatibilität des Netzes gewährleistet. Um ein möglichst unverzerrtes Netz mit annähernd isotropen, rechtwinkligen Viereckselemente zu erhalten ist es wichtig, die oben genannte Reihenfolge, mit der im zweiten Schritt benachbarte Dreieckselemente zusammengefasst werden, einzuhalten. Dabei kommt eine Liste zum Einsatz, in der alle möglichen Kombinationen gemäß ihrer Bewertung nach den von RANK [66] angegebenen Gütekriterien sortiert werden. In Abbildung 5.15 ist links das Dreiecksnetz und rechts das reine Vierecksnetz als Ergebnis der Konvertierung für eine Platte mit Loch dargestellt.

Üblicherweise werden mehrere Regionen sequentiell diskretisiert. Um kompatible Netzübergänge zu gewährleisten, werden die auf dem Strukturrand benötigten Knoten schon vor der Gebietsvernetzung definiert bzw. über den h -Wert festgelegt.

5.3.3 Adaptive Netzgenerierung

Die bisher vorgestellte Methode der rekursiven Gebietsteilung ermöglicht eine Diskretisierung mit einem äquidistanten Knotenabstand auf den Strukturrändern. Verfeinerungen lassen sich

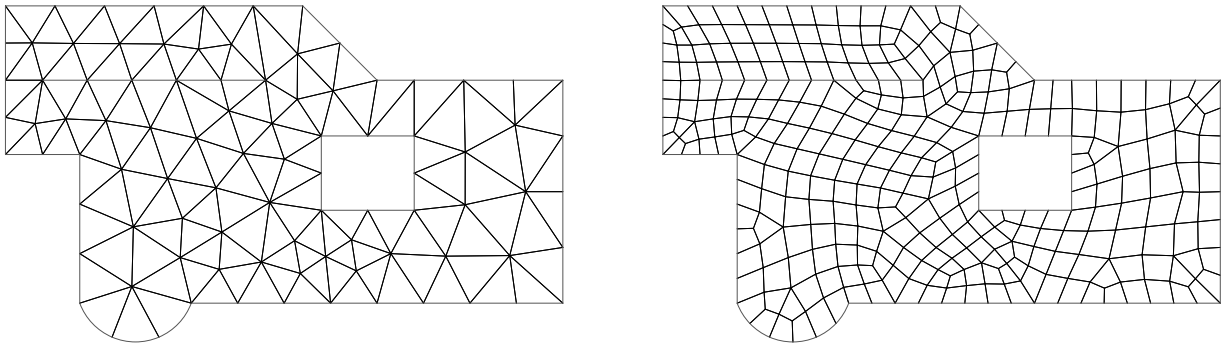


Abbildung 5.15: Dreiecksnetz mit 91 Knoten, 138 Elemente (links) und Vierecksnetz mit 339 Knoten, 295 Elemente (rechts)

dabei nur zu Strukturkanten bzw. -knoten erzeugen (siehe Abbildung 5.16). Um die Verfeinerung des Netzes im Gebiet zu variieren wurde von SCHWEINGRUBER-STRATEN [70] die Technik der adaptiven Netzgenerierung entwickelt. Dabei wird die Netzfeinheit über eine auf einem Hintergrundnetz definierte Dichtefunktion beeinflusst.

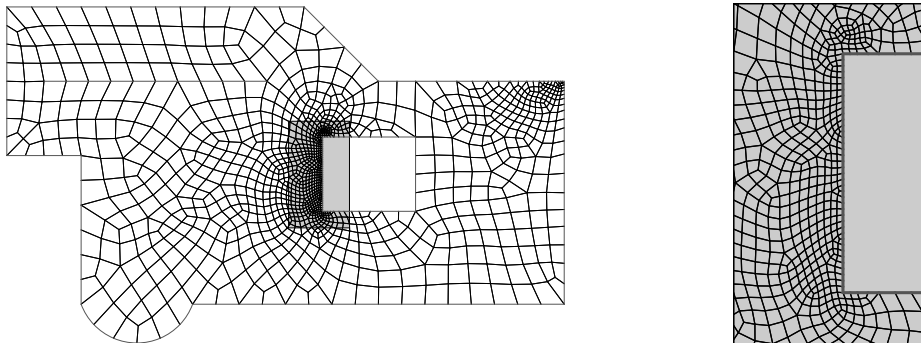


Abbildung 5.16: Verfeinerung durch Vorgabe von Knotenabstandswerten an Strukturknoten, Gesamtstruktur (links) und gezoomter Ausschnitt (rechts)

Bei dem in Abbildung 5.17 dargestellten Netz sind an Ecken, Zwangskanten und -knoten Spannungssingularitäten zu erwarten, welche eine Verfeinerung der Diskretisierung erfordern. Innerhalb des Gebietes liegt ein glatter Lösungsverlauf vor, für den eine Verfeinerung nicht erforderlich ist. Um nun die für dieses dargestellte Problem optimale Diskretisierung mit einer lokalen Verfeinerung zu generieren, bedarf es der adaptiven Netzgenerierung.

Dabei wird zunächst ein Ausgangsnetz mit einer dem verfügbaren Speicher entsprechenden Elementanzahl mithilfe der Methode der rekursiven Gebietsteilungstechnik generiert. Den generierten Knoten werden nun Knotenabstandswerte h über eine Dichtefunktion zugewiesen.

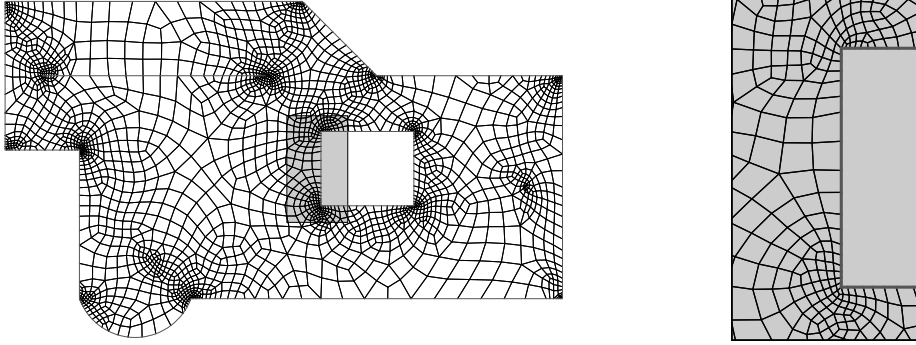


Abbildung 5.17: Verfeinerung mittels einer auf einem Hintergrundnetz definierten Dichtefunktion, Gesamtstruktur (links) und gezoomter Ausschnitt (rechts)

Diese Dichtefunktion kann dabei entweder *a priori* an Orten mit zu erwarteten Lösungssingularitäten manuell oder *a posteriori* automatisch aus Ergebnissen einer Fehlerschätzung definiert werden.

Da die Dichtefunktion Sprünge aufweisen kann, muss die Verträglichkeit der h -Werte untersucht und die Funktion entsprechend geglättet werden, um verzerrte Elemente zu vermeiden. Dazu werden zunächst die Knoten entsprechend ihrer h -Werte in eine Liste sortiert. Beginnend mit dem Knoten mit dem kleinsten h -Wert werden nun für jeden Knoten alle Verbindungskanten zu diesem Knoten bestimmt. Dabei wird für jede Kante der Länge l_{el} der Progressionsfaktor q ermittelt, mit dem der h -Wert des Nachbarknotens erreicht werden kann. Ist dabei $q > q_{\max}$, wobei q_{\max} der für eine bestimmte Diskretisierungsgüte maximal zulässige Progressionsfaktor ist, so wird der h -Wert des Nachbarknotens entsprechend herabgesetzt. Mit diesem Vorgehen wird ein geglätteter Übergang von feinen auf gröbere Bereiche ermöglicht.

Die vorgegebenen Knotenabstandswerte, welche zur Erstellung des in Abbildung 5.17 dargestellten Netzes verwendet wurden, sind in Abbildung 5.18 aufgetragen.

Die Erzeugung der im allgemeinen nicht-äquidistanten Teilungskanten erfolgt nun unter Berücksichtigung der auf dem Hintergrundnetz definierten Verteilung der Knotenabstandswerte. Die für die jeweilige Teilungskante zugehörigen h -Werte werden über Schnittpunkte der Teilungskante mit den Elementkanten des in Abbildung 5.19 dargestellten Hintergrundnetzes mithilfe des in KRAUSE & RANK [45] beschriebenen Quadtree-Algorithmus zur Elementsuche ermittelt. Dabei wird nur ein Element des Hintergrundnetzes gesucht in dem der Anfangs- oder Endpunkt der Teilungskante liegt. Die maßgebenden Schnittpunkte zwischen Teilungskante und Elementen des Hintergrundnetzes können gemäß Abbildung 5.19 über die Nachbarschaftsbeziehungen der Elemente gefunden werden.

Eine detaillierte Darstellung des Vorgehens zur adaptiven Netzerstellung wird in SCHWEIN-GRUBER-STRATEN [70] gegeben.

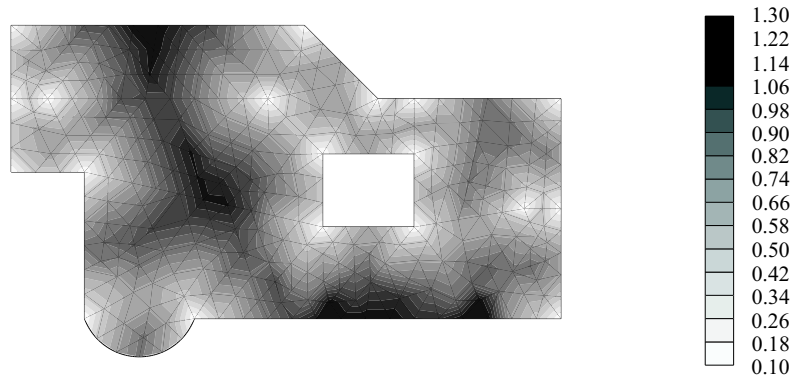


Abbildung 5.18: Knotenabstandswerte der Dichtefunktion

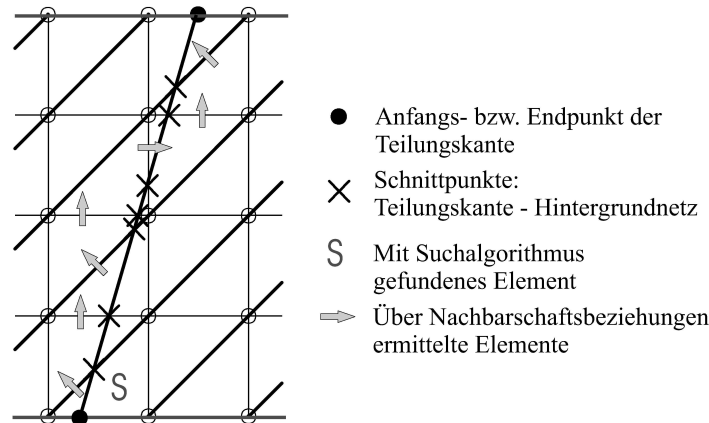


Abbildung 5.19: Schnittpunkte zwischen der Teilungskante und dem Hintergrundnetz

Abschließend sei noch angemerkt, dass die adaptive Netzgenerierung für Diskretisierungen mit der p -Version für die Berechnung der elastischen Rückfederung nicht die Regel ist. Der Grund dafür liegt in der zu erwartenden glatten Lösung und der implizit über die Partitionierung in Regionen festgelegte Elementgröße. Jedoch kann diese Art der Netzgenerierung bei komplexen Geometrien oder in Hinblick auf die Pressenkraftermittlung entscheidende Vorteile bieten. Hier sollte aufgrund der statisch unbestimmten Lagerung eine Verfeinerung der Diskretisierung hin zu den damit verbundenen Singularitäten durchgeführt werden.

5.3.4 Netzgenerierung auf Parameterflächen

Bei DO_MESH handelt es sich um einen 2D-Netzgenerator. Um Finite-Elemente-Diskretisierungen für Oberflächen im euklidischen Raum \mathbb{R}^3 zu generieren, werden die Elemente zunächst in der Parameterebene erzeugt und die Knotenpunkte über Abbildungsfunktionen in den dreidimensionalen Raum abgebildet. Dazu ist es notwendig, eine Kopplung zwischen dem Netzgenerator und der Applikation, welche die Geometrie vorhält, herzustellen. Eine Möglichkeit ist es, Netzgenerator und Geometriemodellierer programmtechnisch getrennt zu halten. Der Datenaustausch bei einem solchen Verfahren würde dann z. B. Datei-basiert erfolgen. Allerdings

stellt sich hier die Frage, welche geometrischen Informationen zur Verfügung gestellt werden müssen, um ein verzerrungsfreies Netz zu erhalten. Gewöhnlich ist dazu eine bidirektionale Kommunikation zwischen Geometriemodellierer und Netzgenerator notwendig. Da sich diese Vorgehensweise als ineffizient darstellt, wird sie hier nicht weiter verfolgt.

Eine weitere Möglichkeit wäre, Modellierungsprogramm und Netzgenerator über eine Interprozesskommunikations-Schnittstelle zu koppeln. Dabei würden beide Programme jeweils einen eigenen Prozess belegen und über eine Kommunikations-Schnittstelle Nachrichten austauschen. Dies könnte über eine Standard-Kommunikationssoftware wie z. B. MPI [25] (zeitaufwändig) oder Sockets geschehen [70]. Allerdings bringen solche Konzepte zusätzliche Komplexität mit sich, welche der Kompatibilität und Portierbarkeit entgegenstehen.

Die in Bezug auf Geschwindigkeit und Wartung optimale Variante ist die direkte Integration des Netzgenerators in den Modellierer. Hierbei ist es wichtig, einheitliche Schnittstellen in Form von Routinen zu definieren, um z. B. den Netzgenerator in mehrere Modellierer einzubetten. Ein Problem wäre hierbei z. B. eine Änderung der Datenstruktur im Modellierer, welche eine Änderung eines Funktionsaufrufs des Netzgenerators und damit eine Änderung der Funktion des Netzgenerators nach sich ziehen würde. Diese Änderung müsste dann folglich für alle Netzgeneratoren, welche diese Funktion verwenden, durchgeführt werden. Eine Abhilfe schaffen so genannte *Wrapper-Funktionen*. Dazu werden neue Funktionen definiert, welche eine Art Zwischenschicht darstellen und wiederum Kommunikations-Funktionen aufrufen. Veränderungen im Modellierer können somit in den Wrapper-Funktionen abgefangen werden und müssen nicht im Netzgenerator berücksichtigt werden. Ein weiterer Vorteil liegt in der Abwärtskompatibilität. Hat eine neue Version des Netzgenerators z. B. ein zusätzliches optionales Argument im Funktionsaufruf, kann dies durch einfache Erweiterung der entsprechenden Wrapper-Funktion um eine statische Variable berücksichtigt werden, ohne Veränderungen im Modellierer vorzunehmen.

Im vorliegenden Fall mussten zusätzlich unterschiedliche Programmiersprachen, in denen DO-MESH und *Mesh & Marry* implementiert wurden, berücksichtigt werden. Die DO-MESH-Routinen, welche in der Programmiersprache FORTRAN 77 umgesetzt wurden, haben einen prozeduralen Aufbau. *Mesh & Marry* dagegen wurde vollständig in der objektorientierten Sprache C++ entwickelt. Es ist nun notwendig alle Schnittstellen-Funktionen im gemeinsamen definierten Sprachraum zu implementieren. Diese Einschränkung betrifft im vorliegenden Fall dabei nur Funktionen, welche vom Modellierer zur Verfügung gestellt werden und durch den Netzgenerator aufgerufen werden. Explizit bedeutet dies, dass die betroffenen Funktionen mit der `extern "C"` Deklaration implementiert werden müssen. Dies schließt natürlich benutzerdefinierte Übergabeparameter wie Geometrieobjekte aus. Innerhalb des Funktionsrumpfes ist jedoch die objektorientierte Sprache C++ erlaubt. Um nun innerhalb einer solchen Funktion auf Geometrieobjekte zuzugreifen, müssen diese global zur Verfügung stehen, was eigentlich einem objektorientierten Konzept entgegensteht. Ein eleganter Weg zur Abhilfe dieser Unzulänglichkeit ist die Verwendung eines *Singleton-Pattern*. Das Singleton-Pattern beschreibt eine Klasse, welche nur einmal instanziiert werden kann, auf dessen einzige Instanz aber ein globaler Zugriff möglich ist. Eine umfassende Darstellung über Singleton-Patterns findet sich in [11, 29].

Weiterhin gilt es verschiedene Datenstrukturen mithilfe von Interfaces gegenseitig abzubilden. In DO_MESH werden die Netzobjekte in der Datenbank-ähnlichen Form *CDBASE* [72] gehalten. *Mesh & Marry* verwendet eine objektorientierte, hierarchische Struktur. Die Interface-Klasse enthält alle notwendigen Informationen, um aus der Datenbankabfrage das richtige Geometrieobjekt zu liefern.

5.3.4.1 Vernetzung gekrümmter Oberflächen

Im folgenden soll die Diskretisierung allgemeiner gekrümmter Oberflächen mithilfe des in Kapitel 5.3.4 vorgestellten Konzepts erläutert werden. Dabei werden die einzelnen Schritte zur Generierung eines Oberflächennetzes mit äquidistanten Knotenabstandswert diskutiert. Die Generierung eines lokal verfeinerten, adaptiven Netzes kann analog zum im Kapitel 5.3.3 beschriebenen Vorgehen durchgeführt werden.

Zunächst werden auf allen Strukturkanten der zu vernetzenden Regionen Knoten entsprechend der h -Werte erzeugt (siehe Abbildung 5.20). Diese Knoten liegen auf der definierenden Geometrie, d. h. werden mithilfe des Geometriemodellierers ermittelt. Dabei können sowohl analytische (z. B. Kreisbögen, Parabeln) als auch parametrische Kantenbeschreibungen (B-Spline, NURBS) verwendet werden.

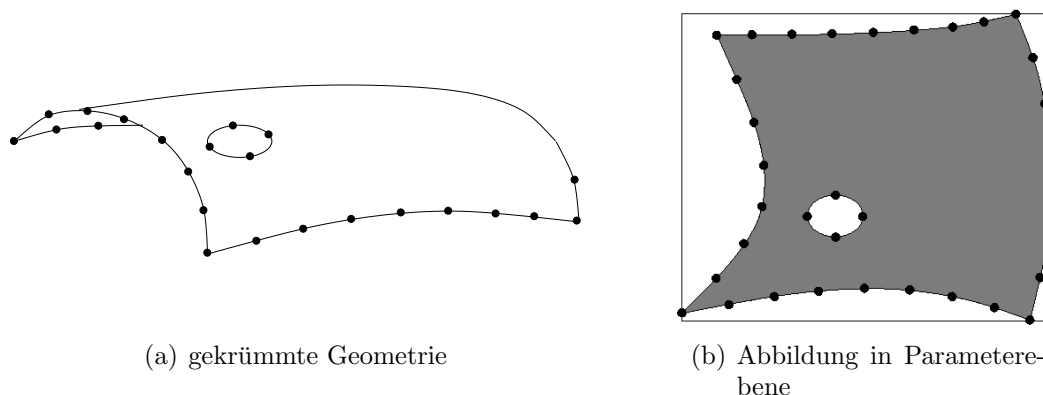


Abbildung 5.20: Randeinteilung der zu vernetzenden Geometrie (nach [70])

Die folgenden Schritte werden sequentiell für alle Regionen der zu diskretisierenden Struktur durchgeführt. Es werden dabei zunächst für alle Knoten i auf den Strukturkanten mit den globalen Koordinaten (x_i, y_i, z_i) die entsprechenden Koordinaten (u_i, v_i) der zu vernetzenden Parameterfläche mithilfe der Funktion

$$(u_i, v_i) = s(x_i, y_i, z_i) \quad (5.6)$$

bestimmt [70]. Die Region wird schließlich in der Parameterebene durch Polygone, welche durch die zuvor ermittelnden Knoten in der Parameterebene bestimmt sind, definiert. Beliebige, gekrümmte Segmente der Strukturkanten werden somit in der Parameterebene durch geradlinige Segmente abgebildet. In den folgenden Schritten wird gezeigt werden, dass damit kein Verlust geometrischer Informationen verbunden ist. Für die nun erzeugte zweidimensionale Region kann mithilfe der rekursiven Gebietsteilungstechnik ein Dreiecks- bzw. Vierecksnetz

erzeugt werden. Dabei dürfen auf den Regionskanten keine zusätzlichen Knoten erzeugt werden, um die Kompatibilität zu benachbarten Regionen zu gewährleisten. Schließlich wird das erzeugte Netz auf der Oberfläche erstellt. Dazu werden die Knoten der erstellten Elemente über die Umkehrabbildung

$$(x_i, y_i, z_i) = r(u_i, v_i) \quad (5.7)$$

auf die Oberfläche abgebildet. Die Geometrie der Elementkanten am Rand kann dabei durch die Geometrie der Randkanten beschrieben werden. Durch diese Vorgehensweise geht keine geometrische Information verloren. Im Inneren der Region ergibt sich zunächst ein bilineares Vierecksnetz, welches später den Ausgangspunkt für die Generierung des Hexaeder-Netzes darstellt. In Abhängigkeit der Parametrisierung der jeweiligen Regionen entsteht dabei ein mehr oder weniger verzerrtes Netz. Um die Güte des Netzes zu bestimmen und um gegebenenfalls Modifikationen durchzuführen werden zwei Kriterien geprüft:

1. Für die *Verzerrung* eines Elementes t mit dem Flächeninhalt a und den Seitenlängen s_1, s_2 und s_3 definiert sich die Güte $G(t)$ nach BANK [6] zu

$$G(t) = \frac{4\sqrt{3}a}{s_1^2 + s_2^2 + s_3^2}. \quad (5.8)$$

Ist $G(t) \leq \alpha$, so ist das Gütekriterium verletzt.

2. Ein weiteres Gütekriterium ist die *Elementgröße*. Ist dabei eine der Dreiecksseiten s_1, s_2 oder s_3 länger als $\beta \cdot h$, so gilt das Gütekriterium als verletzt.

Um ein weitgehend unverzerrtes Finite-Element-Netz zu erhalten wurden empirisch $\alpha = 0.85$ und $\beta = 1.5$ ermittelt. Wird eines der o. g. Gütekriterien verletzt, wird durch eine lokale Neuvernetzung nach der Delaunay-Technik [30] ein neuer Knoten in das Netz eingefügt.

5.3.5 Datenstruktur

Aus dem Geometriemodell in *Mesh & Marry* wird nun für DO_MESH eine Netzstruktur abgeleitet, welche aus

1. Regionen
2. Kanten
3. Knoten
4. Attributen
5. Löchern
6. Zwangslinien und -punkten

besteht. Die Regionen entsprechen im wesentlichen den projizierten Patches und erhalten als geometrische Information Ober-, Unter-, und Mittelfläche. Für die Schalenvernetzung mit DO_MESH wird zunächst die Mittelfläche benötigt. Dabei ist eine Region durch einen geschlossenen Kantenzug im Gegenuhrzeigersinn berandet. Kanten entsprechen im wesentlichen den Edges, es darf also für zwei benachbarte Regionen nur eine Kante erzeugt werden. Kanten erhalten außerdem Geometrieattribute, welche bei Teilung der Kante an die neu entstehenden Kanten vererbt werden. Das gleiche gilt für die Knoten. Attribute können Zwangspunkte oder die Geometrie definieren. Löcher werden definiert wie Regionen, mit dem Unterschied, dass sie durch einen Kantenzug im Uhrzeigersinn berandet sind.

5.3.6 Konvertierung des Vierecksnetzes in ein gekrümmtes Hexaedernetz

Das erzeugte bilineare Vierecksnetz wird nun topologisch auf ein Hexaedernetz erweitert und geometrisch über die berandenden Objekte attribuiert. Dieses Vorgehen wird vor allem durch das Streben nach einem modularen Aufbau der Diskretisierung motiviert. Durch den Zwischenschritt über ein bilineares Vierecksnetz ist es möglich, bei Bedarf eine Vielzahl an verfügbaren Schalenvernetzungsprogrammen zu verwenden.

Da die resultierenden Hexaederelemente eine variable Dicke und beliebige Oberflächengeometrie haben ist es nicht möglich, diese Elemente über eine gewöhnliche Extrusion der bilinearen Schalenelemente zu erzeugen. Vielmehr wurde eine Technik entwickelt, welche die Berandung auf die Ober- und Unterfläche projiziert (siehe Abbildung 5.21). Dabei werden in einem ersten

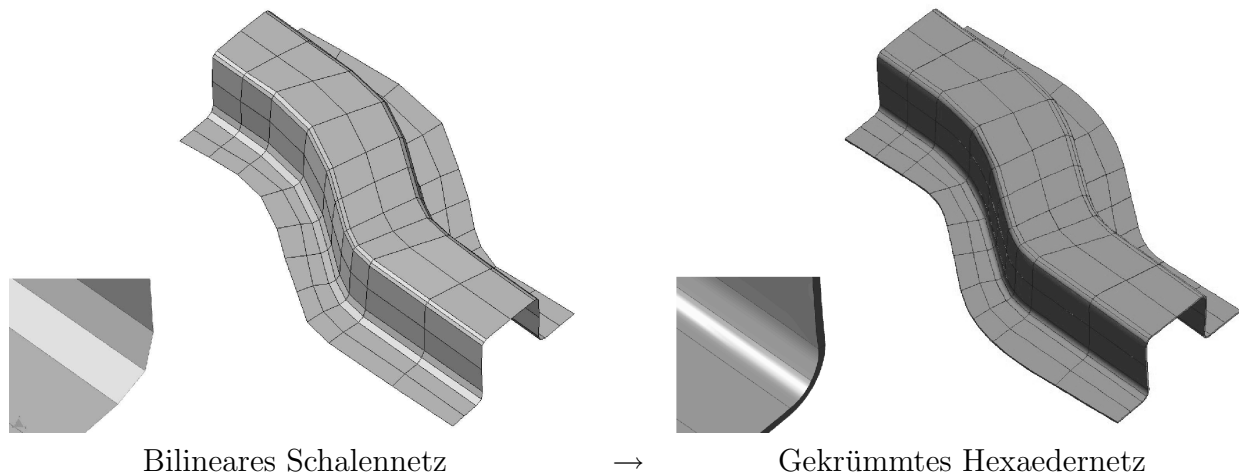


Abbildung 5.21: Diskretisierungsprozess

Schritt die Normalenvektoren an den Knoten des Schalennetzes bezüglich der Mittelfläche ermittelt. Für jede Kante wird nun über die beiden Normalenvektoren an den Randknoten eine bilineare Fläche aufgespannt. Über eine Boolesche Operation wird die Schnittkante dieser Schnittfläche jeweils mit Ober- und Unterfläche des Volumenkörpers bestimmt. In Abbildung 5.22 sind dabei \mathbf{n}_1 und \mathbf{n}_2 die beiden Normalenvektoren, welche die bilineare Schnittfläche

aufspannen, E_{lin} die ursprüngliche lineare Kante, F_o, F_u, F_m die Ober-, Unter- und Mittelfläche sowie E_o und E_u die durch die boolesche Operation gewonnenen gekrümmten Kantenbeschreibungen. Die lokalen Flächen 1 und 6 des Hexaederelements ermitteln sich somit über die Ober-

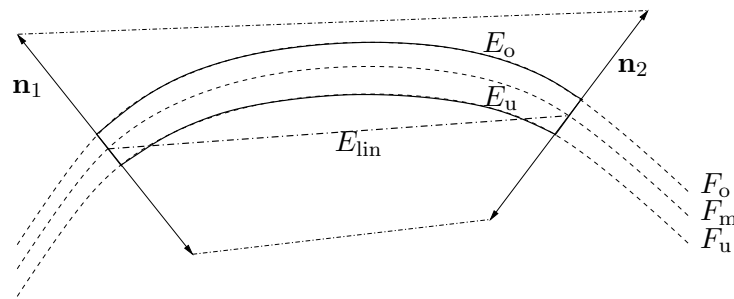


Abbildung 5.22: gekrümmte Kantenbeschreibung durch Verschneidung der bilinearen Schnittfläche mit Ober- und Unterseite

und Unterfläche des Volumenkörpers und den jeweiligen vier Schnittkanten als Berandung. Die vier Seitenflächen des Hexaederelements ergeben sich aus den bilinearen Flächen, den berandenden Normalenvektoren sowie jeweils einer Schnittkante auf Ober- und Unterfläche. Eine weitere Möglichkeit zur Gewinnung der gekrümmten Kantenbeschreibungen besteht darin, die linearen Kanten des Schalennetzes auf die Ober- und Unterfläche des Volumenkörpers normal zu projizieren. Dabei werden die lokalen Seitenflächen des erzeugten Hexaederelements über ein Skinning der jeweiligen Schnittkanten auf Ober- und Unterfläche erzeugt.

Im nächsten Schritt werden die für die quasi-regionale Abbildung erforderlichen Stützstellen des erzeugten Hexaederelements gemäß dem in Kapitel 4.6.2 beschriebenen Verfahren bestimmt.

5.4 Datentransfer

Im vorangegangenen Kapitel wurde die Rediskretisierung für die p -Version der Finite-Elemente-Methode mit Volumenelementen vorgestellt. Ausgangsobjekt war dabei ein bilineares Schalennetz. In vorliegendem Kapitel soll nun der Transfer der Geschichtsvariablen, in diesem Fall der Resteigenspannungen σ , welche die elastische Rückfederung hervorrufen, erläutert werden.

Drei wesentliche Unterschiede der Diskretisierungen für Umform- und Rückfederungsprozess machen eine Transformation der Komponenten des Spannungszustandes erforderlich. Zum einen müssen die Spannungskomponenten, welche von der Umformapplikation in lokaler Koordinatenrichtung des Elements ausgegeben werden in Richtung der globalen Koordinaten des Netzes für die p -Version transformiert werden. Desweiteren verwendet die Umformsimulation mit PAM-STAMP eine *Simpson*-Integration mit äquidistanten Integrationspunkten, *AdhoC*⁴ dagegen die *Gauß*-Integration mit nicht-äquidistanten Stützstellen. Darüber hinaus verändert sich die Lage der Integrationspunkte auch durch die Neuvernetzung. Werden Spannungskomponenten in einem eindimensionalen Feld abgelegt, so muss eine unterschiedliche Reihenfolge und Anzahl der Komponenten für *AdhoC*⁴ und dem für die Umformsimulation verwendeten Schalenprogramm berücksichtigt werden:

Komponente	<i>AdhoC</i>	Schale (PAM-STAMP)
1	σ_{xx}	σ_{xx}
2	σ_{yy}	σ_{yy}
3	σ_{zz}	σ_{xy}
4	σ_{xy}	σ_{xz}
5	σ_{yz}	σ_{yz}
6	σ_{xz}	

Tabelle 5.1: Reihenfolge der Komponenten im Vektor des Spannungszustands

5.4.1 Koordinatentransformation

Die Komponenten des Spannungstensors σ sind in der Umformapplikation PAM-STAMP auf das lokale Elementkoordinatensystem bezogen (siehe Abbildung 5.23). Für eine Berechnung der elastischen Rückfederung mit *AdhoC*⁴ müssen diese bezüglich des globalen Koordinatensystems ermittelt werden. Die globalen Komponenten lassen sich wie folgt bestimmen:

$$\sigma_g = \mathbf{T}^T \sigma_l \mathbf{T} \quad (5.9)$$

mit

$$\mathbf{T} = \begin{bmatrix} \mathbf{e}_1^T \\ \mathbf{e}_2^T \\ \mathbf{e}_3^T \end{bmatrix}. \quad (5.10)$$

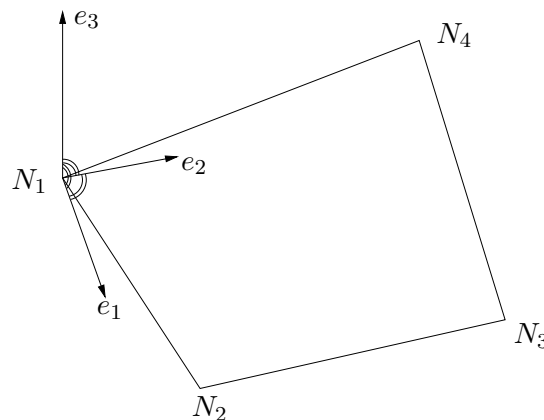


Abbildung 5.23: Lokales Elementkoordinatensystem (PAM-STAMP)

Die einzelnen orthogonalen Basisvektoren ermitteln sich nach [35] wie folgt:

$$\mathbf{e}_3 = \frac{\overrightarrow{N_1 N_3} \times \overrightarrow{N_2 N_4}}{|\overrightarrow{N_1 N_3} \times \overrightarrow{N_2 N_4}|} \quad (5.11)$$

Dabei ist \mathbf{e}_3 der Normalenvektor auf einer Durchschnittsebene durch die Punkte N_1 bis N_4 .

$$\mathbf{e}_1 = \frac{\overrightarrow{N_1 N_2} - (\mathbf{e}_3 \cdot \overrightarrow{N_1 N_2}) \mathbf{e}_3}{|\overrightarrow{N_1 N_2} - (\mathbf{e}_3 \cdot \overrightarrow{N_1 N_2}) \mathbf{e}_3|} \quad (5.12)$$

$\overrightarrow{N_1 N_2}$ repräsentiert den Kantenvektor zwischen Knoten N_1 und N_2 . Der Basisvektor in lokaler η -Richtung bestimmt sich zu

$$\mathbf{e}_2 = \mathbf{e}_3 \times \mathbf{e}_1. \quad (5.13)$$

5.4.2 Integrationspunkte

Für die p -Version der Finite-Elemente-Methode wird als numerisches Integrationsverfahren die Gauß-Integration verwendet. Dabei werden die globalen Koordinaten der Stützstellen mithilfe der quasi-regionalen Abbildung mit Gleichung (4.53) für Kanten bzw. Gleichung (4.54) für Flächen ermittelt. Die Koordinaten der Gaußpunkte im Inneren ergeben sich mithilfe der Blending-Funktionen-Methode (siehe Abbildung 5.24).

Es wurde ein Fehlermessverfahren implementiert, mit dem es möglich ist, die euklidische Norm des mithilfe der quasi-regionalen Abbildung interpolierten und des bezüglich der kontinuierlichen Geometrie exakten Gaußpunktes zu bestimmen. Die Maximumsnorm bezogen auf die euklidische Norm aller Gaußpunkte eines Elementes ist dabei ein Maß für die Güte der transfiniten Geometrieabbildung. Überschreitet die Maximumsnorm einen bestimmten Grenzwert, wird der Polynomgrad p_{geo} der Geometriebeschreibung erhöht.

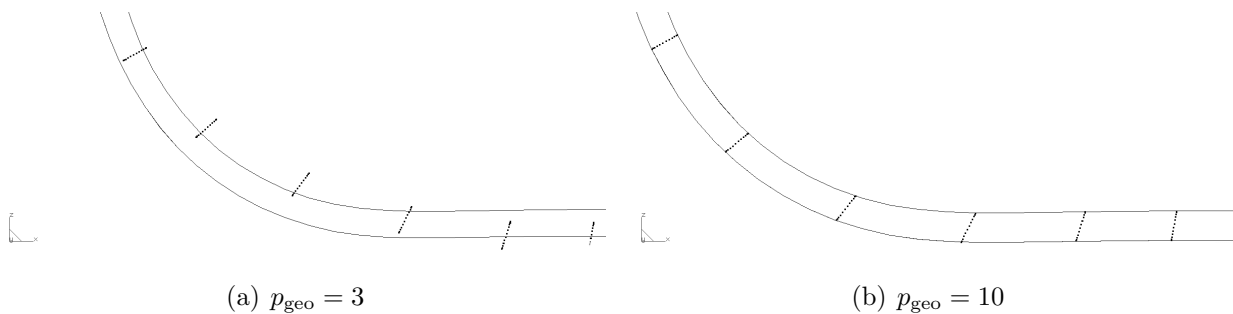


Abbildung 5.24: Gaußpunkte bei verschiedenen Geometriepolynomgraden p_{geo}

5.4.3 Aufbau eines Space-Tree

Grundlage aller Interpolationsverfahren ist es, zu einem Integrationspunkt des neuen p -Netzes die entsprechenden in der Nähe liegenden Integrationspunkte des h -Netzes zu finden und auf diese das Interpolationsverfahren anzuwenden. Denkbare Verfahren sind dabei

- Lineare Suche,
- Advancing-Front Suche,
- Quadtree-basierte Suche auf Parameterflächen sowie die
- Octree-basierte Suche im \mathbb{R}^3 .

Das in Bezug auf den Implementierungsaufwand einfachste Verfahren zur Suche eines Elementes für einen Integrationspunkt stellt die lineare Suche dar. Dabei wird für jedes Element des

Schalennetzes ein Punkt-im-Polygon-Test durchgeführt. Bei kleinen Problemen mit wenigen Elementen mag dieses Verfahren aufgrund seines geringen Aufwands geeignet erscheinen, bei Netzen mit einer großen Elementanzahl n bzw. Anzahl der zu bestimmenden Integrationspunkte g wird dieses Verfahren jedoch ineffizient, da der Aufwand zur Suche mit $O(n \cdot g)$ linear mit n bzw. g zunimmt.

Die Advancing-Front-Suche geht davon aus, dass sowohl das Schalennetz als auch die Integrationspunkte des Rückfederungsnetzes in einer Netzdatenstruktur beschrieben sind. Dabei werden die Nachbarknoten eines ersten gefundenen Knotens in einer Front zusammengefasst. Diese Knoten werden nun nacheinander abgearbeitet und Nachbarknoten des jeweiligen, untersuchten Knoten in die Front aufgenommen. Dies macht eine umfangreiche Suche erforderlich, da die beiden Diskretisierungen stark unterschiedliche Elementgrößen aufweisen und über Ausparungen im Inneren verfügen.

Eine weitere Möglichkeit der Suche ist die Anwendung eines Space-Tree gekoppelt mit einer lokalen Suche über Nachbarschaftsbeziehungen. Eine auf den ersten Blick in Bezug auf Speicherbedarf und Zeitkomplexität der Suche effiziente Datenstruktur ist eine Quadtree-basierte Suche auf Parameterebene. Den Integrationspunkten und Elementknoten müssen zunächst die Parameterkoordinaten der jeweiligen geometrischen Mittelebene zugeordnet werden. Dies stellt jedoch aufgrund der inversen Abbildung je nach Komplexität der Fläche eine mehr oder weniger zeitaufwändige Operation dar. Desweiteren würde dieses Verfahren nur mit einer einzigen Parametrisierung funktionieren, was der Segmentierung in Regionen mit eigenen Geometriebeschreibungen entgegen steht. Im folgenden wird daher eine Octree-basierte Suche vorgestellt, welche eine Erweiterung des Ansatzes von KRAUSE & RANK [45] und HALFMANN [34] darstellt.

Die Datenstruktur des Baumes umschreibt das Schalennetz durch einen Oktanten. Dieses Vatererelement wird nun rekursiv jeweils in acht gleich große Kindelemente so lange unterteilt, bis sich in einem Oktanten der Mittelpunkt der Integrationspunkteteile von genau einem Schalenelement befindet. Die dabei erzeugte Datenstruktur wird als Baumdatenstruktur gemäß Abbildung 5.25 abgespeichert. Jeder Oktant erhält als Information seine räumliche Ausdehnung und Lage über zwei gegenüberliegende Eck-Koordinaten. Dadurch ergibt sich ein einfaches Kriterium, ob der später zu suchende Punkt innerhalb des Oktanten liegt. Als topologische Information wird weiterhin ein Zeiger auf das Vatererelement des aktuellen Elements sowie acht Zeiger auf die Kindelemente gespeichert. Umschreibt ein Oktant keinen Integrationspunkt, so wird der entsprechende Zeiger auf das Kindelement zu Null gesetzt. Der ursprüngliche Oktant, der das gesamte Gebiet umfasst, ist das Wurzelement und besitzt als einziger Oktant kein Vatererelement. Die Zeiger auf die Schalenelemente werden in der Netzklasse in einem Vektor gespeichert. Beim Aufbau des Oktalbaums wird nun dieser Vektor kopiert. Innerhalb des kopierten Vektors werden für alle Kindelemente die Zeiger so sortiert, dass alle Elementzeiger eines Kindelementes hintereinander liegen. Durch dieses Vorgehen benötigt ein Kindelement als einzige Information den Zeiger des ersten Elements sowie die Anzahl der im Kindelement liegenden Elemente, um auf alle Elemente zugreifen zu können. Die Zeitkomplexität für das beim Aufbau des Oktalbaums nötige Umsortieren ist dabei von der Ordnung $O(n \log(n))$.

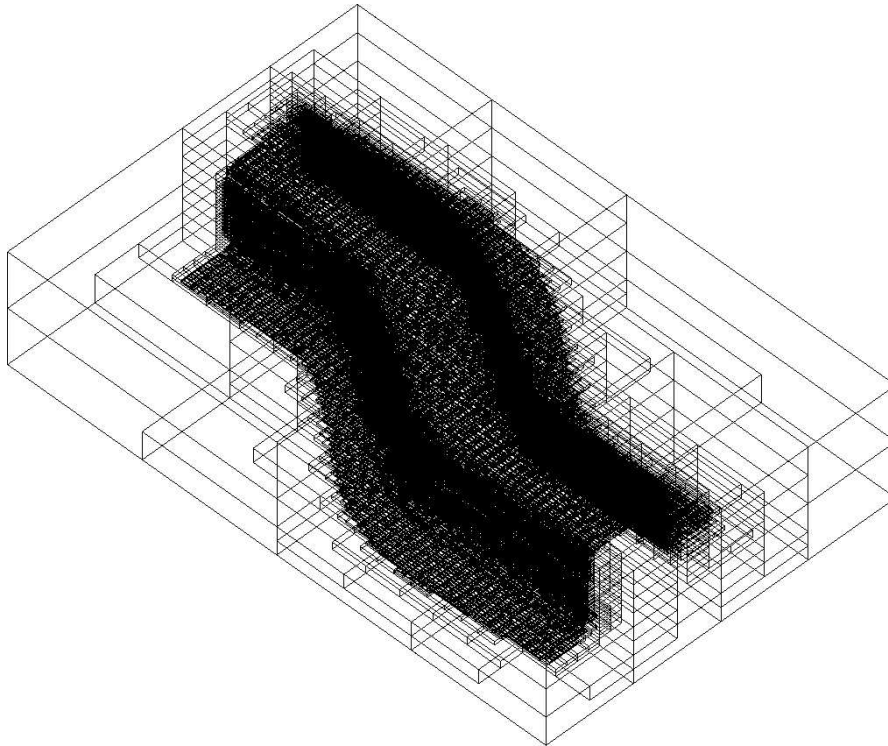


Abbildung 5.25: Oktalbaum für S-Rail-Schalennetz

Im nächsten Schritt wird mithilfe der Parameter-Koordinaten des Gaußpunktes der Baumknoten lokalisiert, der diesen Punkt beinhaltet. Falls dieser Knoten Kindelemente besitzt, wird die darunter liegende Ebene betrachtet. Sobald ein Baumknoten keine untergeordnete Ebene mehr besitzt, wird je nach Interpolationsverfahren eine Suchwelle über die benachbarten Schalenelemente gestartet. Der Vorteil dieses Verfahrens ist, dass die Suche nach den für die Interpolation relevanten Punkten nur über einige wenige Schalenelemente erforderlich ist.

5.4.4 Effiziente Punktsuche

Im folgenden soll einem Integrationspunkt des Volumennetzes ein Element des dimensionsreduzierten Schalennetzes zugeordnet werden. Da das Schalennetz keine Ausdehnung in lokaler Dickenrichtung besitzt, genügt es für alle Integrationspunkte mit $\xi = \eta = \text{const.}$ die Suche nur einmal durchzuführen. Die Suche eines dem Integrationspunkt zugeordneten Schalenelements läuft dabei in zwei Schritten ab:

1. Zunächst wird aufgrund einfacher Vergleichsoperationen der in den jeweiligen Oktanten gespeicherten Koordinaten rekursiv ermittelt, in welchem Oktant der gesuchte Integrationspunkt liegt. Das im Blatt-Oktanten liegende Element wird als Initial-Element für den zweiten Schritt zurückgeliefert.
2. Das mithilfe des Octrees ermittelte Element umschreibt nicht notwendigerweise den gesuchten Punkt, da auch andere Elemente, deren Schwerpunkt nicht im Oktanten liegen, von diesem geschnitten werden können. Daher wird nun zunächst für das Initialelement ein Punkt-im-Polygon-Test [45] durchgeführt. Liegt die Projektion des Punktes im In-

neren, war die Suche erfolgreich und wird beendet. Liegt sie außerhalb des Elements, so wird die Suche mithilfe der Nachbarschaftsbeziehungen in der Umgebung des Initialelements fortgesetzt.

Die Erweiterung des Punkt-im-Polygon-Tests soll im folgenden näher beschrieben werden. Neben der Information, dass sich der projizierte Integrationspunkt innerhalb des Elementes befindet, werden für die Interpolationsalgorithmen in der Regel die lokalen Flächenkoordinaten ξ und η benötigt. Die Projektionsrichtung \mathbf{p} des Punktes \mathbf{G} ergibt aus einer Geraden durch die Integrationspunkte mit $\xi = \eta = \text{const.}$ Zusammen mit der Projektionsrichtung \mathbf{p} kann für den Punkt \mathbf{G} eine Geradengleichung $\mathbf{g}(\lambda) = \mathbf{G} + \lambda\mathbf{p}$ aufgestellt werden. Der gesuchte Schnittpunkt ergibt sich durch Lösen eines bestimmten Gleichungssystems

$$\begin{aligned} \frac{1}{4}((1 - \xi)(1 - \eta)\mathbf{X}_1 + (1 + \xi)(1 - \eta)\mathbf{X}_2 + (1 + \xi)(1 + \eta)\mathbf{X}_3 + (1 - \xi)(1 + \eta)\mathbf{X}_4) = \\ = \mathbf{G} + \lambda\mathbf{p} \end{aligned} \quad (5.14)$$

wobei $\mathbf{X}_i \in \mathbb{R}^3, i = 1, \dots, 4$ die Koordinaten der Elementeckpunkte sind. Aus Effizienzgründen wird zur Lösung dazu ein Newton-Raphson-Verfahren [28] verwendet, welches bei den untersuchten Beispielen mit jeweils ca. 3 Iterationsschritten konvergiert.

5.4.5 Horizontale Interpolation

Gewöhnlich müssen aufgrund der Rediskretisierung oder auch bei gleicher Diskretisierung durch die unterschiedliche Lage der Integrationsstützstellen der Spannungszustand in der lokalen horizontalen Ebene der dünnwandigen Elemente interpoliert werden, d.h in der $\xi - \eta$ -Ebene des Hexaeder-Elements. Die lokale ζ -Koordinate wird dabei zunächst von den entsprechenden interpolierten Simpson-Integrationspunkten beibehalten. Die Interpolation auf die entsprechenden Gaußpunktkoordinaten geschieht schließlich durch eine lineare Interpolation der entsprechenden interpolierten Simpson-Punkte über die Dicke des Elements (siehe dazu Kapitel 5.4.6).

5.4.5.1 Next-Neighbor-Interpolation

Ein gängiges Interpolationsverfahren stellt die so genannte Next-Neighbor-Interpolation dar [34]. Dabei wird einem mit dem in Kapitel 5.4.4 erläuterten Verfahren gefundenen Punkt der Spannungszustand des Simpson-Integrationspunktes aus dem Umform-Netz zugewiesen, der diesem Punkt geometrisch am nächsten liegt. Weiterhin ist es möglich, mehrere Nachbarpunkte des h -Netzes über eine Wichtung der Spannungskomponenten mit der euklidischen Norm zum Gauß-Integrationspunkt des p -Netzes zu berücksichtigen. Die Komponenten $\hat{\sigma}_{\text{GP},i}$ des Spannungszustandes $\hat{\boldsymbol{\sigma}}$ für einen in der $\xi - \eta$ -Ebene interpolierten Punkt ermitteln sich zu

$$\hat{\sigma}_{\text{GP},i} = \frac{\sum_{j=1}^n \frac{\sigma_{\text{SP},j,i}}{d_j}}{\sum_{j=1}^n \frac{1}{d_j}} \quad i = 1, \dots, 6 \quad (5.15)$$

wobei n die Anzahl der zu interpolierenden Simpson-Integrationspunkte $\sigma_{\text{SP}(j)}$ und d_j ihren jeweiligen Abstand zum zu interpolierenden Punkt bezeichnet.

5.4.5.2 Knotenpunktmitteilung

Im folgenden sei σ eine typische Spannungskomponente. Ein einfacher und häufig effektiver Weg zur Verbesserung von Spannungsergebnissen führt über die Extrapolation der berechneten Spannungskomponenten von den Integrationspunkten jedes Elements zum Knoten i . Bei den Ergebnissen der Umformsimulation handelt es sich dabei um eine konstante Extrapolation, da in der Ebene aufgrund der verwendeten reduzierten Integration nur eine Stützstelle vorhanden ist. Auf diese Weise liefert die Extrapolation für jede Spannungskomponente des Knotens genau so viele Werte, wie Elemente zu ihm benachbart sind. Der Mittelwert σ_m^i dieser n Werte wird als gemittelte Spannungskomponente im Knoten i genommen (Abbildung 5.26). Nachdem für

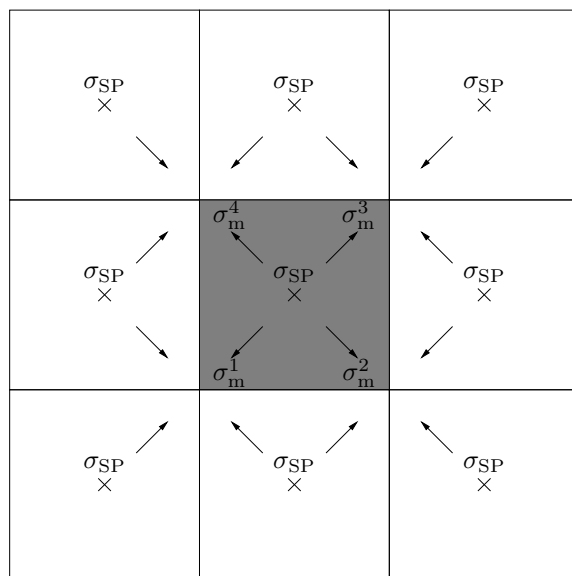


Abbildung 5.26: Mittelung der Spannungskomponenten zu Knoten

alle sechs Spannungskomponenten die Mittelung durchgeführt worden sind, ergibt sich der verbesserte Wert der Spannungskomponente σ innerhalb des Elements zu

$$\sigma_{\text{verb.}} = \sum_{i=1}^4 h_i \sigma_m^i \quad (5.16)$$

wobei h_i die Ergebnisse für die Verschiebungsinterpolationsfunktionen am gesuchten Punkt und somit

$$\hat{\sigma}_{\text{GP}} = \sigma_{\text{verb.}} \quad (5.17)$$

gilt. Ein weitere Möglichkeit σ_m^i für den Knoten i zu erhalten besteht darin, die Methode der kleinsten Quadrate auf die Werte in den Integrationspunkten anzuwenden [7]. Dabei werden der Knoten i selbst, die acht benachbarten Knoten sowie die dem Knoten i nächsten j Integrationspunkte betrachtet.

$$\frac{\partial}{\partial \sigma_m^i} \left[\sum_{j=1}^4 \left(\sigma_{\text{SP}}^j - \sum_{i=0}^9 h_i \sigma_m^i \right)^2 \right] = 0 \quad i = 1, \dots, 9 \quad (5.18)$$

minimiert den Fehler zwischen den gegebenen Komponenten in den Integrationspunkten und den aus der Interpolation der Knotenpunktwerte σ_m^i für die gleichen Koordinaten gewonnenen Werte. Gleichung (5.18) erfordert die Lösung eines Gleichungssystems mit neun Unbekannten. Die Methode der kleinsten Quadrate wird für die Interpolation der Ergebnisse der Tiefziehsimulation nicht weiter betrachtet, da das Verfahren mit der einfachen Mittelung adäquate Ergebnisse bei einer erheblich geringeren Rechenzeit liefert.

5.4.6 Vertikale Interpolation

Ein wesentlicher Unterschied zwischen den für Umform- und Rückfederungsberechnung verwendeten Integrationsverfahren ist die unterschiedliche Lage der Stützstellen. Während bei der Simpson-Integration eine äquidistante Stützstellenverteilung Anwendung findet, verwendet die Gauß-Integration ein nicht-äquidistantes Stützstellenschema mit unterschiedlicher Gewichtung der einzelnen Integrationspunkte (siehe Abbildung 5.4.6).

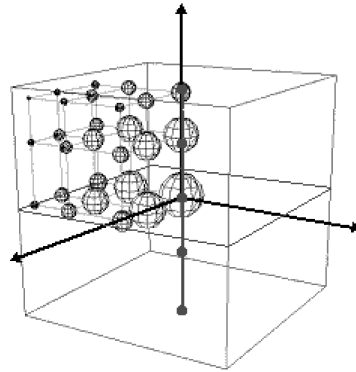


Abbildung 5.27: Stützstellen der Gauß- und Simpsonintegration

Die Simpson-Integration ist dabei ineffizienter als die Gauß-Integration, besitzt allerdings den Vorteil, dass die Integrationspunkte auch auf dem Rand liegen. Die horizontal mit dem Next-Neighbor- oder Knotenpunktmittelungs-Verfahren interpolierten Simpson-Integrationspunkte müssen nun in vertikaler Richtung für das Gauß-Stützstellenschema interpoliert werden. Numerische Untersuchungen haben gezeigt, dass die lineare Interpolation eine sehr gute Genauigkeit liefert. Dabei wird für $n_{\text{Gauß}} = n_{\text{Simpson}}$ eine genaue Interpolation erwartet. Vor allem in Bereichen mit hoher Krümmung ist eine maximale Genauigkeit erforderlich, da dort gewöhnlich der Verlauf der Spannungskomponenten sehr komplex werden kann. Aufgrund der Hebelwirkung können bereits kleine Fehler in der Interpolation der Spannungen zu großen Auswirkungen im Verschiebungsverlauf in anderen Bereichen des Bauteils führen. In ebenen Bereichen können die Verläufe der einzelnen Komponenten jedoch häufig durch einen linearen oder konstanten Verlauf approximiert werden. Um bei sehr feinen Diskretisierungen den Aufwand hinsichtlich Speicher- als auch Rechenzeitbedarf zu reduzieren wurde eine optionale adaptive Integration entwickelt. Dabei wird die für jedes Element für eine bestimmte Genauigkeit erforderliche Integrationsordnung bestimmt. Das Maß für die Genauigkeit ist dabei die

Differenz zwischen Simpsonintegral und Gaußintegral

$$\left| \sum_{i=0}^k |w_i \cdot f(x_i)| - \sum_{i=0}^{n_{\text{Simpson}}} |w_i \cdot f(x_i)| \right| \leq \varepsilon \quad (5.19)$$

wobei

- k die Anzahl der Gaußpunkte, Iterationsparameter,
- w den Wichtungsfaktor,
- $f(x)$ den Funktionswert am Gaußpunkt sowie
- ε die Fehlertoleranz bezeichnet.

Als Startwert wird der Parameter $k = n_{\text{Gauß}}$ gesetzt. k wird nun iterativ solange um 1 reduziert wie Gleichung (5.19) erfüllt ist (siehe Abbildung 5.28). Dabei ist k nach unten außerdem durch den Polynomgrad p_ζ der Finite-Element-Approximation begrenzt.

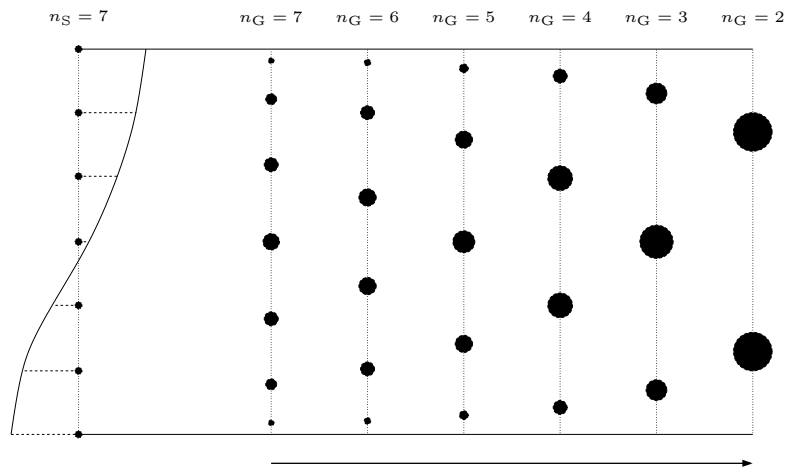


Abbildung 5.28: Adaptive Integration über Elementdicke

5.5 Effiziente Integrationsmethoden

Bei der Berechnung der Finiten Elemente sind Integrationen über den Elementbereich erforderlich. Die Komponenten der Steifigkeitsmatrix \mathbf{K} und des Lastvektors \mathbf{f} des Gleichungssystems

$$\mathbf{K}\mathbf{a} = \mathbf{f} \quad (5.20)$$

werden dabei durch eine numerische Integration bestimmt, da sie in der Regel kein Polynom sondern eine gebrochen rationale Funktion darstellen. Für die p -Version findet dabei aufgrund der Genauigkeit die Gauß-Integration Anwendung. Dabei wird später gezeigt werden, dass sich für den *trunk space* $\mathcal{S}_{\text{ts}}^{p_\xi, p_\eta, p_\zeta}(\Omega_{\text{st}}^h)$ bei $n_G \geq p + 1$ bzw. beim *tensor product space* $\mathcal{S}_{\text{ps}}^{p_\xi, p_\eta, p_\zeta}(\Omega_{\text{st}}^h)$ bei $n_G \geq p + 2$ die Genauigkeit der Lösung nicht weiter verbessert. Da die Resteingenspannungen auf die Stützstellen der Gauß-Integration interpoliert werden, kann auf der anderen

Seite für eine grobe Diskretisierung der Rückfederungsberechnung z. B. mit einem Elementverhältnis des Umformnetzes zum Rückfederungsnetzes > 100 eine hohe Integrationsordnung für das Lastintegral erforderlich sein. Ein effiziente Vorgehensweise zur Integration ist dabei die Anzahl der Integrationspunkte für Lastvektor und Elementsteifigkeitsmatrix unabhängig voneinander definieren zu können. Auf der Lastseite orientiert sich dabei die Ordnung an der zur Abbildung der Distribution des Spannungszustandes erforderlichen Genauigkeit, auf der Steifigkeitsseite am jeweiligen Polynomgrad und Ansatzraum. Da der numerische Aufwand für die Steifigkeitsmatrix von der Ordnung $O(n_G^2)$, der des Lastvektors dagegen von der Ordnung $O(n_G^3)$ ist, liegt der Effizienzgewinn vor allem in einer möglichst niedrigen Integrationsordnung der Steifigkeitsmatrix.

Weiterhin wurde eine anisotrope Integration implementiert. Dadurch ist es möglich, für die Distribution der Resteigenspannungen die Anzahl der Spannungstensoren für alle drei lokalen Koordinatenrichtungen unabhängig voneinander zu definieren. Als günstig hat es sich dabei erwiesen, über die Dicke die Anzahl der Gaußpunkte

$$n_\zeta = n_{\text{SP}} \quad (5.21)$$

gleich der Anzahl der ursprünglichen Simpsonpunkte n_{SP} zu setzen. In der Ebene wird in Hinblick auf eine möglichst genaue Abbildung der Distribution der Spannungszustände

$$n_\xi = n_\eta = \text{int} \left(\sqrt{\frac{n_U}{n_A}} \right) \quad (5.22)$$

gewählt, wobei n_U die Anzahl der Elemente des Umformnetzes und n_A die Anzahl der Elemente des Rückfederungsnetzes für die p -Version der Finite-Elemente-Methode ist.

5.6 Gleichungslöser

Wie in Kapitel 3.6 gezeigt wurde, erfordern nichtlineare Problemstellungen vom Typ

$$\mathbf{K}(\mathbf{a}) \mathbf{a} = \mathbf{f} \quad (5.23)$$

iterative Lösungsverfahren. Dabei werden gewöhnlich Newton-Raphson-Verfahren verwendet. Diese beinhalten den wiederholten Aufbau sowie Lösung eines linearen Gleichungssystems bis eine gewünschte Genauigkeit erreicht ist.

Globale Steifigkeitsmatrixen für diskrete, dreidimensionale Problemstellungen besitzen eine hohe Dimension, welche effiziente Lösungsstrategien erforderlich machen. Dabei weisen sie für elastisches Materialverhalten, wie es bei der Rückfederung vorliegt eine symmetrische Struktur auf. Ein entscheidender Aspekt in Hinblick auf große Gleichungssysteme ist dabei die Skalierbarkeit der Lösungsverfahren. Eine gute Skalierbarkeit bedeutet in diesem Zusammenhang, dass der Aufwand zur Lösung des Gleichungssystems nur linear mit der Anzahl der Unbekannten anwächst. Die auf einen lokalen Bereich beschränkte Kopplung der Freiheitsgrade führt gewöhnlich zu schwach besetzten Matrizen. Dabei besitzt die Anordnung der Freiheitsgrade entscheidenden Einfluss auf die Bandbreite des Gleichungssystems. Bei der p -Version ist diese Eigenschaft im Gegensatz zur h -Version weniger dominierend, da bei Polynomgraden

höherer Ordnung eine stärkere Kopplung der Elemente, insbesondere über die Kanten- und Flächenmoden, zu dichter besetzten Matrizen führt. Die Assemblierung der Elemente hat dabei entscheidenden Einfluss auf die Bandbreite der Steifigkeitsmatrix. Eine geeignete Nummerierung der Elemente führt zu einer schmalen, parallel ausgerichteten Bandstruktur mit einer geringen Anzahl von Nullelementen innerhalb des Bandes. Schließlich ist noch die Konditionierung des Gleichungssystems für die Konvergenz iterativer Gleichungslöser von entscheidender Bedeutung. Maßgebende Größe für die Konditionierung ist dabei die Konditionszahl

$$k(\mathbf{K}) = \frac{\lambda_{\max}}{\lambda_{\min}} \quad (5.24)$$

wobei λ_{\max} den größten, λ_{\min} den kleinsten Eigenwert der Matrix \mathbf{K} darstellt. Die Konditionierung hängt dabei von der Elementformulierung und der Assemblierung der Elemente ab. Für dünne Strukturen, wie sie bei der Blechumformung vorkommen, ergeben sich höhere Konditionszahlen als für dicke Strukturen, da die Eigenwerte und damit auch die Eigenvektoren in der Ebene und in Dickenrichtung aufgrund der unterschiedlichen Steifigkeit stark voneinander abweichen.

Die unterschiedlichen in *AdhoC*⁴ implementierten Arten von Gleichungslösern sollen nun diskutiert und verglichen werden.

5.6.1 Direkte Gleichungslöser

Gleichungslöser für Finite-Element-Probleme nutzen die schwache Besetztheit der globalen Steifigkeitsmatrix aus. In diesem Kontext bedeutet dies, dass nur die von Null verschiedenen Elemente in einem Band- oder Blockspeicherschema gespeichert werden. Ein großer Vorteil von direkten Methoden zur Lösung von Gleichungssystemen liegt in der Robustheit gegenüber schlecht konditionierten Problemen, wie sie z. B. bei dünnen Strukturen vorliegen. Nachteil direkter Gleichungslöser ist der hohe numerische Aufwand der Ordnung $O(nm^2)$, wobei n die Anzahl der Unbekannten und m die Bandbreite bezeichnet. Vor allem bei der p -Version macht sich dies bemerkbar, da eine Erhöhung der Freiheitsgrade hier zwangsläufig aufgrund der höheren Anzahl an gekoppelten Freiheitsgraden zu einer größeren Bandbreite führt. Weiterhin kommt es im Eliminationsprozess zu einem Auffüllen des Profils, so dass die Speicherung nicht mehr kompakt, d. h. ohne Nullelemente erfolgen kann. Vor allem für dreidimensionale Probleme macht sich diese Eigenschaft stark bemerkbar.

In den letzten zehn Jahren wurden Techniken entwickelt um die ungünstigen Eigenschaften des Speicherbedarfs sowie des numerischen Aufwands zu minimieren [4]. Die zugehörigen Implementierungen weisen einen hohen Grad an Komplexität auf [1], zeigen jedoch gerade für große dreidimensionale Probleme eine hohe Effizienz.

5.6.2 Iterative Gleichungslöser

Die bei der Finite-Elemente-Methode entstehenden Gleichungssysteme sind zwar schwach besetzt, besitzen jedoch eine hohe Dimension. Daher ist für große Problemstellungen eine Kompaktspeichertechnik erforderlich, welche sich allerdings nicht bei direkten Gleichungslösern

verwenden lässt, da diese keinen fill-in zulassen. Aus diesem Grund wird für große Gleichungssysteme diese Technik zusammen mit iterativen Lösern angewendet. Zwei dieser Verfahren sind z. B. das *Gauß-Seidel*- und das *Jacobi*-Verfahren. Diese besitzen allerdings nur ein langsames asymptotisches Konvergenzverhalten und verwenden keine Vorkonditionierung. Iterative Gleichungslöser bieten sich aufgrund des geringen Speicherbedarfs für große Systeme an. Allerdings arbeiten sie nur dann effizient, wenn sie einen für das zu lösende Problem angepassten Vorkonditionierer verwenden. Die Methode der konjugierten Gradienten (CG-Verfahren) für positiv definite Gleichungssysteme kann somit zur Methode der vorkonditionierten konjugierten Gradienten (PCG-Verfahren) erweitert werden. Dabei wird das zu lösende Gleichungssystem

$$\mathbf{K}\mathbf{a} + \mathbf{d} = \mathbf{0} \quad (5.25)$$

als ein äquivalentes Minimierungsproblem des quadratischen Funktionals

$$F(\mathbf{a}) = \frac{1}{2}\mathbf{a}^T\mathbf{K}\mathbf{a} + \mathbf{d}^T\mathbf{a} \quad (5.26)$$

betrachtet. Eine ausführliche Beschreibung leistungsfähiger CG-Verfahren findet sich z. B. in [68]. Die wesentliche Idee aller zur Vorkonditionierung entwickelten Verfahren liegt darin, das zu lösende System

$$\mathbf{K}\mathbf{a} + \mathbf{d} = \mathbf{0} \quad (5.27)$$

in ein äquivalentes System

$$\tilde{\mathbf{K}}\tilde{\mathbf{a}} + \tilde{\mathbf{d}} = \mathbf{0} \quad (5.28)$$

mit

$$\tilde{\mathbf{K}} = \mathbf{C}^{-1}\mathbf{K}\mathbf{C}^{-T}, \quad \tilde{\mathbf{a}} = \mathbf{C}^T\mathbf{a}, \quad \tilde{\mathbf{d}} = \mathbf{C}^{-1}\mathbf{d} \quad (5.29)$$

überzuführen, bei dem für die Konditionszahlen

$$k(\tilde{\mathbf{K}}) < k(\mathbf{K}) \quad (5.30)$$

gilt. Es ist wichtig, die Vorkonditionierung dem Problem entsprechend zu wählen, damit der zusätzliche Aufwand zur Vorkonditionierung und die Reduktion des numerischen Aufwands zur Lösung des Gleichungssystems in einem ausgewogenen Verhältnis stehen. Weiterhin wird in WALL ET AL. [81] ein Verfahren zur Konditionierung dünnwandiger dreidimensionaler Strukturen präsentiert, welches in Dickenrichtung eine Skalierung vornimmt. Dabei muss das konstitutive Gesetz entsprechend angepasst werden.

5.6.3 Anmerkungen zur Wahl des Gleichungslösers

Iterative Gleichungslöser haben in der Regel einen geringeren Speicherplatzbedarf. Allerdings erfordert die Vorkonditionierung ein genaue Kenntnis des zu berechnenden Problems, da im ungünstigsten Fall die Vorkonditionierung numerisch teurer werden kann, als der Gewinn bei der Reduzierung des Aufwands zur Lösung des Gesamtgleichungssystems. Direkte Lösungsverfahren besitzen den Vorteil, dass sie selbst bei schlecht konditionierten Problemen, wie sie bei dünnen Strukturen auftreten das Gleichungssystem wesentlich schneller lösen und damit iterativen Verfahren überlegen sind. Allerdings wird bei großen Systemen der Speicherplatzbedarf zum kritischen Faktor, so dass in diesen Fällen auf die iterativen Löser ausgewichen werden muss.

5.7 Ansatz zur Berechnung der erforderlichen Pressenkraft

Die in dieser Arbeit beschriebenen Verfahren beziehen sich auf die Berechnung der elastischen Rückfederung, um mithilfe dieser über Kompensationsverfahren wie z. B. der Verschiebungsanpassung oder der Spring-Forward-Methode (siehe Kapitel 2.2.2) die optimierte Werkzeugform zu finden. Eine weitere Problematik, welche sich vor allem in den letzten Jahren zeigt, liegt in der Dimensionierung der Pressenhydraulik. Vor allem durch den Einsatz hochfester Stähle oder Aluminium für die Werkstücke kommt es aufgrund der höheren Streckgrenze bzw. des höheren E-Moduls zu großen Kräften, welche das Werkzeug aufbringen muss. Aufgrund der langen Laufzeit einer Presse ist die richtige Dimensionierung von höchster Priorität. Der für die Bemessung maßgebende maximale Spannungszustand im Bauteil wird im geschlossenen Zustand der Presse, d. h. unmittelbar vor dem Herausfahren des Werkzeugs, erreicht. Bei diesem Spannungszustand handelt es sich um die Resteigenspannungen, welche auch für die elastische Rückfederung verantwortlich sind. Im Automobilbau übliche Pressen deformieren die Bauteile durch eine lineare Verschiebungssteuerung, gewöhnlich in Richtung der Schwerkraft (globale z -Richtung, siehe Abbildung 5.29). Es entstand die Idee, die erforderliche Pressenkraft durch Integration der Komponenten in z -Richtung des Spannungsvektors normal zur – in Kontakt mit dem Werkzeug stehenden – Oberflächen der jeweiligen Hexaederelemente zu berechnen.

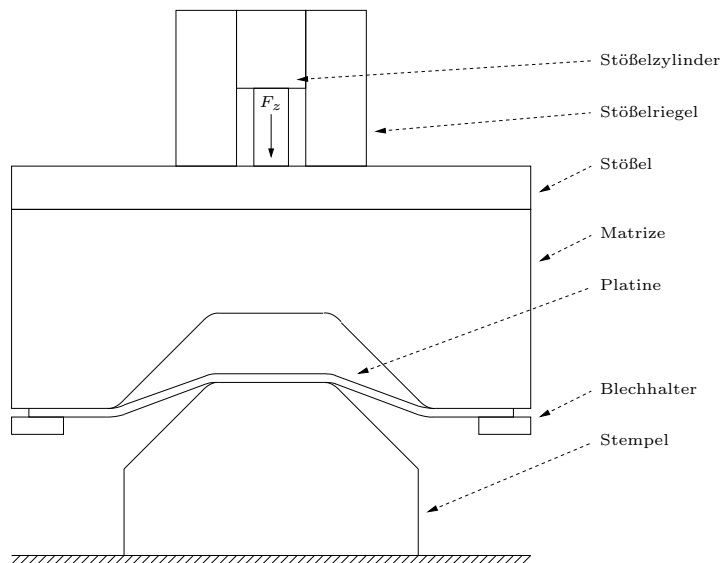


Abbildung 5.29: Ziehvorgang

5.7.1 Motivation

Die maßgebenden Kontaktkräfte treten im Bereich der Elemente auf, deren Normalenvektor der in Kontakt stehenden Oberflächen in globale z -Richtung zeigt. Den wesentlichen Beitrag zu dieser Kraft liefert die z -Komponente des Spannungsvektors normal zur Oberfläche und damit für den betrachteten maßgebenden Fall hauptsächlich die σ_{zz} -Komponente des Spannungszustandes σ . Da jedoch die für die Durchführung der Tiefzieh-Simulation verwendeten

Schalenelemente einen ebenen Spannungszustand annehmen, d. h. $\sigma_{zz} = 0$ gilt, sind diese zur direkten Berechnung der Kraftkomponenten nicht geeignet. Der folgende Ansatz beschreibt die Möglichkeit zur Berechnung der Pressenkraft mithilfe eines dreidimensionalen, auf Volumenelementen basierenden und aus einer Schalenlösung korrigierten Spannungszustandes [22, 53, 63]. Dabei erhebt das nachfolgende akademische Beispiel nicht den Anspruch der Abbildung einer realen Umformsimulation, sondern soll vielmehr die Überlegenheit des Spannungszustandes von Volumen- gegenüber Schalenelementen und die Möglichkeit der Korrektur des Spannungszustandes demonstrieren.

Im folgenden wird eine Platte mit den Abmessungen $l = b = 10 \text{ m}$, $h = 0.35 \text{ m}$ betrachtet. Die Struktur wird für die Schalenlösung des Initialschrittes mit 162 Quad-Elementen bzw. mit 162 Hexaederelementen für die Spannungskorrektur mit dreidimensionalen Volumenelementen diskretisiert (siehe Abbildung 5.30). Dabei wurde das Netz an den Rändern und dem Lastwechsel verfeinert um Randschichten und Singularitäten aufzulösen.

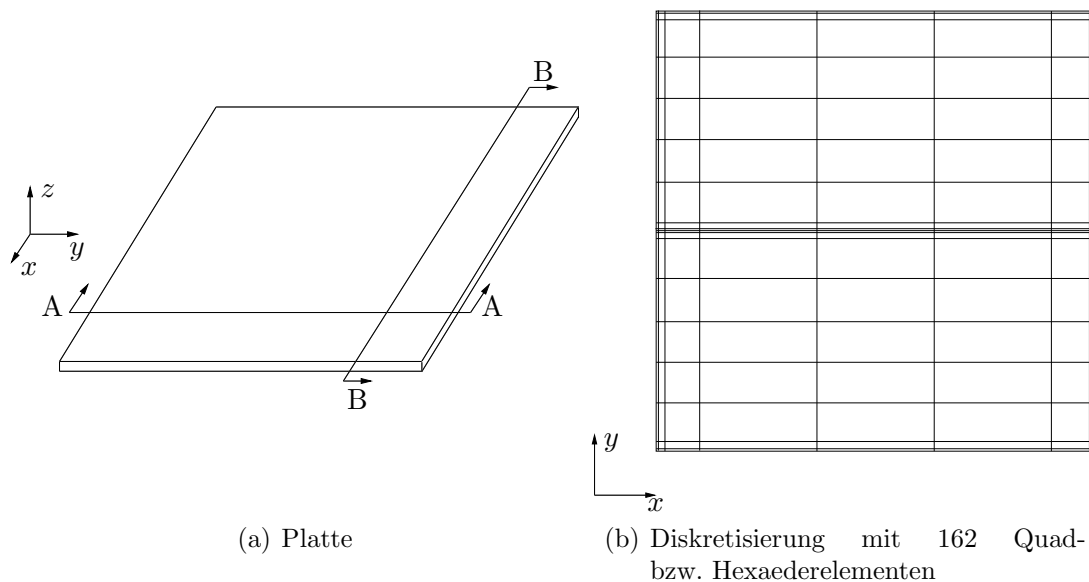


Abbildung 5.30: Struktur zur Berechnung der Spannungskorrektur

Als Material wird Stahl mit einem E-Modul $E = 210000 \frac{\text{N}}{\text{mm}^2}$ und einer Querdehnzahl $\nu = 0.3$ verwendet. Als Dirichlet-Randbedingung wurde eine Einspannung an allen vier Seiten der Platte gewählt. In Normalenrichtung der Platte wirkt eine bezüglich der y -Achse punktsymmetrische Flächenbelastung $q = 1000 \frac{\text{N}}{\text{mm}^2}$ (siehe Abbildung 5.31).

Nach der Ermittlung des Spannungszustandes mithilfe der Reissner-Mindlin-Theorie wird dieser als innerer Spannungszustand auf die Volumenelemente des zweiten Modells für die Spannungskorrektur übertragen. Dieses Modell wird dabei mit einem strikt dreidimensionalen Ansatz mit Hexaederelementen diskretisiert. Um nun eine Korrektur der Komponenten des Spannungszustandes durchzuführen, müssen die Dirichlet-Randbedingungen gegenüber des auf der Reissner-Mindlin-Theorie basierenden Initialschrittes modifiziert werden. Würde die Dirichlet-Randbedingung beibehalten werden, käme es zu einer Relaxation der Spannungen,

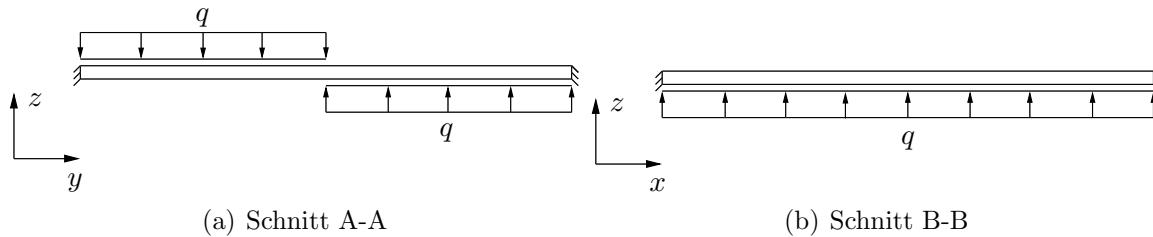


Abbildung 5.31: Randbedingungen des 2D-Modells zur Berechnung der Schalenlösung

bei der die Platte eine im Vergleich zum Initialschritt entgegengerichtete Verschiebung erfahren würde, da sich aufgrund der fehlenden Neumann-Randbedingung ein neues Gleichgewicht einstellen müsste. Daher werden die ursprünglichen Neumann-Randbedingungen durch äquivalente Dirichlet-Randbedingungen ersetzt (siehe Abbildung 5.32). Anstelle der ursprünglichen Randeinspannung werden die Randflächen nur in Richtung ihres Normalenvektors gesperrt, um Starrkörperrotationen zu vermeiden und den ebenen Verzerrungszustand sicherzustellen.

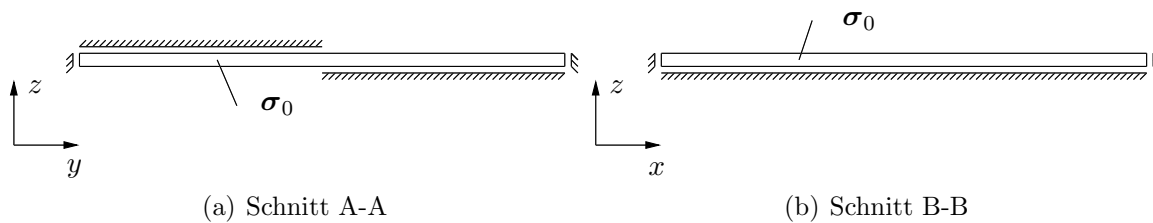


Abbildung 5.32: Randbedingungen des 3D-Modells zur Berechnung der Spannungskorrektur

Die veränderte Kinematik des Volumenmodells sowie das veränderte konstitutive Gesetz erlauben es nun, dass sich ein neuer Spannungszustand in der Struktur einstellt. Durch die Spannungsumlagerung der Schalenlösung im Volumenmodell können die zur Berechnung der Pressenkraft erforderlichen Komponenten ermittelt werden (siehe Abbildung 5.33). Bemerkenswert dabei ist, dass sich die Komponenten in x - bzw. y -Richtung des Schalenmodells und der strikt dreidimensionalen Berechnung kaum unterscheiden und damit auch die Korrekturlösung zu keiner Änderung der Werte führt. Werden jedoch die Komponenten in z -Richtung σ_{zz} , σ_{xz} sowie σ_{yz} betrachtet, so ist deutlich der Unterschied zwischen dimensionsreduzierter und dreidimensionaler Rechnung zu erkennen.

5.7.2 Erweiterungen des Modells

Das im vorangegangenen Kapitel diskutierte, akademische Beispiel möchte nur die Leistungsfähigkeit der Spannungskorrektur demonstrieren und entspricht in keinster Weise einer Umformsimulation. Um Berechnungen der Pressenkraft für reale Probleme durchzuführen, muss das vorgestellte Modell erweitert werden. Dabei muss vor allem auf die Modellierung der Randbedingungen für den Korrekturschritt geachtet werden. Im letzten Kapitel wurden die Dirichlet-Randbedingungen auf Ober- bzw. Unterseite auf der ehemaligen Lastfläche angenommen. Da eine Umformsimulation ein verschiebungsgesteuertes Problem darstellt, ist diese Annahme hier nicht möglich. Vielmehr geht es jetzt darum, die Kontaktbereiche zwischen Werkzeug und Bauteil als Dirichlet-Randbedingungen zu modellieren. Prinzipiell können diese über zwei

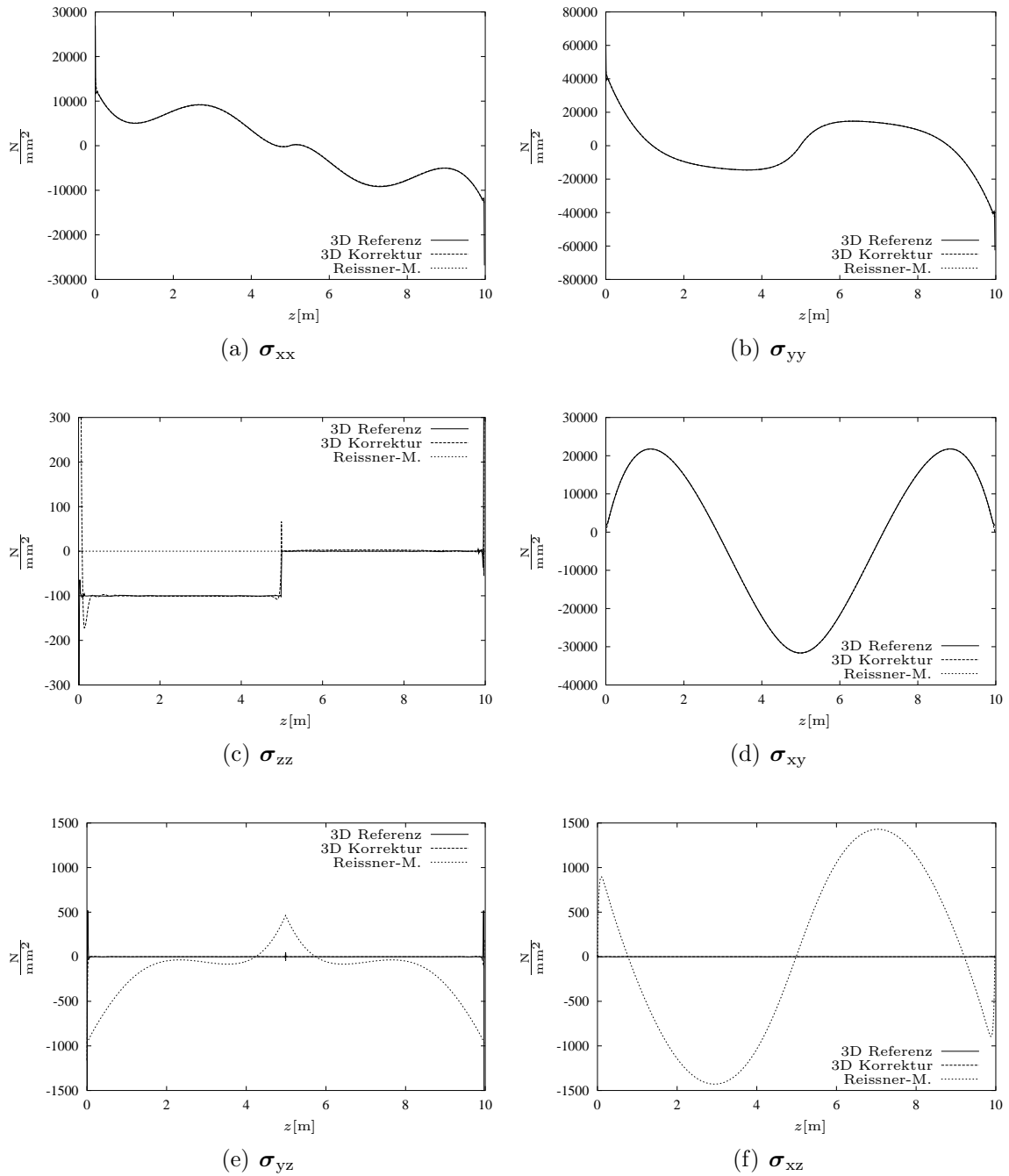


Abbildung 5.33: Komponenten des Spannungstensors des Ausgangs- und Korrekturzustands

Verfahren gewonnen werden. Zum einen kann aus dem Netz mit den in Kapitel 4.3 bzw. 5.2 beschriebenen Verfahren eine kontinuierliche Flächenbeschreibung ermittelt werden. Da die Oberflächenbeschreibung der Werkzeuge ebenfalls als Freiformfläche oder analytische Beschreibung vorliegt, kann mithilfe boolescher Operatoren (siehe Kapitel 4.4) die Schnittmenge beider Flächen als Kontaktfläche ermittelt werden. Eine andere Möglichkeit ist die Ermittlung der Kontaktflächen über den Spannungszustand des Schalenmodells. Da $\sigma_{zz} = 0$ per Definition gilt, kann diese Komponente nicht zur Identifikation herangezogen werden. Üblicherweise müssen an der freien Oberflächen die Schubspannungen $\sigma_{xy}, \sigma_{xz}, \sigma_{xy}$ verschwinden. An Stellen an denen Kontakt bzw. Reibung vorliegt sind die Schubspannungen $\sigma_{xy}, \sigma_{xz}, \sigma_{xy} \neq 0$. Das Vorliegen einer lokalen Schubspannungskomponente kann daher als Indiz für einen Kontaktbereich herangezogen werden. Bei beiden Verfahren muss bedacht werden, dass aufgrund des expliziten Lösungsverfahrens in der Umformsimulation ein Rauschen in der Lösung vorhanden ist. Daher ist es sinnvoll, ein gewisses Toleranzmaß für die Kontaktflächen-Klassifizierung festzulegen. Für die ermittelten Bereiche müssen nun Verschiebungsrandbedingungen definiert werden, bei denen die Fläche in ihrer Normalenrichtung festgehalten und in der Ebene der Kontaktfläche der Reibung entsprechend verschieblich ist (siehe Abbildung 5.34).

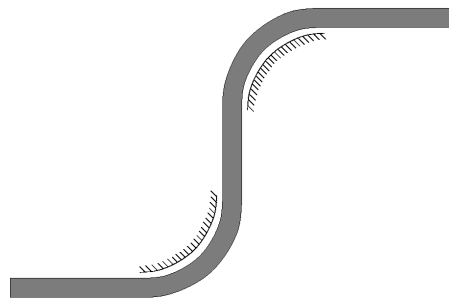


Abbildung 5.34: Normale / tangentielle Randbedingung

Desweiteren ist es erforderlich, dass diese Randbedingung entsprechend der Realität normal zur Fläche nur Druckspannungen aufnehmen kann, da sich sonst die Spannungen bzw. Kräfte lokal nicht korrekt umlagern können. In Kapitel 5.7.1 kann sich aufgrund der Wahl der Randbedingungen kein Doppelkontakt einstellen. Bei realen Bauteilen wird ebenfalls aufgrund der höheren Verschleißerscheinungen bei Werkzeugen ein Doppelkontakt vermieden. Dies ist eine wichtige Tatsache, die auch bei der Modellierung der äquivalenten Dirichlet-Randbedingungen berücksichtigt werden muss. Im Falle einer beidseitigen Lagerung könnte sich in lokaler Dickenrichtung keine Verzerrung ε_{zz} und somit keine Spannungsumlagerung für σ_{zz} einstellen. Sollten solche Kontakte Anwendung finden, müsste eine entsprechend gebettete Verschiebungs-Randbedingung eingeführt werden.

Kapitel 6

Anwendungs-Beispiele

Das folgende Kapitel soll die Umsetzung der Simulation der elastischen Rückfederung mit der p -Version der Finite-Elemente-Methode anhand von vier numerischen Beispielen demonstrieren.

Das erste Beispiel beschreibt einen Ausschnitt eines U-Profiles. Diese rein akademische Anwendung welche aufgrund ihrer unendlichen Ausdehnung mit einem ebenen Verzerrungszustand und mit einer symmetrischen Randbedingung berechnet wird, dient hauptsächlich der Verifikation der Implementierung. Aufgrund der geringen Komplexität können mit dem U-Profil-Streifen verschiedene Fragestellungen wie z. B. die korrekte Implementierung der geometrischen Nichtlinearität anhand einer linear elastischen Umformung beantwortet werden.

Weiter soll das für viele Benchmarkberechnungen herangezogene S-Rail analysiert werden. Das S-Rail diente ursprünglich als Benchmarkbauteil für die NUMIFORM 1996 [47]. Aufgrund der Anzahl an vorhandenen Referenzlösungen lässt sich mithilfe dieses Bauteils die Rückfederungsberechnung verifizieren.

Beide genannten Beispiele kommen aus dem akademischen Bereich und können daher die Leistungsfähigkeit der p -Version der Finite-Elemente-Methode für komplexe Problemstellungen nicht wiedergeben. Daher werden zusätzlich zwei industrielle Bauteile vorgestellt, welche aktuellen Fahrzeugprojekten entstammen. Zunächst wird die elastische Rückfederung für einen Wasserstoff-Tank berechnet. Dieses Bauteil lässt aufgrund seiner glatten Geometrie eine geeignete Diskretisierung für die p -Version der Finite-Elemente-Methode zu. Weiterhin kann durch die doppelt-symmetrische Dirichlet-Randbedingung das Ergebnis der Rückfederung ohne Translationen und Rotationen mit anderen Simulationsprogrammen verglichen werden.

Das bezüglich Größe und Geometrie komplexeste Bauteil stellt der Ende dieses Kapitels präsentierte Längsträger dar. Es wird gezeigt, dass sich dieses Bauteil dank der guten Konvergenzeigenschaften mit der p -Version der Finite-Elemente-Methode und einer strikt dreidimensionalen Formulierung auf einem Standard-PC berechnen lässt, was bei klassischen Implementierungen mit der h -Version selbst auf Workstations noch nicht möglich ist.

6.1 U-Profil

Der U-Profil-Streifen stellt mit einem ebenen Verzerrungszustand ein in y -Richtung unendlich ausgedehntes einfach symmetrisches Bauteil dar. An der Symmetrieachse wird eine entsprechende Einspannung definiert, mit der gleichzeitig die Starrkörpermodi verhindert werden. Es wurden zwei Simulationen mit verschiedenen Materialmodellen durchgeführt. Dabei wurde für das erste Modell ein linear elastisches konstitutives Gesetz angenommen. Dieses rein fiktive Materialverhalten soll dazu dienen, dass sich das Bauteil bei der Rückfederung in seine Ausgangslage zurück verformt. Bei der Umformung traten jedoch numerische Instabilitäten der Simulationsprogramme auf, welche auf die hohen elastischen Spannungen im Bauteil zurückzuführen sind. Daher wurde für diese Simulation der E-Modul um den Faktor 10 reduziert um eine stabile Umformsimulation zu ermöglichen. Für die zweite Simulation wurde ein elastisch-plastisches Materialverhalten angenommen, wie es bei üblichen Umformsimulationen verwendet wird. Weiterhin wurde die Umformsimulation auf einem Schalennetz mit 125 bilinearen Elementen durchgeführt. Für die Rückfederungsberechnung wurde das Bauteil mit 6 Hexaederelementen rediskretisiert. Die Elemententeilung wurde dabei so vorgenommen, dass im Element eine gleiche Krümmungsrichtung und ein möglichst konstanter Radius vorliegt. Dies hat den Vorteil, dass das jeweilige Element mit einem möglichst geringen Polynomgrad für die Geometrie beschrieben werden kann (siehe auch Abbildung 5.24).

6.1.1 Geometrisch nichtlineare Berechnung der Rückfederung

Beim Tiefziehprozess dünner Bleche liegt ein stark nichtlineares Problem vor. Aufgrund der großen Verschiebungen muss eine geometrisch nichtlineare Berechnung durchgeführt werden. Bei der elastischen Rückfederung treten bei praxisrelevanten Fragestellungen von Natur aus kleinere Deformationen auf, da nur die relativ kleinen reversiblen Dehnungen aus dem Umformprozess eine Verschiebung bewirken. Jedoch kann auch hier eine geometrisch nichtlineare Berechnung erforderlich werden, z. B. bei Einsatz hochfester Stähle mit hoher Streckgrenze, Aluminium-Werkstoffen oder komplexen Geometrien mit kleinen Radien.

Im folgenden soll zunächst die elastische Rückfederung für das mit dem physikalisch linearen konstitutiven Gesetz berechnete Bauteil bestimmt werden. Dabei ist aus Abbildung 6.1(a) ersichtlich, dass bei einer geometrisch linearen Berechnung das Bauteil nicht in seinen ursprünglichen Ausgangszustand zurück springt. Bei geometrisch nichtlinearer Berechnung erreicht die Verschiebung eine vollständige Rückfederung in die Ausgangslage vor der Umformung (Abbildung 6.1(b)). Der Fehler in der Verschiebung beträgt dabei 0.12% bezogen auf die Profillänge. Dabei erhebt dieses Beispiel nicht den Anspruch eine reale Umformung zu simulieren. Vielmehr kann damit

- die Rediskretisierung der rückgeführten Geometrie,
- der Transfer der Geschichtsvariablen sowie
- die Implementierung der geometrischen Nichtlinearität

leicht verifiziert werden, da die Abweichung zur Sollgeometrie leicht bestimmt werden kann. Während bei letztgenannten Punkt systematische Fehler identifiziert werden können, sind die

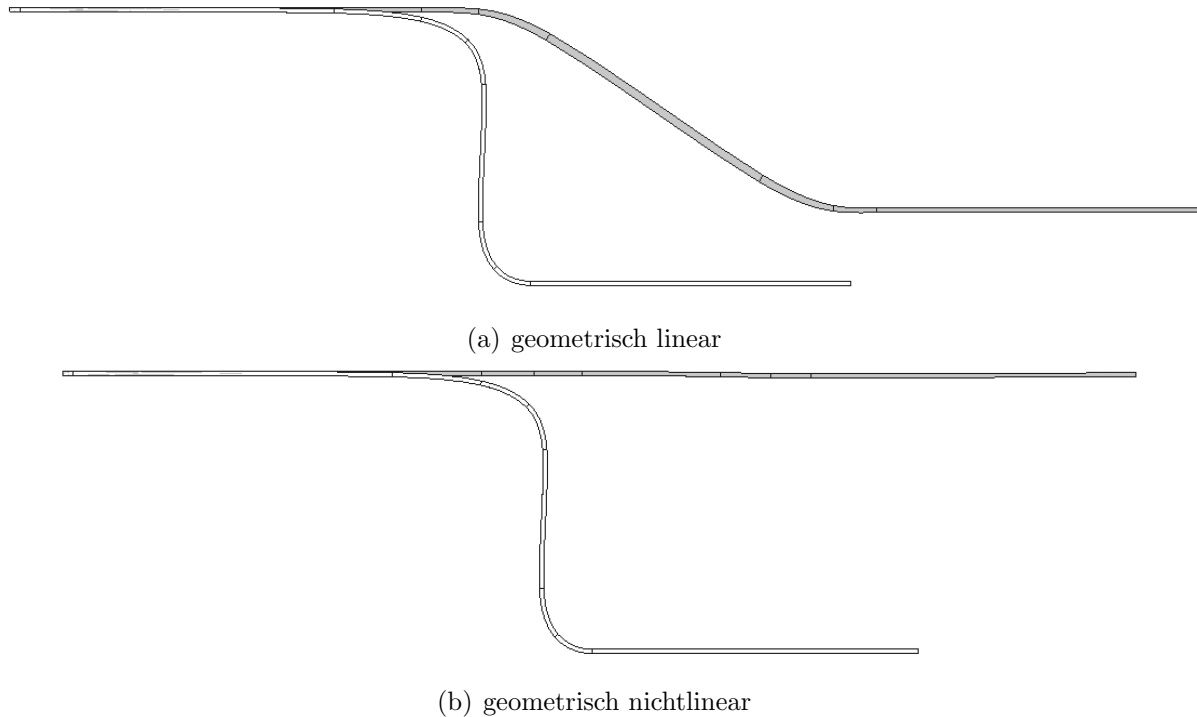
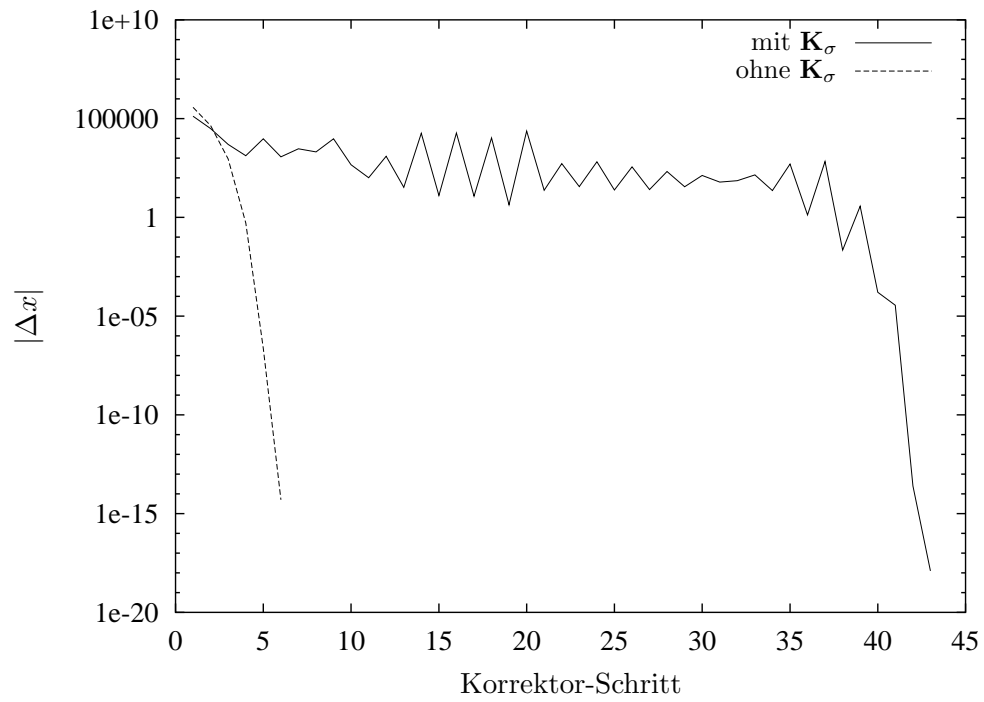


Abbildung 6.1: Rückfederung des linear elastisch tiefgezogenen U-Profil

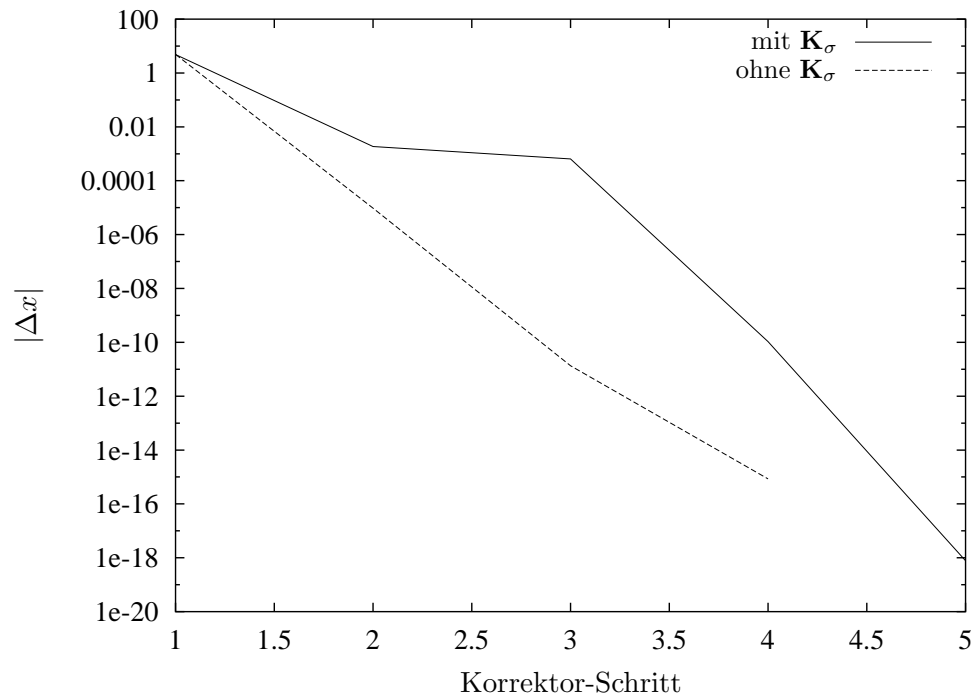
beiden erst genannten Punkte unter anderem auch von Simulationsparametern abhängig. Dieses Beispiel kann daher auch zur Bestimmung von Anwenderrichtlinien herangezogen werden.

Bei der geometrisch nichtlinearen Rückfederung des linear elastischen U-Profiles handelt es sich um ein stark nichtlineares Problem, da dabei die gleichen - allerdings invertierten - Verformungen wie beim Umformprozess auftreten. Daher ist es in diesem Fall üblicherweise sinnvoll, bei einer geometrisch nichtlinearen Berechnung eine für eine effiziente Konvergenz ausreichende Anzahl an Lastschritten für das Newton-Raphson-Verfahren zu wählen. Aufgrund der statisch bestimmten Lagerung kann gezeigt werden, dass durch eine vollständige Vernachlässigung der geometrischen Steifigkeit \mathbf{K}_σ in allen Korrektor-Schritten selbst bei der Wahl von nur einem Lastschritt eine quadratische Konvergenz erreicht werden kann, wogegen bei Berücksichtigung der geometrischen Steifigkeit die Lösung erst nach 40 Korrektor-Schritten dieses Verhalten zeigt (siehe Abbildung 6.2(a)). Bei einer entsprechend klein gewählten Lastschrittweite konvergieren beide Verfahren ungefähr gleich gut (siehe Abbildung 6.2(b)), allerdings ist auch hier das Verfahren mit Vernachlässigung der geometrischen Steifigkeit effizienter, da die Berechnung der geometrischen Steifigkeit in jedem Lastschritt einen nicht unerheblichen Aufwand darstellt.

Wie in Kapitel 3.6.7 bereits erwähnt, ist nur bei statisch bestimmten Systemen eine Erhöhung der Konvergenzgeschwindigkeit durch Vernachlässigung der geometrischen Steifigkeit zu erwarten. Bei statisch überbestimmten Systemen führt das Vorgehen zu einer falschen Tangensteifigkeit und somit zu einer schlechteren Konvergenz. Um dies zu illustrieren wurde im folgenden das linear elastische U-Profil mit einer Dirichlet-Randbedingung versehen, welche eine statisch überbestimmte Lagerung darstellt und somit die (vollständige) Rückfederung



(a) Konstante Schrittweite 1.0



(b) Konstante Schrittweite 0.05

Abbildung 6.2: Konvergenz der geometrisch nichtlinearen Rückfederung

verhindert. Aus Abbildung 6.3 ist deutlich die langsamere Konvergenz bei Vernachlässigung von \mathbf{K}_σ zu erkennen.

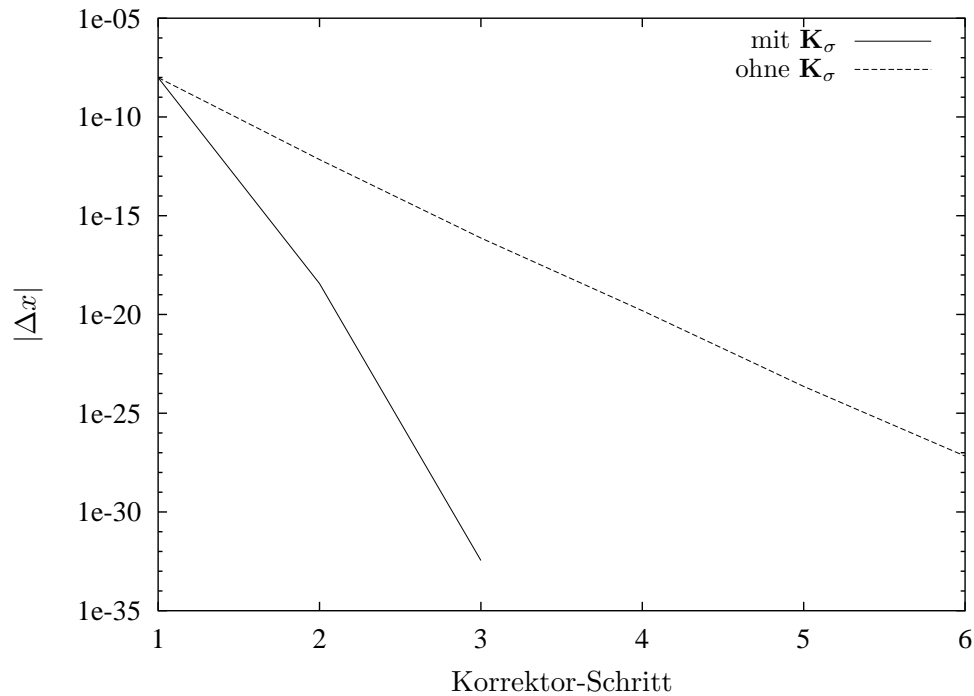


Abbildung 6.3: Konvergenz bei statisch überbestimmter Lagerung

Nun soll die elastische Rückfederung für das Bauteil mit dem elastisch-plastisch konstitutiven Gesetz bestimmt werden. In Abbildung 6.4 ist die elastische Rückfederung sowohl geometrisch nichtlinear als auch linear dargestellt. Es ist offensichtlich, dass die geometrisch lineare Rückfederung die Verschiebung als zu gering ermittelt. Allerdings kann dieser Effekt bei schwach nichtlinearen Problemen vernachlässigt bzw. abgeschätzt werden. Ein weiterer Effekt, der bei der geometrisch linearen Berechnung auffällt (siehe Abbildungen 6.4 und 6.1(a)), ist die Veränderung der Länge bzw. des Querschnitts des Bauteils welche durch die Linearisierung der Verschiebung entsteht.

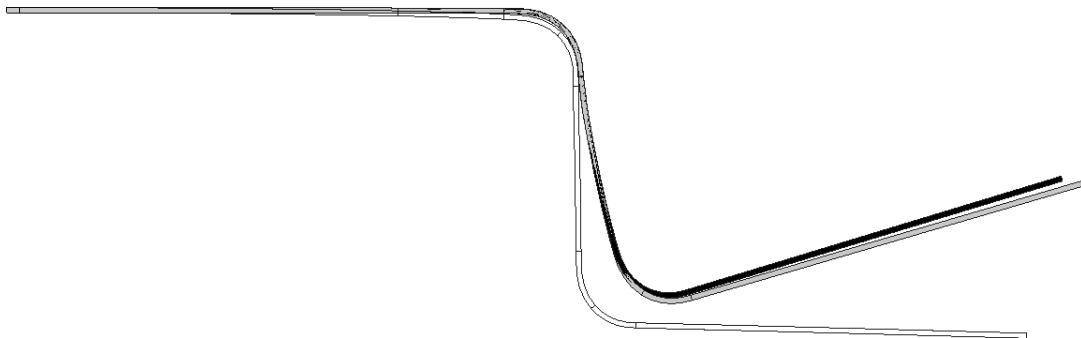


Abbildung 6.4: Rückfederung des elastisch-plastisch umgeformten U-Profiles (schwarz: geometrisch nichtlineare Rückfederung, grau: geometrisch lineare Rückfederung)

6.2 S-Rail

Das S-Rail-Bauteil welches ursprünglich als Benchmarkbauteil für die NUMIFORM 1996 [47] entwickelt wurde, dient heute häufig zum Vergleich von Ergebnissen unterschiedlicher Anwendungsprogramme. Im vorliegenden Fall wurde das Bauteil dabei für den Umformprozess mit ca. 19000 bilinearen Schalenelementen diskretisiert und schließlich für die elastische Rückfederung in ein gekrümmtes Hexaedernetz für die p -Version der Finite-Elemente-Methode übergeführt (siehe Abbildung 6.5).

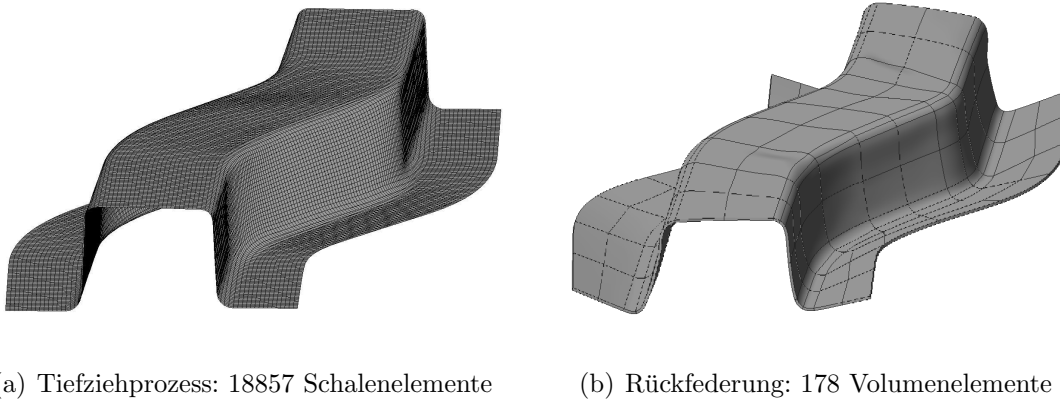


Abbildung 6.5: Umformen als gekoppelter Prozess

6.2.1 Konvergenz der p -Version

Zunächst wird für die Diskretisierung mit 178 Elementen der Polynomgrad für alle Richtungen schrittweise erhöht. Dabei werden die zwei verschiedenen Ansatzräume $\mathcal{S}_{ts}^{p\xi, p\eta, p\zeta}(\Omega_{st}^h)$ und $\mathcal{S}_{ps}^{p\xi, p\eta, p\zeta}(\Omega_{st}^h)$ betrachtet. Um die Konvergenz der p -Version aufzuzeigen, werden die Verschiebungen entlang einer charakteristischen Schnittlinie (siehe Abbildung 6.6) sowie der relative Fehler in der Energienorm miteinander verglichen.

Aus Abbildung 6.7 ist ersichtlich, dass bei einem Polynomgrad $p \geq 8$ für den *trunk space* und bereits bei $p \geq 5$ beim *tensor product space* eine Konvergenz der Verschiebungen eintritt.

Weiter soll der relative Fehler in der Energienorm bestimmt werden. Die zur Bestimmung des Fehlers erforderliche Referenzlösung wird dabei mithilfe der im folgenden beschriebenen *a posteriori* Extrapolationsmethode nach SZABO [75, 77] bestimmt.

Es kann gezeigt werden, dass im Falle einer h - und p -Diskretisierung der Fehler *a priori* mit

$$\|u_{\text{ex.}} - u_{\text{FE}}\|_{E(\Omega)} \leq \frac{k}{N^\beta} \quad (6.1)$$

geschätzt werden kann, wobei k und β positive Konstanten und N die Anzahl der Freiheitsgrade ist. Der nachfolgend beschriebene Ansatz berechnet eine *a posteriori* Schätzung für $\|u_{\text{ex.}} - u_{\text{FE}}\|_{E(\Omega)}$ mithilfe der Ergebnisse von drei aufeinander folgenden FE-Analysen. Der

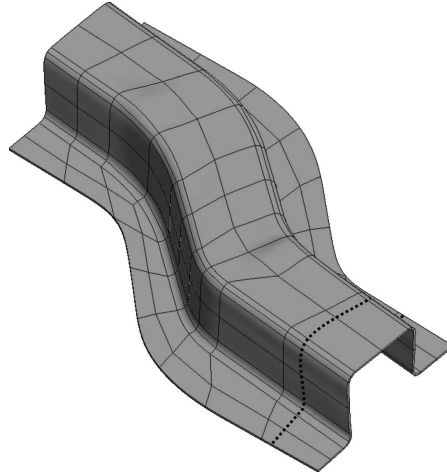


Abbildung 6.6: Auswertelinie der Verschiebung

Fehler in der Energie wird als $e := u_{\text{ex.}} - u_{\text{FE}}$ definiert und kann mithilfe von Gleichung (3.25) als Fehler in der Energienorm

$$\|e\|_{E(\Omega)}^2 = \mathcal{B}(e, e) = \Pi(u_{\text{FE}}) - \Pi(u_{\text{ex.}}) \quad (6.2)$$

ausgedrückt werden. Mit Gleichung (6.2) ergibt sich für Gleichung (6.1)

$$\|u_{\text{ex.}} - u_{\text{FE}}\|_{E(\Omega)}^2 = \Pi(u_{\text{FE}}) - \Pi(u_{\text{ex.}}) \approx \frac{k^2}{N^{2\beta}}. \quad (6.3)$$

Um die drei Unbekannten $\Pi(u_{\text{ex.}})$, k , und β zu bestimmen wird aus Gleichung (6.3) mit drei verschiedenen Werten für $\Pi(u_{\text{FE}})$ und N aus drei aufeinander folgenden p -Diskretisierungen ein bestimmtes Gleichungssystem aufgestellt. Dabei korrespondieren die Anzahl der Freiheitsgrade N_{p-2} , N_{p-1} , N_p bzw. die Energien Π_{p-2} , Π_{p-1} , Π mit den aufeinander folgenden Polynomgraden $p-2$, $p-1$ und p . Die Näherungslösung Π für $\Pi_{u_{\text{ex.}}}$ wird durch Lösen von

$$\frac{\log \frac{\Pi - \Pi_p}{\Pi - \Pi_{p-1}}}{\log \frac{\Pi - \Pi_{p-1}}{\Pi - \Pi_{p-2}}} \approx \frac{\log \frac{N_{p-1}}{N_p}}{\log \frac{N_{p-2}}{N_{p-1}}} \quad (6.4)$$

bestimmt. Das Verfahren setzt dabei eine monotone Konvergenz von Π voraus. Dies ist gleichbedeutend mit der Forderung, dass die Ansatzräume S_{p-2}, S_{p-1}, S_p die Eigenschaft $S_{p-2} \subset S_{p-1} \subset S_p$ besitzen.

Der Verlauf des Fehlers in der Energienorm in Abbildung 6.8 zeigt die für glatte Lösungen mit der p -Version der Finite-Elemente-Methode erzielbare exponentielle Konvergenzrate. Allerdings fällt beim Vergleich der Ansatzräume auf, dass der *trunk space* eine effizientere Diskretisierung als der *tensor product space* darstellt, da hierbei für den gleichen Fehler erheblich weniger Freiheitsgrade nötig sind.

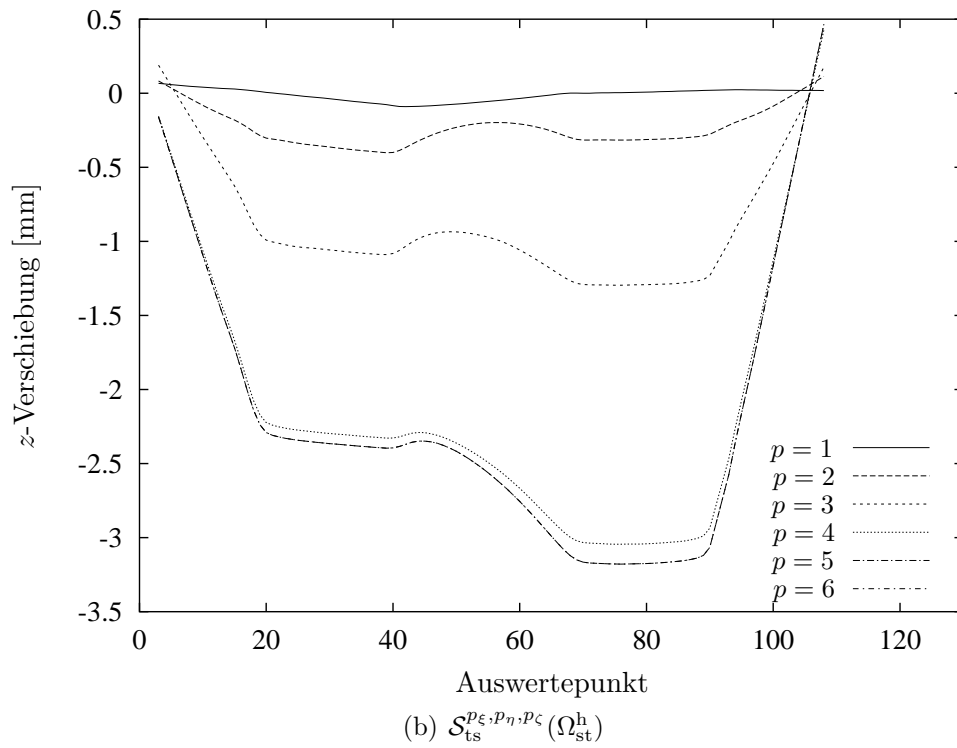
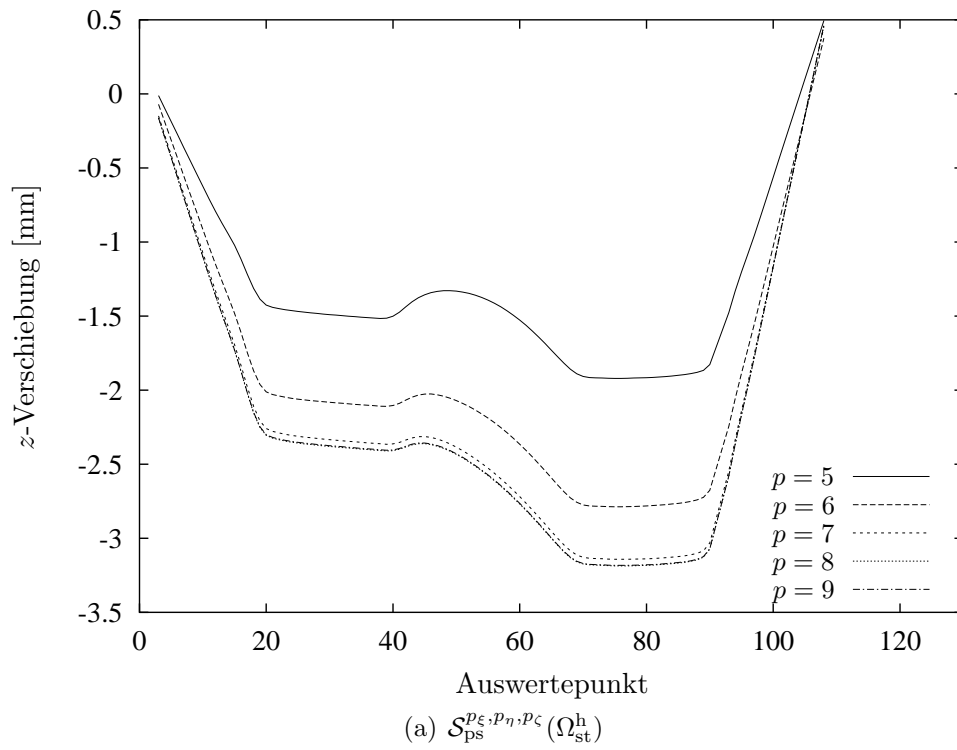


Abbildung 6.7: Konvergenz der Verschiebungen

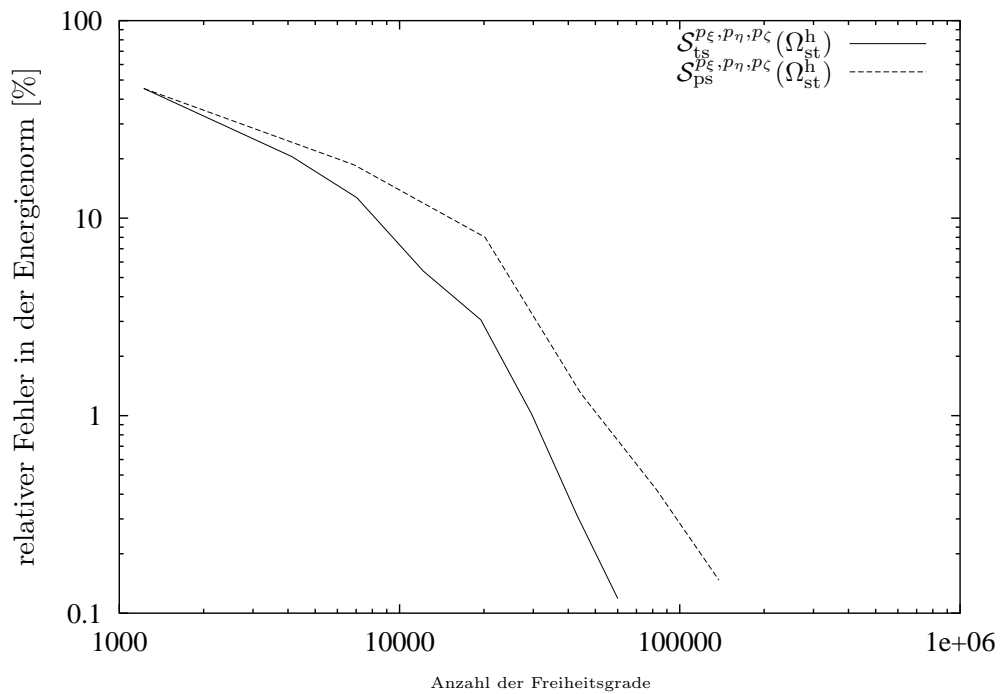


Abbildung 6.8: Konvergenz in der Energienorm

6.2.2 Anisotrope Ansatzräume

Bei anisotropen Ansatzräumen kann über ein Polynomgradtemplete

$$T = \begin{matrix} & u_x & u_y & u_z \\ \begin{matrix} \xi \\ \eta \\ \zeta \end{matrix} & \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{pmatrix} \end{matrix} \quad (6.5)$$

der Polynomgrad für jede Richtung und jede Komponente des Verschiebungsfeldes unabhängig voneinander definiert werden [21, 20]. Dieses Vorgehen ist sehr effizient, da für dünnwandige Strukturen im Gegensatz zu Schalenmodellen der Modellfehler über den Polynomgrad in Dickenrichtung kontrolliert werden kann. Der Modellfehler wird somit zu einem Diskretisierungsfehler. Da die Verschiebungen der Hexaederelemente in den globalen Komponenten u_x, u_y und u_z ausgedrückt werden, ist es bei gekrümmten Elementen zweckmäßig, für die Komponenten einer Richtung einen isotropen Polynomgrad zu wählen. Aufgrund der ähnlichen Abmessungen in der Ebene wird für $p_\xi = p_\eta = p$ angenommen. Somit reduziert sich das Polynomgradtemplete aus Gleichung (6.5) für dünnwandige, gekrümmte Hexaederelemente zu

$$\begin{matrix} & u_x & u_y & u_z \\ \begin{matrix} \xi \\ \eta \\ \zeta \end{matrix} & \begin{pmatrix} p & p & p \\ p & p & p \\ q & q & q \end{pmatrix} \end{matrix} \quad (6.6)$$

wobei gewöhnlich $p > q$ gilt.

Im folgenden soll untersucht werden, welche Polynomgradverteilung eine möglichst effiziente Diskretisierung des Bauteils darstellt. Diese Untersuchung muss für die verschiedenen Ansatzräume durchgeführt werden. Als Kriterium für die Konvergenz dient dabei der relative Fehler in der Energienorm bzw. die Verschiebungen entlang der Schnittlinie (siehe Abbildung 6.6).

Für das Berechnungs-Netz mit 178 Hexaeder-Elementen wird nun eine Finite-Element-Approximation mit dem Ansatzraum $\mathcal{S}_{\text{ts}}^{p,p,q}(\Omega_{\text{st}}^{\text{h}})$ durchgeführt, wobei $p = 8$ und q variabel gewählt wird. p korrespondiert dabei mit den lokalen Richtungen ξ und η und definiert alle Ansatzfunktionen, welche diese Variablen enthalten. q ist somit der Polynomgrad in Dickenrichtung, wobei alle Ansatzfunktionen im *trunk space* definiert sind. Die Referenzlösung stellt die Lösung mit einem isotropen Polynomgrad $p = q = 8$ dar.

Aus Abbildung 6.9 ist ersichtlich, dass für einen Polynomgrad $q \geq 4$ die Verschiebung gegen die isotrope Lösung konvergiert. Somit stellen anisotrope Ansatzräume eine effiziente Diskretisierung dar.

6.2.3 p -Adaptivität

RANK ET AL. präsentieren in [64] ein adaptives Verfahren zur Anpassung des Polynomgrads für jede Verschiebungskomponente der drei lokalen Richtungen des Elementkoordinatensystems. Dabei legt ein hierarchischer Fehlerschätzer im Postprocessing-Schritt für jedes Hexaederelement eine verbesserte Polynomgradverteilung fest. Dieses Verfahren wird so lange fortgesetzt, bis sich die Polynomgradverteilung der Elemente nicht mehr wesentlich ändert.

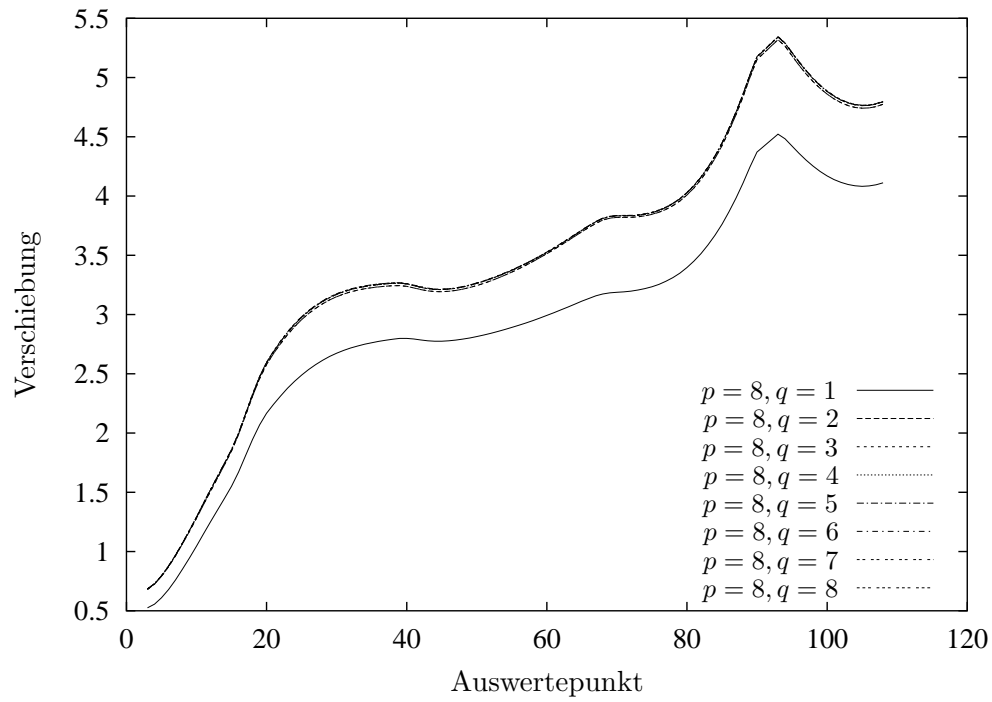
Im folgenden soll die Effizienz des Verfahrens mit dem nicht-adaptiven Ansatz verglichen werden. Dazu wird beim nicht-adaptiven Verfahren der Polynomgrad p der Polynomgradmatrix

$$\begin{array}{c} u_x \quad u_y \quad u_z \\ \xi \\ \eta \\ \zeta \end{array} \begin{pmatrix} p & p & p \\ p & p & p \\ p & p & p \end{pmatrix} \quad (6.7)$$

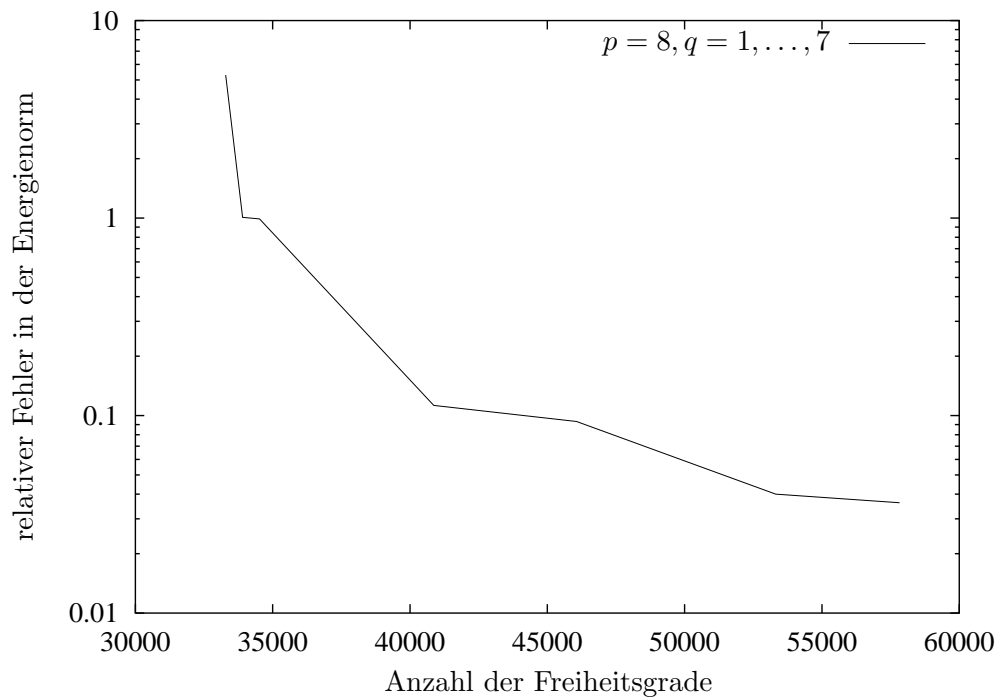
auf dem Ansatzraum $\mathcal{S}_{\text{ts}}^{p_{\xi},p_{\eta},p_{\zeta}}(\Omega_{\text{st}}^{\text{h}})$ schrittweise um 1 erhöht. Für das adaptive Verfahren wird im Initialschritt die Polynomgradverteilung

$$\begin{array}{c} u_x \quad u_y \quad u_z \\ \xi \\ \eta \\ \zeta \end{array} \begin{pmatrix} 6 & 6 & 6 \\ 6 & 6 & 6 \\ 3 & 3 & 3 \end{pmatrix} \quad (6.8)$$

auf dem Ansatzraum $\mathcal{S}_{\text{ts}}^{p_{\xi},p_{\eta},p_{\zeta}}(\Omega_{\text{st}}^{\text{h}})$ für jedes Element festgelegt, welche während der Iteration dann adaptiv angepasst wird. Die Polynomgradverteilung des letzten Iterationsschritts ist in Abbildung 6.10 dargestellt.



(a) Verschiebung



(b) relativer Fehler in der Energienorm

Abbildung 6.9: Anisotrope Polynomgradverteilung in Dickenrichtung mit Ansatzraum $\mathcal{S}_{ts}^{p,p,q}(\Omega_{st}^h)$

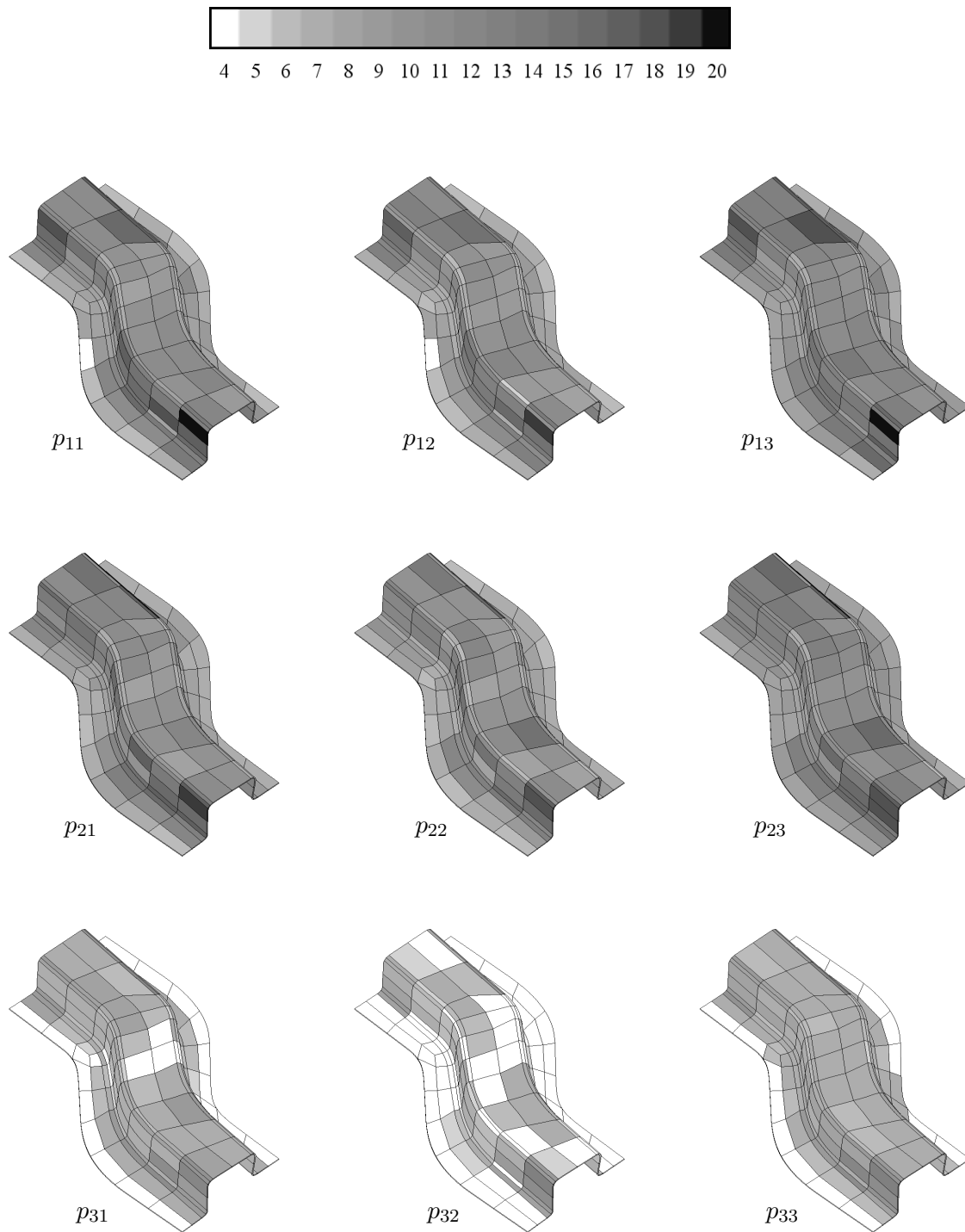


Abbildung 6.10: Polynomgradverteilung nach (6.5) im Iterationsschritt 9

Die Effizienz des Verfahrens ist durch die Analyse des Fehlers in der Energienorm in Abbildung 6.11 ersichtlich. Dabei fällt auf, dass am Anfang der Iteration sowohl das adaptive als auch das nicht-adaptive Verfahren eine ähnliche Konvergenzrate besitzen, da die Polynomgradverteilung beim adaptiven Verfahren noch sehr gleichmäßig und ähnlich zum nicht-adaptiven Verfahren ist. Mit zunehmender Anzahl an Iterationen zeigt sich jedoch die Überlegenheit des adaptiven Verfahrens durch eine schnellere Konvergenz für den Fehler in der Energienorm.

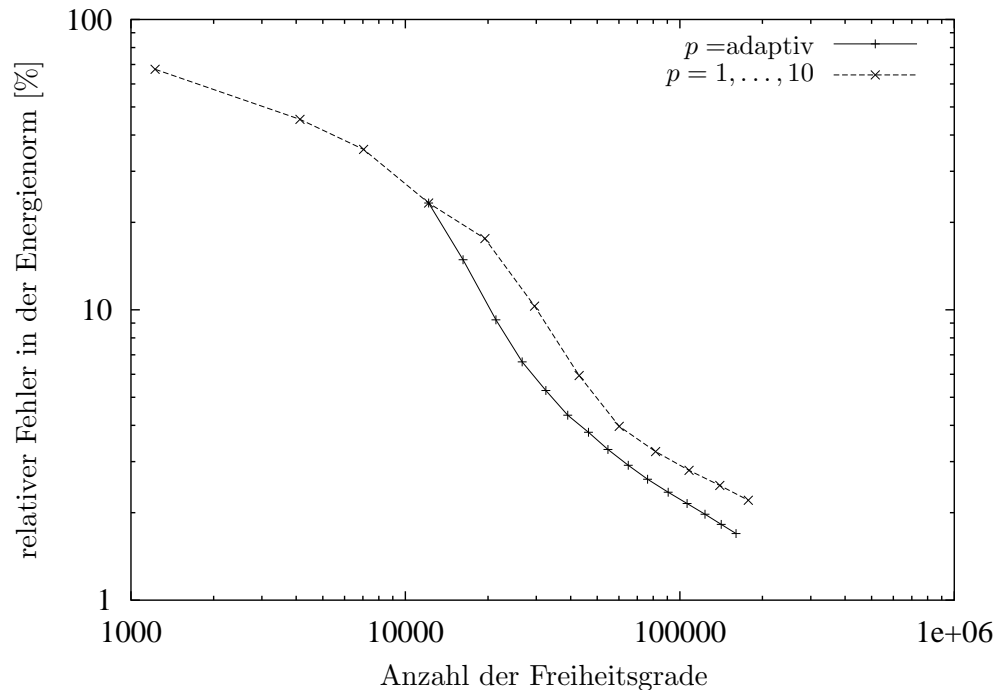


Abbildung 6.11: Vergleich des Fehlers in der Energienorm für adaptive und nicht-adaptive Berechnung

6.2.4 Geometrische Nichtlinearität

Das praxisrelevante S-Rail zeigt nur einen geringen Einfluss der geometrischen Nichtlinearität bei der Berechnung der elastischen Rückfederung. Durch diese schwache Nichtlinearität erreichen Newton-Raphson-Verfahren in der Regel mit nur einem Lastschritt eine hinreichende Konvergenz. Mit dem in Kapitel 3.6.7 beschriebenen Verfahren wird eine ungefähr gleiche Anzahl an Korrektor-Schritten (siehe Abbildung 6.12) benötigt, allerdings ist der Aufwand bei Vernachlässigung der geometrischen Steifigkeit durch eine geringere Anzahl an Matrix-Matrix-Multiplikationen beträchtlich geringer.

Beim Vergleich der Rückfederungsergebnisse der geometrisch nichtlinearen mit der geometrisch linearen Berechnung an der Schnittlinie (Abbildung 6.13) ist eine geringe Abweichung ersichtlich. Gerade bei der virtuellen Prozessabsicherung können jedoch solche Abweichungen vernachlässigt bzw. abgeschätzt werden, zumal eine geometrisch nichtlineare Berechnung ein Vielfaches an zeitlichen Aufwand gegenüber einer geometrisch linearen Berechnung erfordert.

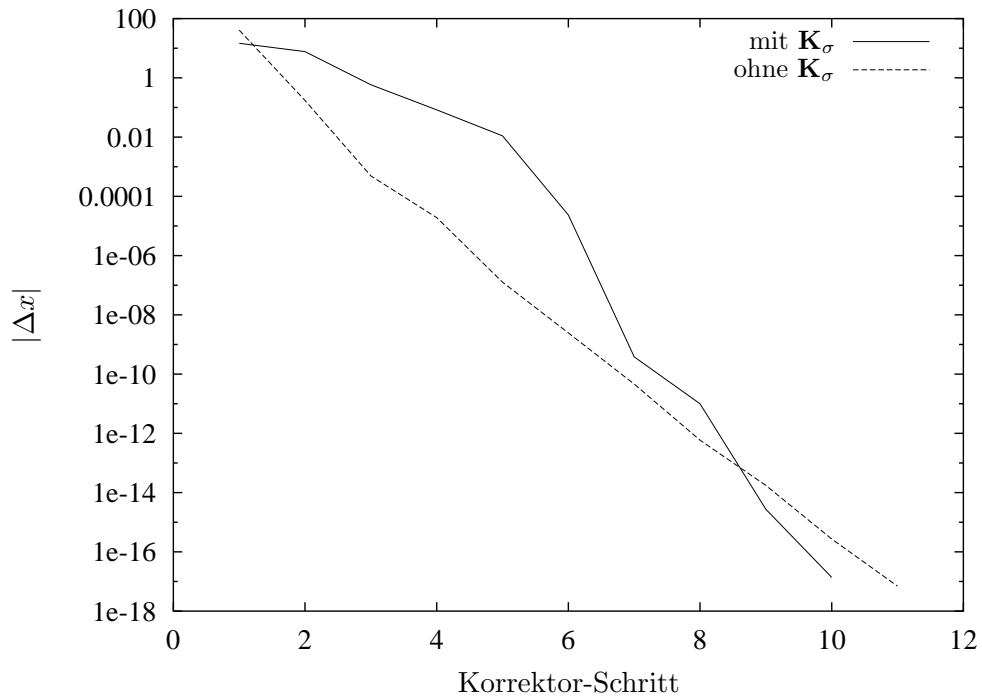


Abbildung 6.12: Konvergenz der geometrisch nichtlinearen Rückfederung

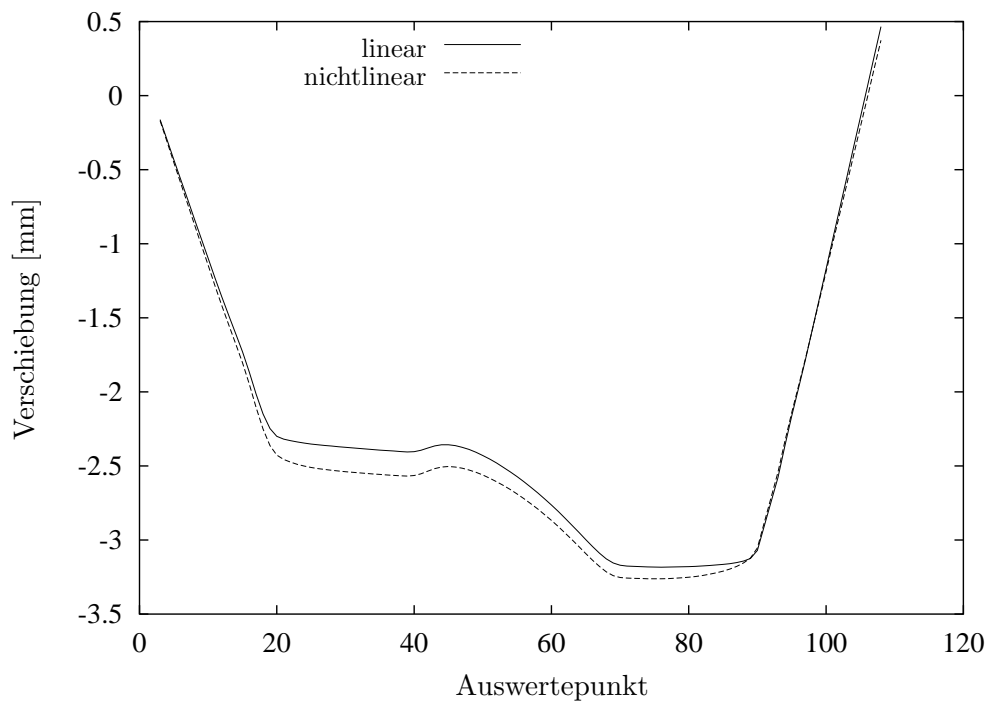


Abbildung 6.13: Verschiebungen der geometrisch linearen und nichtlinearen Berechnung

6.3 Wasserstoff-Tank

In diesem Kapitel soll eine doppelt-symmetrische Schalenstruktur betrachtet werden. Dieses industrielle Bauteil weist aufgrund seiner Geometrie ein sehr gutes Diskretisierungsverhalten für die p -Version der Finite-Elemente-Methode auf. Dabei wird das Bauteil an seinen Symmetrieachsen durch entsprechende Dirichlet-Randbedingungen gelagert. Aufgrund der Geometrieapproximation wurde das Bauteil für die Umformsimulation mit ca. 36000 Schalenelementen diskretisiert (siehe Abbildung 6.14) und für die p -Version mit nur 12 gekrümmten Hexaederelementen rediskretisiert (siehe Abbildung 6.15).

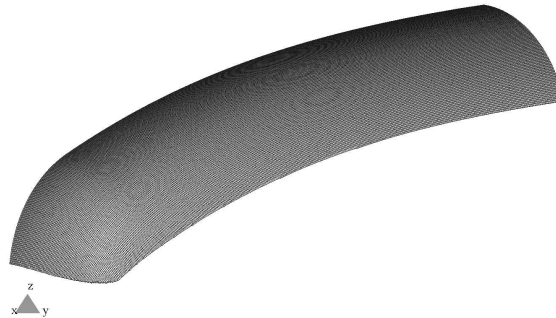


Abbildung 6.14: Diskretisierung der Tiefziehsimulation mit Schalenelementen

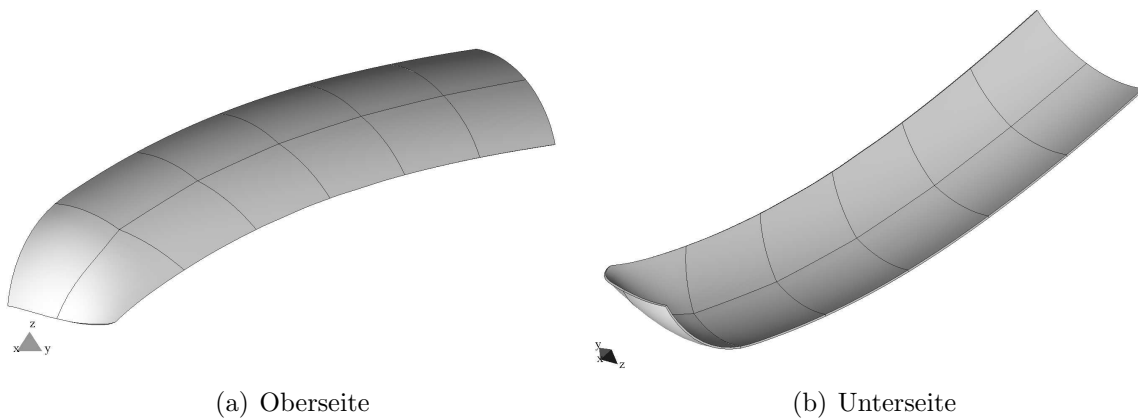


Abbildung 6.15: Diskretisierung für die Rückfederung mit der p -Version der FEM

6.3.1 Genauigkeit

Es ist aufgrund der einfachen Geometrie mit relativ großen Radien zu erwarten, dass die Schalentheorie für das Bauteil ein zuverlässiges Ergebnis für die elastische Rückfederung liefert. Daher soll nun die Rückfederung mit dem strikt dreidimensionalen Ansatz höherer Ordnung mit den Ergebnissen der Schalentheorie niedriger Ordnung verglichen werden. Aus Abbildung 6.16 sind deutlich die Übereinstimmung der Verschiebungsverläufe an der Schnittlinie

zu erkennen. Bemerkenswert dabei ist, dass für die gleiche Genauigkeit bei der p -Version aufgrund ihrer guten Konvergenzeigenschaften eine geringere Rechenzeit benötigt wird als bei der dimensionsreduzierten Schalenlösung.

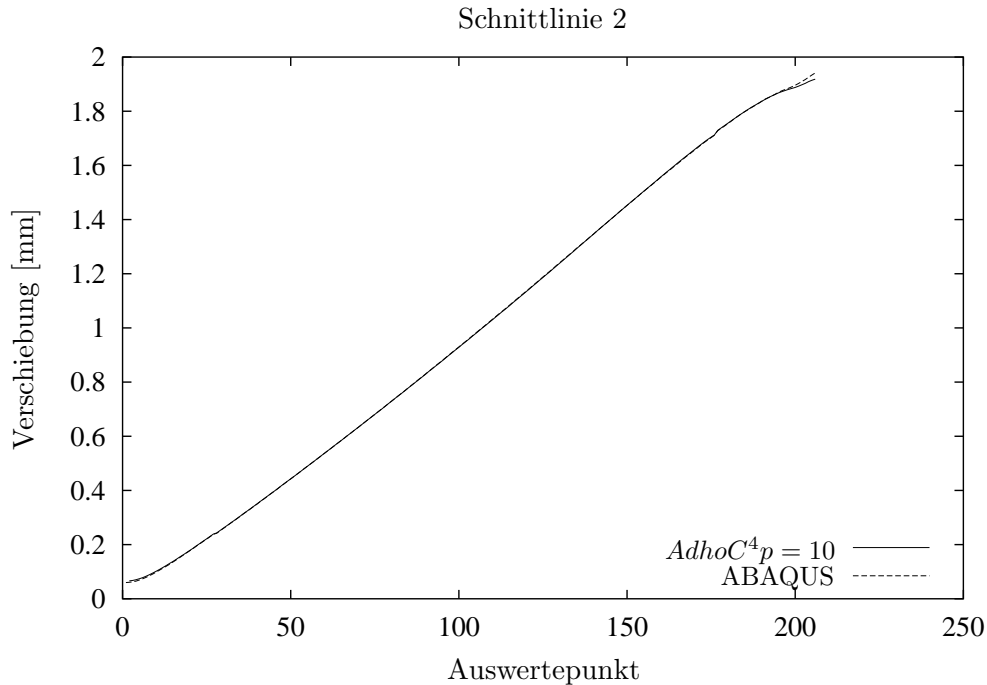


Abbildung 6.16: Vergleich der Schalentheorie h -Version mit Volumenelementen p -Version

6.3.2 Konvergenz der p -Version

Im folgenden soll die Konvergenz von drei unterschiedlichen Diskretisierungen der h - und p -Version mit Hexaederelementen betrachtet werden. Für die p -Version wurde das Hexaedernetz mit 12 Elementen und einem fixen Polynomgrad der Geometrie in der Ebene von $p_{\text{geo}} = 4$ gewählt. Der Polynomgrad der Verschiebungen wurde bei allen Diskretisierungen isotrop gewählt. Die Spannungsverteilung wird mit jeweils 25 Integrationspunkten in ξ - und η -Richtung sowie 10 Integrationspunkten in ζ -Richtung beschrieben. Für die Vernetzung mit 879 Elementen wurde zum einen eine trilineare sowie eine quadratische Interpolation in der Ebene als geometrische Beschreibung betrachtet. Hierbei wurde die Distribution der Spannungen auf jeweils 5 Integrationspunkte in der Ebene und 10 Integrationspunkte über die Dicke des Elements interpoliert. Schließlich soll als dritte Diskretisierung ein Netz mit 3501 Hexaederelementen mit einer Spannungsverteilung an 3 Gaußpunkten in der Ebene sowie 10 Integrationspunkten über die Dicke betrachtet werden. Die Geometrie wird dabei aufgrund des geringen Elementdurchmessers trilinear beschrieben.

In Abbildung 6.18 sind die einzelnen Konvergenzraten für den relativen Fehler in der Energienorm in einem doppelt logarithmischen Maßstab über die Anzahl der Freiheitsgrade aufgetragen. Dabei ist für jedes der betrachteten Netze eine Abnahme des Fehlers mit zunehmenden Polynomgrad ersichtlich. Desweiteren ist aus der parallelen Verschiebung der Linien die

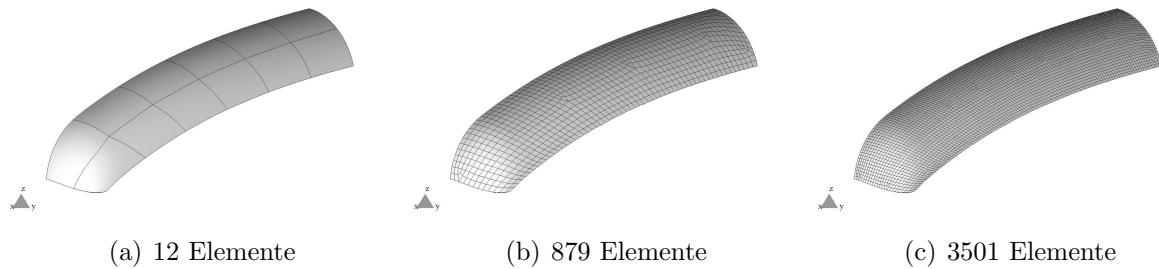


Abbildung 6.17: Diskretisierungen mit Volumenelementen

schnellere Konvergenz der p -Version ersichtlich. Die effizienteste Diskretisierung stellt dabei das Netz mit 12 Elementen dar. Der Einfluss der geometrischen Interpolation mithilfe der quasi-regionalen Abbildung ist beim relativen Fehler in der Energienorm deutlich zu erkennen. Gerade bei niedrigen geometrischen Polynomgraden p_{geo} stellt sich ab einem gewissen Polynomgrad der Verschiebung keine weitere Steigerung der Genauigkeit mehr ein. Dies ist deutlich am horizontal asymptotischen Verlauf der Konvergenzkurve zu erkennen, welcher auf einen konstanten Fehler in der Energienorm hinweist und auf den Geometriefehler zurückgeführt werden kann.

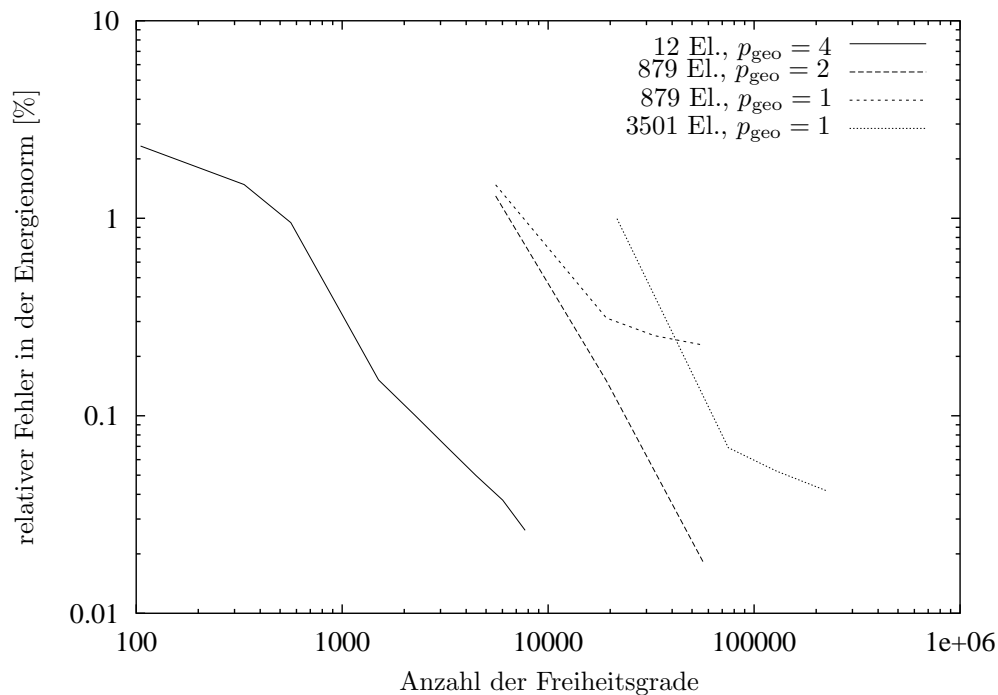


Abbildung 6.18: Konvergenz im Fehler der Energienorm

Aus den betrachteten Diskretisierungen lässt sich die Effizienz der p -Version belegen. In Abbildung 6.18 ist dargestellt, dass mit einer geringen Anzahl an Elementen bei einem gewissen Polynomgrad eine gleiche Güte der Ergebnisqualität wie bei der h -Version mit einer erheblich höheren Anzahl an Freiheitsgraden erreicht werden kann. Die Referenzlösung wurde dabei auf

einen feinen Netz mit einem hohen Polynomgrad ermittelt. Desweiteren wurde gezeigt, dass ein entsprechend hoher Polynomgrad der geometrischen Beschreibung eine gute Approximation erwarten lässt. Allerdings führt die Verwendung eines Polynomgrads der Geometrie, welcher höher ist als der Polynomgrad der Verschiebungen zu superparametrischen Elementen, welche wiederum zu künstlichen Spannungen und damit zu einer zu schlechten Approximation der Verschiebungen führen könnten.

In Abbildung 6.19 ist die elastische Rückfederung für einen Polynomgrad $p = 10$, $q = 5$ und einer Diskretisierung mit 12 Elementen dargestellt.

6.4 Längsträger

Das letzte Beispiel stellt ein komplexes, industrielles Bauteil dar, mit dem die Überlegenheit der p -Version der Finite-Elemente-Methode gegenüber der h -Version demonstriert werden soll. Bei diesem Bauteil handelt es sich um die Unterseite eines zweiteiligen Längsträgers. Dieses auch für die passive Sicherheit relevante Bauteil besitzt eine Materialstärke von ca. 1,4 mm, eine Länge von ca. 1,5 m und einen E-Modul von $210000 \frac{\text{N}}{\text{mm}^2}$.

Um eine Hexaeder-Diskretisierung für die elastische Rückfederung zu erstellen, muss zunächst eine Regionseinteilung für das Bauteil im tiefgezogenen Zustand gefunden werden. Hierbei fand die in Kapitel 5.2.2.2 beschriebene Regionsvererbung der Werkzeuggeometrie (siehe Abbildung 6.20) Anwendung.

Dabei werden zunächst rekursiv Patches, welche klein im Verhältnis zum aktuellen Patch sind, zu diesem Patch hinzugefügt. Als weiteres Kriterium für das Zusammenfügen von Werkzeugen dient der Normalenvektor der Patches an charakteristischen Stellen. Liegt die Abweichung des Normalenvektors eines Patches zum aktuellen Patch unter einer bestimmten Grenze, so wird dieses Patch zum aktuellen hinzugefügt. Mithilfe dieses Verfahrens können die entsprechenden Flächenrückführungen und Regionseinteilungen für die Hexaeder-Netzgenerierung vorgenommen werden (siehe Abbildung 6.21).

Auf Basis des aus ca. 130000 Schalenelementen rückgeführten Volumenkörpers wurde ein Netz mit 739 Hexaederelementen erstellt (siehe Abbildung 6.22). Dabei ist sehr gut die Ausrichtung der Elemente an der Regionsberandung zu erkennen. Die Verteilung des Spannungszustands wurde auf jeweils 12 Integrationspunkte in der Ebene und auf 8 über die Dicke eines jeden Elements interpoliert.

6.4.1 Ansatzräume

Zunächst sollen für den Längsträger verschiedene Ansatzräume verglichen werden, um daraus eine Empfehlung für eine optimale Diskretisierung geben zu können. In Abbildung 6.23 ist der relative Fehler in der Energienorm des *trunk space* $\mathcal{S}_{\text{ts}}^{p_\xi, p_\eta, p_\zeta}(\Omega_{\text{st}}^h)$ und *tensor product space* $\mathcal{S}_{\text{ps}}^{p_\xi, p_\eta, p_\zeta}(\Omega_{\text{st}}^h)$ ersichtlich. Dabei wird beim *trunk space* für den gleichen Fehler in der Energienorm stets eine geringere Anzahl an Freiheitsgraden benötigt als beim *tensor product space*.

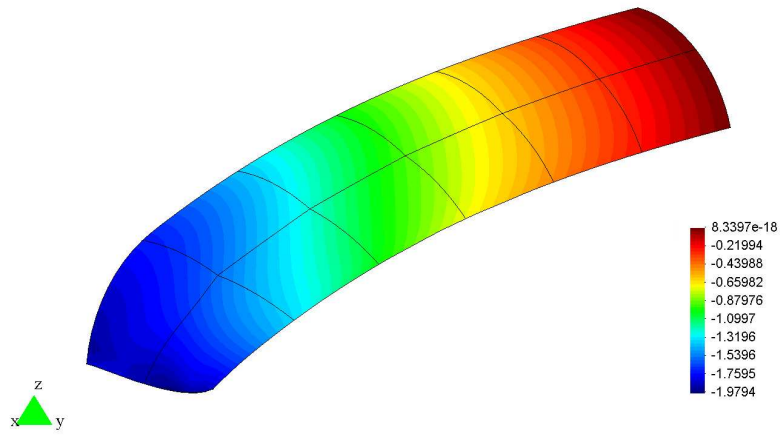
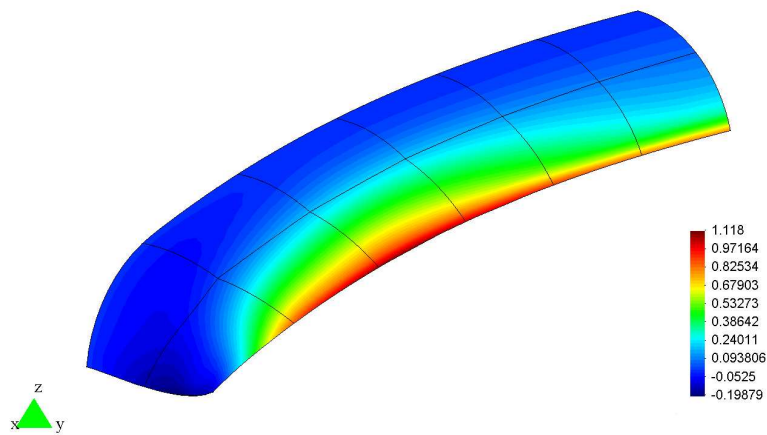
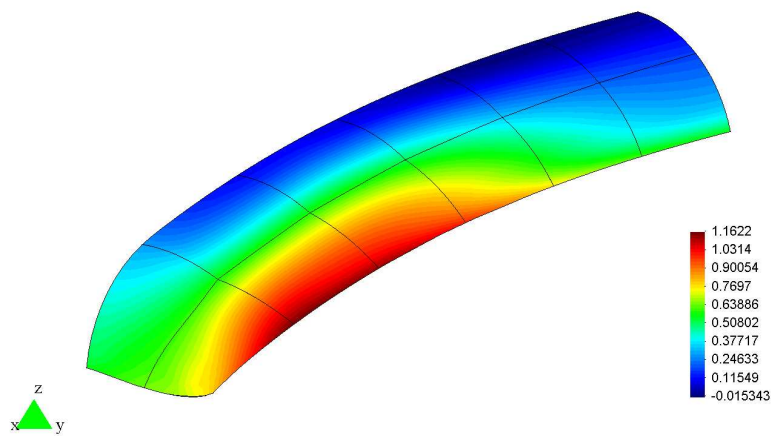
(a) x -Komponente(b) y -Komponente(c) z -Komponente

Abbildung 6.19: Elastische Rückfederung [mm] des Wasserstofftanks

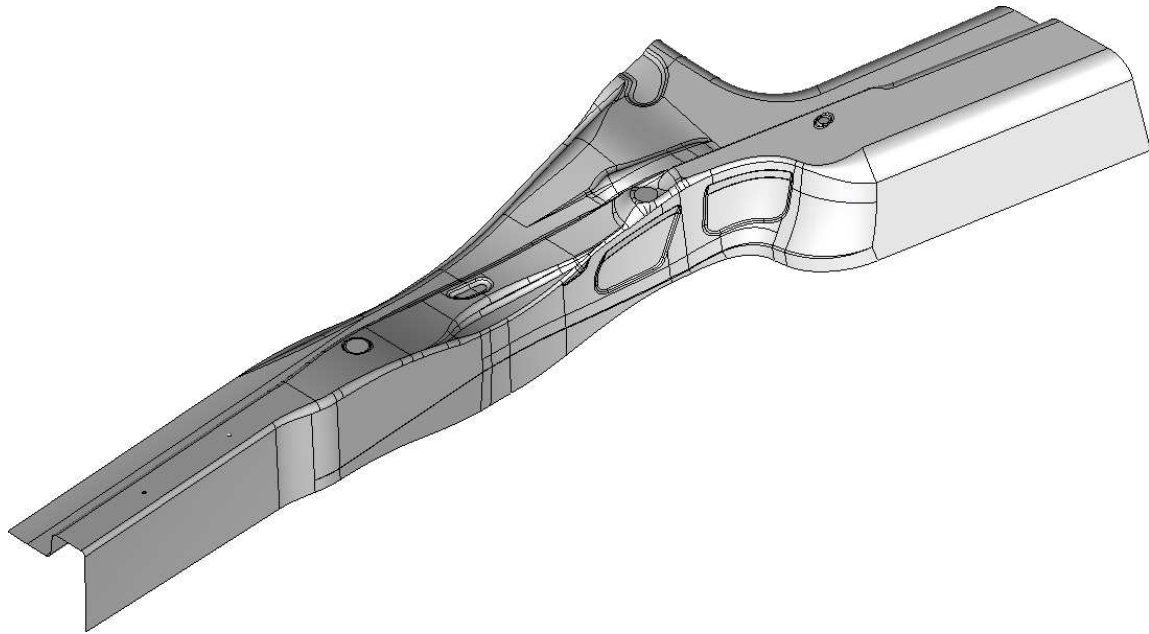


Abbildung 6.20: Werkzeuggeometrie Längsträger

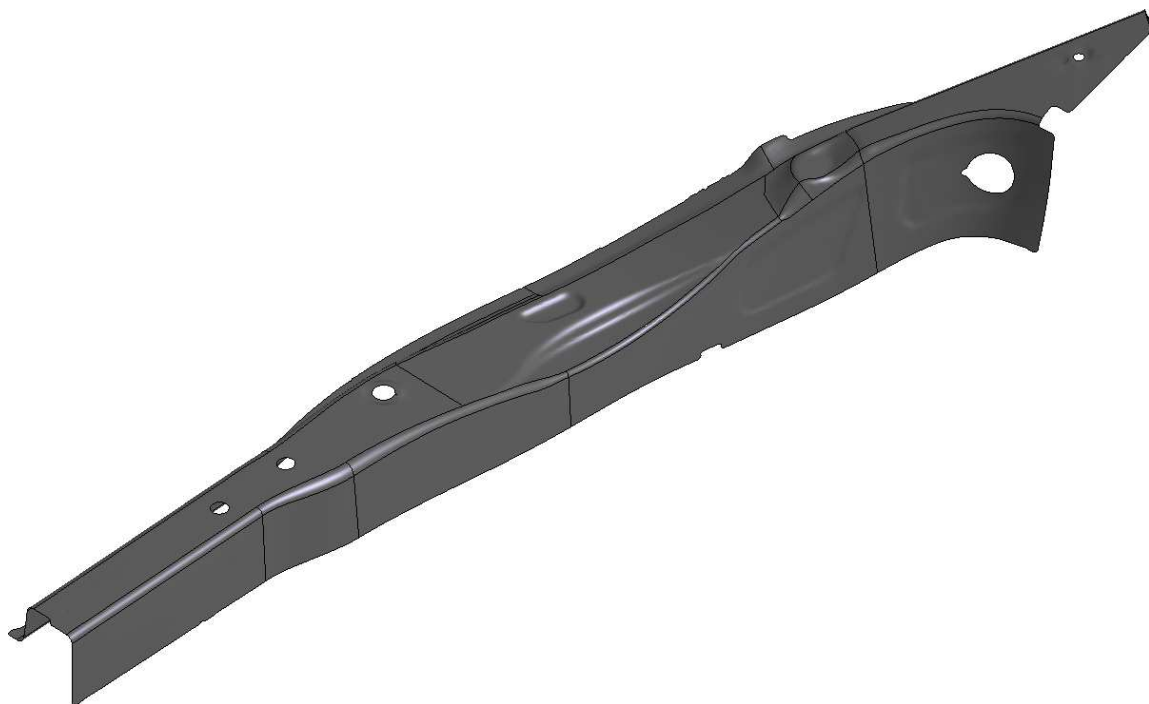


Abbildung 6.21: Rückgeführte Mittelfläche mit Regionseinteilung für Vernetzung

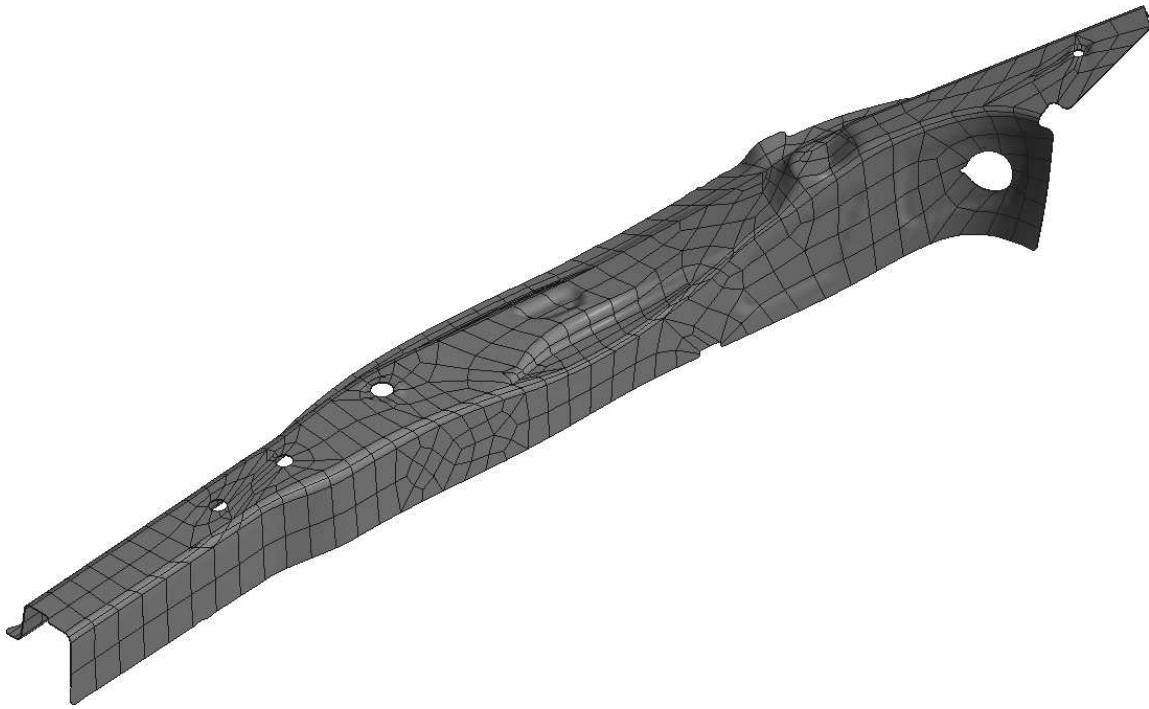


Abbildung 6.22: Rediskretisierte Geometrie mit Hexaederelementen

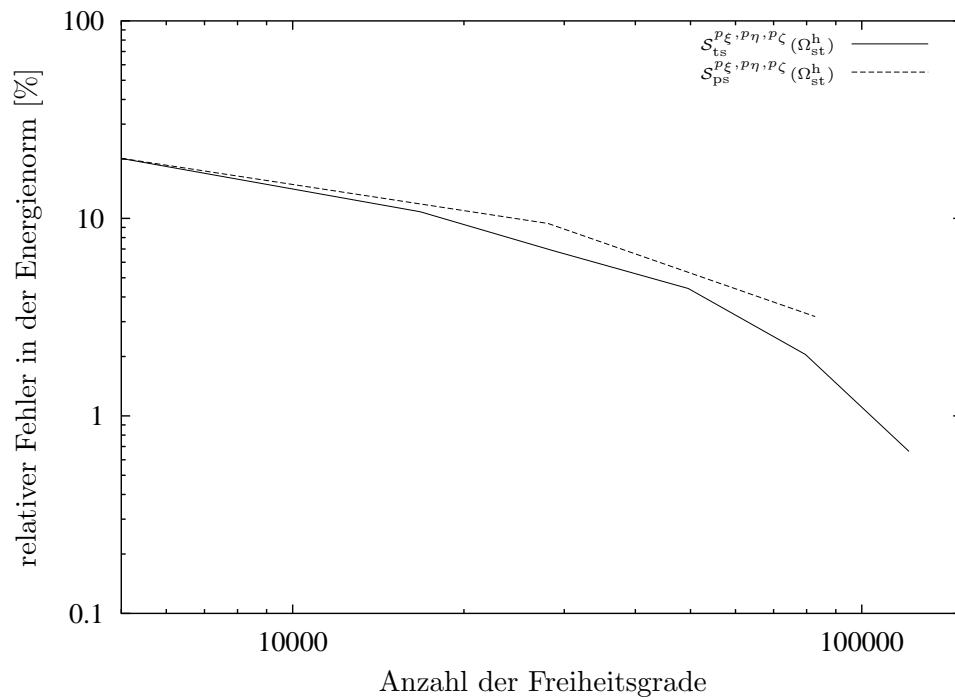


Abbildung 6.23: Relativer Fehler in der Energienorm für $\mathcal{S}_{ts}^{p_{\xi}, p_{\eta}, p_{\zeta}}(\Omega_{st}^h)$ und $\mathcal{S}_{ps}^{p_{\xi}, p_{\eta}, p_{\zeta}}(\Omega_{st}^h)$

Anisotrope Ansatzräume stellen eine effiziente Diskretisierung gegenüber isotropen Räumen dar. Aus Abbildung 6.24 ist dabei ersichtlich, dass der relative Fehler in der Energienorm für einen Polynomgrad $q \leq 3$ über die Dicke gegen einen konstanten Fehler konvergiert. Dies lässt sich darauf zurückführen, dass für $q \leq 3$ der Modellfehler dominiert. Für einen Polynomgrad $q \geq 4$ zeigt sich die bei der p -Version für glatte Probleme erwartete exponentielle Konvergenz. Aus Effizienzgründen sollte die Wahl von q nach oben begrenzt sein. Im vorliegenden Beispiel stellt $q = 4$ den optimalen Polynomgrad über die Dicke dar, da hier für den gleichen relativen Fehler in der Energienorm weniger Freiheitsgrade erforderlich sind als bei einem höheren Polynomgrad. Mit dieser Diskretisierung betrug die Rechenzeit 1521 Sekunden im Vergleich zu 1320 Sekunden bei Berechnung mit dimensionsreduzierten Schalenelementen niedriger Ordnung.

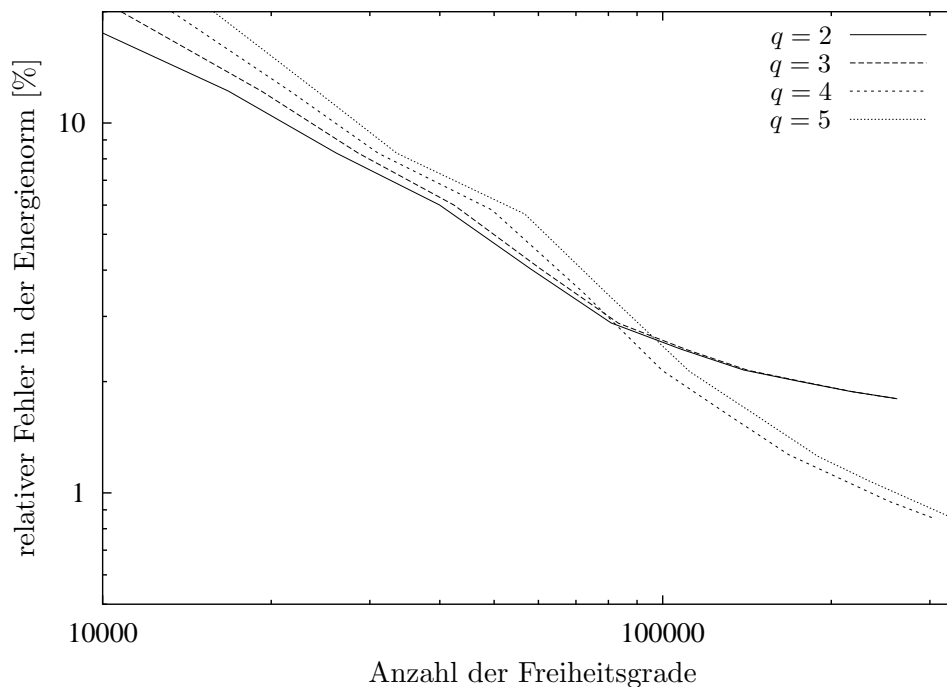


Abbildung 6.24: Relativer Fehler in der Energienorm für anisotrope Ansatzräume

6.4.2 Geometrische Nichtlinearität

Bei der geometrisch nichtlinearen Berechnung für den Längsträger wurde das Newton-Raphson-Verfahren mit einem Lastschritt angewendet. In Abbildung 6.25 sind die Konvergenzraten aufgetragen. Daraus ist ersichtlich, dass das Verfahren mit Vernachlässigung der geometrischen Steifigkeit mit ungefähr der Hälfte an Iterationsschritten wie bei Berücksichtigung von \mathbf{K}_σ konvergiert. Durch den geringeren Aufwand an Multiplikationen ergibt sich ein Vorteil in der Rechenzeit im Verhältnis von etwa 4.

Aus Abbildung 6.26 wird ersichtlich, dass die Orte der maximalen Verschiebungen der geometrisch nichtlinearen Berechnung der elastischen Rückfederung und der geometrisch linearen

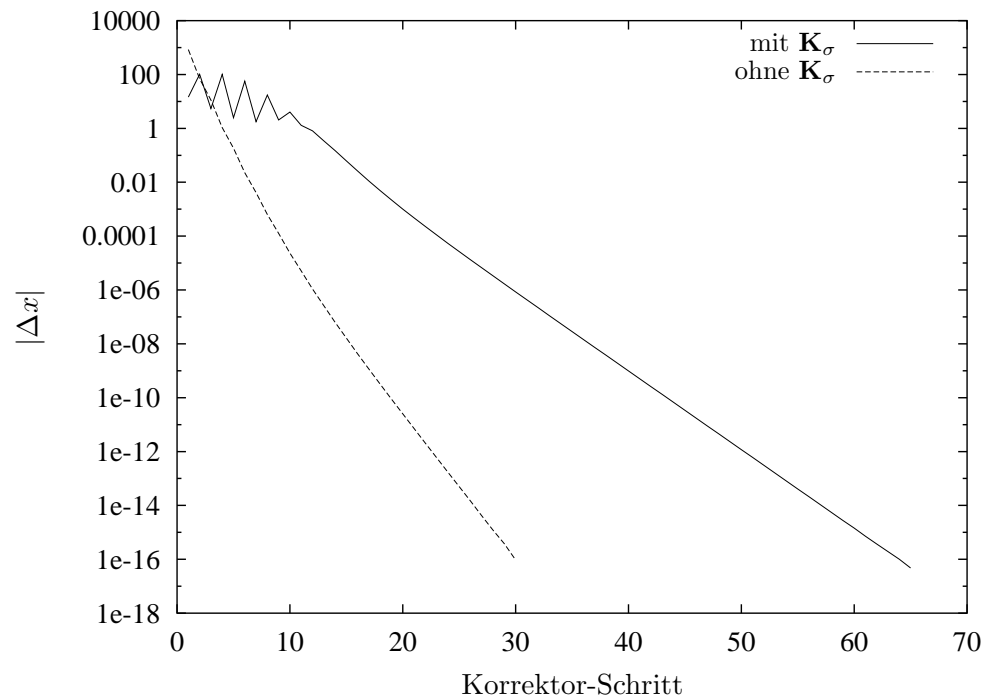


Abbildung 6.25: Konvergenz der geometrisch nichtlinearen elastischen Rückfederung

Lösung (Abbildung 6.27) voneinander abweichen. Bezogen auf den Ergebnis der linearen Berechnung ergibt sich eine maximale Zunahme der Verschiebungen von mehr als 44 % (8.29 mm zu 11.99 mm).

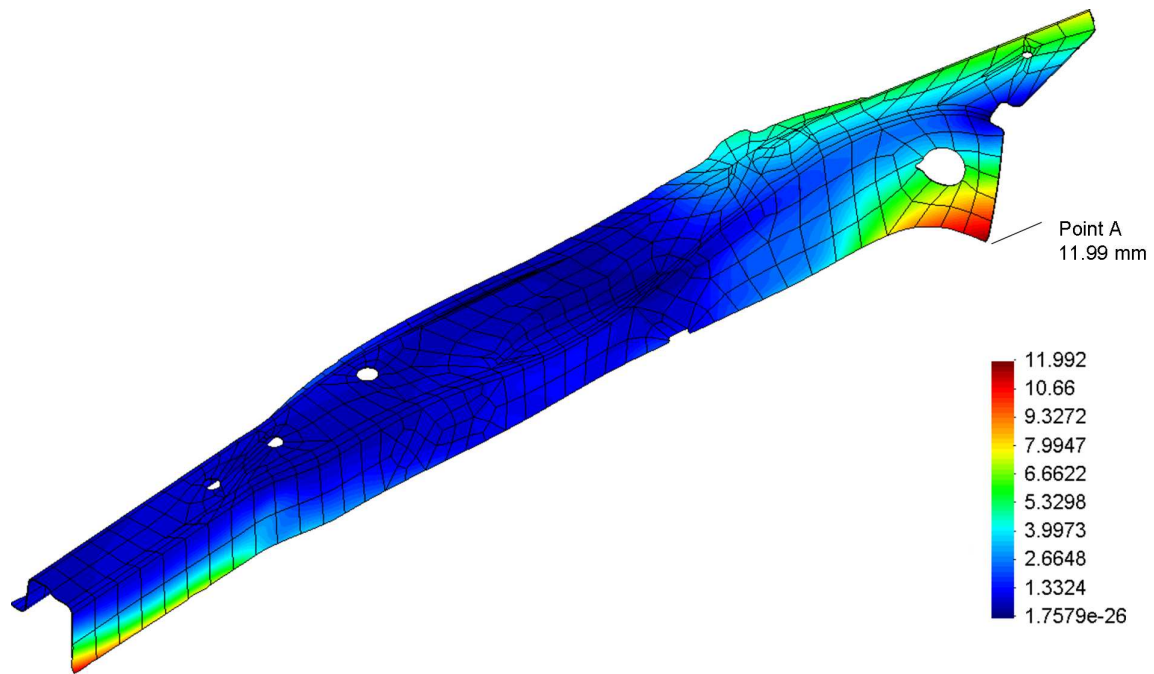


Abbildung 6.26: Geometrisch nichtlineare Rückfederung

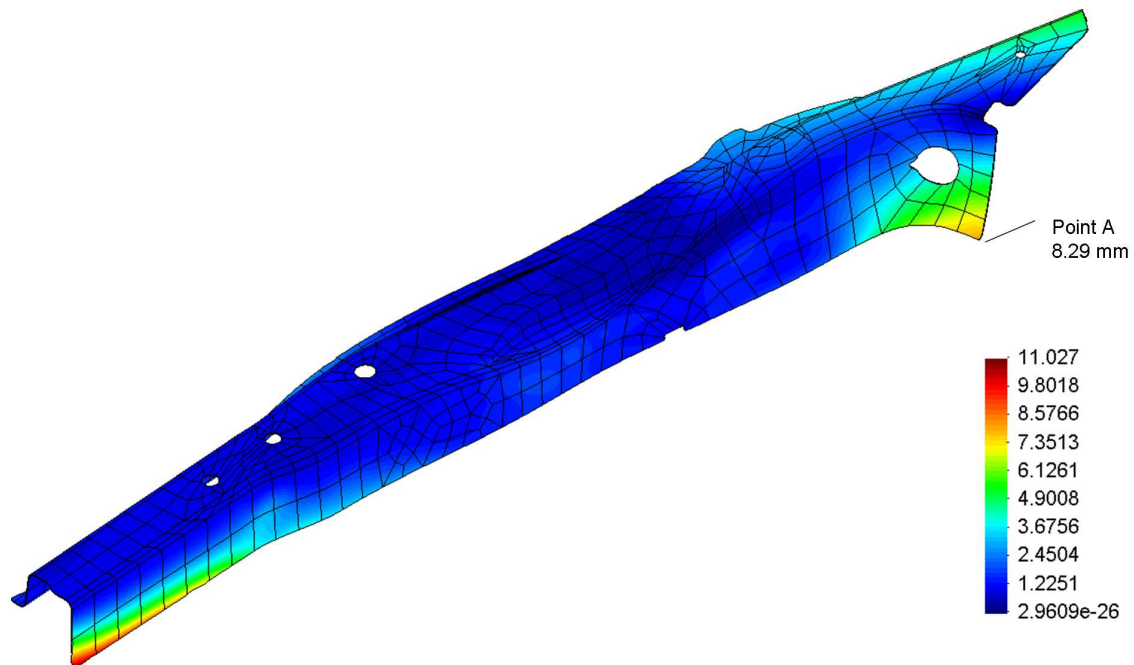


Abbildung 6.27: Geometrisch lineare Rückfederung

Kapitel 7

Zusammenfassung

Die Finite-Elemente-Methode stellt das wichtigste Berechnungsverfahren zur virtuellen Prozessabsicherung in der Umformtechnik dar. Während im Bereich der Tiefziehsimulation mit kommerziellen Applikationen zuverlässige Ergebnisse erzielt werden können, stellt die Berechnung der elastischen Rückfederung ein noch nicht zufrieden stellend gelöstes Problem dar. Berechnungen mit expliziten Verfahren führen häufig aufgrund der fehlenden Gleichgewichtsiteration zu ungenauen Ergebnissen. Implizit durchgeführte Rückfederungsberechnungen dagegen stellen häufig einen erheblichen Bedarf an Speicherkapazität. Weiterhin können die dimensionsreduzierten Modelle die Spannungszustände aufgrund des Modellfehlers in kritischen Bereichen wie engen Radien nicht korrekt abbilden.

Ausgehend von den genannten Problemen wurde im Rahmen dieser Arbeit die Berechnung der elastischen Rückfederung mit einem strikt dreidimensionalen Ansatz hoher Ordnung durchgeführt. Die p -Version der Finite-Elemente-Methode ermöglicht bereits heute auf verfügbarer Hardware Probleme mit Volumenelementen zu berechnen, welche sich bei der h -Version nur mit den gewöhnlichen dimensionsreduzierten Ansätzen lösen lassen. Durch die hohe Konvergenzrate der p -Version bei glatten Lösungsverläufen ist es möglich, eine Lösung gleicher Güte mit einer niedrigeren Anzahl an Freiheitsgraden als bei der h -Version zu erreichen. Dabei liegen die Rechenzeiten in einer vergleichbaren Größenordnung, obwohl die Berechnung bei der p -Version mit Volumenelementen durchgeführt wird.

Durch die Verwendung einer genauen Geometriebeschreibung für die p -Version mithilfe der Blending-Funktionen-Methode und der quasi-regionalen Abbildung wird der Diskretisierungsfehler kontrolliert. Die in diesem Zusammenhang entwickelte Flächenrückführung mit Regionseinteilung auf Basis projizierter Patches der Werkzeuge ermöglicht eine optimale Diskretisierungsstrategie für die p -Version der Finite-Elemente-Methode. Weiterhin ist es bei dieser Vorgehensweise möglich, die Geometrie innerhalb der gesamten Prozesskette beizubehalten und dem Konstrukteur für Modifikationen auf der Ebene einzelner Patches innerhalb der Prozess-Iteration zugänglich zu machen.

Mithilfe der entwickelten Kopplung eines Netzgenerators für Schalenprobleme an den entwickelten Präprozessor ist es möglich, für dreidimensionale Körper über eine bijektive Abbildung vom Parameter- in den euklidischen Raum \mathbb{R}^3 und eine Projektion der linearen Kanten auf die berandende Geometrie ein Netz mit gekrümmten Hexaederelementen für die p -Version

der Finite-Elemente-Methode zu erzeugen.

Anisotrope Ansatzfunktionen ermöglichen die Kontrolle des Modellfehlers durch die Definition des Polynomgrads über die Elementdicke. Im Rahmen dieser Arbeit wurden Untersuchungen zur Bestimmung des optimalen Polynomgrads in Hinblick auf Effizienz und Genauigkeit gemacht. Weiterhin wurde ein adaptives Verfahren auf Basis eines hierarchischen Fehlerschätzers zur Bestimmung der optimalen, elementweisen Polynomgradverteilung angewendet.

Im Rahmen der Untersuchung der geometrischen Nichtlinearität bei der elastischen Rückfederung wurde ein Verfahren zur Beschleunigung der Konvergenzgeschwindigkeit statisch bestimmter Systeme verwendet und untersucht. Insbesondere bei großen Lastschritten zeigte die neue Vorgehensweise einen hohen Grad an Effizienz.

Durch den Einsatz hochfester Stähle mit hoher Streckgrenze oder Aluminiumwerkstoffen im Automobilbau und der damit verbundenen Zunahme elastischer Spannungen gewinnt die Bestimmung der erforderlichen Pressenkraft zunehmend an Bedeutung. Es wurde ein Ansatz entwickelt, der es erlaubt aus dem Spannungszustand dimensionsreduzierter Modelle einen vollständigen dreidimensionalen Spannungszustand zu ermitteln. Über eine anschließende Oberflächenintegration lassen sich die erforderlichen Pressenkräfte bestimmen. In einem Anschlussprojekt soll der genannte Ansatz zu einer zuverlässigen Bestimmung der Pressenkraft weiterentwickelt werden. Dazu müssen unter anderem entsprechende Verschiebungsrandbedingungen, Diskretisierungs- und Integrationsstrategien entwickelt werden. Weiterhin lässt sich durch Berechnung der elastischen Rückfederung mit den korrigierten Spannungen der Einfluss der Komponenten σ_{xz} , σ_{yz} , σ_{zz} des Spannungszustands σ auf die Rückfederung beurteilen.

Im Rahmen dieses Nachfolgeprojekts soll auch die bestehende Implementierung der Flächenrückführung weiterentwickelt werden. Die Möglichkeit der Glättung der Geometrie bzw. Substitution durch analytische Flächenbeschreibungen würde dabei zum einen zu einer Reduktion der Datenmenge führen. Zum anderen würde sich die Anzahl der durch Oszillationen hervorgerufen „Kontaktinseln“ reduzieren. Weiterhin könnte untersucht werden, inwieweit sich die Oberflächenqualität durch Einsatz von Patch-weisen Beziérflächen verändert. Bei der Diskretisierungstechnik könnte eine Kopplung von Makro- mit Freivernetzern zu einer weiteren Reduzierung der Anzahl an Hexaederelementen und damit zu einer noch effizienteren Diskretisierung für die p -Version der Finite-Elemente-Methode führen.

Literaturverzeichnis

- [1] <http://netlib.bell-labs.com/netlib/linalg/spooles/spooles.2.2.html>.
- [2] S. Abramowski und H. Müller. *Geometrisches Modellieren*. BI Wissenschaftsverlag, 1991.
- [3] E.L. Anagnostou and J.M. Papazian. *Optimized Tooling Design Algorithm for Sheet Metal Forming over Reconfigurable Compliant Tooling*. In Somnath Ghosh et al. (editor): *Proceedings NUMIFORM 2004, Numerical Methods in Industrial Forming Processes*, pages 741–748, 2004.
- [4] Cleve Ashcraft and Roger Grimes. *Spooles: An Object-Oriented Sparse Matrix Library*. In *Proceedings of the 1999 SIAM Conference on Parallel Processing for Scientific Computing*, 1999.
- [5] I. Babuška and B. Szabó. *On the rates of convergence of the finite element method*. *International Journal for Numerical Methods in Engineering*, 18:323–341, 1982.
- [6] R.E. Bank. *PLTMG, A software package for solving elliptic partial differential equations*. *SIAM Journal of Numerical Analysis*, 7, 1990.
- [7] K.J. Bathe. *Finite-Elemente-Methoden*. Springer-Verlag, 2002.
- [8] C. Brix. *A three-dimensional p-version approach for elastic spring back simulation in deep drawing processes*. Master thesis, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2001.
- [9] H. Bröker. *Integration von geometrischer Modellierung und Berechnung nach der p-Version der FEM*. Doktorarbeit, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2001.
- [10] H.-J. Bungartz, M. Griebel und Ch. Zenger. *Einführung in die Computergraphik*. Vieweg Verlag, 1996.
- [11] F. Buschmann. *Pattern-orientierte Software-Architektur . Ein Pattern-System*. Addison-Wesley, 1998.
- [12] J. Canny. *A Computational Approach to Edge Detection*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
- [13] Q. Chen and I. Babuška. *Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle*. *Computer Methods in Applied Mechanics and Engineering*, 128:405–417, 1995.

- [14] Q. Chen and I. Babuška. *The optimal symmetrical points for polynomial interpolation of real functions in the tetrahedron*. Computer Methods in Applied Mechanics and Engineering, 137:89–94, 1996.
- [15] J. Corney. *3D modeling with the ACIS kernel and toolkit*. John Wiley & Sons, 1997.
- [16] M.A. Crisfield. *Non-linear finite element analysis of solids and structures, Volume 1*. John Wiley & Sons, 1991.
- [17] M.A. Crisfield. *Non-linear finite element analysis of solids and structures, Volume 2*. John Wiley & Sons, 2003.
- [18] C. De Boor. *A Practical Guide to Splines*. Springer-Verlag, 1978.
- [19] A. Dombrowski. *Implementation und numerische Untersuchung der geometrischen Nicht-linearität für dreidimensionale finite Elemente hoher Ordnung*. Diplomarbeit, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2004.
- [20] A. Düster. *High order finite elements for three-dimensional, thin-walled nonlinear continua*. Doktorarbeit, Lehrstuhl für Bauinformatik, Bauingenieur- und Vermessungswesen, Technische Universität München, 2001.
- [21] A. Düster, H. Bröker, and E. Rank. *The p-version of the finite element method for three-dimensional curved thin walled structures*. International Journal for Numerical Methods in Engineering, 52:673–703, 2001.
- [22] A. Düster and E. Rank. *The p-version of the finite element method compared to an adaptive h-version for the deformation theory of plasticity*. Computer Methods in Applied Mechanics and Engineering, 190:1925–1935, 2001.
- [23] A. Düster, A. Niggel, and E. Rank. *Thermo-elastic computations of geometrically non-linear three-dimensional thin-walled continua based on high order finite elements*. In *Proceedings of WCCM V, Fifth World Congress on Computational Mechanics*, Vienna, Austria, 2002.
- [24] G. Farin. *Curves and surfaces for computer aided geometric design*. Academic Press, 4. edition, 1997.
- [25] Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard*. International Journal of Supercomputer Applications, 8(3/4), 1994.
- [26] J. Fünfänger. *Development of an automated procedure to transfer measuring data onto finite element meshes*. Master thesis, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2004.
- [27] J. Fukuda and J. Suhara. *Automatic mesh generation for finite element analysis*. Advances in Computational Methods in Structural Mechanics and Design, 1992.

- [28] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, M. Booth, and F. Rossi. *GNU Scientific Library Reference Manual - Second Edition*. Network Theory Limited, Bristol, UK, 2002.
- [29] E. Gamma, R. Helm und R. Johnson. *Entwurfsmuster. Elemente wiederverwendbarer objektorientierter Software*. Addison-Wesley, 2004.
- [30] P.L. George. *Automatic Mesh Generation – Application to Finite Element Methods*. John Wiley & Sons, Chichester, 1991.
- [31] W.J. Gordon and Ch.A. Hall. *Construction of curvilinear co-ordinate systems and applications to mesh generation*. International Journal for Numerical Methods in Engineering, 7:461–477, 1973.
- [32] W.J. Gordon and Ch.A. Hall. *Transfinite element methods: Blending function interpolation over arbitrary curved element domains*. Numerische Mathematik, 21:109–129, 1973.
- [33] M. Gruber. *Simulation der elastischen Rückfederung bei einfachen Tiefziehbteileilen mit Hilfe der p -Version der FEM*. Diplomarbeit, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2000.
- [34] A. Halfmann. *Ein geometrisches Modell zur numerischen Simulation der Fluid-Struktur-Interaktion windbelasteter, leichter Flächentragwerke*. Doktorarbeit, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2002.
- [35] E. Haug, J. Clinckemaillie, and F. Aberlenc. *Computational mechanics in crashworthiness analysis*. PAM–STAMP Technical Papers, PAM System International, 1999.
- [36] I. Heinle. *Automatische Verbesserung von Werkzeugziehanlagen durch Aufsprungsimulation*. Diplomarbeit, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2002.
- [37] S. Holzer, E. Rank, and H. Werner. *An implementation of the hp -version of the finite element method for Reissner-Mindlin plate problems*. International Journal for Numerical Methods in Engineering, 30:459–471, 1990.
- [38] S.M. Holzer. *Was kann ein Finite-Element-Programm für Platten in der Praxis leisten?* In: P. Wriggers, U. Meißner, E. Stein und W. Wunderlich (Herausgeber): *Finite Elemente in der Baupraxis*, Seiten 441–450. Ernst & Sohn, 1998.
- [39] S.M. Holzer. *Modellierung und Berechnung mit der p -Version der FEM*. Preprint, Informationsverarbeitung im Konstruktiven Ingenieurbau, Universität Stuttgart, 1999.
- [40] S.M. Holzer and C. Haußer. *Ergebnisverbesserung von Finite-Element-Berechnungen durch exakte Geometriemodellierung*. Forschung im Ingenieurwesen, 56(3):65–70, 1990.
- [41] B. Jähne. *Digital Image Processing: Concepts, Algorithms and Scientific Applications*. Springer-Verlag, 2002.

- [42] A.P. Karafillis and M.C. Boyce. *Tooling design in sheet metal forming using springback*. Int. J. Mach. Tools. Manuf., 34(113), 1992.
- [43] A. Kela, R. Perucchio, and H.B Voelcker. *Toward automatic finite element analysis*. Computer Methods in Applied Mechanics and Engineering, pages 57–71, 1986.
- [44] G. Királyfalvi and B.A. Szabó. *Quasi-regional mapping for the p-version of the finite element method*. Finite Elements in Analysis and Design, 27:85–97, 1997.
- [45] R. Krause and E. Rank. *A fast algorithm for point-location in a finite element mesh*. Computing, 57/1:49–63, 1996.
- [46] K. Lange. *Umformtechnik. Handbuch für Industrie und Wissenschaft. Band 1: Grundlagen*. Springer-Verlag, 2002.
- [47] J.K. Lee et al. (eds.). *Proceedings of the 3rd International Conference on Numerical Simulations of 3-D Sheet Metal Forming Processes*. Dearborn, Michigan, 1996.
- [48] S.H Lo. *A new mesh generation scheme for arbitrary planar domains*. International Journal for Numerical Methods in Engineering, 21:1403–1426, 1985.
- [49] J. Meinhardt et al.. *Virtuelle Entwicklung von Presswerkzeugen im industriellen Umfeld*. In: E. Doege (Herausgeber): *Umformtechnik – Erschließung wirtschaftlicher und technologischer Potenziale 17. Umformtechnisches Kolloquium Hannover*, 2002.
- [50] M. Mäntylä. *An Introduction to Solid modeling*. Computer Science Press, 1988.
- [51] A. Muthler, A. Düster, W. Volk, M. Wagner, and E. Rank. *High order finite elements applied to the computation of elastic spring back in sheet metal forming*. In Somnath Ghosh et al. (editor): *Proceedings NUMIFORM 2004, Numerical Methods in Industrial Forming Processes*, pages 946–950, 2004.
- [52] V. Nübel. *Die adaptive rp-Methode für elastoplastische Probleme*. Dissertation, Lehrstuhl für Bauinformatik, Fakultät für Bauingenieur- und Vermessungswesen, Technische Universität München, 2005.
- [53] A. Niggl, A. Düster, and E. Rank. *Coupling 1d and 2d elasticity problems by using the hp-d version of the finite element method*. In *Proceedings of the Second M.I.T. Conference on Computational Fluid and Solid Mechanics*, Boston, USA, 2003.
- [54] J. Peraire, J. Peiro, L. Formaggia, K. Morgan, and O.C. Ziekiewicz. *Finite element Euler computations in three dimensions*. International Journal for Numerical Methods in Engineering, 26:2135–2159, 1988.
- [55] J. Peraire, M. Vahdati, K. Morgan, and O.C. Ziekiewicz. *Adaptive remeshing for compressible flow computations*. Journal of Computational Physics, 72:449–466, 1987.
- [56] K. Pöhlandt. *Testing strain-rate sensitive materials in the torsion test*. Materialprüfung, 22:399–406, 1980.
- [57] L. Piegl and W. Tiller. *The Nurbs Book*. Springer-Verlag, 2. edition, 1997.

- [58] E. Rank. *A-posteriori-Fehlerabschätzungen und adaptive Netzverfeinerung für Finite-Element- und Randintegralelement-Methoden*. Doktorarbeit, Elektronisches Rechnen im Konstruktiven Ingenieurbau, Bauingenieur- und Vermessungswesen, Technische Universität München, 1985.
- [59] E. Rank, H. Bröker, A. Düster, R. Krause, and M. Rucker. *The p-version of the finite element method for structural problems*. In E. Stein (editor): *Error-Controlled Adaptive Finite Elements in Solid Mechanics*. John Wiley & Sons, 2002.
- [60] E. Rank, H. Bröker, A. Düster, and V. Nübel. *High order solid elements for thin-walled structures: No tricks? - no crimes!* In W.A. Wall, K.-U. Bletzinger, and K. Schweizerhof (editors): *Trends in computational structural mechanics*, CIMNE, Barcelona, Spain, 2001.
- [61] E. Rank, H. Bröker, A. Düster und M. Rucker. *Neue Modellierungskonzepte für FE-Berechnungen mit Ansätzen höherer Ordnung*. In: P. Wriggers, U. Meißner, E. Stein und W. Wunderlich (Herausgeber): *Finite Elemente in der Baupraxis - Modellierung, Berechnung und Konstruktion*, Seiten 421–430. Ernst & Sohn, 1998.
- [62] E. Rank, H. Bröker, A. Düster und M. Rucker. *Modellieren und Berechnen mit der p-Version der FEM*. In: C. Katz und B. Protosaltis (Herausgeber): *SOFISTIK, Software für Statik und Konstruktion 3*, Seiten 185–192, A.A. Balkema, Rotterdam, 2000.
- [63] E. Rank and A. Düster. *Dimensional adaptivity using the p-version of the finite element method*. In *Proceedings of the European Conference on Computational Mechanics 2001*, Cracow, Poland, 2001.
- [64] E. Rank, A. Düster, and Z. Alam. *Adaptive computation of thin-walled structures based on high order solid finite elements*. In *Proceedings the WCCM VI in conjunction with APCOM'04*, Beijing, China, 2004.
- [65] E. Rank, A. Düster, A. Muthler, and R. Romberg. *High order solid elements for thin-walled structures with applications to linear and non-linear structural analysis*. In P. Neittaanmäki (editor): *Proceedings of ECCOMAS 2004, European Congress on Computational Methods in Applied Sciences and Engineering*, Jyväskylä, Finland, 2004.
- [66] E. Rank, M. Rucker und M. Schweingruber. *Automatische Generierung von Finite-Element-Netzen*. Bauingenieur, 69(10):373–379, 1994.
- [67] M. Rucker. *A parallel p-version finite element approach for civil engineering and structural mechanics problems*. Doktorarbeit, Lehrstuhl für Bauinformatik, Bauingenieur- und Vermessungswesen, Technische Universität München, 2000.
- [68] P. Saint-George, G. Warzee, B. Beauwens, and Y. Notay. *High-performance pcg solvers for FEM structural analysis*. International Journal for Numerical Methods in Engineering, 39:1313–1340, 1996.
- [69] H.R. Schwarz. *Methode der Finiten Elemente*. B.G. Teubner, 3. edition, 1991.

- [70] M. Schweingruber-Straten. *Generierung von Oberflächennetzen nach der Gebietsteilungstechnik*. PhD thesis, Fach Numerische Methoden und Informationsverarbeitung, Universität Dortmund, 1999.
- [71] J. Shah and M. Mänty. *Parametric and feature-based CAD/CAM: concepts, techniques and applications*. John Wiley & Sons, 1995.
- [72] SOFiSTiK AG. *CDBASE – Help and documentation*. München, 2004.
- [73] Spatial Technology Inc.. *ACIS 3D Toolkit – Getting Started Guide*. Spatial Technology, USA, 1999.
- [74] Spatial Technology, Inc.. *ACIS 3D Toolkit – Technical Overview*. Spatial Technology, USA, 1999.
- [75] B.A. Szabó. *Estimation and control of error based on p convergence*. In I. Babuška, O.C. Zienkiewicz, J. Gago, and E.R. de A. Olivera (editors): *Accuracy estimates and adaptive refinements in finite element computations*, pages 61–70. John Wiley & Sons, 1986.
- [76] B.A. Szabó. *Mesh design for the p -version of the finite element method*. *Computer Methods in Applied Mechanics and Engineering*, 55:181–197, 1986.
- [77] B.A. Szabó and I. Babuška. *Finite element analysis*. John Wiley & Sons, 1991.
- [78] B.A. Szabó, A. Düster, and E. Rank. *The p -version of the finite element method*. In E. Stein, R. de Borst, and T.J.R. Hughes (editors): *Encyclopedia Of Computational Mechanics*, chapter 5, pages 119–139. John Wiley & Sons, Ltd, Chichester, 2004.
- [79] R. Wagoner. *Fundamental aspects of springback in sheet metal forming*. In D. Yang et al. (editor): *Proceedings Numisheet 2002, International Conference and Workshop on Numerical Simulation of 3D Sheet Metal Forming Processes*, pages 13–19, 2002.
- [80] R. Wagoner. *Design of sheet forming dies for springback compensation*. In V. Brucato (editor): *Proceedings ESAFORM 2003, European Scientific Association for material FORMing*, pages 7–14, 2003.
- [81] W.A. Wall, M. Gee, and E. Ramm. *The challenge of a three-dimensional shell formulation, the conditioning problem*. In *Proceedings of ECCM '99, European Conference on Computational Mechanics*, München, Germany, 1999.
- [82] J. Weiher, B. Rietman, K. Kose, S. Ohnimus, and M. Petzoldt. *Controlling springback with compensation strategies*. In Somnath Ghosh et al. (editor): *Proceedings NUMIFORM 2004, Numerical Methods in Industrial Forming Processes*, pages 1011–1015, 2004.
- [83] P. Wriggers. *Nichtlineare Finite-Element Methoden*. Springer-Verlag, 2001.
- [84] O.C. Zienkiewicz and D.V. Phillips. *An automatic mesh generation scheme for plane and curved surfaces by isoparametric co-ordinates*. *International Journal for Numerical Methods in Engineering*, 3:519–528, 1971.
- [85] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method – Basic Formulations and Linear Problems*, volume 1. McGraw-Hill Book Company, 5th edition, 2000.