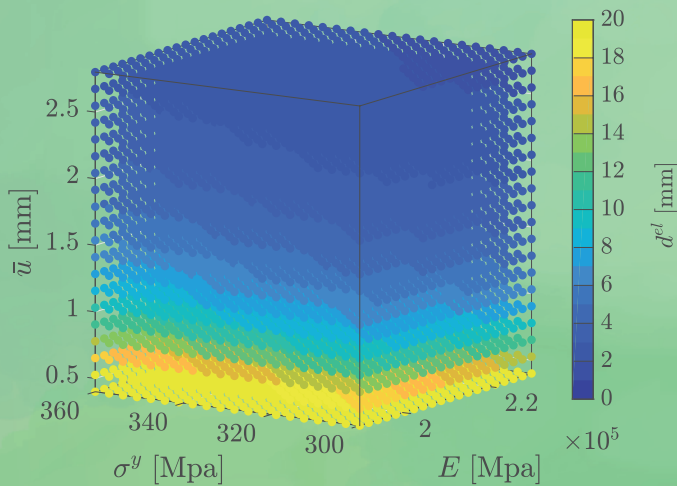


Support Vektor Regression für Anwendungen im Bereich der Elasto-Plastizität



Support Vektor Regression für Anwendungen im Bereich der Elasto-Plastizität

Von der

Fakultät für Bauingenieurwesen und Geodäsie
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation von

Steffen Funk, M.Eng.

Erscheinungsjahr 2022

Referent:

Prof. Dr.-Ing. U. Nackenhorst

Korreferent:

Prof. Dr.-Ing. habil. Dr. h.c. mult.

Dr.-Ing. E.h. P. Wriggers

Tag der Promotion:

14. April 2022

Herausgeber:

Prof. Dr.-Ing. U. Nackenhorst

Verwaltung:

Institut für Baumechanik und

Numerische Mechanik

Gottfried Wilhelm Leibniz Universität

Hannover

Appelstr. 9A

30167 Hannover

Tel.: +49 (0)511 / 762-3560

Fax.: +49 (0)511 / 762-19053

Steffen Funk

Institut für Baumechanik und

Numerische Mechanik

Gottfried Wilhelm Leibniz Universität

Hannover

Appelstr. 9A

30167 Hannover

Alle Rechte, insbesondere das der
Übersetzung in fremde Sprachen,
Vorbehalten. Ohne Genehmigung
des Autors ist es nicht gestattet,
dieses Heft ganz oder teilweise auf
photomechanischem, elektronischem
oder sonstigem Wege zu vervielfältigen

Typeset in FreeSerif with \LaTeX

ISBN 978-3-935732-55-0

Support Vektor Regression für Anwendungen im Bereich der Elasto-Plastizität

Von der Fakultät für Bauingenieurwesen und Geodäsie der Gottfried Wilhelm Leibniz
Universität Hannover

zur Erlangung des Grades eines
Doktor-Ingenieurs

eingereicht von
Steffen Funk M.Eng.

eingereicht am
20. April 2022

Hauptreferent: Prof. Dr.-Ing. Udo Nackenhorst
Korreferent: Prof. Dr.-Ing. habil. Dr. h.c. mult. Dr.-Ing. E.h. Peter Wriggers

Danksagung

Die vorliegende Dissertation entstand im Rahmen meiner Assistenzpromotion am Institut für Baumechanik und numerische Mechanik (IBNM) der Gottfried Wilhelm Leibniz Universität Hannover. Seit meiner Studienzeit sind für mich die Mechanik und die Finite Elemente Methode (FEM) Themen, denen ich mit Enthusiasmus nachgehe. Der weiteren Vertiefung meiner Kenntnisse und der Forschung im Bereich der Festkörpermechanik und der FEM konnte ich mich erstmals uneingeschränkt während der Zeit am Institut widmen. Das hat mich mit Freude und Dankbarkeit erfüllt. Mir ist es daher an dieser Stelle eine Herzensangelegenheit mich bei allen, die mich auf diesem Weg begleitet haben, herzlich zu bedanken.

Ich möchte Herrn Prof. Dr.-Ing. Udo Nackenhorst für die Möglichkeit der Promotion an seinem Institut danken. Als Doktorvater stand er mir in der gesamten Zeit wegweisend für Fragen und Abstimmungen zur Verfügung und hat somit durch seine wissenschaftliche Anleitung maßgebend zum Gelingen dieser Doktorarbeit beigetragen. Auch für das Vertrauen, Lehrveranstaltungen als Übungsleiter zu betreuen, möchte ich mich bei ihm bedanken, da diese Aufgaben mich über mein wissenschaftliches Arbeiten hinaus geprägt haben. Ich habe ihn als sehr fürsorglichen Institutsleiter kennengelernt, der sich immer Zeit für die Anliegen seiner Mitarbeiter nimmt. Herrn Prof. Dr.-Ing. habil. Dr. h.c. mult. Dr.-Ing. E.h. Peter Wriggers möchte ich für die Übernahme des Koreferats und das Interesse an meiner Dissertation Dank aussprechen. Ein Dank geht auch an die weiteren Mitglieder der Promotionskommission Prof. Dr.-Ing. Michael Beer und Prof. Dr.-Ing. Peter Schaumann.

Allen Mitarbeitern des Instituts für Baumechanik und Numerische Mechanik, mit denen ich während meiner Zeit am Institut zusammenarbeiten durfte, möchte ich an dieser Stelle meinen Dank zukommen lassen. Ich habe die Arbeit als wissenschaftlicher Mitarbeiter am Institut immer als sehr harmonisch und kollegial erfahren dürfen. Ich blicke auf zahlreiche konstruktive Gespräche mit Maximilian Bittens während meiner gesamten Promotion zurück. Bei Fragen zu Software oder Programmierung konnte ich immer auf ihn zählen. Gerade am Anfang meiner Promotion hatte ich inhaltlich wertvolle Gespräche mit Shadi Alameddin. Diese haben mich bei meinem Einstieg in den Themenbereich der Modellreduktion begleitet. Die wertvollen Diskussionen mit Stein Stoter in der zentralen Phase meiner Forschungsarbeit haben mir sehr bei der Strukturierung meiner Studien geholfen. Der wissenschaftliche Austausch mit Thirumalalagu Palanichamy, Ammar Basmaji und Jorge Urrea gab mir stets Anregungen während der Bearbeitung meiner Dissertation. Esther Voelsen, Rodolfo Fleury und Fynn Bensel möchte ich für die Unterstützung bei MATLAB[®] und ABAQUS[®] danken. Zur entspannten und hilfsbereiten Arbeitsatmosphäre haben letztendlich alle Mitarbeiterinnen und Mitarbeiter ihren Beitrag geleistet. Die Liste wäre an dieser Stelle zu lang, aber ich bin dankbar dafür, dass ich so viel liebe Kolleginnen und Kollegen in meiner Zeit am Institut kennenlernen durfte.

Ein besonderer Dank geht an meine liebe Freundin Katrin Gräfin von Schwerin. Ihre mentale Unterstützung in der gesamten Zeit war für mich eine Quelle der Kraft. So manches Mal habe ich nach Feierabend mit ihr bei einem Glas Wein über die weiteren Strategien bei der Umsetzung meiner Forschungsarbeit gesprochen. Auch mit dem Korrekturlesen des Manuskriptes hat sie mich tatkräftig unterstützt. Meinen lieben Eltern Magrit und Wilhelm Funk möchte ich für die unendliche und hingebungsvolle Unterstützung in all den Jahren danken. Ohne sie wäre mir mein eingeschlagener Weg nicht so möglich gewesen. Abschließend danken möchte ich außerdem Prof. Wilfried Gawehn, der mein Interesse an der FEM stets gefördert hat. Auch für die zahlreichen Diskussionen und das Korrekturlesen des Manuskripts sei ihm an dieser Stelle nochmals gedankt.

Zusammenfassung

In der vorliegenden Arbeit werden Untersuchungen zur Modellreduktion mechanischer Systeme mit elasto-plastischem Materialverhalten durchgeführt. Das untersuchte Verfahren zur Modellreduktion ist in dieser Arbeit die *Support Vector Regression* (SVR). Nach einem Überblick zum Stand der Technik im Rahmen der Modellreduktion mechanischer Systeme, wird die Theorie der Kontinuumsmechanik und der Finiten Elemente Methode (FEM) bereitgestellt. Im Anschluss an eine ausführliche Darlegung des mathematischen Hintergrunds der *Support Vector Regression* wird die Methode auf den eindimensionalen, rein phänomenologischen elasto-plastischen Fall angewendet. Ein wesentliches Kapitel widmet sich der Anwendung auf den drei- und zweidimensionalen elasto-plastischen Berechnungsfall unter vorangegangener Finite-Elemente Berechnung.

In den Studien wird gezeigt, dass die *Matèrn 5/2* Kernel Funktion sehr gut für Anwendungen im Bereich der Elasto-Plastizität geeignet und mit entsprechend gewählten Parametern anderen *Kernel* Funktionen überlegen ist. Mit verschiedenen, in dieser Dissertation vorgestellten adaptiven Sampling-Methoden kann eine teilweise Verbesserung im Gegensatz zu der homogenen, kartesischen Verfeinerung erzielt werden. Im Rahmen der FE-Anwendung zeigt sich die eigentliche Stärke der Modellreduktion mittels SVR. Die ungleichmäßigen und nicht stetigen Antwortflächen der gewählten Größe von Interesse können mit der Support Vector Regression hinreichend präzise angenähert werden. Die Sensitivität in den verschiedenen Parameter-Richtungen lässt eine zusätzliche Reduktion der nötigen Trainingsdaten mittels anisotropen Gitterstrukturen zu.

Schlüsselwörter: *Modellreduktion, Support Vector Regression (SVR), Nichtlineare Finite Elemente, Kernel Funktion, Hyperparameter, Elasto-Plastizität*

Abstract

In this thesis, investigations on model reduction of mechanical systems with elasto-plastic material behaviour are carried out. The investigated method for model reduction in this work is the *Support Vector Regression* (SVR). After a state of the art in model reduction of mechanical systems, the theory of continuum mechanics and the finite element method (FEM) is provided. Following a detailed presentation of the mathematical background of Support Vector Regression, the method is applied to the one-dimensional, purely phenomenological elasto-plastic case. A essential chapter is addressed towards the application of the method to the three- and two-dimensional elasto-plastic case, preceded by a finite element calculation.

In the numerical studies, it is shown that the *Matèrn 5/2* kernel function is very well suited for applications in the field of elasto-plasticity, and with appropriately chosen parameters is superior to other kernel functions. With different adaptive sampling methods presented in this dissertation, a partial improvement can be achieved compared with the homogeneous cartesian refinement. In the context of the FE application, the real strength of model reduction using SVR becomes apparent. The non-uniform and non-smooth response surfaces of the selected quantity of interest can be approximated with sufficient precision using Support Vector Regression. The sensitivity in the different parameter directions allows an additional reduction of the necessary training data by means of anisotropic grid structures.

Keywords: *Model reduction, Support Vector Regression (SVR), non-linear finite elements, kernel function, hyperparameter, elasto-plasticity*

Inhaltsverzeichnis

1	Einleitung	1
2	Reduzierte Modelle	3
2.1	Mechanische Anwendungen	4
2.2	Verschiedene Techniken der Modellreduktion	4
3	Kontinuumsmechanik	13
3.1	Kinematik	13
3.2	Spannungen	17
3.3	Bilanzgleichungen	19
3.4	Materialgleichungen	20
3.4.1	Elastisches Materialverhalten	20
3.4.2	Elasto-Plastisches Materialverhalten	21
3.4.3	Algorithmische Aspekte der Elasto-Plastizität	24
4	Finite Elemente Methode für hoch aufgelöste Berechnungen	29
4.1	Schwache Formulierung der zugrunde liegenden Gleichungen	29
4.2	Finite Elemente Ansatz	31
4.3	Isoparametrisches Konzept	33
4.4	Numerische Integration	36
4.5	Lösung der Finite Elemente Gleichungssysteme	36
4.5.1	Iteratives Lösungsverfahren: <i>Newton-Raphson</i> Methode	37
5	Support Vector Regression (SVR)	43
5.1	Support Vector Maschinen für Regression	43
5.2	Standardisieren der Eingabedaten	51
5.3	Nicht glatte Antwortspektren	51
5.4	Sampling Methoden	52
6	Anwendung in der 1D-Elasto-Plastizität	55
6.1	Modellaufbau	55
6.2	Studien zur Kernel Funktion und Parametrisierung	58
6.3	Überanpassung (<i>engl. Overfitting</i>)	68
6.4	Studien zur Sampling Methode	73
6.5	Fazit aus den Studien	85
7	Anwendung in der 2D- und 3D-Elasto-Plastizität	87
7.1	Vier-Punkt-Biegebalken	87
7.2	Parameterraum und Ausgabegröße (<i>Quantity of Interest</i>)	88
7.3	Modellaufbau in 3D	89
7.4	Test- und Trainingsdaten für das reduzierte Modell in 3D	91
7.5	Ergebnisse der <i>Support Vector Regression</i> in 3D	97
7.6	Modellaufbau in 2D (Ebener Spannungszustand)	100
7.7	Test- und Trainingsdaten für das reduzierte Modell in 2D	102
7.8	Ergebnisse der <i>Support Vector Regression</i> in 2D	105
7.9	Sensitivität in den Parameterrichtungen	107
7.10	Fazit aus den Untersuchungen und Rechenzeiterparnis	110

8 Zusammenfassung und Ausblick	113
A Anhang	117
A.1 SVR Programmcode in MATLAB [®]	117
A.2 <i>Matèrn</i> 5/2 Kernel Funktion in MATLAB [®]	117
A.3 FE-Eingabedatensätze für die Parameterstudie in ABAQUS [®]	118
A.4 Programmcode für die Parameterstudie in ABAQUS [®]	120
Literaturverzeichnis	125

Verwendete Notation

Skalare, Vektoren und Tensoren

Im Rahmen dieser Arbeit wird mit *kartesischen* Tensoren gearbeitet, sodass keine Unterscheidung zwischen den Basisvektoren und ihren reziproken Basisvektoren nötig ist. Das Konzept des Dualraumes wird nicht benötigt, sodass auch nicht mit dualen Basisvektoren gearbeitet werden muss. Die Unterscheidung zwischen *reziproken* und *dualen* Basisvektoren wird hier in Anlehnung an Mühlich (2017) gemacht, in der *klassischen* Tensorrechnung werden diese Begriffe synonym verwendet. Indizierte Größen im Sinne der Tensorrechnung können im Rahmen dieser Arbeit somit durch ausschließlich tiefgestellte Indices dargestellt werden. Es wird die *Einstein'sche* Summationskonvention verwendet, nach der eine Summation bei doppelt auftretenden Indices in einem Monom über diese Indices ausgeführt wird (Kreißig und Benedix, 2013). Die *symbolische* Schreibweise für *Vektoren*, also Tensoren 1-ter Stufe und allgemein *Tensoren*, also Tensoren mit einer Stufe ≥ 2 , ist in dieser Arbeit durch kursiven Fettdruck, z.B. $\mathbf{A} = A_{ij} \mathbf{e}_i \otimes \mathbf{e}_j$ oder kalligrafischen Fettdruck, z.B. $\mathcal{K} = \mathcal{K}_{ijk} \mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k$ dargestellt. Die für Tensoren verwendeten griechischen Buchstaben sind lediglich in Fettdruck dargestellt, z.B. $\boldsymbol{\sigma} = \sigma_{ij} \mathbf{e}_i \otimes \mathbf{e}_j$. Für *Skalare*, also Tensoren 0-ter Stufe, werden kursive oder aufrecht stehende, aber nicht fett gedruckte Symbole verwendet, z.B. x oder x . Einige der verwendeten Operation für Tensoren sind in der unten stehenden Tabelle aufgeführt.

Matrizen und Voigt-Notation

Matrizen werden in dieser Arbeit durch Symbole in aufrecht stehendem Fettdruck gekennzeichnet, z.B. $\mathbf{S} := [S_{ij}]$. Im Rahmen der *Voigt*-Notation werden Tensoren in spezielle Matrizen-schemata überführt. Hier wird bei griechischen Buchstaben, z.B. $\boldsymbol{\epsilon}$, aus dem Zusammenhang klar, ob eine symbolische Tensorschreibweise oder Matrizen-schreibweise gemeint ist. Einige der verwendeten Operationen für Matrizen sind in der unten stehenden Tabelle aufgeführt.

Symbol	Beschreibung
\cdot	Skalarprodukt: Auch bekannt als einfache Verjüngung oder Kontraktion zwischen Tensoren
$\cdot\cdot$	doppeltes Skalarprodukt: Auch bekannt als doppelte Verjüngung oder zweifache Kontraktion zwischen Tensoren
\otimes	dyadisches Produkt: Auch bekannt als tensorielles Produkt zwischen Tensoren
$\mathbf{e}_{i1} \otimes \mathbf{e}_{i2} \cdots \otimes \mathbf{e}_{in}$	Tensorbasis: Tensorielle Verknüpfung von n Basisvektoren zu einer Basis für einen Tensor n -ter Stufe
$(\cdot)^T$	Transponierte Größe: Transponierter Tensor oder das Tauschen von Spalten und Zeilen einer quadratischen Matrix
$(\cdot)^{-1}$	Inverse Größe: Inverser Tensor oder die Inversion einer quadratischen Matrix
$(\cdot)^{-T}$	Invers-transponierte Größe: Invers-transponierter Tensor oder das Invertieren und anschließendes Transponieren einer quadratischen Matrix
$\delta_{ij} := \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$	Kronecker-Symbol (auch Kroneker-Delta genannt)

1 Einleitung

In zahlreichen Forschungsdisziplinen, wie in Universitäten und externen Einrichtungen, aber auch in Unternehmen, die danach streben, Spitzenprodukte auf den Markt zu bringen, ist eine solide Rechenleistung für computergestützte Modellauswertungen eine Grundvoraussetzung. Die Zusammenschlüsse mehrerer Rechner, aber auch Großrechneranlagen können gezielt zur Berechnung hoch aufgelöster numerischer Simulationen genutzt werden. Berechnungszeiten von Tagen bis hin zu Wochen sind bei komplexen Modellen keine Seltenheit. Gekoppelte physikalische Systeme, die thermische, mechanische und elektrische Feldgleichungen miteinander verbinden, aber auch Mehrskalensysteme in der Festkörper- oder Fluidmechanik erzeugen großen Berechnungsaufwand. Ein weiteres typisches Beispiel mit immensem Rechnerleistungsbedarf ist der Rollkontakt von Reifen (Nackenhorst, 2004). Größere Rechnerleistungen werden in immer kürzeren Abständen von den Hardwareherstellern angeboten. Auch wenn daher die Berechnungszeiten in der Zukunft weiter reduziert werden, steht diesem Trend der Wunsch nach immer detaillierteren und höher aufgelösten physikalischen Beschreibungen entgegen. Auf der anderen Seite ist es vielfach auch geradezu mühsam, eine aufwändige Berechnung erneut durchzuführen, wenn zum Beispiel lediglich „ein“ Parameter geändert wurde. Gerade in Entwicklungsphasen sind oft Trends und grobe Richtungen bei Veränderungen oder Anpassungen am Produkt interessant. Hierbei kommt es nicht auf eine „exakte“ Lösung an, wobei „exakt“ hier relativ zu sehen ist. Ein Forschungsbereich, der an dieser Stelle ansetzt, ist der Bereich der sogenannten *Modellreduktion*. Die Grundidee besteht darin, die Daten eines hoch aufgelösten Modells, sowohl Eingabe- als auch Ergebnisdaten, nach Abschluss einer Berechnung zu konservieren und weiter in einem „günstigeren“ Ersatzmodell zu verarbeiten. Sind beispielsweise 5 hoch aufgelöste Modelle mit leicht unterschiedlichen Parametern berechnet worden, so kann eventuell eine ausreichende *Approximation* für eine sechste Parameterkombination auf der Grundlage der 5 anderen hoch aufgelösten Modelle durchgeführt werden.

Im Rahmen dieser Arbeit wird ein Ersatzmodell bzw. ein reduziertes Modell für die Behandlung von Bauteilen mit elasto-plastischem Materialverhalten vorgestellt. Das Antwortverhalten von mechanischen Modellen mit elasto-plastischem Materialverhalten ist zum einen nichtlinear (ab einer gewissen Fließgrenze). Andererseits ist es von mehr Materialparametern abhängig als ein Modell mit rein linear elastischem Materialverhalten.

In Kapitel 2 wird ein Einblick in die Anwendungsgebiete reduzierter Modelle in der Mechanik gegeben. Ein weiteres Unterkapitel widmet sich den verschiedenen Methoden der Modellreduktion, wobei auch hier der Fokus auf mechanischen Anwendungen liegt.

Das Kapitel 3 führt den nötigen theoretischen Rahmen für die später folgenden mechanischen Anwendungen in der Elasto-Plastizität auf. Ausgehend von der (Verzerrungs-)kinematik werden nach der Einführung des Spannungskonzepts die Bilanzgleichungen der Festkörpermechanik aufgestellt. Ein Hauptteil dieses Kapitels stellen die Materialgleichungen dar, wobei das in den Modellen verwendete elasto-plastische Materialverhalten mit Verfestigung vorgestellt wird.

Für die zu „Trainingszwecken“ hoch aufgelösten Berechnungen wird im Rahmen dieser Arbeit die *Finite Element Methode* (FEM) verwendet. Das Kapitel 4 widmet sich daher diesem heutzutage wohl meist verwendeten numerischen Näherungsverfahren in der Festkörper- und Strukturmechanik. Nach der Herleitung der schwachen Form der Bilanzgleichungen wird der Grundgedanke der FEM vorgestellt. Anschließend folgt ein Unterkapitel über die Lösungsverfahren der FEM-Gleichungssysteme mit besonderer Rücksicht auf nichtlineares bzw. elasto-plastisches Materialverhalten.

Für die Modellreduktion wird im Rahmen dieser Arbeit ein Ersatzmodell mit der sogenannten *Support Vector Regression* (SVR) konstruiert. Dieses Verfahren, zusammen mit der Anwendung auf Elasto-Plastizität, stellt den Kern der vorliegenden Arbeit dar und ist zugleich

nach bestem Wissen des Autors in dieser Form ein neuer Anwendungsbereich. Das Kapitel 5 stellt die mathematisch-numerischen Grundlagen der *Support Vector Regression* vor. Zuerst werden klassische, lineare Verfahren der Regression vorgestellt. Nach der Einführung der für nichtlineare Phänomene wichtigen *Kernel* Funktionen werden die linearen Verfahren auf diesen Anwendungsbereich erweitert. Die Formulierung der *Support Vector Regression* erfolgt über die Einführung einer speziellen Verlustfunktion, für dessen Lösung ein Minimierungsproblem unter Nebenbedingungen und Komplementaritätsbedingungen ausgewertet werden muss. Von entscheidender Bedeutung für die Ergebnisgüte ist die Wahl der *Kernel* Funktion sowie die Wahl der Parameter, die das Verfahren steuern. Vielfach ist es sinnvoll, die Eingabedaten zu standardisieren, weshalb ein separates Unterkapitel hierfür vorgesehen ist. Im Rahmen dieser Arbeit spielen nicht glatte Antwortspektren eine erhebliche Rolle, da dieses Phänomen gerade bei elasto-plastischen Materialverhalten zum Tragen kommt. Aus diesem Grund widmet sich ein eigenes Unterkapitel diesem Phänomen. Für die Güte eines Ersatzmodells ist die Positionierung der *Trainingspunkte* entscheidend. In dem entsprechenden Unterkapitel werden unterschiedliche Methoden vorgestellt, mit denen die Trainingspunkte verteilt werden können.

In Kapitel 6 werden die Studien des Autors zum „reinen“, eindimensionalen elasto-plastischen Materialverhalten vorgestellt, sprich ohne die Analyse mittels mechanischem FE-Modell. Die wesentliche Materialroutine, das sogenannte *Return Mapping* und somit das Aufteilen der Gesamtdehnung in rein elastische Dehnung und rein plastische Dehnung mit entsprechender Spannungsantwort wird hier rein phänomenologisch durchgeführt. Die Auswirkungen der *Kernel* Funktionen und der wesentlichen Parameter für die *Support Vector Regression* auf das nichtlineare und nicht glatte Antwortspektrum lassen sich hierbei sehr gut studieren und später auf den dreidimensionalen FEM-Fall übertragen. Im Anschluss an die Studien zur *Kernel* Funktion und der Kalibrierung der Parameter folgt eine Untersuchung zum Phänomen der Überanpassung (*engl. Overfitting*). Das Kapitel schließt mit einer umfangreichen Studie zu *adaptiven* Sampling Methoden ab. Dabei werden zwei Verfahren vorgestellt und mit einer homogen Verfeinerung der Trainingsdaten verglichen.

Die gewonnenen Erkenntnisse aus den Studien am eindimensionalen, rein phänomenologischen Fall der Elasto-Plastizität werden in Kapitel 7 auf den dreidimensionalen (3D) und zweidimensionalen (2D) Fall angewandt. Zuerst wird das mechanische Modell vorgestellt und die *Quantity of Interest* definiert, für die das reduzierte Modell Näherungslösungen liefern soll. Im ersten Schritt werden 3D FE-Modelle mit unterschiedlichen Vernetzungsdichten untersucht. Eine weitere Erhöhung der Netzdichte wird für ein 2D FE-Modell durchgeführt, wobei im Rahmen dieser Studie in einem zweiten Schritt die Verteilung der Testdaten geändert wird. In einem Unterkapitel wird die Sensitivität des reduzierten Modells in den Parameter-Richtungen untersucht.

Das abschließende Kapitel 8 fasst die Arbeit und insbesondere die gewonnenen Erkenntnisse zusammen und weist weiter auf Ausbaumöglichkeiten der *Support Vector Regression* im Rahmen von nichtlinearem Materialverhalten hin.

2 Reduzierte Modelle

Für Parameterstudien oder Berechnungen mechanischer Strukturen, die mehrfach mit nur leichten Modifikationen durchgeführt werden, ist eine *Modellreduktion* vorteilhaft. Dabei wird im Folgenden unter einer Modellreduktion ein Ablauf verstanden, wie er in der Abbildung 2.1 dargestellt ist. Im ersten Schritt werden n hoch aufgelöste Berechnungen \mathcal{F}^{HF} (*HF* engl. für *High Fidelity*) durchgeführt. Für einen Eingabedatensatz $\mathbf{x}_i = [x_{i1} \cdots x_{im}]$ verschiedener Parameter x_{ij} wird beispielsweise im Rahmen der Mechanik ein Finite Elemente Modell (FE-Modell) berechnet. Das ermittelte Ergebnis dieser Berechnung, die sogenannte *Quantity of Interest* \mathbf{y}_i (diese kann aus einer oder mehreren Größen bestehen), wird zusammen mit dem Eingabedatensatz \mathbf{x}_i abgelegt. Die Eingabe- und Ergebnisdatensätze der n Berechnungen, zusammengefasst in \mathbf{X} bzw. \mathbf{Y} , werden im nächsten Schritt zum *Trainieren* des reduzierten Modells \mathcal{F}^{RM} (*RM* engl. für *Reduced Model*) herangezogen. Im letzten Schritt, sozusagen die Anwendung, wird

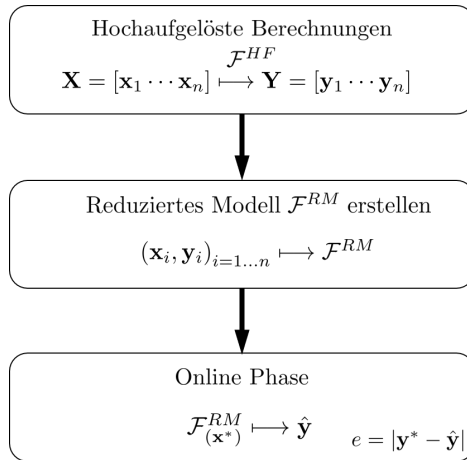


Abbildung 2.1: Flussdiagramm einer Modellreduktion

mit dem Modell \mathcal{F}^{RM} , für einen neuen Eingabedatensatz \mathbf{x}^* , ein *Näherungsergebnis* $\hat{\mathbf{y}}$ für die *Quantity of Interest* berechnet. Dies geschieht mit erheblich weniger Berechnungsaufwand als bei der hoch aufgelösten Berechnung. Ziel ist es, neben der Zeitersparnis die Abweichung $e = |\mathbf{y}^* - \hat{\mathbf{y}}|$ zwischen der hoch aufgelösten Berechnung \mathbf{y}^* und der approximierten Lösung $\hat{\mathbf{y}}$ möglichst gering zu halten.

Die ersten beiden Blöcke in Abbildung 2.1 werden zusammen vielfach *Trainingsphase* genannt und der letzte Block *Online Phase*. Die Trainingsphase stellt den Anwender vor zwei wesentliche Herausforderungen:

1. Welches Modellreduktionsverfahren ist geeignet, bei ausreichendem Training hinreichende Approximationen zu liefern?
2. Wie viel hoch aufgelöste Berechnungen werden für das Training benötigt, um ein valides reduziertes Modell zu erzeugen?

Die genannten Fragestellungen sind in erster Linie von dem zu reduzierenden Ausgangssystem abhängig, wobei dieser Sachverhalt genauso wie das gesamte Gebiet der Modellreduktion heutzutage intensiver Gegenstand der Forschung ist. Aus diesem Grund widmet sich Kapitel 2.1 der Vielfalt mechanischer Anwendungen, um das Spektrum der Modellreduktion, gerade in Bezug auf die Mechanik, zu veranschaulichen. In dem anschließenden Kapitel 2.2 wird ein Überblick über die verschiedenen Verfahren der Modellreduktion gegeben.

2.1 Mechanische Anwendungen

Die Modellreduktion hat in all ihrer Vielfalt Einzug in die unterschiedlichsten wissenschaftlichen Disziplinen gehalten. Im Rahmen dieser Arbeit liegt der Fokus auf der Mechanik, sodass im Folgenden einige interessante Veröffentlichungen diesbezüglich genannt werden. Die Auflistung erfolgt für „reduzierte Modelle“ in der Mechanik allgemein, sodass an dieser Stelle noch keine Unterscheidung zwischen *intrusiven* und *nicht intrusiven* Methoden der Modellreduktion durchgeführt wird. Die Auflistung ist bei weitem nicht allumfassend und soll nur eine Idee von dem aufzeigen, was mittels Modellreduktion möglich ist.

Eine umfangreiche Übersicht zum derzeitigen Stand der Technik mit einer Einführung in die grundlegenden Techniken der Modellreduktion ist in Chinesta et al. (2017) gegeben. In Carlberg und Farhat (2011) erfolgt die Anwendung auf eine parametrisierte Problemstellung aus dem Luft- und Raumfahrtbereich, speziell auf ein Standard Forschungs-Tragflügelmodell. Für den Tragflügel soll in dieser Arbeit mithilfe der Modellreduktion eine optimale Struktur gefunden werden. In der Arbeit von Serani et al. (2020) kommt eine nichtlineare Dimensionsreduktion im Bereich der Hydrodynamik von Schiffen zum Einsatz. Die Veröffentlichung von Wang et al. (2008) zielt auf die Parameteroptimierung hochgradig nichtlinearer Systeme ab. Zum einen wird hier eine Studie zur Energieaufnahme bei Crashtest Simulationen durchgeführt. Zum anderen wird eine Topologieoptimierung im Bereich der Metallumformung vorgestellt. In Moustapha und Sudret (2019) wird die Modellreduktion unter anderem auf ein Zwei-Stab-Tragsystem angewendet, dass bei geometrisch nichtlinearer Betrachtung ein Durchschlagsproblem darstellt.

2.2 Verschiedene Techniken der Modellreduktion

In der vorliegenden Arbeit wird die *Support Vector Regression* (SVR) für die Modellreduktion herangezogen. SVR stellt ein *nicht-intrusives* Verfahren der Modellreduktion dar. Im Gegensatz hierzu stehen *intrusive* Verfahren wie beispielsweise die *Proper Generalized Decomposition* (PGD), die unter anderem in den Büchern von Chinesta et al. (2013) und Cueto et al. (2016) dargestellt wird. Erweiterungen für die *Proper Generalized Decomposition* werden beispielsweise in den Veröffentlichungen von Alameddin et al. (2017) und Bhattacharyya et al. (2018) vorgestellt.

Im Folgenden soll das grundsätzliche Vorgehen von Modellreduktionsverfahren vorgestellt werden. Dies soll anhand eines Modells aus dem Bereich der Strukturmechanik geschehen, welche auch ausführlich in den späteren Kapiteln 3 und 4 zur Kontinuumsmechanik bzw. FEM beschrieben werden. Die Größe von Interesse bzw. die *Quantity of Interest* sei das kontinuierliche Verschiebungsfeld $\mathbf{u} = [u_x \ u_y \ u_z]^T$, für das im Rahmen der FEM eine Näherungslösung der Form

$$\mathbf{u} \approx \sum_{i=1}^n h^i \hat{\mathbf{u}}^i, \quad (2.1)$$

gefunden werden soll, wobei die Summe über die Anzahl n der Knoten des FE-Modells zu bilden ist. Mit h^i sind hierbei die *Ansatzfunktionen* gekennzeichnet und $\hat{\mathbf{u}}^i$ steht für die zu bestimmenden *Knotenverschiebungen*. Die Freiheitsgrade (*engl. Degrees of Freedom*, DOF) resultieren hier aus der FE-Diskretisierung und betragen für den angesetzten dreidimensionalen

Fall $\text{DOF}^{HF} = 3n$. Bei der Methode der reduzierten Basen (engl. *Reduced Bases Methods*) wird jetzt davon ausgegangen, dass die angenäherte Lösung für das Modell mit einer deutlich geringeren Anzahl an Freiheitsgraden DOF^{RM} konstruiert werden kann: $\text{DOF}^{RM} \ll \text{DOF}^{HF} = 3n$. Nach Quarteroni et al. (2015), Patera und Rozza (2007) und Quarteroni et al. (2011) kann aus mathematischer Sicht festgehalten werden, dass die Lösung des Problems auf einer *niedrigdimensionalen Mannigfaltigkeit* neben der wahren, hochdimensionalen Mannigfaltigkeit entsprechend des FE-Ansatzes „lebt“. Dieser Sachverhalt kann ingenieurmäßig auf eine anschauliche Weise interpretiert werden. Liegt beispielsweise ein linear elastisches, strukturelmechanisches Problem vor, bei dem die Parameter „Elastizitätsmodul“ E und „Querkontraktionszahl“ ν variiert werden sollen, so ist es ausreichend, in der *Trainingsphase* „wenige“ hoch aufgelöste Berechnungen für verschiedene Parameter E und ν mittels der FEM durchzuführen und in der *Online Phase* dann hieraus die *Approximationen* zu generieren. Das Verschiebungsfeld wird sich bei Vorgabe neuer Parameter E^* und ν^* nicht „grundlegend“ ändern, vielmehr nimmt es eine Lösung auf der niedrigdimensionalen „Lösungs“-Mannigfaltigkeit an. Ein ähnliches Vorgehen ist aus der *Modalanalyse* der linearen Strukturodynamik bekannt, bei der aus wenigen, niedrigfrequenten Schwingungsmoden eine Lösungsantwort durch Superposition ermittelt wird. Nach Swischuk et al. (2019) kann ein Näherungsansatz für das Verschiebungsfeld der Form

$$\tilde{\mathbf{u}}(t; \mathbf{p}) \approx \sum_{i=1}^d \Psi_i \mathbf{b}_i(t; \mathbf{p}), \quad (2.2)$$

gemacht werden, wobei mit Ψ_i die *globalen*¹ Ansatzfunktionen gekennzeichnet sind. Die vektorwertigen Gewichtungsfaktoren \mathbf{b}_i hängen in diesem Ansatz von der Zeit t und einer Kollektion an Parametern \mathbf{p} ab, in dem gewählten Beispiel gilt z.B. $\mathbf{p} = [E^* \nu^*]$. Die Summe in Gleichung (2.2) ist hierbei über eine deutlich geringe Anzahl $d \ll n$ an Freiheitsgraden zu bilden als in dem FE-Ansatz (2.1). In Swischuk et al. (2019) wird gezeigt, wie der Ansatz (2.2), genauer gesagt die Berechnung der globalen Basen Ψ_i , weiter quantifiziert werden kann. Hierfür werden zunächst in der Trainingsphase k hoch aufgelöste FE-Berechnungen für verschiedene Parameterkombinationen $[E^j \nu^j]_{j=1 \dots k}$ durchgeführt. Die Ergebnisse $\hat{\mathbf{u}}_j = [\hat{u}_x^1 \hat{u}_y^1 \hat{u}_z^1 \dots \hat{u}_x^n \hat{u}_y^n \hat{u}_z^n]^T$ mit $j = 1 \dots k$ werden in der Matrix $\mathbf{Q} \in \mathbb{R}^{3n \times k}$ zusammengefasst, wobei in der Regel bei FE-Modellen $3n > k$ gilt. Die wesentlichen Informationen über das Modellverhalten sind in der sogenannten Schnapsschuss-Matrix $\mathbf{Q} = [\hat{\mathbf{u}}_1 \dots \hat{\mathbf{u}}_k]$ (engl. *Snapshot Matrix*) bereits enthalten. In dieser Form ist es aber noch nicht möglich, mit \mathbf{Q} reduzierte Basen zu erstellen, da in den Spalten der Matrix redundante Informationen vorhanden sind. In der linearen Algebra zeichnet sich eine Basis dadurch aus, dass die Basisvektoren linear unabhängig sind. Lineare Unabhängigkeit liegt beispielsweise vor, wenn jeder Basisvektor orthogonal zu den restlichen Basisvektoren in der Spanne der Basisvektoren ist. Um eine solche Basis zu erzeugen, wird \mathbf{Q} einer Singulärwertzerlegung (engl. *Singular Value Decomposition*, SVD) unterzogen, die dadurch gekennzeichnet ist, dass sie anders als eine „klassische“ Eigenwertberechnung direkt auf nicht quadratische Matrizen angewendet werden kann. Die Singulärwertzerlegung kann wie folgt angegeben werden,

$$\mathbf{Q} = \mathbf{V} \mathbf{\Sigma} \mathbf{W}^T, \quad (2.3)$$

wobei $\mathbf{V} \in \mathbb{R}^{3n \times 3n}$ die Matrix der linken Singulärvektoren kennzeichnet und $\mathbf{W} \in \mathbb{R}^{k \times k}$ die Matrix der rechten Singulärvektoren. Die Matrix $\mathbf{\Sigma} \in \mathbb{R}^{3n \times k}$ ist eine zusammengesetzte Matrix der Form

$$\mathbf{\Sigma} = \begin{bmatrix} \mathbf{\Lambda} \\ \mathbf{0} \end{bmatrix}, \quad (2.4)$$

¹Die FE-Ansatzfunktionen h^i nehmen an dem jeweiligen Knoten i den Wert 1 an und an allen anderen Knoten im Gebiet den Wert 0. Die Ansatzfunktionen Ψ_j seien zunächst einmal unabhängig von der FE-Diskretisierung global und kontinuierlich über dem Gebiet aufgespannt.

mit der Diagonalmatrix $\mathbf{A} = \text{diag}(\sigma_1 \cdots \sigma_k) \in \mathbb{R}^{k \times k}$ der Singulärwerte σ_i der Matrix \mathbf{Q} und einer „Nullmatrix“ $\mathbf{0} \in \mathbb{R}^{3n-k \times k}$ (Quarteroni et al., 2015). Die Singulärwerte σ_i von $\mathbf{Q} \in \mathbb{R}^{3n \times k}$ stehen mit den Eigenwerten λ_i der Matrix $\mathbf{G} = \mathbf{Q}^T \mathbf{Q}$ in dem Zusammenhang $\sigma_i = \sqrt{\lambda_i}$ (Quarteroni et al., 2011). Die Matrix $\mathbf{G} \in \mathbb{R}^{k \times k}$ wird als *Gramsche* Matrix bezeichnet und beinhaltet in diesem Fall die Skalarprodukte $G_{ij} = \mathbf{u}_i^T \mathbf{u}_j$ der *Snapshots*. Eine geeignete Basis für den allgemeinen Ansatz (2.2) kann mit den ersten d Spalten $\{\boldsymbol{\vartheta}_1 \cdots \boldsymbol{\vartheta}_d\} \subset \{\mathbf{v}_1 \cdots \mathbf{v}_d \cdots \mathbf{v}_{3n}\}$ aus der Matrix der linken Singulärvektoren \mathbf{V} generiert werden, wobei $d \leq k$ gilt. Der allgemeine Ansatz (2.2) kann hiermit in die folgende Form umgeschrieben werden,

$$\hat{\mathbf{u}}(t; \mathbf{p}) \approx \sum_{i=1}^d \boldsymbol{\vartheta}_i \beta_i(t; \mathbf{p}) = \boldsymbol{\Theta} \boldsymbol{\beta}, \quad (2.5)$$

mit der Matrix $\boldsymbol{\Theta} = [\boldsymbol{\vartheta}_1 \cdots \boldsymbol{\vartheta}_d]$ der herangezogenen d Basisvektoren und der Matrix der reduzierten, skalaren Freiheitsgrade $\boldsymbol{\beta} = [\beta_1 \cdots \beta_d]^T$. Aus dem Ansatz (2.5) wird sofort die auch vielfach verwendete und zutreffende (englische) Bezeichnung *Proper Orthogonal Decomposition* (POD) deutlich, handelt es sich doch um eine „geeignete“, orthogonale Zerlegung von $\hat{\mathbf{u}}$ in die Basisvektoren $\boldsymbol{\vartheta}_i$. Während $\hat{\mathbf{u}} = [\tilde{u}_x \tilde{u}_y \tilde{u}_z]^T$ in Gleichung (2.2) eine „direkte“ Näherung für \mathbf{u} darstellt, ist $\hat{\mathbf{u}} \in \mathbb{R}^{3n \times 1}$ in Gleichung (2.5) wieder eine Formulierung in den Knotenverschiebungen im Sinne der FE-Diskretisierung. Die Lösungsfindung erfolgt in der *Online Phase* dann in den Freiheitsgraden β_i unter Eingabe der neuen, unbekannt Parameter t^* und $\mathbf{p} = [E^* \nu^*]$. Das Finite Elemente Gleichungssystem² der hoch aufgelösten Berechnung $\mathbf{K} \hat{\mathbf{u}} = \mathbf{f}_{ext}$ kann jetzt wie folgt mit $\hat{\mathbf{u}} \approx \boldsymbol{\Theta} \boldsymbol{\beta}$ in das *reduzierte* Gleichungssystem überführt werden,

$$\underbrace{\boldsymbol{\Theta}^T \mathbf{K} \boldsymbol{\Theta}}_{\tilde{\mathbf{K}} \in \mathbb{R}^{d \times d}} \boldsymbol{\beta} = \underbrace{\boldsymbol{\Theta}^T \mathbf{f}_{ext}}_{\tilde{\mathbf{f}}_{ext} \in \mathbb{R}^{d \times 1}}, \quad (2.6)$$

wobei mit $\tilde{\mathbf{K}}$ die reduzierte Steifigkeitsmatrix und mit $\tilde{\mathbf{f}}_{ext}$ der reduzierte äußere Kraftvektor gekennzeichnet sind. Die unterschiedliche Lösungsfindung für das Verschiebungsfeld in den Variablen $\hat{\mathbf{u}}$ und $\boldsymbol{\beta}$ ist in Abbildung 2.2 dargestellt. Die Abbildung 2.2 verdeutlicht, dass die Lösungsfindung in den reduzierten Variablen $\boldsymbol{\beta}$ das Spektrum der möglichen Verschiebungsfelder $\hat{\mathbf{u}}$ deutlich eingrenzt und die „Suche“ auf die niedrigdimensionale *Lösungs-Mannigfaltigkeit* beschränkt.

Das vorgestellte Vorgehen für *lineare* mechanische Systeme ist aus dem Grunde sehr effizient, weil die Parameter zum einen lediglich als Konstanten in die Elementbeiträge für die reduzierte Steifigkeitsmatrix $\tilde{\mathbf{K}}$ und den reduzierten äußeren Kraftvektor $\tilde{\mathbf{f}}_{ext}$ eingehen. Zum Anderen sind $\tilde{\mathbf{K}}$ und $\tilde{\mathbf{f}}_{ext}$ bei linearen Modellen selbst unveränderliche Größen, sodass die Berechnung (2.6) nur *einmalig* bei Abschluss der Trainingsphase durchgeführt werden muss. In der *Online Phase* werden die neuen Parameter von Interesse einfach in das reduzierte Modell eingesetzt. Bei *nichtlinearen* mechanischen Systemen kann die vorgestellte Vorgehensweise nicht direkt oder auch gar nicht angewendet werden. Dieser Sachverhalt ist der Tatsache geschuldet, dass ein nichtlineares Finite Elemente Gleichungssystem³ der Form $\mathbf{f}_{int}(\hat{\mathbf{u}}) - \mathbf{f}_{ext} = \mathbf{0}$ gelöst werden muss, bei dem eine Umformung wie in Gleichung (2.6) nicht mehr „allein“ zielführend ist und somit zu einer Reduktion der Freiheitsgrade beiträgt. Der sogenannte „Berechnungsgenpass“ (*engl. Computational Bottleneck*) liegt hier in der Auswertung des internen Kraftvektors ${}^n \mathbf{f}_{int}(\hat{\mathbf{u}})$ und dessen Ableitungen $\left. \frac{\partial \mathbf{f}_{int}}{\partial \hat{\mathbf{u}}} \right|_n$ für die tangentielle Steifigkeitsmatrix ${}^n \mathbf{K}$ zum Lastschritt n (Ghavamian et al., 2017). Denn, auch wenn eine analoge Matrizenmultiplikation wie in Gleichung (2.6) für das *linearisierte* FE-Gleichungssystem möglich ist, so hängt die

²Für die genau Definition der Größen des FE-Gleichungssystems wird an dieser Stelle auf das Kapitel 4, insbesondere auf die Gleichung (4.35) verwiesen

³Die iterative Lösung eines nichtlinearen FE-Gleichungssystems wird ausführlich in Kapitel 4.5.1 beschrieben

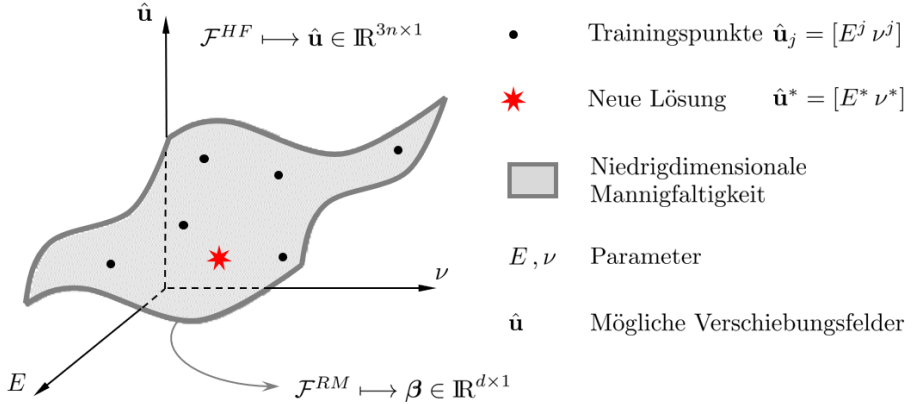


Abbildung 2.2: Lösungsfindung im Rahmen der reduzierten Basen Methode auf einer niedrigdimensionalen Mannigfaltigkeit in den Variablen $\beta \in \mathbb{R}^{d \times 1}$ neben der wahren, hochdimensionalen Mannigfaltigkeit in den Variablen $\hat{\mathbf{u}} \in \mathbb{R}^{3n \times 1}$ mit $d \ll 3n$

Berechnung der einzelnen Einträge⁴ in der tangentialen Steifigkeitsmatrix und dem internen Kraftvektor immer noch von der Anzahl $3n$ der Freiheitsgrade des vollen Systems ab (Ghavamian et al., 2017). Dieser Berechnungsengpass kann für das bereits vorgestellte Verfahren mittels einer sogenannten Hyperreduktion (engl. *Hyper-reduction*) verringert werden. In Chaturantabud und Sorensen (2010) und Ghavamian et al. (2017) wird hierfür beispielsweise die Diskrete Empirische Interpolationsmethode (engl. *Discrete Empirical Interpolation Method*), kurz „DEIM“ vorgestellt, die als *intrusives* Modellreduktionsverfahren eine Kombination aus dem bereits vorgestellten POD-Verfahren und einer Interpolation darstellt. Das Ziel mittels DEIM ist es, die Lösungssuche weiterhin in den d reduzierten Variablen $\beta = [\beta_1 \cdots \beta_d]^T$ durchzuführen und den Ansatz $\hat{\mathbf{u}} \approx \Theta \beta$ mit den Basisvektoren $\Theta = [\vartheta_1 \cdots \vartheta_d]$ wie gewohnt zu verwenden. Die jetzt *nichtlinearen Snapshots* werden vorab mittels Berechnung des vollen FE-Systems $\mathbf{f}_{int}(\hat{\mathbf{u}}(t; \mathbf{p})) - \mathbf{f}_{ext} = \mathbf{0}$ durchgeführt, wobei t und \mathbf{p} wieder Parameter darstellen. Die nichtlinearen Snapshots $\hat{\mathbf{u}}_j$ für die Parameter $[t^j \mathbf{p}^j]_{j=1 \dots k}$ werden wie gewohnt in der Matrix $\mathbf{Q} \in \mathbb{R}^{3n \times k}$ zusammengefasst. Die im Rahmen der DEIM-Methode notwendigen „Kraftvektor“-Snapshots werden in der Matrix $\mathbf{Q}_f = [\mathbf{f}_{int}(\hat{\mathbf{u}}_1) \cdots \mathbf{f}_{int}(\hat{\mathbf{u}}_k)] \in \mathbb{R}^{3n \times k}$ zusammengefasst. Die Singulärwertzerlegung wird im Rahmen der der Trainingsphase jeweils auf die beiden Matrizen \mathbf{Q} und \mathbf{Q}_f angewandt. Im Anschluss an die Singulärwertzerlegung können dann jeweils die d Basisvektoren $\Theta = [\vartheta_1 \cdots \vartheta_d] \in \mathbb{R}^{3n \times d}$ und $\Theta_f = [\Upsilon_1 \cdots \Upsilon_d] \in \mathbb{R}^{3n \times d}$ extrahiert werden. Auf der Grundlage von Θ_f wird jetzt ein Indices-Vektor \mathbf{i} und eine Indices-Matrix \mathbf{P} erstellt,

$$\begin{aligned}
 \text{DEIM}(\Theta_f) &\mapsto \mathbf{i} \in \mathbb{R}^{d \times 1}, \mathbf{P} \in \mathbb{R}^{3n \times d} \\
 \mathbf{P} &= [\mathbf{e}_{I_1} \cdots \mathbf{e}_{I_d}] \\
 \mathbf{i} &= [I_1 \cdots I_d], \tag{2.7}
 \end{aligned}$$

wobei die I_j in \mathbf{i} Indices aus der gesamten Spanne $3n$ der DOF's des vollen FE-Modells annehmen können. Die \mathbf{e}_{I_j} in \mathbf{P} kennzeichnen jeweils die j -te Spalte der Identitätsmatrix $\mathbf{I} \in \mathbb{R}^{3n \times 3n}$

⁴Für *Materialnichtlinearitäten* ist beispielsweise eine Materialroutine auf Integrationspunktebene für $\frac{\partial \sigma}{\partial \mathbf{e}}$ durchzuführen, während bei *linearen* mechanischen Systemen durchgehend die konstante 6×6 Elastizitätsmatrix verwendet werden kann.

und DEIM(\cdot) beschreibt die Anwendung des DEIM-Algorithmus aus Ghavamian et al. (2017). Der DEIM-Algorithmus gibt sozusagen über eine Auswahl an DOF's des vollen FE-Modells Aufschluss darüber, welche Elemente der FE-Vernetzung unter der gegebenen Belastung einen wesentlichen Einfluss auf das Lösungsverhalten haben und welche weniger. Nach Ghavamian et al. (2017) kann folgende Umformung für ${}^n\mathbf{f}_{int}$ und ${}^n\mathbf{K} = \left. \frac{\partial \mathbf{f}_{int}}{\partial \boldsymbol{\Theta}} \right|_n$ zum Lastschritt n durchgeführt werden

$$\begin{aligned} {}^n\mathbf{f}_{int} &\approx \boldsymbol{\Theta}_f \left(\mathbf{P}^T \boldsymbol{\Theta}_f \right)^{-1} \underbrace{\mathbf{P}^T {}^n\mathbf{f}_{int}}_{{}^n\mathbf{f}_{int}^{\text{DEIM}}} \\ {}^n\mathbf{K} &\approx \boldsymbol{\Theta}_f \left(\mathbf{P}^T \boldsymbol{\Theta}_f \right)^{-1} \underbrace{\mathbf{P}^T {}^n\mathbf{K}}_{{}^n\mathbf{K}^{\text{DEIM}}} . \end{aligned} \quad (2.8)$$

Der entscheidende „DEIM-Part“ in der Modellreduktion ist die Berechnung bzw. genauer gesagt das Assemblieren von ${}^n\mathbf{f}_{int}^{\text{DEIM}}$ und ${}^n\mathbf{K}^{\text{DEIM}}$. Da die Index-Matrix \mathbf{P} dünn besetzt ist, reicht es für den Assemblierungsprozess aus, von vornherein nur die Element(-beiträge) zu berechnen, die nach dem Indices-Vektor $\mathbf{i} = [I1 \cdots Id]$ überhaupt relevant sind. Ist für ein betrachtetes Element mindestens ein zugehöriger (globaler) Freiheitsgrad in \mathbf{i} enthalten, so wird das Element ausgewertet und in die Assemblierung einbezogen. Trifft dies nicht zu, wird das Element weggelassen und verursacht somit auch keinen Berechnungsaufwand. Der inkrementelle oder inkrementell-iterative Lösungsprozess kann dann mit dem reduzierten (linearisierten) System

$$\underbrace{\boldsymbol{\Theta}^T {}^n\mathbf{K} \boldsymbol{\Theta}}_{{}^n\tilde{\mathbf{K}} \in \mathbb{R}^{d \times d}} \boldsymbol{\beta} = \underbrace{\boldsymbol{\Theta}^T {}^{n+1}\mathbf{f}_{ext}}_{{}^{n+1}\tilde{\mathbf{f}}_{ext} \in \mathbb{R}^{d \times 1}} - \underbrace{\boldsymbol{\Theta}^T {}^n\mathbf{f}_{int}}_{{}^n\tilde{\mathbf{f}}_{int} \in \mathbb{R}^{d \times 1}} \quad (2.9)$$

stattfinden, wobei ${}^n\mathbf{f}_{int}$ und ${}^n\mathbf{K}$ die Approximationen aus (2.8) sind. In Ghavamian et al. (2017) wird das Verfahren noch weiter optimiert, indem eine zusätzliche Klassifizierung der in Betracht kommenden nichtlinearen Snapshots vorgeschaltet wird (*engl. Localized Discrete Empirical Interpolation Method*).

Eine weitere Möglichkeit der Hyperreduktion für das POD-Verfahren wird in Martynov und Wever (2019) über das Heranziehen von *Ansatz-Polynomen* vorgestellt. Ohne an dieser Stelle ins Detail zu gehen, soll hier nur die grundlegende Idee vorgestellt werden, in der ein Ansatz der Form

$$\boldsymbol{\Theta}^T \mathbf{f}_{int}(\hat{\mathbf{u}}) \approx \mathbf{a} + \mathbf{b}_i \beta_i + \mathbf{c}_{ij} \beta_i \beta_j + \mathbf{d}_{ijk} \beta_i \beta_j \beta_k \quad (2.10)$$

gemacht wird, wobei auch hier wieder der klassische POD-Ansatz $\hat{\mathbf{u}} \approx \boldsymbol{\Theta} \boldsymbol{\beta}$ integriert ist. Die Größen \mathbf{a} , \mathbf{b}_i , \mathbf{c}_{ij} and \mathbf{d}_{ijk} sind *vektorielle Polynom Koeffizienten*.

Ein ebenfalls effizientes und intrusives Modellreduktionsverfahren wird in der Arbeit von Millán und Arroyo (2013) durch das Heranziehen von *datenbasierten generalisierten Koordinaten* erzielt. Der Ansatz in dieser Arbeit unterscheidet sich von dem bisher vorgestellten Ansatz indem hier vorab eine nichtlineare Dimensionsreduktion (*engl. Nonlinear Dimensionality Reduction, NLDR*) auf Basis der Snapshotmatrix \mathbf{Q} durchgeführt wird. NLDR dient hier zur *Einbettung* der hochaufgelösten $3n$ Koordinaten in die d generalisierten Koordinaten. Millán und Arroyo (2013) zeigen in ihrer Arbeit anhand eines nichtlinearen elastodynamischen mechanischen Systems, wie die generalisierten d Koordinaten zur Formulierung der Systemgleichungen verwendet werden können.

Ein an dieser Stelle ebenfalls erwähnenswerter Ansatz wird in Hernández et al. (2014) mit der *Aufspaltung* der Basis $\boldsymbol{\Theta}$ in rein inelastische $\boldsymbol{\Theta}^{inel}$ und rein elastische Moden $\boldsymbol{\Theta}^{el}$ vorgestellt. In seiner Arbeit wendet er dieses Vorgehen auf repräsentative Volumenelemente (*engl. Representative Volume Element, RVE*) für die Mehrskalen-Homogenisierung (*engl. Multiscale*

(Homogenization) an.

Für Modellreduktionsverfahren, die in der *Online Phase* die zugrunde liegenden Gleichungen des physikalischen Modells nicht mit einbeziehen, ist das „Erkennen“ des Lösungsverhaltens in der *Trainingsphase* von entscheidender Bedeutung. In der angelsächsischen Literatur werden für diese Verfahren weitestgehend die Begriffe *Surrogate Model*, zu deutsch Ersatzmodell oder auch *Metamodel* für diese Art der Modellreduktion verwendet. In dem Lehrbuch von Jiang et al. (2020) werden die Verfahren⁵

- Polynomial Response Surface Models
- Radial Basis Funktion Models
- Support Vector Regressions Models
- Gaussian Process Models
- Backpropagation Neural Network Models

in kompakter Form vorgestellt. Der unterschiedliche mathematische Hintergrund jedes der aufgezählten Verfahren soll an dieser Stelle nicht weiter vertieft werden, zumal im Rahmen dieser Arbeit die **Support Vector Regression** detailliert in Kapitel 5 behandelt wird und als gewähltes Verfahren der Modellreduktion in Kapitel 6 und 7 auf mechanische Systeme angewendet wird. In dem Buch von Shawe-Taylor et al. (2004) wird in dem Kapitel *Pattern Analysis Using Eigen-Decomposition* unter anderem auch eine Einführung in die Hauptkomponentenregression (engl. *Principal Component Regression*, PCR) gegeben. Dieses Regressionsverfahren beinhaltet eine Hauptkomponentenanalyse (engl. *Principal Component Analysis*, PCA), die ein weit verbreitetes Verfahren zur *Dimensionsreduktion* darstellt. Zur Motivation dieses Verfahrens sei ein mechanisches System zugrunde gelegt, bei dem für einen Eingabedatensatz $\mathbf{x}_i = [x_{i1} \cdots x_{im}]^T \in \mathbb{R}^{m \times 1}$ (hier als *Snapshot*) nach Auswertung des Modells ein Ergebnis für die *Quantity of Interest* $\mathbf{y}_i = [y_{i1} \cdots y_{iw}]^T \in \mathbb{R}^{w \times 1}$ zurückgegeben wird. In der *Trainingsphase* werden beispielsweise k hoch aufgelöste Berechnungen durchgeführt und die Ergebnisse in den Matrizen⁶ $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_k] \in \mathbb{R}^{m \times k}$ und $\mathbf{Y} = [\mathbf{y}_1 \cdots \mathbf{y}_k] \in \mathbb{R}^{w \times k}$ abgelegt. Für die Hauptkomponentenanalyse wird zuerst die *Kovarianz Matrix* $\mathbf{C} = \frac{1}{k} \mathbf{X} \mathbf{X}^T \in \mathbb{R}^{m \times m}$ aus der *Snapshot Matrix* $\mathbf{X} \in \mathbb{R}^{m \times k}$ berechnet. Die *Eigenwertberechnung* $k \mathbf{C} \mathbf{v} = \mathbf{v} \lambda$ resultiert in die m Eigenvektor-Eigenwert Paare $(\mathbf{v}_i, \lambda_i)_{i=1 \dots m}$, die in einem direkten Zusammenhang mit den k Eigenvektor-Eigenwert Paaren $(\mathbf{w}_i, \lambda_i)_{i=1 \dots k}$ der *Gramschen Matrix* $\mathbf{G} = \mathbf{X}^T \mathbf{X} \in \mathbb{R}^{k \times k}$ stehen

$$\mathbf{v} = \frac{1}{\sqrt{\lambda_i}} \mathbf{X} \mathbf{w}_i \iff \mathbf{w}_i = \frac{1}{\sqrt{\lambda_i}} \mathbf{X}^T \mathbf{v}_i. \quad (2.11)$$

Die Dimensionsreduktion im eigentlichen Sinne entspricht dann einer Projektion der *Snapshots* $\mathbf{x}_i = [x_{i1} \cdots x_{im}]^T$ oder auch neuer Daten \mathbf{x}^* auf die Hauptachsen \mathbf{v}_i wobei nur eine geringe Anzahl r der „größten“ Hauptachsen herangezogen wird

$$\tilde{\mathbf{x}}_i = \tilde{\mathbf{S}}^T \mathbf{x}_i; \tilde{\mathbf{x}}^* = \tilde{\mathbf{S}}^T \mathbf{x}^*. \quad (2.12)$$

Hierbei enthält die Matrix $\tilde{\mathbf{S}} = [\mathbf{v}_1 \cdots \mathbf{v}_r]$ die ersten r Eigenvektoren von \mathbf{C} und die $\tilde{\mathbf{x}} \in \mathbb{R}^{r \times 1}$ sind die sogenannten Hauptkoordinaten (engl. *Principal Coordinates*) (Shawe-Taylor et al.,

⁵An dieser Stelle werden die englischsprachigen Begriffe verwendet. Aktuelle Beiträge der *Scientific Community* in diesem Feld können bei Kenntnis der englischen Bezeichnungen schneller gefunden und eingeordnet werden.

⁶Im Hinblick auf die bereits vorgestellte *Singulärwertzerlegung* sei $m > k$ vorausgesetzt, was lediglich einer kompakten Darstellung im Rahmen dieses Kapitels dient. In Quarteroni et al. (2011) werden die Fälle $m > k$ und $m < k$ ausführlich beschrieben.

2004). Das Vorgehen der Hauptkomponentenanalyse kann auf *nichtlineare* Zusammenhänge mit Hilfe von *Kernel* Funktionen $k(\mathbf{x}_i, \mathbf{x}_j)$ erweitert werden. Auf *Kernel* Funktionen sowie *Kernel* Matrizen⁷ \mathbf{K} mit den Einträgen $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ wird in dem Kapitel 5.1 detailliert eingegangen. In der *kernelbasierten* Hauptkomponentenanalyse (*engl. Kernel Principal Component Analysis*, Kernel PCA) werden zuerst die Eigenwerte und Eigenvektoren $(\mathbf{w}_i, \lambda_i)_{i=1 \dots k}$ der *Kernel* Matrix \mathbf{K} berechnet. Unter Berücksichtigung des Zusammenhangs aus Gleichung (2.11) kann eine Projektion auch mit den Eigenvektoren von \mathbf{K} durchgeführt werden. Nach Rosipal et al. (2000) oder auch Shawe-Taylor et al. (2004) können jetzt die *nichtlinearen* Hauptkoordinaten (*engl. Nonlinear Principal Components*) $\tilde{\mathbf{x}}'_l$ der *Snapshots* \mathbf{x}_l , mit $l = 1 \dots k$, bezüglich der r größten Eigenvektoren von $\mathbf{K} \in \mathbb{R}^{k \times k}$ wie folgt angegeben werden

$$\begin{aligned}\tilde{x}'_{ij}(\mathbf{x}_l) &= \frac{1}{\sqrt{\lambda_i}} \sum_{i=1}^k w_{ji} k(\mathbf{x}_i, \mathbf{x}_l) \\ \mathbf{w}_j &= [w_{j1} \dots w_{jk}]^T, \quad j = 1 \dots r \\ \tilde{\mathbf{x}}'_l &= [\tilde{x}'_{l1} \dots \tilde{x}'_{lr}]^T, \quad l = 1 \dots k.\end{aligned}\tag{2.13}$$

Für einen neuen Datenpunkt \mathbf{x}^* , der nicht im Bereich der Trainingsdaten (*Snapshots*) \mathbf{X} liegt, muss in Gleichung (2.13) lediglich \mathbf{x}_l durch \mathbf{x}^* ersetzt werden. Für die Hauptkomponentenregression (Kernel PCR) werden die Daten $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_k] \in \mathbb{R}^{m \times k}$ und $\mathbf{Y} = [\mathbf{y}_1 \dots \mathbf{y}_k] \in \mathbb{R}^{w \times k}$ zusammen in Betracht gezogen, um ein reduziertes (Regressions-)Modell $\mathcal{F}_{(\mathbf{x}^*)}^{RM} = \hat{\mathbf{y}}$ für das hoch aufgelöste Modell $\mathcal{F}_{(\mathbf{x}^*)}^{HF} = \mathbf{y}$ zu erstellen. Nach Shawe-Taylor et al. (2004) kann die Regression wie folgt berechnet werden

$$\begin{aligned}\zeta_j &= \sum_{i=1}^r \frac{1}{\sqrt{\lambda_i}} \left(\mathbf{w}_i^T \check{\mathbf{y}}_j \right) \mathbf{w}_i \\ \check{\mathbf{y}}_j &\mapsto j\text{-te Spalte von } \mathbf{Y}^T \in \mathbb{R}^{k \times w} \\ \zeta_j &= [\zeta_{1j} \dots \zeta_{kj}], \quad j = 1 \dots w \\ \hat{y}_j &= \sum_{i=1}^k \zeta_{ij} k(\mathbf{x}_i, \mathbf{x}^*) \\ \hat{\mathbf{y}} &= [\hat{y}_1 \dots \hat{y}_w]^T,\end{aligned}\tag{2.14}$$

wobei die ζ_{ij} als Gewichtungsfaktoren bezüglich der $j = 1 \dots w$ Ergebnisfreiheitsgrade von \mathbf{y} betrachtet werden können. Die Verwendung einer *zentrierten Kernel* Matrix

$$\bar{\mathbf{K}} = \frac{1}{k} \mathbf{j} \mathbf{j}^T \mathbf{K} - \frac{1}{k} \mathbf{K} \mathbf{j} \mathbf{j}^T + \frac{1}{k^2} \left(\mathbf{j}^T \mathbf{K} \mathbf{j} \right) \mathbf{j} \mathbf{j}^T,\tag{2.15}$$

mit $\mathbf{j}^T = [1 \dots 1] \in \mathbb{R}^{1 \times k}$, bringt durch eine Transformation des „Datenursprungs“ in den „Datenschwerpunkt“ für die Berechnung Vorteile, ist aber nicht zwingend erforderlich (Shawe-Taylor et al., 2004). In Wibowo und Yamamoto (2012) wird eine überarbeitete Version der kernelbasierten Hauptkomponentenregression (*engl. Revised Kernel PCR*) vorgestellt, die die Auswahl der in Betracht zu ziehenden Eigenvektoren optimiert.

Die vorliegende Arbeit zielt auf die Anwendung der Support Vector Regression (SVR) im Rahmen der Strukturmechanik mit elasto-plastischen Materialverhalten ab. SVR wurde bereits in zahlreichen mechanische Anwendungen erfolgreich eingesetzt. In der Arbeit von Zhu et al.

⁷Wenn für die *Kernel* Funktion ein *linearer* Ansatz der Form $k(\mathbf{x}_i, \mathbf{x}_j) := \mathbf{x}_i^T \mathbf{x}_j$ verwendet wird so entspricht die *Kernel* Matrix \mathbf{K} der *Gramschen* Matrix $\mathbf{G} = \mathbf{X}^T \mathbf{X}$.

(2012) wird die Support Vector Regression im Bereich der *Crashtest* Simulation herangezogen. In einer Studie wird die Widerstandskraft gegen das Eindringen des Fahrzeugdaches optimiert. In einer anderen Studie geht es um die Optimierung der Fahrzeugfront im Hinblick auf eine Leichtbauweise. In Clarke et al. (2004) wird SVR mit den Verfahren *Response Surface Methodology*, *Radial Basis Functions*, *Kriging* und *Multivariate Adaptive Regression Splines* verglichen. Die Anwendung findet hier in erster Linie auf Stab- und Balkenstrukturen sowie einen Druckkessel und eine Stahlfeder statt. Der Autor zeigt, dass die Support Vector Regression hierbei den anderen genannten Verfahren überlegen ist. In Wang et al. (2009) findet die Anwendung von SVR auf die Optimierung eines Blechumformenprozesses statt.

Nicht-intrusive Methoden können auch gezielt nur in Teilbereichen eines mechanischen Berechnungsmodells eingesetzt werden. In Hartmaier (2020) wird unter der Überschrift *Datenorientierte Konstitutivmodellierung der Metalplastizität* die *Support Vector Classifizierung* (SVC) genutzt, um ein Ersatzmodell für die Fließfunktion der Elasto-Plastischen-Materialroutine zu generieren. Mit dieser *maschinell gelernten* Fließfunktion kann die Auswertung der Fließfunktion umgangen werden. Der wesentliche Vorteil gegenüber der konventionellen Auswertung der Fließfunktion ist nach Hartmaier (2020) der, dass neben den Spannungen auch mikrostrukturelle Parameter des Materials selbst mit in die Auswertung einbezogen werden können. Eine Einführung in das Thema *Datenbasierte Konstitutivgleichungen* mit einer Übersicht einiger Arbeiten zu diesem Thema wird in Chinesta et al. (2020) gegeben.

In der praktischen Anwendung von Parameterstudien sind häufig mechanische Größen von Interesse, die aus den Ergebnissen von Finite Elemente Berechnungen abgeleitet werden. Beispiele hierfür sind Verschiebungen an einem *ausgezeichneten* Punkt des FE-Modells, maximale Spannungsspitzen in einem *kritischen* Bereich oder der *quantitative* Anteil an äquivalenter plastischer Dehnung in Bezug auf das Gesamtvolumen des FE-Modells, um nur einige Möglichkeiten zu nennen. Das Antwortverhalten dieser Größen muss nicht zwangsläufig „glatt“ sein, sodass vielfach Unstetigkeiten und Sprünge sowie konstante Bereiche die Erstellung eines reduzierten Modells erschweren. Daher soll im Rahmen dieser Arbeit mittels der *Support Vektor Regression* und einer geeigneten Wahl der *Kernel* Funktion sowie der regulierenden Parameter eine Möglichkeit aufgezeigt werden, wie die genannten Schwierigkeiten im Rahmen einer Modellreduktion berücksichtigt werden können.

3 Kontinuumsmechanik

In diesem Kapitel werden die notwendigen kontinuumsmechanischen Grundlagen für den Rahmen dieser Arbeit zugrunde gelegt. Die Darlegungen orientieren sich hier an den Büchern von Stein (1989), de Souza Neto et al. (2008), Parisch (2003), Wriggers (2013), de Borst et al. (2014), Kreißig und Benedix (2013) und Ibrahimbegovic (2009).

3.1 Kinematik

Die Beschreibung der von einem Körper ausgeführten Bewegung wird in der Festkörpermechanik der sogenannten *Kinematik* zugeordnet. Ein Körper kann dabei eine Bewegung mit oder ohne Volumen- und Formänderung ausführen. Letzteres wird als *Starrkörperbewegung* bezeichnet, wo hingegen Ersteres eine *Deformation* beinhaltet. Für die Untersuchung der mechanischen Beanspruchung von Strukturen ist es notwendig, *Verzerrungsmaße* auf der Grundlage von Deformationen zu definieren. In diesem Kapitel soll ein geeignetes Verzerrungsmaß für die Theorie *kleiner* Deformationen hergeleitet werden. Unter kleinen Deformationen soll verstanden werden, dass sowohl die *Verschiebungen* als auch die *Verschiebungsgradienten* klein gegenüber den Bauteilabmessungen sind. Die Theorie kleiner Deformationen bringt weitere vereinfachende Annahmen mit sich, die am Ende dieses Kapitels genannt werden und Einfluss auf die Folgekapitel haben. Die Beschreibung der Kinematik im *dreidimensionalen euklidischen Raum* kann entweder in *kartesischen* oder *krummlinigen* Koordinaten stattfinden, wobei für Letztere eine zweckmäßige Spezialisierung mit sogenannten *konvektiven* Koordinaten möglich ist (Steinmann, 2015). Im Rahmen dieser Arbeit findet die Beschreibung der Kontinuumsmechanik ausschließlich in *kartesischen* Koordinaten statt.

Der betrachtete Körper nimmt zu jedem Zeitpunkt eine Position im *dreidimensionalen Euklidischen Anschauungsraum* \mathcal{E} ein. Zu einem bestimmten Zeitpunkt¹ $t = 0$ nehme er die (meist unverformte) *Konfiguration* \mathcal{B} ein. Zu einem späteren Zeitpunkt nehme er die Konfiguration $\varphi(\mathcal{B})$ ein. In der sogenannten *Lagrange'schen* Formulierung wird die Konfiguration \mathcal{B} als *Referenzkonfiguration* bezeichnet und die Konfiguration $\varphi(\mathcal{B})$ als *Momentankonfiguration*. Das Wesentliche an dieser Formulierung ist, dass zu jedem Zeitpunkt eine *Abbildung* von der Referenzkonfiguration auf die Momentankonfiguration angegeben werden kann,

$$\mathcal{B} \mapsto \varphi(\mathcal{B}) . \tag{3.1}$$

Für die Beschreibung der Referenzkonfiguration wird ein globales *kartesisches* Basissystem $\{\mathbf{E}_i\}_{i=1,2,3}$ eingeführt. Für die Beschreibung der Momentankonfiguration wird entsprechend ein globales *kartesisches* Basissystem $\{\mathbf{e}_i\}_{i=1,2,3}$ zugrunde gelegt. Die beiden globalen Basissysteme fallen im Rahmen dieser Arbeit zusammen bzw. werden übereinander gelegt, was das Transformationsverhalten zwischen den beiden Systemen erheblich vereinfacht (Parisch, 2003). Die Transformation vereinfacht sich in diesem Fall zu (Stein, 1989; Ibrahimbegovic, 2009)

$$\begin{aligned} \mathbf{E}_i \cdot \mathbf{e}_j &= \delta_{ij} \\ \mathbf{E}_i &= \delta_{ij} \mathbf{e}_j . \end{aligned} \tag{3.2}$$

Punkte, die zur Referenzkonfiguration gehören, werden durch die Ortsvektoren $\mathbf{X} = X_i \mathbf{E}_i$ ausgedrückt und Punkte, die zur Momentankonfiguration gehören, durch die Ortsvektoren $\mathbf{x} = x_i \mathbf{e}_i$. Aus der Gleichung (3.1) kann eine punktweise, funktionelle Abhängigkeit $\mathbf{x} = \mathbf{x}(\mathbf{X})$

¹Im Weiteren wird der Übersichtlichkeit halber auf eine zusätzlichen Indizierung „ t “ für die Zeitabhängigkeit an den Größen verzichtet.

zwischen diesen Ortsvektoren hergestellt werden. Es ist nun auch möglich, einen auf die Referenzkonfiguration bezogenen *Verschiebungsvektor* \mathbf{u} einzuführen (Stein, 1989),

$$\mathbf{u} = \mathbf{x} - \mathbf{X} = (\delta_{ij}x_i - X_j) \mathbf{E}_j \quad (3.3)$$

Die bisher aufgeführten kinematischen Zusammenhänge sind in der Abbildung 3.1 dargestellt. Für die Herleitung eines Verzerrungsmaßes ist zuerst ein *Gradient* einzuführen, den sogenannten *Deformationsgradient* (Wriggers, 2013)

$$\mathbf{F} = \frac{\partial x_i}{\partial X_j} \mathbf{e}_i \otimes \mathbf{E}_j . \quad (3.4)$$

Die Koeffizienten des Deformationsgradienten können auch in einer 3×3 -Matrix $[F_{ij}] = [\partial x_i / \partial X_j]$

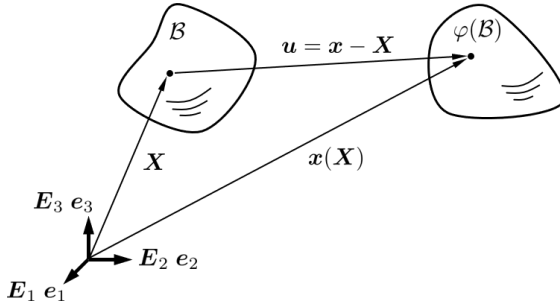


Abbildung 3.1: Verformungskinematik des Körpers \mathcal{B}

zusammengefasst werden. Werden das Transformationsgesetz (3.2) zusammen mit der Gleichung (3.3) für den Verschiebungsvektor und ein auf die Referenzkonfiguration bezogener *Nabla*-Operator in Vektorform $\nabla = \frac{\partial}{\partial X_j} \mathbf{E}_j$ herangezogen, so lässt sich der Deformationsgradient auch in der folgenden Form angeben,

$$\begin{aligned} \mathbf{x} &= u_j \delta_{ji} \mathbf{e}_i + X_j \delta_{ji} \mathbf{e}_i \\ \mathbf{1} &= X_j \delta_{ji} \mathbf{e}_i \otimes \frac{\partial}{\partial X_j} \mathbf{E}_j = \delta_{ij} \mathbf{e}_i \otimes \mathbf{E}_j \\ \mathbf{H} &= u_j \delta_{ji} \mathbf{e}_i \otimes \frac{\partial}{\partial X_j} \mathbf{E}_j = \frac{\partial u_i}{\partial X_j} \mathbf{e}_i \otimes \mathbf{E}_j \\ \mathbf{F} &= \mathbf{1} + \mathbf{H} = \left(\delta_{ij} + \frac{\partial u_i}{\partial X_j} \right) \mathbf{e}_i \otimes \mathbf{E}_j \\ \mathbf{F} &= \mathbf{1} + \mathbf{H} = \mathbf{1} + \mathbf{u} \otimes \nabla . \end{aligned} \quad (3.5)$$

In der Darstellung (3.5) ist $\mathbf{H} = \mathbf{u} \otimes \nabla$ der sogenannte *Verschiebungsgradient* und $\mathbf{1}$ der *Einheitstensor*, beide jeweils mit *Tensorbasis*-Anteilen in der Referenz- und Momentankonfiguration: $\mathbf{e}_i \otimes \mathbf{E}_j$. Der Deformationsgradient ist ein Tensor zweiter Stufe, genauer gesagt ein

*Zweifeldtensor*², was ihm eine entscheidende Rolle im Rahmen von *geometrisch nichtlinearen* Berechnungen beimisst. Er „steht“ sozusagen mit einem Bein in der Referenzkonfiguration und mit dem anderen Bein in der Momentankonfiguration (Parisch, 2003).

Für Verzerrungsmaße müssen die der Verformung überlagerten Starrkörperbewegungen isoliert werden. Die *reinen* Translationen sind schon durch die Bildung des Deformationsgradienten eliminiert (de Borst et al., 2014). Zur Eliminierung der Starrkörperrotationen wird der auf die Referenzkonfiguration bezogene *Green-Lagrange'sche* Verzerrungstensor \mathbf{E} wie folgt definiert (Stein, 1989; Wriggers, 2013)

$$\begin{aligned} \mathbf{C} &= \mathbf{F}^T \cdot \mathbf{F} = F_{ki} F_{lj} \mathbf{E}_i \otimes \overbrace{\mathbf{e}_k \cdot \mathbf{e}_l}^{\delta_{kl}} \otimes \mathbf{E}_j \\ \mathbf{C} &= F_{li} F_{lj} \mathbf{E}_i \otimes \mathbf{E}_j = C_{ij} \mathbf{E}_i \otimes \mathbf{E}_j \\ \mathbf{1} &= \delta_{ij} \mathbf{E}_i \otimes \mathbf{E}_j \\ \mathbf{E} &:= \frac{1}{2} (\mathbf{C} - \mathbf{1}) = \frac{1}{2} (C_{ij} - \delta_{ij}) \mathbf{E}_i \otimes \mathbf{E}_j. \end{aligned} \quad (3.6)$$

Die Größe \mathbf{C} wird als *rechter Cauchy-Green* Tensor bezeichnet und ist auf die Referenzkonfiguration bezogen (Wriggers, 2013). In (3.6) ist der Identitätstensor $\mathbf{1}$, anders als in der Darstellung (3.5), mit beiden Basisvektoren in der Referenzkonfiguration definiert. Der *Green-Lagrange'sche* Verzerrungstensor kann auch mit dem Verschiebungsvektor \mathbf{u} und dem daraus abgeleiteten *Verschiebungsgradienten* \mathbf{H} dargestellt werden, was für die weiteren Herleitungen vorteilhaft ist,

$$\mathbf{E} := \frac{1}{2} (\mathbf{H} + \mathbf{H}^T + \mathbf{H}^T \cdot \mathbf{H}). \quad (3.7)$$

Für eine konsistente Basisdarstellung ist bei den ersten beiden Summanden auf die Transformation (3.2) zurückzugreifen, sodass $\mathbf{H} = \partial u_i / \partial X_j \mathbf{e}_i \otimes \mathbf{E}_j$ mit $u_i \mathbf{e}_i = u_j \delta_{ji} \mathbf{E}_i$ als $\mathbf{H} = \partial u_i / \partial X_j \mathbf{E}_i \otimes \mathbf{E}_j$ dargestellt werden kann. Bei dem Produktterm in (3.7) ist der Bezug zur Basis in der Referenzkonfiguration analog zur Berechnung in (3.6) automatisch gegeben. Somit nimmt der *Green-Lagrange'sche* Verzerrungstensor in der Basisdarstellung folgende Form an,

$$\mathbf{E} = \frac{1}{2} \left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} + \frac{\partial u_k}{\partial X_i} \frac{\partial u_k}{\partial X_j} \right) \mathbf{E}_i \otimes \mathbf{E}_j. \quad (3.8)$$

Der *Green-Lagrange'sche* Verzerrungstensor eignet sich gut um Verzerrungen auch unter großen Deformationen, also auch großen Starrkörperrotationen und Formänderungen, zu bestimmen. Bei vielen Ingenieur Anwendungen ist diese Berücksichtigung nicht erforderlich, sodass nach Wriggers (2013) unter der Voraussetzung kleiner Verschiebungsgradienten ($\|\mathbf{H}\| \ll 1$), der quadratische Teil vernachlässigt werden kann. Das so gewonnene Verzerrungsmaß wird im Rahmen kleiner Deformationen verwendet und als *linearer* Verzerrungstensor $\boldsymbol{\varepsilon}$ bezeichnet,

$$\boldsymbol{\varepsilon} = \frac{1}{2} \left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} \right) \mathbf{E}_i \otimes \mathbf{E}_j. \quad (3.9)$$

²Die Kennzeichnung des Deformationsgradienten als Zweifeldtensor hat ihren Ursprung in der Differentialgeometrie. Dort wird gezeigt, dass der Deformationsgradient $\mathbf{F}|_X := T_X \mathcal{B} \rightarrow T_x \varphi(\mathcal{B})$ eine lineare Abbildung zwischen den Tangentialräumen $T_X \mathcal{B}$ und $T_x \varphi(\mathcal{B})$ zu den Punkten $X \in \mathcal{B}$ und $x \in \varphi(\mathcal{B})$ herstellt. Lehrbücher, die diesen Ansatz verfolgen, sind beispielsweise Bertram (1989), Marsden und Hughes (1994), Epstein (2010), Romano (2014), Steinmann (2015) und Lychev und Koifman (2019). Aber auch Publikationen und Veröffentlichungen wie Aubram (2009), Miehe (1993), Stumpf und Hoppe (1997), Kollmann und Hackenberg (1993), Epstein und Segev (1980), Barthold (2002), Franze (2013) und Romano et al. (2014) behandeln die Kontinuumsmechanik auf der Grundlage der Differentialgeometrie. In dem Lehrbuch von Mühlich (2017) wird der notwendige mathematische Rahmen Ingenieursstudenten mit der nötigen Sauberkeit zugänglich gemacht, wobei das Augenmerk immer auf der Mechanik liegt.

Die Annahme kleiner Deformationen hat über das lineare Verzerrungsmaß hinaus weitere Konsequenzen für die Aufstellung und Berechnung der zugrundeliegenden Gleichungen in der Festkörpermechanik. So können Integrationen und Differenzierungen mit Bezug auf die Momentankonfiguration mit Bezug auf die Referenzkonfiguration ausgeführt werden (Ibrahimbegovic, 2009). Dies bedingt, dass die Gleichgewichtsbedingungen bzw. Bilanzgleichungen am unverformten Körper, also in der Referenzkonfiguration aufgestellt werden können, was die Berechnungen erheblich vereinfacht. Darüber hinaus können *virtuelle* Verschiebungsfelder im Rahmen der Finite Elemente Methode mit Bezug auf die Referenzkonfiguration angesetzt werden, obgleich sie auf der Momentankonfiguration „wirken“. Diese wichtigen Vereinfachungen sind in den Definitionsgleichungen (3.10), (3.11) und (3.12) dargestellt,

$$\mathbf{X} \approx \mathbf{x} \quad (3.10)$$

$$\begin{aligned} \int_{\mathcal{B}} dV &\approx \int_{\varphi(\mathcal{B})} dV^\varphi \\ \int_{\partial\mathcal{B}} dA &\approx \int_{\partial\varphi(\mathcal{B})} dA^\varphi \end{aligned} \quad (3.11)$$

$$\frac{\partial}{\partial X_i} \approx \frac{\partial}{\partial x_i}. \quad (3.12)$$

Der schon eingeführte *Nabla*-Operator in Vektorform wird im Hinblick auf Gleichung (3.12) im Rahmen kleiner Deformationen als $\nabla = \frac{\partial}{\partial x_j} \mathbf{e}_j$ mit $\mathbf{u} = u_i \mathbf{e}_i$ verwendet, sodass

$$\nabla \mathbf{u} := \mathbf{u} \otimes \nabla = \frac{\partial u_i}{\partial x_j} \mathbf{e}_i \otimes \mathbf{e}_j \quad (3.13)$$

angesetzt werden kann. In der Gleichung (3.13) wird auf der linken Seite der *Operator* Charakter deutlich, wo hingegen der mittlere Term die tensorielle Berechnung veranschaulicht. Der Gradient erhöht somit die tensorielle Stufe um „eins“. Der auf die Referenzkonfiguration bezogene Gradientenoperator wird oft mit $\text{Grad}(\cdot)$ und der auf die Momentankonfiguration mit $\text{grad}(\cdot)$ bezeichnet, was einer besseren Unterscheidung dienlich ist (Wriggers, 2013). Im Fall kleiner Deformationen gilt $\text{Grad}(\cdot) \approx \text{grad}(\cdot)$, sodass wie in Gleichung (3.13) $\nabla \mathbf{u} := \text{grad} \mathbf{u}$ geschrieben werden kann. Mit der Formulierung aus Gleichung (3.13) kann der *lineare* Verzerrungstensor (3.9) auch folgendermaßen angegeben werden (Ibrahimbegovic, 2009),

$$\boldsymbol{\varepsilon} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \mathbf{e}_i \otimes \mathbf{e}_j := \frac{1}{2} \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T \right). \quad (3.14)$$

Die Darstellung (3.14) in der Basis $\{\mathbf{e}_i\}_{i=1,2,3}$ hat für die folgenden Kapitel entscheidende Vorteile, da beispielsweise der *Cauchy*-Spannungstensor $\boldsymbol{\sigma}$ in der Momentankonfiguration definiert ist und somit von „Natur aus“ in dieser Basis dargestellt wird. Die Größen $\boldsymbol{\sigma}$ und $\boldsymbol{\varepsilon}$ können somit im Rahmen kleiner Deformation „direkt“ miteinander verknüpft werden, wobei dann alle Rechenoperationen, wie Differenzierung und Integration, in den Koordinaten der Referenzkonfiguration ausgeführt werden. Die Koordinaten des linearen Verzerrungstensors können in einer 3×3 -Matrix zusammengefasst werden,

$$[\boldsymbol{\varepsilon}_{ij}] = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33} \end{bmatrix}, \quad (3.15)$$

wobei die einzelnen Einträge ε_{ij} der Gleichung (3.14) zu entnehmen sind. Dadurch, dass der *lineare* Verzerrungstensor $\boldsymbol{\varepsilon}$ symmetrisch ist, lässt er sich im Rahmen der sogenannten *Voigt*-Notation als 6×1 -Spaltenmatrix angeben

$$\boldsymbol{\varepsilon} = [\varepsilon_{11} \ \varepsilon_{22} \ \varepsilon_{33} \ 2\varepsilon_{12} \ 2\varepsilon_{23} \ 2\varepsilon_{13}]^T . \quad (3.16)$$

Die Unterscheidung zwischen einer tensoriellen Darstellung, wie in (3.14), oder der Darstellung (3.16) in *Voigt*-Notation, wird aus dem Zusammenhang klar, sodass $\boldsymbol{\varepsilon}$ stellvertretend für beides verwendet wird. Im Folgenden wird der Begriff „Verzerrungstensor“ synonym für *linearer* Verzerrungstensor verwendet, da im Rahmen kleiner Deformationen hier kein Missverständnis auftreten kann.

3.2 Spannungen

Die Beanspruchung eines Bauteils unter Last wird maßgebend durch die Magnitude der inneren, die Struktur zusammenhaltenden Bindungskräfte, gekennzeichnet. In der Festkörpermechanik dient das Konzept der *Spannungen* dazu, diese Beanspruchung punktuell zu bewerten. Die *wahre*, in der betrachteten Struktur vorliegende Spannung, wird als *Cauchy*-Spannung bezeichnet und mit dem Symbol $\boldsymbol{\sigma}$ gekennzeichnet. Im Folgenden soll die *Cauchy*-Spannung definiert und daraus folgende Konzepte und Berechnungsvorschriften aufgeführt werden.

Der betrachtete Körper befinde sich in einer belasteten und verformten Lage, die von einer zugrunde gelegten kartesischen Basis $\{\mathbf{e}_i\}_{i=1,2,3}$ aus beschrieben wird (Momentankonfiguration, siehe Annahmen Kapitel 3.1). Durch den Körper wird „gedanklich“ ein Schnitt gelegt, siehe Abbildung 3.2. In der Schnittfläche wird ein Flächenelement ΔA betrachtet, auf das durch die freigelegten inneren Kräfte eine Schnittkraft $\Delta \mathbf{f}$ wirkt. Nach de Borst et al. (2014) kann durch die Grenzwertbildung

$$\mathbf{t} = \lim_{\Delta a \rightarrow 0} \frac{\Delta \mathbf{f}}{\Delta A} \quad (3.17)$$

der *Spannungsvektor* \mathbf{t} ermittelt werden. Dem Flächenstück ΔA ist eine Orientierung in Form eines Normalenvektors \mathbf{n} zuzuordnen, sodass auch von einem orientierten Flächenelement $\Delta \mathbf{A} = \Delta A \mathbf{n}$ gesprochen werden kann. Der Spannungsvektor \mathbf{t} hängt also nicht nur von der Richtung der Schnittkraft $\Delta \mathbf{f}$, sondern auch von der Flächenorientierung \mathbf{n} ab. Eine direkte Verbindung zwischen dem Spannungsvektor \mathbf{t} und dem Flächen-Normalenvektor \mathbf{n} wird durch die *Cauchysche* Formel (Kreißig und Benedix, 2013) hergestellt,

$$\mathbf{t} = \boldsymbol{\sigma} \cdot \mathbf{n} . \quad (3.18)$$

Der in Gleichung (3.18) enthaltene *Cauchysche* Spannungstensor $\boldsymbol{\sigma}$ (Tensor zweiter Stufe) repräsentiert die Belastung in einem Punkt des Körpers. Seine Koordinaten σ_{ij} können durch das Kräftegleichgewicht an einem infinitesimalen Tetraeder (auch genannt *Cauchy's* Tetraeder (Ibrahimbegovic, 2009)) ermittelt werden. Mit Bezug auf die Basis $\{\mathbf{e}_i\}_{i=1,2,3}$ kann der *Cauchysche* Spannungstensor folgendermaßen angegeben werden,

$$\boldsymbol{\sigma} = \sigma_{ij} \mathbf{e}_i \otimes \mathbf{e}_j . \quad (3.19)$$

Die in Gleichung (3.18) aufgeführte *Abbildung* zwischen zwei Vektoren kann nun folgendermaßen aufgeschrieben werden,

$$\mathbf{t} = \sigma_{ij} n_k \mathbf{e}_i \otimes \overbrace{\mathbf{e}_j \cdot \mathbf{e}_k}^{\delta_{jk}} = \sigma_{ij} n_j \mathbf{e}_i , \quad (3.20)$$

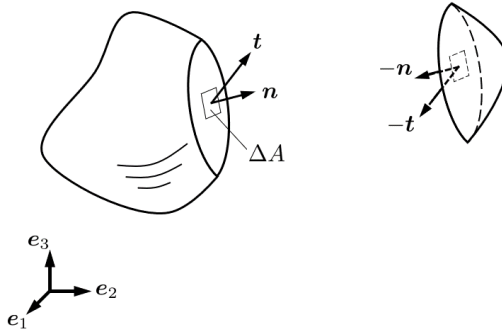


Abbildung 3.2: Spannungen am Schnitt durch den Körper \mathcal{B}

wobei $\mathbf{n} = n_k \mathbf{e}_k$ den Flächen-Normalenvektor in der Basisdarstellung repräsentiert. Die Abbildung 3.2 verdeutlicht die Ausführungen, wobei zusätzlich das sogenannte *Schnittprinzip* deutlich wird. Demnach wirken an dem gegenüberliegenden *Schnittufer* die beschriebenen Größen mit entgegengesetztem Vorzeichen. Die Koordinaten des *Cauchy Spannungstensors*, im Folgenden nur noch Spannungstensor genannt, können analog zu den Verzerrungen im Kapitel 3.1 in einer 3×3 -Matrix zusammengefasst werden

$$[\sigma_{ij}] = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}. \quad (3.21)$$

Der Spannungstensor stellt analog zum Verzerrungstensor einen symmetrischen Tensor zweiter Stufe dar und lässt sich somit in *Voigt-Notation* als 6×1 -Spaltenmatrix angeben

$$\boldsymbol{\sigma} = [\sigma_{11} \ \sigma_{22} \ \sigma_{33} \ \sigma_{12} \ \sigma_{23} \ \sigma_{13}]^T. \quad (3.22)$$

Die Unterscheidung zwischen der Darstellung in (3.22) und (3.19) wird aus dem Zusammenhang klar. Anstelle der Darstellung in der Basis $\{\mathbf{e}_i\}_{i=1,2,3}$ kann eine Darstellung des Spannungszustandes in einem hierzu gedrehten Koordinatensystem stattfinden. Das sogenannte *Hauptachsensystem* $\{\mathbf{e}_i^*\}_{i=I,II,III}$ ist dadurch ausgezeichnet, dass der Spannungstensor lediglich durch Normalspannungen dargestellt werden kann. Die *Hauptachsen* und die zugehörigen sogenannten *Hauptspannungen* lassen sich durch eine Eigenwertberechnung ermitteln. Die Koordinatenmatrix ist dann lediglich auf der Diagonalen besetzt

$$\begin{aligned} \{\mathbf{e}_i\}_{i=1,2,3} &\longrightarrow \{\mathbf{e}_i^*\}_{i=I,II,III} \\ \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} &\longrightarrow \begin{bmatrix} \sigma_I & 0 & 0 \\ 0 & \sigma_{II} & 0 \\ 0 & 0 & \sigma_{III} \end{bmatrix}. \end{aligned} \quad (3.23)$$

Nach der Ermittlung der Hauptspannungen kann ein Spannungszustand auch im sogenannten *Hauptspannungsraum* dargestellt werden, wobei die Koordinatenachsen hier als $\{\boldsymbol{\sigma}_i\}_{i=I,II,III}$ gekennzeichnet sind. In dem Abschnitt über Materialeigenschaften, Kapitel 3.4, werden weitere Eigenschaften des Spannungstensors aufgeführt und verwendet. Letztendlich ist es das Ziel, eine Verbindung zwischen den Verzerrungen und Spannungen für die Lösung mechanischer Analysen herzustellen.

3.3 Bilanzgleichungen

Für die Herleitung der sogenannten *starken* Form des Gleichgewichts wird der Weg über die *Impulsbilanz* gewählt. Eine andere Möglichkeit ist die Aufstellung des Kräftegleichgewichts am infinitesimalen Quader im Inneren des Körpers.

Die Impulsbilanz am Körper \mathcal{B} kann beispielsweise nach de Borst et al. (2014) mit

$$\int_{\partial\mathcal{B}} \mathbf{t} \, dA + \int_{\mathcal{B}} \rho \mathbf{g} \, dV = \int_{\mathcal{B}} \rho \ddot{\mathbf{u}} \, dV \quad (3.24)$$

angegeben werden, wobei dA ein Flächenelement auf der Oberfläche $\partial\mathcal{B}$ des Körpers ist, auf das eine Oberflächenlast \mathbf{t} (engl. *surface traction*) wirkt. Die übrigen Integrale sind über das Volumen des Körpers \mathcal{B} zu führen, mit dem infinitesimalen Volumenelement dV im Integranden. Die Dichte ρ , multipliziert mit der Erdbeschleunigung \mathbf{g} und dem Volumenelement dV , ergibt die punktuell im Körper vorliegende Gewichtskraft. Auf der rechten Seite in (3.24) steht der *Trägheitsterm* in Form des Produktes aus der Dichte ρ , der Beschleunigung $\ddot{\mathbf{u}} = \frac{\partial^2}{\partial t^2} \mathbf{u}$ und dem Volumenelement dV . Die Oberflächenlast \mathbf{t} kann mit *Cauchy's* Formel (3.18) und dem zum Flächenelement dA zugehörigen Flächen-Normalenvektor \mathbf{n} auch als $\mathbf{t} = \boldsymbol{\sigma} \cdot \mathbf{n}$ geschrieben werden. Mithilfe des *Gauß'schen* Integralsatzes $\int_{\partial\mathcal{B}} \boldsymbol{\sigma} \cdot \mathbf{n} \, dA = \int_{\mathcal{B}} \nabla \cdot \boldsymbol{\sigma} \, dV$ kann die Gleichung (3.24) auch folgendermaßen umgeschrieben werden

$$\int_{\mathcal{B}} (\nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g} - \rho \ddot{\mathbf{u}}) \, dV = \mathbf{0} . \quad (3.25)$$

Das Skalarprodukt mit dem *Nabla*-Operator in Vektorform entspricht der Divergenzbildung, sodass auch die Schreibweise $\nabla \cdot \boldsymbol{\sigma} := \text{div } \boldsymbol{\sigma}$ üblich ist. Die Divergenzbildung verringert die tensorielle Stufe um „eins“. Die Divergenz des Spannungstensors kann in der Basis $\{\mathbf{e}_i\}_{i=1,2,3}$ angegeben werden

$$\nabla \cdot \boldsymbol{\sigma} = \frac{\partial \sigma_{ij}}{\partial x_k} \mathbf{e}_i \otimes \overbrace{\mathbf{e}_j \cdot \mathbf{e}_k}^{\delta_{jk}} = \frac{\partial \sigma_{ij}}{\partial x_j} \mathbf{e}_i := \sigma_{ij,j} \mathbf{e}_i . \quad (3.26)$$

Die Impulsbilanz (3.25) kann nur erfüllt werden, wenn der Klammerausdruck auf der linken Seite Null ist. Dies ist gerade die *lokale*, also punktuelle Erfüllung des Gleichgewichts, sodass in jedem Punkt des Körpers

$$\nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g} - \rho \ddot{\mathbf{u}} = \mathbf{0} \quad (3.27)$$

erfüllt sein muss, mit dem Nullvektor $\mathbf{0}$ auf der rechten Seite. Im Rahmen dieser Arbeit werden nur *quasi statische* Prozesse betrachtet. Der *Trägheitseffekt* der Masse muss demnach nicht berücksichtigt werden, da die zweite Zeitableitung der Verschiebung $\ddot{\mathbf{u}} = \frac{\partial^2}{\partial t^2} \mathbf{u}$ vernachlässigbar klein ist. Somit ergibt sich die *starke* Form im statischen Fall zu

$$\begin{aligned} \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g} &= \mathbf{0} \\ \sigma_{ij,j} + \rho g_i &= 0 ; \quad i = 1, 2, 3 . \end{aligned} \quad (3.28)$$

Dies sind drei gekoppelte partielle Differenzialgleichungen, was aus der Indexschreibweise in (3.28) deutlich wird. Die *Drehimpulsbilanz* am Körper \mathcal{B} , mit Bezug auf einen Ursprung \mathcal{O} , mündet in die Symmetrie des Spannungstensors

$$\sigma_{ij} = \sigma_{ji} \rightarrow \boldsymbol{\sigma} = \boldsymbol{\sigma}^T . \quad (3.29)$$

Zur Erfüllung des Gleichgewichts sind sowohl Gleichung (3.28) und Gleichung (3.29) zu erfüllen. Letztgenannte Gleichung wird automatisch durch die Verwendung des *Cauchy* Spannungstensors erfüllt (Kreißig und Benedix, 2013).

3.4 Materialgleichungen

Nachdem in den Kapiteln 3.1 und 3.2 der Verzerrungstensor $\boldsymbol{\varepsilon}$ und der Spannungstensor $\boldsymbol{\sigma}$ für die Berechnungen im Rahmen kleiner Deformationen vorgestellt wurden, soll im nächsten Schritt eine Beziehung zwischen diesen beiden Größen hergestellt werden. Ohne diese Materialgleichungen, auch *Konstitutivbeziehungen* genannt, sind die Bilanzgleichungen aus dem Kapitel 3.3 nicht oder nur in Spezialfällen lösbar.

3.4.1 Elastisches Materialverhalten

In Ibrahimbegovic (2009) wird zwischen *hypoelastischem* und *hyperelastischem* Materialverhalten im Rahmen der Elastizität unterschieden. Ersteres wird auch *Cauchy's* Elastizität genannt und soll im Rahmen dieser Arbeit nicht weiter betrachtet werden. Das *hyperelastische* Materialverhalten wird auch *Green's* Elastizität genannt und beruht auf einem Potential ψ . Das Potential ergibt sich formal aus einer Integration der mechanischen Arbeit (Formänderungsarbeit), die die Spannungen an den Verzerrungen vollbringen, sodass sich nach Kreißig und Benedix (2013) die Spannungen als partielle Ableitungen nach den Verzerrungen ergeben

$$\boldsymbol{\sigma} = \frac{\partial \psi}{\partial \boldsymbol{\varepsilon}}. \quad (3.30)$$

Das Potenzial ψ wird auch *Verzerrungsenergiedichte* genannt. Im einfachsten Fall, der *linearen* Elastizität, beschrieben durch das *Hooke'sche* Gesetz, ist das Potenzial ψ durch eine quadratische Form gegeben (Ibrahimbegovic, 2009)

$$\begin{aligned} \psi &= \frac{1}{2} \boldsymbol{\varepsilon} \cdot \cdot \boldsymbol{\mathcal{C}} \cdot \cdot \boldsymbol{\varepsilon} \\ \psi &= \frac{1}{2} \varepsilon_{ij} \mathcal{C}_{ijkl} \varepsilon_{kl}. \end{aligned} \quad (3.31)$$

Dabei ist $\boldsymbol{\mathcal{C}}$ der Elastizitätstensor vierter Stufe, der sich zur Basis $\{\mathbf{e}_i\}_{i=1,2,3}$ auch in Basisdarstellung angeben lässt

$$\boldsymbol{\mathcal{C}} = \mathcal{C}_{ijkl} \mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l. \quad (3.32)$$

Der Elastizitätstensor ist durch Hauptsymmetrien $\mathcal{C}_{ijkl} = \mathcal{C}_{klij}$ und Nebensymmetrien $\mathcal{C}_{ijkl} = \mathcal{C}_{jikl} = \mathcal{C}_{ijlk}$ gekennzeichnet. Im Fall der isotropen linearen Elastizität kann der Elastizitätstensor allein durch zwei materialspezifische Größen beschrieben werden

$$\begin{aligned} \boldsymbol{\mathcal{C}} &= \lambda \mathbf{1} \otimes \mathbf{1} + 2\mu \mathbf{I} \\ \mathbf{1} \otimes \mathbf{1} &= \delta_{ij} \delta_{kl} \mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l \\ \mathbf{I} &= \frac{1}{2} (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}) \mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l. \end{aligned} \quad (3.33)$$

Der Identitätstensor zweiter Stufe ist mit $\mathbf{1} = \delta_{ij} \mathbf{e}_i \otimes \mathbf{e}_j$ gegeben, wohingegen \mathbf{I} in (3.33) einen Tensor vierter Stufe darstellt, der in de Souza Neto et al. (2008) als *symmetrische Projektion* oder *symmetrische Identität* bezeichnet wird. Die sogenannten *Lamé* Konstanten λ und μ können auch im gängigeren Elastizitätsmodul E und der Querkontraktionszahl ν ausgedrückt werden,

$$\begin{aligned} \lambda &= \frac{\nu E}{(1 + \nu)(1 - 2\nu)}, \\ \mu &= \frac{E}{2(1 + \nu)}. \end{aligned} \quad (3.34)$$

Ausgehend von Gleichung (3.30) und unter Berücksichtigung von (3.31), kann jetzt der lineare Zusammenhang zwischen der Spannung $\boldsymbol{\sigma}$ und der Verzerrung $\boldsymbol{\varepsilon}$ folgendermaßen angegeben werden,

$$\begin{aligned}\boldsymbol{\sigma} &= \mathbf{C} \cdot \boldsymbol{\varepsilon} \\ \boldsymbol{\sigma} &= C_{ijkl} \varepsilon_{mn} \mathbf{e}_i \otimes \mathbf{e}_j \otimes \underbrace{\mathbf{e}_k \otimes \mathbf{e}_l \cdot \mathbf{e}_m \otimes \mathbf{e}_n}_{\delta_{lm} \delta_{kn}} = C_{ijkl} \varepsilon_{lk} \mathbf{e}_i \otimes \mathbf{e}_j.\end{aligned}\quad (3.35)$$

Im Falle der linearen Elastizität, insbesondere der isotropen linearen Elastizität lassen sich die Koordinaten des Elastizitätstensors in einer 6×6 -Matrix zusammenfassen

$$\mathbf{C} = \frac{\nu E}{(1 + \nu)(1 - 2\nu)} \begin{bmatrix} (1 - \nu) & \nu & \nu & 0 & 0 & 0 \\ \nu & (1 - \nu) & \nu & 0 & 0 & 0 \\ \nu & \nu & (1 - \nu) & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(1 - 2\nu)}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1 - 2\nu)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(1 - 2\nu)}{2} \end{bmatrix}. \quad (3.36)$$

Die Darstellung (3.36) ist speziell geeignet im Zusammenhang mit der *Voigt*-Notation. Damit lassen sich die Spannungen einfach als Matrizenmultiplikation aus dem Elastizitätstensor und den Verzerrungen berechnen,

$$\boldsymbol{\sigma} = \mathbf{C} \boldsymbol{\varepsilon}, \quad (3.37)$$

wobei hier $\boldsymbol{\sigma}$ und $\boldsymbol{\varepsilon}$ in den *Voigt* Notationen (3.22) bzw. (3.16) anzusetzen sind. Bei einem rein elastischen Materialverhalten ist der Spannungszustand allein vom vorliegenden Verzerrungszustand abhängig; der deformierte Körper kehrt nach Rücknahme der Belastung in seinen ursprünglichen Zustand zurück (Paris, 2003).

3.4.2 Elasto-Plastisches Materialverhalten

Die Eigenschaft eines Materials, nach Entlastung wieder in seinen ursprünglichen Zustand zurückzukehren, ist bei vielen Werkstoffen nur bis zu einer gewissen Lastgrenze gegeben. Wird diese Lastgrenze überschritten, so geht bei Rücknahme der Belastung nur ein Teil der Verformung zurück, ein anderer Teil bleibt auch nach der Entlastung im Bauteil bestehen. Das sogenannte (plastische) *Fließen* des Materials setzt in den hoch belasteten Regionen des Bauteils ein. Der Deformationsprozess ruft demnach nicht nur *elastische* Verzerrungen $\boldsymbol{\varepsilon}^e$ im Körper hervor, sondern nach Überschreiten der elastischen Lastgrenze auch *plastische* Verzerrungen $\boldsymbol{\varepsilon}^p$. Im Rahmen der hier betrachteten kleinen Deformationen können diese beiden Anteile additiv zur Gesamtdehnung $\boldsymbol{\varepsilon}$ zusammengefasst werden

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p. \quad (3.38)$$

In der Regel versagt ein Bauteil nicht schlagartig nach Überschreiten der elastischen Lastgrenze, sondern ist in der Lage, weiter Last aufzunehmen. In einem Last-Verformungs-Diagramm stellt sich dies nach Überschreiten der elastischen Grenze als ein weiter steigender, wenn auch meist nicht mehr linearer Verlauf dar. Zum einen kann hierfür eine sogenannte *Spannungsumlagerung* im Bauteil verantwortlich sein, zum anderen auch eine sogenannte *Verfestigung* des Materials selbst. In der Praxis treten meist beide Effekte gemeinsam auf und bedingen sich gegenseitig. Ein Material ohne Verfestigung wird als *ideal plastisch* bezeichnet und ist in der Praxis eher

selten vorzufinden. Da plastische Deformationen volumentreu sind, bietet es sich weiter an, den Verzerrungstensor folgendermaßen aufzuteilen,

$$\begin{aligned}\boldsymbol{\varepsilon} &= \frac{1}{3}e\mathbf{1} + \tilde{\boldsymbol{\varepsilon}} \\ e &= \text{tr}(\boldsymbol{\varepsilon}) \\ \mathbf{1} &= \delta_{ij} \mathbf{e}_i \otimes \mathbf{e}_j \\ \tilde{\boldsymbol{\varepsilon}} &= \boldsymbol{\varepsilon} - \frac{1}{3}e\mathbf{1},\end{aligned}\tag{3.39}$$

wobei e die *Volumendilatation* für den hydrostatischen Anteil darstellt und $\tilde{\boldsymbol{\varepsilon}}$ der *Deviator* des Verzerrungstensors ist. Die Spurbildung ist mit $\text{tr}(\cdot)$ gekennzeichnet. Eine ähnliche Aufteilung kann für den Spannungstensor durchgeführt werden. Die Begründung hierbei liegt darin, dass plastisches Fließen nicht vom hydrostatischen Druckzustand abhängig ist. Somit kann, analog zum Verzerrungstensor, eine hydrostatisch-deviatorische Aufteilung des Spannungstensors durchgeführt werden,

$$\begin{aligned}\boldsymbol{\sigma} &= -p\mathbf{1} + \tilde{\boldsymbol{\sigma}} \\ p &= -\frac{1}{3}\text{tr}(\boldsymbol{\sigma}) \\ \mathbf{1} &= \delta_{ij} \mathbf{e}_i \otimes \mathbf{e}_j \\ \tilde{\boldsymbol{\sigma}} &= \boldsymbol{\sigma} + p\mathbf{1},\end{aligned}\tag{3.40}$$

mit dem *hydrostatischen Druck* p und dem *Spannungsdeviator* $\tilde{\boldsymbol{\sigma}}$.

Die Verfestigungsmechanismen im Material werden in *isotrope* und *kinematische* Verfestigung eingeteilt. Für die kontinuumsmechanische Beschreibung der Verfestigungsmechanismen ist das Konzept der *inneren Variablen* geeignet, die für die Formulierung *inelastischen* Materialverhaltens eingeführt werden. Für den *isotropen* Anteil der Verfestigung wird die skalare Variable a eingeführt und für den *kinematischen* Anteil die tensorielle Variable $\boldsymbol{\beta}$. Während im elastischen Fall die mechanische Arbeit vollständig in Form der Verzerrungsenergiedichte als Potenzial gespeichert wird, muss im elasto-plastischen Fall von dieser Theorie abgewichen werden. Hier geht ein Teil der im Belastungsprozess vollbrachten mechanischen Arbeit in Form von *Dissipation* \mathcal{D}^{in} , also Verlustarbeit durch das plastische Fließen verloren. Mithilfe der internen Variablen ist es nun möglich, die Arbeitsanteile in gespeicherte Verzerrungsenergiedichte ψ und Dissipation \mathcal{D}^{in} aufzuteilen

$$\psi := \psi(\boldsymbol{\varepsilon}^e, a, \boldsymbol{\beta})\tag{3.41}$$

$$\mathcal{D}^{in} = \boldsymbol{\sigma} \cdot \dot{\tilde{\boldsymbol{\varepsilon}}}^p - \frac{\partial \psi}{\partial a} \dot{a} - \frac{\partial \psi}{\partial \boldsymbol{\beta}} \cdot \dot{\boldsymbol{\beta}}.\tag{3.42}$$

Durch die Erweiterung der Argumente in der Verzerrungsenergiedichte (3.41), mit den internen Variablen a und $\boldsymbol{\beta}$, soll diese ab jetzt *freie Energiedichte-Funktion* ψ genannt werden. In Gleichung (3.42) beschreibt $(\dot{\cdot}) := \frac{d(\cdot)}{dt}$ eine *Rate* der Größe, wobei t hier als eine *Prozesszeit* gesehen werden kann, da alle Betrachtungen im quasi statischen Fall formuliert werden. Demnach ist $\dot{\tilde{\boldsymbol{\varepsilon}}}^p$ die Rate des deviatorischen Anteils des Verzerrungstensors. Durch die Verfestigung wird das Maß der durch $\dot{\tilde{\boldsymbol{\varepsilon}}}^p$ in Form von Dissipation verloren gegangenen mechanischen Arbeit an den Spannungen $\boldsymbol{\sigma}$ reduziert, was durch die beiden zusätzlichen Subtrahenden in Gleichung (3.42) ausgedrückt wird. Dabei sind \dot{a} und $\dot{\boldsymbol{\beta}}$ die Raten der internen Variablen und

$$\boldsymbol{\alpha} = -\frac{\partial \psi}{\partial a}\tag{3.43}$$

$$\boldsymbol{\tau} = -\frac{\partial\psi}{\partial\boldsymbol{\beta}} \quad (3.44)$$

die zu den *thermodynamischen Flüssen konjugierten thermodynamischen Kräfte*. Dabei kann $\boldsymbol{\tau}$ als eine *generalisierte Spannung* aufgefasst werden, die auch als *back stress* bezeichnet wird. Für die Formulierung eines elasto-plastischen *Fließgesetzes* ist es zuallererst nötig, eine Grenze zu definieren, ab der ein plastisches Fließen einsetzt. Im Rahmen dieser Arbeit wird die *von Mises'sche Fließbedingung* für isotrope und kinematische Verfestigung angesetzt,

$$f = \|\tilde{\boldsymbol{\sigma}} - \boldsymbol{\tau}\| - \sqrt{\frac{2}{3}}(\sigma^y + \alpha) \leq 0, \quad (3.45)$$

hierbei ist σ^y die anfängliche Fließgrenze des Materials. Die Betonung „anfällig“ ist nötig, da das Material nach einmaliger Verfestigung einen Anstieg in der Fließgrenze für eine erneute Belastung erfährt. Für eine zyklische Belastung ist es wichtig, den *Bauschinger-Effekt* zu erfassen, was durch die kinematische Verfestigungsbeschreibung gewährleistet ist. Die zur *von Mises'schen Fließbedingung* gehörende *Fließfläche* kann im Hauptspannungsraum³ als Zylinder mit der hydrostatischen Achse als Rotationsachse dargestellt werden.

Die elasto-plastische Materialformulierung erfolgt in *Ratenform* $(\dot{\cdot}) := \frac{d(\cdot)}{dt}$, sodass im Folgenden für die eingeführten Größen $\dot{\boldsymbol{\varepsilon}}^p$, $\dot{\boldsymbol{\beta}}$, $\dot{\boldsymbol{\alpha}}$, $\dot{\boldsymbol{\tau}}$ und $\dot{\boldsymbol{\sigma}}$ Zusammenhänge formuliert werden müssen. Hierfür wird ein *Optimierungsproblem* für die Dissipation $\mathcal{D}^{in} \rightarrow \max$, unter Berücksichtigung einer Nebenbedingung, bestehend aus der Fließfunktion (3.45) $f(\tilde{\boldsymbol{\sigma}}, \boldsymbol{\tau}, \alpha)$ formuliert. Die Nebenbedingung wird über einen *Lagrange-Multiplikator* $\dot{\lambda}$ eingebunden, sodass Folgendes gilt

$$\mathcal{L} = -\mathcal{D}^{in} + \dot{\lambda}f(\tilde{\boldsymbol{\sigma}}, \boldsymbol{\tau}, \alpha) \longrightarrow \text{stationär}. \quad (3.46)$$

Die Gleichung (3.46) stellt ein sogenanntes *Lagrangesches Funktional* dar, zu dessen Erfüllung noch die sogenannten *Kuhn-Tucker Bedingungen* der Plastizität einzuführen sind,

$$\dot{\lambda} \geq 0, \quad f \leq 0, \quad \dot{\lambda}f = 0. \quad (3.47)$$

Die Bedingungen (3.47) können in Hinblick auf die Fließfläche interpretiert werden und besagen, dass eine elastische Zustandsänderung innerhalb der Fließfläche und eine plastische Zustandsänderung auf der Fließfläche stattfindet (Parisch, 2003). Zusätzlich zu den *Kuhn-Tucker Bedingungen* muss berücksichtigt werden, dass plastisches Fließen nur eintritt, wenn sowohl $f = 0$ als auch $\dot{f} = 0$ gegeben ist. Letzteres bedeutet, dass die Fließfunktion zumindest eine zeitlang gleich null sein muss und nicht nur die Fließfläche „kurz“ berührt, was auch als *Praeger'sche Konsistenzbedingung* bezeichnet wird (de Borst et al., 2014). Tritt plastisches Fließen ein, so ist der *Lagrange-Multiplikator* $\dot{\lambda}$ ein Maß für die Größe des plastischen Verzerrungsinkrements und der Gradient $\frac{\partial f}{\partial \tilde{\boldsymbol{\sigma}}}$ zur Fließfläche

$$\begin{aligned} \frac{\partial f}{\partial \tilde{\boldsymbol{\sigma}}} &= \frac{\tilde{\boldsymbol{\sigma}} - \boldsymbol{\tau}}{\|\tilde{\boldsymbol{\sigma}} - \boldsymbol{\tau}\|} = \mathbf{n} \\ \frac{\partial f}{\partial \boldsymbol{\tau}} &= -\mathbf{n} \end{aligned} \quad (3.48)$$

gibt die Richtung $\mathbf{n} = n_{ij} \mathbf{e}_i \otimes \mathbf{e}_j$ des plastischen Fließens an (Wriggers, 2013). Die Ermittlung der Fließrichtung normal zur Fließfläche aus Gleichung (3.48) zusammen mit der Verwendung einer freien Energiedichte-Funktion wird als *assoziierte Fließregel* bezeichnet. Die Lösung des Optimierungsproblems (3.46) mit den genannten zusätzlichen Bedingungen und

³Siehe hierzu Gleichung (3.23) in Kapitel 3.2.

der Beschränkung auf *lineare* isotrope und kinematische Verfestigung führt letztendlich auf die folgenden *Evolutionsgleichungen*,

$$\dot{\tilde{\boldsymbol{\varepsilon}}}^p = \dot{\lambda} \mathbf{n} \quad (3.49)$$

$$\dot{a} = \dot{\lambda} \frac{\partial f}{\partial \alpha} = -\dot{\lambda} \sqrt{\frac{2}{3}} \quad (3.50)$$

$$\dot{\boldsymbol{\beta}} = \dot{\lambda} \frac{\partial f}{\partial \boldsymbol{\tau}} = -\dot{\lambda} \mathbf{n} \quad (3.51)$$

$$\dot{\alpha} = -A\dot{a} = A\dot{\lambda} \sqrt{\frac{2}{3}} \quad (3.52)$$

$$\dot{\boldsymbol{\tau}} = -\frac{2}{3}B\dot{\boldsymbol{\beta}} = \frac{2}{3}\dot{\lambda} \mathbf{n} \quad (3.53)$$

$$\dot{j} = \mathbf{n} \cdot \cdot \left(\dot{\tilde{\boldsymbol{\varepsilon}}} - \dot{\tilde{\boldsymbol{\varepsilon}}}^p \right) - \frac{2}{3}A\dot{\lambda} - \frac{2}{3}B\dot{\lambda} \mathbf{n} \cdot \cdot \mathbf{n} = 0 \quad (3.54)$$

$$\dot{\lambda} = \frac{\mathbf{n} \cdot \cdot \dot{\tilde{\boldsymbol{\varepsilon}}}}{\left(1 + \frac{A+B}{3\mu} \right)} \quad (3.55)$$

$$\dot{\boldsymbol{\sigma}} = \tilde{\mathbf{C}}^{ep} \cdot \cdot \dot{\tilde{\boldsymbol{\varepsilon}}} \quad (3.56)$$

$$\tilde{\mathbf{C}}^{ep} = 2\mu \left(\mathbf{I} - \frac{\mathbf{n} \otimes \mathbf{n}}{1 + \frac{A+B}{3\mu}} \right), \quad (3.57)$$

wobei $\tilde{\mathbf{C}}^{ep}$ den elasto-plastischen *Tangentenmodul* (Tensor vierter Stufe) darstellt. Die Größen A und B sind Parameter für die isotrope und kinematische Verfestigung und \mathbf{I} ist der schon in Gleichung (3.33) eingeführte symmetrische Identitätstensor vierter Stufe. Die Größe $\mu = \frac{E}{2(1+\nu)}$ ist eine der beiden *Lamé* Konstanten, wobei an dieser Stelle angemerkt sei, dass der *Lagrange-Multiplikator* $\dot{\lambda}$ nicht zu verwechseln ist mit der *Lamé* Konstante $\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}$. In Abschnitt 3.4.3 soll nun die *Integration* der Evolutionsgleichungen vorgestellt werden.

3.4.3 Algorithmische Aspekte der Elasto-Plastizität

In diesem Abschnitt sollen die vorgestellten *Raten* Formulierungen des elasto-plastischen Materialverhaltens aus Kapitel 3.4.2 in einen numerischen Rahmen eingebettet werden. In einem inkrementellen oder inkrementell-iterativen Lösungsalgorithmus auf *globaler* Ebene hat sich für die *Materialroutine*⁴ ein sehr effizienter Algorithmus etabliert, das sogenannte *radial-return-mapping*, was einen *Prädiktor-Korrektor-Algorithmus* darstellt (de Souza Neto et al., 2008; Parisch, 2003). Im Folgenden soll $\Delta t = {}^{n+1}t - {}^n t$ die Schrittweite darstellen und ${}^n(\cdot)$ bzw. ${}^{n+1}(\cdot)$ die betrachteten Größen zum *Prozesszeitpunkt* t bzw. $t + \Delta t$. Aus dem globalen Lösungsalgorithmus wird ein neuer Verzerrungszustand ${}^{n+1}\boldsymbol{\varepsilon}$ ermittelt⁵. Die Verzerrung ${}^{n+1}\boldsymbol{\varepsilon}$ muss nun innerhalb der Materialroutine in einen elastischen und plastischen Anteil aufgeteilt werden. Zudem muss eine resultierende Spannung und eine Materialtangente an den globalen Lösungsalgorithmus zurückgegeben werden. Für die wiederkehrenden Folgeschritte müssen

⁴In der FEM erfolgt die Materialroutine auf Integrationspunktebene, siehe Kapitel 4.

⁵In der FEM resultiert der Verzerrungszustand ${}^{n+1}\boldsymbol{\varepsilon}$ an den Integrationspunkten aus den Knotenverschiebungen des Elementes, siehe Kapitel 4.

innerhalb der Materialroutine auch immer die plastischen Variablen und die Fließfunktion upgedatet werden. Im Sinne des *radial-return-mappings* werden im Prädikatorschritt zuerst *Test-Größen* wie folgt ermittelt

$$\begin{aligned} {}^{n+1}e &= \text{tr}({}^{n+1}\boldsymbol{\varepsilon}) \\ {}^{n+1}\tilde{\boldsymbol{\varepsilon}} &= {}^{n+1}\boldsymbol{\varepsilon} - \frac{1}{3} {}^{n+1}e \mathbf{1} \end{aligned} \quad (3.58)$$

$${}^{n+1}\tilde{\boldsymbol{\sigma}}^{\text{test}} = 2\mu ({}^{n+1}\tilde{\boldsymbol{\varepsilon}} - {}^n\tilde{\boldsymbol{\varepsilon}}^p) \quad (3.59)$$

$${}^{n+1}\tilde{\boldsymbol{\sigma}}^{\text{test}} = {}^{n+1}\tilde{\boldsymbol{\sigma}}^{\text{test}} - {}^n\boldsymbol{\tau} \quad (3.60)$$

$${}^{n+1}\alpha^{\text{test}} = {}^n\alpha \quad (3.61)$$

$${}^{n+1}f^{\text{test}} = \|{}^{n+1}\tilde{\boldsymbol{\sigma}}^{\text{test}}\| - \sqrt{\frac{2}{3}} (\sigma^y + {}^{n+1}\alpha^{\text{test}}) . \quad (3.62)$$

Zunächst muss die Test-Fließfunktion (3.62) ausgewertet werden. Für den Fall ${}^{n+1}f^{\text{test}} \leq 0$ liegt ein rein elastischer Schritt vor, sodass die Größen einfach als

$${}^{n+1}\tilde{\boldsymbol{\varepsilon}}^p = {}^n\tilde{\boldsymbol{\varepsilon}}^p \quad (3.63)$$

$${}^{n+1}\tilde{\boldsymbol{\sigma}} = {}^{n+1}\tilde{\boldsymbol{\sigma}}^{\text{test}} \quad (3.64)$$

$${}^{n+1}\boldsymbol{\tau} = {}^n\boldsymbol{\tau} \quad (3.65)$$

$${}^{n+1}\alpha = {}^n\alpha \quad (3.66)$$

$${}^{n+1}\tilde{\boldsymbol{\mathcal{C}}}^{ep} = {}^n\tilde{\boldsymbol{\mathcal{C}}}^{ep} , \quad (3.67)$$

angegeben werden können. In dem Fall ${}^{n+1}f^{\text{test}} > 0$ liegt ein plastischer Schritt vor, sodass zusätzlich das Inkrement des *Lagrange-Multiplikators* $\Delta\lambda = {}^{n+1}\lambda - {}^n\lambda$ und die Richtung des plastischen Fließens ${}^{n+1}\mathbf{n}$ berechnet werden müssen. Die Größen lassen sich wie folgt berechnen

$${}^{n+1}\mathbf{n} = \frac{{}^{n+1}\tilde{\boldsymbol{\sigma}}^{\text{test}}}{\|{}^{n+1}\tilde{\boldsymbol{\sigma}}^{\text{test}}\|} \quad (3.68)$$

$$\Delta\lambda = \frac{{}^{n+1}f^{\text{test}}}{2\mu \left(1 + \frac{A+B}{3\mu}\right)} \quad (3.69)$$

$${}^{n+1}\tilde{\boldsymbol{\varepsilon}}^p = \Delta\lambda {}^{n+1}\mathbf{n} \quad (3.70)$$

$${}^{n+1}\tilde{\boldsymbol{\sigma}} = {}^{n+1}\tilde{\boldsymbol{\sigma}}^{\text{test}} - 2\mu \Delta\lambda {}^{n+1}\mathbf{n} \quad (3.71)$$

$${}^{n+1}\boldsymbol{\tau} = {}^n\boldsymbol{\tau} + \frac{2}{3} \Delta\lambda B {}^{n+1}\mathbf{n} \quad (3.72)$$

$${}^{n+1}\alpha = {}^n\alpha + \sqrt{\frac{2}{3}} \Delta\lambda A \quad (3.73)$$

$${}^{n+1}\tilde{\mathbf{s}} = {}^{n+1}\tilde{\mathbf{s}} - \Delta\lambda \left(2\mu + \frac{2}{3}B \right) {}^{n+1}\mathbf{n} \quad (3.74)$$

$${}^{n+1}f = {}^{n+1}f^{\text{test}} - \left(2\mu + \frac{2}{3}(A+B) \right) \Delta\lambda \quad (3.75)$$

$${}^{n+1}\tilde{\mathbf{C}}^{ep} = 2\mu \left(\left(1 - \frac{2\mu\Delta\lambda}{\|{}^{n+1}\tilde{\mathbf{s}}^{\text{test}}\|} \right) \mathbf{I} - \left(\frac{1}{1 + \frac{A+B}{3\mu}} - \frac{2\mu\Delta\lambda}{\|{}^{n+1}\tilde{\mathbf{s}}^{\text{test}}\|} \right) {}^{n+1}\mathbf{n} \otimes {}^{n+1}\mathbf{n} \right). \quad (3.76)$$

Die Gleichung (3.71) zur Bestimmung der *zulässigen* Spannung ${}^{n+1}\tilde{\boldsymbol{\sigma}}$ ist in Abbildung 3.3 veranschaulicht. Bei der Darstellung in 3.3 handelt es sich um die sogenannte π -Ebene, bei der im Hauptspannungsraum die Ebene durch den Ursprung senkrecht zur Raumdiagonalen $\boldsymbol{\sigma}_I = \boldsymbol{\sigma}_{II} = \boldsymbol{\sigma}_{III}$ als Projektionsebene genutzt wird. Die Achsen werden in dieser zweidimensionalen Ansicht mit $\tilde{\boldsymbol{\sigma}}_I$, $\tilde{\boldsymbol{\sigma}}_{II}$ und $\tilde{\boldsymbol{\sigma}}_{III}$ gekennzeichnet, da nur die ausschlaggebenden deviatorischen Anteile dargestellt werden. Ausgehend von einem in der Fließfläche $f(\tilde{\boldsymbol{\sigma}}, \boldsymbol{\tau}, \alpha)$

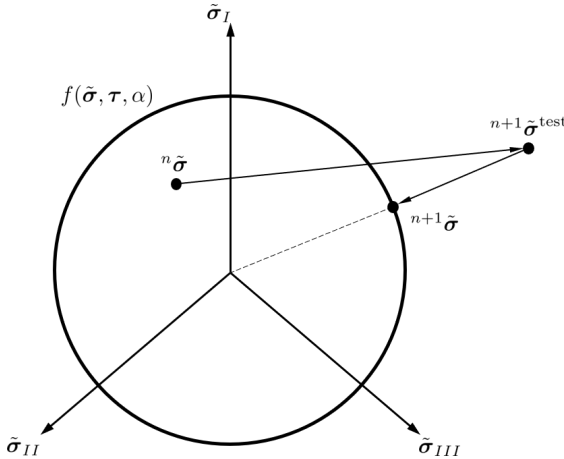


Abbildung 3.3: Schema des radial-return-mapping in der π -Ebene für den Fließflächen-Zylinder nach von Mises

liegenden Spannungszustandes ${}^n\tilde{\boldsymbol{\sigma}}$ wird mit den ermittelten deviatorischen Verzerrungen ${}^{n+1}\tilde{\boldsymbol{\epsilon}}$ die deviatorische Test-Spannung ${}^{n+1}\tilde{\boldsymbol{\sigma}}^{\text{test}}$ berechnet. Im Sinne des *radial-return-mappings* muss dieser außerhalb der Fließfläche liegende und damit nicht zulässige Spannungszustand zurück auf die Fließfläche gebracht werden. Die Richtung ${}^{n+1}\mathbf{n}$ des Fließens ist nach Gleichung (3.68) gerade die normierte deviatorische Testspannung und somit *radial* gerichtet zur hydrostatischen Achse des *von Mises'schen* Fließflächen-Zylinders im Hauptspannungsraum.

Abschließend muss von der Materialroutine die Spannung ${}^{n+1}\boldsymbol{\sigma}$ und die *vollständig, algorithmisch konsistente Materialtangente* ${}^{n+1}\mathbf{C}^{ep}$ an den *globalen* Lösungsalgorithmus zurückgegeben werden⁶. Die Spannung ${}^{n+1}\boldsymbol{\sigma}$ muss um den hydrostatischen Anteil erweitert werden und ergibt

⁶Die Größen werden auf globaler Ebene für die Berechnung der internen Knotenkräfte und der tangentialen Steifigkeitsmatrix benötigt, siehe Kapitel 4 zur FEM.

sich zu

$$\begin{aligned}
 {}^{n+1}\boldsymbol{\sigma} &= \kappa {}^{n+1}\mathbf{e}\mathbf{1} + {}^{n+1}\tilde{\boldsymbol{\sigma}} \\
 {}^{n+1}\mathbf{e} &= \text{tr}({}^{n+1}\boldsymbol{\varepsilon}) \\
 \kappa &= \frac{3\lambda + 2\mu}{3},
 \end{aligned} \tag{3.77}$$

wobei κ der Kompressionsmodul ist und λ und μ wieder die *Lamé* Konstanten repräsentieren. Für die vollständige algorithmisch konsistente Materialtangente muss das inkrementelle Stoffgesetz $\Delta\boldsymbol{\sigma} = \mathbf{C}^{ep} \cdot \Delta\boldsymbol{\varepsilon}$ berücksichtigt werden, sodass die „deviatorische“ Materialtangente ${}^{n+1}\tilde{\mathbf{C}}^{ep}$ aus Gleichung (3.76) zu erweitern ist,

$$\begin{aligned}
 {}^{n+1}\mathbf{C}^{ep} &= {}^{n+1}\mathbf{C}_{vol}^e + {}^{n+1}\tilde{\mathbf{C}}^{ep} \cdot \mathcal{I} \\
 \mathcal{I} &= \mathbf{I} - \frac{1}{3}\mathbf{1} \otimes \mathbf{1},
 \end{aligned} \tag{3.78}$$

mit dem *Projektionstensor*⁷ \mathcal{I} und dem volumetrischen Anteil der Materialtangente ${}^{n+1}\mathbf{C}_{vol}^e$. Die aus rein deviatorischen Anteilen berechnete Materialtangente ${}^{n+1}\tilde{\mathbf{C}}^{ep}$ muss wieder in den „Spannungs-Dehnungs-Raum“ zurückprojiziert werden und um den volumetrischen Anteil ergänzt werden. Nach Wriggers (2013) ist eine Darstellung in der Form

$$\begin{aligned}
 {}^{n+1}a &= 1 - \frac{2\mu\Delta\lambda}{\|{}^{n+1}\tilde{\mathbf{s}}^{\text{test}}\|} \\
 {}^{n+1}b &= \frac{1}{1 + \frac{A+B}{3\mu}} - \frac{2\mu\Delta\lambda}{\|{}^{n+1}\tilde{\mathbf{s}}^{\text{test}}\|} \\
 {}^{n+1}\mathbf{C}^{ep} &= \kappa\mathbf{I} \otimes \mathbf{1} + 2\mu({}^{n+1}a\mathcal{I} - {}^{n+1}b\mathbf{n} \otimes \mathbf{n})
 \end{aligned} \tag{3.79}$$

möglich. Auf die Verwendung der Spannungen ${}^{n+1}\boldsymbol{\sigma}$ und der vollständig algorithmisch konsistenten Materialtangente ${}^{n+1}\mathbf{C}^{ep}$ wird im Rahmen der FEM (Kapitel 4.5.1) an entsprechender Stelle hingewiesen.

⁷Dabei sind $\mathbf{I} = \frac{1}{2}(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})\mathbf{e}_i \otimes \mathbf{e}_j \otimes \mathbf{e}_k \otimes \mathbf{e}_l$ und $\mathbf{1} = \delta_{ij}\mathbf{e}_i \otimes \mathbf{e}_j$ die bereits vorgestellten Identitätstensoren.

4 Finite Elemente Methode für hoch aufgelöste Berechnungen

In diesem Kapitel wird die Finite Elemente Methode (FEM) aufgezeigt, die im Rahmen dieser Arbeit für die hoch aufgelösten Berechnungen der 3D-Elasto-Plastizität verwendet wird. Eine ausführliche Darstellung der FEM findet sich beispielsweise in den Büchern von Wriggers (2013), de Borst et al. (2014) und Gawehn und Funk (2020).

4.1 Schwache Formulierung der zugrunde liegenden Gleichungen

Die Aufgabenstellung, die mittels der FEM gelöst werden soll, kann im Hinblick auf die kontinuumsmechanischen Zusammenhänge aus Kapitel 3 formuliert werden. Nach Gawehn und Funk (2020) stehen für die unbekanntenen Größen

$$\begin{aligned}
 \text{Verschiebungsfeld} & \quad \mathbf{u} & (3 \text{ Unbekannte}) \\
 \text{Verzerrungstensor} & \quad \boldsymbol{\varepsilon} & (6 \text{ Unbekannte}) \\
 \text{Spannungstensor} & \quad \boldsymbol{\sigma} & (6 \text{ Unbekannte}),
 \end{aligned} \tag{4.1}$$

die folgenden Gleichungen für die Lösung zur Verfügung

$$\begin{aligned}
 \text{Kinematische Gleichungen} & \quad \boldsymbol{\varepsilon} = \frac{1}{2} (\nabla \mathbf{u} + \nabla \mathbf{u}^T) & (6 \text{ Gleichungen}) \\
 \text{Gleichgewichtsbedingungen} & \quad \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g} = \mathbf{0} & (3 \text{ Gleichungen}) \\
 \text{Materialgleichungen} & \quad \Delta \boldsymbol{\sigma} \approx \mathbf{C}(\boldsymbol{\varepsilon}) \cdot \Delta \boldsymbol{\varepsilon} & (6 \text{ Gleichungen}).
 \end{aligned} \tag{4.2}$$

Somit stehen den 15 Unbekannten aus (4.1) die 15 Gleichungen aus (4.2) gegenüber, was grundsätzlich eine Lösung des Problems möglich macht. Die Materialgleichungen sind in inkrementeller Näherungsform gegeben, da im allgemeinen Fall der Materialtensor $\mathbf{C}(\boldsymbol{\varepsilon})$ vom Verzerrungstensor abhängig ist. Eine *Ein-Schritt-Lösung* ist lediglich bei linear elastischem Materialverhalten möglich. Vielmehr muss ein *iteratives* Lösungsverfahren im Rahmen der FEM formuliert werden. Die Gleichungen (4.2) müssen zudem durch Randbedingungen ergänzt werden, die für eine eindeutige Lösungsfindung notwendig sind

$$\begin{aligned}
 \text{Dirichlet Randbedingungen} & : \quad \mathbf{u} = \mathbf{u}_0 & \text{Rand } \partial_u \mathcal{B} \\
 \text{Neumann Randbedingungen} & : \quad \mathbf{t} = \mathbf{t}_0 & \text{Rand } \partial_t \mathcal{B}.
 \end{aligned} \tag{4.3}$$

Dabei beschreiben $\partial_u \mathcal{B}$ und $\partial_t \mathcal{B}$ den kompletten Rand des Körpers \mathcal{B} . Dies kann auch als Vereinigung $\partial \mathcal{B} = \partial_u \mathcal{B} \cup \partial_t \mathcal{B}$ geschrieben werden, bei der eine Überschneidung der Ränder ausgeschlossen ist: $\emptyset = \partial_u \mathcal{B} \cap \partial_t \mathcal{B}$. Die zu lösenden Gleichungen (4.2) für die gesuchten Größen (4.1) werden in Wriggers (2013) auch als statisches Randwertproblem der Kontinuumsmechanik bezeichnet, im Rahmen dieser Arbeit für den Fall kleiner Deformation und ohne Beschleunigungsterm. In der Mechanik sind nun zwei klassische Herangehensweisen für die Lösung des beschriebenen Randwertproblems üblich, die unter die Überschrift *Prinzipien der Mechanik* fallen. Zum einen das *Prinzip vom Minimum des elastischen Gesamtpotentials* und zum anderen das *Prinzip der virtuellen Verschiebungen* (Wriggers, 2013; Kreißig und Benedix, 2013; Stein, 1989; Ibrahimbegovic, 2009; Gawehn und Funk, 2020). Die Methoden führen beide auf die gleiche sogenannte *schwache* Form des Gleichgewichts und können somit als einander äquivalent

angesehen werden (Gawehn und Funk, 2020; Ibrahimbegovic, 2009).

Im Folgenden soll die *schwache* Form des Gleichgewichts mit dem *Prinzip der virtuellen Verschiebungen* hergeleitet werden. Im ersten Schritt wird hierzu die lokal geltende Gleichgewichtsbedingung (3.28), auch *starke* Form genannt, mit einer *Testfunktion* multipliziert und über das gesamte Gebiet integriert

$$\begin{aligned} \int_{\mathcal{B}} \delta \mathbf{u} \cdot (\nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g}) dV &= 0 \\ \int_{\mathcal{B}} (\nabla \cdot \boldsymbol{\sigma}) \cdot \delta \mathbf{u} dV + \int_{\mathcal{B}} \rho \mathbf{g} \cdot \delta \mathbf{u} dV &= 0. \end{aligned} \quad (4.4)$$

Die Testfunktion $\delta \mathbf{u}$ ist im Rahmen der Mechanik als die schon angedeutete *virtuelle* Verschiebung zu deuten. Die Forderung nach Erfüllung des Gleichgewichts erfolgt nicht mehr punktuell sondern im gewichteten, integralen Mittel. Für den ersten Term in Gleichung (4.4) wird im Rahmen der Tensorrechnung, beispielsweise in de Boer (1982), oder auch im Anhang zur Tensorrechnung in Wriggers (2013) folgende Rechenvorschrift¹ bereitgestellt

$$(\nabla \cdot \boldsymbol{\sigma}) \cdot \delta \mathbf{u} = \nabla \cdot (\boldsymbol{\sigma} \cdot \delta \mathbf{u}) - \boldsymbol{\sigma} \cdot \cdot \nabla \delta \mathbf{u}. \quad (4.5)$$

Der Term $\nabla \delta \mathbf{u} := \delta \mathbf{u} \otimes \nabla$ stellt einen Tensor zweiter Stufe dar und kann als zu $\delta \mathbf{u}$ zugehöriger *virtueller* Verzerrungszustand $\delta \boldsymbol{\varepsilon} := \nabla \delta \mathbf{u}$ im Körper interpretiert werden. Somit kann die *schwache* Form folgendermaßen umgeschrieben werden

$$\int_{\mathcal{B}} \nabla \cdot (\boldsymbol{\sigma} \cdot \delta \mathbf{u}) dV - \int_{\mathcal{B}} \boldsymbol{\sigma} \cdot \cdot \delta \boldsymbol{\varepsilon} dV + \int_{\mathcal{B}} \rho \mathbf{g} \cdot \delta \mathbf{u} dV = 0. \quad (4.6)$$

Der erste Term in Gleichung (4.6) kann jetzt mit dem *Gauß'schen* Integralsatz folgendermaßen ausgedrückt werden,

$$\int_{\mathcal{B}} \nabla \cdot (\boldsymbol{\sigma} \cdot \delta \mathbf{u}) dV = \int_{\partial \mathcal{B}} \overbrace{\boldsymbol{\sigma} \cdot \mathbf{n}}^{\mathbf{t}} \cdot \delta \mathbf{u} dA, \quad (4.7)$$

mit dem auf dem Rand $\partial \mathcal{B}$ des Körpers nach außen zeigenden Flächen Normalenvektor \mathbf{n} und dem Spannungsvektor \mathbf{t} . Mit dem Ergebnis aus dem *Gauß'schen* Integralsatz kann die schwache Form (4.6) in

$$\underbrace{\int_{\mathcal{B}} \boldsymbol{\sigma} \cdot \cdot \delta \boldsymbol{\varepsilon} dV}_{\delta W} = \underbrace{\int_{\partial \mathcal{B}} \mathbf{t} \cdot \delta \mathbf{u} dA + \int_{\mathcal{B}} \rho \mathbf{g} \cdot \delta \mathbf{u} dV}_{\delta A} \quad (4.8)$$

umgeschrieben werden, mit der virtuellen inneren Arbeit δW und der virtuellen äußeren Arbeit δA . Nach dem Prinzip der virtuellen Verschiebungen kann Gleichung (4.8) folgendermaßen interpretiert werden:

Prinzip der Virtuellen Verschiebungen

Befindet sich ein belasteter Körper unter einem Verschiebungsfeld \mathbf{u} im Gleichgewicht, so ist die virtuelle innere Arbeit δW , die die virtuellen Verzerrungen $\delta \boldsymbol{\varepsilon}$ an den wahren Spannungen $\boldsymbol{\sigma}$ verrichten, äquivalent zu der virtuellen äußeren Arbeit δA , die die virtuellen Verschiebungen $\delta \mathbf{u}$ an den äußeren Lasten \mathbf{t} und $\rho \mathbf{g}$ verrichten.

¹Diese Rechenregel aus der Tensorrechnung kann in Analogie zur Produktregel für das Ableiten zweier skalarer Funktionen gesehen werden.

4.2 Finite Elemente Ansatz

Für eine Finite Elemente Formulierung wird im Folgenden auf die *Voigt*-Notation umgestiegen. Damit nimmt die schwache Form (4.8) folgende Gestalt an,

$$\int_{\mathcal{B}} \delta \boldsymbol{\varepsilon}^T \boldsymbol{\sigma} dV = \int_{\partial \mathcal{B}} \delta \mathbf{u}^T \mathbf{t} dA + \rho \int_{\mathcal{B}} \delta \mathbf{u}^T \mathbf{g} dV, \quad (4.9)$$

mit den Größen in *Voigt*-Notation

$$\begin{aligned} \delta \boldsymbol{\varepsilon} &= [\delta \varepsilon_{11} \quad \delta \varepsilon_{22} \quad \delta \varepsilon_{33} \quad 2\delta \varepsilon_{12} \quad 2\delta \varepsilon_{23} \quad 2\delta \varepsilon_{13}]^T \\ \boldsymbol{\sigma} &= [\sigma_{11} \quad \sigma_{22} \quad \sigma_{33} \quad \sigma_{12} \quad \sigma_{23} \quad \sigma_{13}]^T \\ \delta \mathbf{u} &= [\delta u_1 \quad \delta u_2 \quad \delta u_3]^T \\ \mathbf{t} &= [t_1 \quad t_2 \quad t_3]^T \\ \mathbf{g} &= [g_1 \quad g_2 \quad g_3]^T. \end{aligned} \quad (4.10)$$

Zwischen den virtuellen Verzerrungen $\delta \boldsymbol{\varepsilon}$ und den virtuellen Verschiebungen $\delta \mathbf{u}$ wird folgender Zusammenhang hergestellt,

$$\delta \boldsymbol{\varepsilon} = \mathbf{D} \delta \mathbf{u}, \quad (4.11)$$

mit der *Differenzialoperator-Matrix* \mathbf{D} , die folgendermaßen angegeben werden kann

$$\mathbf{D} = \begin{bmatrix} \frac{\partial}{\partial x_1} & 0 & 0 \\ 0 & \frac{\partial}{\partial x_2} & 0 \\ 0 & 0 & \frac{\partial}{\partial x_3} \\ \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_1} & 0 \\ 0 & \frac{\partial}{\partial x_3} & \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_3} & 0 & \frac{\partial}{\partial x_1} \end{bmatrix}. \quad (4.12)$$

Mit Gleichung (4.11) kann die schwache Form (4.9) auch in folgender Form angegeben werden,

$$\int_{\mathcal{B}} (\mathbf{D} \delta \mathbf{u})^T \boldsymbol{\sigma} dV = \int_{\partial \mathcal{B}} \delta \mathbf{u}^T \mathbf{t} dA + \rho \int_{\mathcal{B}} \delta \mathbf{u}^T \mathbf{g} dV. \quad (4.13)$$

Ziel ist es, eine numerische Näherungslösung für die Gleichung (4.13) zu finden. Pionierarbeit wurde unter anderem von *Walter Ritz* mit seinem *Ritzschen Verfahren* geleistet (Kreißig und Benedix, 2013). Dabei wird ein Überlagerungsansatz für das kontinuierliche Verschiebungsfeld $\mathbf{u} = [u_1 \quad u_2 \quad u_3]^T$ gewählt, bei dem sogenannte *Ansatzfunktionen* mit Wichtungsfaktoren versehen werden. Eingesetzt in die schwache Form lassen sich so Bestimmungsgleichungen für die Wichtungsfaktoren ermitteln. Das Verfahren hat den Nachteil, dass es schwer ist, für komplizierte Geometrien kinematisch zulässige Ansatzfunktionen *global* zu formulieren (Kreißig und Benedix, 2013). Im Gegensatz hierzu wird in der Finite Elemente Methode (FEM) das Gebiet bzw. hier der Körper \mathcal{B} in eine *finite*, also endliche Anzahl Elemente bzw. Teilgebiete zerlegt. Nach Wriggers (2013) kann dies durch

$$\mathcal{B} \approx \mathcal{B}^h = \bigcup_{e=1}^{n_e} \Omega_e \quad (4.14)$$

formuliert werden, wobei \mathcal{B} durch eine Vereinigung \mathcal{B}^h von n_e finiten Elementen Ω_e angenähert wird. Ein finites Element Ω_e kann dabei geometrisch durch die Angabe von (Eck-)Knoten beschrieben werden. Der wesentliche Unterschied der FEM zum Ritz'schen Verfahren ist nun der, für die gesuchten Feldgrößen solche Funktionen zu definieren, die an einem Knoten den Wert 1 annehmen und an allen umliegenden Knoten in \mathcal{B}^h den Wert 0. Die Funktionen werden dann wie bei dem Ritz'schen Verfahren mit Wichtungsfaktoren versehen und überlagert. Die Funktionen müssen gewissen Stetigkeitsansprüchen genügen, in unserm Fall C^0 . Das bedeutet, dass lediglich die primäre Größe, bei uns das Verschiebungsfeld \mathbf{u} stetig abgebildet werden soll. Mit einem linearen Funktionsansatz wäre diese Stetigkeitsanforderung erfüllt und könnte im ein- oder zweidimensionalen Fall als typische *Hütchenfunktion* dargestellt werden. In Ibrahimbegovic (2009) wird gezeigt, dass gerade diese Eigenschaft der Ansatzfunktionen und die anschließende Überlagerung für das Gebiet \mathcal{B}^h ein entscheidender Vorteil der FEM ist. Es ist somit möglich, die schwache Form alternativ auf Elementebene zu formulieren und die resultierenden Gleichungen anschließend zu *assemblieren*. Dies ist genau die numerische Überlegenheit, die die Finite Elemente Methode so auszeichnet. Ein Ansatz für das Verschiebungsfeld in einem Element Ω_e kann wie folgt angegeben werden,

$$\mathbf{u}_e = \mathbf{H}\hat{\mathbf{u}}_e, \quad (4.15)$$

wobei \mathbf{H} die auf einem Element Ω_e formulierten Ansatzfunktionen sind und $\hat{\mathbf{u}}_e$ die zugehörigen Verschiebungen der Elementknoten. Für ein Elementtyp mit n Knoten kann die Matrix der Ansatzfunktionen \mathbf{H} folgendermaßen angegeben werden

$$\mathbf{H} = \begin{bmatrix} h^1(\mathbf{x}) & 0 & 0 & \cdots & h^n(\mathbf{x}) & 0 & 0 \\ 0 & h^1(\mathbf{x}) & 0 & \cdots & 0 & h^n(\mathbf{x}) & 0 \\ 0 & 0 & h^1(\mathbf{x}) & \cdots & 0 & 0 & h^n(\mathbf{x}) \end{bmatrix} \quad (4.16)$$

$$\hat{\mathbf{u}}_e = [\hat{u}_1^1 \quad \hat{u}_2^1 \quad \hat{u}_3^1 \quad \cdots \quad \hat{u}_1^n \quad \hat{u}_2^n \quad \hat{u}_3^n]^T. \quad (4.17)$$

Eine *Ansatzfunktion* $h^i(\mathbf{x})$ auf dem Element nimmt am Knoten i den Wert 1 und an allen restlichen Knoten den Wert 0 an. Somit kann beispielsweise durch einen trilinearen Funktionsansatz in den Elementkoordinaten $\mathbf{x}_e = [x_1 \ x_2 \ x_3]_e \in \Omega_e$ ein stetiges Verschiebungsfeld in Ω_e durch die Knotenverschiebungen $\hat{\mathbf{u}}_e$ dargestellt werden. Es ist zudem gesichert, dass das Verschiebungsfeld \mathbf{u} im gesamten Gebiet \mathcal{B}^h stetig ist, wobei die Gradienten nicht zwangsläufig stetig sein müssen. Für das virtuelle Verschiebungsfeld wird der gleiche Ansatz $\delta\mathbf{u}_e = \mathbf{H}\delta\hat{\mathbf{u}}_e$ gewählt, mit dem virtuellen Verschiebungsfeld $\delta\hat{\mathbf{u}}_e$ auf dem Element Ω_e . Hier ist \mathbf{H} wieder die auf einem Element Ω_e definierte Matrix der Ansatzfunktion (4.16) und $\delta\hat{\mathbf{u}}_e$ die zugehörigen virtuellen Knotenverschiebungen. Zusammen mit der Definition (4.11) kann für das virtuelle Verzerrungsfeld

$$\begin{aligned} \delta\boldsymbol{\varepsilon}_e &= \mathbf{D}\mathbf{H}\delta\hat{\mathbf{u}}_e \\ \delta\boldsymbol{\varepsilon}_e &= \mathbf{B}\delta\hat{\mathbf{u}}_e \end{aligned} \quad (4.18)$$

geschrieben werden. Die \mathbf{B} -Matrix ist hierbei eine $6 \times 3n$ -Matrix der Gradienten der Ansatzfunktionen, wobei n die Anzahl der Knoten pro Element ist. Für eine FE-Analyse sind alle Knoten und Elemente in einer *Vernetzung* \mathcal{B}^h durchnummeriert, benachbarte Elemente sind dabei durch gemeinsame Knoten miteinander verbunden. Die Zuordnung von Knoten zu Elementen wird *Konnektivität* (engl. *Connectivity*) genannt. Für den Assemblierungsprozess werden die *Inzidenzmatrizen* \mathbf{Z}_e für jedes Element eingeführt, die die Konnektivität widerspiegeln

(de Borst et al., 2014). Damit kann zwischen den globalen Freiheitsgraden $\hat{\mathbf{u}}$ bzw. $\delta\hat{\mathbf{u}}$ und den Elementfreiheitsgraden \mathbf{u}_e bzw. $\delta\mathbf{u}_e$ transformiert werden

$$\begin{aligned}\hat{\mathbf{u}}_e &= \mathbf{Z}_e \hat{\mathbf{u}} \\ \delta\hat{\mathbf{u}}_e &= \mathbf{Z}_e \delta\hat{\mathbf{u}} \\ \mathbf{u}_e &= \mathbf{H}\mathbf{Z}_e \hat{\mathbf{u}} \\ \delta\mathbf{u}_e &= \mathbf{H}\mathbf{Z}_e \delta\hat{\mathbf{u}}.\end{aligned}\quad (4.19)$$

Dabei sind $\hat{\mathbf{u}}$ bzw. $\delta\hat{\mathbf{u}}$ $N \times 1$ -Spaltenmatrizen mit den N globalen Knotenfreiheitsgraden des Modells. Die Inzidenzmatrizen \mathbf{Z}_e haben die Dimension $3n \times N$ und sind lediglich mit Nullen und Einsen besetzt (de Borst et al., 2014). Die schwache Form (4.13) kann jetzt wie folgt aufgeschrieben werden,

$$\delta\hat{\mathbf{u}}^T \left(\sum_{e=1}^{n_e} \mathbf{Z}_e^T \int_{\Omega_e} \mathbf{B}^T \boldsymbol{\sigma} dV^e - \sum_{e=1}^{n_e} \mathbf{Z}_e^T \int_{\partial\Omega_e} \mathbf{H}^T \mathbf{t} dA^e - \rho \sum_{e=1}^{n_e} \mathbf{Z}_e^T \int_{\Omega_e} \mathbf{H}^T \mathbf{g} dV^e \right) = 0. \quad (4.20)$$

Die Gleichung (4.20) kann nur erfüllt werden, wenn der Term in den Klammern verschwindet, da die virtuellen Knotenverschiebungen $\delta\hat{\mathbf{u}}$ beliebig sein können. Demzufolge ist das algebraische Gleichungssystem

$$\begin{aligned}\sum_{e=1}^{n_e} \mathbf{Z}_e^T \int_{\Omega_e} \mathbf{B}^T \boldsymbol{\sigma} dV^e &= \sum_{e=1}^{n_e} \mathbf{Z}_e^T \int_{\partial\Omega_e} \mathbf{H}^T \mathbf{t} dA^e + \rho \sum_{e=1}^{n_e} \mathbf{Z}_e^T \int_{\Omega_e} \mathbf{H}^T \mathbf{g} dV^e \\ \mathbf{f}_{int} &= \sum_{e=1}^{n_e} \mathbf{Z}_e^T \int_{\Omega_e} \mathbf{B}^T \boldsymbol{\sigma} dV^e \\ \mathbf{f}_{ext} &= \sum_{e=1}^{n_e} \mathbf{Z}_e^T \int_{\partial\Omega_e} \mathbf{H}^T \mathbf{t} dA^e + \rho \sum_{e=1}^{n_e} \mathbf{Z}_e^T \int_{\Omega_e} \mathbf{H}^T \mathbf{g} dV^e \\ \mathbf{f}_{int} &= \mathbf{f}_{ext}\end{aligned}\quad (4.21)$$

zu lösen, mit der Spaltenmatrix der internen Knotenkräfte \mathbf{f}_{int} und der Spaltenmatrix der externen Knotenkräfte \mathbf{f}_{ext} . Mechanisch kann die Gleichung (4.21) als das Auswerten des Kräftegleichgewichts an den Knotenpunkten interpretiert werden. Bevor das Gleichungssystem (4.21) gelöst werden kann, muss ein Verfahren für die Lösung der Integralausdrücke auf Elementebene bereitgestellt werden. Zudem entscheidet der Zusammenhang $\boldsymbol{\sigma}(\boldsymbol{\varepsilon}(\mathbf{u}))$ darüber, ob am Ende ein lineares oder nichtlineares Gleichungssystem gelöst werden muss.

4.3 Isoparametrisches Konzept

Für die Berechnung der Integralausdrücke in Gleichung (4.21) hat sich das sogenannte *isoparametrische* Konzept etabliert. Dabei handelt es sich um eine *Koordinatentransformation* vom Element Ω_e auf ein Referenzelement Ω_r . Isoparametrisch wird diese Transformation dadurch, dass für die Formulierung der Elementgeometrie und das Verschiebungsfeld der gleiche Ansatz in den Koordinaten $\boldsymbol{\xi} = [\xi_1 \xi_2 \xi_3]$ des Referenzelementes gewählt wird. Die Matrix der Ansatzfunktionen \mathbf{H} wird jetzt in den Referenzelement-Koordinaten $\boldsymbol{\xi}$ formuliert,

$$\mathbf{H} = \begin{bmatrix} h^1(\boldsymbol{\xi}) & 0 & 0 & \cdots & h^n(\boldsymbol{\xi}) & 0 & 0 \\ 0 & h^1(\boldsymbol{\xi}) & 0 & \cdots & 0 & h^n(\boldsymbol{\xi}) & 0 \\ 0 & 0 & h^1(\boldsymbol{\xi}) & \cdots & 0 & 0 & h^n(\boldsymbol{\xi}) \end{bmatrix}. \quad (4.22)$$

Mit der Matrix (4.22) werden die Elementgeometrie und das Verschiebungsfeld gleichermaßen in den Koordinaten ξ formuliert,

$$\mathbf{x}_e = \mathbf{H}\hat{\mathbf{x}}_e \quad (4.23)$$

$$\mathbf{u}_e = \mathbf{H}\hat{\mathbf{u}}_e, \quad (4.24)$$

wobei $\hat{\mathbf{u}}_e$ wieder die Knotenverschiebungen (4.17) des Elements sind und $\hat{\mathbf{x}}_e$ die Knotenkoordinaten

$$\hat{\mathbf{x}}_e = [\hat{x}_1^1 \quad \hat{x}_2^1 \quad \hat{x}_3^1 \quad \cdots \quad \hat{x}_1^n \quad \hat{x}_2^n \quad \hat{x}_3^n]^T. \quad (4.25)$$

Die isoparametrische Koordinatentransformation ist in Abbildung 4.1 dargestellt. Die Integrale

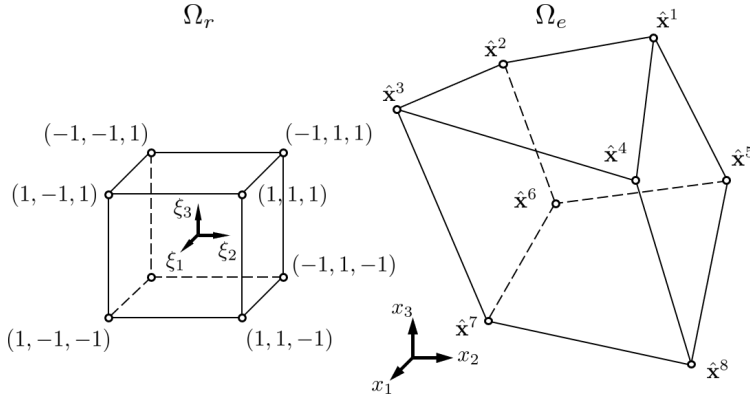


Abbildung 4.1: Transformation eines 8-Knoten Kontinuumelements Ω_e auf das Referenzelement Ω_r ; Knotenkoordinaten $\hat{\mathbf{x}}^i = [\hat{x}_1^i \hat{x}_2^i \hat{x}_3^i]^T$

über das Element Ω_e können mithilfe der Koordinatentransformation in Integrale über das Referenzelement Ω_r überführt werden,

$$\int_{\Omega_e} dV^e = \int_{\Omega_r} \det \mathbf{J} dV^r. \quad (4.26)$$

Der Ausdruck $\det \mathbf{J}$, die Jacobi Determinante, gibt dabei an, wie sich ein infinitesimales Volumenelement dV^e in das entsprechende Volumenelement $\det \mathbf{J} dV^r$ transformiert. Die zugehörige Jacobi Matrix \mathbf{J} berechnet sich folgendermaßen

$$\mathbf{J} = \frac{\partial \mathbf{x}}{\partial \xi} = \begin{bmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_3}{\partial \xi_1} \\ \frac{\partial x_1}{\partial \xi_2} & \frac{\partial x_2}{\partial \xi_2} & \frac{\partial x_3}{\partial \xi_2} \\ \frac{\partial x_1}{\partial \xi_3} & \frac{\partial x_2}{\partial \xi_3} & \frac{\partial x_3}{\partial \xi_3} \end{bmatrix}. \quad (4.27)$$

Mit den Ansatzfunktionen in Referenzelement-Koordinaten $h^i(\xi)$ aus (4.22) und den Knotenkoordinaten $\hat{\mathbf{x}}^i = [\hat{x}_1^i \hat{x}_2^i \hat{x}_3^i]^T$ aus (4.25) kann die Jacobi Matrix auch in einfacher Summenform

angegeben werden (de Borst et al., 2014),

$$\begin{aligned} \frac{\partial h^i}{\partial \boldsymbol{\xi}} &= \left[\frac{\partial h^i}{\partial \xi_1} \quad \frac{\partial h^i}{\partial \xi_2} \quad \frac{\partial h^i}{\partial \xi_3} \right]^T \\ \mathbf{J} &= \sum_{i=1}^n \frac{\partial h^i}{\partial \boldsymbol{\xi}} \hat{\mathbf{x}}^i T, \end{aligned} \quad (4.28)$$

wobei die Summe über die Anzahl n der Elementknoten geht. Durch den isoparametrischen Ansatz kann die in \mathbf{x} formulierte Matrix der Ansatzfunktionen \mathbf{H} aus (4.16) direkt durch die Matrix der Ansatzfunktionen \mathbf{H} aus (4.22) in den Referenzelement-Koordinaten $\boldsymbol{\xi}$ ersetzt werden. In der Regel ist die Gravitation in Gleichung (4.21) \mathbf{g} als Konstante gegeben, sodass hier keine weiteren Umformulierungen nötig sind. Der Ausdruck der Oberflächenlast $\mathbf{t} dA^e$, mit $\mathbf{t} = \mathbf{t}(\mathbf{x}_e)$, muss noch in $\mathbf{t} dA^r$ transformiert werden, mit $\mathbf{t} = \mathbf{t}(\boldsymbol{\xi})$. Die Integralauswertung über die Oberfläche des Elementes $\partial\Omega_e$ wird im Referenzelement über $\partial\Omega_r$ ausgeführt, wobei die Ansatzfunktionen an der Oberfläche $\mathbf{H}|_{\partial\Omega_r}$ des Referenzelementes auszuwerten sind. Nach Wriggers (2013) sind für die Oberflächenintegrale Ansatzfunktionen eines um eine Dimension reduzierten Elementes einzusetzen. Für die Berechnung der internen Kräfte muss die \mathbf{B} -Matrix mit den Ableitungen $\frac{\partial h^i}{\partial x_j}$ noch in den Koordinaten $\boldsymbol{\xi}$ des Referenzelementes formuliert werden.

Für die Transformation der Gradienten der Ansatzfunktion $\frac{\partial h^i}{\partial \mathbf{x}} = \left[\frac{\partial h^i}{\partial x_1} \quad \frac{\partial h^i}{\partial x_2} \quad \frac{\partial h^i}{\partial x_3} \right]^T$ in den Koordinaten \mathbf{x} kann die Inverse der Jacobi Matrix $\mathbf{J}^{-1} = \frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}}$ herangezogen werden², sodass die transformierten Einträge der \mathbf{B} -Matrix, in den Koordinaten $\boldsymbol{\xi}$, den Ausdrücken

$$\frac{\partial h^i}{\partial \mathbf{x}} = \mathbf{J}^{-1} \frac{\partial h^i}{\partial \boldsymbol{\xi}} \quad (4.29)$$

entnommen werden können (de Borst et al., 2014). Mit der \mathbf{B} -Matrix in den Koordinaten des Referenzelementes können die Ausdrücke aus (4.21) in Integrale über das Referenzelement überführt werden,

$$\begin{aligned} \mathbf{f}_{int} &= \sum_{e=1}^{n_e} \mathbf{z}_e^T \int_{\Omega_r} \mathbf{B}^T \boldsymbol{\sigma} \det \mathbf{J} dV^r \\ \mathbf{f}_{ext} &= \sum_{e=1}^{n_e} \mathbf{z}_e^T \int_{\partial\Omega_r} \mathbf{H}^T \mathbf{t} dA^r + \rho \sum_{e=1}^{n_e} \mathbf{z}_e^T \int_{\Omega_r} \mathbf{H}^T \mathbf{g} \det \mathbf{J} dV^r. \end{aligned} \quad (4.30)$$

Der Vorteil des isoparametrischen Konzeptes wird in Gleichung (4.30) deutlich. Für jedes Element sind die Integrationsgrenzen gleich

$$\int_{\Omega_r} \det \mathbf{J} dV^r = \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} \det \mathbf{J} d\xi_1 d\xi_2 d\xi_3. \quad (4.31)$$

Die Knotenkoordinaten $\hat{\mathbf{x}}_e$ sind in den Ausdrücken \mathbf{B} und $\det \mathbf{J}$ als konstante Faktoren in den Polynomen der Koordinaten $\boldsymbol{\xi}$ enthalten. Der Nachteil ist allerdings, dass durch die Inverse Jacobi Matrix \mathbf{J}^{-1} gebrochene rationale Ausdrücke in $\boldsymbol{\xi}$ entstehen.

²Der entscheidende Vorteil ist, dass $\frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}}$ nie explizit berechnet wird, sondern die Jacobi Matrix \mathbf{J} aus (4.28) mit den Regeln der Matrizenalgebra invertiert werden kann.

i	w_i	ξ_1	ξ_2	ξ_3
1	1	$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$
2	1	$+\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$
3	1	$+\frac{1}{\sqrt{3}}$	$+\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$
4	1	$+\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	$+\frac{1}{\sqrt{3}}$
5	1	$-\frac{1}{\sqrt{3}}$	$+\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$
6	1	$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	$+\frac{1}{\sqrt{3}}$
7	1	$-\frac{1}{\sqrt{3}}$	$+\frac{1}{\sqrt{3}}$	$+\frac{1}{\sqrt{3}}$
8	1	$+\frac{1}{\sqrt{3}}$	$+\frac{1}{\sqrt{3}}$	$+\frac{1}{\sqrt{3}}$

Tabelle 4.1: Gauß-Punkte i mit Wichtungsfaktoren w_i für ein 8-Knoten-Kontinuumsselement

4.4 Numerische Integration

Für eine effiziente numerische Auswertung der Integrale in (4.30) hat sich das *Gauß-Quadratur-Verfahren* durchgesetzt. Die Integrale werden durch gewichtete Summen der Integranden, ausgewertet an entsprechenden Stützstellen, approximiert. Für ein 8-Knoten-Kontinuumsselement sind für eine zuverlässige Integration nach dem *Gauß-Quadratur-Verfahren* 8 Stützstellen nötig. Die Stützstellen, auch *Gauß-Punkte* genannt, sind für das 8-Knoten-Kontinuumsselement in Tabelle 4.1 dargestellt. Die Wichtungsfaktoren w_i haben in diesem Fall alle den Wert 1. Die Integrale aus Gleichung (4.30) können in Summen über die Integrationspunkte umgeformt werden,

$$\begin{aligned} \mathbf{f}_{int} &= \sum_{e=1}^{n_e} \mathbf{Z}_e^T \sum_{i=1}^{n_i} w_i \mathbf{B}_i^T \boldsymbol{\sigma}_i \det \mathbf{J}_i \\ \mathbf{f}_{ext} &= \sum_{e=1}^{n_e} \mathbf{Z}_e^T \sum_{i=1}^{n_i} w_i \mathbf{H}_i^T \mathbf{t}_i + \rho \sum_{e=1}^{n_e} \mathbf{Z}_e^T \sum_{i=1}^{n_i} w_i \mathbf{H}_i^T \mathbf{g} \det \mathbf{J}_i, \end{aligned} \quad (4.32)$$

wobei die erste Summe über alle Elemente n_e der Vernetzung \mathcal{B}^h geht und die zweite Summe über alle *Gauß-Punkte* n_i eines Elementes Ω_e zu führen ist. Die numerische Integration leistet einen wesentlichen Beitrag für die effiziente Implementierung in einem computergestützten Berechnungsverfahren. Die Ausdrücke (4.32) werden mit dem *Gauß-Quadratur-Verfahren* allerdings nicht exakt gelöst. Aufgrund der gebrochenrationalen Ausdrücke ist dies nicht möglich. Der Fehler ist allerdings in einer tolerierbaren Größenordnung.

4.5 Lösung der Finite Elemente Gleichungssysteme

Um das FE-Gleichungssystem $\mathbf{f}_{int} = \mathbf{f}_{ext}$ für die gesuchten Knotenverschiebungen $\hat{\mathbf{u}}$ zu lösen, ist das Materialgesetz heranzuziehen. Im Fall der linearen isotropen Elastizität kann der Materialtensor \mathbf{C} in *Voigt*-Notation aus (3.36) herangezogen werden. In *Voigt*-Notation kann die Spannung in einem Element dann folgendermaßen angegeben werden,

$$\boldsymbol{\sigma} = \underbrace{\mathbf{C} \mathbf{B} \mathbf{Z}_e}_e \hat{\mathbf{u}}. \quad (4.33)$$

Wird Gleichung (4.33) in den Ausdruck für \mathbf{f}_{int} eingesetzt, so kann die Spaltenmatrix der Knotenverschiebungen $\hat{\mathbf{u}}$ ausgeklammert werden

$$\mathbf{f}_{int} = \underbrace{\left(\sum_{e=1}^{n_e} \mathbf{z}_e^T \left(\sum_{i=1}^{n_i} w_i \mathbf{B}_i^T \mathbf{C} \mathbf{B}_i \det \mathbf{J}_i \right) \mathbf{z}_e \right)}_{\mathbf{K}} \hat{\mathbf{u}}. \quad (4.34)$$

In diesem Fall ist dann ein lineares Gleichungssystem

$$\mathbf{K} \hat{\mathbf{u}} = \mathbf{f}_{ext} \quad (4.35)$$

zu lösen. Hierin ist \mathbf{K} die $N \times N$ -Steifigkeitsmatrix der Struktur. N ist hierbei die Anzahl der globalen Freiheitsgrade bzw. Knotenverschiebungen $\hat{\mathbf{u}}$. In das Gleichungssystem müssen noch die *Dirichlet* Randbedingungen aus (4.3) eingebaut werden. Die *Neumann* Randbedingungen sind schon in \mathbf{f}_{ext} enthalten. Die *Dirichlet* Randbedingungen werden als vorgegebene Knotenverschiebungen in $\hat{\mathbf{u}}$ eingesetzt. Durch das Tauschen von Zeilen und Spalten kann das Gleichungssystem (4.35) in folgende *Unterstrukturen* aufgeteilt werden

$$\begin{bmatrix} \mathbf{K}^{11} & \mathbf{K}^{12} \\ \mathbf{K}^{21} & \mathbf{K}^{22} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}}^{\bar{t}} \\ \hat{\mathbf{u}}^{\bar{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{ext}^{\bar{t}} \\ \mathbf{f}_{ext}^{\bar{u}} \end{bmatrix}. \quad (4.36)$$

Dabei sind $\hat{\mathbf{u}}^{\bar{t}}$ die unbekanntenen Knotenverschiebungen der Struktur \mathcal{B}^h . Die vorgegebenen Knotenkräfte $\mathbf{f}_{ext}^{\bar{t}}$ setzen sich aus den Oberflächenlasten auf dem *Neumann*-Rand und den Volumenkräften über \mathcal{B}^h zusammen. Der Anteil $\hat{\mathbf{u}}^{\bar{u}}$ beschreibt die vorgegebenen Knotenverschiebungen, also die *Dirichlet*-Randbedingungen. Die Knotenkräfte $\mathbf{f}_{ext}^{\bar{u}}$ können nach dem Schnittprinzip als wirkende Lagerkräfte auf dem *Dirichlet*-Rand betrachtet werden, die zunächst unbekannt sind. Zuerst wird das Gleichungssystem

$$\mathbf{K}^{11} \hat{\mathbf{u}}^{\bar{t}} = \mathbf{f}_{ext}^{\bar{t}} - \mathbf{K}^{12} \hat{\mathbf{u}}^{\bar{u}} \quad (4.37)$$

für $\hat{\mathbf{u}}^{\bar{t}}$ gelöst. Im Anschluss können die Lagerkräfte berechnet werden,

$$\mathbf{f}_{ext}^{\bar{u}} = \mathbf{K}^{21} \hat{\mathbf{u}}^{\bar{t}} + \mathbf{K}^{22} \hat{\mathbf{u}}^{\bar{u}}. \quad (4.38)$$

Die Steifigkeitsmatrizen \mathbf{K}^{ij} ($i = 1, 2$) sind symmetrische positiv definite Bandmatrizen, sodass für die Lösung des FE-Gleichungssystems (4.37) beispielsweise der LR-Algorithmus oder das *Cholesky*-Verfahren angewendet werden können (Gawehn und Funk, 2020).

4.5.1 Iteratives Lösungsverfahren: *Newton–Raphson* Methode

Wenn kein linear elastisches Materialverhalten vorliegt, lässt sich das Gleichungssystem $\mathbf{f}_{int} = \mathbf{f}_{ext}$ nicht in ein lineares Gleichungssystem wie in (4.35) überführen. In einem solchen Fall von nichtlinearem Materialverhalten muss dann eine *angenäherte* Lösungsfindung formuliert werden. Aus mathematischer Sicht kann dies als eine Nullstellensuche in $\hat{\mathbf{u}}$ für den Ausdruck $\mathbf{f}_{int} - \mathbf{f}_{ext} = \mathbf{0}$ betrachtet werden. Durch das nichtlineare Materialverhalten ist es zweckmäßig, die externe Belastung \mathbf{f}_{ext} nicht in einem Schritt aufzubringen. Vielmehr ist ein inkrementelles Aufbringen der Last vorteilhaft, wodurch ein sogenannter *Lastpfad* entsteht. Grundsätzlich wird zwischen *inkrementellen* und *inkrementell-iterativen* Lösungsverfahren unterschieden. Die rein inkrementellen Lösungsverfahren haben den Nachteil, dass sie vom vorgegebenen Lastpfad abdriften (de Borst et al., 2014; Ibrahimbegovic, 2009). Unter „Abdriften“ ist zu verstehen, dass \mathbf{f}_{int} dem vorgegebenen Lastpfad der äußeren Belastung nicht folgt und somit ein immer größer werdendes Ungleichgewicht entsteht. Dies kann durch kleine Schritte im Lastpfad minimiert, aber nie ganz unterbunden werden, sodass ein Abdriften in jedem Fall in Kauf zu nehmen ist.

Allerdings sind diese Verfahren relativ schnell. Das Defizit des Abdriften vom Lastpfad wird bei den *inkrementell-iterativen* Verfahren durch eine *Gleichgewichtsiteration* in jedem Lastschritt beseitigt. Das bekannteste Verfahren ist dabei das nach *Newton-Raphson*, was im Folgenden hergeleitet werden soll³.

Für das Aufbringen der Last wird ein Kopfzeiger ${}^n(\cdot)$ eingeführt, der den n -ten Lastschritt kennzeichnet. Damit kann folgender Ausdruck für den Lastschritt $n + 1$ angegeben werden,

$$\begin{aligned} {}^{n+1}\mathbf{f}_{int} &= {}^{n+1}\mathbf{f}_{ext} \\ {}^n\mathbf{f}_{int} + \Delta\mathbf{f}_{int} &= {}^{n+1}\mathbf{f}_{ext} \\ \Delta\mathbf{f}_{int} &= {}^{n+1}\mathbf{f}_{ext} - {}^n\mathbf{f}_{int}, \end{aligned} \quad (4.39)$$

wobei $\Delta\mathbf{f}_{int}$ der Zuwachs der internen Knotenkräfte ist. Die Größen auf der rechten Seite in Gleichung (4.39) sind bekannt. Unbekannt ist das Inkrement $\Delta\hat{\mathbf{u}}$, welches das Inkrement der internen Kräfte $\Delta\mathbf{f}_{int}$ auf der rechten Seite der Gleichung erzeugt,

$$\Delta\mathbf{f}_{int} = \sum_{e=1}^{n_e} \mathbf{Z}_e^T \sum_{i=1}^{n_i} w_i \mathbf{B}_i^T \Delta\boldsymbol{\sigma}_i \det\mathbf{J}_i. \quad (4.40)$$

Das Inkrement $\Delta\hat{\mathbf{u}}$ kann durch eine *Linearisierung* approximiert bzw. angenähert werden. Hierfür muss das Inkrement des Spannungstensors $\Delta\boldsymbol{\sigma}_i$ linearisiert werden, wobei i sich auf einen Integrationspunkt⁴ bezieht

$$\Delta\boldsymbol{\sigma}_i = \left. \frac{\partial\boldsymbol{\sigma}(i)}{\partial\boldsymbol{\varepsilon}(i)} \right|_n \underbrace{\mathbf{B}_i \mathbf{Z}_e \Delta\hat{\mathbf{u}}}_{\Delta\boldsymbol{\varepsilon}_i}. \quad (4.41)$$

Dabei ist die Ableitung der Spannungen nach den Dehnungen zum Lastschritt n die sogenannte *Materialtangente* bzw. der verzerrungsabhängige Materialtensor $\mathbf{C}(\boldsymbol{\varepsilon})$. Im Rahmen dieser Arbeit wird das elasto-plastische Materialverhalten aus Kapitel 3.4.2 herangezogen. Die algorithmischen Aspekte aus dem Kapitel 3.4.3 können direkt auf Integrationspunktebene umgesetzt werden, sodass für die Materialtangente im Integrationspunkt i

$$\left. \frac{\partial\boldsymbol{\sigma}(i)}{\partial\boldsymbol{\varepsilon}(i)} \right|_n = {}^n\mathbf{C}_i^{ep} \quad (4.42)$$

angegeben werden kann. Die Matrixeinträge für die 6×6 -Materialtangente ${}^n\mathbf{C}_i^{ep}$ am Integrationspunkt i können der vollständig algorithmisch konsistenten Materialtangente ${}^n\mathbf{C}^{ep}$ aus Gleichung (3.79) entnommen werden. Die Koeffizienten werden nach Ibrahimbegovic (2009) folgendermaßen für die *Voigt*-Notation in die 6×6 -Matrix eingetragen:

$${}^n\mathbf{C}_i^{ep} = \begin{bmatrix} {}^n\mathbf{C}_{1111}^{ep} & {}^n\mathbf{C}_{1122}^{ep} & {}^n\mathbf{C}_{1133}^{ep} & {}^n\mathbf{C}_{1112}^{ep} & {}^n\mathbf{C}_{1123}^{ep} & {}^n\mathbf{C}_{1131}^{ep} \\ & {}^n\mathbf{C}_{2222}^{ep} & {}^n\mathbf{C}_{2233}^{ep} & {}^n\mathbf{C}_{2212}^{ep} & {}^n\mathbf{C}_{2223}^{ep} & {}^n\mathbf{C}_{2231}^{ep} \\ & & {}^n\mathbf{C}_{3333}^{ep} & {}^n\mathbf{C}_{3312}^{ep} & {}^n\mathbf{C}_{3323}^{ep} & {}^n\mathbf{C}_{3331}^{ep} \\ & & & {}^n\mathbf{C}_{1212}^{ep} & {}^n\mathbf{C}_{1223}^{ep} & {}^n\mathbf{C}_{1231}^{ep} \\ & & & & {}^n\mathbf{C}_{2323}^{ep} & {}^n\mathbf{C}_{2331}^{ep} \\ & & & & & {}^n\mathbf{C}_{3131}^{ep} \end{bmatrix}_i \quad (4.43)$$

³Im eindimensionalen Fall einer skalarwertigen Funktion $y = f(x)$ entspricht dieses Verfahren, dann auch *Newton*-Verfahren genannt, einem Tangentenverfahren zur Nullstellenfindung $f(x_0) = 0$.

⁴Der Ausdruck $\left. \frac{\partial\boldsymbol{\sigma}(i)}{\partial\boldsymbol{\varepsilon}(i)} \right|_n$ löst aufgrund der geklammerten Indices in diesem Fall nicht die *Einstein'sche* Summationskonvention aus.

Für das Inkrement der internen Kräfte ergibt sich somit folgender Ausdruck,

$$\Delta \mathbf{f}_{int} = \underbrace{\left(\sum_{e=1}^{n_e} \mathbf{Z}_e^T \left(\sum_{i=1}^{n_i} w_i \mathbf{B}_i^T {}^n \mathbf{C}_i^{ep} \mathbf{B}_i \det \mathbf{J}_i \right) \mathbf{Z}_e \right)}_{{}^n \mathbf{K}} \Delta \hat{\mathbf{u}}, \quad (4.44)$$

mit der *tangentialen* Steifigkeitsmatrix ${}^n \mathbf{K}$ im Lastschritt n . Die linearisierte Variante der Gleichung (4.39) wird folgendermaßen angegeben

$${}^n \mathbf{K} \Delta \hat{\mathbf{u}} = {}^{n+1} \mathbf{f}_{ext} - {}^n \mathbf{f}_{int}. \quad (4.45)$$

In einem *rein* inkrementellen Verfahren werden nach der Bestimmung von $\Delta \hat{\mathbf{u}}$ nicht nur die Größen ${}^n \mathbf{K}$ und ${}^n \mathbf{f}_{int}$ neu berechnet bzw. aktualisiert, sondern auch die externe Last ${}^{n+1} \mathbf{f}_{ext}$ weiter erhöht. Dies führt, wie eingangs beschrieben, zum Abdriften der Lösung. Um ein Abdriften zu verhindern, wird im *Newton-Raphson*-Verfahren eine Gleichgewichtsiteration innerhalb eines Lastschrittes durchgeführt. Hierfür wird ein zusätzlicher Fußzeiger ${}_k(\cdot)$ für die Gleichgewichtsiteration eingeführt. In der sogenannten *Materialroutine* $\text{Mat}(\boldsymbol{\varepsilon})$ nach Kapitel 3.4.3 findet die Berechnung der Materialtangente ${}^n \mathbf{C}_i^{ep}$ und der Spannungen ${}^n \boldsymbol{\sigma}_i$ statt. Eine typische Gleichgewichtsiteration innerhalb des *Newton-Raphson*-Verfahrens ist in Algorithmus 1 dargestellt. Dabei ist ϵ eine vorzugebende Genauigkeit. Die mit ${}_0(\cdot)$ indizierten Startgrößen in

Algorithmus 1 Gleichgewichtsiteration innerhalb des *Newton-Raphson* Verfahrens

- 1: ${}_0 \mathbf{C}_i^{ep}, {}_0 \boldsymbol{\sigma}_i \mapsto {}_0 \mathbf{K}, {}_0 \mathbf{f}_{int}$
 - 2: ${}_0 \mathbf{K} \Delta \hat{\mathbf{u}} = {}_0^{+1} \mathbf{f}_{ext} - {}_0 \mathbf{f}_{int} \mapsto \Delta \hat{\mathbf{u}}$
 - 3: $\Delta \hat{\mathbf{u}} \mapsto {}_{k+1} \boldsymbol{\varepsilon}_i = {}_k \boldsymbol{\varepsilon}_i + \mathbf{B}_i \mathbf{Z}_e \Delta \hat{\mathbf{u}}$
 - 4: $\text{Mat} \left({}_{k+1} \boldsymbol{\varepsilon}_i \right) \mapsto {}_{k+1} \mathbf{C}_i^{ep}, {}_{k+1} \boldsymbol{\sigma}_i$
 - 5: ${}_{k+1} \mathbf{C}_i^{ep}, {}_{k+1} \boldsymbol{\sigma}_i \mapsto {}_{k+1} \mathbf{K}, {}_{k+1} \mathbf{f}_{int}$
 - 6: ${}^{n+1} \mathbf{f}_{ext} - {}_{k+1} \mathbf{f}_{int} > \epsilon \mapsto \begin{cases} \text{erfüllt :} & \text{Weiter mit Schritt 7} \\ \text{nicht erfüllt :} & \text{Ende der Iteration} \mapsto {}^{n+1} \mathbf{f}_{ext} + \Delta \mathbf{f}_{ext} \end{cases}$
 - 7: ${}_{k+1} \mathbf{K} \Delta \hat{\mathbf{u}} = {}^{n+1} \mathbf{f}_{ext} - {}_{k+1} \mathbf{f}_{int} \mapsto \Delta \hat{\mathbf{u}} : \text{Weiter mit Schritt 3}$
-

Schritt 1 resultieren dabei aus einem *konvergierenden* vorangehenden Schritt n . In Schritt 6 bedeutet das Nicht-Erfüllen der Bedingung, dass die Gleichgewichtsiteration beendet werden kann und die geforderte Genauigkeit ϵ erreicht ist. Demzufolge kann die externe Last \mathbf{f}_{ext} weiter erhöht werden und mit der Gleichgewichtsiteration im nächsten Lastschritt erneut begonnen werden. Ist die Bedingung im Schritt 6 erfüllt, ist die vorgegebene Genauigkeit ϵ in der Gleichgewichtsiteration noch nicht erreicht. In diesem Fall wird in Schritt 7 ein weiteres Inkrement $\Delta \hat{\mathbf{u}}$ berechnet und mit Schritt 3 beginnend, der gleiche Vorgang wiederholt. In der Abbildung 4.2 ist die Gleichgewichtsiteration für einen Lastschritt gezeigt. Die Gesamtverschiebung $\hat{\mathbf{u}}$ in einer Iteration ergibt sich dabei immer nach folgender Gleichung,

$${}^n \hat{\mathbf{u}} = {}_0 \hat{\mathbf{u}} + \sum_{k=1}^{n_k} {}_k \Delta \hat{\mathbf{u}}, \quad (4.46)$$

wobei ${}_0 \hat{\mathbf{u}}$ der Verschiebungszustand nach einem konvergierten vorangegangenen Lastschritt ist und n_k die Anzahl der erforderlichen Iterationen bis zur Konvergenz darstellt. Da in jedem Iterationsschritt die tangentiale Steifigkeit und die internen Kräfte neu berechnet werden müssen,

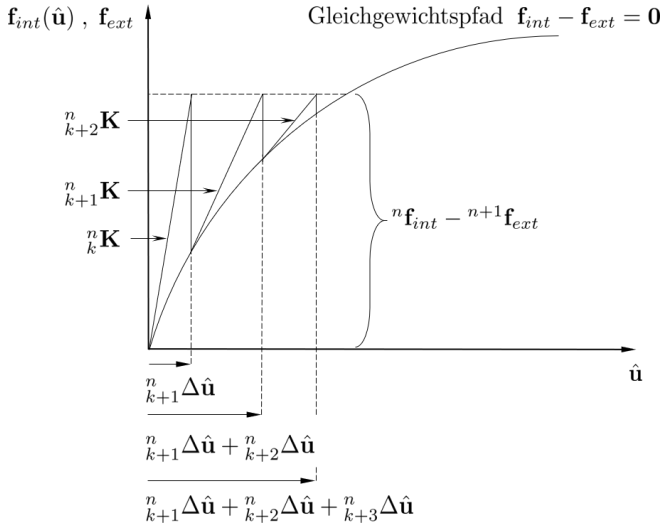


Abbildung 4.2: Gleichgewichtsiteration im Newton-Raphson-Verfahren für einen Lastschritt

ist das vollständige Newton-Raphson-Verfahren sehr rechenintensiv. Ein modifiziertes Newton-Raphson Verfahren sieht vor, die tangentiale Steifigkeitsmatrix lediglich zu Beginn eines jeden Lastschrittes zu berechnen und dann durchgehend für die Gleichgewichtsiteration zu verwenden, was den Berechnungsaufwand deutlich verringert. Allerdings geht dadurch die quadratische Konvergenz des Newton-Raphson Verfahrens verloren.

Bei der Vorgabe der Strukturbelastung wird zwischen Kraftgesteuert und Verschiebungsgesteuert unterschieden. Bei einer kraftgesteuerten Berechnung werden die externe Last auf dem Neumann-Rand $\partial_t \mathcal{B}^h$ und die Volumenlasten in \mathcal{B}^h vorgegeben, sodass \mathbf{f}_{ext}^t bekannt ist und inkrementell erhöht wird. Die Knotenverschiebungen $\hat{\mathbf{u}}^{\bar{u}} = \mathbf{0}$ auf dem Dirichlet-Rand $\partial_u \mathcal{B}^h$ werden dabei blockiert. Bei einer verschiebungsgesteuerten Analyse wird die Verschiebung $\hat{\mathbf{u}}^{\bar{u}} \neq \mathbf{0}$ auf dem Dirichlet-Rand $\partial_u \mathcal{B}^h$ vorgegeben bzw. inkrementell erhöht. Der Anteil der Oberflächenlast in den externen Knotenkräften \mathbf{f}_{ext}^t auf dem Neumann-Rand $\partial_t \mathcal{B}^h$ ist dabei Null sodass nur die Anteile aus der Volumenlast enthalten sind. Wenn das Eigengewicht der Struktur und somit die Volumenlasten vernachlässigbar sind, kann von $\mathbf{f}_{ext}^t = \mathbf{0}$ ausgegangen werden. Dies ist in vielen Ingenieur Anwendungen mit hinreichender Genauigkeit gerechtfertigt. Die Knotenkräfte \mathbf{f}_{ext}^u auf dem Dirichlet-Rand $\partial_u \mathcal{B}^h$ sind dann die zu den vorgegebenen Verschiebungen $\hat{\mathbf{u}}^{\bar{u}}$ gehörigen unbekannt Knotenkräfte. Das linearisierte Gleichungssystem (4.45) kann in Analogie zum linearem Fall (4.36) folgendermaßen aufgeteilt werden,

$$\begin{bmatrix} {}^n \mathbf{K}^{11} & {}^n \mathbf{K}^{12} \\ {}^n \mathbf{K}^{21} & {}^n \mathbf{K}^{22} \end{bmatrix} \begin{bmatrix} \Delta \hat{\mathbf{u}}^t \\ \Delta \hat{\mathbf{u}}^{\bar{u}} \end{bmatrix} = \begin{bmatrix} {}^{n+1} \mathbf{f}_{ext}^t \\ {}^{n+1} \mathbf{f}_{ext}^{\bar{u}} \end{bmatrix} - \begin{bmatrix} {}^n \mathbf{f}_{int}^t \\ {}^n \mathbf{f}_{int}^{\bar{u}} \end{bmatrix}. \quad (4.47)$$

Bei einer kraftgesteuerten Berechnung mit $\hat{\mathbf{u}}^{\bar{u}} = \mathbf{0}$ wird die Gleichung

$${}^n \mathbf{K}^{11} \Delta \hat{\mathbf{u}}^t = {}^{n+1} \mathbf{f}_{ext}^t - {}^n \mathbf{f}_{int}^t \quad (4.48)$$

mit dem Newton-Raphson-Verfahren gelöst, siehe Algorithmus 1 und Abbildung 4.2. Die tangentialen Steifigkeitsmatrizen ${}^n \mathbf{K}^{11}$ sind dabei symmetrische positiv definite Bandmatrizen, die

beispielsweise wieder mit dem *Cholesky*-Verfahren gelöst werden können.

In einer *reinen* verschiebungsgesteuerten Berechnung mit $\mathbf{f}_{ext}^{\bar{t}} = \mathbf{0}$ muss für einen Lastschritt ein Verschiebungsinkrement ${}_0^{n+1}\Delta\hat{\mathbf{u}}^{\bar{u}}$ vorgegeben werden, sodass zum Beginn einer *Newton-Raphson* Gleichgewichtsiteration folgendes System gegeben ist

$${}_0^n\mathbf{K}^{11}\Delta\hat{\mathbf{u}}^{\bar{t}} = {}_0^n\mathbf{K}^{12} {}_0^{n+1}\Delta\hat{\mathbf{u}}^{\bar{u}} - {}_0^n\mathbf{f}_{int}^{\bar{t}}. \quad (4.49)$$

Dabei kann der Term ${}_0^n\mathbf{K}^{12} {}_0^{n+1}\Delta\hat{\mathbf{u}}^{\bar{u}}$ als äquivalente externe Kraft angesehen werden (de Borst et al., 2014). Der Schritt in Gleichung (4.49) ist nötig, um die Gleichgewichtsiteration für das vorgegebene Verschiebungsinkrement ${}_0^{n+1}\Delta\hat{\mathbf{u}}^{\bar{u}}$ zu starten, da zu Beginn ${}_0^n\mathbf{f}_{int}^{\bar{t}} = \mathbf{0}$ gilt. Innerhalb der folgenden Schritte in der Gleichgewichtsiteration $k = 1 \dots n_k$ erfährt das vorgegebene Inkrement ${}_0^{n+1}\Delta\hat{\mathbf{u}}^{\bar{u}}$ keinen Zuwachs, sodass ${}_{k>0}^{n+1}\Delta\hat{\mathbf{u}}^{\bar{u}} = \mathbf{0}$ gilt (de Borst et al., 2014). Demnach leistet die zugehörige tangentielle Steifigkeitsmatrix ${}_{k>0}^n\mathbf{K}^{12} = \mathbf{0}$ auch keinen Beitrag mehr, sodass in der Gleichgewichtsiteration folgendes System für ${}_k^n\Delta\hat{\mathbf{u}}^{\bar{t}}$ „ausiteriert“ werden muss

$${}_k^n\mathbf{K}^{11} {}_k^n\Delta\hat{\mathbf{u}}^{\bar{t}} = {}_k^n\mathbf{f}_{int}^{\bar{t}}. \quad (4.50)$$

Am Ende der Gleichgewichtsiteration $k = 1 \dots n_k$ gilt für die ins *Ungleichgewicht* gebrachten internen Knotenkräfte ${}_{k=n_k}^n\mathbf{f}_{int}^{\bar{t}} \approx \mathbf{0}$ und das zugehörige Verschiebungsinkrement ${}_{k=n_k}^n\Delta\hat{\mathbf{u}}^{\bar{t}}$ ist bestimmt. Die zum vorgegebenen Verschiebungsinkrement ${}_0^{n+1}\Delta\hat{\mathbf{u}}^{\bar{u}}$ gehörenden Knotenkräfte ${}_{k=n_k}^n\mathbf{f}_{ext}^{\bar{u}}$ können dann in einer Nachlaufrechnung bestimmt werden. Auch wenn die vorgegebenen Verschiebungsinkremente ${}_0^{n+1}\Delta\hat{\mathbf{u}}^{\bar{u}}$ nicht „offensichtlich“ in der eigentlichen Gleichgewichtsiteration (4.50) ersichtlich sind, so haben sie Einfluss auf die Berechnung der tangentialen Steifigkeitsmatrizen ${}_{k=0}^n\mathbf{K}^{11}$, da die Elemente am vorgegebenen *Dirichlet*-Rand auch Knoten im restlichen Gebiet \mathcal{B}^h haben und die Matrizen mit den Verzerrungen am Integrationspunkt berechnet werden.

In der Regel ist eine verschiebungsgesteuerte Analyse stabiler im Konvergenzverhalten als eine kraftgesteuerte Berechnung. Darüber hinaus gibt es viele Strategien, um das Konvergenzverhalten in beiden Verfahren der Belastung zu verbessern. In den zu Verfügung stehenden kommerziellen FEM Softwarepaketen sind in der Regel Optimierungsalgorithmen zur Sicherung der Konvergenz integriert.

5 Support Vector Regression (SVR)

Die *Support Vector Regression* (SVR) stellt ein effektives, *nicht-intrusives* Verfahren der Modellreduktion dar. In den Lehrbüchern von Cristianini et al. (2000), Campbell und Ying (2011), Shawe-Taylor et al. (2004), Jiang et al. (2020), Fasshauer und McCourt (2015), Kung (2014) und Forrester et al. (2008) wird dieses Verfahren ausführlich vorgestellt, um nur einige an dieser Stelle zu nennen. Veröffentlichungen in diesem Bereich enthalten in der Regel eine kurze Einführung in das Thema, wie beispielsweise in Cristianini und Shawe-Taylor (2007), Smola und Schölkopf (2004), Clarke et al. (2004), Steiner et al. (2019) Moustapha et al. (2018), An et al. (2020), Moustapha und Sudret (2019), Wang et al. (2009), Ciccazzo et al. (2014), Xiang et al. (2017), Jiang et al. (2018) und Zhu et al. (2012), um auch hier nur einige zu nennen. Im Rahmen dieser Arbeit wird die in MATLAB[®] integrierte *Support Vector Regression* herangezogen, für die neben der Anwenderdokumentation eine Theorie-Dokumentation von MATLAB[®] zur Verfügung gestellt wird. Im folgenden Kapitel soll die Theorie zur *Support Vector Regression* in dem Umfang vorgestellt werden wie sie für das Verständnis der weiteren Kapitel 6 und 7 notwendig ist.

5.1 Support Vector Maschinen für Regression

Die folgende Darlegung der Theorie erfolgt auf Grundlage der unter Kapitel 5 aufgeführten Quellen. In erster Linie sind hierbei die Lehrbücher von Cristianini et al. (2000), Shawe-Taylor et al. (2004), Fasshauer und McCourt (2015) und Jiang et al. (2020) sowie die Publikationen von Moustapha et al. (2018), Smola und Schölkopf (2004) und Cristianini und Shawe-Taylor (2007) zu nennen.

Die *Support Vector Machines*¹ (SVM) werden für *Klassifizierungen* und *Regressionen* verwendet. Im Rahmen dieser Arbeit wird das Verfahren für Problemstellungen aus dem Bereich der Regression herangezogen. Die Herleitung des Verfahrens findet im Folgenden somit für diesen Anwendungsfall statt. Ein Trainingsdatensatz für das reduzierte Modell sei folgendermaßen vorgegeben,

$$S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}, \quad (5.1)$$

wobei $\mathbf{x}_i \in \mathbb{R}^{m \times 1}$ als Argument einer Zielfunktion (*engl. Target Function*) $y = t(\mathbf{x})$ angesehen werden kann und $y_i \in \mathbb{R}$ die Funktionsantwort für diesen entsprechenden i -ten Trainingspunkt darstellt.

Klassische Regressionansätze

In der „klassischen“ *linearen* Regression sollen ein Vektor $\mathbf{w} \in \mathbb{R}^{m \times 1}$ und eine Konstante b so bestimmt werden, dass für den Ausdruck (Cristianini et al., 2000)

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b \quad (5.2)$$

die Differenz $t(\mathbf{x}) - f(\mathbf{x})$ zwischen dem reduzierten (Regressions-) Modell f und der Zielfunktion t möglichst gering wird. In der Gleichung (5.2) wird das *innere* Produkt $\langle \cdot, \cdot \rangle$ eingeführt. In diesem Fall entspricht es dem Matrizenprodukt $\mathbf{w}^T \mathbf{x} := \langle \mathbf{w}, \mathbf{x} \rangle$, ist aber nicht hierauf beschränkt, sodass auch Funktionen oder Elemente aus Vektorräumen unendlicher Dimension als

¹Die Bezeichnung wird im Hinblick auf die angelsächsische Literatur verwendet. Die deutsche Übersetzung „Unterstützungs-Vektor Maschinen“ wird nicht verwendet.

Argumente eingesetzt werden können. In der *Methode der kleinsten Quadrate* wird vorgeschlagen, den Ausdruck

$$L(\mathbf{w}, b) = \sum_{i=1}^n (t(\mathbf{x}_i) - f(\mathbf{x}_i))^2 = \sum_{i=1}^n (y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b)^2 \quad (5.3)$$

zu minimieren. Die Größe $L(\cdot)$ kennzeichnet hierbei den zu *minimierenden* Ausdruck, mit der Angabe der diesbezüglich zu variierenden Argumente in der Klammer. In diesem Zusammenhang wird $L(\cdot)$ auch als *quadratische Verlustfunktion* bezeichnet. In Cristianini et al. (2000) wird die Überführung in ein Matrizenschema mit $\hat{\mathbf{w}} = [\mathbf{w}^T b]^T$ und

$$\hat{\mathbf{X}} = \begin{bmatrix} \hat{\mathbf{x}}_1^T \\ \vdots \\ \hat{\mathbf{x}}_n^T \end{bmatrix} \quad (5.4)$$

vorgeschlagen, wobei $\hat{\mathbf{x}}_i = [\mathbf{x}_i^T 1]^T$ ist. Das Minimierungsproblem (5.3) kann somit wie folgt aufgeschrieben werden,

$$L(\hat{\mathbf{w}}) = (\mathbf{y} - \hat{\mathbf{X}}\hat{\mathbf{w}})^T (\mathbf{y} - \hat{\mathbf{X}}\hat{\mathbf{w}}), \quad (5.5)$$

wobei $\mathbf{y} = [y_1 \cdots y_n]^T$ ist. Die Lösung für $\hat{\mathbf{w}}$ kann im Sinne einer Minimierungsaufgabe durch das Ableiten von L nach $\hat{\mathbf{w}}$ und anschließendes „zu Null“ setzen ermittelt werden,

$$\frac{\partial L}{\partial \hat{\mathbf{w}}} = -2\hat{\mathbf{X}}^T \mathbf{y} + 2\hat{\mathbf{X}}^T \hat{\mathbf{X}} \hat{\mathbf{w}} = \mathbf{0}. \quad (5.6)$$

Die Umstellung dieser Gleichung, die eine Matrizeninversion beinhaltet, führt auf die *optimale* Lösung

$$\hat{\mathbf{w}} = (\hat{\mathbf{X}}^T \hat{\mathbf{X}})^{-1} \hat{\mathbf{X}}^T \mathbf{y}. \quad (5.7)$$

Wenn die Matrix $\hat{\mathbf{X}}^T \hat{\mathbf{X}}$ keinen vollen Rang hat, kann es zu Stabilitätsproblemen während des Lösungsverfahrens der Gleichung (5.7) kommen, sodass eine Erweiterung in Form der sogenannten *Ridge* Regression hier Abhilfe schaffen kann. Bei diesem Verfahren wird eine *penalisierte* Verlustfunktion angesetzt, in der zwischen geringen quadratischen Abweichungen $(y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b)^2$ und geringen Normen $\|\mathbf{w}\|^2 = \langle \mathbf{w}, \mathbf{w} \rangle$ über einen Parameter $\lambda \in \mathbb{R}^+$ ausgeglichen wird (Cristianini et al., 2000),

$$L(\mathbf{w}, b) = \lambda \langle \mathbf{w}, \mathbf{w} \rangle + \sum_{i=1}^n (y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b)^2. \quad (5.8)$$

Für die Lösung des Minimierungsproblems wird über die Ableitung $\frac{\partial L}{\partial \mathbf{w}} = \mathbf{0}$ die Bestimmungsgleichung

$$\begin{aligned} \lambda \mathbf{w} + \sum_{i=1}^n (y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b) \mathbf{x}_i &= \mathbf{0} \\ \mathbf{w} &= - \sum_{i=1}^n \frac{1}{\lambda} (y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b) \mathbf{x}_i \end{aligned} \quad (5.9)$$

erhalten, aus der sich entnehmen lässt, dass die Lösung \mathbf{w} als eine gewichtete Summe der Trainingsdaten \mathbf{x}_i darstellbar ist. Dies führt auf die sogenannte *duale* Darstellung,

$$\begin{aligned}\mathbf{w} &= \sum_{i=1}^n \alpha_i \mathbf{x}_i \\ \alpha_i &= -\frac{1}{\lambda} (y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b),\end{aligned}\quad (5.10)$$

in der die inneren Produkte nicht mehr direkt auftauchen und stattdessen in den *dualen* Variablen $\boldsymbol{\alpha} = [\alpha_1 \cdots \alpha_n]^T$ die Lösung gesucht wird. Mit der Matrix $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_n]^T \in \mathbb{R}^{n \times m}$ der Trainingsdaten lässt sich die erste Gleichung (5.10) in der Matrixform $\mathbf{w} = \mathbf{X}^T \boldsymbol{\alpha}$ ausdrücken. Die *penalisierte* Verlustfunktion (5.8) kann jetzt in eine (*duale*) Matrizenform umgeschrieben werden,

$$\begin{aligned}L(\boldsymbol{\alpha}) &= \lambda \boldsymbol{\alpha}^T \mathbf{X} \mathbf{X}^T \boldsymbol{\alpha} + \sum_{i=1}^n (\boldsymbol{\alpha}^T \mathbf{X} \mathbf{x}_i - y_i)^T \\ L(\boldsymbol{\alpha}) &= \lambda \boldsymbol{\alpha}^T \mathbf{G} \boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{G} \boldsymbol{\alpha} - 2\mathbf{y}^T \mathbf{G} \boldsymbol{\alpha} + \mathbf{y}^T \mathbf{y},\end{aligned}\quad (5.11)$$

mit der *Gramschen* Matrix $\mathbf{G} = \mathbf{X} \mathbf{X}^T$, deren Einträge die Produkte $G_{ij} = \mathbf{x}_i^T \mathbf{x}_j$ sind. Das Minimierungsproblem wird jetzt über $\frac{\partial L}{\partial \boldsymbol{\alpha}} = \mathbf{0}$ in den *dualen* Variablen gelöst, was auf folgende Bestimmungsgleichung für $\boldsymbol{\alpha}$ führt,

$$\boldsymbol{\alpha} = (\lambda \mathbf{I} + \mathbf{G})^{-1} \mathbf{y}.\quad (5.12)$$

In seiner *dualen* Form kann das reduzierte (*Ridge* Regressions-) Modell anders als in der *primären* Form (5.2) folgendermaßen angegeben werden,

$$f(\mathbf{x}) = \sum_{i=1}^n \alpha_i z_i = \boldsymbol{\alpha}^T \mathbf{z},\quad (5.13)$$

wobei $\mathbf{z} = [z_1 \cdots z_n]^T$ der Vektor mit den Einträgen $z_i = \mathbf{x}_i^T \mathbf{x} := \langle \mathbf{x}_i, \mathbf{x} \rangle$ ist.

Kernel Funktion

Die *Methode der kleinsten Quadrate* und ihre Erweiterung in Form der *Ridge* Regression stellen sogenannte *lineare* Lernalgorithmen² dar. Die Anwendung dieser Verfahren stößt in der Praxis schnell an ihre Grenzen, wenn die Zielfunktion $y = t(\mathbf{x})$ und somit auch die Trainingsdaten $(\mathbf{x}_i, y_i)_{i=1 \dots n}$ beispielsweise *Nichtlinearitäten* und *Unregelmäßigkeiten* enthalten. Eine Möglichkeit, mit der auch in solchen Fällen weiter mit den vorgestellten und effizienten *linearen* Algorithmen gearbeitet werden kann, ist die Einführung sogenannter *Kernel*³ Funktionen. Die Idee ist es, die *Datenstruktur* in einen anderen mathematischen Raum zu transformieren, in dem dann lineare Algorithmen wieder erfolgreich eingesetzt werden können. Dieser Raum wird als Merkmal-Raum (*engl. Feature Space*) F bezeichnet und dient dazu, vorliegende nichtlineare Strukturen im *Eingaberaum* in lineare Strukturen zu überführen. Um diesen Schritt zu motivieren ist als Beispiel in Abbildung 5.1 die Überführung von unstrukturierten Daten aus dem Eingaberaum in korrespondierende strukturierte Daten im Merkmal-Raum aufgezeigt. In der Abbildung ist auch angedeutet, dass die Dimension des Eingaberaumes nicht zwangsläufig der

²Der Zusatz „lernen“ hat hierbei seinen Ursprung im umfassenden Bereich des *maschinellen* Lernens, in den die bisher vorgestellten Verfahren aber auch die *Support Vector Regression* eingeordnet werden können.

³Der Begriff *Kernel* hat seinen Ursprung in der Integral Operator Theorie (Cristianini et al., 2000).

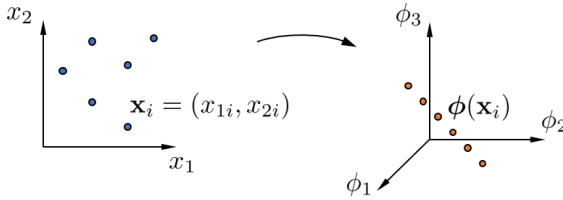


Abbildung 5.1: Unstrukturierte Datenpunkte $\mathbf{x}_i = (x_{1i}, x_{2i})$ (links); Transformierte Datenpunkte $\phi(\mathbf{x}_i)$ im Feature Space F (rechts)

des Merkmal-Raumes entsprechen muss. Eine Transformation in den Merkmal-Raum kann wie folgt angegeben werden

$$\mathbf{x} = [x_1 \cdots x_m] \mapsto \phi(\mathbf{x}) = [\phi_1(\mathbf{x}) \cdots \phi_M(\mathbf{x})]. \quad (5.14)$$

In den bereits vorgestellten Regressionsverfahren nehmen die *inneren* Produkte $\langle \cdot, \cdot \rangle$ eine entscheidende Betreuung ein, was durch die *Gramsche* Matrix \mathbf{G} in Gleichung (5.12) und den Vektor \mathbf{z} in Gleichung (5.13) deutlich wird. Der entscheidende Schritt ist, diese inneren Produkte im Merkmal-Raum zu bilden und dann indirekt über passende *Kernel* Funktionen auszudrücken

$$\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = K(\mathbf{x}_i, \mathbf{x}_j). \quad (5.15)$$

Die Einführung einer Kernel Funktion für die inneren Produkte bedeutet die Einführung einer anderen *Metrik* (Cristianini et al., 2000). Die *Kernel* Funktionen müssen gewissen Eigenschaften genügen, im Wesentlichen ist hierbei das *Mercer's* Theorem zu nennen, das darüber Auskunft gibt, ob eine Funktion eine gültige Kernel Funktion darstellt. Hierauf soll an dieser Stelle nicht weiter eingegangen werden, da im Rahmen dieser Arbeit keine neuen *Kernel* Funktionen entwickelt werden, sondern vielmehr effektive und bereits erprobte *Kernel* Funktionen für die Problemstellungen der Elasto-Plastizität herangezogen werden. Es sei nur soviel erwähnt, dass die Größen $K(\mathbf{x}_i, \mathbf{x}_j)$ in einer *Kernel* Matrix \mathbf{K} zusammengefasst werden und diese in jedem Fall *positiv semidefinit* ist und somit keine negativen Eigenwerte besitzt (Cristianini et al., 2000). Mit der *Kernel* Matrix \mathbf{K} kann die vorgestellte *Ridge* Regression im Merkmal-Raum F ausgeführt werden, was zur Bezeichnung *Kernel Ridge* Regression führt. Dieses Verfahren ist äquivalent zu dem aus der Statistik bekannten sogenannten *Gaußprozess*-Regression (Cristianini und Shawe-Taylor, 2007). Die Koeffizienten aus Gleichung (5.12) werden nun mit der *Kernel* Matrix folgendermaßen berechnet

$$\boldsymbol{\alpha} = 2\lambda (\lambda \mathbf{I} + \mathbf{K})^{-1} \mathbf{y}. \quad (5.16)$$

Das reduzierte Modell kann im Fall der *Kernel Ridge* Regression wie folgt dargestellt werden,

$$f(\mathbf{x}) = \sum_{i=1}^n \alpha_i k_i = \boldsymbol{\alpha}^T \mathbf{k}, \quad (5.17)$$

wobei die $k_i = K(\mathbf{x}_i, \mathbf{x})$ in dem Vektor $\mathbf{k} = [k_1 \cdots k_n]^T$ zusammengefasst werden.

Die zu wählende *Kernel* Funktion ist von der Art der Problemstellung und der Datenstruktur abhängig. Im Kapitel 6 werden verschiedene *Kernel* Funktionen vorgestellt und für die Anwendungen im Bereich der Elasto-Plastizität erprobt.

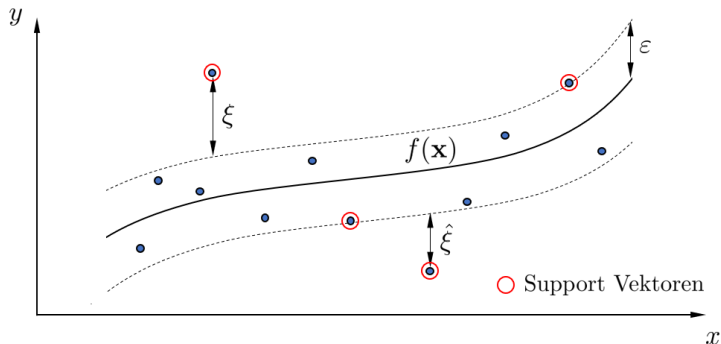


Abbildung 5.2: Nichtlineare Support Vector Regression: ε -„Schlauch“; $\xi, \hat{\xi}$ Slack Variablen; Regressionsfunktion $f(\mathbf{x})$

Formulierung der Support Vector Regression

Nach Einführung der grundlegenden Regressionsverfahren und ihrer Überführung in den Merkmal-Raum mittels *Kernel* Funktionen soll nun das Konzept der *Support Vector Regression* (SVR) vorgestellt werden. Ziel ist es, eine optimale Regressionsfunktion der Form

$$f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b \quad (5.18)$$

zu finden. Hierfür wird eine sogenannte ε -intensive Verlustfunktion eingeführt (Cristianini et al., 2000; Shawe-Taylor et al., 2004)

$$\begin{aligned} \text{linear:} \quad \mathcal{L}^\varepsilon(\mathbf{x}, y, f(\mathbf{x})) &= |y - f(\mathbf{x})|_\varepsilon = \max(0, |y - f(\mathbf{x})| - \varepsilon) \\ \text{quadratisch:} \quad \mathcal{L}_2^\varepsilon(\mathbf{x}, y, f(\mathbf{x})) &= |y - f(\mathbf{x})|_\varepsilon^2, \end{aligned} \quad (5.19)$$

wobei zwischen *linearer* und *quadratischer* Verlustfunktion unterschieden wird. Die vom Benutzer vorzugebende Größe ε ist in Abbildung 5.2 dargestellt und kann als eine Art „Schlauch“ um die zu findende Regressionsfunktion $f(\mathbf{x})$ angesehen werden. In der angelsächsischen Literatur wird auch von der *Intensive Tube Width* gesprochen, was soviel bedeutet wie „intensive Schlauchweite“. Als weitere relevante Größe werden die sogenannten *Slack* Variablen⁴ eingeführt. Die *Slack* Variablen ξ_i und $\hat{\xi}_i$ geben an, wie weit der Wert y_i eines Trainingspunktes \mathbf{x}_i über bzw. unter dem ε -„Schlauch“ liegt. Befindet sich beispielsweise ein Punkt y_i oberhalb des ε -„Schlauches“ so ist gleichzeitig $\hat{\xi}_i = 0$, sodass immer gilt $\xi_i \hat{\xi}_i = 0$. Liegt ein Punkt innerhalb des ε -Bereichs, so gilt $\xi_i = \hat{\xi}_i = 0$. Die *Slack* Variablen für einen Trainingspunkt (\mathbf{x}_i, y_i) sind somit äquivalent zur Auswertung der ε -intensive Verlustfunktion

$$\xi_i + \hat{\xi}_i = \mathcal{L}^\varepsilon(\mathbf{x}_i, y_i, f(\mathbf{x}_i)). \quad (5.20)$$

Die *Slack* Variablen zu den Trainingspunkten können in den Matrizen $\boldsymbol{\xi} = [\xi_1 \cdots \xi_n]^T$ und $\hat{\boldsymbol{\xi}} = [\hat{\xi}_1 \cdots \hat{\xi}_n]^T$ zusammengefasst werden. Neben der vom Anwender festzulegenden Größe ε wird eine weitere Größe C zur Steuerung der Regressionsqualität eingeführt, wobei $C > 0$

⁴Zu deutsch so viel wie „Schlupfvariablen“. Im Rahmen dieser Arbeit wird die englische Bezeichnung verwendet.

gilt (Jiang et al., 2020). Der Parameter C wird in der angelsächsischen Literatur auch als *Penalty Term* oder *Box Constraint* bezeichnet und stellt eine Art „Ausgleichsparameter“ dar, der die Komplexität der Trainingsdatenstruktur und die „Verluste“ in Form der außerhalb des ε -„Schlauches“ liegenden Punkte in Betracht zieht (Cristianini et al., 2000). Mit den eingeführten „freien“ Variablen und den gebunden bzw. festzulegenden „Steuer“-Größen kann die Optimierungsaufgabe für die *Support Vektor Regression* formuliert werden. Hierfür sind sowohl die mit C gewichteten ε -intensiven Verluste $\mathcal{L}^\varepsilon(\mathbf{x}_i, y_i, f(\mathbf{x}_i))$ aus Gleichung (5.19) zu minimieren als auch die Norm des Lösungsvektors $\|\mathbf{w}\|^2 = \langle \mathbf{w}, \mathbf{w} \rangle$. In der *primären* Form kann somit für die *lineare* Verlustfunktion das folgende Optimierungsproblem aufgestellt werden (Shawe-Taylor et al., 2004; Cristianini et al., 2000),

$$\min_{\mathbf{w}, b, \hat{\xi}, \xi} \left(\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n (\xi_i + \hat{\xi}_i) \right), \quad (5.21)$$

wobei der Zusammenhang aus Gleichung (5.20) für die *Slack* Variablen berücksichtigt wird. Für das Minimierungsproblem (5.21) müssen folgende Nebenbedingungen eingehalten werden

$$\begin{aligned} \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b - y_i &\leq \varepsilon + \xi_i, \quad i = 1 \dots n \\ y_i - \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle - b &\leq \varepsilon + \hat{\xi}_i, \quad i = 1 \dots n \\ \hat{\xi}_i &\geq 0, \quad i = 1 \dots n \\ \xi_i &\geq 0, \quad i = 1 \dots n. \end{aligned} \quad (5.22)$$

Für die Lösung von Optimierungsproblemen mit Nebenbedingungen in Form von Ungleichungen werden in der *Lagrange*-Theorie sogenannte *Lagrange*-Multiplikatoren eingeführt (Cristianini et al., 2000). In diesem Fall sind die Nebenbedingungen (5.22) mit den Multiplikatoren α_i , $\hat{\alpha}_i$, β_i und $\hat{\beta}_i$ zu versehen. Die sogenannte *Lagrange*-Funktion kann dann folgendermaßen aufgestellt werden (Jiang et al., 2020; Smola und Schölkopf, 2004)

$$\begin{aligned} L = & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n (\xi_i + \hat{\xi}_i) - \sum_{i=1}^n \alpha_i (\varepsilon + \xi_i - \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle - b + y_i) \\ & - \sum_{i=1}^n \hat{\alpha}_i (\varepsilon + \hat{\xi}_i - y_i + \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b) - \sum_{i=1}^n (\beta_i \xi_i + \hat{\beta}_i \hat{\xi}_i). \end{aligned} \quad (5.23)$$

Im nächsten Schritt sind die partiellen Ableitungen der *Lagrange*-Funktion nach \mathbf{w} , b , ξ_i und $\hat{\xi}_i$ „gleich Null“ zu setzen. Für die partielle Ableitung nach \mathbf{w} ergibt dies

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^n (\hat{\alpha}_i - \alpha_i) \phi(\mathbf{x}_i) = \mathbf{0}, \quad (5.24)$$

woraus ersichtlich wird, dass die *Lagrange*-Multiplikatoren α_i und $\hat{\alpha}_i$ die *dualen* Variablen darstellen. Nach Umformung der Gleichung (5.24) kann die *primäre* Variable \mathbf{w} folgendermaßen mit den *dualen* Variablen dargestellt werden (Fasshauer und McCourt, 2015),

$$\mathbf{w} = \sum_{i=1}^n (\hat{\alpha}_i - \alpha_i) \phi(\mathbf{x}_i) = \sum_{i=1}^n \tilde{\alpha}_i \phi(\mathbf{x}_i), \quad (5.25)$$

mit $\tilde{\alpha}_i = \hat{\alpha}_i - \alpha_i$. Die *dualen* Variablen können in den Matrizen $\boldsymbol{\alpha} = [\alpha_1 \dots \alpha_n]^T$ und $\hat{\boldsymbol{\alpha}} = [\hat{\alpha}_1 \dots \hat{\alpha}_n]^T$ zusammengefasst werden, womit sich für die spätere Ergebnisdarstellung ein kombinierter Lösungsvektor der Form

$$\tilde{\boldsymbol{\alpha}} = \hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha} \quad (5.26)$$

angeben lässt. Die weiteren partiellen Ableitungen der *Lagrange*-Funktion (5.23) ergeben

$$\begin{aligned}\frac{\partial L}{\partial b} &= \sum_{i=1}^n (\alpha_i - \hat{\alpha}_i) = 0 \\ \frac{\partial L}{\partial \xi_i} &= C - \alpha_i - \beta_i = 0 \\ \frac{\partial L}{\partial \hat{\xi}_i} &= C - \hat{\alpha}_i - \hat{\beta}_i = 0.\end{aligned}\quad (5.27)$$

Die Rücksubstitution der Gleichung (5.24) bzw. (5.25) und der Gleichungen (5.27) in die *Lagrange*-Funktion (5.23) führt auf die sogenannte *duale Lagrange*-Funktion (Smola und Schölkopf, 2004; Jiang et al., 2020; Cristianini et al., 2000; Shawe-Taylor et al., 2004)

$$L = \sum_{i=1}^n (\hat{\alpha}_i - \alpha_i) y_i - \varepsilon \sum_{i=1}^n (\hat{\alpha}_i + \alpha_i) - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\hat{\alpha}_i - \alpha_i) (\hat{\alpha}_j - \alpha_j) K(\mathbf{x}_i, \mathbf{x}_j). \quad (5.28)$$

Die *Lagrange*-Multiplikatoren β_i und $\hat{\beta}_i$ können dabei über die beiden letzten Gleichungen (5.27) eliminiert werden. Zusätzlich werden die Zusammenhänge (Cristianini et al., 2000)

$$\begin{aligned}\|\mathbf{w}\|^2 = \langle \mathbf{w}, \mathbf{w} \rangle &= \left\langle \left(\sum_{i=1}^n (\hat{\alpha}_i - \alpha_i) \phi(\mathbf{x}_i) \right), \left(\sum_{i=1}^n (\hat{\alpha}_i - \alpha_i) \phi(\mathbf{x}_i) \right) \right\rangle \\ &= \sum_{i=1}^n \sum_{j=1}^n (\hat{\alpha}_i - \alpha_i) (\hat{\alpha}_j - \alpha_j) K(\mathbf{x}_i, \mathbf{x}_j)\end{aligned}\quad (5.29)$$

und (Fasshauer und McCourt, 2015)

$$\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle = \sum_{j=1}^n (\hat{\alpha}_j - \alpha_j) K(\mathbf{x}_j, \mathbf{x}_i). \quad (5.30)$$

berücksichtigt. Im Sinne der *Lagrange*-Theorie ist bei der *dualen* Form des Optimierungsproblems ein Maximum in den *dualen* Variablen bzw. *Lagrange*-Multiplikatoren $\boldsymbol{\alpha}$ und $\hat{\boldsymbol{\alpha}}$ zu suchen. Als Erweiterung zu den schon vorgestellten Regressionsverfahren handelt es sich bei der *dualen* Form um eine Optimierungsaufgabe mit *Nebenbedingungen* und zusätzlichen *Komplementaritätsbedingungen*. Letztere steuern die Lösungsfindung in Form der sogenannten *Karush-Kuhn-Tucker* Bedingungen. Das *duale* Optimierungsproblem zu dem *primären* Optimierungsproblem (5.21) kann nun wie folgt angegeben werden,

$$\begin{aligned}\max_{\hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha}} \quad & L(\hat{\boldsymbol{\alpha}}, \boldsymbol{\alpha}) \\ L = \sum_{i=1}^n (\hat{\alpha}_i - \alpha_i) y_i - \varepsilon \sum_{i=1}^n (\hat{\alpha}_i + \alpha_i) - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\hat{\alpha}_i - \alpha_i) (\hat{\alpha}_j - \alpha_j) K(\mathbf{x}_i, \mathbf{x}_j),\end{aligned}\quad (5.31)$$

unter Berücksichtigung der *Nebenbedingungen*

$$\begin{aligned}0 \leq \hat{\alpha}_i, \alpha_i \leq C, \quad & i = 1 \dots n \\ \sum_{i=1}^n (\hat{\alpha}_i - \alpha_i) &= 0,\end{aligned}\quad (5.32)$$

sowie den *Karush-Kuhn-Tucker* Komplementaritätsbedingungen

$$\begin{aligned}
 \alpha_i (\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b - y_i - \varepsilon - \xi_i) &= 0, \quad i = 1 \cdots n \\
 \hat{\alpha}_i \left(y_i - \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle - b - \varepsilon - \hat{\xi}_i \right) &= 0, \quad i = 1 \cdots n \\
 \alpha_i \hat{\alpha}_i &= 0, \quad i = 1 \cdots n \\
 \xi_i \hat{\xi}_i &= 0, \quad i = 1 \cdots n \\
 (\alpha_i - C) \xi_i &= 0, \quad i = 1 \cdots n \\
 (\hat{\alpha}_i - C) \hat{\xi}_i &= 0, \quad i = 1 \cdots n.
 \end{aligned} \tag{5.33}$$

Hierbei können die *inneren* Produkte in den ersten beiden Gleichungen (5.33) mithilfe der *Kernel* Funktion entsprechend Gleichung (5.30) ausgedrückt werden. Für die *quadratische* Verlustfunktion aus Gleichung (5.19) kann das entsprechende *duale* Optimierungsproblem⁵ wie folgt angegeben werden,

$$\begin{aligned}
 &\max_{\hat{\alpha}, \alpha} L(\hat{\alpha}, \alpha) \\
 L &= \sum_{i=1}^n (\hat{\alpha}_i - \alpha_i) y_i - \varepsilon \sum_{i=1}^n (\hat{\alpha}_i - \alpha_i) - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\hat{\alpha}_i - \alpha_i) (\hat{\alpha}_j - \alpha_j) \left(K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{C} \delta_{ij} \right),
 \end{aligned} \tag{5.34}$$

mit dem *Kronecker* Delta δ_{ij} . Die *Nebenbedingungen* werden in diesem Fall wie folgt angegeben

$$\begin{aligned}
 \hat{\alpha}_i \geq \alpha_i \geq 0, \quad i &= 1 \cdots n \\
 \sum_{i=1}^n (\hat{\alpha}_i - \alpha_i) &= 0.
 \end{aligned} \tag{5.35}$$

Die *Karush-Kuhn-Tucker* Komplementaritätsbedingungen lauten für den *quadratischen* Fall

$$\begin{aligned}
 \alpha_i (\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + b - y_i - \varepsilon - \xi_i) &= 0, \quad i = 1 \cdots n \\
 \hat{\alpha}_i \left(y_i - \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle - b - \varepsilon - \hat{\xi}_i \right) &= 0, \quad i = 1 \cdots n \\
 \alpha_i \hat{\alpha}_i &= 0, \quad i = 1 \cdots n \\
 \xi_i \hat{\xi}_i &= 0, \quad i = 1 \cdots n.
 \end{aligned} \tag{5.36}$$

Die Lösung der Optimierungsprobleme (5.31) und (5.34) resultiert in den *optimalen* Werten für die *dualen* Variablen $\alpha = [\alpha_1 \cdots \alpha_n]^T$ und $\hat{\alpha} = [\hat{\alpha}_1 \cdots \hat{\alpha}_n]^T$ sowie in den optimalen Wert für b . Der Wert b wird auch als *Bias*-Term bezeichnet und berechnet sich im Rahmen der *Support Vector Regression* über die *Karush-Kuhn-Tucker* Komplementaritätsbedingungen (Fasshauer und McCourt, 2015). Für b gilt

$$\begin{aligned}
 b &= y_i - \sum_{j=1}^n (\hat{\alpha}_j - \alpha_j) K(\mathbf{x}_i, \mathbf{x}_j) - \varepsilon \quad \text{für } \alpha_i \in (0, C) \\
 b &= y_i - \sum_{j=1}^n (\hat{\alpha}_j - \alpha_j) K(\mathbf{x}_i, \mathbf{x}_j) + \varepsilon \quad \text{für } \hat{\alpha}_i \in (0, C).
 \end{aligned} \tag{5.37}$$

⁵Auf eine Herleitung wird an dieser Stelle verzichtet. Hierfür sei auf die eingangs aufgeführten Quellen verwiesen.

Die geschlossene Lösung für die *Support Vector Regression* kann damit folgendermaßen angegeben werden,

$$f(\mathbf{x}) = \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b = \sum_{i=1}^n \tilde{\alpha}_i K(\mathbf{x}_i, \mathbf{x}) + b = \tilde{\boldsymbol{\alpha}}^T \mathbf{k} + b, \quad (5.38)$$

mit $\tilde{\alpha}_i = \hat{\alpha}_i - \alpha_i$ und $\tilde{\boldsymbol{\alpha}} = [\tilde{\alpha}_1 \cdots \tilde{\alpha}_n]^T$ sowie $\mathbf{k} = [k_1 \cdots k_n]^T$ mit $k_i = K(\mathbf{x}_i, \mathbf{x})$.

Die Punkte (\mathbf{x}_i, y_i) , die nicht *strikt* innerhalb des ε -„Schlauches“ liegen werden *Support Vektoren* genannt (siehe hierzu Abbildung 5.2). Die entsprechenden *dualen* Variablen der *Support Vektoren* erfüllen die Bedingung $\hat{\alpha}_i - \alpha_i \neq 0$ und bestimmen somit maßgebend das Lösungsverhalten der *Support Vector Regression*, siehe hierzu Gleichung (5.25) und (5.38). Die Punkte, die komplett außerhalb des ε -„Schlauches“ liegen werden *gebundene* Support Vektoren genannt und solche, die diesen lediglich berühren werden als *ungebundene* Support Vektoren bezeichnet. Bei den *gebundenen* Support Vektoren nehmen die *dualen* Variablen $\hat{\alpha}_i$ und α_i den Wert C an. Bei den *ungebundenen* Support Vektoren ergibt sich ein Wert zwischen 0 und C . Zu beachten ist, dass $\hat{\alpha}_i$ und α_i nicht gleichzeitig ungleich 0 sein können. (Fasshauer und McCourt, 2015; Moustapha et al., 2018; Cristianini et al., 2000)

Mit dem vorgestellten Modellreduktionsverfahren der *Support Vector Regression* (SVR) werden in den Kapiteln 6 und 7 Untersuchungen im Rahmen der Elasto-Plastizität durchgeführt. Das Entscheidende ist dabei die Wahl der *Kernel* Funktion $K(\mathbf{x}_i, \mathbf{x}_j)$ sowie der Parameter ε und C , die auch als *Hyperparameter* bezeichnet werden (Moustapha et al., 2018). Es wird gezeigt, dass mit einer passenden Auswahl dieser Größen ein effizienter Einsatz im Anwendungsbereich der Elasto-Plastizität möglich ist.

5.2 Standardisieren der Eingabedaten

Im Kapitel 2.2, Gleichung (2.15) wird eine Zentrierung der *Kernel* Matrix vorgestellt, die den „Datenursprung“ in den „Datenschwerpunkt“ verschiebt. Eine weitere Möglichkeit die Eingabedaten aufzuarbeiten ist die *Standardisierung*. Vielfach liegen die Daten x_{ij} in $\mathbf{x}_i = [x_{i1} \cdots x_{im}]$, $i = 1 \cdots n$ in unterschiedlichen Einheiten vor. Dann ist es zweckmäßig, diese vor der Modellreduktion zu standardisieren. In Shawe-Taylor et al. (2004) wird eine Standardisierung für die j -te Variable folgendermaßen vorgeschlagen

$$\begin{aligned} \hat{x}_{ij} &= \frac{x_{ij} - \mu_{xj}}{\sigma_{xj}} \\ \mu_{xj} &= \frac{\sum_{i=1}^n x_{ij}}{n} \\ \sigma_{xj} &= \sqrt{\left(\sum_{i=1}^n x_{ij} - \mu_{xj} \right)^2}. \end{aligned} \quad (5.39)$$

Nach der Standardisierung liegen die Werte für die Variablen $\hat{x}_{ij} \in (0, 1)$, $i = 1 \cdots n$, $j = 1 \cdots m$, im Einheitsintervall vor.

5.3 Nicht glatte Antwortspektren

Im Rahmen mechanischer Anwendungen kann das Antwortverhalten der *Quantity of Interest* nichtlinear, unstetig, sprunghaft oder auch bereichsweise eine Konstante sein. Die generellen Strukturen des Antwortverhaltens, die im Rahmen dieser Arbeit auftreten, sind in Abbildung 5.3 vereinfacht für den skalaren Fall $y = f(x)$ dargestellt. Im Rahmen einer Modellreduktion sind solche Verläufe schwer zu approximieren, da gerade „Ecken“ oder „konstante“ Verläufe

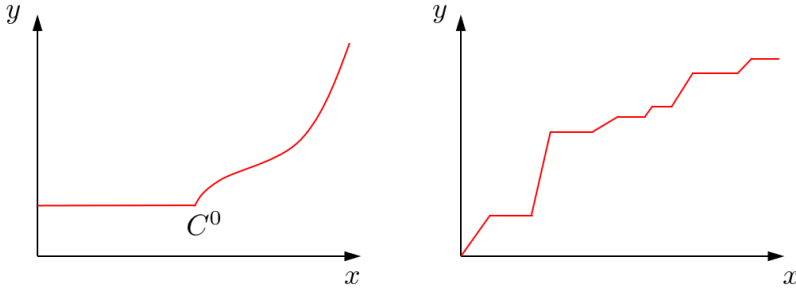


Abbildung 5.3: Antwortverhalten $y = f(x)$: konstanter Bereich, der C^0 -stetig in einen nicht-linearen Bereich übergeht (links); Verlauf stückweise linear und stückweise Konstant(rechts)

zu Problemen führen. Wird beispielsweise eine Approximation mittels Polynom-Superposition in Betracht gezogen, so kann es bei einer zu großen Anzahl Stützstellen in einem *kritischen* Bereich im restlichen Bereich zu großen *Oszillationen* kommen. Generell sind solche Verläufe schwer *global*, also über den gesamten Datenbereich mit nur einem reduzierten Modell zu approximieren. In Moustapha und Sudret (2019) wird ein Ansatz vorgeschlagen, bei dem vorab die einzelnen, für sich stetigen Bereiche mittels einer Klassifikation getrennt werden. Im Anschluss werden separate, reduzierte Modelle für die Bereiche erstellt. Das Vorgehen wird in Moustapha und Sudret (2019) unter anderem auf ein *Zwei-Stab-Durchschlagproblem* der Statik angewandt. Im Rahmen dieser Arbeit soll ein *global* reduziertes Modell erstellt werden. Der Vorteil ist die Vermeidung einer vorab durchgeführten Klassifizierung, was unter Umständen dann einen Einfluss auf die Genauigkeit der Approximation hat. In den späteren Kapiteln wird gezeigt, dass bei der geeigneten Wahl der Kernel Funktion und der Parameter für die hier herangezogenen Anwendungen im Bereich der Elasto-Plastizität auch mit *global* reduzierten Modellen gute Ergebnisse erzielt werden können.

5.4 Sampling Methoden

Unter *Sampling Methoden* ist die Vorgehensweise zu verstehen, mit der die Trainingsdatenpunkte (\mathbf{x}_i, y_i) für die Modellreduktion in dem Gebiet von Interesse (*engl. Domain*) verteilt werden. In Jiang et al. (2020) wird zwischen *One-Shot* Sampling Methoden und *adaptive* Sampling Methoden unterschieden. Bei den *One-Shot* Sampling Methoden werden die Trainingspunkte *einmalig* vor der Modellerstellung in dem betrachteten Gebiet verteilt. Bei den *adaptive* Sampling Methoden werden ausgehend von einer „Initialverteilung“ der Trainingspunkte sukzessiv Punkte in bestimmten Bereichen des Gebietes hinzugefügt, um die Approximationsgenauigkeit des reduzierten Modells zu erhöhen. *Sampling Methoden* stellen im Bereich der Modellreduktion nach wie vor einen Forschungsschwerpunkt dar. Neue Ansätze zu *adaptive* Sampling Methoden werden in den Arbeiten von Garud et al. (2016), Steiner et al. (2019), Xiang et al. (2017), Zhang et al. (2020), Mandelli und Smith (2012), Xu et al. (2014), Jin et al. (2002) und Garbo (2019) aufgeführt, um an dieser Stelle nur einige zu nennen. In Fuhg et al. (2020) wird ein umfangreicher Überblick über *adaptive* Sampling Methoden gegeben.

Bei den *One-Shot* Sampling Methoden ist ein klassischer und einfach zu implementierender Ansatz das *kartesische* Gitter. Dabei werden die Punkte äquidistant über die Gebiets-Koordinaten verteilt und dann im *kartesischen Produkt* kombiniert. Ein zweidimensionales *kartesisches* Gitter für $n = 1000$ Trainingspunkte ist in der linken Abbildung 5.4 dargestellt. Ein *kartesisches* Gitter kann auch in *anisotroper* Form erstellt werden, beispielsweise wenn die Anzahl der Da-

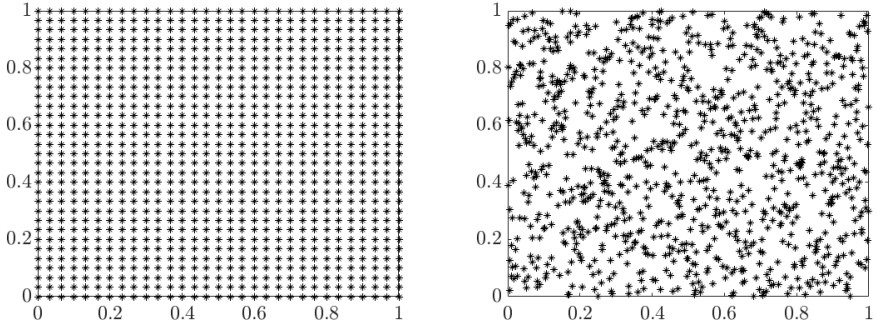


Abbildung 5.4: *One-Shot Sampling Methoden für den zweidimensionalen Fall mit $n = 1000$ Trainingspunkten im Einheitsquadrat: Kartesisches Gitter (links); Latin Hypercube (rechts)*

tenpunkte in den Koordinatenrichtungen unterschiedlich ist. Dies kann nützlich für Modelle sein, bei denen die Sensitivität in den Variablen unterschiedlich ist.

Eine weitere Möglichkeit ist die Verteilung der Punkte im sogenannten *Latin Hypercube*⁶ (Jiang et al., 2020). In der rechten Abbildung 5.4 ist ein *Latin Hypercube Sampling* (LHS) für $n = 1000$ Trainingspunkte dargestellt. Zuerst wird ein m -dimensionales Einheitsgebiet⁷ für die m Variablen aus $\mathbf{x}_i = [x_{i1} \cdots x_{im}]$ erstellt $(0, 1)^m$ (Jiang et al., 2020). Im nächsten Schritt werden für die n Trainingspunkte n *Hyperwürfel* mit einer Kantenlänge von $\frac{1}{n}$ erstellt. Nach Forrester et al. (2008) wird jetzt in jedem der n *Hyperwürfel* ein Trainingspunkt so verteilt, dass er bei seiner Projektion in jede der m Koordinatenrichtungen auf keinen weiteren Trainingspunkt in einem andern *Hyperwürfel* trifft. Die Verteilung in einem *Hyperwürfel* ist dabei zufällig, muss aber die genannte Bedingung „Einzigartigkeit“ in den m Koordinatenrichtungen erfüllen.

Weitere Sampling Verfahren sind beispielsweise das *Monte-Carlo* Sampling, das *quasi Monte-Carlo* Sampling (z.B. *Sobol-Sequenz*) sowie die Verteilung der Trainingspunkte in Form der *Gauß-Legrende* Quadraturstellen, um nur einige zu nennen. Im Rahmen dieser Arbeit werden für das *One-Shot* Sampling *kartesische* Gitter und *Latin Hypercubes* verwendet, da es in erster Linie um die Erprobung der *Support Vector Regression* im Bereich der Elasto-Plastizität geht. In Kapitel 6 werden zudem Techniken des *adaptiven* Samplings vorgestellt und erprobt.

⁶Auf deutsch „lateinische Hyperwürfel Verteilung“. Im Rahmen dieser Arbeit wird einheitlich die Bezeichnung aus der angelsächsischen Literatur verwendet.

⁷Nach dem Sampling kann eine Rücktransformation von dem Einheitsgebiet auf das Ursprungsgebiet stattfinden.

6 Anwendung in der 1D-Elasto-Plastizität

Die in diesem Kapitel vorgestellten Studien stellen einen neuen Anwendungsfall des Autors für die *Support Vector Regression* (SVR) dar. Die SVR wird auf das rein phänomenologische, eindimensionale elasto-plastische Materialverhalten angewendet. Ein Eingabedatensatz für das Modell besteht dabei aus der Gesamtdehnung, der Fließgrenze und einem Verfestigungsparameter. Das trainierte Modell soll dann in der Lage sein, die zugehörige plastische Dehnung, in diesem Fall die *quantity of interest*, vorauszusagen bzw. zu approximieren. Die Berechnungen finden mit der *Statistics and Machine Learning Toolbox* in MATLAB[®] statt.

6.1 Modellaufbau

Der rein phänomenologische und eindimensionale Modellaufbau unterliegt hier keinem Finite Elemente Modell (FE-Modell), sodass direkt mit *skalarwertigen* Spannungen und Dehnungen gearbeitet werden kann. Die in den Kapiteln 3 und 4 im Rahmen der Kontinuumsmechanik bzw. der Finite Elemente Methode beschriebene Vorgehensweise beinhaltet den allgemeinen dreidimensionalen Fall. Der Weg über die Berechnung des Verschiebungsfeldes ist somit im Folgenden nicht nötig, zumal kein FE-Modell zugrunde gelegt wird. Viele der vorgestellten Konzepte vereinfachen sich demnach, sodass in Anlehnung an die Veröffentlichung von Yaw (2012) die wesentlichen Schritte der *eindimensionalen* Elasto-Plastizität noch einmal vorgestellt werden.

Im Rahmen dieser Arbeit wird ein lineares Verfestigungsgesetz vorausgesetzt. Der zugehörige Verfestigungsparameter wird mit k bezeichnet. Ausgehend von einer bekannten Gesamtdehnung ${}^n\varepsilon$, einer plastischen Dehnung ${}^n\varepsilon^p$ und einer internen Variable ${}^n\alpha$ zum Prozessschritt n sollen die Größen ${}^{n+1}\varepsilon^p$, ${}^{n+1}\alpha$ und ${}^{n+1}\sigma$ unter Vorgabe einer neuen Gesamtdehnung ${}^{n+1}\varepsilon$ berechnet werden. Dabei beschreibt ${}^{n+1}\sigma$ die resultierende Spannung im Schritt $n + 1$. Zuerst wird die elastische Testspannung ${}^{n+1}\sigma^{\text{test}}$ und die Test-Fließfunktion ${}^{n+1}f^{\text{test}}$ berechnet,

$$\begin{aligned} {}^{n+1}\sigma^{\text{test}} &= E ({}^{n+1}\varepsilon - {}^n\varepsilon^p) \\ {}^{n+1}f^{\text{test}} &= |{}^{n+1}\sigma^{\text{test}}| - (\sigma^y + k {}^n\alpha) , \end{aligned} \tag{6.1}$$

wobei σ^y die Fließgrenze kennzeichnet und E für den Elastizitätsmodul steht. Im nächsten Schritt muss anhand der Test-Fließfunktion ausgewertet werden, ob die Gesamtdehnung ${}^{n+1}\varepsilon$ ein plastisches Fließen zur Folge hat. Ist die Test-Fließfunktion kleiner oder gleich null, so ergeben sich die Größen für den Schritt $n + 1$ wie folgt zu

$$\begin{aligned} {}^{n+1}f^{\text{test}} &\leq 0 \\ {}^{n+1}\sigma &= {}^{n+1}\sigma^{\text{test}} \\ {}^{n+1}\varepsilon^p &= {}^n\varepsilon^p \\ {}^{n+1}\alpha &= {}^n\alpha , \end{aligned} \tag{6.2}$$

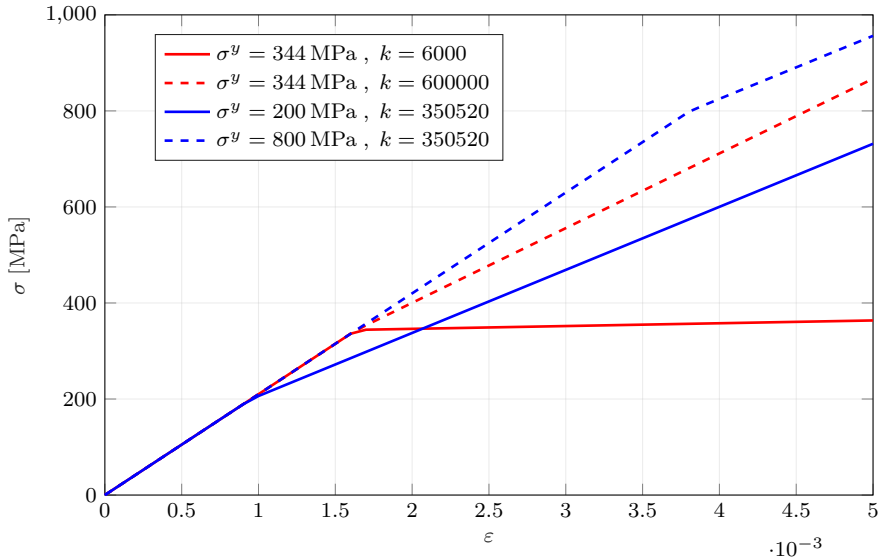


Abbildung 6.1: Spannungs-Dehnungs-Diagramme $\sigma(\varepsilon)$ der 1D-Elasto-Plastizität bei verschiedenen (σ^y, k) -Kombinationen

somit tritt kein plastisches Fließen ein. Ist die Test-Fließfunktion größer null, so setzt plastisches Fließen ein und die neuen Größen für den Schritt $n + 1$ ergeben sich wie folgt zu

$$\begin{aligned}
 {}^{n+1}f^{\text{test}} &> 0 \\
 \Delta\gamma &= \frac{{}^{n+1}f^{\text{test}}}{E + k} \\
 {}^{n+1}\sigma &= \left(1 - \frac{\Delta\gamma E}{|{}^{n+1}\sigma^{\text{test}}|}\right) {}^{n+1}\sigma^{\text{test}} \\
 {}^{n+1}\varepsilon^p &= {}^n\varepsilon^p + \Delta\gamma \operatorname{sign}({}^{n+1}\sigma^{\text{test}}) \\
 {}^{n+1}\alpha &= {}^n\alpha + \Delta\gamma,
 \end{aligned} \tag{6.3}$$

wobei $\operatorname{sign}(\cdot)$ die Vorzeichenfunktion kennzeichnet

$$\operatorname{sign}({}^{n+1}\sigma^{\text{test}}) \begin{cases} -1 & \text{if } {}^{n+1}\sigma^{\text{test}} < 0 \\ 0 & \text{if } {}^{n+1}\sigma^{\text{test}} = 0 \\ 1 & \text{if } {}^{n+1}\sigma^{\text{test}} > 0 \end{cases}. \tag{6.4}$$

Für den Elastizitätsmodul wird ein Wert von $E = 210\,000 \text{ N mm}^{-2}$ angesetzt. Die Fließgrenze σ^y und der Verfestigungsparameter k bilden zusammen mit der Gesamtdehnung ε den Eingabedatensatz für das zu reduzierende Modell. In der Abbildung 6.1 sind für verschiedene Kombinationen (σ^y, k) die zugehörigen Spannungs-Dehnungs-Diagramme dargestellt. Die roten Linien stehen für eine gleiche Fließgrenze, aber unterschiedliche Verfestigungsparameter. Bei den blauen Linien sind die Verfestigungsparameter gleich, aber die Fließgrenzen unterschiedlich.

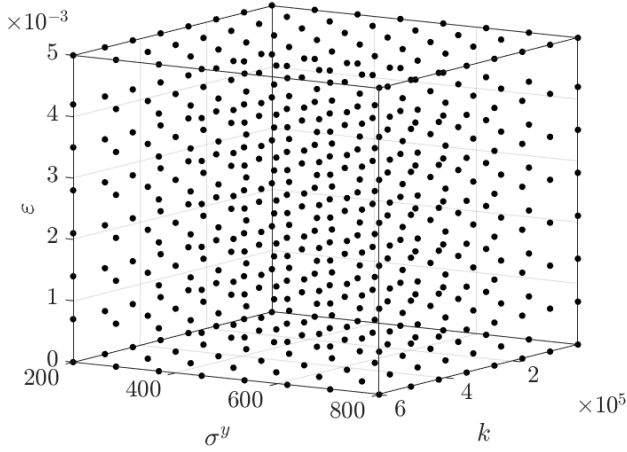


Abbildung 6.2: Gitter $\mathfrak{X}_{8 \times 8 \times 8}$ der Eingabedatensätze $\mathbf{x}_i = [\varepsilon_i, k_i, \sigma_i^y]_{i=1 \dots 512}$; σ^y in [MPa]

Im Zuge der Modellreduktion werden die Daten folgendermaßen zusammengefasst

$$\begin{aligned} \mathbf{X} &= [\mathbf{x}_1 \cdots \mathbf{x}_n]^T, \quad \mathbf{x}_i = [\varepsilon_i, k_i, \sigma_i^y]^T \\ \mathbf{Y} &= [y_1 \cdots y_n]^T, \quad y_i = \varepsilon_i^p. \end{aligned} \quad (6.5)$$

Mit einer hoch aufgelösten Berechnung \mathcal{F}^{HF} werden für die Eingabedatensätze \mathbf{x}_i die entsprechenden Antworten y_i des Systems berechnet

$$\begin{aligned} \mathcal{F}^{HF} : \mathbf{X} &\longrightarrow \mathbf{Y} \\ \mathcal{F}_{(\mathbf{x}_i)}^{HF} &= y_i. \end{aligned} \quad (6.6)$$

In den folgenden Untersuchungen wird der Bereich für die Eingabedatensätze $\mathbf{x}_i = [\varepsilon_i, k_i, \sigma_i^y]$ eingegrenzt. Die Bereichsgrenzen sind

$$\begin{aligned} 0 &\leq \varepsilon_i \leq 0,005 \\ 6000 &\leq k_i \leq 600\,000 \\ 200 \text{ MPa} &\leq \sigma_i^y \leq 800 \text{ MPa}. \end{aligned} \quad (6.7)$$

Zur Visualisierung seien beispielsweise jeweils 8 Werte der drei Eingabegrößen ε_i , k_i und σ_i^y gegeben ($i = 1, \dots, 8$). Die Punkte sind dabei äquidistant über die Intervalle in (6.7) verteilt. Die Datenpunkte werden so miteinander kombiniert, dass ein rechtwinkliges Gitter entsteht,

$$\begin{aligned} \mathfrak{X}_{8 \times 8 \times 8} &= \{\varepsilon_i\}_{i=1 \dots 8} \times \{k_i\}_{i=1 \dots 8} \times \{\sigma_i^y\}_{i=1 \dots 8} \\ &\times : \text{kartesisches Produkt}. \end{aligned} \quad (6.8)$$

Das Gitter $\mathfrak{X}_{8 \times 8 \times 8}$ ist in der Abbildung 6.2 dargestellt und besteht in diesem Fall aus $8^3 = 512$ Datenpunkten. Zu jedem der Punkte aus Gleichung (6.8) gibt es eine entsprechende plastische Dehnung $\varepsilon_i^p(\varepsilon_i, k_i, \sigma_i^y)_{i=1 \dots 512}$. Eine plastische Dehnung tritt erst ab der Grenze von

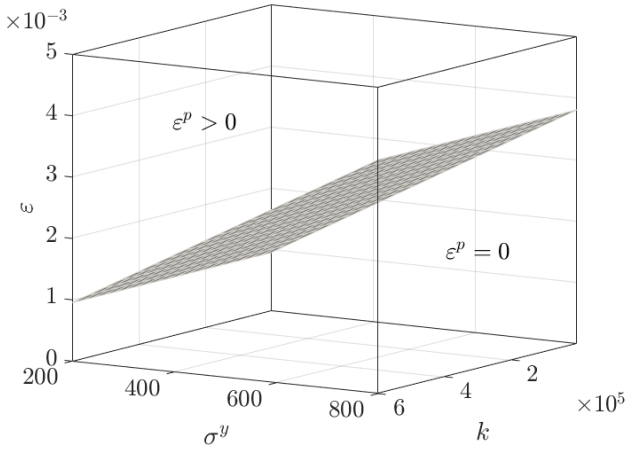


Abbildung 6.3: Grenze zwischen Datenpunkten mit plastischer Dehnung $\varepsilon^p > 0$ und keiner plastischen Dehnung $\varepsilon^p = 0$; σ^y in [MPa]

σ^y/E auf, sodass der untere Bereich im Gitter keine plastische Dehnung hervorruft und somit $\varepsilon^p(\varepsilon_i, k_i, \sigma_i^y) = 0$ ist. Die Grenze, bei der plastische Dehnung einsetzt, ist in Abbildung 6.3 dargestellt. Dieser Sachverhalt hat eine entscheidende Konsequenz für die Erstellung des reduzierten Modells. Es muss in der Lage sein, den hieraus resultierenden *nicht glatten* Verlauf von $\varepsilon^p(\varepsilon_i, k_i, \sigma_i^y)$ hinreichend abzubilden. Zur Visualisierung des nicht glatten Verlaufs dienen die Abbildungen 6.4 und 6.5. In der Abbildung 6.4 ist die oberste Lage aus dem Gitter $\mathfrak{X}_{8 \times 8 \times 8}$ mit einer Gesamtdehnung von $\varepsilon = 0,005$ dargestellt (schwarze Punkte). Die zu diesen Punkten zugehörigen plastischen Dehnungen sind mit roten Punkten dargestellt und zusätzlich zur besseren Illustration durch ein Flächengitter miteinander verbunden. Hierbei ist deutlich zu erkennen, dass ein nichtlinearer, hier aber noch glatter Verlauf in der plastischen Dehnung vorliegt. In der Abbildung 6.5 ist die Ebene des Gitters mit einer Gesamtdehnung von $\varepsilon = 0,0028$ aufgetragen, also ca. die Hälfte der maximalen Gesamtdehnung von $\varepsilon = 0,005$. Es ist deutlich zu erkennen, dass ab einer Grenze von $\sigma^y \geq 588 \text{ MPa} = 0,0028E$ ein konstanter Verlauf mit $\varepsilon^p = 0 := \text{const}$ vorliegt. In dem Bereich $\sigma^y \leq 588 \text{ MPa}$ liegt wieder ein glatter, nichtlinearer Verlauf vor, wobei der Übergang zwischen diesen beiden Bereichen scharfkantig ist.

6.2 Studien zur Kernel Funktion und Parametrisierung

Für die Erstellung eines Surrogate Modells mit Hilfe der Support Vector Regression muss die Kernel Funktion richtig gewählt werden. Darüber hinaus sind die in das SVR Modell eingehenden Parameter passend zu wählen. Die hierfür durchgeführten Studien werden im Folgenden vorgestellt.

Für die Untersuchungen zur passenden Kernel Funktion werden im ersten Schritt die drei „klas-

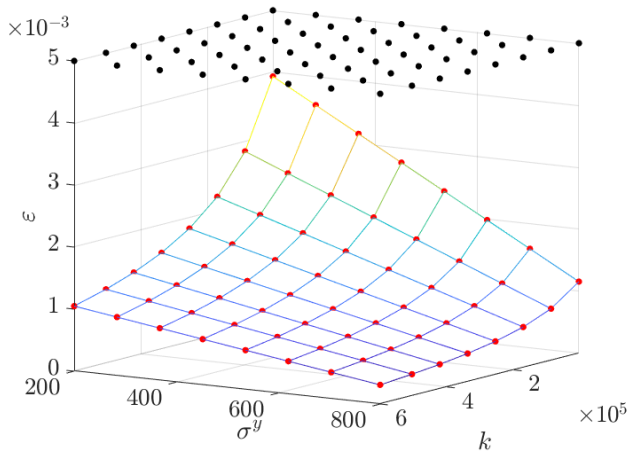


Abbildung 6.4: Gesamtdehnung (\bullet) $\varepsilon = 0,005$ mit zugehöriger plastischer Dehnung (\bullet) ε^p ; σ^y in [MPa]

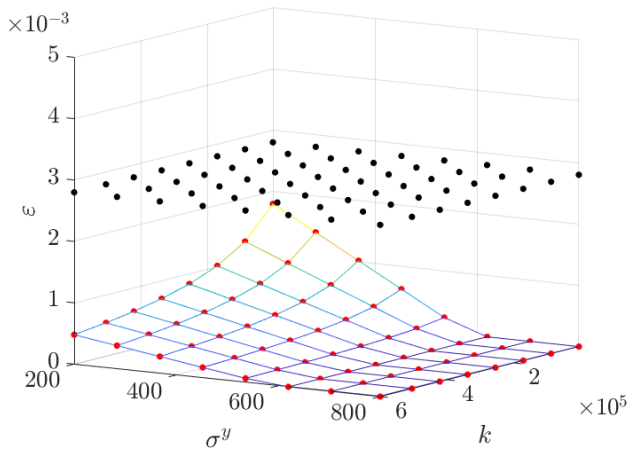


Abbildung 6.5: Gesamtdehnung (\bullet) $\varepsilon = 0,0028$ mit zugehöriger plastischer Dehnung (\bullet) ε^p ; σ^y in [MPa]

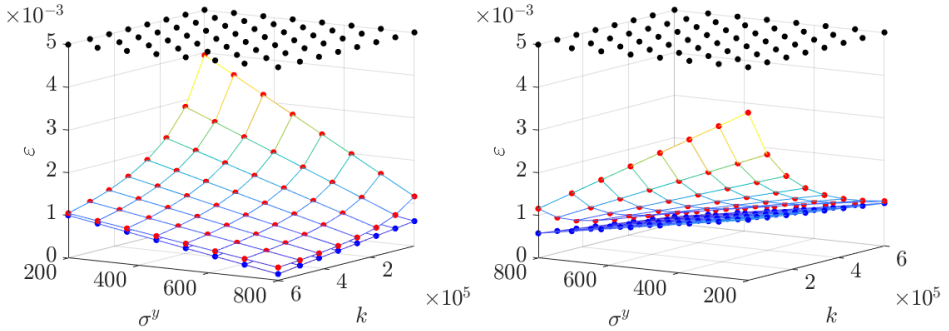


Abbildung 6.6: Gesamtdehnung (●) $\varepsilon = 0,005$ mit zugehöriger plastischer Dehnung (●) ε^P ; Approximation mit linearer Kernel Funktion (●) $\hat{\varepsilon}^P$; σ^y in [MPa]

sischen“ Kernel Funktion untersucht,

Linear	$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$	
Polynomial	$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^p$; p : Polynomgrad	(6.9)
Gauß	$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\ \mathbf{x}_i^T \mathbf{x}_j\ ^2)$,	

wobei $\mathbf{x}_i = [\varepsilon_i, k_i, \sigma_i^y]^T$ ist. Das Modell wird mit einem Gitter $\mathfrak{X}_{10 \times 10 \times 10}$ trainiert, also mit $10^3 = 1000$ Datenpunkten $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_{1000}]^T$. Die Daten werden vorab standardisiert¹. Um eine Aussage darüber zu treffen, wie gut sich die jeweiligen Kernel Funktionen an das Lösungsspektrum $\mathbf{Y} = [y_1 \cdots y_n]^T$ mit $y_i = \varepsilon_i^P$ anschmiegen, eignet sich die Betrachtung einer einzelnen Ebene bzw. „Fläche“ im Gitter. Die approximierte Ergebnis-Fläche sollte sich möglichst gut an die Soll-Fläche annähern. Zudem kann so auch ein mögliches Oszillieren der Lösung festgestellt werden. Als Test-Flächen werden zum einen die oberste Ebene im dünneren Gitter $\mathfrak{X}_{8 \times 8 \times 8}$ bei $\varepsilon = 0,005$ herangezogen. Zum Anders ist es zweckmäßig, die Ebene bei $\varepsilon = 0,0028$ zu betrachten, da hier noch die Schwierigkeit des nicht glatten Verlaufes im Antwortspektrum sichtbar wird.

In Abbildung 6.6 ist das Ergebnis für die lineare Kernel Funktion bei einer Gesamtdehnung von $\varepsilon = 0,005$ (schwarze Punkte) dargestellt. Für eine bessere Darstellung sind zwei verschiedene Perspektiven des Ergebnisses gezeigt. Die zugehörigen plastischen Soll-Dehnungen $y_i = \varepsilon_i^P$ sind wieder mit roten Punkten markiert und die approximierte Lösung $\hat{y}_i = \hat{\varepsilon}_i^P$ zur besseren Unterscheidung mit blauen Markierungspunkten. Um die Ergebnisse zu interpretieren, ist wieder jeweils ein Flächengitter mit dargestellt. Es ist gut zu erkennen, dass die lineare Kernel Funktion nur einen linearen Verlauf abbilden kann. Im Bereich größerer Gradienten löst sich die Approximation von der hoch aufgelösten Berechnung bzw. Soll-Fläche deutlich ab. Bei einem hohen Verfestigungsparameter $k > 400\,000$ stimmen die Verläufe noch qualitativ gut überein, wobei die Approximation die plastische Dehnung leicht unterschätzt. Die Ergebnisse für eine Gesamtdehnung von $\varepsilon = 0,0028$ sind in Abbildung 6.7 dargestellt. In diesem Fall durchdringen sich die Flächen der hoch aufgelösten Berechnung ε^P und des reduzierten Modells $\hat{\varepsilon}^P$ in zwei Bereichen. In der linken Darstellung in Abbildung 6.7 ist gut zu erkennen, wie die approximierte Lösung (blaue Punkte) durch den flachen Bereich der hoch aufgelösten Berechnung mit

¹Siehe hierzu Kapitel 5.2.

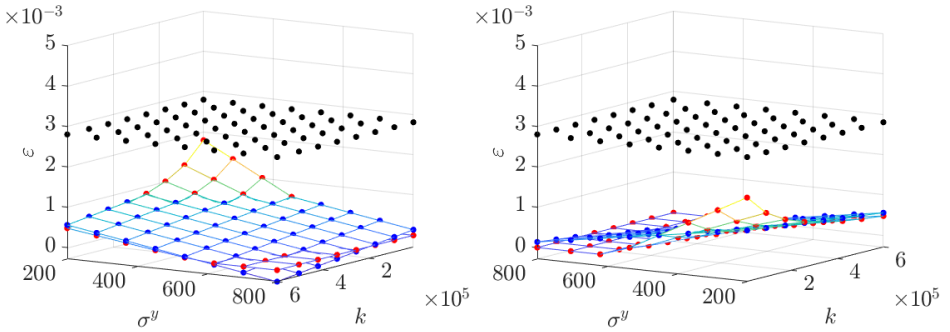


Abbildung 6.7: Gesamtdehnung (●) $\varepsilon = 0,0028$ mit zugehöriger plastischer Dehnung (●) ε^P ; Approximation mit linearer Kernel Funktion (●) $\hat{\varepsilon}^P$; σ^y in [MPa]

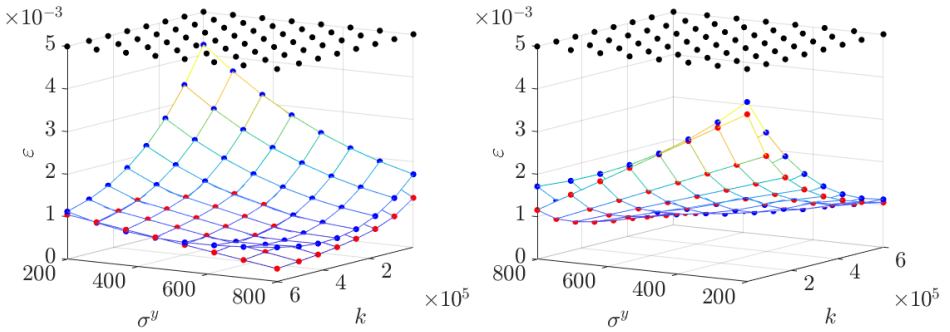


Abbildung 6.8: Gesamtdehnung (●) $\varepsilon = 0,005$ mit zugehöriger plastischer Dehnung (●) ε^P ; Approximation mit polynomialer Kernel Funktion (●) $\hat{\varepsilon}^P$; σ^y in [MPa]

$\varepsilon^P = 0$ durchtritt und in den negativen Bereich übergeht. Dies ist ein physikalisch unrealistisches Ergebnis. Im Bereich hoher Gradienten löst sich die approximierten Lösung wieder von der Zielfläche ab. Weitere Studien mit mehr Trainingspunkten konnten die Ergebnisse mit der linearen Kernel Funktion nicht wesentlich verbessern.

Die Ergebnisse für die Polynom Kernel Funktion sind in Abbildung 6.8 und 6.9 dargestellt. Bei den dargestellten Ergebnissen wurde ein Polynomgrad $p=3$ verwendet, was der Voreinstellung in MATLAB[®] entspricht. Im Falle der glatten Antwortfläche in Abbildung 6.8 (rote Punkte) ist die Approximation (blaue Punkte) deutlich besser im Gegensatz zur der Approximation mit der linearen Kernel Funktion. Es ist gut zu erkennen, dass die Bereiche hoher Gradienten durch die Polynomfunktion mit abgedeckt werden. Ein wesentliches Problem bei der Verwendung von polynomialen Kernel Funktionen wird in Abbildung 6.9 im flachen Bereich mit $\varepsilon^P = 0$ deutlich. Hier ist die Approximation nicht in der Lage, den flachen Verlauf abzubilden. Es kommt zu mehrfachem Durchtreten der Soll-Fläche aufgrund des stetigen polynomialen Verlaufs. Auch bei höheren und niedrigeren Polynomgraden p und mehr Trainingspunkten konnte diese Eigenschaft des Über- und Unterschwingens im flachen Bereich nicht minimiert bzw. unterbunden werden.

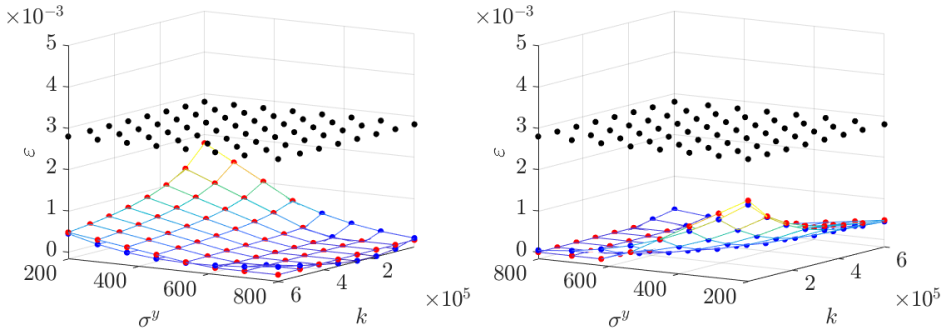


Abbildung 6.9: Gesamtdehnung (\bullet) $\varepsilon = 0,0028$ mit zugehöriger plastischer Dehnung (\bullet) ε^P ; Approximation mit polynomialer Kernel Funktion (\bullet) $\hat{\varepsilon}^P$; σ^y in [MPa]

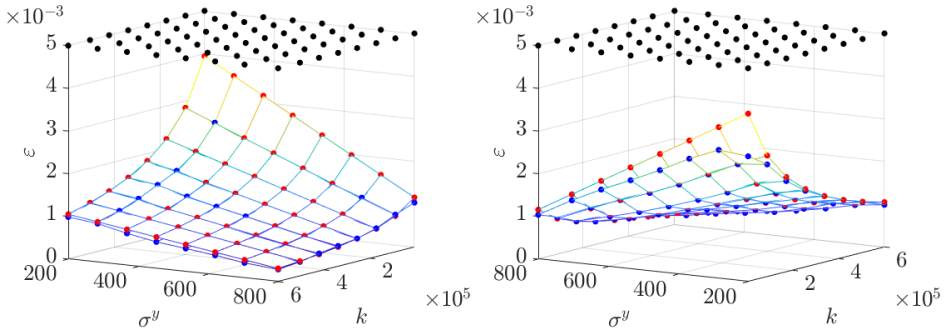


Abbildung 6.10: Gesamtdehnung (\bullet) $\varepsilon = 0,005$ mit zugehöriger plastischer Dehnung (\bullet) ε^P ; Approximation mit Gauß'scher Kernel Funktion (\bullet) $\hat{\varepsilon}^P$; σ^y in [MPa]

Weitaus bessere Ergebnisse lassen sich mit der *Gauß'schen* Kernel Funktion erzielen. Für den glatten Verlauf der Soll-Funktion mit $\varepsilon = 0,005$ ist das Ergebnis in Abbildung 6.10 dargestellt. Der Verlauf der Approximation (blaue Punkte) schmiegt sich größtenteils sehr gut an die Soll-Fläche (rote Punkte) an. Im Bereich hoher Gradienten löst sich die Approximation von der hoch aufgelösten Berechnung ab. Der Fehler konnte mit mehr Trainingspunkten etwas verringert werden. Im Gegensatz zur linearen und polynomialen Kernel Funktion ist der entscheidende Vorteil aber in der Approximation des flachen Bereiches bei $\varepsilon^P = 0$ zu sehen. In Abbildung 6.11 ist gut zu erkennen, wie die Approximation (blaue Punkte) den Verlauf der Soll-Funktion (rote Punkte) auch in diesem Bereich gut abbildet. Somit lässt sich im Vergleich der drei Kernel Funktionen (linear, Polynom, *Gauß*) festhalten, dass die *Gauß'sche* Kernel Funktion grundsätzlich die Eigenschaft hat, den nicht glatten Verlauf abzubilden.

Als Nächstes werden die passenden Parameter für die Support Vector Regression ermittelt. In der Veröffentlichung von Moustapha et al. (2018) wird vorgeschlagen, den Parameter ε (*intensive tube width*) möglichst klein zu wählen und den Parameter C (*penalty term*) möglichst groß.

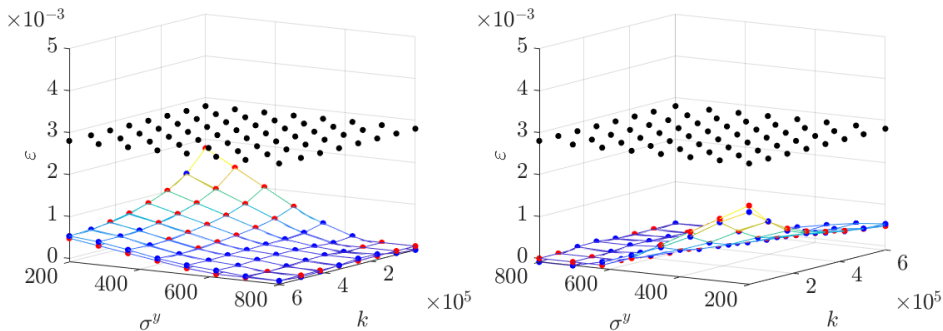


Abbildung 6.11: Gesamtdéhnung (●) $\varepsilon = 0,0028$ mit zugehöriger plastischer Déhnung (●) ε^P ; Approximation mit Gauß'scher Kernel Funktion (●) $\hat{\varepsilon}^P$; σ^y in [MPa]

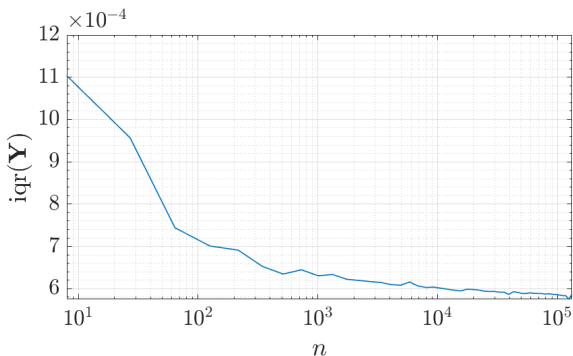


Abbildung 6.12: Interquartilsabstand $\text{iqr}(\cdot)$ über der Anzahl der Trainingspunkte n pro Durchlauf; n in logarithmischer Darstellung

Die Standard Werte in MATLAB[®] sind

$$\varepsilon = \frac{\text{iqr}(\mathbf{Y})}{13,49} \quad (6.10)$$

$$C = \frac{\text{iqr}(\mathbf{Y})}{1,349}, \quad (6.11)$$

wobei $\text{iqr}(\cdot)$ für den *Interquartilsabstand* steht. Der Verlauf des Interquartilsabstandes $\text{iqr}(\mathbf{Y})$ in Abhängigkeit von der Anzahl an Trainingspunkten n ist in Abbildung 6.12 gezeigt. Der Wert des Interquartilsabstandes verringert sich im Anfangsbereich und nimmt ab einer Anzahl von ca. $10^2 = 100$ Trainingspunkten Werte im Bereich zwischen $5,8 \cdot 10^{-4}$ und $7 \cdot 10^{-4}$ an, sodass bei steigender Anzahl an Trainingspunkten annähernd von einer Konstanten ausgegangen werden kann. Um die Parameter für das Surrogate Modell anzupassen wird zuerst eine Konvergenzstudie mit einem rein glatten Antwortspektrum durchgeführt. Sowohl die Trainingspunkte für das SVR Modell wie auch die Testpunkte für die eine Approximation gesucht werden soll, befinden sich im Bereich $\varepsilon^P > 0$. Die Punkte für ein Test-Gitter $\mathfrak{X}_{20 \times 20 \times 20}$, bei dem die Punkte mit

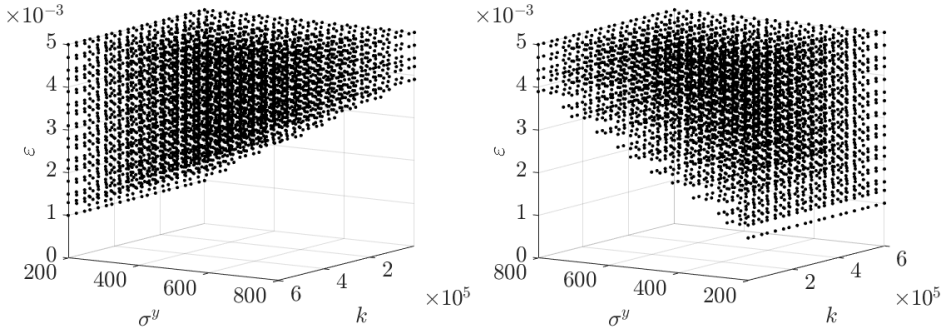


Abbildung 6.13: Gitter $\mathfrak{X}_{20 \times 20 \times 20}$; Gesamtdehnung (\bullet) ohne den Bereich $\varepsilon^P = 0$; σ^y in [MPa]

$\varepsilon^P = 0$ ausgeschlossen sind, sind in Abbildung 6.13 dargestellt. Für die Konvergenzstudien werden zwei verschiedene Ziel- bzw. Test-Gitter $\mathfrak{X}_{20 \times 20 \times 20}$ und $\mathfrak{X}_{51 \times 51 \times 51}$ verwendet. Das dünnere Gitter dient dazu, eine Richtung für die Parameterwahl festzustellen, da es nicht so rechenintensiv ist und somit schneller Aussagen getroffen werden können. Das dichte Gitter wird dann für eine abschließende Untersuchung herangezogen, mit der dann die Prognose aus der Berechnung mit dem dünneren Gitter bestätigt oder widerlegt wird. Beginnend mit sehr dünnen Trainings-Gittern, z.B. $\mathfrak{X}_{3 \times 3 \times 3}$, wird das SVR Modell dann in jedem Durchlauf mit einem immer dichter werdenden Gitter trainiert, bis schließlich die Dichte des Test-Gitters erreicht ist. In jedem Durchlauf wird geprüft, wie gut das SVR Modell das dichte Test-Gitter approximiert. Dafür werden zwei Fehlermaße verwendet. Um eine Aussage für den flächendeckenden Fehler zu bekommen, wird ein *relatives* Fehlermaß ϵ_{rel} eingeführt,

$$\begin{aligned} \tilde{y} &= \frac{\sum_{i=1}^n y_i}{n} \\ \epsilon_i &= \frac{|y_i - \hat{y}_i|}{|\tilde{y}|} \\ \epsilon_{rel} &= \frac{\sum_{i=1}^n \epsilon_i}{n}, \end{aligned} \quad (6.12)$$

wobei y_i das Soll-Ergebnis darstellt. Die Approximation mit dem SVR Modell wird mit \hat{y}_i gekennzeichnet und n ist die Gesamtzahl an Testpunkten. Für ein *absolutes* Fehlermaß wird die Maximum-Norm, auch *Tschebyschew-Norm* genannt, verwendet

$$\begin{aligned} e_i &= |y_i - \hat{y}_i| \\ \epsilon_{max} &:= \max\{e_1 \cdots e_n\}. \end{aligned} \quad (6.13)$$

Die Studien haben gezeigt, dass im Falle des glatten Antwortspektrums (ohne den Bereich $\varepsilon^P = 0$, siehe Abbildung 6.13) die Werte für die Parameter ε sehr klein und für C sehr groß gewählt werden können. Das Ergebnis wird dadurch zunehmend besser. Der maximale Fehler ϵ_{max} für eine Parameterwahl $\varepsilon = \text{iqr}(\mathbf{Y})/20000$ und $C = \text{iqr}(\mathbf{Y})/0,00001$ ist in Abbildung 6.14 in zweifach logarithmischer Skalierung dargestellt. Der entsprechende relative Fehler ϵ_{rel} ist in Abbildung 6.15, ebenfalls in zweifach logarithmischer Skalierung, dargestellt. Die Erweiterung der Konvergenzstudie für optimale Parameter auf volle Gitter, wie beispielsweise in der Abbildung 6.2, zeigt ein divergierendes Verhalten bei zu kleinen und zu großen Werten für ε bzw. C .

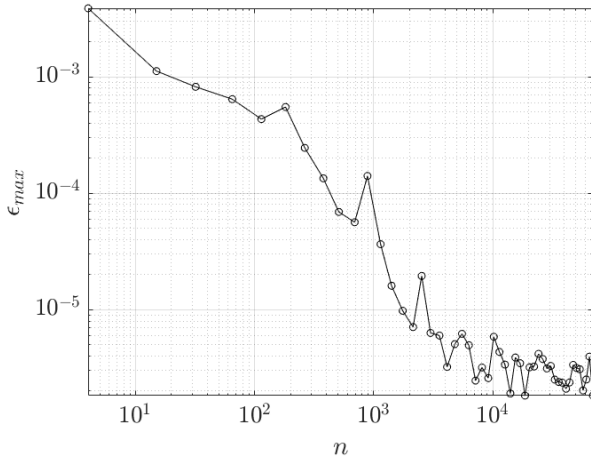


Abbildung 6.14: Maximum Norm ϵ_{max} : Gauß'sche Kernel Funktion; Parameter $\epsilon = iqr(\mathbf{Y})/20000$ und $C = iqr(\mathbf{Y})/0,00001$; Gitter ohne $\epsilon^p = 0$; n : Anzahl der Trainingspunkte im Gitter; Zweifach logarithmische Darstellung

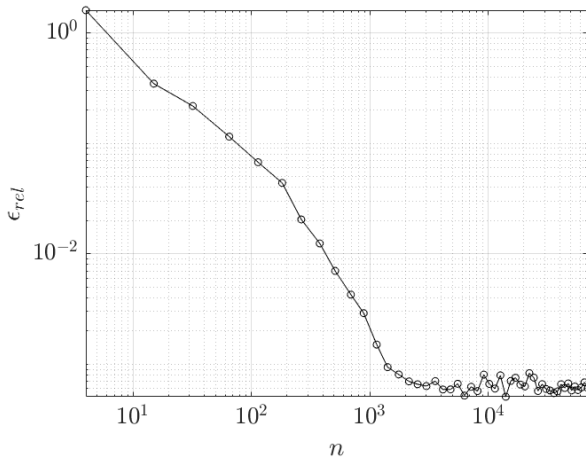


Abbildung 6.15: Relativer Fehler ϵ_{rel} : Gauß'sche Kernel Funktion; Parameter $\epsilon = iqr(\mathbf{Y})/20000$ und $C = iqr(\mathbf{Y})/0,00001$; Gitter ohne $\epsilon^p = 0$; n : Anzahl der Trainingspunkte im Gitter; Zweifach logarithmische Darstellung

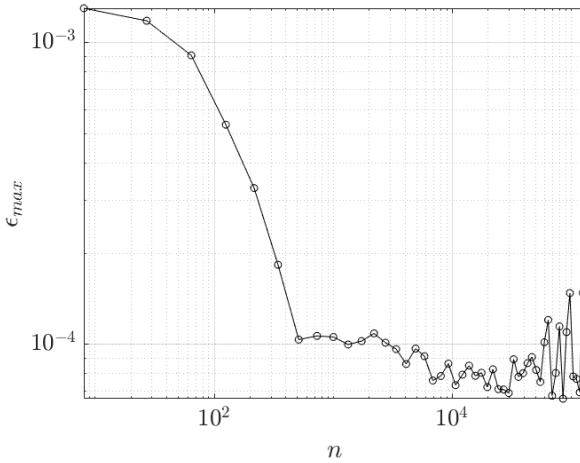


Abbildung 6.16: Maximum Norm ϵ_{max} : Gauß'sche Kernel Funktion; Parameter $\epsilon = \text{iqr}(\mathbf{Y})/200$ und $C = \text{iqr}(\mathbf{Y})/0,001$; Gitter mit $\epsilon^p = 0$; n : Anzahl der Trainingspunkte im Gitter; Zweifach logarithmische Darstellung

Im Rahmen dieser Arbeit wurden gute Erfahrungen mit den Werten

$$\epsilon_{opt} = \frac{\text{iqr}(\mathbf{Y})}{200} \quad (6.14)$$

$$C_{opt} = \frac{\text{iqr}(\mathbf{Y})}{0,001}, \quad (6.15)$$

gemacht. Allerdings hat die *Gauß'sche* Kernel Funktion einen entscheidenden Nachteil. Sie divergiert und oszilliert ab einer bestimmten Dichte des Trainings-Gitters. Das oszillierende Verhalten lässt sich sehr gut im Verlauf des absoluten Fehlers ϵ_{max} in Abbildung 6.16 erkennen. Das divergierende Verhalten ist dem Verlauf des relativen Fehlers ϵ_{rel} in Abbildung 6.17 zu entnehmen.

In Moustapha et al. (2018) werden gute Erfahrungen mit der *Matèrn 5/2* Kernel Funktion gemacht. Die *Matèrn 5/2* Kernel Funktion wird folgendermaßen angegeben (Moustapha et al., 2018),

$$K(\mathbf{x}_i, \mathbf{x}_j) = \left(1 + \sqrt{5} \frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\rho} + \frac{5}{3} \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\rho^2}\right) \exp\left(-\sqrt{5} \frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\rho}\right), \quad (6.16)$$

wobei ρ ein zu wählender Parameter ist. Im Rahmen der hier durchgeführten Konvergenzstudie wird $\rho = 1$ gesetzt. Zusammen mit den Parametern ϵ_{opt} und C_{opt} aus Gleichung (6.14) und (6.15) kann das Konvergenzverhalten im Gegensatz zur *Gauß'schen* Kernel Funktion deutlich verbessert werden. Der maximale Fehler mit der *Matèrn 5/2* Kernel Funktion ist in Abbildung 6.18 dargestellt. Zusätzlich ist als Referenz eine rote Linie eingezeichnet, die den minimalen Fehler kennzeichnet, der mit der *Gauß'schen* Kernel Funktion erreicht wird. Dadurch ist deutlich zu erkennen, dass die *Matèrn 5/2* Kernel Funktion an der Stelle, wo das Divergieren der *Gauß'schen* Kernel Funktion beginnt, weiter konvergiert. Das Ergebnis für den relativen Fehler

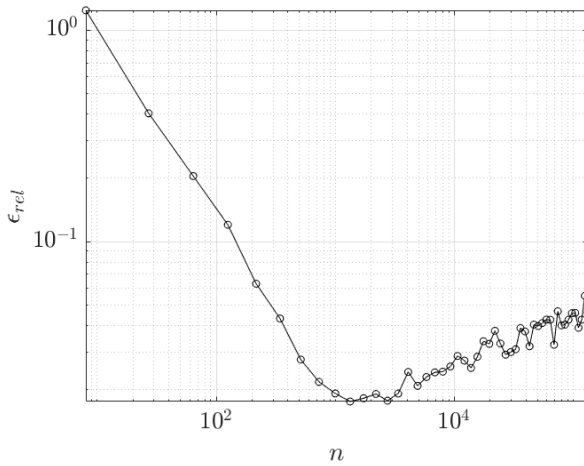


Abbildung 6.17: Relativer Fehler ϵ_{rel} : Gauß'sche Kernel Funktion; Parameter $\varepsilon = iqr(\mathbf{Y})/200$ und $C = iqr(\mathbf{Y})/0,001$; Gitter mit $\varepsilon^p = 0$; n : Anzahl der Trainingspunkte im Gitter; Zweifach logarithmische Darstellung

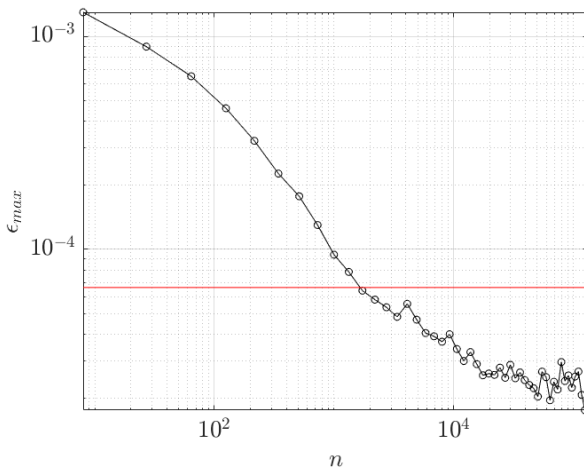


Abbildung 6.18: Maximum Norm ϵ_{max} : Matérn 5/2 Kernel Funktion; Parameter $\varepsilon = iqr(\mathbf{Y})/200$ und $C = iqr(\mathbf{Y})/0,001$; Gitter mit $\varepsilon^p = 0$; n : Anzahl der Trainingspunkte im Gitter; Zweifach logarithmische Darstellung; (—) Bestes Ergebnis mit der Gauß'schen Kernel Funktion als Vergleich

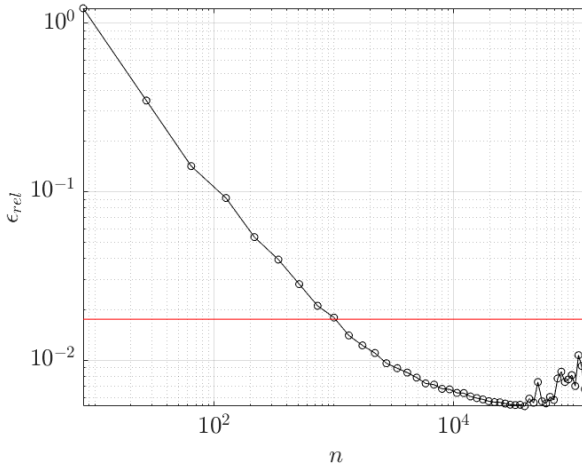


Abbildung 6.19: Relativer Fehler ϵ_{rel} : Matèrn 5/2 Kernel Funktion; Parameter $\epsilon = iqr(\mathbf{Y})/200$ und $C = iqr(\mathbf{Y})/0,001$; Gitter mit $\epsilon^p = 0$; n : Anzahl der Trainingspunkte im Gitter; Zweifach logarithmische Darstellung; (—) Bestes Ergebnis mit der Gauß'schen Kernel Funktion als Vergleich

mit der *Matèrn 5/2* Kernel Funktion ist in Abbildung 6.19 dargestellt. Auch hier ist wieder die rote Linie für den Vergleich mit der *Gauß'schen* Kernel Funktion mit angegeben. Wird die rote Linie aus Abbildung 6.18 mit einem Wert von $7 \cdot 10^{-5} = 0,00007$ als Referenz genommen, so ist daraus zu schließen, dass der absolute Fehler bei einem Dehnungsbereich $0 \leq \epsilon \leq 0,005$ sehr gering ist. Das Gleiche gilt für die rote Linie in Abbildung 6.19. Hier liegt der relative Fehler bei einem Wert von $2 \cdot 10^{-2} = 2\%$. Der absolute Fehler ϵ_{max} und der relative Fehler ϵ_{rel} können also mit der *Matèrn 5/2* Kernel Funktion in einen Bereich gebracht werden, der im Rahmen dieser mechanischen Anwendung absolut ausreichend ist. Im Rahmen weiterer Studien wurde untersucht, wie sich die automatische Parameterwahl in MATLAB[®] verhält. Bei der automatisierten Parameterwahl wird in der SVR Routine über die Minimierung des *cross-validation* Fehlers versucht, die optimalen Parameter zu finden. Die Optimierung kann dabei gezielt auf einzelne Parameter angewendet werden, oder aber auf einen Satz mehrerer Parameter. Der Versuch, mit dieser Möglichkeit, einzeln oder in Kombination, die Parameter ϵ , C , die Kernel Skalierung σ oder auch die Polynomordnung p zu optimieren, brachte keine zufriedenstellenden Ergebnisse.

Eine weitere Möglichkeit in MATLAB[®] besteht darin, bestimmte Daten in der Modellbildung zu gewichten. Durch den nicht glatten Verlauf im Antwortspektrum liegt es nahe, beispielsweise den Bereich mit $\epsilon^p = 0$ anders zu gewichten als den glatten Bereich. Bei einem Trainingsdatensatz $\{\mathbf{X}, \mathbf{Y}\}$ mit n Trainingspunkten, können mit einer Spaltenmatrix \mathbf{W} der Länge n gezielt Wichtungsfaktoren für die Punkte übergeben werden. Untersuchungen haben gezeigt, dass im Rahmen dieses Anwendungsfalls keine nennenswerten Verbesserungen erzielt werden können.

6.3 Überanpassung (engl. *Overfitting*)

Zur einer sogenannten Überanpassung (engl. *Overfitting*) kann es bei einer zu großen Anzahl an Trainingsdaten oder auch einem verrauschten Trainingsdatensatz kommen (Cristianini et al.,

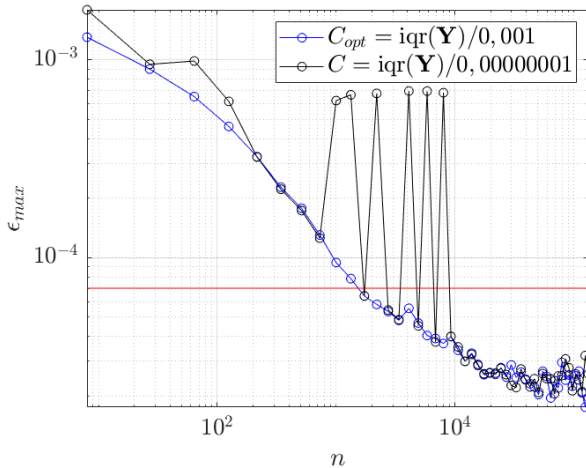


Abbildung 6.20: Maximum Norm ϵ_{max} : Matèrn 5/2 Kernel Funktion; Parameter $C_{opt} = iqr(\mathbf{Y})/200$ und $C = iqr(\mathbf{Y})/0,00000001$; Gitter mit $\varepsilon^p = 0$; n : Anzahl der Trainingspunkte im Gitter; Zweifach logarithmische Darstellung; (—) Bestes Ergebnis mit der Gauß'schen Kernel Funktion als Vergleich

2000). Der zweit-genannte Fall eines verrauschten Trainingsdatensatzes tritt bei Messdaten auf und ist hierbei auf sogenannte „Ausreißer“-Datenpunkte zurückzuführen. Im Rahmen dieser Arbeit liegen deterministische Trainingsdaten vor, die durch ein zugrundeliegendes mechanisches Modell ermittelt werden. Demzufolge kann eine Überanpassung nur auf eine zu große Anzahl an Trainingspunkten zurückgeführt werden. Der Verlauf des relativen Fehlers ϵ_{rel} in Abbildung 6.19 weist ab ca. $4 \cdot 10^4 = 40000$ Trainingspunkten n einen leicht divergierenden und oszillierenden Verlauf auf. Das Oszillieren lässt sich auch bei der Maximum Norm ϵ_{max} in Abbildung 6.18 ab einer Anzahl von $4 \cdot 10^4 = 40000$ Trainingspunkten beobachten. Das Hinzufügen von weiteren Trainingspunkten bewirkt somit ab ca. $n = 4 \cdot 10^4$ ein *Overfitting* des Modells. Die in diesem Fall beste Approximation des Testgitters $51 \times 51 \times 51$ mit $n_{test} = 132651$ Testpunkten erfolgt demnach „schon“ mit ca. $n \approx \frac{n_{test}}{3}$ Trainingspunkten und kann ab dort nicht durch das Hinzufügen weiterer Punkte verbessert werden.

Im Rahmen dieser Arbeit wurden im Hinblick auf eine Überanpassung weitere Studien zum Parameter C (*penalty term* (Moustapha et al., 2018) oder auch *box constraint* genannt (Cristianini et al., 2000)) durchgeführt. Dieser Parameter ist maßgeblich für die Vermeidung von *Overfitting*-Phänomenen bei der Support Vector Regression verantwortlich. In Moustapha et al. (2018) wird empfohlen, diesen Wert möglichst groß zu wählen, um gute Ergebnisse zu erzielen. Im Kapitel 6.2 werden bis zum Einsetzen des *Overfittings* gute Ergebnisse mit dem Wert $C_{opt} = \frac{iqr(\mathbf{Y})}{0,001}$ (Gleichung 6.15) erzielt, der 1349-fach größer ist als der voreingestellte Wert in MATLAB[®] aus Gleichung 6.11. Weitere Untersuchungen haben gezeigt, dass der Parameter nicht beliebig groß gewählt werden darf. Exemplarisch sind die Ergebnisse für einen 134900000-fach größeren Wert $C = \frac{iqr(\mathbf{Y})}{0,00000001}$ in Abbildung 6.20 für die Maximum Norm ϵ_{max} und in Abbildung 6.21 für den relativen Fehler ϵ_{rel} gezeigt. Als Referenz ist in beiden Abbildungen der Kurvenverlauf für C_{opt} mit dargestellt. Die Fehlerkurven von ϵ_{max} und ϵ_{rel} weisen für $C = iqr(\mathbf{Y})/0,00000001$ große „Ausreißer“ auf. Zudem kann das *Overfitting* ab $n = 4 \cdot 10^4$

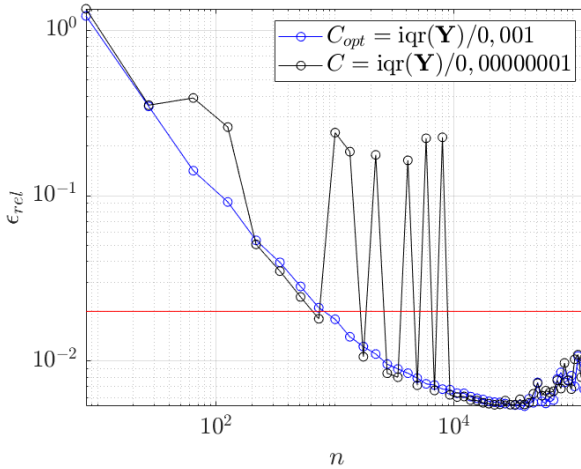


Abbildung 6.21: Relativer Fehler ϵ_{rel} : Matérn 5/2 Kernel Funktion; Parameter $C_{opt} = iqr(\mathbf{Y})/200$ und $C = iqr(\mathbf{Y})/0,00000001$; Gitter mit $\epsilon^p = 0$; n : Anzahl der Trainingspunkte im Gitter; Zweifach logarithmische Darstellung; (—) Bestes Ergebnis mit der Gauß'schen Kernel Funktion als Vergleich

Trainingspunkten nicht unterbunden werden.

In MATLAB[®] besteht die Möglichkeit, den Parameter C automatisiert anzupassen. Dies geschieht über die Minimierung eines Kreuzvalidierungs-Fehlers (engl. *Cross-Validation-Error*). Bei dieser Minimierung wird der optimale Wert für C in dem Bereich zwischen 10^{-3} und 10^3 gesucht. Die automatisierte Anpassung über eine Kreuzvalidierung hat einen deutlich erhöhten Berechnungsaufwand zur Folge. In Abbildung 6.22 ist der Verlauf des maximalen Fehlers ϵ_{max} für die automatisierte Anpassung des Parameters C dargestellt. Als Vergleich ist der Verlauf für C_{opt} mit dargestellt. Der automatisch angepasste Parameter ist mit C_{CVE} gekennzeichnet (**C**ross-**V**alidation-**E**rror). In Abbildung 6.23 sind die entsprechenden Verläufe des relativen Fehlers ϵ_{rel} dargestellt. Die Verläufe von C_{CVE} in Abbildung 6.22 und Abbildung 6.23 stimmen weitestgehend mit den Verläufen von C_{opt} überein. Die Berechnung wurde nach 38 Schleifendurchläufen beendet, da die automatisierte Anpassung des Parameters C_{CVE} über den Kreuzvalidierungs-Fehler bei einer größeren Anzahl an Trainingspunkten, beispielsweise $n > 6 \cdot 10^4$, schon mehrere Tage in Anspruch nimmt. Die Auswirkungen der automatisierten Anpassung auf den Bereich des *Overfittings* sind aber auch schon nach 38 Schleifendurchläufen feststellbar. In Abbildung 6.24 ist der Bereich, in dem das *Overfitting* einsetzt, für den Verlauf des maximalen Fehlers ϵ_{max} detailliert dargestellt. Es ist gut zu erkennen, dass die automatisierte Anpassung des Parameters C_{CVE} die Überanpassung in diesem Bereich nicht unterbinden kann. Es kommt zu ähnlichen Oszillationen wie bei dem Verlauf des fest gewählten Parameters C_{opt} . Die Verläufe der Parameter C_{opt} und C_{CVE} sind in Abbildung 6.25 dargestellt. Während der Parameter $C_{opt} \approx 0,61$ annähernd konstant verläuft, oszilliert der Wert C_{CVE} durchgehend. Die automatisierte Anpassung des Parameters C hat somit keinen positiven Einfluss auf das Unterbinden der Überanpassung. Zudem ist der Berechnungsaufwand über die zusätzliche Kreuzvalidierung sehr groß und in diesem Anwendungsfall nicht Zielführend.

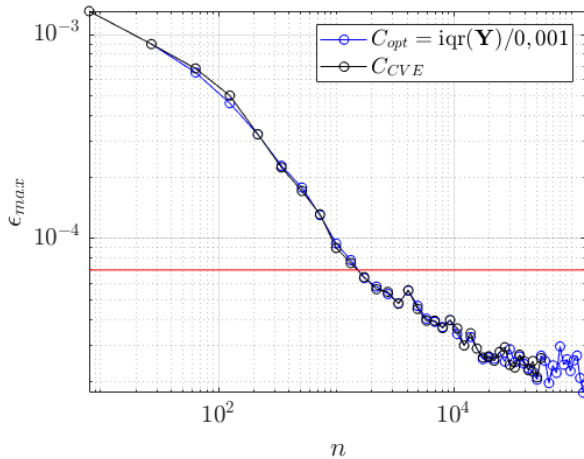


Abbildung 6.22: Maximum Norm ϵ_{max} : Matérn 5/2 Kernel Funktion; Parameter $C_{opt} = \text{iqr}(\mathbf{Y})/200$ und C_{CVE} ; Gitter mit $\varepsilon^p = 0$; n : Anzahl der Trainingspunkte im Gitter; Zweifach logarithmische Darstellung; (—) Bestes Ergebnis mit der Gauß'schen Kernel Funktion als Vergleich

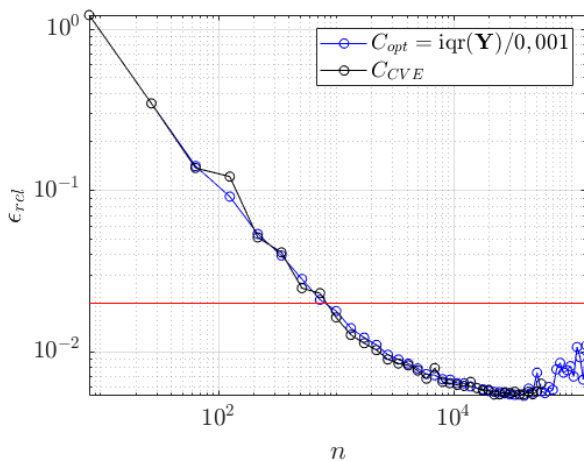


Abbildung 6.23: Relativer Fehler ϵ_{rel} : Matérn 5/2 Kernel Funktion; Parameter $C_{opt} = \text{iqr}(\mathbf{Y})/200$ und C_{CVE} ; Gitter mit $\varepsilon^p = 0$; n : Anzahl der Trainingspunkte im Gitter; Zweifach logarithmische Darstellung; (—) Bestes Ergebnis mit der Gauß'schen Kernel Funktion als Vergleich

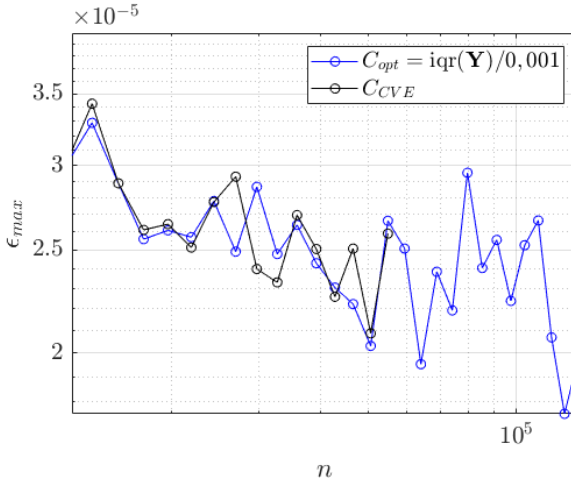


Abbildung 6.24: (Detail Overfitting) Maximum Norm ϵ_{max} : Matérn 5/2 Kernel Funktion; Parameter $C_{opt} = iqr(\mathbf{Y})/200$ und C_{CVE} ; Gitter mit $\varepsilon^p = 0$; n : Anzahl der Trainingspunkte im Gitter; Zweifach logarithmische Darstellung

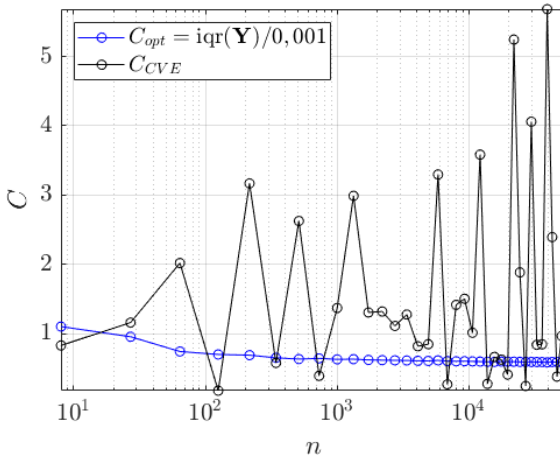


Abbildung 6.25: Verlauf der Parameter C_{opt} und C_{CVE} : Matérn 5/2 Kernel Funktion; Gitter mit $\varepsilon^p = 0$; n : Anzahl der Trainingspunkte im Gitter; Einfach logarithmische Darstellung für die Anzahl der Trainingspunkte n

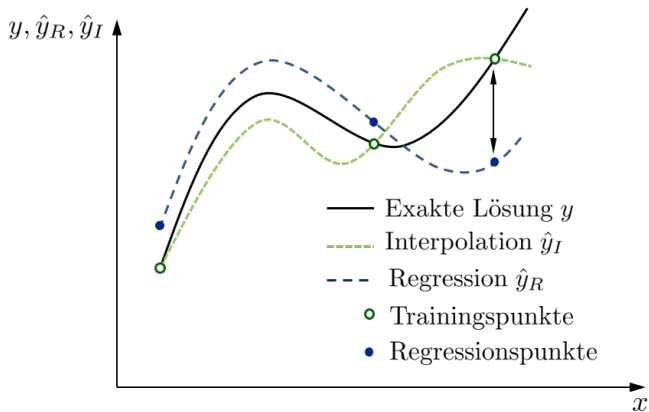


Abbildung 6.26: Unterschied zwischen Regression und Interpolation

6.4 Studien zur Sampling Methode

Im Kapitel 6.2 werden die Konvergenzstudien mit *kartesischen* Datenpunktgittern durchgeführt. Dabei sind sowohl die Test-Gitter als auch die Trainings-Gitter für die *Support Vector Regression* äquidistant in allen drei Parametern ε_i , k_i und σ_i^y . Um die Leistungsfähigkeit der SVR zu bewerten, wird das reduzierte Modell ausgehend von einem groben Gitter mit immer dichteren Trainings-Gittern trainiert. Eine Erhöhung der Anzahl der Trainingspunkte auf diese Weise wird auch als *homogene Verfeinerung* bezeichnet.

In diesem Kapitel werden die durchgeführten Studien zur *adaptiven Verfeinerung*, d.h. zum sogenannten *adaptiven Sampling* vorgestellt. Im Rahmen dieser Forschungsarbeit wurde eine Vielzahl von Methoden untersucht. In den ersten Studien wird die kartesische Struktur beim Hinzufügen von Datenpunkten weitestgehend beibehalten. Da es sich bei SVR um ein Regressionsverfahren handelt, ist es möglich, ein Modell mit vorgegebenen Datenpunkten zu trainieren und bei anschließender Auswertung derselben Punkte einen Unterschied zur exakten Lösung zu erhalten. Dies ist bei einer Interpolation nicht der Fall; der Sachverhalt ist in Abbildung 6.26 motiviert. Während die Interpolation die vorgegebenen Trainingspunkte genau abbildet, „kann“ die Regression hier Abweichungen aufweisen. Dies gibt allerdings noch keinen Aufschluss über die Approximationsgüte der beiden Verfahren allgemein, siehe Abbildung 6.26. Für das Hinzufügen von Datenpunkten kann beispielsweise der Punkt aus den vorgegebenen Trainingspunkten x_i herangezogen werden, dessen Lösung den größten Abstand $|y(x_i) - \hat{y}_R(x_i)|$ zur exakten Lösung vorweist. Angewendet auf die Trainingspunkte $\mathbf{x}_i = [\varepsilon_i, k_i, \sigma_i^y]^T$ wird somit erst das Modell trainiert und dann für dieselben (Trainings-)Punkte ausgewertet. Ist dann der Trainingspunkt $\tilde{\mathbf{x}}_i$ mit der größten Abweichung zur exakten Lösung gefunden, so bietet sich im kartesischen Gitter aufgrund der einfachen Berechnung ein Hinzufügen von Trainingspunkten in *Sternform* an. Die Tilde über dem Datenpunkt dient zur Unterscheidung von den restlichen Punkten im Datenpunktgitter. Für den betreffenden Trainingspunkt $\tilde{\mathbf{x}}_i$ werden die Abstände zu den nächstgelegenen vorhandenen Nachbarpunkten berechnet. Die additiven Punkte werden dann genau mittig auf den Verbindungslinien zwischen dem kritischen Punkt und seinen nächsten Nachbarpunkten platziert. Dabei können, wie in Abbildung 6.27 dargestellt, vier verschiedene Fälle unterschieden werden, je nachdem, ob der Fehler im Inneren des Sampling-Bereiches, in einer Ecke, auf einer Fläche oder auf einer Kante liegt. Im Vergleich zum *homogenen Verfeinern* hat diese Vorgehensweise keine Verbesserung erzielen können; der Approximationsfehler

ϵ_{max} aus Gleichung (6.13) ist in den durchgeführten Berechnungen bei ähnlicher Anzahl an Trainingspunkten größer.

Der nächste Schritt besteht darin, das Kriterium für die Ermittlung des kritischen Punktes zu verändern. Eine Möglichkeit ist die *Kreuzvalidierung*, insbesondere die *Leave-One-Out Cross-Validation* (LOOCV) (Fuhg et al., 2020). Dabei wird das Modell mit einem reduzierten Datensatz trainiert und mit dem „voll“ trainierten Modell verglichen. Stehen beispielsweise n Trainingspunkte \mathbf{x}_i ($i = 1, \dots, n$) zu Verfügung, so wird nacheinander immer einer der Trainingspunkte zur Berechnung des reduzierten Modells entfernt. Mit diesen reduzierten Modellen wird jeweils eine Approximation $\hat{\mathbf{Y}}_{-i}(\mathbf{X})$ für die n zu Verfügung stehenden Trainingspunkte $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_n]^T$ selbst durchgeführt. Dabei steht $\hat{\mathbf{Y}}_{-i} = [\hat{y}_1 \cdots \hat{y}_n]^T$ für diejenige Approximation, für die der Datenpunkt \mathbf{x}_i aus dem Trainingsdatensatz ausgelassen wird und das Modell mit $n-1$ Trainingspunkten berechnet wird. In Jiang et al. (2020) wird der quadratische mittlere Fehler der *Leave-One-Out Cross-Validation* wie folgt angegeben,

$$\epsilon_{LOOCV} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\hat{\mathbf{Y}}_{-i}(\mathbf{X}) - \hat{\mathbf{Y}}(\mathbf{X}) \right)^2}, \quad (6.17)$$

wobei $\hat{\mathbf{Y}}(\mathbf{X})$ die Approximation des mit allen n Trainingspunkten trainierten Modells bezeichnet. Ein Indikator für eine lokale Verfeinerung kann beispielsweise der Punkt $\tilde{\mathbf{x}}_i$ sein, bei dem der Wert

$$\epsilon_{LOOCV}^{(-i)} = \| \hat{\mathbf{Y}}_{-i}(\mathbf{X}) - \hat{\mathbf{Y}}(\mathbf{X}) \| \quad (6.18)$$

maximal wird. In MATLAB[®] ist die Berechnung der *Leave-One-Out Cross-Validation* im Rahmen der Support Vector Regression direkt möglich. Ist der Punkt $\tilde{\mathbf{x}}_i$ mit dem maximalen Fehler (6.18) gefunden, kann eine Verfeinerung in der eingangs beschriebenen *Sternform* stattfinden (siehe hierzu Abbildung 6.27). Eine weitere Möglichkeit besteht darin, die Fehler (6.18) in absteigender Reihenfolge zu ordnen und die k größten Punkte $\tilde{\mathbf{x}}_i$ ($i = 1, \dots, k$) $_{k \leq n}$ als Region zur Verfeinerung heranzuziehen. Dieses Vorgehen ist in Algorithmus 2 dargestellt, wobei j den „Schleifenzähler“ kennzeichnet und i wieder den Index für die Datenpunkte. Der Algo-

Algorithmus 2 LOOCV mit Verfeinerung in Sternform

- 1: **for** $j = 1$ to N **do**
 - 2: $\mathbf{X}_j \mapsto$ Auswertung LOOCV
 - 3: Sortierung ϵ_{LOOCV}^{-i} , $i = 1, \dots, n$ in absteigender Reihenfolge
 - 4: Auswahl der größten k ($k \leq n$) Werte aus Schritt 3 $\mapsto \tilde{\mathbf{x}}_i$ ($i = 1, \dots, k$) $_{k \leq n}$
 - 5: $\tilde{\mathbf{x}}_i$ ($i = 1, \dots, k$) $_{k \leq n} \mapsto$ Berechnung der jeweils 6 nächsten Nachbarpunkte
 - 6: Hinzufügen von adaptiven Punkten in *Sternform* $\mapsto \mathbf{X}^{adaptiv}$
 - 7: $\mathbf{X}_{j+1} = \begin{bmatrix} \mathbf{X}_j \\ \mathbf{X}^{adaptiv} \end{bmatrix}$
-

rithmus 2 ist stark von den Start-Trainingspunkten \mathbf{X}_0 und der Anzahl k der in Betracht zu ziehenden größten Fehlerpunkte abhängig. Als Beispiel ist im Folgenden das Ergebnis für ein $\mathfrak{X}_{3 \times 3 \times 3}$ Start-Gitter \mathbf{X}_0 aufgezeigt. Ausgehend von diesen $3^3 = 27$ Trainingspunkten wurden in jedem Schleifendurchlauf $k = 2$ kritische Fehlerpunkte herangezogen. Mögliche Dopplungen

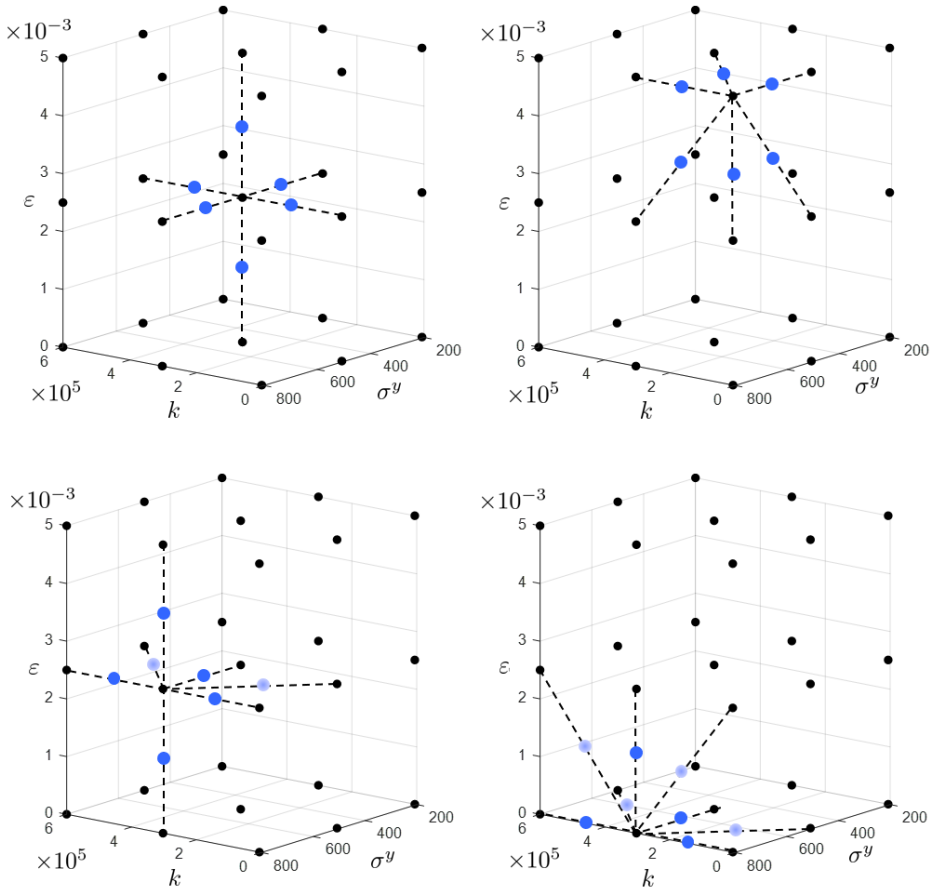


Abbildung 6.27: Sternförmige Verfeinerung um Punkte mit großem Fehler, (●) neue Datenpunkte, (●) vorhandene Datenpunkte, (oben links) Fehler innerhalb, (oben rechts) Fehler in Ecke, (unten links) Fehler auf Fläche, (unten rechts) Fehler auf Kante

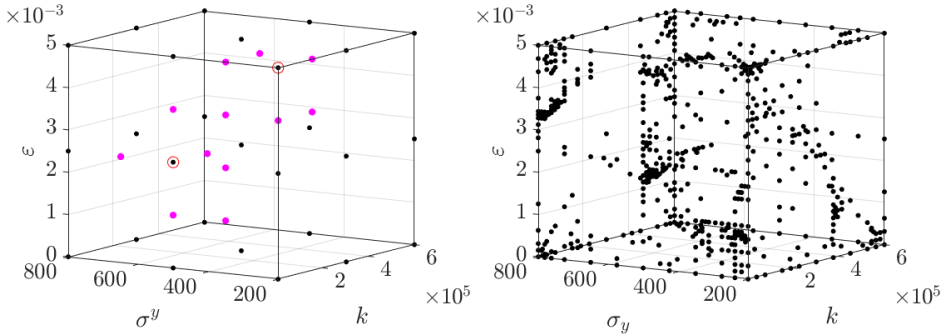


Abbildung 6.28: LOOCV mit Verfeinerung in Sternform: Datenpunkte nach ersten Schleifendurchlauf, (●) \mathbf{X}_0 , (○) $\tilde{\mathbf{x}}_i$ $i = 1, 2$, (●) $\mathbf{X}^{adaptive}$ (links); Datenpunkte nach 51 Schleifendurchläufen, (●) $\mathbf{X}_{i=51}$ (rechts); σ^y in [MPa]

beim Hinzufügen von Trainingspunkten in Sternform von benachbarten Fehlerpunkten wurden unterbunden bzw. entfernt. Die Datenpunktstruktur nach dem ersten Schleifendurchlauf und dem Schleifendurchlauf $j = 51$ (618 Trainingspunkte) sind in Abbildung 6.28 zu sehen. Um die Qualität der adaptiven Verfeinerung zu bewerten, werden die Maximum Norm ϵ_{max} und der relative Fehler ϵ_{rel} mit den Ergebnissen aus Kapitel 6.2 verglichen. Das Test-Gitter $\mathfrak{X}_{51 \times 51 \times 51}$ wird dabei nach jedem Schleifendurchlauf approximiert, um eine Vergleichbarkeit mit der homogenen Verfeinerung herzustellen. Das Ergebnis für die Maximum Norm nach $j = 72$ Schleifendurchläufen und somit 848 Trainingspunkten ist in Abbildung 6.29 dargestellt. Der relative Fehler ist in Abbildung 6.30 zu sehen. Aus der Abbildung 6.29 wird deutlich, dass ab 512 Datenpunkten im kartesischen Gitter $\mathfrak{X}_{8 \times 8 \times 8}$ die adaptive Verfeinerung nicht mehr unter der Kurve der homogenen Verfeinerung verläuft. Hier tritt ein „Durchstechen“ der blauen Kurve durch die schwarze Kurve nach dem Schleifendurchlauf $j = 52$ auf. Der relative Fehler der adaptiven Verfeinerung in Abbildung 6.30 stimmt anfangs weitestgehend mit der homogenen Verfeinerung überein und driftet ab 125 Datenpunkten im kartesischen Gitter $\mathfrak{X}_{5 \times 5 \times 5}$ stark ab. Die maximale Anzahl der Schleifendurchläufe beträgt in dieser Studie $j = 72$.

In einer weiteren Studie soll der verfolgte Ansatz der sternförmigen Verfeinerung im Rahmen der *Leave-One-Out Cross-Validation* erweitert und abgeändert werden. Die Idee ist es, Informationen über das Verhalten des Antwortspektrums mit in den adaptiven Sampling-Prozess einzubauen. Die plastische Dehnung ϵ^p ist in einem großen Datenpunktbereich $\mathbf{x}_i = [\epsilon_i, k_i, \sigma_i^y]^T$ gleich null und geht C^0 -stetig in einen glatten, nichtlinearen Bereich über (siehe hierzu auch Abbildung 6.3). In einer Vorstudie wurden Punkte additiv aus dem $\mathfrak{X}_{51 \times 51 \times 51}$ Test-Gitter hinzugenommen, bei denen die Abweichung zur exakten Lösung am größten waren. Dabei wurden Doppelungen in den Trainingspunkten ausgeschlossen, wenn beispielsweise die maximale Abweichung bei einem der Trainingspunkte selbst lag. Diese Untersuchung zeigt eine allmähliche Anhäufung der hinzukommenden Trainingspunkte im Bereich der C^0 -„Trennebene“ aus Abbildung 6.3. Hierzu sei erwähnt, dass diese Vorstudie keine Samplingmethode darstellt, da die hinzugefügten Trainingspunkte aus dem zu approximierenden Test-Gitter entnommen werden. Um nun gezielt Punkte in diesen Trennbereich einzufügen, bietet sich ein Sampling für eine *Klassifizierung* der Datenpunkte an. Für das elasto-plastische, eindimensionale Materialmodell würde eine Klassifizierung wie folgt aussehen

$$l_i = \begin{cases} 1 & \epsilon_i^p(\epsilon_i, k_i, \sigma_i^y) > 0 \\ -1 & \epsilon_i^p(\epsilon_i, k_i, \sigma_i^y) = 0 \end{cases} \quad (6.19)$$

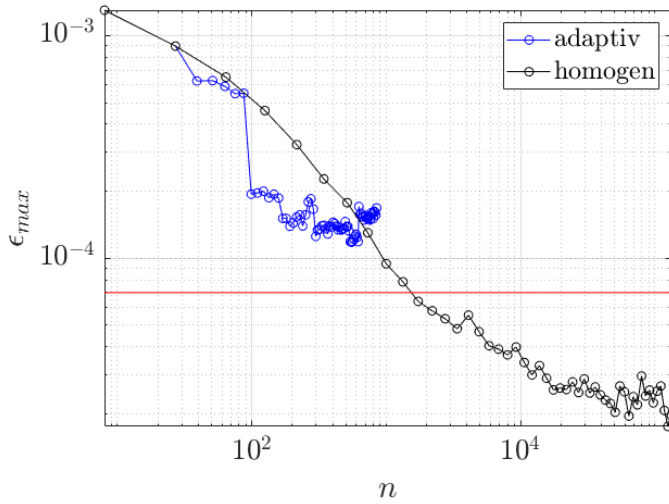


Abbildung 6.29: Maximum Norm ϵ_{max} : Vergleich der homogenen Verfeinerung mit der adaptiven Verfeinerung (Sternform); (—) Bestes Ergebnis mit der Gauß'schen Kernel Funktion als Vergleich

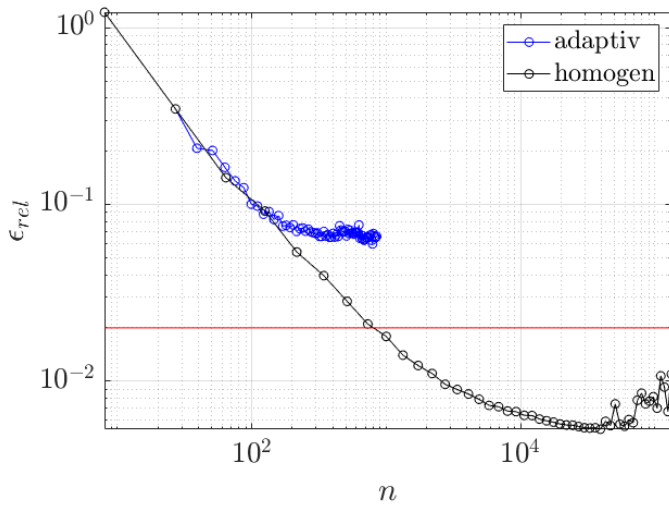


Abbildung 6.30: Relativer Fehler ϵ_{rel} : Vergleich der homogenen Verfeinerung mit der adaptiven Verfeinerung (Sternform); (—) Bestes Ergebnis mit der Gauß'schen Kernel Funktion als Vergleich

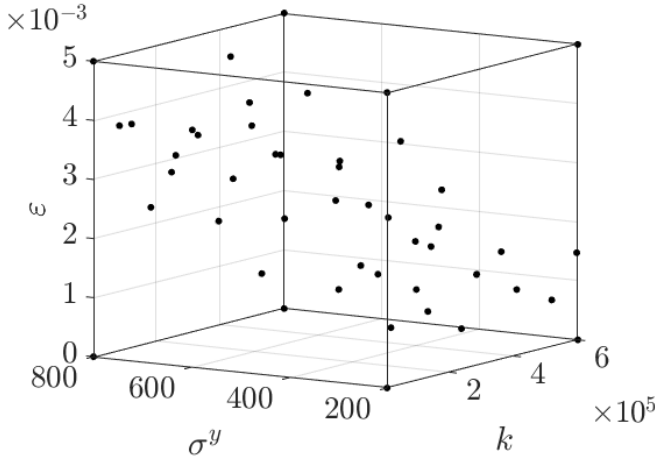


Abbildung 6.31: Trainingspunkte nach 40 adaptiv hinzugefügten Punkten mit dem „max-min“ Sampling der CODES[®]-Toolbox

Dabei beschreibt l_i die Markierung (engl. Label) des Datenpunktes, die entweder den Wert 1 oder den Wert -1 annimmt. Im Rahmen dieser Arbeit wird die CODES[®]-Toolbox von Lacaze und Missoum (2015) für die Klassifizierung und das hierfür mit bereitgestellte adaptive Sampling verwendet. Insbesondere wird das zur Verfügung gestellte und in Basudhar und Missoum (2010) vorgestellte „max-min“ Sampling herangezogen. Die *Support Vector Machine* (SVM) zur Klassifizierung wird mit der in der CODES[®]-Toolbox zur Verfügung stehenden *Gauß'schen* Kernel Funktion trainiert. Um die Qualität zu bewerten, wird ein Start-Gitter \mathbf{X}_0 mit lediglich 8 Eckpunkten des Sampling Bereiches vorgegeben. Additiv werden dann $k^{SVM} = 40$ Trainingspunkte hinzugefügt. Die Verteilung der Trainingspunkte nach dem Sampling Prozess ist in Abbildung 6.31 dargestellt. Es ist deutlich zu erkennen, dass die hinzugekommenen Trainingspunkte nahe an der Trennfläche der Bereiche $\varepsilon^p = 0$ und $\varepsilon^p > 0$ liegen. Um die Güte der SVM zu bewerten, wird die Klassifizierung (6.19) für ein Test-Gitter $\mathfrak{X}_{20 \times 20 \times 20}$, also 8000 homogen verteilte Testpunkte mit der CODES[®]-Toolbox durchgeführt. Das Ergebnis ist in Abbildung 6.32 gezeigt. Es ist gut zu erkennen, dass die Klassifizierung mit der geringen Anzahl an Trainingspunkten schon recht gute Ergebnisse liefert.

Ein weiteres mögliches Element des adaptiven Samplings ist es, den noch nicht abgedeckten Raum zu „ertasten“ und nicht nur lokal in Bereichen großer Fehler zu verfeinern. Daher kann es sinnvoll sein, einen *Explorations*-Anteil in das adaptive Sampling mit einzubinden. Im Rahmen der Untersuchungen werden hierfür Triangulierungen über den Samplingbereich und seine Begrenzung durchgeführt. Dabei entstehen auf dem Rand des Samplingbereiches Dreiecke und im Samplingbereich Tetraeder. Zur Verfeinerung werden dann die Schwerpunkte der größten Dreiecke und Tetraeder hinzugezogen. Um die Dreiecke auf dem Rand zu berechnen, wird zuerst die konvexe Hülle für die Trainingspunkte berechnet. Dabei handelt es sich um eine Standardfunktion in MATLAB[®], die eine Konnektivität als Ergebnis liefert, also welche 3 Trainingspunkte jeweils ein Dreieck auf dem Rand des Samplingbereiches bilden. Die Dreiecke überdecken dabei in Summe den gesamten Rand. Für die Berechnung der Schwerpunkte in den Dreiecken wird die frei zu Verfügung stehende MATLAB[®]-Funktion von Morales (2021) gewählt. Im Rahmen dieser Arbeit wird die Funktion erweitert, um zusätzlich den Flächeninhalt

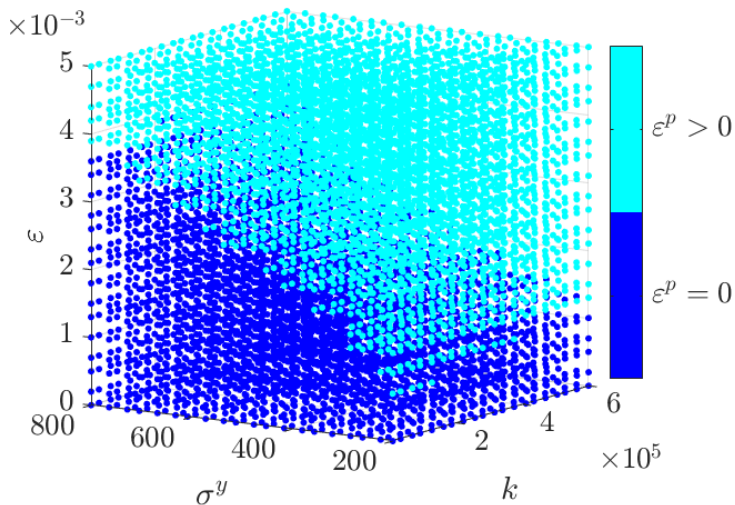


Abbildung 6.32: Klassifizierung von 8000 Testpunkten $\mathfrak{X}_{20 \times 20 \times 20}$ nach adaptiver Verfeinerung mit dem „max-min“ Sampling der CODES[®]-Toolbox

als Ausgabewert zu erhalten. Mithilfe des Flächeninhaltes können die Dreiecke der Größe nach sortiert werden. Die Schwerpunkte der größten k^{Tri} Dreiecke werden dann als neue Trainingspunkte zu den vorhandenen Trainingspunkten hinzugefügt. Beispielhaft ist eine Triangulierung mittels konvexer Hülle in Abbildung 6.33 gezeigt, bei der die größten $k^{Tri} = 20$ Dreiecke in Betracht gezogen werden. Eine Triangulierung im Samplingbereich über die Trainingspunkte \mathbf{X} resultiert in ein Tetraedernetz. In MATLAB[®] steht hierfür die *Delaunay*-Triangulierung zur Verfügung. Als Ergebnis wird die Konnektivität zurückgegeben, also welche 4 Trainingspunkte jeweils ein Tetraeder erzeugen. Mit der MATLAB[®]-Standardfunktion „incenter“ ist es dann möglich, die notwendigen Schwerpunkte der Tetraeder zu berechnen. Das Volumen jedes Tetraeders kann wieder über die konvexe Hülle berechnet werden. Als Beispiel ist in Abbildung 6.34 die Hinzunahme der Schwerpunkte der größten $k^{Tet} = 10$ Tetraeder dargestellt. Mit der Triangulierung im Samplingbereich und auf dem Rand werden Trainingspunkte hinzugefügt, die helfen, den gesamten Definitionsbereich zu füllen bzw. zu erkunden. Allerdings ist bei diesem Vorgehen lediglich der Volumen- bzw. Flächeninhalt für das Hinzufügen von Punkten entscheidend. Somit ist es sinnvoll, im Anschluss an die drei beschriebenen Samplingmethoden wieder eine *Leave-One-Out Cross-Validation* (LOOCV) durchzuführen und Punkte in den Bereichen großer Fehler zu ergänzen. Von dem eingangs beschriebenen sternförmigen Hinzufügen von Samplingpunkten wird im Folgenden abgewichen, da die kartesische Gitterstruktur durch die beschriebenen Methoden nicht mehr bestehen bleibt. Es wird stattdessen ein Weg gewählt, bei dem auf der Verbindungslinie zwischen zwei kritischen LOOCV-Punkten neue Punkte platziert werden. Ausgehend vom Mittelpunkt dieser Verbindungslinie werden dann zwei additive Punkte in die Nähe der kritischen Punkte „geschoben“. Wie dicht ein additiver Punkt an einen kritischen Punkt geschoben wird, kann dabei über einen Wichtungsfaktor gesteuert werden. Der Faktorwert $w = 1$ schiebt den Punkt direkt auf die Position des kritischen Punktes. Der Wert $w = 0$ bedeutet eine Platzierung genau in der Mitte der Verbindungslinie zweier kritischer

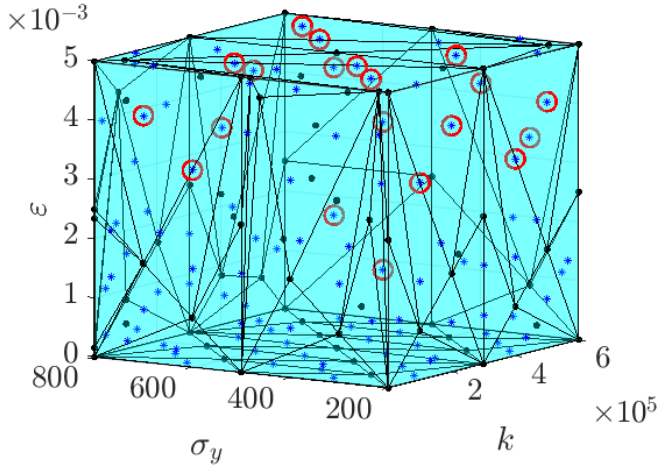


Abbildung 6.33: Triangulierung mittels konvexer Hülle und Hinzufügen der Schwerpunkte der größten Dreiecke mithilfe der „Center“-Funktion, (●) \mathbf{X}_j , (*) Schwerpunkte der Dreiecke, (○) $\mathbf{X}^{adaptiv}$; σ^y in [MPa]

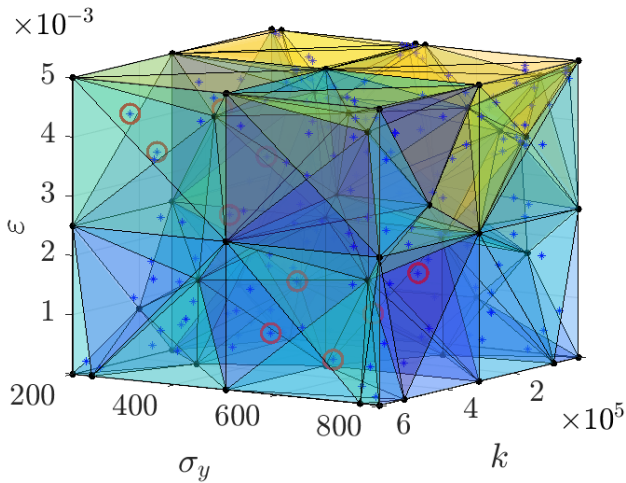


Abbildung 6.34: Erstellung der Tetraeder mittels Delaunay-Triangulierung und Hinzufügen der Schwerpunkte der größten Tetraeder, (●) \mathbf{X}_j , (*) Schwerpunkte der Tetraeder, (○) $\mathbf{X}^{adaptiv}$; σ^y in [MPa]

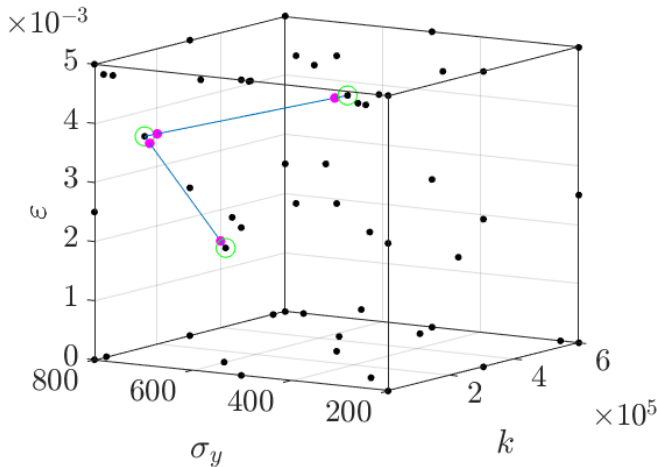


Abbildung 6.35: Hinzufügen von Trainingspunkten basierend auf LOOCV und Verbindungslinien zwischen den kritischen Punkten, (●) \mathbf{X}_j , (○) kritische LOOCV Punkte, (—) Verbindungslinie zwischen den kritischen Punkten, (●) $\mathbf{X}^{adaptive}$; σ^y in [MPa]

Punkte. Im Rahmen der Untersuchungen haben Werte zwischen $w = 0,7$ und $w = 0,9$ gute Resultate erzielt. In der Abbildung 6.35 ist ein Ergebnis für den Wert $w = 0,875$ dargestellt. In dem gezeigten Beispiel werden drei kritische LOOCV-Punkte in absteigender Reihenfolge ermittelt und die Verbindungslinien berechnet. In der Abbildung 6.35 sind die Verbindungslinien zwischen den kritischen Punkten mit dargestellt. Somit werden in diesem Beispiel $k^{LOOCV} = 4$ Trainingspunkte hinzugefügt.

Die vier beschriebenen Samplingmethoden werden in einer „äußeren“ Schleife N -mal wiederholt nacheinander ausgeführt. Zudem wurden „innere“ Schleifen für die einzelnen Anteile eingefügt, was dem Algorithmus mehr Flexibilität verleiht. Dabei wird der betreffende Sampling-Anteil mehrfach ausgeführt. Der „max-min“ Sampling Prozess mit der CODES[®]-Toolbox wird dabei N^{SVM} -mal wiederholt, wobei immer $k^{SVM} = 1$ Trainingspunkt hinzugefügt und dann die *Support Vector Machine* zur Klassifizierung anschließend neu trainiert wird. Bei der Triangulierung des Randes wird der Prozess N^{Tri} -mal wiederholt, wobei immer die Schwerpunkte der k^{Tri} größten Dreiecke hinzugenommen werden. Bei der *Delanunay*-Triangulierung werden in den N^{Tet} inneren Schleifendurchläufen jeweils die Schwerpunkte der k^{Tet} größten Tetraeder hinzugefügt. Abschließend wird die *Leave-One-Out Cross-Validation* N^{LOOCV} -mal ausgeführt und jeweils k^{LOOCV} Trainingspunkte hinzugefügt. In jedem der N^{LOOCV} Durchläufe wird die *Support Vector Regression* neu trainiert und die LOOCV ausgewertet. Das beschriebene Vorgehen mit „inneren“ Schleifen hat allerdings mehr zu wählende Parameter zur Folge, was das Finden einer optimalen Lösung schwierig gestaltet. In Algorithmus 3 ist der strukturelle Aufbau und die Abfolge dieses zusammengesetzten Samplingprozesses dokumentiert, wobei die „inneren“ Schleifen zur besseren Übersichtlichkeit der prinzipiellen Struktur nicht mit aufgeführt sind.

Es wurden gute Ergebnisse mit den Parametern in Tabelle 6.1 erzielt. Die Trainingspunkte nach $N = 55$ äußeren Schleifendurchläufen und einem kartesischen Startgitter $\mathfrak{X}_{3 \times 3 \times 3}$ sind in Abbildung 6.36 dargestellt. Der Wichtungsfaktor für die im LOOCV Verfahren hinzugefügten Trainingspunkte beträgt hierbei $w = 0,8795$. Die Maximum Norm für das zusammengesetzte, adaptive Verfahren ist in Abbildung 6.37 dargestellt und der relative Fehler in Abbildung 6.38.

Algorithmus 3 Erweitertes additives Sampling

-
- 1: **for** $j = 1$ to N **do**
 - 2: $\mathbf{X}_j \mapsto$ SVM mit der CODES[®]- Toolbox
 - 3: „max-min“ Sampling $\mapsto \mathbf{X}^{adaptiv}$
 - 4: $\mathbf{X}_j^* = \begin{bmatrix} \mathbf{X}_j \\ \mathbf{X}^{adaptiv} \end{bmatrix}$
 - 5: $\mathbf{X}_j^* \mapsto$ Triangulierung des Randes mittels Konvexer Hülle
 - 6: Schwerpunkte der größten Dreiecke auf dem Rand $\mapsto \mathbf{X}^{adaptiv}$
 - 7: $\mathbf{X}_j^{**} = \begin{bmatrix} \mathbf{X}_j^* \\ \mathbf{X}^{adaptiv} \end{bmatrix}$
 - 8: $\mathbf{X}_j^{**} \mapsto$ *Delaunay* Triangulierung im Gebiet
 - 9: Schwerpunkte der größten Tetraeder im Gebiet $\mapsto \mathbf{X}^{adaptiv}$
 - 10: $\mathbf{X}_j^{***} = \begin{bmatrix} \mathbf{X}_j^{**} \\ \mathbf{X}^{adaptiv} \end{bmatrix}$
 - 11: $\mathbf{X}_j^{***} \mapsto$ LOOCV
 - 12: Trainingspunkte auf Verbindungslinien $\mapsto \mathbf{X}^{adaptiv}$
 - 13: $\mathbf{X}_{j+1} = \begin{bmatrix} \mathbf{X}_j^{***} \\ \mathbf{X}^{adaptiv} \end{bmatrix}$
-

Startgitter \mathbf{X}_0 : $\mathfrak{X}_{3 \times 3 \times 3}$ (27 Trainingspunkte)		
Wichtungsfaktor für LOOCV: $w = 0,8795$		
Sampling Methode	„Innere“ Schleifen	Neue Trainingspunkte
Klassifizierung	$N^{SVM} = 2$	$k^{SVM} = 1$
Schwerpunkte Dreiecke	$N^{Tri} = 1$	$k^{Tri} = 2$
Schwerpunkte Tetraeder	$N^{Tet} = 2$	$k^{Tet} = 2$
LOOCV	$N^{LOOCV} = 4$	$k^{LOOCV} = 4$

Tabelle 6.1: Parameter für das zusammengesetzte adaptive Sampling

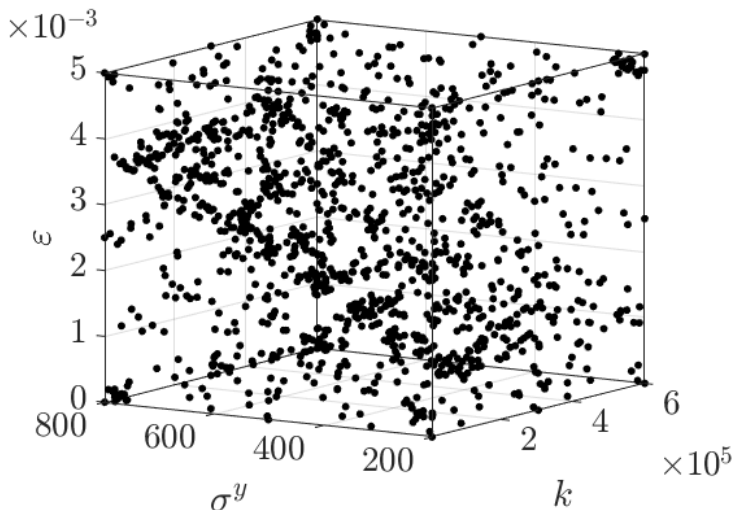


Abbildung 6.36: Trainingspunkte nach $N = 55$ äußeren Schleifendurchläufen des zusammengesetzten Sampling Verfahrens

Bei der Maximum Norm in Abbildung 6.37 wird wieder deutlich, dass das adaptive Sampling im Anfangsbereich der homogenen Verfeinerung überlegen ist. Allerdings tritt auch hier wieder eine Verschlechterung bei einer höheren Anzahl an Trainingspunkten im Vergleich zum homogenen Verfeinern auf. Ab 512 Datenpunkten im kartesischen Gitter $\mathfrak{X}_{8 \times 8 \times 8}$ sticht die blaue Kurve der adaptiven Verfeinerung durch die Kurve der homogenen Verfeinerung hindurch. Der relative Fehler in Abbildung 6.38 verläuft weitestgehend analog zur homogenen Verfeinerung und driftet ab 343 Punkten im kartesischen Gitter $\mathfrak{X}_{7 \times 7 \times 7}$ leicht ab.

Im Vergleich mit der homogenen Verfeinerung zeigen sowohl das sternförmige LOOCV Sampling als auch das zusammengesetzte Sampling im Anfangsbereich wesentliche Vorteile. Als Maßgebend ist hierbei der Kurvenverlauf der Maximum Norm anzusehen, der in Abbildung 6.29 und Abbildung 6.37 jeweils unter der Kurve der homogenen Verfeinerung verläuft. Allerdings ist der Verlauf beider Samplingstrategien „nur“ bis ca. 500 Trainingspunkte der homogenen Verfeinerung überlegen. Ab hier zeigt die homogene Verfeinerung einen besser konvergierenden Verlauf an, während die adaptiven Verfahren stagnieren. Bei der Auswahl der richtigen Parameter aus Tabelle 6.1 können weitere Kombinationen ausprobiert werden, hier besteht noch Forschungsbedarf. Es kann nicht ausgeschlossen werden, dass bessere Ergebnisse als das hier gezeigte Resultat möglich sind.

In Fuhg et al. (2020) werden verschiedene adaptive Samplingmethoden miteinander verglichen, wobei die Modellerstellung hier mit dem sogenannten *Kriging* durchgeführt wird. Bei *Kriging* handelt es sich wie bei SVR um ein effizientes Verfahren um reduzierte bzw. *Surrogate* Modelle auf der Basis von Trainingspunkten zu erstellen. Beide Verfahren haben ihre Vorteile und wurden in zahlreichen Anwendungen getestet, sodass nicht von vornherein eines der genannten Verfahren bevorzugt werden kann. Für das in dieser Arbeit herangezogene eindimensionale elasto-plastische Modell konnten allerdings keine zufriedenstellenden Ergebnisse für *globale* Ersatzmodelle erzielt werden. Bei der homogenen Verfeinerung zeigt *Kriging* hier ein oszillierendes Verhalten im Verlauf der Maximum Norm. Für die *Autokorrelationsfunktion* des *Krigings* wird

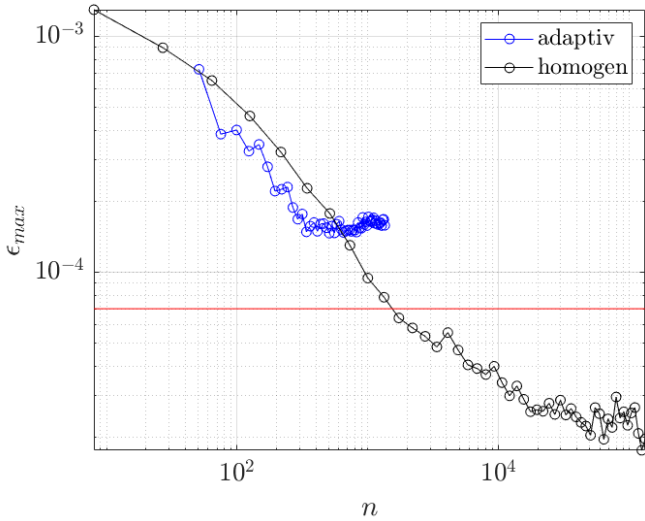


Abbildung 6.37: Maximum Norm ϵ_{max} : Vergleich der homogenen Verfeinerung mit der adaptiven, zusammengesetzten Verfeinerung; (—) Bestes Ergebnis mit der Gauß'schen Kernel Funktion als Vergleich

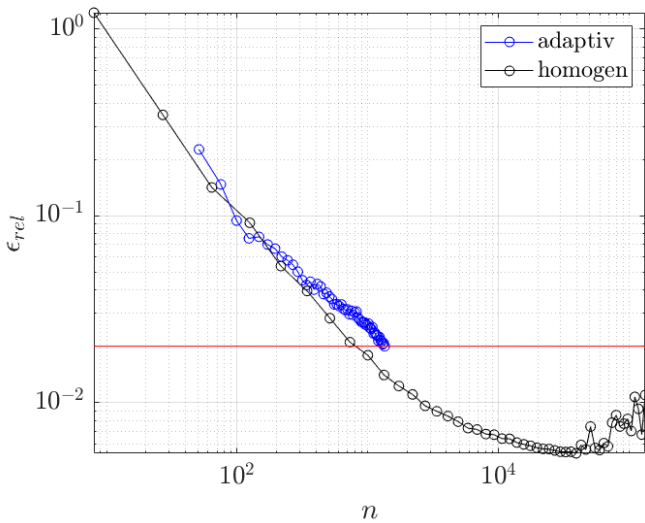


Abbildung 6.38: Relativer Fehler ϵ_{rel} : Vergleich der homogenen Verfeinerung mit der adaptiven, zusammengesetzten Verfeinerung; (—) Bestes Ergebnis mit der Gauß'schen Kernel Funktion als Vergleich

wie bei der *Support Vector Regression* die *Matèrn 5/2* Kernel Funktion verwendet. Ohne auf die Details einzugehen sei erwähnt, das die mathematische Herleitung des Verfahrens auf einem stochastischen Ansatz beruht und für die Erstellung des reduzierten Modells ein Optimierungsproblem in den *Hyperparametern* gelöst werden muss, insbesondere die Korrelationsparameter θ (Fuhg et al., 2020). Die mathematische Formulierung des Verfahrens und des zugehörigen Optimierungsproblems unterscheidet sich deutlich von der *Support Vector Regression*. Der große Bereich mit $\varepsilon^p = 0$ und der C^0 -stetige Übergang in einen glatten nichtlinearen Bereich wirken sich negativ auf die durch *Kriging* erzielten Ergebnisse aus. Bei einer Approximation mittels *Kriging* wird der Einfluss der Trainingspunkte auf den zu approximierenden Trainingspunkt berechnet. Ist das Antwortspektrum in einem Bereich konstant gleich 0, so geht der Einfluss gegen Unendlich. In einem glatten, stetigen und nichtlinearen Bereich nimmt dieser Einfluss endliche Werte an. Dies führt voraussichtlich zu den genannten Oszillationen im Verlauf der Maximum Norm, die somit als ein „Pendeln“ um einen optimalen Mittelwert zwischen den beiden Bereichen angesehen werden können².

6.5 Fazit aus den Studien

Im Kapitel 6 werden wesentliche Eigenschaften der Support Vector Regression (SVR) untersucht. Hierbei liegt der Fokus auf der Approximationsgüte der plastischen Dehnung ε^p des eindimensionalen elasto-plastischen Materialmodells unter Vorgabe der Gesamtdehnung ε , der Fließgrenze σ^y und des Verfestigungsparameters k . Es wird gezeigt, dass mit der *Gauß'schen* Kernel Funktion der „flachen“ Bereich mit $\varepsilon^p = 0$ quantitativ gut approximiert werden kann. Mit der Verwendung der geeigneten Parameter ε_{opt} (*intensive tube width*) und C_{opt} (*penalty term*) können auch qualitativ gute Ergebnisse erzielt werden. Ab einer höheren Anzahl an Trainingspunkten ist allerdings ein Oszillieren und Divergieren in den Fehlerkurven zu beobachten. Eine weitere Verbesserung ist mit der *Matèrn 5/2* Kernel Funktion möglich, die im Gegensatz zur *Gauß'schen* Kernel Funktion bei größeren Anzahlen an Trainingspunkten weiter konvergiert. Das Phänomen der Überanpassung (*engl. Overfitting*) ist ab einer Trainingspunktanzahl von $n = 4 \cdot 10^4$ zu beobachten. Untersuchungen hierzu haben gezeigt, dass ein vollständiges Unterbinden des *Overfittings* weder mit dem weiteren Erhöhen von $C > C_{opt}$ als auch mit der automatischen Anpassung von C_{CVE} unterbunden werden kann. Ingenieurtechnisch gesehen ist der Bereich, in dem ein *Overfitting* aufgrund zu vieler Trainingspunkte stattfindet, nicht von Interesse. Des weiteren werden zwei verschiedene Algorithmen zum adaptiven Sampling vorgestellt. Beide Verfahren zeigen im Anfangsbereich bis ca. $n \approx 500$ Trainingspunkte gute Ergebnisse. Bei größeren Mengen an Trainingspunkten können mit homogenen Gittern bessere Approximationen erzielt werden. Für praktische Zwecke, bei denen nach Möglichkeit mit weniger Trainingspunkten gute Ergebnisse erzielt werden sollen, ist es somit sinnvoll, eine adaptive Verfeinerung vorzunehmen.

²Der Autor dankt an dieser Stelle Jan Niklas Fuhg für den wissenschaftlichen Austausch zu *Kriging* und adaptiven Sampling.

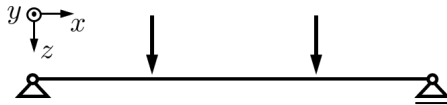
7 Anwendung in der 2D- und 3D-Elasto-Plastizität

In dem vorliegenden Kapitel wird die *Support Vector Regression* auf Daten angewendet, die mittels einer Finite-Elemente-Berechnung generiert werden. Die Erkenntnisse aus Kapitel 6, der 1D-Elasto-Plastizität, können hierfür weiter genutzt und verifiziert werden.

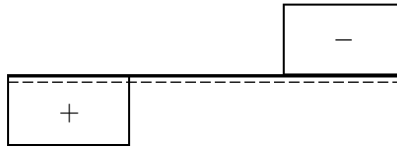
7.1 Vier-Punkt-Biegebalken

Als Modell wird der sogenannte *Vier-Punkt-Biegeversuch* herangezogen. Dabei handelt es sich um einen Balken auf zwei Stützen, der durch zwei gleich große, parallele Kräfte belastet wird. Das statische 1D-Modell, der qualitative Querkraftverlauf sowie der qualitative Momentenverlauf sind in Abbildung 7.1 dargestellt. Eine wesentliche Eigenschaft dieses Modells ist der

Statisches Modell



Querkraftverlauf



Momentenverlauf



Abbildung 7.1: Statisches 1D-Modell des Vier-Punkt-Biegebalkens mit den zugehörigen qualitativen Schnittgrößenverläufen für Querkraft und Moment

querkraftfreie mittlere Bereich zwischen den aufgebrachtten Kräften. Hieraus resultiert ein konstanter Momentenverlauf im mittleren Bereich. Im Rahmen dieser Arbeit wird für das Modell ein rechteckiger Balkenquerschnitt verwendet, wie er in Abbildung 7.2 zu sehen ist. Der Querschnitt hat eine Breite von 10 mm und eine Höhe von 20 mm. Die Gesamtlänge des Balkens beträgt 200 mm, wobei die Lager jeweils 10 mm von den Balkenenden versetzt angeordnet sind. Die Kräfte werden jeweils 60 mm versetzt von den Balkenenden aufgebracht. Eine schematische Darstellung des Vier-Punkt-Biegeversuches ist in Abbildung 7.2 dargestellt. Die Versuchsdurchführung soll im Rahmen dieser Arbeit als verschiebungsgesteuert angesehen werden. Dabei werden die Lager auf der Balkenoberseite in Abbildung 7.2 kontrolliert abgesenkt. Von Interesse ist in der Regel der mittlere Bereich, in dem lediglich das Schnittmoment als innere Belastung zum Tragen kommt. Lokale Spannungsspitzen an den Lager- bzw. Lastpunkten haben aufgrund

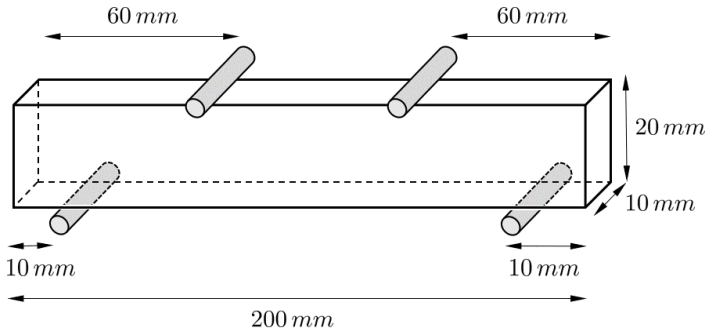


Abbildung 7.2: Vier-Punkt-Biegeversuch mit den zugehörigen Abmessungen und Lagerungen

des Prinzips von de Saint Venant auf den mittleren Balkenbereich keinen Einfluss (Kreißig und Benedix, 2013, Kapitel 3.5). Dies gilt auch für ein FE-Modell, bei dem beispielsweise die Kräfte oder Verschiebungsrandbedingungen an den entsprechenden Stellen Knotenweise aufgebracht werden. Die Normalspannungen in Balkenlängsrichtung verlaufen linear über die Balkenhöhe und nehmen ihre Maximal- und Minimalwerte an der Balkenunter- bzw. oberseite an. Abgesehen von den schon genannten und vernachlässigbaren Spannungsspitzen an den Last- und Lagerpunkten findet somit im mittleren Bereich, bei Überschreiten des elastischen Bereichs, ein kontrolliertes Fließen des Werkstoffes statt.

7.2 Parameterraum und Ausgabegröße (*Quantity of Interest*)

Der Vier-Punkt-Biegeversuch wird für die Ermittlung der Trainings- und Testdaten in ein FE-Modell überführt. Dabei wird auf der Oberseite des Balkens an zwei Stellen eine Verschiebung $\bar{u} > 0$ mm aufgebracht, siehe hierfür Abbildung 7.3. Je nachdem, wie groß diese Verschiebung

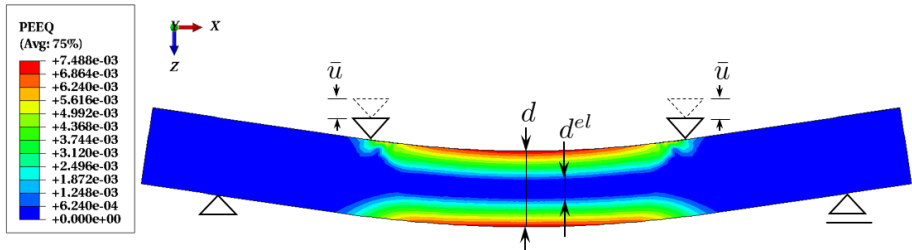


Abbildung 7.3: FE-Berechnung eines Vier-Punkt-Biegeversuches mit schematischer Darstellung der Randbedingungen: Äquivalente plastische Dehnung ε_{eq}^p (PEEQ); eingeprägte Verschiebung \bar{u} [mm]; Balkenhöhe d [mm]; elastischer, nicht plastifizierter Querschnitt d^{el} [mm]

\bar{u} ist, verhält sich das Material rein elastisch oder geht in den elasto-plastischen Bereich über. Das Einsetzen von elasto-plastischem Materialverhalten ist zusätzlich von der Größe der Fließgrenze σ^y und dem Elastizitätsmodul E abhängig. Im Rahmen dieser Arbeit wird ein *isotroper* und *ideal plastifizierender* Stahl als Werkstoff herangezogen. Für die *Support Vector Regression* (SVR) soll bei Vorgabe einer Verschiebung \bar{u} , einer Fließgrenze σ^y und einem Elastizitätsmodul

E eine möglichst genaue Vorhersage für den nicht plastifizierten, elastischen Querschnitt d^{el} gefunden werden, siehe hierzu Abbildung 7.3. Wenn keine plastische Dehnung vorliegt, entspricht der nicht plastifizierte, elastische Querschnitt d^{el} der Balkenhöhe $d^{el} = d$. Die Eingabe und Ausgabegrößen zum Trainieren des reduzierten Modells können in Matrizen wie folgt zusammengefasst werden

$$\begin{aligned} \mathbf{X} &= [\mathbf{x}_1 \cdots \mathbf{x}_n]^T, \quad \mathbf{x}_i = [E_i, \sigma_i^y, \bar{u}_i]^T \\ \mathbf{Y} &= [y_1 \cdots y_n]^T, \quad y_i = d^{el}. \end{aligned} \quad (7.1)$$

Für einen neuen Eingabedatensatz $\mathbf{x}^* = [E, \sigma^y, \bar{u}]^T$ soll das reduzierte Modell \mathcal{F}^{RM} in der Lage sein, den entsprechenden elastischen Querschnitt $\hat{y} = d^{el}$ vorauszusagen,

$$\mathcal{F}^{RM}(\mathbf{x}^*) \mapsto \hat{y}. \quad (7.2)$$

Die Bereiche (*engl. Domain*) für die Trainingsdaten $\mathbf{x}_i = [E_i, \sigma_i^y, \bar{u}_i]^T$ werden im Hinblick auf die Materialeigenschaften von Baustählen folgendermaßen eingegrenzt,

$$\begin{aligned} 190\,000 \text{ MPa} &\leq E \leq 230\,000 \text{ MPa} \\ 300 \text{ MPa} &\leq \sigma^y \leq 360 \text{ MPa} \\ 0,4 \text{ mm} &\leq \bar{u} \leq 2,8 \text{ mm}, \end{aligned} \quad (7.3)$$

wobei der Bereich für die aufgebrachte Verschiebung \bar{u} die Betrachtung im Rahmen *kleiner* Deformationen rechtfertigt. In einer FE-Analyse liegt der nicht plastifizierte, elastische Querschnitt d^{el} , in diesem Fall die Größe von Interesse (*engl. Quantity of Interest*), nicht primär nach der EE-Berechnung vor. Hierfür werden im Rahmen dieser Arbeit zwei Nachlaufberechnungen benötigt (*engl. Postprocessing*). Im ersten Schritt wird die äquivalente plastische Dehnung ε_{eq}^p berechnet,

$$\varepsilon_{eq}^p := \int_0^t \sqrt{\frac{2}{3}} \|\dot{\varepsilon}^p\| d\tau = \sqrt{\frac{2}{3}} \varepsilon^p \cdots \varepsilon^p, \quad (7.4)$$

die als skalare Vergleichsgröße für die vorliegende plastische Dehnung angesehen werden kann. In ABAQUS[®] wird diese Größe mit PEEQ bezeichnet und ist exemplarisch in Abbildung 7.3 als Farbplot dargestellt. Für den Ort des nicht plastifizierten, elastischen Querschnitts d^{el} wird die Mitte des Balkens gewählt, siehe hierzu Abbildung 7.3. Ausgehend von der Gesamthöhe des Balkens d kann dann d^{el} wie folgt definiert werden,

$$d^{el} = d (\varepsilon_{eq}^p = 0). \quad (7.5)$$

Im folgenden Kapitel 7.3 wird beschrieben, wie diese Größe im Rahmen der FEM angenähert wird.

7.3 Modellaufbau in 3D

Die Modellierung des Vier-Punkt-Biegebalkens findet in ABAQUS[®] CAE statt, der grafischen Benutzeroberfläche dieser Finite-Elemente-Software. Bei der Modellierung werden Symmetrieeigenschaften ausgenutzt, sodass nur ein Viertel des Balkens modelliert werden muss. Der modellierte Teil des Balkens sowie die Orientierung und der Ursprung des Koordinatensystems sind in Abbildung 7.4 schematisch dargestellt. Die Verschiebungsrandbedingungen für das 3D-Modell in ABAQUS[®] sind in Abbildung 7.5 abgebildet. An der yz-Ebene wird die Verschiebung in x-Richtung gesperrt und an der xz-Ebene die Verschiebung in y-Richtung. Die restlichen Richtungen sind in diesen beiden Ebenen frei beweglich, sodass keine Zwängungen entstehen. Auf

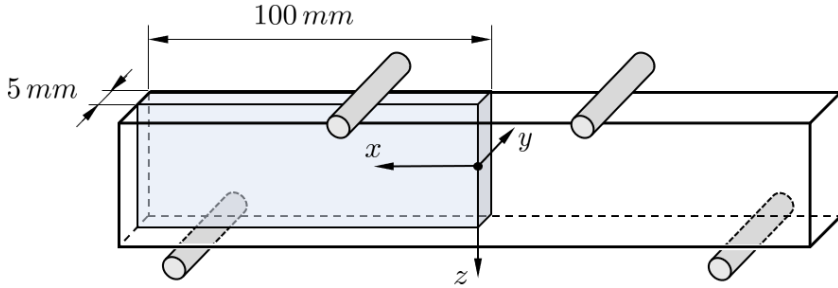


Abbildung 7.4: Modellierung des Vier-Punkt-Biegeversuches unter Ausnutzung der Symmetriebedingungen bezüglich der zy -Ebene und xy -Ebene: Viertel Balken

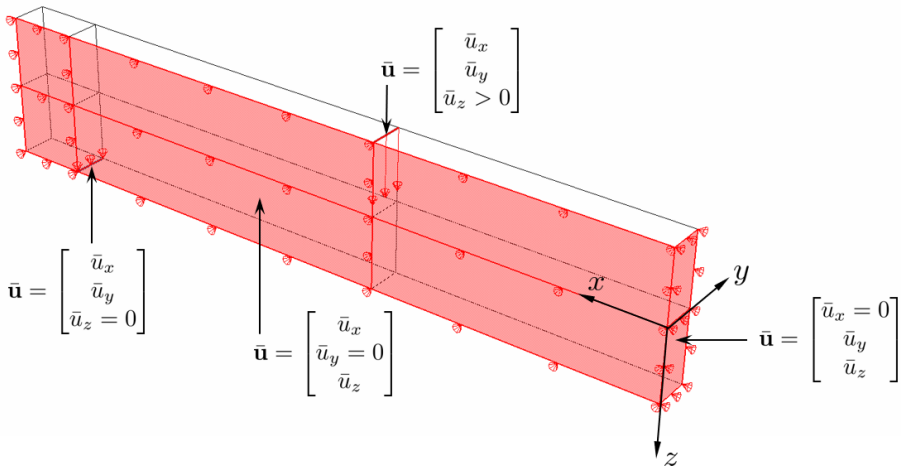


Abbildung 7.5: Verschiebungsrandbedingungen $\bar{\mathbf{u}}$ des 3D-Modells für den Vier-Punkt-Biegebalken

der Balkenunterseite wird die Verschiebung auf einer Partitionslinie in z -Richtung gesperrt. Die anderen Richtungen sind auch hier frei beweglich, sodass ein Loslager abgebildet werden kann. Auf der Balkenoberseite wird eine Verschiebung $\bar{u} > 0$ mm auf einer Partitionslinie in z -Richtung aufgebracht, wodurch eine Verschiebungssteuerung im Rahmen der Finite-Elemente-Methode realisiert wird. Im Rahmen der Untersuchungen werden zwei verschiedene Vernetzungen herangezogen. Die Vernetzungsdichte an der Kante, die mit der zentralen z -Achse zusammenfällt, ist für die Qualität der hoch aufgelösten Berechnung d^{el} (engl. *High-Fidelity-Solution*) von entscheidender Bedeutung. An dieser Kante wird die für die Berechnung von d^{el} relevante äquivalente plastische Dehnung ε_{eq}^p abgegriffen, da hier der maximal belastete Querschnitt des Vier-Punkt-Biegeversuchs vorliegt. Nach einer konvergierten FE-Berechnung liegen die Werte für die plastischen Dehnungen an den Integrationspunkten der Elemente vor. Somit ist die Anzahl der Elemente über die Balkenhöhe d für die Präzision des nicht plastifizierten, elastischen Querschnitts d^{el} von entscheidender Bedeutung. In Abbildung 7.6 ist eine Vernetzung mit 40 Elementen über die Höhe im vorderen Bereich dargestellt. In Abbildung 7.7 ist eine weitere Ver-

feinerung der Vernetzung mit 80 Elementen über die Höhe dargestellt. Bei beiden FE-Netzen ist ein Großteil der Geometrie mit einem groben, strukturierten Netz versehen, da der entsprechende Bereich von untergeordneter Bedeutung ist. Ausgehend von diesem grob vernetzten Bereich wird das Netz zum vorderen Bereich des Modells hin verfeinert, sodass jeweils die gewünschte Elementanzahl von 40 bzw. 80 Elementen über die Höhe erreicht wird. Für die Elementierung wird das in Abbildung 4.1, Kapitel 4.3 vorgestellte 8-Knoten Kontinuumsselement verwendet. In ABAQUS® ist dieser Elementtyp mit C3D8 deklariert (*8-node linear brick*). Die äquivalente plastische Dehnung ε_{eq}^p wird an den 40 bzw. 80 Elementen abgegriffen, bei denen eine Elementkante mit der zentralen z-Achse zusammenfällt (siehe hierzu Abbildung 7.5 und 7.10). Es ist zweckmäßig, den Wert für ε_{eq}^p je Element am Elementschwerpunkt zu ermitteln. In ABAQUS® wird der Wert im Elementschwerpunkt aus den Werten an den Integrationspunkten des Elements interpoliert. Der Zusammenhang zwischen den Elementknoten, den Integrationspunkten und den Elementschwerpunkten ist in Abbildung 7.8 dargestellt. Der Abstand zwischen den Elementschwerpunkten zweier benachbarter Elemente stimmt für die betrachteten Elemente ungefähr mit der Elementkantenlänge überein. Somit kann bei Kenntnis von ε_{eq}^p im Schwerpunkt entschieden werden, ob dieses Element bzw. die zugehörige Elementkantenlänge dem rein elastischen oder dem elasto-plastischen Bereich zugeschrieben werden kann. Die Berechnung des nicht plastifizierten, elastischen Querschnitts d^{el} kann somit wie folgt angegeben werden,

$$d^{el} = \left(\sum \text{Element} \Big|_{\varepsilon_{eq}^p=0} \right) l_e, \quad (7.6)$$

wobei l_e für die Elementkantenlänge eines Elementes im betrachteten Bereich steht. Aus der Gleichung (7.6) wird ersichtlich, dass die Güte der hoch aufgelösten FE-Berechnung mit der Elementgröße korreliert. In der Abbildung 7.9 ist exemplarisch ein Ergebnis-Farbplot für ε_{eq}^p (PEEQ) und eine Vernetzung mit 40 Elementen über die Höhe dargestellt. Es ist gut zu erkennen, wie sich die plastischen Zonen von Balkenober- und unterseite aus annähernd symmetrische zur Spannungsnulllinie im Querschnitt ausbreiten. Die für den nicht plastifizierten, elastischen Querschnitt d^{el} relevanten Elemente im Zentrum an der globalen z-Achse können in einem Element-Pfad zusammengefasst werden. Speziell für das Berechnungsergebnis aus Abbildung 7.9 ist die äquivalente plastische Dehnung ε_{eq}^p entlang des Element-Pfades in Abbildung 7.10 dargestellt. Aus der Abbildung 7.10 kann entnommen werden, dass für die 6 Elemente 18 bis 23 ein Wert von $\varepsilon_{eq}^p = 0$ vorliegt. Bei 40 Elementen über die Höhe kann die Elementkantenlänge mit $l_e = 20 \text{ mm}/40 = 0,5 \text{ mm}$ angegeben werden. Die Auswertung von Gleichung (7.6) ergibt somit einen nicht plastifizierten, elastischen Querschnitt d^{el} von $6l_e = 3 \text{ mm}$. Für die Vernetzung mit 80 Elementen kann die Elementlänge mit $l_e = 20 \text{ mm}/80 = 0,25 \text{ mm}$ angegeben werden. Die Elementkantenlänge ist somit als maximale Präzision der hoch aufgelösten FE-Berechnung anzusehen.

7.4 Test- und Trainingsdaten für das reduzierte Modell in 3D

Für die Test- und Trainingsdatenstruktur der *Support Vector Regression* werden kartesische Gitter verwendet. Die Punkte werden hierfür in den Bereichen aus Gleichung (7.3) äquidistant verteilt und im Anschluss zu einem kartesischen Gitter kombiniert. Die Erstellung der Test- und Trainingsdaten erfolgt mittels einer *Parameterstudie*, für die in ABAQUS® eine spezielle Syntax im Rahmen von Python-Skripten vorgesehen ist. Um sicherzustellen, dass die Test- und Trainingspunkte nicht zusammenfallen, werden zwei kartesische Gitter mit verschiedenen Dichten erstellt

$$\begin{aligned} \mathfrak{X}_{20 \times 20 \times 20} &= \{E_i\}_{i=1 \dots 20} \times \{\sigma_i^y\}_{i=1 \dots 20} \times \{\bar{u}_i\}_{i=1 \dots 20} \\ \mathfrak{X}_{19 \times 19 \times 19} &= \{E_i\}_{i=1 \dots 19} \times \{\sigma_i^y\}_{i=1 \dots 19} \times \{\bar{u}_i\}_{i=1 \dots 19} \\ &\times : \text{kartesisches Produkt.} \end{aligned} \quad (7.7)$$

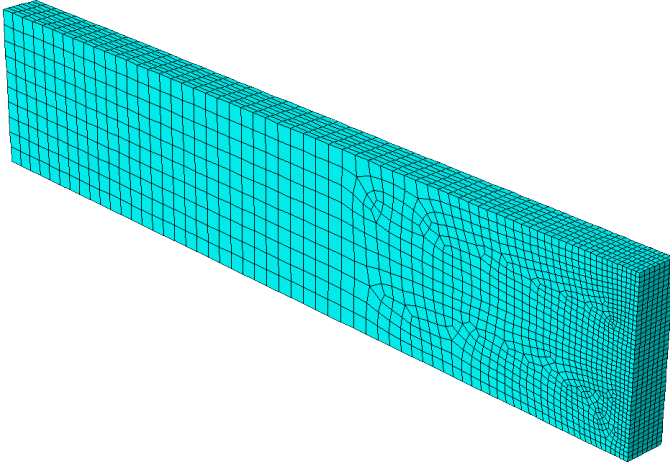


Abbildung 7.6: Vernetzung des 3D-Modells zum Vier-Punkt-Biegeversuch: 40 Elemente über der Balkenhöhe im Zentrum; Elementtyp: 8-Knoten Kontinuumsselement (C3D8); Anzahl der Elemente: 8953

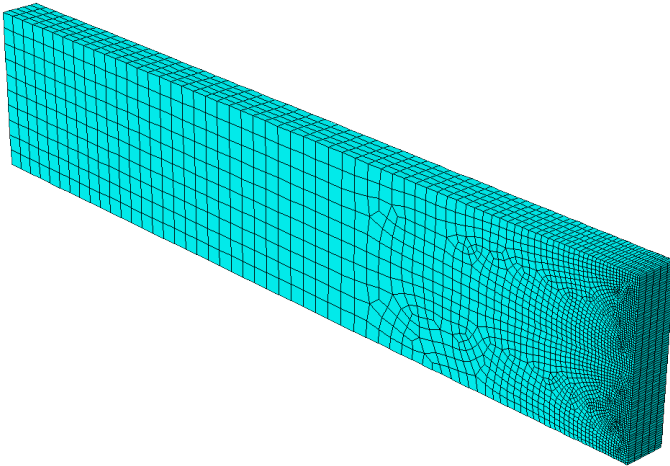


Abbildung 7.7: Vernetzung des 3D-Modells zum Vier-Punkt-Biegeversuch: 80 Elemente über der Balkenhöhe im Zentrum; Elementtyp: 8-Knoten Kontinuumsselement (C3D8); Anzahl der Elemente: 11985

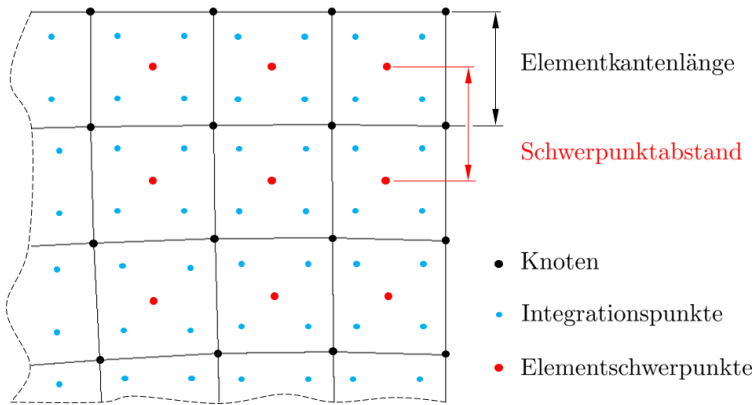


Abbildung 7.8: Schematische Darstellung der Elementknoten, Integrationspunkte und Elementschwerpunkte

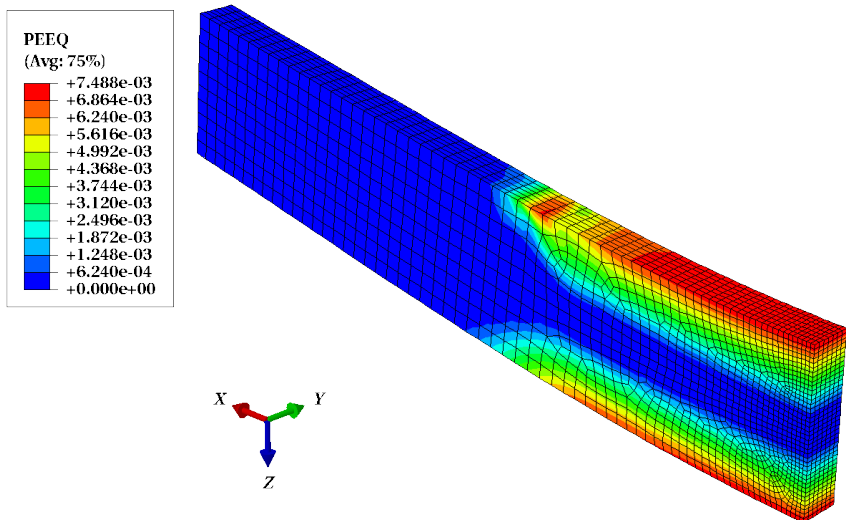


Abbildung 7.9: Farbplot für ε_{eq}^p (PEEQ) nach einer konvergierten FE-Berechnung für die Vernetzung mit 40 Elementen über die Höhe: Aufgebrachte Verschiebung $\bar{u}_z = 2$ mm, Fließgrenze $\sigma^y = 360$ MPa, Elastizitätsmodul $E = 210\,000$ MPa, Querkontraktionszahl $\nu = 0,3$, ideal plastisches Materialverhalten

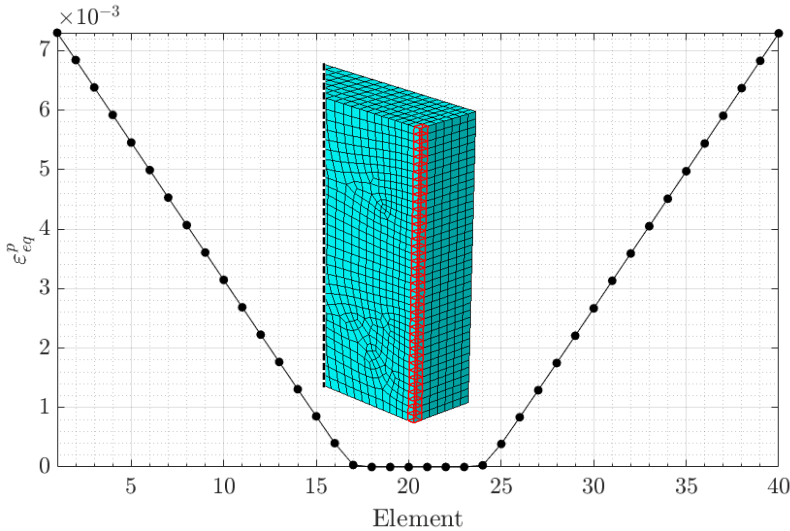


Abbildung 7.10: Äquivalente Plastische Dehnung ε_{eq}^p (PEEQ) für den Element-Pfad ■ nach einer konvergierten FE-Berechnung für die Vernetzung mit 40 Elementen über die Höhe: Aufgebrachte Verschiebung $\bar{u}_z = 2$ mm, Fließgrenze $\sigma^y = 360$ MPa, Elastizitätsmodul $E = 210\,000$ MPa, Querkontraktionszahl $\nu = 0,3$, ideal plastisches Materialverhalten

Das Gitter $\mathfrak{X}_{20 \times 20 \times 20}$ mit 8000 Datenpunkten wird als Testgitter verwendet, welches es mithilfe der SVR zu approximieren gilt. Das um eine Stufe gröbere Gitter $\mathfrak{X}_{19 \times 19 \times 19}$ mit 6859 Datenpunkten wird für die Trainingsphase der Support Vector Regression verwendet. Bei beiden Gittern fallen lediglich die $2^3 = 8$ Eckpunkte der Samplingdomain (7.3) zusammen. In Abbildung 7.11 sind die beiden disjunkten Datengitter für die Test- und Trainingspunkte dargestellt. Die Darstellung des zugehörigen nicht plastifizierten, elastischen Querschnitts $d^{el}(E, \sigma^y, \bar{u})$ ist mittels Farbplot möglich. In Abbildung 7.12 sind die Farbplots für das $\mathfrak{X}_{20 \times 20 \times 20}$ Testgitter dargestellt. Auf der linken Seite ist das Ergebnis für eine Netzdichte von 40 Elementen über die Balkenhöhe zu sehen. Auf der rechten Seite ist der Farbplot für das feinere Netz mit 80 Elementen über die Balkenhöhe dargestellt. In beiden Farbplots ist gut zu erkennen, dass bei einer niedrigen Fließgrenze und einem hohen Elastizitätsmodul der elastische, nicht plastifizierte Querschnitt d^{el} bei hohen Verschiebungsmagnituden sehr gering ist (dunkelblaue Färbung in der oberen rechten Ecke). Im unteren Bereich ist überwiegend der „volle“ elastische, nicht plastifizierte Querschnitt $d^{el} \approx d$ vorhanden (gelbe Färbung im unteren Bereich). Die Übergänge zwischen den Farbbereichen sind nichtlinear und stufenförmig. Für einen besser zu interpretierenden Vergleich der Ergebnisse beider Vernetzungsdichten ist es sinnvoll, einzelne Ergebnisflächen gegenüberzustellen. Dabei wird der nicht plastifizierte, elastische Querschnitt für eine konstante Verschiebung $d^{el}|_{\bar{u}=const.}$ über der Fließgrenze σ^y und dem Elastizitätsmodul E aufgetragen. Die Ergebnisfläche für die minimale Verschiebung von $\bar{u} = 0,4$ mm ist in Abbildung 7.13 dargestellt. Hier ist deutlich zu erkennen, dass der elastische, nicht plastifizierte Querschnitt vorwiegend dem vollen Querschnitt entspricht: $d^{el} = d$. Lediglich im Bereich einer niedrigen Fließgrenze und einem hohen Elastizitätsmodul fängt das Material bei dieser geringen Verschiebung an zu plastifizieren. Das feinere Netz mit 80 Elementen über die Höhe ist hier bereits in der Lage, eine feinere Auflösung zu erzielen, was an der „Zwischenstufe“ bei $d^{el} = 19,5$ mm im rechten Bild 7.13 zu erkennen ist. Noch deutlicher wird dieser Unterschied

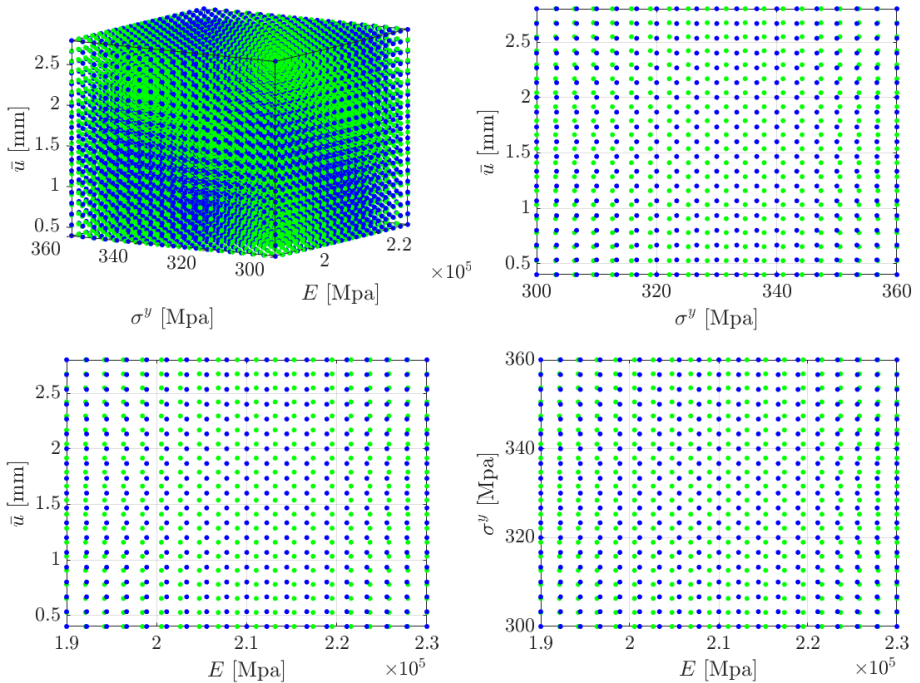


Abbildung 7.11: Testpunkte $\mathfrak{X}_{20 \times 20 \times 20}$ (●), Trainingspunkte $\mathfrak{X}_{19 \times 19 \times 19}$ (●): Isometrische Ansicht (oben links), \bar{u} σ^y –Ebene (oben rechts), \bar{u} E –Ebene (unten links), σ^y E –Ebene (unten rechts)

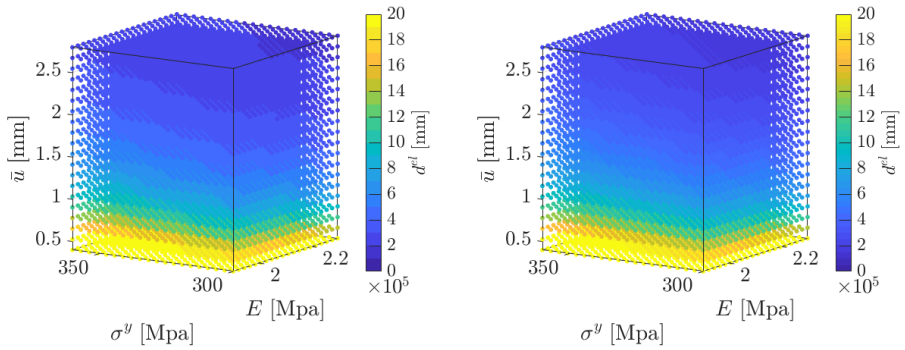


Abbildung 7.12: Testgitter $\mathfrak{X}_{20 \times 20 \times 20}$ mit Ergebnis-Farbplot für den nicht plastifizierten, elastischen Querschnitt d^{el} : 40 Elemente über die Balkenhöhe (links), 80 Elemente über die Balkenhöhe (rechts)

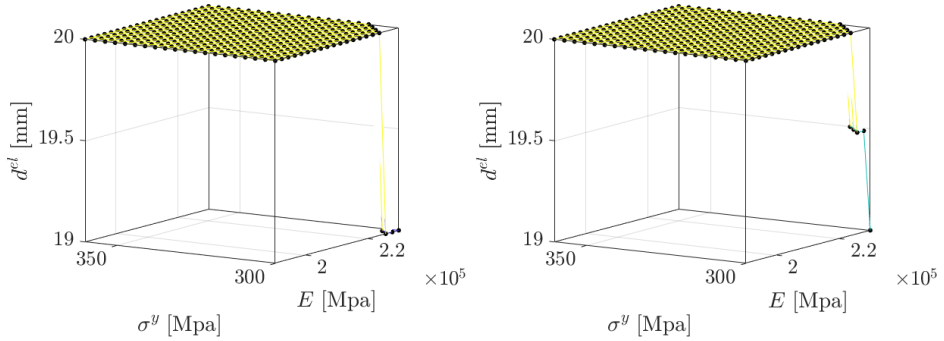


Abbildung 7.13: Nicht plastifizierter, elastischer Querschnitt $d^{el}|_{\bar{u}=0,4 \text{ mm}}$ für die minimale Verschiebung: 40 Elemente über die Balkenhöhe (links), 80 Elemente über die Balkenhöhe (rechts)

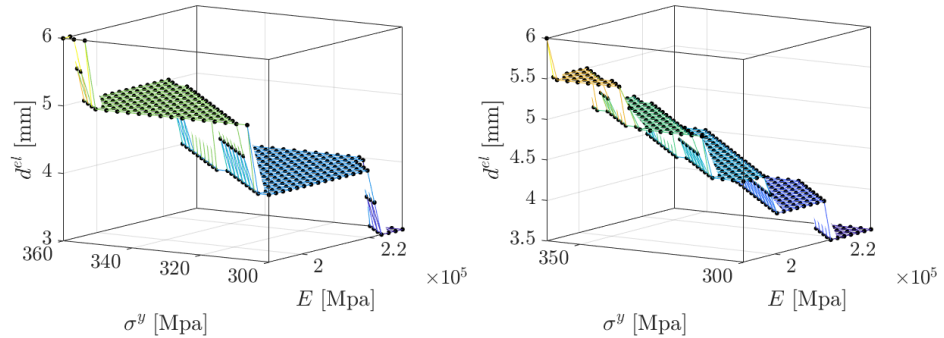


Abbildung 7.14: Nicht plastifizierter, elastischer Querschnitt $d^{el}|_{\bar{u}=1,5368 \text{ mm}}$ für eine mittlere Verschiebung: 40 Elemente über die Balkenhöhe (links), 80 Elemente über die Balkenhöhe (rechts)

zwischen den verschiedenen Vernetzungsdichten in Abbildung 7.14 für eine Verschiebung von $\bar{u} = 1,5368 \text{ mm}$. Die entspricht ungefähr dem mittleren Wert des Verschiebungsbereiches (*Domain*) von $0,4 \text{ mm} \leq \bar{u} \leq 2,8 \text{ mm}$. Mit der feineren Vernetzung ist es möglich, die resultierenden elastischen Querschnittshöhen d^{el} feiner aufzulösen, was aus der Anzahl der „Stufen“ in der rechten Abbildung 7.14 deutlich wird. Die *Stufenhöhe* für die gröbere Vernetzung beträgt 1 mm während mit der feinen Vernetzung 0,5 mm realisiert werden können. Für die maximale Verschiebung von $\bar{u} = 2,8 \text{ mm}$ sind die Ergebnisse beider Vernetzungsdichten in Abbildung 7.15 abgebildet. Hier ist ebenfalls wieder deutlich der Zusammenhang zwischen der Anzahl der Elemente über die Höhe und der Auflösungsgüte von d^{el} erkennbar. Es lässt sich erahnen, dass bei zunehmender Vernetzungsdichte die Anzahl der Stufen in d^{el} für das hoch-aufgelöste FE-Ergebnis steigt. In der Praxis kann eine sehr feine Auflösung vielfach nicht realisiert werden, weshalb das reduzierte Modell in der Lage sein muss, auch bei einer gröberen „Stufenstruktur“ der Trainingsdaten hinreichend genaue Vorhersagen zu treffen.

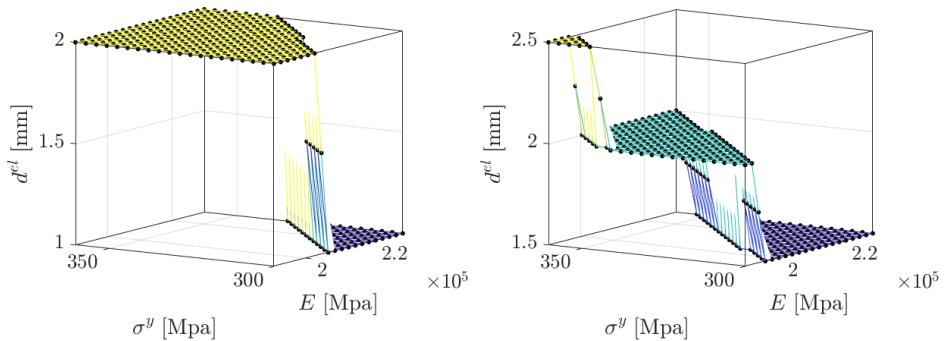


Abbildung 7.15: Nicht plastifizierter, elastischer Querschnitt $d^{el}|_{\bar{u}=2,8 \text{ mm}}$ für die maximale Verschiebung: 40 Elemente über die Balkenhöhe (links), 80 Elemente über die Balkenhöhe (rechts)

7.5 Ergebnisse der Support Vector Regression in 3D

Das reduzierte Modell wird beginnend mit einem groben Trainingsgitter mit immer dichter werdenden Gittern trainiert bzw. erstellt. Dabei wird das Gitter $\mathfrak{X}_{19 \times 19 \times 19}$ herangezogen, aus dem dann „äquidistant“ Punkte in kartesischer Gitterform zum Trainieren entnommen werden. Die äquidistanten Abstände in den Parameter-Richtungen der entnommenen Gitter können nicht in jedem Schleifendurchlauf „exakt“ eingehalten werden. Dies liegt an der Entnahme der Trainingsdatenpunkte aus einer festen Spanne von 19 Punkten des finalen Trainingsgitters in Abhängigkeit eines *Schleifenzählers* $i \in \mathbb{N}_+$. Hier hängt es davon ab, ob i grade oder ungerade ist und sich dementsprechend äquidistant in der Spanne $\{1 \cdots 19\} \in \mathbb{N}_+$ aufteilen lässt. Der verwendete Extraktions-Algorithmus sorgt aber dafür dass die Trainingspunkte in jedem Fall „näherungsweise“ äquidistant verteilt sind¹. Das gröbste Gitter besteht dabei lediglich aus den $2^3 = 8$ Eckpunkten des Samplingbereiches (7.3). Das feinste Gitter zum Trainieren der Support Vector Regression ist dann das $\mathfrak{X}_{19 \times 19 \times 19}$ Gitter selbst. Für jedes Trainingsgitter wird eine Approximation für das $\mathfrak{X}_{20 \times 20 \times 20}$ Testgitter durchgeführt. Die Fehlermessung erfolgt wie in Kapitel 6 zur 1D-Elasto-Plastizität. Das Ergebnis der Approximation $\hat{y}_i = \tilde{d}_i^{el}$ mittels SVR soll das Ergebnis der hoch aufgelösten FE-Berechnung $y_i = d_i^{el}$ möglichst gut abbilden, sodass der absolute Fehler mit der Maximum-Norm folgendermaßen angegeben werden kann

$$\begin{aligned} e_i &= |y_i - \hat{y}_i| \\ \epsilon_{max} &:= \max\{e_1 \cdots e_n\}. \end{aligned} \quad (7.8)$$

Der relative Fehler

$$\begin{aligned} \tilde{y} &= \frac{\sum_{i=1}^n y_i}{n} \\ \epsilon_i &= \frac{|y_i - \hat{y}_i|}{|\tilde{y}|} \\ \epsilon_{rel} &= \frac{\sum_{i=1}^n \epsilon_i}{n} \end{aligned} \quad (7.9)$$

¹Für die *qualitativen* Aussagen in dieser Arbeit spielt es nur eine untergeordnete Rolle, ob die Trainingspunkte „strenge“ äquidistant verteilt sind. Die Anzahl der Trainingspunkte in den Parameter-Richtungen bei ungefähr gleichmäßiger Verteilung ist wesentlich entscheidender, was in Kapitle 7.9 gezeigt wird

gibt einen Überblick, wie gut die Testpunkte insgesamt abgebildet werden. Die Anzahl der Testpunkte ist wieder mit n gekennzeichnet und beträgt im $\mathfrak{X}_{20 \times 20 \times 20}$ Gitter 8000. Der absolute Fehler (7.9) ist im Gegensatz zum eindimensionalen Fall aus Kapitel 6 in diesem Anwendungsfall, für den elastischen Querschnitt d^{el} , mit der Einheit [mm] behaftet. Die Parameter ε (*intensive tube width*) und C (*penalty term*) für die Support Vector Regression werden entsprechend der Gleichung (6.14) bzw. Gleichung (6.15) berechnet, da mit diesen Werten in Kapitel 6 gute Erfahrungen gemacht wurden. Der Verlauf des *absoluten* Fehlers ϵ_{max} für die Vernetzungsdichte mit 40 Elementen über die Balkenhöhe ist in Abbildung 7.16 dargestellt. Der entsprechende Verlauf des relativen Fehlers ϵ_{rel} ist in Abbildung 7.17 abgebildet. In den

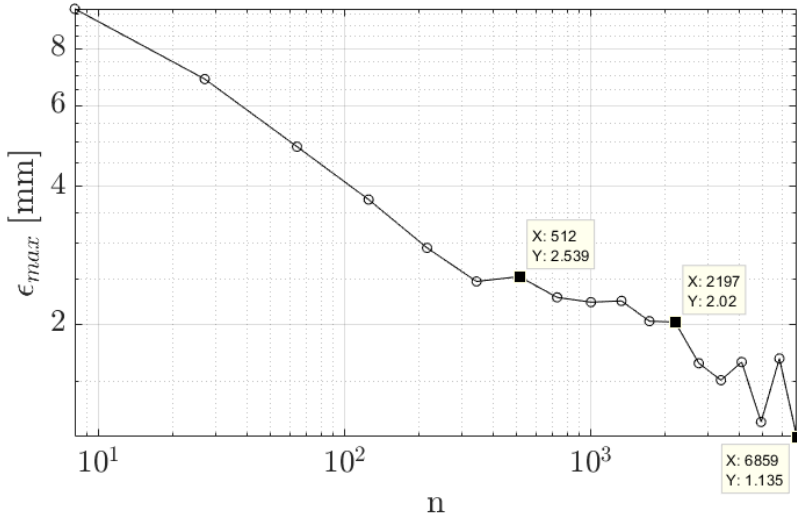


Abbildung 7.16: Maximum Norm ϵ_{max} für die Vernetzung mit 40 Elementen über die Balkenhöhe; Matèrn 5/2 Kernel Funktion; Parameter $C_{opt} = iqr(\mathbf{Y}) / 0,001$ und $\varepsilon = iqr(\mathbf{Y}) / 200$; n : Anzahl der Trainingspunkte im Gitter; zweifach logarithmische Darstellung

Abbildungen sind zusätzlich markante Ergebnisse mit genauer Angabe der Werte hervorgehoben. Aus dem Verlauf des absoluten Fehlermaßes in Abbildung 7.16 wird deutlich, dass die bestmögliche Approximation mit den $n = 6859$ Trainingspunkten des vollen $\mathfrak{X}_{19 \times 19 \times 19}$ Trainingsgitters nicht unter den Wert von 1,135 mm kommt. Der Wert von 1 mm kann als grober Richtwert für die mögliche Auflösung des FE-Modells mit 40 Elementen über die Höhe angesehen werden. Bei annäherungsweise symmetrischer Plastifizierung, von der Balkenober- und unterseite ausgehend, ist bei einer Elementlänge von 0,5 mm kein besseres Ergebnis als $2 \cdot 0,5 \text{ mm} = 1,0 \text{ mm}$ zu erwarten. Auffällig ist zudem das Oszillieren im hinteren Bereich der Kurve für ϵ_{max} . Die Oszillation um den Wert von $\epsilon_{max} \approx 1,5 \text{ mm}$ kann auf den genannten Einfluss der Vernetzungsdichte zurückgeführt werden und verhindert hier eine weitere Konvergenz. Der Verlauf des *relativen* Fehlers ϵ_{rel} in Abbildung 7.17 zeigt einen stabilen Verlauf. Der Wert von $\epsilon_{rel} = 0,038 \approx 4\%$ bei $n = 6859$ Trainingspunkten ist für praktische Anwendungen mehr als ausreichend. Relative Fehler $\epsilon_{rel} < 10\%$ werden schon ab $n = 10^2$ Trainingspunkten erreicht. Der Verlauf des *absoluten* Fehlers ϵ_{max} für die Vernetzungsdichte mit 80 Elementen über die Balkenhöhe ist in Abbildung 7.18 dargestellt. Der entsprechende Verlauf des relativen Fehlers ϵ_{rel} ist in Abbildung 7.19 abgebildet. Bei der Vernetzung mit 80 Elementen über

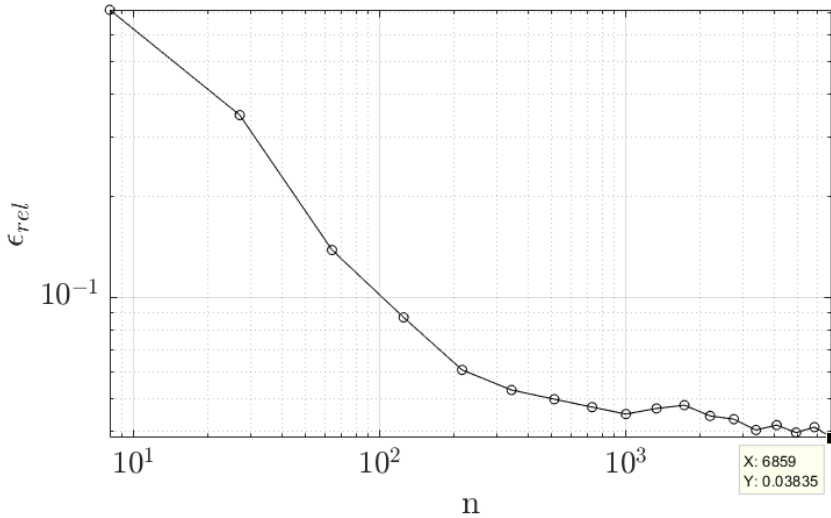


Abbildung 7.17: Relativer Fehler ϵ_{rel} für die Vernetzung mit 40 Elementen über die Balkenhöhe: Matérn 5/2 Kernel Funktion; Parameter $C_{opt} = iqr(\mathbf{Y})/0,001$ und $\varepsilon = iqr(\mathbf{Y})/200$; n : Anzahl der Trainingspunkte im Gitter; zweifach logarithmische Darstellung

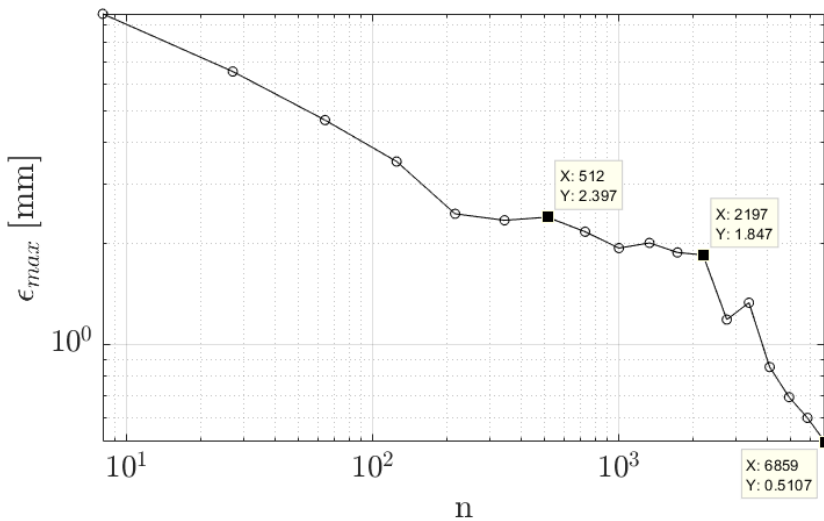


Abbildung 7.18: Maximum Norm ϵ_{max} für die Vernetzung mit 80 Elementen über die Balkenhöhe: Matérn 5/2 Kernel Funktion; Parameter $C_{opt} = iqr(\mathbf{Y})/0,001$ und $\varepsilon = iqr(\mathbf{Y})/200$; n : Anzahl der Trainingspunkte im Gitter; zweifach logarithmische Darstellung

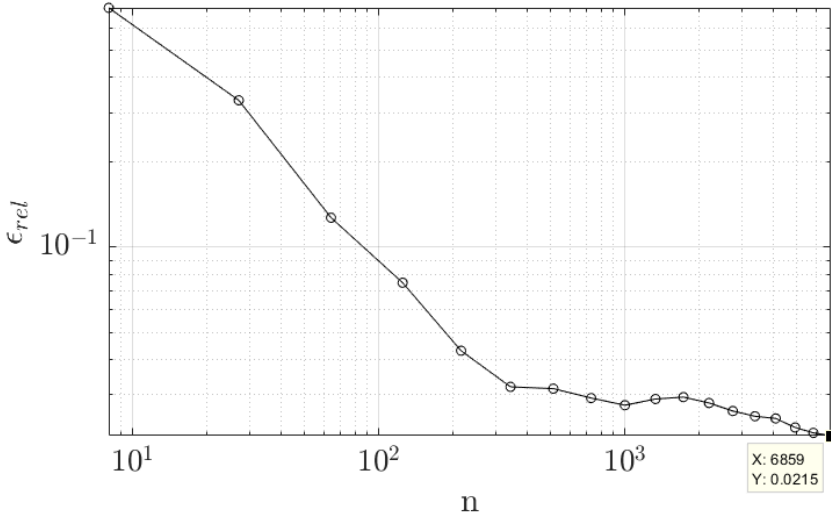


Abbildung 7.19: Relativer Fehler ϵ_{rel} für die Vernetzung mit 80 Elementen über die Balkenhöhe: Matèrn 5/2 Kernel Funktion; Parameter $C_{opt} = iqr(\mathbf{Y})/0,001$ und $\epsilon = iqr(\mathbf{Y})/200$; n : Anzahl der Trainingspunkte im Gitter; zweifach logarithmische Darstellung

die Balkenhöhe kann ein *absoluter* Fehler von $\epsilon_{max} = 0,5107$ mm mit $n = 6859$ Trainingspunkten erreicht werden. Die *absolute* Fehler von $0,5$ mm = $2 \cdot 0,25$ mm korreliert auch hier wieder mit der Elementkantenlänge von $l_e = 0,25$ mm. Ab $n = 3000$ Trainingspunkten ist in Abbildung 7.18 zudem eine Konvergenz von ϵ_{max} zu beobachten. Dies kann ebenfalls auf die höhere Vernetzungsdichte mit 80 Elementen über die Balkenhöhe zurückgeführt werden. Für die meisten praktischen Anwendungen im Ingenieurwesen ist die Vorhersage des nicht plastifizierten, elastischen Querschnitts d^{el} mit einer Abweichung von $< 1,0$ mm bei einer Balkenhöhe von $d = 20$ mm ausreichend. Diese Werte werden schon ab ca. $n = 4000$ Trainingspunkten in Abbildung 7.18 erreicht. Der relative Fehler in Abbildung 7.19 nimmt für die feine Vernetzung schon ab $n = 300$ Trainingspunkten Werte von $\epsilon_{rel} < 4\%$ an. Der Fehler bei $n = 6859$ beträgt hier sogar $0,0215 \approx 2\%$, was für praktische Anwendungen in den meisten Fällen ausreichend ist.

7.6 Modellaufbau in 2D (Ebener Spannungszustand)

Die Belastung des *Vier-Punkt-Biegeversuchs* aus Kapitel 7.1 erfolgt ausschließlich in der xz -Ebene. Die Ausdehnung des Balkens in die z -Richtung ist zudem nicht durch Randbedingungen unterbunden oder beeinträchtigt. Somit ist es mit ausreichender Genauigkeit möglich, den *Vier-Punkt-Biegeversuch* als ebenen Spannungszustand in einem zweidimensionalen Modell abzubilden. Hierfür wird im Rahmen dieser Arbeit ein ABAQUS® Modell in der xy -Ebene erstellt. Die Breite von 10 mm wird für die Berechnung lediglich als Parameter übergeben. Das 2D-Modell des *Vier-Punkt-Biegebalkens* ist in Abbildung 7.20 dargestellt, wobei aufgrund der Symmetriebedingungen in 2D nur ein halber Balken modelliert werden muss. Die Verschiebung des halben Balkens in x -Richtung wird an der zentralen Symmetriekante unterbunden. Die y -

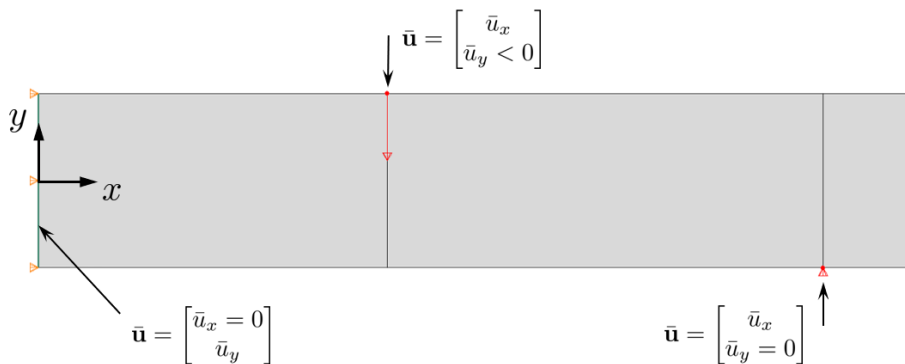


Abbildung 7.20: Modellierung des Vier-Punkt-Biegeversuches als ebener Spannungszustand (2D-Modell) in der xy -Ebene unter Ausnutzung der Symmetriebedingungen: Halber Balken

Richtung ist hier frei verschieblich, sodass eine Durchbiegung in y -Richtung gewährleistet wird. An der Balkenunterseite wird die Verschiebung in y -Richtung am Auflager unterbunden, die Verschiebung in x -Richtung ist hier frei, um Zwängungen zu vermeiden. An der Balkenoberseite wird die Verschiebungssteuerung mit $\bar{u}_y < 0$ realisiert. Auch hier ist wieder die Verschiebung in x -Richtung frei. Die Vernetzung erfolgt mit bilinearen 4-Knoten Elementen. In ABAQUS® gibt es für den ebenen Spannungszustand speziell das Element CPS4 (*Continuum Plane Stress 4 Nodes*). Die Vernetzung des Modells ist in Abbildung 7.21 dargestellt. An der Symmetrie-

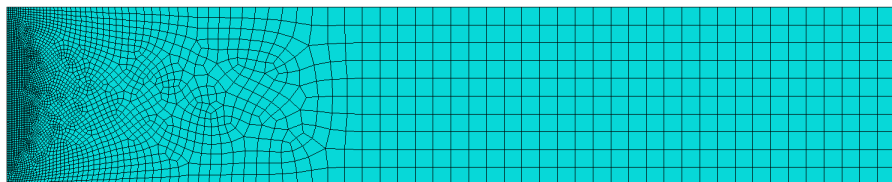


Abbildung 7.21: Vernetzung des 2D-Modells zum Vier-Punkt-Biegeversuch mit 4-Knoten Elementen für den ebenen Spannungszustand (CPS4); Anzahl der Elemente: 3416

kante ist die Vernetzung feiner ausgeführt, da hier die äquivalenten plastischen Dehnungen abgegriffen werden. Über die Balkenhöhe sind in diesem Bereich 100 Elemente angeordnet, sodass eine Elementkantenlänge von $l_e = 0,2 \text{ mm}$ vorliegt. Die äquivalente plastische Dehnung ε_{eq}^p wird an den 100 Elementen abgegriffen, bei denen eine Elementkante mit der zentralen y -Achse bzw. der Symmetriekante zusammenfällt (siehe Abbildung 7.23, Element-Pfad). Dabei wird ε_{eq}^p immer im Schwerpunkt des Elementes ermittelt. In Abbildung 7.22 ist der Ergebnis-Farbplot für ε_{eq}^p (PEEQ) einer konvergierten Berechnung dargestellt. Die Verschiebung beträgt hierbei $\bar{u}_y = -2 \text{ mm}$. Das Ergebnis des 2D-Modells in Abbildung 7.22 stimmt weitestgehend mit dem Ergebnis für das 3D-Modell aus Abbildung 7.9 überein und beträgt in Balkenmitte jeweils $\varepsilon_{eq}^p \approx 7,5 \cdot 10^{-3}$. Im 2D-Modell erfährt der Bereich um den Lasteinleitungspunkt allerdings deutlich mehr plastische Dehnung von $\varepsilon_{eq}^p \approx 1,3 \cdot 10^{-2}$. Die Verschiebung wird in diesem Bereich lediglich auf einen Knoten aufgebracht, wohingegen im 3D-Modell die Verschiebung auf einer Knotenreihe erfolgt. Der Einfluss auf den Bereich von Interesse ist hier allerdings ver-

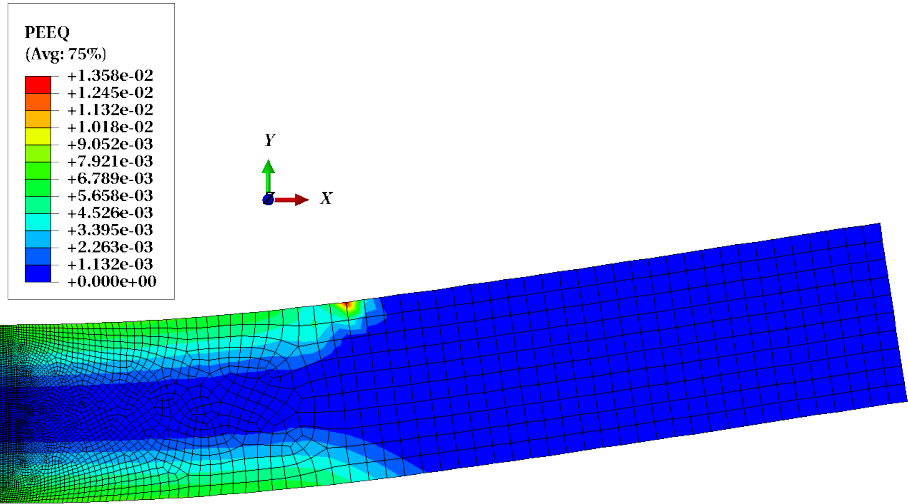


Abbildung 7.22: Farbplot für ε_{eq}^p (PEEQ) nach einer konvergierten FE-Berechnung für die Vernetzung des 2D-Modells mit 100 Elementen über die Balkenhöhe: Aufgebrachte Verschiebung $\bar{u}_y = -2$ mm, Fließgrenze $\sigma^y = 360$ MPa, Elastizitätsmodul $E = 210\,000$ MPa, Querkontraktionszahl $\nu = 0,3$, ideal plastisches Materialverhalten

nachlässigbar (*Prinzip von de Saint Venant*). In Abbildung 7.23 ist die äquivalente plastische Dehnung ε_{eq}^p für den Element-Pfad in der Mitte des Balkens dargestellt. Der Vergleich mit dem Element-Pfad aus dem 3D-Modell mit 40 Elementen über die Balkenhöhe, Abbildung 7.10, verdeutlicht noch mal sehr schön den Einfluss der Vernetzungsdichte auf die Präzision der hoch aufgelösten FE-Berechnung. In beiden Abbildungen sind die Materialwerte und die aufgebrachte Verschiebungsmagnitude gleich. Mit den 100 Elementen über die Balkenhöhe kann im Vergleich zu den 40 bzw. 80 Elementen über die Balkenhöhe des 3D-Modells die Präzision der hoch aufgelösten FE-Berechnung nochmals gesteigert werden.

7.7 Test- und Trainingsdaten für das reduzierte Modell in 2D

Die *Support Vector Regression* wird für das 2D-Modell mit kartesischen Gittern trainiert. Es werden zwei Studien mit verschiedenen, zu approximierenden Testdatenstrukturen durchgeführt.

Die erste Studie entspricht im Hinblick auf die Test- und Trainingsdatenstruktur der Studie, die auch für das 3D-Modell vorgestellt wurde. Hierfür werden in jedem Approximations-Durchlauf äquidistante Gitter aus einem $\mathfrak{X}_{19 \times 19 \times 19}$ Trainingsgitter mit $n = 6859$ Punkten extrahiert. Beginnend mit einem dünnen $\mathfrak{X}_{2 \times 2 \times 2}$ Gitter der Domain-Eckpunkte wird das reduzierte Modell dann jeweils trainiert und im Anschluss für die Approximation der Testpunkte angewendet. Das finale Trainingsgitter ist dann das $\mathfrak{X}_{19 \times 19 \times 19}$ Gitter selbst. Die zu approximierenden $n = 8000$ Testpunkte entstammen dabei wieder aus einem dichten $\mathfrak{X}_{20 \times 20 \times 20}$ Testgitter. Mit dieser Studie ist ein direkter Vergleich mit den Studien zum 3D-Modell möglich, womit der Einfluss der feineren Vernetzung, mit 100 Elementen über die Balkenhöhe, bewertet werden kann.

In einer zweiten Studie wird die Struktur der Testdaten geändert. Die Testdaten sind dabei nicht mehr in einem kartesischen Gitter angeordnet, sondern zufällig verteilt. Die Trainingsphase des

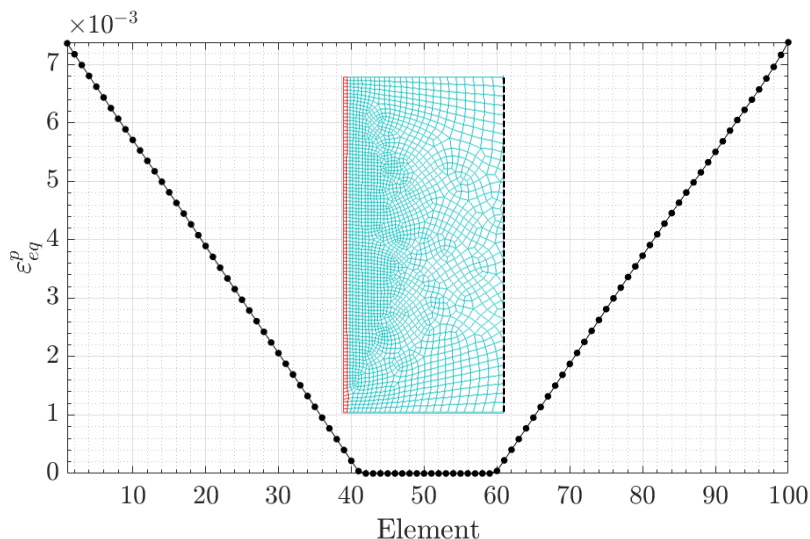


Abbildung 7.23: Äquivalente plastische Dehnung ε_{eq}^p (PEEQ) für den Element-Pfad ■ nach einer konvergierten FE-Berechnung für die Vernetzung des 2D-Modells mit 100 Elementen über die Balkenhöhe: Aufgebrachte Verschiebung $\bar{u}_y = -2$ mm, Fließgrenze $\sigma^y = 360$ MPa, Elastizitätsmodul $E = 210\,000$ MPa, Querkontraktionszahl $\nu = 0,3$, ideal plastisches Materialverhalten

reduzierten Modells erfolgt aber auch hier wieder mit einem kartesischen Gitter. Allerdings werden die Trainingspunkte jetzt in jedem Durchlauf aus dem dichten $\mathfrak{X}_{20 \times 20 \times 20}$ Gitter mit den $n = 8000$ Datenpunkten entnommen. Das $\mathfrak{X}_{20 \times 20 \times 20}$ Gitter ist in der Abbildung 7.24 oben links dargestellt. Die Verschiebung \bar{u} hat aufgrund der Orientierung der y-Achse im 2D-Modell ein negatives Vorzeichen, die Werte entsprechen aber betragsmäßig den Verschiebungen des 3D-Modells, sodass ein direkter Vergleich möglich ist. In der Abbildung 7.24 sind zusätzlich einzelne Antwortflächen für $d^{el}|_{\bar{u}=const.}$ dargestellt. Oben rechts ist die Antwortfläche für eine Verschiebung von $\bar{u} = -0,4$ mm zu sehen. Durch die feine Vernetzung mit 100 Elementen über die Balkenhöhe befinden sich mehr Punkte auf der „Zwischenebene“ bei $d^{el} = 19,6$ mm als im 3D-Modell mit 80 Elementen über die Balkenhöhe (vgl. mit Abbildung 7.13, rechte Seite). In der linken unteren Abbildung 7.24 ist die Antwortfläche für eine Verschiebung von $\bar{u} = -1,5368$ mm dargestellt. Auch hier ist wieder deutlich zu erkennen, dass mit den 100 Elementen über der Balkenhöhe im 2D-Modell eine genauere FE-Lösung für d^{el} erzielt werden kann als mit den 80 Elementen im 3D-Modell, was aus der größeren Anzahl an „Stufen“ hervorgeht (vgl. mit Abbildung 7.14, rechte Seite). Unten rechts in der Abbildung 7.24 ist die Antwortfläche für die maximale Verschiebung von $\bar{u} = -2,8$ mm dargestellt. Auch hier ist im Vergleich zu der Vernetzung mit 80 Elementen über der Balkenhöhe im 3D-Modell die Auflösung für d^{el} feiner (vgl. mit Abbildung 7.15, rechte Seite). Die Genauigkeit der FE-Berechnung für d^{el} mit der Elementkantenlänge $l_e = 0,2$ mm (100 Elemente) im 2D-Modell hat somit leicht im Vergleich zur Elementkantenlänge $l_e = 0,25$ mm (80 Elemente) im 3D-Modell zugenommen.

Wie eingangs schon erwähnt, soll für die Testdatenstruktur in der zweiten Studie kein kartesisches Gitter mehr verwendet werden. Vielmehr soll für die Vergleiche mit den Ergebnissen zu den kartesischen Testdatenstrukturen nun eine zufällige Verteilung der Testdaten verwendet werden. Abweichend von einer „reinen“ zufälligen Verteilung der Testpunkte wird die in

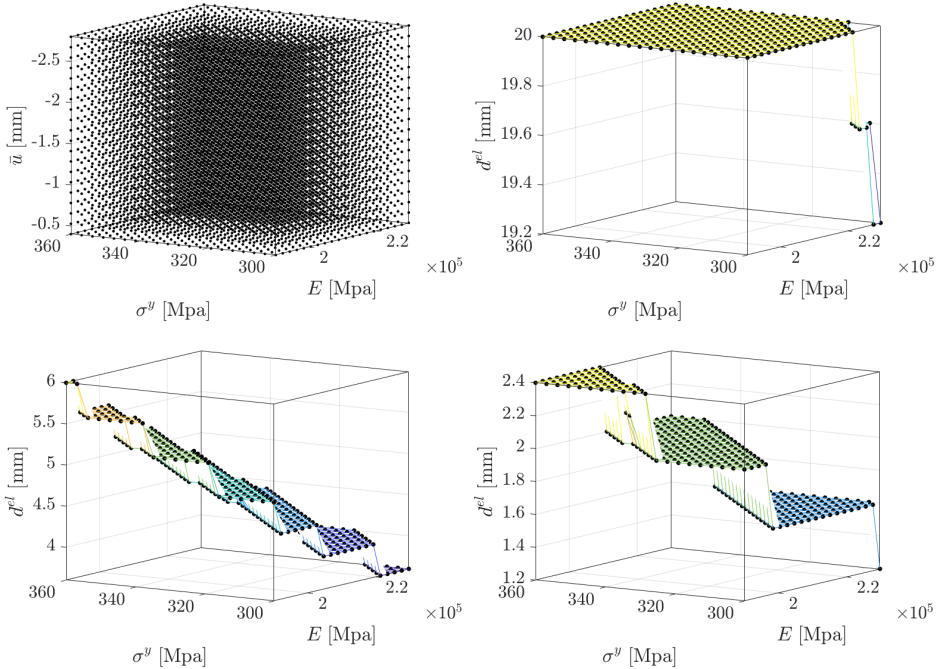


Abbildung 7.24: Trainingsdaten $\mathfrak{X}_{20 \times 20 \times 20}$ mit $n = 8000$ Punkten (oben links); nicht plastifizierter, elastischer Querschnitt $d^{el}|_{\bar{u}=-0,4 \text{ mm}}$ für die minimale Verschiebung (oben rechts); nicht plastifizierter, elastischer Querschnitt $d^{el}|_{\bar{u}=-1,5368 \text{ mm}}$ für die mittlere Verschiebung (unten links); nicht plastifizierter, elastischer Querschnitt $d^{el}|_{\bar{u}=-2,8 \text{ mm}}$ für die maximale Verschiebung (unten rechts); Anzahl der Elemente über die Balkenhöhe: 100

Kapitel 5.4 vorgestellte Lateinische Hyperwürfel² Verteilung (engl. *Latin Hypercube Sampling*, *LHS*) verwendet. Die Anzahl der Testpunkte und die maximale Anzahl Trainingspunkte sind in dieser zweiten Studie gleich ($n = 8000$). In der Abbildung 7.25 ist der nicht plastifizierte, elastische Querschnitt d^{el} als Ergebnis-Farbplot über der *Latin Hypercube* Verteilung aufgetragen. Der raumfüllende Charakter dieses Sampling Verfahrens, ohne „Nesterbildung“, wie es bei „rein“ zufallsbasierten Sampling Verfahren auftreten kann, wird dabei sehr gut deutlich. Gut zu erkennen ist hierbei, dass bei einer geringen Verschiebung von $\bar{u} \approx -0,5 \text{ mm}$ weitestgehend rein elastisches Verhalten vorliegt: $d^{el} \approx d \approx 20 \text{ mm}$ (gelber unterer Bereich). Die kleinsten nicht plastifizierten, elastischen Querschnitte d^{el} liegen wieder bei einer geringen Fließgrenze σ^y , einem hohen Elastizitätsmodul E und einer großen Verschiebung \bar{u} vor (obere rechte Ecke mit dunkelblauer Färbung).

²Der Autor dankt Fynn Benschel für die Bereitstellung des Codes zum Erstellen der Test- und Trainingsdaten dieser zweiten Studie.

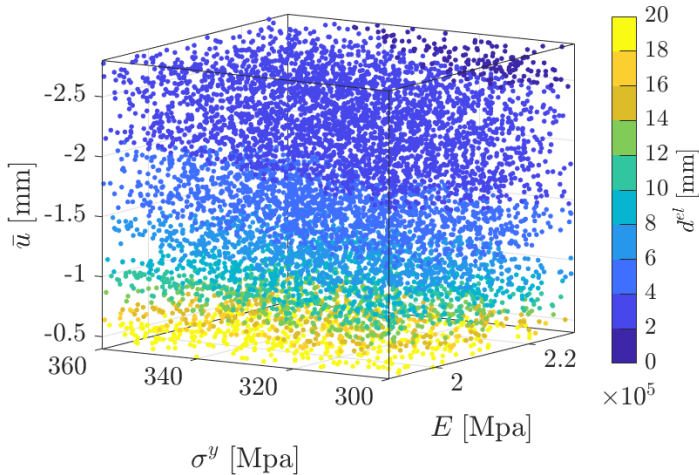


Abbildung 7.25: Testdaten mit $n = 8000$ Punkten in einer Latin Hypercube Verteilung mit Ergebnis-Farbplot für den nicht plastifizierten, elastischen Querschnitt d^{rel}

7.8 Ergebnisse der Support Vector Regression in 2D

Für die Bewertung der Approximationsgüte werden wieder die Fehlermaße ϵ_{max} aus Gleichung (7.8) und ϵ_{rel} aus Gleichung (7.9) herangezogen. Hierdurch ist ein direkter Vergleich mit den Ergebnissen des 3D-Modells aus Kapitel 7.5 möglich. Die Parameter ε (intensive tube width) und C (penalty term) für die Erstellung des reduzierten Modells werden auch im 2D-Fall wieder entsprechend der Gleichung (6.14) bzw. Gleichung (6.15) gewählt.

Der Verlauf der Maximum-Norm ϵ_{max} für die Studie mit dem kartesischen Trainingsgitter $\mathfrak{X}_{19 \times 19 \times 19}$ und dem $\mathfrak{X}_{20 \times 20 \times 20}$ Testgitter ist in Abbildung 7.26 dargestellt. Zusätzlich sind die drei Wertepaare $(n, \epsilon_{\text{max}})$ des Kurvenverlaufes hervorgehoben, die auch schon für den Verlauf des 3D-Modells mit 80 Elementen über die Balkenhöhe hervorgehoben sind (vgl. Abbildung 7.18, Kapitel 7.5). Wie zu erwarten war, stellt sich auch bei der Anzahl von 100 Elementen über der Balkenhöhe im 2D-Modell ein ähnliches Ergebnis wie für das 3D-Modell mit 80 Elementen über die Balkenhöhe ein. Im hinteren Bereich ist wieder eine Konvergenz zu beobachten, wobei das finale Ergebnis von $\epsilon_{\text{max}} = 0,4396$ mm bei $n = 6859$ Trainingspunkten noch mal eine Steigerung zu dem 3D-Modell mit 80 Elementen über die Balkenhöhe darstellt. In der Abbildung 7.27 ist der zugehörige relative Fehler für die Studie mit den kartesischen Test- und Trainingsgittern dargestellt. Als Vergleich dient auch hier wieder das Ergebnis des 3D-Modells mit den 80 Elementen über die Balkenhöhe (vgl. Abbildung 7.19, Kapitel 7.5). Der finale Fehler bei $n = 6859$ Trainingspunkten von $\epsilon_{\text{rel}} = 0,01703 \approx 1,7\%$ stellt auch hier noch mal eine Steigerung im Vergleich zu dem Ergebnis für das 3D-Modell ($\approx 2\%$) dar.

In der zweiten Studie wird die Approximation für die Testpunkte in der Latin Hypercube Verteilung durchgeführt. Der Verlauf der Maximum-Norm ϵ_{max} für die Latin Hypercube Approximation ist in Abbildung 7.28 dargestellt. Zusätzlich sind drei Wertepaare $(n, \epsilon_{\text{max}})$ des Kurvenverlaufes hervorgehoben. Trotz der feinen Vernetzung mit 100 Elementen über die Balkenhöhe geht der Wert für ϵ_{max} nicht unter 1 mm. Ab ca. $n = 5000$ Trainingspunkten läuft die Fehlerkurve in eine Sättigung ein und bleibt bei Werten von ca. $\epsilon_{\text{max}} \approx 1,0$ mm stehen. In der Abbildung 7.29 ist der Verlauf des zugehörigen relativen Fehlers ϵ_{rel} für die Latin Hypercube Approximation

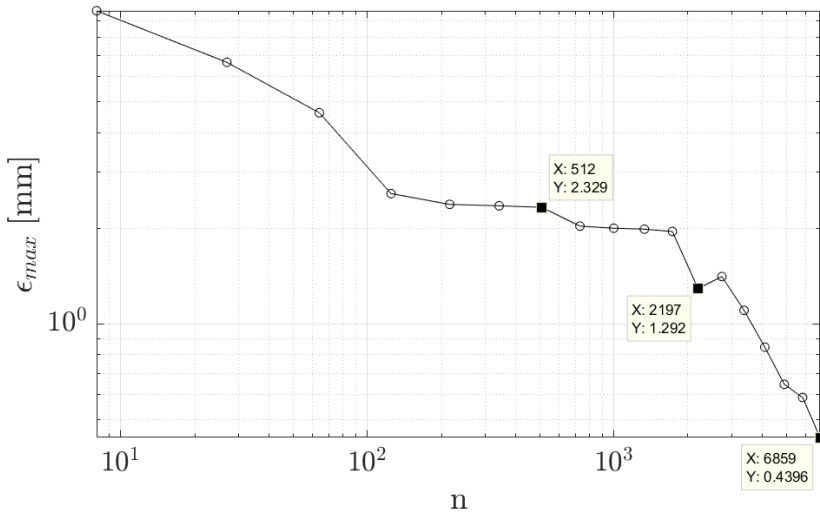


Abbildung 7.26: Maximum Norm ϵ_{max} für die Vernetzung mit 100 Elementen über der Balkenhöhe im 2D-Modell und das kartesische Testgitter $\mathfrak{X}_{20 \times 20 \times 20}$: Matèrn 5/2 Kernel Funktion; Parameter $C_{opt} = iqr(\mathbf{Y})/0,001$ und $\varepsilon = iqr(\mathbf{Y})/200$; n : Anzahl der Trainingspunkte im Gitter; Zweifach logarithmische Darstellung

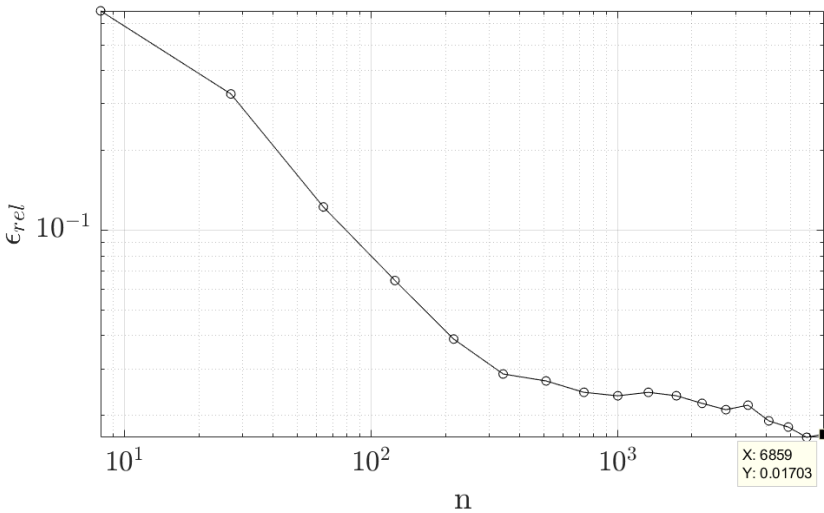


Abbildung 7.27: Relativer Fehler ϵ_{rel} für die Vernetzung mit 100 Elementen über der Balkenhöhe im 2D-Modell und das kartesische Testgitter $\mathfrak{X}_{20 \times 20 \times 20}$: Matèrn 5/2 Kernel Funktion; Parameter $C_{opt} = iqr(\mathbf{Y})/0,001$ und $\varepsilon = iqr(\mathbf{Y})/200$; n : Anzahl der Trainingspunkte im Gitter; zweifach logarithmische Darstellung

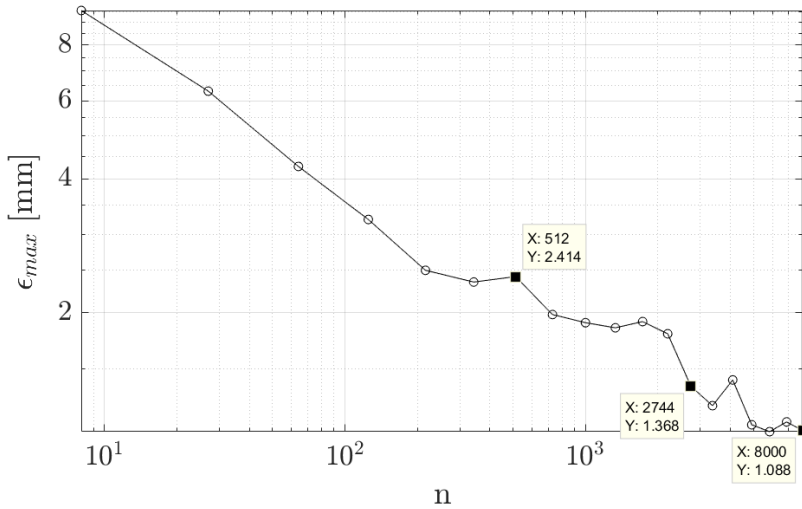


Abbildung 7.28: Maximum Norm ϵ_{max} für die Vernetzung mit 100 Elementen über der Balkenhöhe im 2D-Modell und die Latin Hypercube Verteilung der Testpunkte : Matérn 5/2 Kernel Funktion; Parameter $C_{opt} = \text{igr}(\mathbf{Y})/0,001$ und $\varepsilon = \text{igr}(\mathbf{Y})/200$; n : Anzahl der Trainingspunkte im Gitter; zweifach logarithmische Darstellung

dargestellt. Der minimale Wert von $\epsilon_{rel} \approx 2\%$ ist am Ende der Kurve bei 8000 Trainingspunkten hervorgehoben. Der Verlauf des relativen Fehlers entspricht weitestgehend dem Verlauf für das kartesische Testgitter aus Abbildung 7.27. Qualitativ hat sich die Approximationsgüte ϵ_{rel} für die Testdaten in der *Latin Hypercube* Verteilung im Gegensatz zur Approximation eines kartesischen Testgitters also nicht wesentlich verschlechtert.

Wesentlich für die praktische Anwendung ist aber die Sicherstellung von $\epsilon_{max} \leq \epsilon_{max}^{Tol}$ für beliebig gewählte Testpunkte, wobei ϵ_{max}^{Tol} für einen Toleranzwert steht. Aus dem Verlauf des absoluten Fehlers ϵ_{max} in Abbildung 7.28 kann auf einen Einfluss der Testdatenstruktur auf die Approximationsgüte geschlossen werden. Durch die Erhöhung der Elementanzahl von 80 auf 100 und somit eine feinere Auflösung der *Quantity of Interest* d^{el} in den Trainingsdaten wäre auch hier eine Verbesserung im Vergleich zum Verlauf für das 3D-Modell aus Abbildung 7.18 zu erwarten gewesen. Die Kurve stagniert allerdings bei $\epsilon_{max} \approx 1$ mm. Aus der Abbildung 7.28 wird allerdings auch deutlich, dass trotz der zufälligen Verteilung der $n = 8000$ Testpunkte mit kartesischen Trainingsgittern von $n \geq 700$ Punkten schon Fehler von $\epsilon_{max} < 2$ mm möglich sind. Für die praktische Anwendung ist die Vorhersage von d^{el} mit einer Abweichung von $\frac{1}{10}$ bei einer Gesamtbalkenhöhe von $d = 20$ mm vielfach ausreichend. Für beliebige Testpunkten ist somit in jedem Fall $\epsilon_{max} \leq \epsilon_{max}^{Tol} = 2$ mm bei $n \geq 700$ möglich.

7.9 Sensitivität in den Parameterrichtungen

Die kartesischen Trainings-Gitter der vorgestellten Studien haben in den Parameter-Richtungen σ^y , E und \bar{u} äquidistante Abstände. Solche Gitterstrukturen werden als *isotrop* bezeichnet. Aus den Ergebnis-Farbplots sowie den einzelnen Ergebnisflächen für d^{el} , die in den Kapiteln 7.4 und 7.7 abgebildet sind, lässt sich jedoch ein unterschiedliches Antwortverhalten der hoch aufgelösten Berechnung in den drei Parameter-Richtungen erkennen. Dies legt die Vermutung

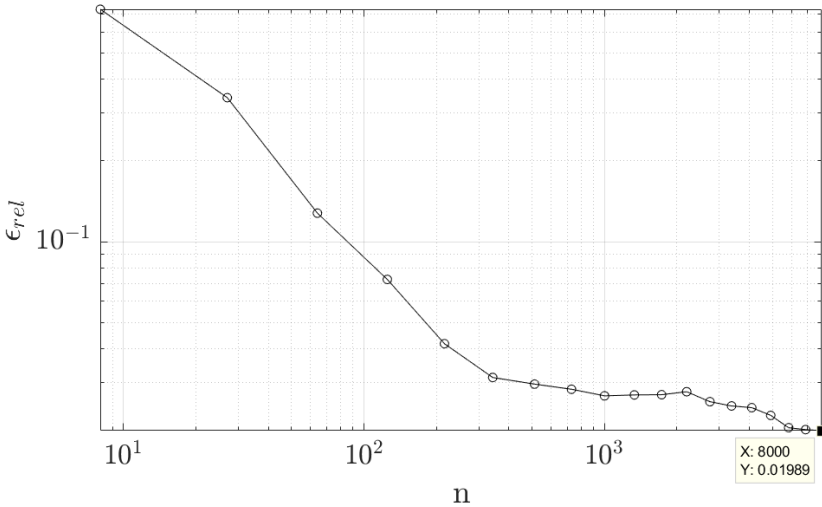


Abbildung 7.29: Relativer Fehler ϵ_{rel} für die Vernetzung mit 100 Elementen über der Balkenhöhe im 2D-Modell und die Latin Hypercube Verteilung der Testpunkte : Matèrn 5/2 Kernel Funktion; Parameter $C_{opt} = iqr(\mathbf{Y}) / 0,001$ und $\varepsilon = iqr(\mathbf{Y}) / 200$; n : Anzahl der Trainingspunkte im Gitter; zweifach logarithmische Darstellung

nahe, dass auch das reduzierte Modell unterschiedlich *sensibel* in den Parametern σ^y , E und \bar{u} reagiert. Im Folgenden soll daher eine Studie mit *anisotropen* Trainingsgittern durchgeführt werden. Hierunter sind kartesische Gitter zu verstehen, bei denen die Parameter-Richtungen σ^y , E und \bar{u} mit einer unterschiedlichen Anzahl an Trainingspunkten versehen sind.

Bei Betrachtung der Antwortflächen im $\mathfrak{X}_{20 \times 20 \times 20}$ Gitter in Abbildung 7.24 ist ein annähernd linearer, wenn auch stufenartiger Verlauf in den Parametern σ^y und E zu erkennen. Für $d^{el}(\bar{u})$ kann ein nichtlineares Verhalten entnommen werden, da die „Stufenanzahl“ bei minimaler und maximaler Verschiebung \bar{u} geringer ist als bei der mittleren Verschiebung. Diese beiden Unterscheidungen in den Parameter-Richtungen können wie folgt festgehalten werden,

$$\begin{aligned} d^{el}(\bar{u} = \text{const.}, \sigma^y, E) &\sim \text{linear}, \\ d^{el}(\bar{u}) &\sim \text{nichtlinear}. \end{aligned} \quad (7.10)$$

Für die durchgeführte *Sensitivitätsanalyse* werden nacheinander immer dichtere Trainingsgitter aus dem $\mathfrak{X}_{19 \times 19 \times 19}$ extrahiert. Der Unterschied zu den bisher vorgestellten Studien ist der, dass jetzt in jedem Trainingsdurchlauf ein *anisotropes* Gitter aus dem vollen Gitter extrahiert wird. Die Approximation wird dann in jedem Durchlauf für das $\mathfrak{X}_{20 \times 20 \times 20}$ Testgitter durchgeführt, sodass ein direkter Vergleich mit den Ergebnissen der Studie aus Kapitel 7.8, speziell den Kurven aus den Abbildungen 7.26 und 7.27, gegeben ist, bei der in jedem Schleifendurchlauf ein *isotropes* Gitter verwendet wird. Entsprechend der Zusammenstellung aus Gleichung (7.10) wird der Faktor 2 für das Anzahl-Verhältnis der Samplingpunkte in der Parameter-Richtung \bar{u} zu den Parameter-Richtungen σ^y , E getestet. Ein *anisotropes* Trainingsgitter kann somit in dieser Studie wie folgt dargestellt werden

$$\begin{aligned} \mathfrak{X}_{2n \times n \times n} &= \{\bar{u}\}_{i=1 \dots 2n} \times \{\sigma_i^y\}_{i=1 \dots n} \times \{E_i\}_{i=1 \dots n} \\ &\times : \text{kartesisches Produkt}. \end{aligned} \quad (7.11)$$

Durch den Faktor 2 wird die Anzahl der Trainingspunkte in der \bar{u} -Richtung somit „schneller“ erhöht, da hier die *Sensitivität* des reduzierten Modells vermutet wird. Der Vergleich der Ergebnisse findet mit den Ergebnissen aus Kapitel 7.8 statt. In Abbildung 7.30 ist der absolute Fehler ϵ_{max} für die *isotropen* und *anisotropen* Trainingsgitter dargestellt. Der Verlauf der zugehörigen relativen Fehler ϵ_{rel} für beide Gittertypen ist in Abbildung 7.31 abgebildet. Zusätzlich sind in

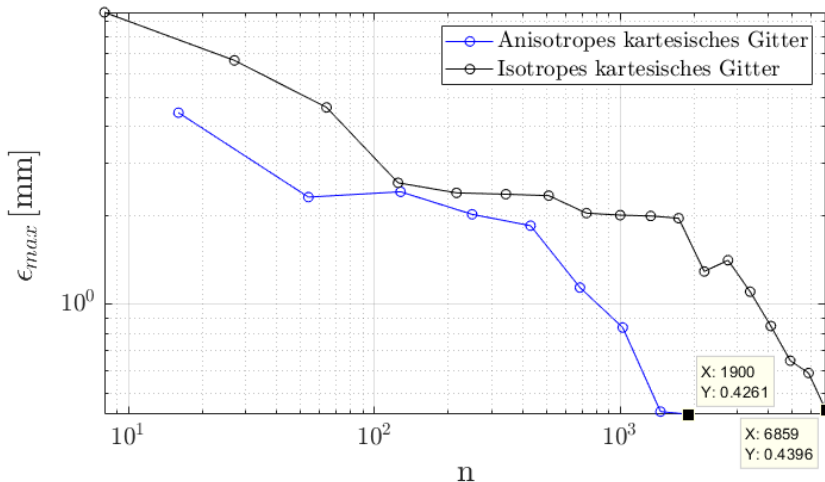


Abbildung 7.30: Maximum Norm ϵ_{max} für die Vernetzung mit 100 Elementen über der Balkenhöhe im 2D-Modell und das kartesische Testgitter $\mathfrak{X}_{20 \times 20 \times 20}$; Vergleich der *isotropen* und *anisotropen* kartesischen Trainingsgitter : Matèrn 5/2 Kernel Funktion; Parameter $C_{opt} = iqr(\mathbf{Y})/0,001$ und $\varepsilon = iqr(\mathbf{Y})/200$; n : Anzahl der Trainingspunkte im Gitter; zweifach logarithmische Darstellung

den beiden Abbildungen die Fehlerwerte für die finale Anzahl an Trainingspunkten des *isotropen* und des *anisotropen* Trainingsgitters hervorgehoben. Für das finale *isotrope* Gitter mit $n = 6859$ Trainingspunkten und das finale *anisotrope* Gitter mit $n = 1900$ Trainingspunkten stimmen die Trainingspunkte für die Verschiebungsmagnitude \bar{u}_i mit $i = 1 \dots 19$ überein. Die Gitter unterscheiden sich jedoch entsprechend der Gleichung (7.11) in den Parameter-Richtungen σ^y und E . Die Draufsichten der $\sigma^y - E$ -Ebenen für das *anisotrope* und *isotrope* Gitter der finalen Trainingspunkteverteilungen sind in Abbildung 7.32 dargestellt. Aus der Abbildung 7.30 wird deutlich, dass in den Parameter-Richtungen σ^y und E deutlich weniger Punkte notwendig sind, um gute Ergebnisse in der Approximation zu erzielen. Das beste Ergebnis $\epsilon_{max} \approx 0,43$ mm für das *isotrope* Gitter wird mit dem *anisotropen* Gitter schon bei $\frac{1900}{6859} \approx \frac{1}{4}$ der Anzahl an Trainingspunkten des *isotropen* Gitters erreicht. Das Gleiche gilt für den „besten“ relativen Fehler $\epsilon_{rel} \approx 1,7\%$ in Abbildung 7.31, der auch hier mit ungefähr $\frac{1}{4}$ der Trainingspunkte des *isotropen* Gitters erreicht wird. Die Kurven für ϵ_{max} in Abbildung 7.30 verlaufen im hinteren, konvergenten Bereich parallel, wenn auch mit großem Abstand. Das zeigt noch einmal deutlich, dass die Auflösung in der Parameter-Richtung \bar{u} maßgeblich für eine gute Approximation ist.

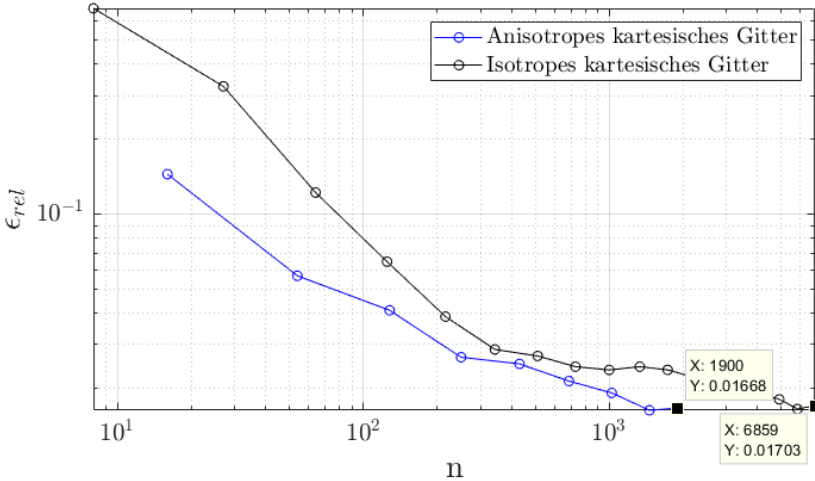


Abbildung 7.31: Relativer Fehler ϵ_{rel} für die Vernetzung mit 100 Elementen über der Balkenhöhe im 2D-Modell und das kartesische Testgitter $\mathfrak{X}_{20 \times 20 \times 20}$; Vergleich der *isotropen* und *anisotropen* kartesischen Trainingsgitter: Matèrn 5/2 Kernel Funktion; Parameter $C_{opt} = iqr(\mathbf{Y})/0,001$ und $\epsilon = iqr(\mathbf{Y})/200$; n : Anzahl der Trainingspunkte im Gitter; zweifach logarithmische Darstellung

7.10 Fazit aus den Untersuchungen und Rechenzeitersparnis

Das vorgestellte Kapitel hat die Leistungsfähigkeit der *Support Vector Regression* für elasto-plastische 3D- und 2D-Modelle aufgezeigt, die mittels der Finiten Elemente Methode berechnet werden. Für die Erstellung eines Ersatzmodells mittels SVR ist zuerst die Definition der *Quantity of Interest*, also der „Ausgabegröße von Interesse“ nötig. Im Rahmen dieser Arbeit wird der elastische, nicht plastifizierte Querschnitt d^{el} eines Vier-Punkt-Biegebalkens mit der Balkenhöhe d als relevante Größe gewählt. Die Eingabegrößen sind die Fließgrenze σ^y , der Elastizitätsmodul E und die Verschiebungsmagnitude \bar{u} der Balkenbelastung. Eine wesentliche Erweiterung im Hinblick auf das 1D-Modell Kapitel 6 ist die Datenstruktur. Im 1D-Fall liegt eine glattes Antwortverhalten vor, dass C^0 -Stetig in einen konstanten „0“-Bereich übergeht. In dem vorgestellten 3D-Modell ist das Antwortverhalten der Größe d^{el} zudem „Stufenhaft“, was der gewählten FE-Auflösung bzw. Elementgröße geschuldet ist. Ein solches Verhalten ist im Rahmen der Modellreduktion schwer zu approximieren. Aber die Support Vector Regression hat hier ihre Stärken gezeigt und konnte ausreichend genaue Approximation liefern.

Bei dem in dieser Arbeit gewählten Beispiel kommt die Netzabhängigkeit deutlich zum tragen. Eine feine Vernetzung bei der hier gewählten *Quantity of Interest* d^{el} trägt maßgeblich zur Güte der Approximation bei. Die Elementgröße muss vorab sinnvoll gewählt werden, was in Kapitel 7.5 anhand der unterschiedlichen Approximationsergebnisse für Elementkantenlängen von $l_e = 0,5$ mm und $l_e = 0,25$ mm gezeigt wurde. Der Anwender hat somit vorab zu prüfen, wie sensibel seine *Quantity of Interest* auf die Netzfeinheit bei der Modellreduktion reagiert. Die Struktur der Testdaten hat einen weiteren wesentlichen Einfluss auf das Ergebnis der Approximation, was im Rahmen der Untersuchungen zum 2D-Modell deutlich wird. Aber auch hier konnte gezeigt werden, dass die Support Vector Regression, trainiert mit „relativ“ wenig Punkten in kartesischen Gittern, in der Lage ist, beliebig verteilte Testpunkte mit einer Genauigkeit von mindestens $\frac{1}{10}$ Abweichung in Bezug auf die Referenzgröße (hier die Balkenhöhe

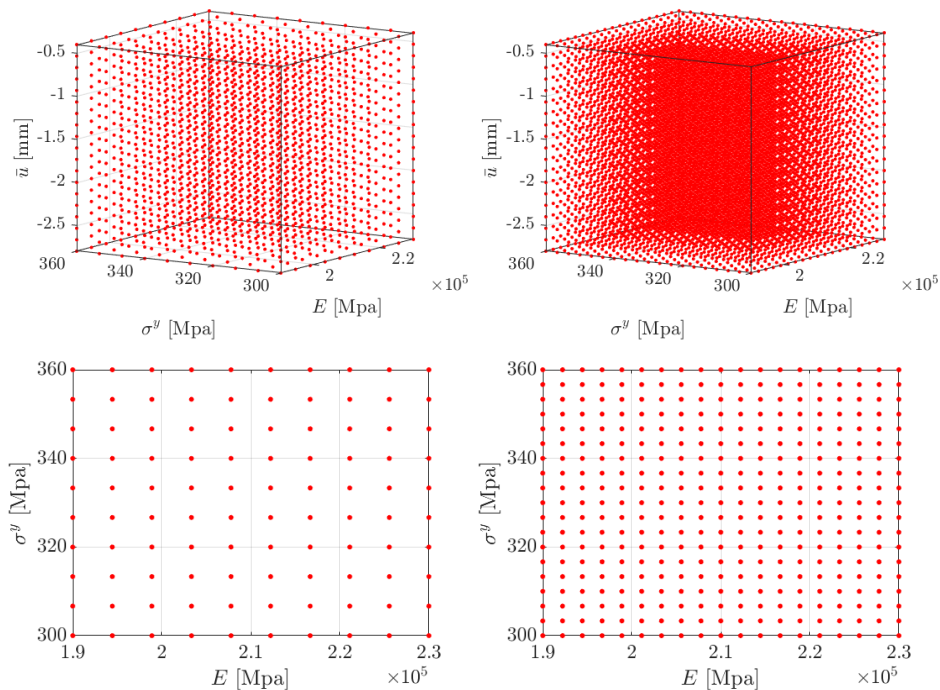


Abbildung 7.32: Finales, *anisotropes* $\mathfrak{X}_{19 \times 10 \times 10}$ Trainingsgitter mit $n = 1900$ Trainingspunkten (links oben); Draufsicht des finalen, *anisotropen* Trainingsgitters in der $\sigma^y - E$ -Ebene (links unten); Finales, *isotropes* $\mathfrak{X}_{19 \times 19 \times 19}$ Trainingsgitter mit $n = 6859$ Trainingspunkten (rechts oben); Draufsicht des finalen, *isotropen* Trainingsgitters in der $\sigma^y - E$ -Ebene (rechts unten)

d) vorherzusagen.

Die Untersuchung der Sensitivität in den drei Parameterrichtungen σ^y , E and \bar{u} hat ergeben, dass eine unterschiedliche Anzahl der Trainingspunkte in den drei Richtungen einen erheblichen Einfluss auf das Approximationsergebnis des reduzierten Modells hat. Besonders *sensibel* ist in dieser Anwendung die Richtung der Verschiebungsmagnitude \bar{u} , da hier der nichtlineare Teil des hoch aufgelösten Modells liegt. Mehr Trainingspunkte in dieser Richtung, bei gleichzeitig „dünnere“ Belegung der Richtungen σ^y und E haben zu einer schnelleren Konvergenz im Gegensatz zu *isotropen* kartesischen Gittern geführt.

Das wesentliche Entscheidungskriterium „für“ oder „gegen“ ein *Modellreduktionsverfahren* ist die effektive (Rechen-) Zeitersparnis. Der größte Anteil des Berechnungsaufwands für eine Modellreduktion liegt in der *Trainingsphase*. Als Beispiel sind in Tabelle 7.1 die Berechnungszeiten für das vorgestellte 2D-Modell und das *anisotrope* Trainingsgitter mit $n = 1900$ Trainingspunkten aufgeführt. Mithilfe der Modellreduktion können in der *Online Phase* die Berechnungen in einem Bruchteil der Zeit im Vergleich zur Auswertung des hoch aufgelösten Modells ausgewertet werden. Für die Werte aus Tabelle (7.1) kann eine $\frac{2\text{s}}{0,18\text{s}} \approx 11$ mal schnellere Berechnung in der *Online Phase* erzielt werden. Für „fein“ aufgelösten *Parameterstudien* oder bei *Optimierungsaufgaben* für Strukturen im Ingenieurwesen ist somit eine Modellreduktion sinnvoll. Hinzu kommt auch noch die Frage, wie genau die Approximation sein muss. Bei einer geringeren Ap-

Berechnungsschritte	Berechnungszeit
Eine hoch aufgelösten Berechnung	2 s
$n = 1900$ hoch aufgelöste Berechnungen	3800 s \approx 1 h
SVR Modellerstellung mit $n = 1900$ Trainingspunkten	60,7 s
Auswertung des reduzierten Modells für einen Testpunkt	0,18 s

Tabelle 7.1: Rechenzeiten der einzelnen Abschnitte einer Modellreduktion für das 2D-Modell und ein anisotropes Trainingsgitter mit $n = 1900$ Punkten und einem absoluten Fehler $\epsilon_{max} \approx 0,43$ mm.

proximationsgenauigkeit genügt in den meisten Fällen schon eine Trainingsphase mit deutlich weniger Trainingspunkten, sodass hier weitere Einsparungen möglich sind. Ein weiterer Faktor ist die Komplexität des hoch aufgelösten Modells. Für das herangezogen Beispiel aus Tabelle (7.1) ist die Berechnungszeit mit 2 s für die hoch aufgelöste Berechnung selbst sehr gering. Der Faktor für die Rechenzeiterparnis wird deutlich größer wenn aufwändigere Strukturen oder Materialmodelle herangezogen werden.

8 Zusammenfassung und Ausblick

In der vorliegenden Dissertation wird die *Support Vector Regression* auf den Bereich der Elasto-Plastizität angewendet. Ziel ist es, ein global reduziertes Modell für eine Größe (*Quantity of Interest*) zu erstellen, die aus einer nichtlinearen Finite Elemente Berechnung hervorgeht.

Nach einer Einleitung in Kapitel 1, welche die Notwendigkeit der Modellreduktion aufzeigt, wird in Kapitel 2 ein Überblick über gängige Modellreduktionsverfahren gegeben. Hierbei stehen sowohl die Vielfalt der mechanischen Anwendungen als auch unterschiedliche Techniken der Modellreduktion im Vordergrund.

Der für die Anwendung im Bereich der Elasto-Plastizität notwendige kontinuumsmechanische Rahmen wird in Kapitel 3 bereitgestellt. Nach der Herleitung der kinematischen Beziehungen und einem hieraus abgeleiteten Verzerrungsmaß für kleine Deformationen wird das Konzept der mechanischen Spannungen vorgestellt. Daran anschließend werden die notwendigen Bilanzgleichungen für die Lösung des mechanischen Feldproblems hergeleitet. Das Kapitel schließt mit den Materialgleichungen ab, die eine Beziehung zwischen den Spannungen und Dehnungen herstellen. Hierbei liegt der Fokus auf dem elasto-plastischen Materialverhalten, insbesondere der *von Mises* Plastizität.

In Kapitel 4 wird die Finite Elemente Methode vorgestellt, die für die numerische Lösung des Feldproblems der Mechanik herangezogen wird. Ausgehend von der schwachen Formulierung der zugrunde liegenden Gleichungen wird der Finite Elemente Ansatz eingeführt. Daran anschließend wird das isoparametrische Konzept und die numerische Integration vorgestellt, welche wesentlich zur Effizienz der FEM beitragen. Mit den Lösungsverfahren der Finiten Elemente Gleichungssysteme schließt das Kapitel ab. Der Fokus liegt hierbei auf den iterativen Lösungsverfahren für die sowohl die Last- als auch die Verschiebungssteuerung vorgestellt wird. Der mathematische Hintergrund für die Modellreduktion mittels der *Support Vector Regression* wird in Kapitel 5 bereitgestellt. Ausgehend von der klassischen, linearen Regression werden stückweise die erweiterten Konzepte eingeführt. Die erste wesentliche Erweiterung besteht in der Einführung von *Kernel* Funktionen, die aus einer Transformation der Trainingsdaten in einen Merkmal-Raum hervorgehen. Die inneren Produkte im Merkmal-Raum werden durch passende *Kernel* Funktionen ersetzt, wodurch auch nichtlinear zusammenhängende Datenstrukturen mit den Techniken der linearen Lösungsverfahren einer Regression unterzogen werden können. Für die *Support Vector Regression* wird eine ε -*intensive* Verlustfunktion eingeführt, für dessen Lösung ein entsprechendes Minimierungsproblem formuliert werden kann. Mit der weiteren Einführung der sogenannten *Slack* Variablen und dem Begrenzungsparameter C kann die Flexibilität der Regression deutlich erhöht werden. Die *Slack* Variablen und die Parameter ε und C werden in den Nebenbedingungen und den sogenannten *Karush-Kuhn-Tucker* Komplementaritätsbedingungen für das Minimierungsproblem berücksichtigt. Im Weiteren wird die Standardisierung der Eingabedaten vorgestellt, die auch in den späteren Anwendung eingesetzt wird. Da im Rahmen dieser Arbeit nicht glatte Antwortspektren auftreten, widmet sich ein Unterkapitel diesem Phänomen. Das Kapitel schließt mit den Sampling Methoden für die Modellreduktion ab. In erster Linie werden die beiden *One-Shot* Sampling Methoden, *kartesische* Gitter und *Latin Hypercube* Sampling, vorgestellt, die auch im Rahmen dieser Arbeit verwendet werden.

In Kapitel 6 werden die Untersuchungen zur rein phänomenologischen 1D-Elasto-Plastizität vorgestellt. Zuerst wird das Modell und dessen Antwortverhalten vorgestellt, bei dem eine Unstetigkeit von einem nichtlinearen in einen konstant verlaufenden Bereich vorliegt. Hier wird eine Schwierigkeit deutlich, da solche Verläufe global nicht hinreichend mit einfachen Polynomansätzen abgebildet werden können. In den vorgestellten Studien wird zuerst nach einer geeigneten *Kernel* Funktion gesucht. Dabei zeigt die *Gauß'sche Kernel* Funktion ein gutes Ap-

proximationsverhalten im Vergleich mit der *linearen Kernel* Funktion und der *Polynom Kernel* Funktion auf. Für die *Support Vector Regression* sind der Parameter ε für die ε -*intensive* Verlustfunktion und der Parameter C für die „Boxberandung“ (*engl. Box Constraint*) von entscheidender Bedeutung. Diese müssen vorab problemspezifisch ausgewählt werden. In den vorgestellten Studien werden verschiedene Werte getestet und schließlich ein optimaler Parametersatz ($\varepsilon_{opt}, C_{opt}$) ermittelt. In einer weiteren Konvergenzstudie wird deutlich, dass die *Gauß'sche Kernel* Funktion ab einer größeren Anzahl an Trainingspunkten einen divergierenden und oszillierenden Verlauf in den Fehlerkurven aufweist. Eine deutliche Verbesserung kann hier mit der *Matèrn 5/2 Kernel* Funktion erzielt werden. Dort, wo die *Gauß'sche Kernel* Funktion einen divergierenden und oszillierenden Verlauf aufweist, verläuft die Fehlerkurve für die *Matèrn 5/2 Kernel* Funktion weiter konvergent. In einer weiteren Studie wird das Phänomen der Überanpassung (*engl. Overfitting*) untersucht, da auch die *Matèrn 5/2 Kernel* Funktion ab einer sehr hohen Anzahl an Trainingspunkten im Verlauf der Fehlerkurven anfängt zu oszillieren und zu divergieren. In dieser Studie werden die optimalen Parameter ($\varepsilon_{opt}, C_{opt}$) nochmal bestätigt, da weder eine weitere manuelle Anpassung, als auch eine automatische Anpassung des Parameters C das Oszillieren und Divergieren unterbinden können. Abschließend werden im Rahmen der 1D Elasto-Plastizität verschiedene adaptive Samplingverfahren untersucht. Mit den beiden vorgestellten Techniken können im Bereich „weniger“ Trainingspunkte bessere Ergebnisse im Vergleich zu einer Verfeinerung mit den homogenen, kartesischen Gittern erzielt werden.

Aufbauend auf den Erkenntnissen der 1D Elasto-Plastizität wird die *Support Vector Regression* in Kapitel 7 auf den drei- und zweidimensionalen elasto-plastischen Fall angewendet. Die „Größe von Interesse“ (*engl. Quantity of Interest*) ist der nicht plastifizierte elastische Querschnitt eines Vier-Punkt-Biegebalkens. Diese Größe wird im Rahmen des Postprocessing aus einer Finite Elemente Berechnung ermittelt. Nach der Vorstellung des mechanischen Modells werden der Parameterraum und die Ausgabegröße (*engl. Quantity of Interest*) definiert. Im Anschluss erfolgt die Überführung des mechanischen Modells in ein Finite Elemente Modell, hierbei wird zuerst der dreidimensionale Fall betrachtet. Auf der Grundlage der vorgestellten Test- und Trainingsdatenverteilung kann die *Support Vector Regression* erfolgreich auf den elasto-plastischen dreidimensionalen Fall angewendet werden. Die Fehlerkurven weisen einen konvergenten Verlauf auf, die Ergebnisse sind allerdings in diesem Fall stark von der Vernetzungsdichte im interessierenden Bereich abhängig. Darüber hinaus kann in den Untersuchungen gezeigt werden, dass die *Support Vector Regression* in der Lage ist, für praktische Ingenieur Anwendungen eine ausreichende Präzision in der Approximation zu erzielen. In einer weiteren Studie wird der zweidimensionale elasto-plastische Fall im Rahmen eines ebenen Spannungszustandes untersucht. Dabei wird die Vernetzungsdichte im interessierenden Bereich noch einmal im Vergleich zum dreidimensionalen Modell erhöht. Zudem wird neben den kartesischen Testgittern auch eine Approximation für eine *Latin Hypercube* Verteilung der Testpunkte durchgeführt. In den Untersuchungen kann die Abhängigkeit der Vernetzungsdichte nochmals bestätigt werden. Die Studien zeigen, dass auch für den zweidimensionalen elasto-plastischen Fall eine hinreichend genaue Approximation mittels der *Support Vector Regression* erzielt werden kann, sodass dieses Verfahren ein geeignetes Modellreduktionsverfahren für elasto-plastische mechanische Anwendungen in der Ingenieurpraxis darstellt. Abschließend wird eine Untersuchung zur Sensitivität in den Parameterrichtungen durchgeführt. Dabei kann gezeigt werden, dass die Anzahl der Trainingspunkte in Form von *anisotropen* Gittern reduziert werden kann, ohne die Approximationsgüte zu beeinträchtigen. Dies zeigt nochmal die Effizienz der *Support Vector Regression* für diesen Anwendungsfall auf, da auch mit drastischer Reduktion der Trainingspunkte immer noch hinreichend genaue Ergebnisse erzielt werden.

Mit der vorliegenden Dissertation kann eine Erweiterung des Anwendungsbereichs von SVR für global reduzierte Modelle der Elasto-Plastizität aufgezeigt werden. Die Methode lässt sich im Rahmen von Parameterstudien, bei denen große Mengen an Daten benötigt werden, in der Praxis einsetzen. Das Verhältnis von erforderlichen Berechnungen und den Berechnungen, die

für die Trainingsphase zur Modellerstellung nötig sind, muss dabei in Betracht gezogen werden. Dies hängt stark von der erforderlichen Präzision des reduzierten Modells ab. Hier muss die Ingenieurin bzw. der Ingenieur entscheiden, was für den vorliegenden Anwendungsfall ausreichend ist. Abschließend kann in der vorliegenden Arbeit festgehalten werden, dass die *Support Vector Regression* ein geeignetes Modellreduktionsverfahren für Anwendungen in der nichtlinearen FEM, speziell im Bereich der Elasto-Plastizität, bereitstellt.

In weiteren Studien können unterschiedliche mechanische Modelle herangezogen werden. Dabei können zum einen die Eingabeparameter verändert werden. Zum anderen können verschiedene „Größen von Interesse“ (*engl. Quantity of Interest*) in Betracht gezogen werden. Als mechanisches Modell wäre ein Druckgussbauteil unter mehrachsiger Belastung denkbar. Hier könnte beispielsweise die Verschiebung an einem kritischen Punkt bei elasto-plastischen Materialverhalten untersucht werden.

Auch die *adaptiven* Sampling Methoden können weiter verbessert werden. Hier können verschieden erprobte Verfahren ausprobiert und neue Techniken implementiert werden. Im Rahmen der *One-Shot* Sampling Methoden können anisotrope Gitter für die Sensitivitätsanalyse weiter untersucht werden. Hier können gegebenenfalls weitere Reduktionen bei der Anzahl an Trainingspunkten vorgenommen werden, ohne die Approximationsgüte zu beeinträchtigen.

Eine weitere Möglichkeit ist das Einbeziehen *stochastischer* Aspekte in die Modellreduktion mittels SVR. Stochastische Methoden im Rahmen der FEM sind intensiver Gegenstand der Forschung am Institut für Baumechanik und Numerische Mechanik (IBNM), sodass beispielsweise die Möglichkeit der Kombination verschiedener stochastischer Verfahren mit der *Support Vector Regression* besteht.

A Anhang

A.1 SVR Programmcode in MATLAB[®]

```
% Trainingsdaten
X      % n x 3 Matrix
Y      % n x 1 Matrix

% Testdaten
X_test % N x 3 Matrix

% Berechnung der Hyperparameter
BoxConst=iqr(Y)/0.001;
EpsTube=iqr(Y)/200;

% Berechnung des reduzierten Modells
Mdl = fitcsvm(X,Y,'Standardize',true,'KernelFunction','mykernel',...
    'BoxConstraint',BoxConst,'Epsilon',EpsTube);

% Approximation für die Testdaten
Y_reg = predict(Mdl,X_test);
```

A.2 *Matèrn* 5/2 Kernel Funktion in MATLAB[®]

```
function G = mykernel(x_i,x_j)

    % Parameter für die Kernel Funktion
    para=1;

    % Differenz-Norm der Trainingsdatenpunkte
    D = pdist2(x_i,x_j,'euclidean');

    % Berechnung der Matèrn 5/2 Kernel Matrix
    G=(1+sqrt(5).* (D) ./para+(5/3) .* (D.^2) ./ (para^2)) ...
        .*exp(-sqrt(5) .* (D) ./para);

end
```

A.3 FE-Eingabedatensätze für die Parameterstudie in ABAQUS®

```

*Heading
** Job name: ParaStudy Model name: Model-1
** Generated by: Abaqus/CAE 2017
**Preprint, echo=NO, model=NO, history=YES, contact=NO
**
*PARAMETER
E=210000.
SigY=360.
DispMag=2.
**
** PARTS
**
*Part, name=Part-1
*Node
    1,          40.,          0.,          0.
    ⋮           ⋮           ⋮           ⋮
10976, 5.16880274, 4.28571415, 4.01488352
*Element, type=C3D8
    1, ... 1350
    ⋮  ⋮  ⋮
8953, ... 10905
*Nset, nset=Set-1, generate
    1, 10976, 1
*Elset, elset=Set-1, generate
    1, 8953, 1
** Section: Section-1
*Solid Section, elset=Set-1, material=Material-1
,
*End Part
**
**
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name=Part-1-1, part=Part-1]
*End Instance
**
*Nset, nset=Set-32, instance=Part-1-1
⋮
*Elset, elset=Set-35, instance=Part-1-1

```

```
*End Assembly
**
** MATERIALS
**
*Material, name=Material-1
*Elastic
<E>, 0.3
*Plastic
<SigY>,0.
** -----
**
** STEP: Step-1
**
*Step, name=Step-1, nlgeom=NO
*Static
0.25, 1., 1e-05, 1.
**
** BOUNDARY CONDITIONS
**
** Name: BC-1 Type: Displacement/Rotation
*Boundary
Set-32, 2, 2
** Name: BC-2 Type: Displacement/Rotation
*Boundary
Set-33, 1, 1
** Name: BC-3 Type: Displacement/Rotation
*Boundary
Set-34, 3, 3, <DispMag>
** Name: BC-4 Type: Displacement/Rotation
*Boundary
Set-35, 3, 3
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=0
**
** FIELD OUTPUT: F-Output-1
**
*Output, field
**
*Element Output, position=centroidal, directions=YES
PEEQ
*Output, history, frequency=0
*End Step
```

A.4 Programmcode für die Parameterstudie in ABAQUS®

```

#Parametric Study
BendingBeam=ParStudy(par=('E','SigY','DispMag'))
#Definig Parameter
BendingBeam.define(CONTINUOUS, par='E', domain=(190000., 230000.))
BendingBeam.define(CONTINUOUS, par='SigY', domain=(300., 360.))
BendingBeam.define(CONTINUOUS, par='DispMag', domain=(0.4, 2.8))
#Sampling
BendingBeam.sample(NUMBER, par='E', number=19)
BendingBeam.sample(NUMBER, par='SigY', number=19)
BendingBeam.sample(NUMBER, par='DispMag', number=19)
#Combine Samples
BendingBeam.combine(MESH, name='grid')
#Generate analysis
BendingBeam.generate(template='ParaStudy')
#Execute the jobs
BendingBeam.execute(ALL)
#Read Results
BendingBeam.output(step=1, overlay=ON, request=FIELD, file=ODB)
BendingBeam.gather(results='PEEQ_2153', variable='PEEQ', centroid=ON, ...
    element=2153, request=FIELD, instance="Part-1-1")
    :
BendingBeam.gather(results='PEEQ_5625', variable='PEEQ', centroid=ON, ...
    element=5625, request=FIELD, instance="Part-1-1")
BendingBeam.report(FILE,results=('PEEQ_2153',..., 'PEEQ_5625'), ...
par=('E','SigY','DispMag'), truncation=OFF, file='BendingBeam.PSE.psr')

```

Literaturverzeichnis

- Alameddin, S., Bhattacharyya, M., Fau, A., Nackenhorst, U., Néron, D., und Ladevèze, P. (2017). Large time increment approach for fatigue damage computations. *PAMM*, 17(1):231–232.
- An, C., Xie, C., Meng, Y., Shi, X., und Yang, C. (2020). Large Deformation Modeling of Wing-Like Structures Based on Support Vector Regression. *Applied Sciences*, 10(17):5995.
- Aubram, D. (2009). *Differential geometry applied to continuum mechanics*. Shaker Verlag GmbH.
- Barthold, F.-J. (2002). *Zur Kontinuumsmechanik inverser Geometrieprobleme*. Mechanik-Zentrum der Technischen Universität Braunschweig, Braunschweiger Schriften zur Mechanik Nr. 44–2002.
- Basudhar, A. und Missoum, S. (2010). An improved adaptive sampling scheme for the construction of explicit boundaries. *Structural and Multidisciplinary Optimization*, 42(4):517–529.
- Bertram, A. (1989). *Axiomatische Einführung in die Kontinuumsmechanik*. BI-Wiss.-Verlag.
- Bhattacharyya, M., Fau, A., Nackenhorst, U., Néron, D., und Ladevèze, P. (2018). A LATIN-based model reduction approach for the simulation of cycling damage. *Computational Mechanics*, 62(4):725–743.
- Campbell, C. und Ying, Y. (2011). Learning with support vector machines. *Synthesis lectures on artificial intelligence and machine learning*, 5(1):1–95.
- Carlberg, K. und Farhat, C. (2011). A low-cost, goal-oriented ‘compact proper orthogonal decomposition’ basis for model reduction of static systems. *International Journal for Numerical Methods in Engineering*, 86(3):381–402.
- Chaturantabut, S. und Sorensen, D. C. (2010). Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764.
- Chinesta, F., Cueto, E., Abisset-Chavanne, E., Duval, J. L., und El Khaldi, F. (2020). Virtual, digital and hybrid twins: a new paradigm in data-based engineering and engineered data. *Archives of computational methods in engineering*, 27(1):105–134.
- Chinesta, F., Huerta, A., Rozza, G., und Willcox, K. (2017). *Model Reduction Methods*. In *Encyclopedia of Computational Mechanics* (Herausgeber E. Stein, R. Borst und T.J.R. Huges). John Wiley & Sons, Ltd, 2. Auflage.
- Chinesta, F., Keunings, R., und Leygue, A. (2013). *The proper generalized decomposition for advanced numerical simulations: a primer*. Springer Science & Business Media.
- Cicczazzo, A., Di Pillo, G., und Latorre, V. (2014). Support vector machines for surrogate modeling of electronic circuits. *Neural Computing and Applications*, 24(1):69–76.
- Clarke, S. M., Griebisch, J. H., und Simpson, T. W. (2004). Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses. *Journal of Mechanical Design*, 127(6):1077–1087.

- Cristianini, N. und Shawe-Taylor, J. (2007). Support vector and kernel methods. In *Intelligent data analysis*, Seiten 169–197. Springer.
- Cristianini, N., Shawe-Taylor, J., u a. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.
- Cueto, E., González, D., und Alfaro, I. (2016). *Proper generalized decompositions: an introduction to computer implementation with Matlab*. Springer.
- de Boer, R. (1982). *Vektor- und Tensorrechnung für Ingenieure*. Springer.
- de Borst, R., Crisfield, M. A., Remmers, J. J., und Verhoosel, C. V. (2014). *Nichtlineare Finite-Elemente-Analyse von Festkörpern und Strukturen*. John Wiley & Sons.
- de Souza Neto, E. A., Peric, D., und Owen, D. R. (2008). *Computational Methods for Plasticity: Theory and Applications*. John Wiley & Sons.
- Epstein, M. (2010). *The geometrical language of continuum mechanics*. Cambridge University Press.
- Epstein, M. und Segev, R. (1980). Differentiable manifolds and the principle of virtual work in continuum mechanics. *Journal of Mathematical Physics*, 21(5):1243–1245.
- Fasshauer, G. E. und McCourt, M. J. (2015). *Kernel-based approximation methods using Matlab*, Band 19. World Scientific Publishing Company.
- Forrester, A., Sobester, A., und Keane, A. (2008). *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons.
- Franze, A. (2013). Ein Beitrag zur Behandlung nichtmaterieller Randbedingungen in der Kontinuumsmechanik. *Technische Universität Dresden, Berichte des Instituts für Mechanik und Flächentragwerke Heft 6*.
- Fuhg, J. N., Fau, A., und Nackenhorst, U. (2020). State-of-the-art and comparative review of adaptive sampling methods for kriging. *Archives of Computational Methods in Engineering*, Seiten 1–59.
- Garbo, A. (2019). *A sequential adaptive sampling technique based on a local linear model for computer experiment applications*. Dissertation, Georgia Institute of Technology.
- Garud, S. S., Karimi, I. A., und Kraft, M. (2016). Smart adaptive sampling for surrogate modelling. In *Computer Aided Chemical Engineering*, Band 38, Seiten 631–636. Elsevier.
- Gawehn, W. und Funk, S. (2020). *Finite Elemente Methode-FEM-Grundlagen zur Statik und Dynamik*. <http://www.mathematik-fundamente.de/>.
- Ghavamian, F., Tiso, P., und Simone, A. (2017). POD–DEIM model order reduction for strain-softening viscoplasticity. *Computer Methods in Applied Mechanics and Engineering*, 317:458–479.
- Hartmaier, A. (2020). Data-oriented constitutive modeling of plasticity in metals. *Materials*, 13(7):1600.
- Hernández, J., Oliver, J., Huespe, A. E., Caicedo, M., und Cante, J. (2014). High-performance model reduction techniques in computational multiscale homogenization. *Computer Methods in Applied Mechanics and Engineering*, 276:149–189.

- Ibrahimbegovic, A. (2009). *Nonlinear solid mechanics: theoretical formulations and finite element solution methods*, Band 160. Springer Science & Business Media.
- Jiang, C., Cai, X., Qiu, H., Gao, L., und Li, P. (2018). A two-stage support vector regression assisted sequential sampling approach for global metamodeling. *Structural and Multidisciplinary Optimization*, 58(4):1657–1672.
- Jiang, P., Zhou, Q., und Shao, X. (2020). *Surrogate Model-Based Engineering Design and Optimization*. Springer.
- Jin, R., Chen, W., und Sudjianto, A. (2002). On sequential sampling for global metamodelling in engineering design. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Band 36223, Seiten 539–548.
- Kollmann, F. und Hackenberg, H.-P. (1993). On the algebra of two-point tensors on manifolds with applications in nonlinear solid mechanics. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 73(11):307–314.
- Kreißig, R. und Benedix, U. (2013). *Höhere technische Mechanik: Lehr- und Übungsbuch*. Springer-Verlag.
- Kung, S. Y. (2014). *Kernel methods and machine learning*. Cambridge University Press.
- Lacaze, S. und Missoum, S. (2015). CODES: A toolbox for computational design. version 1.0. *University of Arizona*. URL: www.codes.arizona.edu/toolbox.
- Lychev, S. und Koifman, K. (2019). Geometry of Incompatible Deformations. In *Geometry of Incompatible Deformations*. De Gruyter.
- Mandelli, D. und Smith, C. (2012). Adaptive sampling using support vector machines. Idaho National Laboratory (INL). Technischer bericht.
- Marsden, J. E. und Hughes, T. J. (1994). *Mathematical foundations of elasticity*. Dover Publications, Inc. New York, Courier Corporation.
- Martynov, K. und Wever, U. (2019). On polynomial hyperreduction for nonlinear structural mechanics. *International Journal for Numerical Methods in Engineering*, 118(12):701–717.
- Miehe, C. (1993). *Kanonische Modelle multiplikativer Elasto-Plastizität: thermodynamische Formulierung und numerische Implementation*. Inst. f. Baumechanik u. Numerische Mechanik, Univ. Hannover.
- Millán, D. und Arroyo, M. (2013). Nonlinear manifold learning for model reduction in finite elastodynamics. *Computer Methods in Applied Mechanics and Engineering*, 261:118–131.
- Morales, G. (2021). „Center“: Calculates and plots centers of a triangle. *MATLAB Central File Exchange*. Retrieved July 15, 2021.
- Moustapha, M., Bourinet, J.-M., Guillaume, B., und Sudret, B. (2018). Comparative study of Kriging and support vector regression for structural engineering applications. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, 4(2):04018005.
- Moustapha, M. und Sudret, B. (2019). A two-stage surrogate modelling approach for the approximation of models with non-smooth outputs. *hal-02379136*.

- Mühlich, U. (2017). *Fundamentals of tensor calculus for engineers with a primer on smooth manifolds*, Band 230. Springer.
- Nackenhorst, U. (2004). The ALE-formulation of bodies in rolling contact: Theoretical foundations and finite element approach. *Computer Methods in Applied Mechanics and Engineering*, 193(39):4299–4322. The Arbitrary Lagrangian-Eulerian Formulation.
- Parisch, H. (2003). *Festkörper-Kontinuumsmechanik*. BG Teubner, 144.
- Patera, A. T. und Rozza, G. (2007). *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations, Version 1.0*. to appear (tentative rubric) in MIT Pappalardo Graduate Monographs in Mechanical Engineering. Copyright MIT 2006, Massachusetts Institute of Technology, Cambridge, MA, US.
- Quarteroni, A., Manzoni, A., und Negri, F. (2015). *Reduced basis methods for partial differential equations: an introduction*, Band 92. Springer.
- Quarteroni, A., Rozza, G., und Manzoni, A. (2011). Certified reduced basis approximation for parametrized partial differential equations and applications. *Journal of Mathematics in Industry*, 1(1):1–49.
- Romano, G. (2014). *Geometry & Continuum Mechanics*. CreateSpace Independent Publishing Platform.
- Romano, G., Barretta, R., und Diaco, M. (2014). Geometric continuum mechanics. *Meccanica*, 49(1):111–133.
- Rosipal, R., Trejo, L. J., und Cichocki, A. (2000). *Kernel principal component regression with em approach to nonlinear principal components extraction*. University of Paisley, Scotland, UK.
- Serani, A., D’Agostino, D., Campana, E. F., und Diez, M. (2020). Assessing the interplay of shape and physical parameters by unsupervised nonlinear dimensionality reduction methods. *Journal of Ship Research*, 64(04):313–327.
- Shawe-Taylor, J., Cristianini, N., u a. (2004). *Kernel methods for pattern analysis*. Cambridge university press.
- Smola, A. J. und Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222.
- Stein, E. (1989). *Nichtlineare Berechnungen im Konstruktiven Ingenieurbau*. Springer.
- Steiner, M., Bourinet, J.-M., und Lahmer, T. (2019). An adaptive sampling method for global sensitivity analysis based on least-squares support vector regression. *Reliability Engineering & System Safety*, 183:323–340.
- Steinmann, P. (2015). *Geometrical foundations of continuum mechanics*, Band 2 in *Lecture Notes in Applied Mathematics and Mechanics*. Springer.
- Stumpf, H. und Hoppe, U. (1997). The application of tensor algebra on manifolds to nonlinear continuum mechanics—invited survey article. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 77(5):327–339.

- Swischuk, R., Mainini, L., Peherstorfer, B., und Willcox, K. (2019). Projection-based model reduction: Formulations for physics-based machine learning. *Computers & Fluids*, 179:704–717.
- Wang, H., Li, E., Li, G., und Zhong, Z. (2008). Development of metamodeling based optimization system for high nonlinear engineering problems. *Advances in Engineering Software*, 39(8):629–645.
- Wang, H., Li, E., und Li, G. Y. (2009). The least square support vector regression coupled with parallel sampling scheme metamodeling technique and application in sheet forming optimization. *Materials & design*, 30(5):1468–1479.
- Wibowo, A. und Yamamoto, Y. (2012). A note on kernel principal component regression. *Computational Mathematics and Modeling*, 23(3):350–367.
- Wriggers, P. (2013). *Nichtlineare Finite-Element-Methoden*. Springer-Verlag.
- Xiang, H., Li, Y., Liao, H., und Li, C. (2017). An adaptive surrogate model based on support vector regression and its application to the optimization of railway wind barriers. *Structural and Multidisciplinary Optimization*, 55(2):701–713.
- Xu, S., Liu, H., Wang, X., und Jiang, X. (2014). A robust error-pursuing sequential sampling approach for global metamodeling based on voronoi diagram and cross validation. *Journal of Mechanical Design*, 136(7).
- Yaw, L. L. (2012). Nonlinear static—1D plasticity—various forms of isotropic hardening. *Walla Walla University*, 25.
- Zhang, Y., Kim, N. H., und Haftka, R. T. (2020). General-surrogate adaptive sampling using interquartile range for design space exploration. *Journal of Mechanical Design*, 142(5).
- Zhu, P., Pan, F., Chen, W., und Zhang, S. (2012). Use of support vector regression in structural optimization: application to vehicle crashworthiness design. *Mathematics and Computers in Simulation*, 86:21–31.

Lebenslauf

Steffen Funk

geboren am 16.03.1983 in Husum

Werdegang

- 03/2017 - 04/2022 **Leibniz Universität Hannover**
Fakultät für Bauingenieurwesen und
Geodäsie
Institut für Baumechanik und Numerische
Mechanik (IBNM)
Wissenschaftlicher Mitarbeiter und Pro-
motionsstudent
Promotion zum Dr.-Ing. (Note: sehr gut)
- 08/2012 – 02/2017 **Ingenieurbüro Dr. Schippke**
+ Partner
- 01/2012 – 07/2012 **Komatsu Hanomag GmbH**
- 03/2010 – 12/2011 **Fachhochschule Flensburg**
Masterstudiengang Systemtechnik
Master of Engineering (Note: 1,2)
- 09/2006 – 03/2010 **Fachhochschule Flensburg**
Bachelorstudiengang Maschinenbau
Bachelor of Engineering (Note: 1,8)
- 08/2003 – 06/2006 **Berufliche Schule des Kreises Nord-
friesland in Niebüll**
Fachgymnasium Fachbereich Technik
Abitur (Note: 2,3)

Forschungs- und Seminarberichte

Bisher in dieser Schriftenreihe erschienene Berichte:

- S 73/1 Seminar über Thermodynamik und Kontinuumsmechanik, Hannover 1973.
- F 75/1 “Die Spannungsberechnung im Rahmen der Finite-Element-Methode”, R. Ahmad, Dissertation, April 1975.
- F 76/1 “Zur Theorie und Anwendung der Stoffgleichungen elastisch-plastisch-viskoser Werkstoffe”, H. Mentlein, Dissertation, April 1976.
- F 77/1 Seminar über lineare und geometrisch nichtlineare Schalentheorie einschließlich Stabilitätstheorie, Hannover 1978.
- F 77/2 “Beitrag zur Berechnung von Gründungsplatten mit Hilfe der Finite-Element-Methode”, H. Meyer, Dissertation, Juli 1977.
- F 77/3 “Zur Berechnung der Eigenfrequenzen und Eigenschwingungsformen räumlich vorgekrümmter und vorverwundener Stäbe” J. Möhlenkamp, Dissertation, Dezember 1977.
- F 77/4 “Zur Theorie und Berechnung geometrisch und physikalisch nichtlinearer Kontinua mit Anwendung der Methode der finiten Elemente”, J. Paulun, Dissertation, Dezember 1977.
- F 78/1 2. Seminar über Thermodynamik und Kontinuumsmechanik, Hannover 1978.
- F 79/1 “Theoretische und numerische Behandlung geometrisch nichtlinearer viskoplastischer Kontinua”, K.-D. Klee, Dissertation, Februar 1979.
- F 79/2 “Zur Konstruierbarkeit von Variationsfunktionalen für nichtlineare Probleme der Kontinuumsmechanik”, J. Siefer, Dissertation, Oktober 1979.
- F 80/1 “Theoretische und numerische Behandlung gerader Stäbe mit endlichen Drehungen”, M. Kessel, Dissertation, Februar 1980.

- F 81/1 “Zur Berechnung von Kontakt- und Stoßproblemen elastischer Körper mit Hilfe der Finite-Element-Methode”, P. Wriggers, Dissertation, Januar 1981.95
- F 81/2 “Stoffgleichungen für Steinsalze unter mechanischer und thermischer Beanspruchung”, J. Olschewski, E. Stein, W. Wagner, D. Wetjen, geänderte Fassung eines Zwischenberichtes zum BMFT-Forschungsvorhaben KWA 1608/5.
- F 82/1 “Konvergenz und Fehlerabschätzung bei der Methode der Finiten Elemente”, R. Rohrbach, E. Stein, Abschlußbericht eines VW- Forschungsvorhabens, Februar 1982.
- F 82/2 “Alternative Spannungsberechnung in Finite-Element- Verschiebungsmodellen”, C. Klöhn, Dissertation, November 1982
- F 83/1 Seminar über nichtlineare Stabtheorie, Hannover 1983.
- F 83/2 “Beiträge zur nichtlinearen Theorie und inkrementellen Finite- Element-Berechnung dünner elastischer Schalen”, A. Berg, Dissertation, Juli 1983.
- F 83/3 “Elastoplastische Plattenbiegung bei kleinen Verzerrungen und großen Drehungen”, J. Paulun, Habilitation, September 1983.
- F 83/4 “Geometrisch nichtlineare FE-Berechnung von Faltwerken mit plastisch / viskoplastischem Deformationsverhalten”, M. Krog, Dissertation, Dezember 1983.
- F 85/1 Verleihung der Ehrendoktorwürde des Fachbereichs Bauingenieur- und Vermessungswesen der Universität Hannover an die Herren Prof. Dr. Drs. h.c. J.H. Argyris, Dr.-Ing. H. Wittmeyer.
- F 85/2 “Eine geometrisch nichtlineare Theorie schubelastischer Schalen mit Anwendung auf Finite-Element-Berechnungen von Durchschlag- und Kontaktproblemen”, W. Wagner, Dissertation, März 1985.
- F 85/3 “Geometrisch/physikalisch nichtlineare Probleme - Struktur und Algorithmen - ”, GAMM-Seminar im Februar 1985 in Hannover.
- F 87/1 “Finite-Elemente-Berechnungen ebener Stabtragwerke mit Fließgelenken und großen Verschiebungen”, R. Kahn, Dissertation, Oktober 1987.
- F 88/1 “Theorie und Numerik schubelastischer Schalen mit endlichen Drehungen unter Verwendung der Biot-Spannungen”, F. Gruttmann, Dissertation, Juni 1988.

- F 88/2 “Optimale Formgebung von Stabtragwerken mit Nichtlinearitäten in der Zielfunktion und in den Restriktionen unter Verwendung der Finite-Element-Methode”, V. Berkhahn, Dissertation, Oktober 1988.
- F 88/3 “Beiträge zur Theorie und Numerik großer plastischer und kleiner elastischer Deformationen mit Schädigungseinfluß”, R. Lammering, Dissertation, November 1988.
- F 88/4 “Konsistente Linearisierungen in der Kontinuumsmechanik und ihrer Anwendung auf die Finite-Elemente-Methode”, P. Wriggers, Habilitation, November 1988.96
- F 88/5 “Mathematische Formulierung und numerische Methoden für Kontaktprobleme auf der Grundlage von Extremalprinzipien”, D. Bischoff, Habilitation, Dezember 1988.
- F 88/6 “Zur numerischen Behandlung thermomechanischer Prozesse”, C. Miehe, Dissertation, Dezember 1988.
- F 89/1 “Zur Stabilität und Konvergenz gemischter finiter Elemente in der linearen Elastizitätstheorie”, R. Rolfes, Dissertation, Juni 1989.
- F 89/2 “Traglastberechnungen von Faltenwerken mit elastoplastischen Deformationen”, K.-H. Lambertz, Dissertation, November 1989.
- F 89/3 “Transientes Kriechen und Kriechbruch im Steinsalz”, U. Heemann, Dissertation, November 1989.
- F 89/4 “Materialgesetze zum Verhalten von Betonkonstruktionen bei harten Stößen”, E. Stein, P. Wriggers, T. Vu Van & T. Wedemeier, Dezember 1989.
- F 89/5 “Lineare Konstruktion und Anwendungen von Begleitmatrizen”, C. Carstensen, Dissertation, Dezember 1989.
- F 90/1 “Zur Berechnung prismatischer Stahlbetonbalken mit verschiedenen Querschnittformen für allgemeine Beanspruchungen”, H. N. Lucero-Cimas, Dissertation, April 1990.
- F 90/2 “Zur Behandlung von Stoß- Kontaktproblemen mit Reibung unter Verwendung der Finite-Element-Methode”, T. Vu Van, Dissertation, Juni 1990.
- F 90/3 “Netzadaption und Mehrgitterverfahren für die numerische Behandlung von Faltenwerken”, L. Plank, Dissertation, September 1990.

- F 90/4 “Beiträge zur Theorie und Numerik finiter inelastischer Deformationen”, N. Müller-Hoeppe, Dissertation, Oktober 1990.
- F 90/5 “Beiträge zur Theorie und Numerik von Materialien mit innerer Reibung am Beispiel des Werkstoffes Beton”, T. Wedemeier, Dissertation, Oktober 1990.
- F 91/1 “Zur Behandlung von Stabilitätsproblemen der Elastostatik mit der Methode der Finiten Elemente”, W. Wagner, Habilitation, April 1991.
- F 91/2 “Mehrgitterverfahren und Netzadaption für lineare und nichtlineare statische Finite-Elemente-Berechnungen von Flächentragwerken”, W. Rust, Dissertation, Oktober 1991.
- F 91/3 “Finite Elemente Formulierung im Trefftzchen Sinne für dreidimensionale anisotrop-elastische Faserverbundstrukturen”, K. Peters, Dissertation, Dezember 1991.
- F 92/1 “Einspielen und dessen numerische Behandlung von Flächentragwerken aus ideal plastischem bzw. kinematisch verfestigendem Material”, G. Zhang, Dissertation, Februar 1992.97
- F 92/2 “Strukturoptimierung stabilitätsgefährdeter Systeme mittels analytischer Gradientenermittlung”, A. Becker, Dissertation, April 1992.
- F 92/3 “Duale Methoden für nichtlineare Optimierungsprobleme in der Strukturmechanik”, R. Mahnken, Dissertation, April 1992.
- F 93/1 “Kanonische Modelle multiplikativer Elasto-Plastizität. Thermodynamische Formulierung und numerische Implementation”, C. Miehe, Habilitation, Dezember 1993.
- F 93/2 “Theorie und Numerik zur Berechnung und Optimierung von Strukturen aus isotropen, hyperelastischen Materialien”, F.-J. Barthold, Dissertation, Dezember 1993.
- F 94/1 “Adaptive Verfeinerung von Finite-Element-Netzen für Stabilitätsprobleme von Flächentragwerken”, E. Stein, B. Seifert, W. Rust, Forschungsbericht, Oktober 1994.
- F 95/1 “Adaptive Verfahren für die Formoptimierung von Flächentragwerken unter Berücksichtigung der CAD-FEM-Kopplung”, A. Falk, Dissertation, Juni 1995.
- F 96/1 “Theorie und Numerik dünnwandiger Faserverbundstrukturen”, F. Gruttmann, Habilitation, Januar 1996.

- F 96/2 “Zur Theorie und Numerik finiter elastoplastischer Deformationen von Schalenstrukturen”, B. Seifert, Dissertation, März 1996.
- F 96/3 “Theoretische und algorithmische Konzepte zur phänomenologischen Beschreibung anisotropen Materialverhaltens”, J. Schröder, Dissertation, März 1996.
- F 96/4 “Statische und dynamische Berechnungen von Schalen endlicher elastischer Deformationen mit gemischten finiten Elementen”, P. Betsch, Dissertation, März 1996.
- F 96/5 “Kopplung von Finiten Elementen und Randelementen für ebene Elastoplastizität mit Impelementierung auf Parallelrechnern”, M. Kreienmeyer, Dissertation, März 1996.
- F 96/6 “Theorie und Numerik dimensions- und modeladaptiver Finite-Elemente-Methoden von Flächentragwerken”, S. Ohnimus, Dissertation, Juni 1996.
- F 96/7 “Adaptive Finite Elemente Methoden für MIMD-Parallelrechner zur Behandlung von Strukturproblemen mit Anwendung auf Stabilitätsprobleme”, O. Klaas, Dissertation, Juli 1996.
- F 96/8 “Institutsbericht 1971-1996 aus Anlaß des 25-jährigen Dienstjubiläums von Prof. Dr.-Ing. Dr.-Ing. E.h. Dr. h.c. mult. Erwin Stein”, Dezember 1996.
- F 97/1 “Modellierung und Numerik duktiler kristalliner Werkstoffe”, P. Steinmann, Habilitation, August 1997.
- F 97/2 “Formoptimierung in der Strukturodynamik”, L. Meyer, Dissertation, September 1997.
- F 97/3 “Modellbildung und Numerik für Versagensprozesse in Gründungen von Caissonwellenbrechern”, M. Lengnick, Dissertation, November 1997.
- F 98/1 “Adaptive gemischte finite Elemente in der nichtlinearen Elastostatik und deren Kopplung mit Randelementen”, U. Brink, Dissertation, Februar 1998.
- F 98/2 “Theoretische und numerische Aspekte zur Parameteridentifikation und Modellierung bei metallischen Werkstoffen”, R. Mahnken, Habilitation, Juli 1998.
- F 98/3 “Lokalisierung und Stabilität der Deformation wassergesättigter bindiger und granularer Böden”, J. M. Panesso, Dissertation, August 1998.

- F 98/4 “Theoretische und numerische Methoden in der angewandten Mechanik mit Praxisbeispielen”, R. Mahnen (Hrsg.), Festschrift anlässlich der Emeritierung von Prof. Dr.-Ing. Dr.-Ing. E.h. h.c. mult. Erwin Stein, November 1998.
- F 99/1 “Eine h-adaptive Finite-Element-Methode für elasto-plastische Schalenproblem in unilateralem Kontakt”, C.-S. Han, Dissertation, Juli 1999.
- F 00/1 “Ein diskontinuierliches Finite-Element-Modell für Lokalisierungsversagen in metallischen und granularen Materialien”, C. Leppin, Dissertation, März 2000.
- F 00/2 “Untersuchungen von Strömungen in zeitlich veränderlichen Gebieten mit der Methode der Finiten Elementen”, H. Braess, Dissertation, März 2000.
- F 00/3 “Theoretische und algorithmische Beiträge zur Berechnung von Faserverbundschalen”, J. Tessmer, Dissertation, März 2000.
- F 00/4 “Theorie und Finite-Element-Methode für die Schädigungsbeschreibung in Beton und Stahlbeton”, D. Tikhomirov, Dissertation, August 2000.
- F 01/1 “A C1 - continuous formulation for finite deformation contact”, L. Krstulovic-Opara, Dissertation, Januar 2001.
- F 01/2 “Strain Localisation Analysis for Fully and Partially Saturated Geomaterials”, H. Zhang, Dissertation, Januar 2001.
- F 01/3 “Meso-makromechanische Modellierung von Faserverbundwerkstoffen mit Schädigung”, C. Döbert, Dissertation, April 2001.
- F 01/4 “Thermomechanische Modellierung gummiartiger Polymerstrukturen”, S. Reese, Habilitation, April 2001.
- F 01/5 “Thermomechanisches Verhalten von Gummimaterialien während der Vulkanisation - Theorie und Numerik -”, M. Andre, Dissertation, April 2001.
- F 01/6 “Adaptive FEM für elastoplastische Deformationen - Algorithmen und Visualisierung”, M. Schmidt, Dissertation, Juni 2001.
- F 01/7 “Verteilte Algorithmen für h-, p- und d-adaptive Berechnungen in der nichtlinearen Strukturmechanik”, R. Niekamp, Dissertation, Juni 2001.99
- F 01/8 “Theorie und Numerik zur Berechnung und Optimierung von Strukturen mit elastoplastischen Deformationen”, K. Wiechmann, Dissertation, Juli 2001.

- F 01/9 “Direct Computation of Instability Points with Inequality using the Finite Element Method”, H. Tschöpe, Dissertation, September 2001.
- F 01/10 “Theorie und Numerik residualer Fehlerschätzer für die Finite-Elemente-Methode unter Verwendung äquibrierter Randspannungen”, S. Ohnimus, Habilitation, September 2001.
- F 02/1 “Adaptive Algorithmen für thermo-mechanisch gekoppelte Kontaktprobleme”, A. Rieger, Dissertation, August 2002.
- F 02/2 “Consistent coupling of shell- and beam-models for thermo-elastic problems”, K.Chavan, Dissertation, September 2002.
- F 03/1 “Error-controlled adaptive finite element methods in large strain hyperelasticity and fracture mechanics”, M. Rüter, Dissertation, Mai 2003.
- F 03/2 “Formulierung und Simulation der Kontaktvorgänge in der Baugrund-Tragwerks- Interaktion”, A. Haraldsson, Dissertation, Juni 2003.
- F 03/3 “Concepts for Nonlinear Orthotropic Material Modeling with Applications to Membrane Structures”, T. Raible, Dissertation, Juni 2003.
- F 04/1 “On Single- and Multi-Material arbitrary Lagrangian-Eulerian Approaches with Application to Micromechanical Problems at Finite Deformations”, D. Freßmann, Dissertation, Oktober 2004.
- F 04/2 “Computational Homogenization of Microheterogeneous Materials at Finite Strains Including Damage”, S. Löhnert, Dissertation, Oktober 2004.
- F 05/1 “Numerical Micro-Meso Modeling of Mechanosensation driven Osteonal Remodeling in Cortical Bone”, C. Lenz, Dissertation, Juli 2005.
- F 05/2 “Mortar Type Methods Applied to Nonlinear Contact Mechanics”, K.A. Fischer, Dissertation, Juli 2005.
- F 05/3 “Models, Algorithms and Software Concepts for Contact and Fragmentation in Computational Solid Mechanics”, C. Hahn, Dissertation, November 2005.
- F 06/1 “Computational Homogenization of Concrete”, S. Moftah, Dissertation, Januar 2006.
- F 06/2 “Reduction Methods in Finite Element Analysis of Nonlinear Structural Dynamics”, H. Spiess, Dissertation, Februar 2006.

- F 06/3 “Theoretische und algorithmische Konzepte zur Beschreibung des beanspruchungsadaptiven Knochenwachstums”, B. Ebbecke, Dissertation, März 2006.100
- F 06/4 “Experimentelle Untersuchungen an elastomeren Werkstoffen”, M. Dämgen, Dissertation, Dezember 2006.
- F 07/1 “Numerische Konzepte zur Behandlung inelastischer Effekte beim reibungsbehafteten Rollkontakt”, M. Zieffe, Dissertation, Februar 2007.
- F 07/2 “Begleitbuch zur Leibniz-Ausstellung”, Hrsg: E. Stein, P. Wriggers, 2007.
- F 07/3 “Modellierung und Simulation der hochfrequenten Dynamik rollender Reifen”, M. Brinkmeier, Dissertation, Juni 2007.
- F 07/4 “Computational Homogenization of micro-structural Damage due to Frost in Hardened Cement Paste”, M. Hain, Dissertation, Juli 2007.
- F 07/5 “Elektromechanisch gekoppelte Kontaktmodellierung auf Mikroebene”, T. Helmich, Dissertation, August 2007.
- F 07/6 “Dreidimensionales Diskretes Elemente Modell für Superellipsoide”, C. Lillie, Dissertation, Oktober 2007.
- F 07/7 “Adaptive Methods for Continuous and Discontinuous Damage Modeling in Fracturing Solids”, S.H. Reese, Dissertation, Oktober 2007.
- F 08/1 “Student Projects of Micromechanics”, Hrsg: U. Nackenhorst, August 2008.
- F 09/1 “Theory and Computation of Mono- and Poly- crystalline Cyclic Martensitic Phase Transformations”, G. Sagar, Dissertation, August 2009.
- F 09/2 “Student projects of Micromechanics”, D. Balzani and U. Nackenhorst, Course Volume, Oktober 2009.
- F 09/3 “Multiscale Coupling based on the Quasicontinuum Framework, with Application to Contact Problems”, W. Shan, Dissertation, November 2009.
- F 10/1 “A Multiscale Computational Approach for Microcrack Evolution in Cortical Bone and Related Mechanical Stimulation of Bone Cells”, D. Kardas, Dissertation, September 2010.
- F 11/1 “Ein Integrales Modellierungskonzept zur numerischen Simulation der Osseointegration und Langzeitstabilität von Endoprothesen”, A.Lutz, Dissertation, Oktober 2011.

- F 12/1 “Ein physikalisch motiviertes Reifen-Fahrbahnmodell für die Gesamtfahrzeugsimulation”, R. Chiarello, Dissertation, Februar 2012.
- F 13/1 “Thermomechanical Analysis of Tire Rubber Compounds in Rolling Contact”, A.Suwannachit, Dissertation, September 2012.
- F 13/2 “Towards a Finite Element Model for Fluid Flow in the Human Hip Joint”, K. Fietz, Dissertation, September 2013.101
- F 14/1 “Micro-Mechanically Based Damage Analysis of Ultra High Performance Fibre Reinforced Concrete Structures with Uncertainties”, A. Hürkamp, Dissertation, Dezember 2013.
- F 14/2 “Numerical Solution of High-Dimensional Fokker-Planck Equations with Discontinuous Galerkin Methods”, F. Loerke, Dissertation, Dezember 2013.
- F 14/3 “Numerische Simulation probabilistischer Schädigungsmodelle mit der Stochastischen Finite Elemente Methode”, P. Jablonski, Dissertation, September 2014.
- F 15/1 “On a Finite Element Approach for the Solution of a Mechanically Stimulated Biochemical Fracture Healing Model”, A. Sapotnick, Dissertation, November 2015.
- F 15/2 “Simulation of Elastic-Plastic Material Behaviour with Uncertain Material Parameters. A Spectral Stochastic Finite Element Method Approach”, S. Fink, Dissertation, November 2015.
- F 15/3 “A Fully Micro-mechanically Motivated Material Law for Filled Elastomer”, O. Stegen, Dissertation, Februar 2016.
- F 16/1 “A modified adaptive harmony search algorithm approach on structural identification and damage detection”, M.Jahjouh, Dissertation, Januar 2016,
- F17/1 “Computation Simulation of Piezo-electrically Stimulated Bone Adaption Surrounding Activated Teeth Implants”, A.Shirazibeheshtiha, Dissertation, Januar 2017.
- F 17/2 “A Constitutive Contact Model for Homogenized Tread-Road Interaction in Rolling Resistance Computations”, R.Bayer, Dissertation, Februar 2017.
- F 17/3 “A Posteriori Error Estimates for Advanced Galerkin Methods”, M.O. Rüter, Habilitation, November 2017.

- F 17/4 “Probabilistische Finite Element Modellierung des mechanischem Materialverhaltens von Salzgestein”, M. Grehn, Dissertation, Dezember 2017.
- F 18/1 “Modelling and numerical simulation for the prediction of the fatigue strength of airsprings”, N.K.Jha, Dissertation, März 2018.
- F 18/2 “A model reduction approach in space and time for fatigue damage simulation”, M.Bhattacharya, Dissertation, Mai 2018.
- F 18/3 “Numerical investigation on hydrogen embrittlement of metallic pipeline structures”, M.Möhle, Dissertation, Mai 2018.
- F 18/5 “A stochastic fatigue model for casted aluminium structures”, Govindarajan Narayanan, Dissertation, August 2018.
- F 20/1 “A Micro-mechanically Motivated Approach for Modelling the Oxidative Aging Process of Elastomers”, Darcy Beurle, Dissertation, December 2019.
- F 20/2 “A Semi-incremental Model Order Reduction Approach for Fatigue Damage Computations”, Shadi Alameddin, Dissertation, January 2020.
- F 20/3 “A Coupled ALE Lagrangian Approach for the Simulation of Treaded Tires”, Thirumal Alagu Palanichamy, Dissertation, August 2020.
- F 21/1 “A parametric modeling concept for predicting biomechanical compatibility in total hip arthroplasti”, Maximilian Bittens, Dissertation, July 2021.
- F 21/2 “Stochastic Modelling and Numerical Simulation of Fatigue Damage”, Weiran Zhang, Dissertation, July 2021.

In der vorliegenden Arbeit werden Untersuchungen zur Modellreduktion mechanischer Systeme mit elasto-plastischem Materialverhalten durchgeführt. Das untersuchte Verfahren zur Modellreduktion ist in dieser Arbeit die Support Vektor Regression (SVR). Nach einem Überblick zum Stand der Technik im Rahmen der Modellreduktion mechanischer Systeme, wird die Theorie der Kontinuumsmechanik und der Finiten Elemente Methode (FEM) bereitgestellt. Im Anschluss an eine ausführliche Darlegung des mathematischen Hintergrunds der Support Vektor Regression wird die Methode auf den eindimensionalen, rein phänomenologischen elasto-plastischen Fall angewendet. Ein wesentliches Kapitel widmet sich der Anwendung auf den drei- und zweidimensionalen elasto-plastischen Berechnungsfall unter vorangegangener Finite-Elemente Berechnung.

In den Studien wird gezeigt, dass die Matérn 5/2 Kernel Funktion sehr gut für Anwendungen im Bereich der Elasto-Plastizität geeignet und mit entsprechend gewählten Parametern anderen Kernel Funktionen überlegen ist. Mit verschiedenen, in dieser Dissertation vorgestellten adaptiven Sampling-Methoden kann eine teilweise Verbesserung im Gegensatz zu der homogenen, kartesischen Verfeinerung erzielt werden. Im Rahmen der FE-Anwendung zeigt sich die eigentliche Stärke der Modellreduktion mittels SVR. Die ungleichmäßigen und nicht stetigen Antwortflächen der gewählten Größe von Interesse können mit der Support Vektor Regression hinreichend präzise angenähert werden. Die Sensitivität in den verschiedenen Parameter-Richtungen lässt eine zusätzliche Reduktion der nötigen Trainingsdaten mittels anisotropen Gitterstrukturen zu.