



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Masterarbeit

Tim von Ahlen

**Optimale Auslegung von Luftfahrtstrukturen aus  
Faserverbundwerkstoffen mit krummlinig  
verlaufenden Faserorientierungen**

*Fakultät Technik und Informatik  
Studiendepartment Fahrzeugtech-  
nik und Flugzeugbau*

*Faculty of Engineering and Computer Science  
Department of Automotive and Aeronautical En-  
gineering*

Tim von Ahlen

**Optimale Auslegung von Luftfahrtstrukturen aus  
Faserverbundwerkstoffen mit krummlinig verlaufenden  
Faserorientierungen**

Masterarbeit eingereicht im Rahmen der Masterprüfung

im Studiengang Master of Science Flugzeugbau  
am Department Fahrzeugtechnik und Flugzeugbau  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuende Prüfer

Erstprüfer: Prof. Dr. Michael Seibel  
Zweitprüfer: Prof. Dr.-Ing. Jens Baaran

Eingereicht am: 25.05.2016

**Tim von Ahlen**

**Thema der Arbeit**

Optimale Auslegung von Luftfahrtstrukturen aus Faserverbundwerkstoffen mit krummlinig verlaufenden Faserorientierungen

**Stichworte**

Krummlinige Faserverläufe, Faserverbundwerkstoffe, Strukturoptimierung, Bezier-Fläche

**Kurzzusammenfassung**

Ziel der Arbeit ist die Erstellung eines Optimierungsprogramms zur automatischen Auslegung idealer Faserverläufe. Anhand eines Beispiels wird die Funktionalität des Tools nachgewiesen und das Potential krummliniger Faserverläufe zur Verbesserung mechanischer Eigenschaften aufgezeigt. Im Beispiel wird der Index des Festigkeitskriteriums von Tsai-Wu gegenüber einer konventionellen Auslegung um bis zu 87 Prozent gesenkt. Die Modellierung der Faserverläufe erfolgt durch individuelle Winkel in allen finiten Elementen. Mittelpunkt des Programms ist eine Funktion, die mittels einer Bezier-Fläche aus wenigen Variablen eine deutlich größere Anzahl Winkel generiert. Um die Prozessierbarkeit einzubeziehen, wird die Faserkrümmung als Restriktion betrachtet.

**Tim von Ahlen**

**Title of the paper**

Optimum design of airframe structures made of composite materials using curvilinear fiber paths

**Keywords**

Curvilinear fiber paths, Composites, Structure optimization, Bezier surface

**Abstract**

Aim of the study is the creation of an optimisation tool that automatically designs ideal fibre paths. The functionality and the potentials of curvilinear fiber paths are demonstrated by an example calculation, in which the index of the Tsai–Wu failure criterion is decreased by 87 per cent referring to a conventional lay-up. The fibre paths are modelled with individual angles for each finite element. Centre point of the tool is a function that generates many different angles by few variables. The processibility is taken into account by a curvature restriction.

Master Thesis  
- Tim von Ahlen -

**Optimale Auslegung von Luftfahrtstrukturen aus  
Faserverbundwerkstoffen mit krummlinig verlaufenden Faserorientierungen**

**Optimum design of airframe structures  
made of composite materials using curvilinear fiber paths**

General description:

Glass and carbon fiber reinforced plastics are used since the nineteen-eighties in primary structures of civil transport aircrafts. Till this day, the design, the analysis and the manufacturing of these composite structures is still under development in order to take full advantage of all their beneficial characteristics. Especially the mechanical properties are of interest to the structural design, due to the superior stiffness to mass and strength to mass ratios.

One of the most difficult challenges during the design is to define the optimum laminate for the envisaged purposes, viz to find the optimum thickness distribution as well as to align the fibers along the load paths. Due to the quantity of potential solutions, the application of trial and error procedures or engineering judgment to find suitable designs and laminate definitions is not reasonable. In fact, numerical optimization procedures should be used to identify the optimum structural configuration and laminate definition.

Objective:

The objective of the master thesis is to develop a tool to describe two-dimensional curvilinear fiber paths for individual plies of a laminate. The traditional design feature of using straight fiber paths only shall be overcome and replaced by curvilinear fiber paths. In other words, rather than designing laminates with constant ply-angles, each ply shall be designed with curvilinear fiber paths that improve the mechanical properties of the structure. Thus, the advantageous properties of fiber reinforced composites may be fully exploited as their fibers are properly placed in their optimal spatial orientations. The manufacturing of such laminates is possible using tow placement (fiber placement) machines. They are capable of steering multiple tows in prescribed paths with constraints on minimum radius of curvature depending on the tow width.

The developed tool shall be implemented in a numerical optimization procedure to demonstrate its capability and performance based on the use case of unstiffened and stiffened, rectangular composite panels subjected to strength and buckling loads. The curvilinear fiber paths shall be derived from the contour lines of a Beziér-surface (two-dimensional parametric function), as depicted in Fig. 1. Parametric studies shall be executed in order to find suitable numbers of control points (i.e. required control point density) that allows an adequate and sound description of the fiber paths of an individual ply of a laminate.

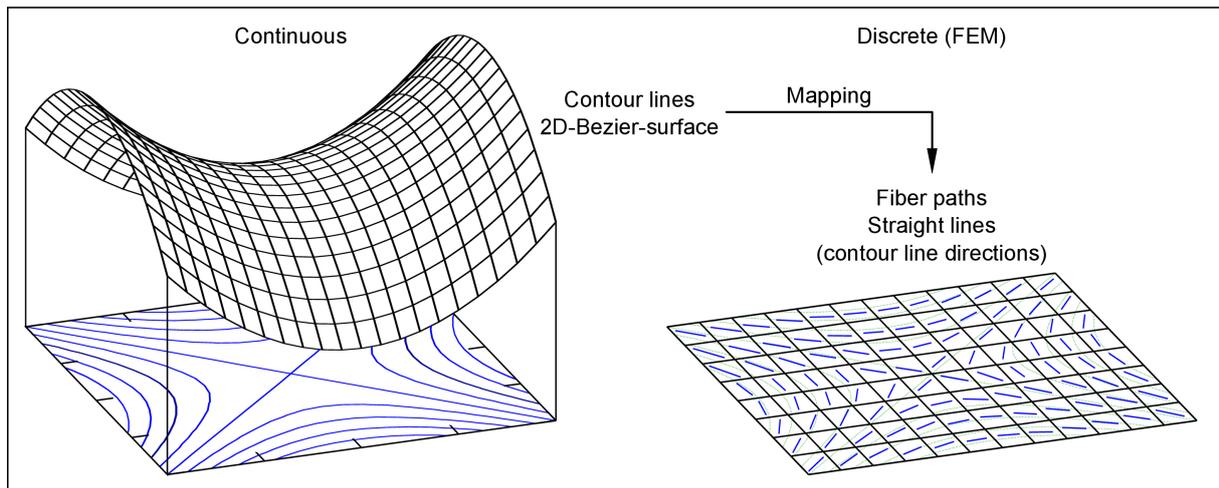


Fig. 1: Two-dimensional curvilinear fiber paths for individual plies of a laminate based on 2D- parametric surface

#### Workflow:

- Familiarization concerning:
  - fiber reinforced plastics, numerical optimization, 2D parametric surface descriptions
- Development of the mathematical tool, describing two-dimensional curvilinear fiber paths for individual plies of a laminate based on parametric surface descriptions
- System programming and implementation:
  - Optimization procedure: Matlab®
  - Structural analysis: MSC/Nastran®
  - Programming language: Matlab®
- Performance demonstration using a notched composite plate considering strength constraints
- Performance demonstration using a stiffened or unstiffened plate considering buckling constraints
- Execution of a parametric study to find a suitable control point density of the Beziér-surface, from which the contour lines are derived
- Development and application of methods to consider changes of the fiber paths within a single layer (e.g. curvature of fiber paths)
- Development and application of methods to consider differences of the fiber orientations in adjacent layers
- Detailed documentation (in English or German language) of the work performed in accordance with the guidelines of the department of automotive and aeronautical engineering

Supervisor:

Prof. Dr.-Ing. Michael Seibel

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>viii</b>
<b>Tabellenverzeichnis</b>	<b>ix</b>
<b>Abbildungsverzeichnis</b>	<b>x</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Motivation der Arbeit . . . . .	1
1.2. Ziel der Arbeit . . . . .	2
1.3. Aufbau der Arbeit . . . . .	3
<b>2. Literaturrecherche</b>	<b>4</b>
2.1. Honda-Untersuchung . . . . .	4
2.2. Weitere Untersuchungen . . . . .	5
<b>3. Theoretische Grundlagen</b>	<b>10</b>
3.1. Optimierungsprozedur . . . . .	10
3.2. Strukturoptimierung . . . . .	10
3.2.1. Allgemeines . . . . .	11
3.2.2. Entwurfsvariablen . . . . .	12
3.2.3. Zielfunktion . . . . .	13
3.2.4. Restriktionen . . . . .	14
3.2.5. Algorithmen . . . . .	15
3.2.6. Konvexität . . . . .	16
3.2.7. Sensitivitätsanalyse . . . . .	17
3.2.8. Optimierungs-Programme . . . . .	17
3.2.9. Universal Framework for Optimization . . . . .	18
3.3. Faserverbundwerkstoffe . . . . .	18
3.3.1. Allgemeines FVW . . . . .	19
3.3.2. Faserverläufe . . . . .	19
3.3.3. Herstellungsverfahren . . . . .	20
3.3.4. Mechanische Kennwerte . . . . .	21
3.3.5. Festigkeitskriterien . . . . .	21
3.4. FEM . . . . .	22
3.4.1. Allgemeines FEM . . . . .	23
3.4.2. Art der Berechnung . . . . .	23
3.4.3. FE-Programme . . . . .	24

3.4.4.	Relevante Dateiformate . . . . .	24
3.4.5.	Parameter-Karten . . . . .	25
3.5.	Mathematik . . . . .	26
3.5.1.	Bezierflächen . . . . .	26
3.5.2.	Bestimmung geometrischer Schwerpunkte . . . . .	28
3.5.3.	Funktionsableitungen . . . . .	28
3.5.4.	Kreisselmeier-Steinhauser-Funktion . . . . .	29
<b>4.</b>	<b>FE-Modellbildung</b>	<b>30</b>
4.1.	Vorüberlegungen zum FE-Modell . . . . .	30
4.1.1.	Ansatz der Umsetzung . . . . .	31
4.1.2.	Unterschiede bei einem FVW-Modell . . . . .	31
4.1.3.	Rechenansatz und Vernetzung . . . . .	32
4.1.4.	Vereinfachungen des Modells . . . . .	32
4.1.5.	Art der Einbindung . . . . .	33
4.2.	Realisierung FE-Modell . . . . .	34
4.2.1.	Eingangsparameter . . . . .	34
4.2.2.	Geometrie . . . . .	36
4.2.3.	Vernetzung . . . . .	36
4.2.4.	Material/Eigenschaften . . . . .	38
4.2.5.	Äußere Kräfte und Randbedingungen . . . . .	39
4.2.6.	Ausgangsparameter . . . . .	40
4.3.	Erzeugung von Vergleichswerten . . . . .	40
4.4.	Validierung des Modells . . . . .	41
<b>5.</b>	<b>Bézier-Tool</b>	<b>43</b>
5.1.	Ansatz Variablenerzeugung . . . . .	43
5.2.	Realisierung Kurvenerzeugung . . . . .	45
5.3.	Anwendung Béziertool . . . . .	46
5.4.	Ausgabewerte . . . . .	46
5.4.1.	Winkel . . . . .	46
5.4.2.	Krümmung . . . . .	47
5.5.	Geometrische Restriktionen . . . . .	47
5.6.	Programmiertechnische Umsetzung des Bézier-Tools . . . . .	48
5.6.1.	Eingabeparameter . . . . .	48
5.6.2.	Berechnung der Bézierfläche . . . . .	49
5.6.3.	Erzeugung der Höhenschnittlinien . . . . .	50
5.6.4.	Berechnung der Winkel und Krümmung . . . . .	52
5.6.5.	Methoden zur Begrenzung/Winkelrestriktionen . . . . .	53
5.6.6.	Darstellung . . . . .	54
5.7.	Problematiken . . . . .	55
5.7.1.	Probleme des Ansatzes . . . . .	55
5.7.2.	Probleme der programmiertechnischen Umsetzung . . . . .	57

<b>6. Optimierungstool</b>	<b>58</b>
6.1. Ansatz der Optimierung	58
6.1.1. Einbindung des UFO-Tools	58
6.1.2. Entwurfsvariablen	59
6.1.3. Algorithmus	59
6.1.4. Zielfunktion	60
6.1.5. Restriktionen	61
6.2. Programmiertechnische Umsetzung	62
6.2.1. Aufbau des Optimierungstools	62
6.2.2. Hauptmodule	67
6.2.3. Untermodule	70
6.2.4. Hilfsmodule	76
6.3. Anwendung des Optimierungstools	77
6.3.1. Eingabe	77
6.3.2. Ausgabe	80
6.4. Sensitivitätsanalyse	81
<b>7. Demonstration des Optimierungstools</b>	<b>84</b>
7.1. Festigkeitsbetrachtung einer gelochten Platte	84
7.1.1. Unrestringierte Optimierung	84
7.1.2. Restringierte Optimierung	89
7.1.3. Auswertung der Mehrzieloptimierung	91
7.2. Parameterstudie zu Kontrollpunkten	92
7.2.1. z-Koordinaten der Kontrollpunkte	92
7.2.2. Anzahl der Kontrollpunkte	94
7.2.3. Position der Kontrollpunkte	96
7.2.4. Kombination	99
<b>8. Auswertung und Fazit</b>	<b>101</b>
8.1. Auswertung	101
8.1.1. Betrachtung der Optimierungsergebnisse	101
8.1.2. Verifikation mit Referenzwerten	103
8.2. Verbesserungsmöglichkeiten	107
8.3. Fazit	110
8.4. Ausblick	111
<b>Literaturverzeichnis</b>	<b>115</b>
<b>A. Anhang Darstellungen zu 7.2.1</b>	<b>115</b>
<b>B. Anhang Darstellungen zu 7.2.3</b>	<b>119</b>
<b>C. Anhang Darstellungen zu 7.2.4</b>	<b>122</b>

# Tabellenverzeichnis

3.1. Arten von Restriktionen . . . . .	15
3.2. Restriktionsfälle . . . . .	15
3.3. Ablauf Optimierungsalgorithmus [ <b>Schumacher 2013</b> ] . . . . .	16
3.4. Materialparameter / Basisfestigkeiten . . . . .	21
3.5. Optionen PCOMP-Karte . . . . .	25
3.6. Optionen MAT8-Karte . . . . .	26
4.1. Abmaße der Lochplatte . . . . .	35
4.2. Materialkonstanten . . . . .	36
4.3. Einteilung Mesh-Seeds . . . . .	37
4.4. Konvergenzstudie . . . . .	42
5.1. Eingabeparameter des Bézier-Tools . . . . .	49
6.1. Optionale Parameter für Nastran . . . . .	79
6.2. Weitere Eingabeparameter . . . . .	80
6.3. Differenzen der Strukturantwort . . . . .	83
7.1. Vergleich restringierter Optimierungen . . . . .	89
7.2. Startvariationen der z-Koordinaten . . . . .	93
7.3. Ergebnisvergleich unterschiedlicher KP-Anzahlen . . . . .	95
7.4. Ergebnisse variabler Kontrollpunkte . . . . .	97
7.5. Ergebnisse manueller Positionierung . . . . .	98
7.6. Ergebnisse Kombinationen . . . . .	99
8.1. Top 6 Optimierungsergebnisse . . . . .	102
8.2. Vergleich mit Ergebnissen von <b>Honda</b> . . . . .	104

# Abbildungsverzeichnis

3.1.	Ablauf einer Strukturoptimierung [ <b>Schumacher 2013</b> ] . . . . .	11
3.2.	Schematische Darstellung einer Bézier-Fläche [ <b>Zhang 2006</b> ] . . . . .	27
4.1.	Darstellung der Lochplatte . . . . .	30
4.2.	Entstehung der Zellwinkel [ <b>Honda</b> ] . . . . .	31
4.3.	Schematische Darstellung der Struktur [ <b>Honda</b> ] . . . . .	35
4.4.	Darstellung der Geometrie . . . . .	36
4.5.	Darstellung der FE-Vernetzung . . . . .	37
4.6.	Darstellung der FE-Zellen im Lochbereich . . . . .	38
5.1.	Schematische Darstellung der Winkelbestimmung . . . . .	44
5.2.	Format der Höhenschnittlinien-Koordinaten [ <b>MathWorks 2016</b> ] . . . . .	51
6.1.	Schematische Darstellung der variablen Kontrollpunkte . . . . .	59
6.2.	Flussdiagramm Optimierungsablauf . . . . .	64
6.3.	Datenfluss des Optimierungsprozesses . . . . .	65
6.4.	Flussdiagramm Erstellung Konfigurationsdatei . . . . .	66
6.5.	Flussdiagramm Erstellung bdf-Datei . . . . .	67
6.6.	Darstellung der grafischen Oberfläche . . . . .	78
7.1.	Faserwinkel im Lochbereich bei unrestringierter Optimierung . . . . .	85
7.2.	Höhenschnittlinien bei unrestringierter Optimierung . . . . .	86
7.3.	Bézier-Fläche bei unrestringierter Optimierung . . . . .	87
7.4.	Faserwinkel im hinteren Bereich bei unrestringierter Optimierung . . . . .	87
7.5.	Spannungsverläufe in Faserrichtung . . . . .	88
7.6.	Spannungsverläufe quer zur Faserrichtung . . . . .	88
7.7.	Faserwinkel im Lochbereich bei restringierter Optimierung . . . . .	90
7.8.	Ergebnisse der restringierten Optimierungen . . . . .	92
7.9.	Bézier-Fläche bei zufälligen Startwerten . . . . .	94
7.10.	Optimierte Winkel im Lochbereich bei 6x6 Kontrollpunkten . . . . .	96
8.1.	Optimierte Winkel im Lochbereich bei 6x6 Kontrollpunkten . . . . .	105
8.2.	Optimierte Winkel im Lochbereich bei 6x6 Kontrollpunkten . . . . .	106

# 1. Einleitung

Um einen Einstieg in diese Arbeit zu finden, wird in diesem Kapitel die Motivation und das Ziel ebendieser erläutert. Zudem wird im Kapitel *Aufbau der Arbeit* eine Zusammenfassung der Inhalte der einzelnen Kapitel gegeben.

## 1.1. Motivation der Arbeit

Das Thema *Leichtbau* ist heutzutage in vielen technischen Branchen von großer Bedeutung. Häufig geht damit der Einsatz von Faserverbundwerkstoffen (FVW) einher, da diese bezogen auf ihre Masse überdurchschnittlich gute mechanische Eigenschaften besitzen. Im Flugzeugbau werden unterschiedliche Faserverbundstrukturen daher bereits seit mehr als drei Jahrzehnten eingesetzt. Dennoch ist die Auslegung, die Berechnung und die Fertigung noch nicht in der Lage, die Potentiale dieser Werkstoffe in hohem Maße zu nutzen. Dieses Phänomen ist nicht flugzeugbauspezifisch, sondern ein allgemeines Problem.

Ein Ansatzpunkt, dies zu ändern, ist die gezielte Ausnutzung der anisotropen Eigenschaften der Faserverbundwerkstoffe. Viele Beispiele aus der Natur belegen, dass speziell lokal veränderte Faserverläufe, bspw. beim Baumwachstum im Bereich von Verästelungen, die Belastbarkeit stark erhöhen. Dieser Ansatz widerspricht jedoch dem aus konservativen Gründen weitverbreiteten Vorgehen, Lagen mit geradlinigen Faserverläufen so zu schichten, dass ein möglichst isotroper Verbund entsteht. Auf diese Weise wird viel Masse in die Struktur eingebracht, die keinen funktionalen Zweck erfüllt. Damit wird der Vorteil der FVW gegenüber den Metallen, nur präzise die Menge an Material einbringen zu müssen, die die Belastungsart und -größe tatsächlich erfordert, vergeben. Durch eine gezieltere Nutzung der Eigenschaften können entweder bei unverminderter Funktionalität die Masse reduziert oder bei gleichbleibender Masse die mechanische Belastbarkeit verbessert werden. Um diese Vorteile zu erreichen und so die Potentiale der FVW möglichst gut zu nutzen, müssen Faserverläufe so definiert werden, dass die mechanische Belastbarkeit der Gesamtstruktur exakt an die Belastung angepasst ist.

Um diese Anpassung zu ermöglichen, sind *krummlinige Faserverläufe* notwendig. Die Idee der krummlinigen Fasern ist nicht neu, jedoch rückt sie in den letzten Jahren durch Fortschritte im Bereich der Herstellungsverfahren, die die industrielle Fertigung solcher Strukturen ermöglichen, mehr in den Fokus. Durch die unzähligen Kombinations- und Entwurfsmöglichkeiten ist eine händische Auslegung wirtschaftlich jedoch nicht realisierbar. Um einen effizienten und wirtschaftlichen Einsatz krummliniger Faserverläufe zu ermöglichen, wurden in den letzten Jahren bereits vermehrt Untersuchungen zu numerischen Berechnungsmethoden und automatisierten Optimierungsmethoden durchgeführt. Da in diesem Bereich aber weiterhin Entwicklungsbedarf besteht, befasst sich diese Arbeit mit ebendiesem Thema.

Zusammenfassend besteht die Motivation der Arbeit darin

- die Berechnung von FE-Modellen mit krummlinigen Faserverläufen zu ermöglichen,
- ein Programm zur Optimierung der Auslegung dieser Modelle zu entwickeln sowie
- Erkenntnisse über die Potentiale krummliniger Faserverläufe zu sammeln.

### 1.2. Ziel der Arbeit

Die Arbeit verfolgt zwei Hauptziele. Zum einen soll eine Optimierungsumgebung geschaffen werden, die eine Auslegung von Faserverbundstrukturen mit optimal verlaufenden, krummlinigen Fasern ermöglicht. Das zweite Ziel ist die Optimierung der mechanischen Eigenschaften einer Struktur mithilfe des erzeugten Optimierungsprogramms. Dies dient dem Nachweis der Funktionalität des Programms und seiner Potentiale.

Die erste Zielsetzung erfordert ein Modul, welches auf einfache Weise viele zweidimensionale Kurven beschreiben kann. Hierbei ist vorgegeben, dass die zweidimensionalen Linien aus der Form einer Bézier-Fläche resultieren sollen. Die Steigungsinformationen dieser Verläufe bestimmen die Orientierung der einzelnen Faserverläufe. Das entwickelte Modul soll in ein bestehendes Framework implementiert werden, um so ein Optimierungsprogramm zu bilden, welches automatisch Faserverläufe in Faserverbundstrukturen optimieren kann. Die Optimierungsumgebung soll variabel anpassbar an unterschiedliche Eingangsparameter und Optimierungsziele sein, weshalb ein modularer Aufbau zur Umsetzung der verschiedenen Funktionen zu erzeugen ist. Um die Programmnutzung verständlich zu machen und Modifikationen zu ermöglichen, ist eine ausführliche Dokumentation der programmiertechnischen

Umsetzung essentieller Teil der Arbeit.

Das zweite Ziel sieht die aktive Anwendung des entwickelten Programms zur Optimierung einer Faserverbundstruktur vor. Die Intention der Optimierung ist einerseits eine optimale Auslegung zu bestimmen, andererseits einen idealen Kompromiss aus mechanischen Kennwerten und der Herstellbarkeit zu bestimmen. Um zu validieren, dass das Tool verlässliche Optimierungsergebnisse erzeugt, sollen diese mit Referenzwerten verglichen werden. Da im Rahmen dieser Arbeit nicht die Möglichkeit der realen Verifizierung anhand eines physikalischen Bauteils besteht, werden hierzu Werte aus einer bereits durchgeführten Untersuchung herangezogen, die das gleiche Ziel mit einem anderen Ansatz verfolgte. Um die Vergleichbarkeit zu gewährleisten, werden die Geometriemaße und alle weiteren relevanten Parameter übernommen. Mit der Intention, die tatsächlich bestmögliche Auslegung zu finden, soll zusätzlich eine Parameterstudie bezüglich der Anzahl und Anordnung der Kontrollpunkte der Bézier-Fläche erfolgen.

### 1.3. Aufbau der Arbeit

Um einen aktuellen Stand des zu behandelnden Themas zu bieten, folgt nach der Einleitung eine Literaturrecherche zu bereits durchgeführten Untersuchungen. Daraufhin werden die zum Verständnis dieser Arbeit erforderlichen Grundlagen der verschiedenen Disziplinen zusammengefasst. Anschließend wird die Erstellung des zu optimierenden FE-Modells beschrieben. Die zwei folgenden Kapitel befassen sich mit den erforderlichen Optimierungsmodulen und ihrer programmiertechnischen Umsetzungen. Hierbei befasst sich Kapitel 5 mit dem Bézierflächen-Modul und Kapitel 6 mit dem Optimierungstool, das alle Komponenten zusammenführt. Nach Abschluss der Programmierung beginnt in Kapitel 7 die aktive Untersuchung. Es werden diverse Optimierungsdurchläufe mit unterschiedlichen Startparametern durchgeführt und dargestellt. Abschließend werden die Ergebnisse in Kapitel 8 bewertet und verglichen sowie diverse Verbesserungsmöglichkeiten diskutiert. Zudem wird ein Fazit gezogen und ein Ausblick gegeben.

## 2. Literaturrecherche

In Kapitel 1.1 wurde erwähnt, dass in den vergangenen Jahren die Untersuchungen zum Thema der krummlinigen Faserverläufe zugenommen haben. Um einen aktuellen Stand der Forschung zu vermitteln, wird in diesem Kapitel eine Auswahl an bereits veröffentlichten Arbeiten in chronologischer Reihenfolge vorgestellt. Hierbei wird jeweils das Ziel der jeweiligen Arbeit, die Umsetzung sowie die Ergebnisse zusammengefasst. In Unterkapitel 2.1 wird besonders auf die Arbeit von **Honda** eingegangen, da sie als Referenzuntersuchung für diese Arbeit dient.

### 2.1. Honda-Untersuchung

*Multi-Objective Optimization of Curvilinear Fiber Shapes for Laminated Composite Plates by Using NSGA-II* von Shinya **Honda**, Teruki Igarashi und Yoshihiro Narita (Hokkaido University, Japan, 2013)

Die Arbeit verfolgt den Ansatz einer Mehrziel-Optimierung, um mit krummlinig verlaufenden Faserverläufen eine Struktur hinsichtlich ihrer mechanischen Belastbarkeit und der Prozessierbarkeit zu optimieren. Es finden zwei getrennte Untersuchungen zur Optimierung unterschiedlicher mechanischer Eigenschaften statt. Zum einen ist die Zielfunktion die Reduzierung des Tsai-Wu-Versagensindex, welcher sich aus den Spannungen im Bereich des Loches in einer Platte ergibt. Im zweiten Fall ist das Ziel die Erhöhung der Eigenfrequenz einer ebenen Platte. Als jeweils zweite Zielfunktion wird die Minimierung der durchschnittlichen Krümmung als Index für die Prozessierbarkeit definiert. Da die Ziele der beiden jeweils aktiven Funktionen kollidieren, kann keine einzeln ihr Optimum erreichen, weswegen lediglich optimale Kompromisse aus beiden Zielen gefunden werden.

Zur Definition der Faserverläufe wird der Ansatz einer kubischen Polynomfunktion gewählt, welche eine dreidimensionale Fläche aufspannt. Mittels Projektion von Höhenschnittlinien werden zweidimensionale Kurven definiert, aus denen für jede Finite Element ein konstanter Winkel bestimmt wird. Die Designvariablen sind dabei die Koeffizienten der Polynomfunktion. Es werden jeweils Platten mit einem symmetrischen Laminataufbau aus acht Lagen und

alternierenden + und - Orientierungen betrachtet. Die Winkel der + Lage ergeben sich aus den Höhengschnittlinien und die der - Lage sind bezogen auf die x-Achse zu denen der + Lage gespiegelt. Zur Bestimmung eines *PARETO-optimalen Randes* wird ein generischer Mehrziel-Algorithmus eingesetzt. Der verwendete NSGA-II (non-dominated sorting genetic algorithm) ist gut geeignet zur Suche von weitläufigen PARETO-Rändern bei divergierenden Zielfunktionen. Seine Funktionsweise basiert grundsätzlich auf Bewertung und Sortierung der Ergebnisse unterschiedlicher Entwürfe. In jedem Durchlauf werden schlechte Kombinationen von Entwurfsvariablen entfernt und mittels sogenannter Mutation neue eingebracht. Am Ende der Optimierung stellen alle verbleibenden Entwürfe den PARETO-optimalen Rand dar.

Anhand der Untersuchungen kann nachgewiesen werden, dass mittels krummliniger Fasern sowohl höhere Eigenfrequenzen als auch eine Verbesserung der Festigkeit gegenüber konventionellen Faserverläufen erreicht werden kann. Die Berechnungen ergeben einen weitläufigen PARETO-Rand, der Entwürfe mit geraden Fasern bis hin zu Auslegungen mit stark gekrümmten Faserverläufen umfasst. Bilder und Tabellen der Ergebnisse sind an den entsprechenden Stellen in Kapitel 8.1 aufgeführt.

## 2.2. Weitere Untersuchungen

*Innovative design of composite structures: The use of curvilinear fiber format in structural design of composites* von R. F. **Charette** und M. W. Hye (University of Maryland, USA, 1990)

Die Arbeit ist eine sehr frühe Untersuchung zu dem Thema und ermöglicht einen Einblick in die Anfänge der Betrachtung von krummlinigen Faserverläufen. Das Ziel der Arbeit ist es, den Einfluss von krummlinigen Fasern in einer belasteten Faserverbundstruktur mit einem mittigen Loch zu untersuchen. Es werden mehrere Lamine mit krumm- und geradlinigen Faserverläufen sowie unterschiedliche Plattengeometrien betrachtet, wobei ein quasi-isotropes Laminat mit geradlinigen Fasern als Referenzdesign gewählt wird. In einem Iterationsprozess werden analytisch die krummlinigen Faserverläufe entlang der Hauptspannungsrichtungen bestimmt. Mittels einer Finite Elemente Analyse (FEA) werden anschließend die Spannungen berechnet und anhand des Festigkeitskriteriums der maximalen Dehnungen und des Tsai-Wu-Kriteriums die Versagenslast und -art ermittelt. Zusätzlich wird das Beulverhalten unter uniaxialem Druck betrachtet. Die Untersuchungen ergeben höhere Lastkapazitäten unter Zugbelastung und vergleichbares Beulverhalten.

*Design of composite layers with Curvilinear Fiber Paths using Cellular Automata* von Shahriar **Setoodeh** und Zafer Gürdal (Virginia Polytechnic Institute and State University, USA, 2003)

Der Fokus dieser Arbeit liegt auf der Anwendung des sog. *Cellular Automata* Methode (Zellulärer Automat) zur Optimierung von krummlinigen Faserverläufen. Mit diesem Ansatz unterscheidet sich die Arbeit von der üblichen FEM-gestützten Herangehensweise. Die Struktur wird hierbei ebenfalls in Zellen eingeteilt. Durch Betrachtung von Zellen und ihren benachbarten Zellen auf lokale Veränderungen erfolgt eine iterative Optimierung. Diese findet hinsichtlich eines Formänderungsenergie-Kriteriums statt. Zur Gewährleistung der Prozessierbarkeit werden Restriktionen bezüglich der Tendenz der Orientierungen benachbarter Zellen eingesetzt. Betrachtet werden sowohl Kragbalken als auch Platten mit einer Aussparung. Bei beiden Strukturen können erhebliche Verbesserungen der Steifigkeiten nachgewiesen werden.

*Optimum design of composite structures with curved fiber courses* von Levend **Parnas**, Süha Oral und Ümit Ceyhan (Middle East Technical University, Türkei, 2003)

Ziel der Arbeit ist die Masseoptimierung einer Faserverbundstruktur unter Berücksichtigung von Spannungsrestriktionen (Tsai-Hill-Kriterium). Hierbei sind sowohl die Faserverläufe als auch die Lagendicken variabel. Die Faserverläufe werden mittels kubischer Bézierkurven dargestellt, während die Lagendicken von bikubischen Bézier-Flächen abgebildet wird. Die Koordinaten der jeweiligen Bézier-Kontrollpunkte sind die Entwurfsvariablen. Diese werden mit Hilfe des SQP-Algorithmus verändert und optimiert. Abschließend wird zusammengefasst, dass das Ziel der Massereduzierung zwar erfüllt werden kann, die Kosten aufgrund der teuren Fertigungsverfahren jedoch steigen.

*Optimization of tow-placed, tailored composite laminates* von Adriana W. **Blom**, Mostafa M. Abdalla und Zafer Gürdal (Delft University of Technology, Niederlande, 2007)

Im Gegensatz zu den meisten anderen Untersuchungen liegt der Fokus dieser Arbeit nicht auf der Optimierung der Faserverläufe hinsichtlich maximaler Belastbarkeit oder Steifigkeit. Da krummlinige Faserverläufe z.B. durch Überlappungen zu lokal variierenden Laminatdicken führen können, entstehen Auswirkungen auf die Fertigung, die mechanischen Eigenschaften sowie die Oberflächenqualität. An dieser Stelle setzt die Arbeit an. Die Ziele sind einerseits die Minimierung der größten Lagendicke sowie die Maximierung der Ebenheit des Laminats. Hierzu wird die Laminatdicke über einen verschmierten Ansatz berechnet. (Die Arbeit wurde 2009 leicht überarbeitet unter dem Titel *Optimization of Course Locations in Fibre-Placed*

*Panels for General Fibre Angle Distributions* nochmals veröffentlicht.)

*A Methodology for Finite Element Analysis of Curvilinear Fiber Laminates with Defects, Fabricated by Automated Fiber Placement Technique* von Suhas **Prabhakar** (McGill University, Kanada, 2011)

Die Arbeit befasst sich damit, dass bei der Herstellung von Laminaten mit krummlinig verlaufenden Fasern Lücken und Überlappungen, sogenannte Defekte, entstehen. Während Faserlücken mit Harz aufgefüllt werden, entstehen bei Überlappungen Erhöhungen des Laminats. Ziel der Arbeit ist, diese Defekte zu analysieren und deren Effekte bereits während der Auslegung zu berücksichtigen. Hierzu wird eine neue Methode, die ein sog. Defekt-Element beinhaltet, entwickelt. Dieser Ansatz soll ermöglichen, dass Modellierungen und Berechnungen von Laminaten mit krummlinigen Faserverläufen und Defekten schneller und realitätsnäher durchgeführt werden können.

*Optimization of Tow Steered Fiber Orientation Using the Level Set Method* von Christopher J. **Brampton** und H. Alicia Kim (University of Bath, UK, 2013)

Mittelpunkt dieser Optimierungsuntersuchung ist der Einsatz der aus dem Bereich der Topologieoptimierung stammenden *Level Set Methode*. Diese bestimmt eine Referenzfunktion, zu der parallel in konstanten Abständen weitere Kurven erstellt werden. Anhand dieser Kurven werden die jeweiligen Faserwinkel bestimmt. Zur Optimierung dieser Funktion respektive der mit ihr verbundenen Faserverläufe dienen lokale Sensitivitätsanalysen, die die Auswirkungen der Faseränderungen hinsichtlich Spannungen und Dehnungen betrachten. Ergebnis der Arbeit ist eine Methode, die Faserverläufe optimieren kann, welche zusätzlich einen kontinuierlichen Verlauf aufweisen. Diese Methode wird abschließend auf einfache Geometrien angewandt und anhand der Ergebnisse belegt, dass die Level Set Methode zur Faseroptimierung erfolgreich genutzt werden kann.

*Postbuckling Optimization of Variable Angle Tow Composite Plates* von Zhangming **Wu**, Paul M. Weaver und Gangadharan Raju (Advanced Composite Centre for Innovation and Science, UK, 2013)

Das Thema der Arbeit ist die Optimierung des Nachbeulverhaltens einer auf Druck belasteten Platte mit Hilfe von krummlinigen Faserverläufen. Das Nachbeulverhalten wird sowohl an der maximalen Querverschiebung sowie an der Stauchung bewertet. Um die einzelnen Faserwinkel anhand weniger Entwurfsvariablen bestimmen zu können, werden Verläufe mit Hilfe der Lagrange-Polynome bestimmt. Zur iterativen Lösungssuche kommt ein generischer

Algorithmus zum Einsatz. Ein Vergleich mit geradlinig verlaufenden Fasern zeigt, dass krummlinige Verläufe das Nachbeulverhalten positiv beeinflussen können.

*Analysis and Optimization of cylindrical Structures manufactured by automated Fiber Placement Technique* von Sedat **Güldü** (Middle East Technical University, Türkei, 2014)

Die durchgeführten Untersuchungen befassen sich mit dem Einsatz von krummlinigen Faserverläufen in zylindrischen Strukturen. Ziel hierbei ist es, den Beulfaktor unter verschiedenen Belastungen zu optimieren. Die Faserorientierung der einzelnen Zellen werden anhand einer Referenzkurve mit zwei Parametern bestimmt. Als Suchalgorithmus wird die sog. Partikelschwarmoptimierung (PSO) genutzt, deren Ablauf sich am Schwarmverhalten der Natur orientiert, und die jeweilige Strukturantwort wird mit Hilfe der FEM berechnet. Die Ergebnisse werden mit denen eines Zylinders konventioneller Auslegung verglichen und es kann eine Erhöhung des Beulfaktors nachgewiesen werden. Zusätzlich werden hinsichtlich der Fertigung die Lagendicken inklusive entstehender Überlappungen und Lücken ermittelt.

*Postbuckling Analysis and Optimization of stiffened Fuselage Panels utilizing Variable-Stiffness Laminates* von Tanut **Ungwattanapanit** und Horst Baier (Technische Universität München, Deutschland, 2014)

Die Untersuchungen befassen sich mit der Optimierung eines Fensterbereichs in einem Flugzeugrumpf mit Hilfe eines aus krummlinigen Faserverläufen erzeugten Laminats. Die Beulstabilität wird anhand der maximal auftretenden Verschiebungen bewertet. Als Optimierungsalgorithmus wird die sog. Global Response Surface Method (GRSM) eingesetzt, welche Vorteile bei der Suche nach dem globalen Optimum bietet. Die Strukturantworten werden unter Berücksichtigung des nicht-linearen geometrischen Verhaltens berechnet. Zur Abbildung der krummlinigen Faserverläufe wird der zu untersuchende Bereich in horizontale Abschnitte eingeteilt, in denen jeweils eine bestimmte Faserorientierung definiert wird. Bezogen auf eine konventionelle Auslegung kann ein signifikant verbessertes Nachbeulverhalten erreicht werden. Nachträgliche Auswertungen ergeben zudem eine Verbesserung des Tsai-Wu-Kriteriums. Abschließend sprechen die Autoren krummlinig verlaufenden Fasern große Potentiale zur Massereduzierung zu.

*Optimization of composite structures with curved fiber trajectories* von Etienne **Lemaire**, Semih Zein und Michael Bruyneel (SAMTECH s.a., Belgien, 2015)

Hauptaspekt der Arbeit ist ein neuartiger Ansatz zur Erstellung und Optimierung von nicht-linearen Faserverläufen, der auch die Arbeit mit allgemeinen, nicht ebenen Flächen

ermöglichen soll. Zudem erzeugt die Methode kontinuierliche Faserverläufe, wodurch Lamine ohne Überlappungen und Lücken entstehen. Dieser Ansatz wird beispielhaft zur Steifigkeitsoptimierung verschiedener Strukturen eingesetzt. Hierbei kommen verschiedene Algorithmen zum Einsatz. Die Faserverläufe werden mittels der Eikonal-Gleichung anhand einer Referenzkurve erzeugt. Deren Form und der jeweilige Abstand zur Referenzkurve sind hierbei die Entwurfsvariablen. Vorteilhaft ist die geringe Anzahl an Entwurfsvariablen, nachteilhaft die beschränkte Variabilität der Faserverläufe. Hinsichtlich der Herstellbarkeit wird ein Augenmerk auf die Begrenzung der Faserkrümmung gelegt.

Die Literaturrecherche hat gezeigt, dass das Thema der krummlinigen Faserverläufe auf viele unterschiedliche Weisen untersucht werden kann. Speziell in den Bereichen der Abbildung der Faserverläufe und der Optimierungsalgorithmen gibt es diverse Ansätze. Zudem unterscheiden sich bei den meisten Arbeiten die betrachteten Strukturen, was sich allerdings aus dem breiten Anwendungsgebiet der Faserverbundwerkstoffe ergibt.

## 3. Theoretische Grundlagen

Das Ziel der Arbeit wurde in Kapitel 1.2 vorgegeben. Um dieses nun in die Praxis umzusetzen, werden Wissen und Fähigkeiten in verschiedenen Disziplinen benötigt. Die Grundlagen ebendieser sollen in diesem Kapitel erläutert werden. Da die Arbeit drei große Themengebiete verbindet, wird sich darauf beschränkt, nur für die Arbeit tatsächlich relevante Grundlagen darzustellen. Für weiteres Hintergrundwissen wird die zur Erstellung dieses Kapitels genutzte Literatur empfohlen.

Bevor auf die notwendigen spezifischen Grundlagen eingegangen wird, bietet das nachfolgende Unterkapitel vorab einen zusammenfassenden Überblick über die geplante Optimierungsprozedur. Dies dient zu verstehen, wie die einzelnen Teildisziplinen genutzt werden und ineinandergreifen.

### 3.1. Optimierungsprozedur

Es gilt eine Struktur zu optimieren, indem der Faserverlauf der verwendeten Faserverbundwerkstoffe ideal ausgelegt wird. Essenziell hierfür sind Kenntnisse von Optimierungsabläufen, speziell der Strukturoptimierung (s. Kapitel 3.2). Zur idealen Auslegung von Faserverläufen ist Wissen zu den Eigenschaften, Versagenskriterien und Herstellungsverfahren von Faserverbundwerkstoffen ebenfalls wichtig (s. Kapitel 3.3). Um die Belastung der Struktur zu berechnen, die entscheidend für die Optimierung ist, hat sich die computergestützte Berechnung mittels der Finite-Elemente-Methode etabliert (s. Kapitel 3.4). Zur FE-Berechnung werden des Weiteren Informationen für den Faserverlauf benötigt. Für die Bereitstellung dieser und weiterer Daten wird sich verschiedener mathematischer Funktionen bedient (s. Kapitel 3.5).

### 3.2. Strukturoptimierung

Das Themengebiet der Strukturoptimierung beinhaltet diverse unterschiedliche Ansätze, Vorgehensweisen und Rechenmethoden. Wie eingangs erwähnt soll daher an dieser Stelle lediglich ein Überblick über die für diese Arbeit relevanten Bereiche der Strukturoptimierung gegeben

werden. Einen detaillierteren Einstieg in die Materie bieten **Harzheim 2014** und **Schumacher 2013**, die auch zur Erstellung der nachfolgenden Grundlagen dienen.

#### 3.2.1. Allgemeines

Ziel der Strukturoptimierung (STO) ist es, den grundsätzlichen Wunsch einer optimalen Lösung in eine mathematische Form zu bringen und auf diese Weise berechenbar zu machen. Wichtig hierbei ist, festzulegen, was als *optimal* angesehen wird. Häufigstes Ziel der STO ist es bspw. eine Struktur so auszulegen, dass die Funktionalität bei geringstmöglicher Masse erhalten bleibt. Hier unterscheidet sich die STO von der lange Zeit und teilweise auch heute noch üblichen Herangehensweise, eine neue Struktur rein nach Erfahrungswerten nur mit dem Ziel der Funktionserfüllung zu entwerfen.

Der grundsätzliche Ablauf eines Entwicklungsprozesses mit Optimierungsprozedur ist in Bild 3.1 dargestellt. Während eine konventionelle Strukturauslegung bereits mit der Auswertung beendet ist, schließt sich dieser bei einer Optimierung der *Optimierungsalgorithmus* an. Hierbei werden die Ergebnisse der Auswertung, die sog. *Strukturantwort*, geprüft und die Entwurfsvariablen (siehe Kapitel 3.2.2) solange verändert, bis die Auswertung ein Optimum ergibt. Die Strukturantwort kann bei einfachen Problemen ggf. noch analytisch berechnet werden, in den meisten Fällen wird jedoch ein numerisches Verfahren wie die FEM benötigt.

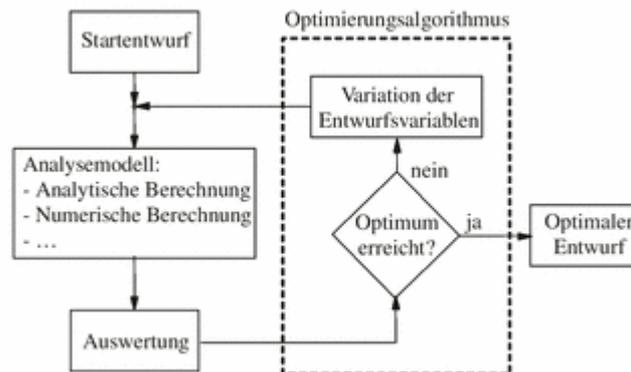


Abbildung 3.1.: Ablauf einer Strukturoptimierung [**Schumacher 2013**]

Das Prinzip einer (Struktur-)Optimierung ist, das Problem zunächst in die Form einer mathematischen Funktion zu bringen, diese abzuleiten und gleich null zu setzen. Auf

diese Weise wird ein Extrempunkt der Funktion gefunden, was gleichbedeutend mit einem Optimum der Strukturauslegung ist. Für eine Optimierung ist es daher notwendig, ein reales Problem mit all seinen Facetten auf eine mathematische Funktion zu reduzieren. Hierbei ist es wichtig, alle wichtigen Informationen des Problems einzubeziehen, damit am Ende ein Optimum entsteht, das realisierbar ist und alle Anforderungen erfüllt. Daher ist bereits die Definition des Optimierungsproblems ein essenzieller Bestandteil einer Strukturoptimierung. Zur mathematischen Definition der Probleminformationen dienen Zielfunktionen und Restriktionen (siehe Kapitel 3.2.3 und Kapitel 3.2.4).

Die Strukturoptimierung wird grundsätzlich in die drei unterschiedlichen Arten Dimensionierung, Formoptimierung und Topologieoptimierung eingeteilt. Während es bei der Formoptimierung um die Veränderung der äußeren Form geht, behandelt die Topologieoptimierung die Materialverteilung innerhalb der vorgegebenen Form, beispielsweise durch Einbringung von Löchern oder lokales Ab-/Auftragen von Material. Bei der Dimensionierung hingegen werden Geometrieparameter an einer ansonsten fixen Form variiert. Hierzu können Strukturdicken, Winkel, Radien, Durchmesser und diverse andere Parameter zählen. Des Weiteren gibt es noch zwei zusätzliche Optimierungsansätze, die jedoch deutlich weniger Raum für Variationen bieten. Dies ist zum einen die Wahl der Bauweise (Fachwerk, Vollwandträger, etc.), zum anderen die Wahl des verwendeten Materials (Stahl, Aluminium, FVW, etc.). Im Falle der FVW ist jedoch eine umfangreichere Materialoptimierung möglich, da diese Werkstoffe aufgrund ihrer Zusammensetzung diverse veränderbare Parameter (z.B. Anzahl Faserlagen, Faserorientierung) besitzen.

#### 3.2.2. Entwurfsvariablen

Grundlage jeder Optimierung sind die Entwurfsvariablen, denn sie unterscheiden einen optimierbaren von einem festen Entwurf. Es kann sich bei ihnen um diverse Faktoren wie z.B. Längen, Radien oder Winkel handeln. Durch Ändern der Variablen ist es dem Optimierungsalgorithmus möglich, Einfluss auf die Struktur zu nehmen. Anhand der Strukturantwort vergleicht der Algorithmus anschließend die Änderung des Ergebnisses und stellt so die Entwurfsvariablen iterativ optimal ein ( $x_i = x_i^*$ ).

Kommen unterschiedlichen Arten von Entwurfsvariablen zum Einsatz, z.B. Profildicken und Winkel, sollte stets auf eine ausgeglichene Skalierung geachtet werden. Wenn die Einflüsse der einzelnen Variablenarten auf das Ergebnis sich stark voneinander unterscheiden, kann es andernfalls zu einem Sensitivitätsproblem kommen. Dadurch benötigt der Algorithmus

mehr Iterationen zur Optimierung aller Variablen und im schlimmsten Fall führt es zur Nicht-Berücksichtigung einer Variablenart. Gleiches gilt für den Zusammenhang der Entwurfsvariablen mit den fix definierten Parametern und den Ausgabeparametern. Führen z.B. die Änderungen der Variablen bei den vorhandenen Belastungen nur zu minimalen Ergebnisvariationen, arbeitet der Algorithmus nicht effizient oder gar fehlerhaft.

Wenn sehr viele Entwurfsvariablen vorliegen, beispielsweise wenn die Anzahl proportional zu der Zellanzahl einer FE-vernetzten Struktur ist, sind hohe Rechenkapazitäten erforderlich, um das Optimierungsproblem zu berechnen. Bei begrenzten Kapazitäten kann daher die Verwendung sog. *Ersatzvariablen* sinnvoll sein. Hierbei werden die tatsächlichen Entwurfsvariablen mittels eines mathematischen Ansatzes in Abhängigkeit von einer ansatzabhängigen Anzahl Ersatzvariablen bestimmt. Auf diese Weise ist es möglich, die Zahl der Variablen im Optimierungsprozess signifikant zu reduzieren und falls eine Abhängigkeit von der FE-Vernetzung besteht, diese zu lösen.

#### 3.2.3. Zielfunktion

Essenzieller Bestandteil einer jeden Optimierung ist die Zielfunktion. Sie stellt eine mathematische Formulierung des Konstruktionsziels dar. Übliche Ziele sind die Reduzierung der Masse, der maximalen Spannungen oder der maximalen Verformungen. In der Regel werden Zielfunktionen so formuliert, dass ein Funktionsminimum gesucht wird, welches die optimale Lösung des Problems darstellt. Mit  $\max f(x) = \min -f(x)$  können auch Maximierungsaufgaben leicht umgeformt werden. Um den Umgang mit der Zielfunktion und den Restriktionen (s. Kapitel 3.2.4) mathematisch zu vereinfachen, können alle Elemente mit der sog. *Lagrange-Funktion* zusammengefügt werden (s. Formel 3.1). Werden auf diese Weise Ungleichheitsbedingungen eingebunden, wird diese auch *Karush-Kuhn-Tucker-Funktion* (KKT) genannt.  $\alpha$  und  $\beta$  sind die sog. Lagrange- oder KKT-Multiplikatoren.

$$L(x, \alpha, \beta) = f(x) + \sum_{i=1}^q \alpha_i h_i(x) + \sum_{k=1}^p \beta_k g_k(x) \quad (3.1)$$

Die notwendige Bedingung für ein Funktionsminimum lautet, dass der Vektor der partiellen Ableitungen der Zielfunktion nach den Entwurfsvariablen gleich dem Nullvektor ist (s. Formel 3.2). Eine hinreichende Bedingung ist, dass die Hesse-Matrix positiv definit ist, das heißt, alle Eigenwerte positiv sind. Die Hesse-Matrix beinhaltet die zweiten partiellen Ableitungen der Zielfunktion (s. Formel 3.3). Alternativ stellen die aus der *Lagrange-Funktion* hergelei-

teten *Kuhn-Tucker*-Kriterien notwendige Bedingungen für die Existenz eines Optimums bei restringierten Problemen dar.

$$\begin{pmatrix} \frac{\partial f}{\partial x_1}(x) \\ \frac{\partial f}{\partial x_2}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{pmatrix}_{x^*} = \mathbf{0} \quad (3.2)$$

$$\mathbf{H}(x^*) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \frac{\partial^2 f}{\partial x_2 \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(x) & \frac{\partial^2 f}{\partial x_n \partial x_2}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n}(x) \end{pmatrix}_{x^*} \quad (3.3)$$

Liegen zwei oder mehr Zielfunktionen vor, wird von einer Mehrzieloptimierung gesprochen. Zur Lösung einer solchen Mehrzieloptimierung bedarf es spezieller Methoden, da in der Regel ein Zielkonflikt vorliegt. Zielkonflikte zeichnen sich dadurch aus, dass die Optima der einzelnen Zielfunktionen nicht erreicht werden können. Dadurch ergeben sich diverse Entwürfe, die als *PARETO-optimal* oder *funktionaleffizient* bezeichnet werden. Diese Kompromisse aus beiden Zielfunktionen können in ein Diagramm eingetragen und auf diese Weise ein sog. PARETO-Rand erzeugt werden. Um diese Mehrzieloptimierungsprobleme (PARETO-Optimierungen) zu lösen, gibt es unterschiedliche Verfahren. In dieser Arbeit wird die Methode der *restriktionsorientierten Transformation* verwendet. Bei dieser wird eine Zielfunktion als Hauptziel deklariert, während die restlichen Zielfunktionen als Nebenziele zu Restriktionen umformuliert werden. Wenn diese Restriktionen aktiv sind (vgl. Kapitel 3.2.4), haben die entsprechenden Nebenziele ein Minimum. Durch Wahl unterschiedlicher Werte für die Nebenziel-Restriktionen und wiederholten Optimierungsdurchläufen kann so ein PARETO-Rand erzeugt werden. Laut **Schumacher 2013** ist diese Vorgehensweise zwar zeitaufwändig, jedoch die beste Vorgehensweise zur Lösung von Mehrzieloptimierungen.

### 3.2.4. Restriktionen

Wie in Kapitel 3.2.1 angedeutet, müssen alle einzuhaltenden Bedingungen bzw. Forderungen an die Konstruktion in die Optimierung einbezogen werden. Hierzu stehen *Gleichheits-* und *Ungleichheitsbedingungen* zur Verfügung, deren mathematische Formulierungen in Tabelle 3.1 aufgeführt sind. Mit ihnen können verschiedenste Arten von Restriktionen mathematisch beschrieben werden. Zusätzlich können Entwurfsvariablen mittels *expliziter Restriktionen* auf

einen Wertebereich, in dem der Algorithmus sie variieren kann, begrenzt werden. Ist durch die Problemstellung kein expliziter Bereich vorgegeben, sollte der Entwurfsraum dennoch sinnvoll eingeschränkt werden, um die Optimierung schneller und zielführender zu gestalten.

Tabelle 3.1.: Arten von Restriktionen

Art	mathematische Formulierung
Gleichheitsrestriktionen	$h_k(x) = 0$
Ungleichheitsrestriktionen	$g_j(x) \leq 0$
explizite Restriktionen	$x_i^{min} < x_i < x_i^{max}$

Eine Restriktion gilt als aktiv, wenn die Entwurfsvariablen bei der optimalen Lösung die Restriktionsgleichung exakt erfüllen. Das heißt, der optimale Entwurf ist durch die Restriktion beschränkt. In der Regel könnte die Zielfunktion also ohne die Restriktion weiter verbessert werden. Speziell bei Masseoptimierungen müssen jedoch Restriktionen vorliegen, da die Masse sonst zu null wird. In Tabelle 3.2 sind die möglichen Fälle, die eine Ungleichheitsrestriktion einnehmen kann, aufgeführt. Gleichheitsrestriktionen sind in der Praxis selten, da es nur in den wenigsten Fällen Sinn macht, einen exakten Zielwert zu fordern. Werden alle Restriktionen eingehalten, befindet sich der Entwurf im *zulässigen Bereich*.

Tabelle 3.2.: Restriktionsfälle

$g_j(x) < 0$	Restriktion erfüllt, aber nicht aktiv
$g_j(x) = 0$	Restriktion erfüllt und aktiv
$g_j(x) > 0$	Restriktion verletzt

Restriktionen können unterschiedlichste Gründe haben, beispielsweise maximale Belastungen zur Verhinderung mechanischen Versagens, maximale Verformungen zur Erhaltung der Kompatibilität zu anderen Systemelementen oder aus fertigungsbedingten Gründen mini- oder maximale geometrische Eigenschaften.

### 3.2.5. Algorithmen

Theoretisch ist es möglich, die Systemantworten manuell auszuwerten und die Entwurfsvariablen beispielsweise anhand der Ergebnisse nach Erfahrungswerten zu verändern. In der Praxis ist dieses Vorgehen jedoch nicht praktikabel, weswegen zur automatischen Durchführung dieses Prozesses mathematische Algorithmen eingesetzt werden. Durch die Algorithmen wird

die Optimierung schneller, wirtschaftlicher und effektiver. Allerdings haben Algorithmen unterschiedliche Vor- und Nachteile, sodass kein universell empfehlenswerter Algorithmus existiert. Um die Optimierung zu beschleunigen oder bessere Optima zu finden, können Algorithmen jedoch kombiniert werden. In Tabelle 3.3 ist der Ablauf eines Optimierungsalgorithmus beschrieben. Mittels der Abbruchkriterien wird vorgegeben, unter welchen Voraussetzungen die Optimierung beendet werden soll.

Tabelle 3.3.: Ablauf Optimierungsalgorithmus [Schumacher 2013]

1. Festlegung des Startentwurfs  $x^{(k)}$  mit  $k = 0$
2. Änderung des Entwurfs nach einem bestimmten Kriterium  $x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$
3. Überprüfung der Abbruchkriterien (z.B. Kuhn-Tucker-Bedingungen), wenn nicht erfüllt, gehe zu 2 mit  $k = k+1$
4. Optimale Lösung  $x^* = x^{(k+1)}$

Die Sequentielle Quadratische Programmierung (SQP) ist ein approximationsbasierter Algorithmus, der die Zielfunktion quadratisch und die Restriktionen linear annähert. Er ist robust und daher vielfältig einsetzbar. Bei nicht-konvexen Problemen ist die Wahrscheinlichkeit, dass der Algorithmus aus einem lokalen Minimum nicht wieder herausfindet, jedoch groß. Dem kann mittels Durchführung mehrerer Optimierungsabläufe mit jeweils unterschiedlichen Startvariablen entgegengewirkt werden. Alternativ können *generische* Algorithmen, auch *stochastische* Verfahren genannt, verwendet werden, um das Problem der lokalen Minima zu verkleinern. Im Gegensatz zu den gradientenbasierten Algorithmen beinhalten diese meist eine Zufallskomponente und haben eine erhöhte Wahrscheinlichkeit, das globale Optimum zu erreichen. Allerdings brauchen generische Algorithmen meist dementsprechend viele Iterationen, um ein Ergebnis zu finden. Als Vertreter der generischen Algorithmen seien *Evolutionalgorithmen* genannt.

#### 3.2.6. Konvexität

Speziell bei komplexen Problemstellungen liegen häufig hochgradig nicht-lineare Funktionen vor, wodurch die Wahrscheinlichkeit groß ist, dass mehrere Extremstellen (Optima) existieren. Ein *konvexes* Problem zeichnet sich dadurch aus, dass nur ein lokales Minimum vorliegt und dieses somit auch das globale Minimum ist. Wenn der gesamte Kurvenverlauf bekannt ist, kann mit einer mathematischen Bedingung die Konvexität geprüft werden. Diese Bedingung ist zwar hinreichend, aber nicht notwendig, weswegen auch nicht-konvexe Funktionen nur ein

Minimum aufweisen können. Für ein konvexes Problem müssen sowohl die Zielfunktion als auch die Restriktionsfunktionen konvex sein. In der Praxis kann bei Problemstellungen jedoch zumeist keine Aussage über die Konvexität getroffen werden. Daher muss von nicht-konvexen Problemen mit mehreren Extremstellen ausgegangen werden und dieser Umstand in der Ergebnisbetrachtung berücksichtigt werden. Um diese Unsicherheit zu minimieren, ist es wichtig, ein Optimierungsprozess mehrfach mit jeweils unterschiedlichen Startwerten durchzuführen. Stellt sich hierbei häufig ein identisches Minimum heraus, ist die Wahrscheinlichkeit groß, dass es sich dabei um das globale Minimum handelt.

#### 3.2.7. Sensitivitätsanalyse

Bei der Verwendung eines gradientenbasierten Algorithmus ist die Sensitivitätsanalyse hinsichtlich des Zeitaufwandes einer Optimierung von großer Bedeutung, da nur mit passenden Gradienten eine effektive und schnelle Optimierung möglich ist. Dafür prüft eine Sensitivitätsanalyse den Einfluss der einzelnen Entwurfsvariablen auf die Strukturantwort. Auf diese Weise kann vorab getestet werden, welchen und wie großen Einfluss die einzelnen Variablen auf das Ergebnis haben. Da es verschiedene Ansätze zur Ermittlung der Sensitivitäten gibt, ist je nach Problem ein sinnvolles Verfahren zu wählen. Methoden für lokale oder globale Sensitivitätsanalysen sind z. B. die Bildung finiter Differenzen, analytische und semi-analytische Sensitivitätsanalysen oder Varianzanalysen. Da bei lokalen Sensitivitäten nicht gewährleistet ist, dass sie das globale Verhalten wiedergeben, sind sie nur bei annähernd glatten Funktionen aussagekräftig und daher grundsätzlich nicht empfehlenswert.

#### 3.2.8. Optimierungs-Programme

Es existieren speziell entwickelte Optimierungsprogramme zur Lösung von Optimierungsproblemen und auch viele FE-Programme bieten mittlerweile integrierte Optimierungsfunktionen. Eine dritte Alternative stellen Programme zur Behandlung allgemeiner mathematischer Probleme, z.B. Matlab, dar. Matlab ist hinsichtlich seiner Fähigkeiten speziellen Optimierungsprogrammen zwar unterlegen, bietet jedoch ein wesentlich offeneres System, wodurch die Optimierung variabler und freier gestaltbar ist. Es können problemlos kleine Zusatztools in den Optimierungsprozess integriert und externe Daten gelesen und geschrieben werden. Zudem besteht die Möglichkeit, weitere Programme, wie bspw. externe FE-Solver, in den Optimierungsprozess einzubinden. Aus diesen Gründen wird zur Programmierung des Optimierungs-Tools Matlab eingesetzt.

### 3.2.9. Universal Framework for Optimization

Das *Universal Framework for Optimization* (UFO) ist mit dem Ziel programmiert, eine einfache und günstige Alternative zu teuren Optimierungsprogrammen bereitzustellen. Dies geschieht durch Einbindung der Optimierungsfunktionen Matlabs, sodass eine einfache Handhabung für unterschiedliche Probleme ermöglicht wird. Um das Framework zu nutzen, werden sämtliche Informationen des zu untersuchenden Optimierungsproblems in die Konfigurationsdatei *config.m* geschrieben. Das config-File ist in zwei Abschnitte unterteilt. Im Konfigurationsabschnitt werden alle Daten für den Optimierungsprozess, bspw. die Zielfunktion, die Restriktionen, die Entwurfsvariablen und der Algorithmus, definiert. Im darauffolgenden Berechnungsabschnitt werden die einzelnen Berechnungsschritte, z.B. der Aufruf externer Programme, festgelegt. Durch Ausführung der Datei *Run.m* mit Übergabe der Konfigurationsdatei wird die Optimierungsprozedur gestartet. Für weitere Informationen zu dem Framework wird die zugehörige Dokumentation **Geisberger 2013** empfohlen, aus der auch das hier aufgeführte Wissen entnommen ist.

Innerhalb des UFO-Programms wird die Zielfunktion in den Matlab-Solver *fmincon* eingebunden (wenn keine Randbedingungen vorliegen *fminunc*). Dieser Solver optimiert die Zielfunktion unter Einbeziehung aller Restriktionen (siehe Formel 3.4). Der hierzu notwendige Lösungsalgorithmus kann in der Konfigurationsdatei inkl. aller relevanten Parameter definiert werden. Welche Algorithmen zur Verfügung stehen und welche Informationen diese benötigen, kann **MathWorks 2016** entnommen werden.

$$\min_x f(x) \begin{cases} x_{min} < x < x_{max} \\ g(x) < 0 \\ h(x) = 0 \end{cases} \quad (3.4)$$

### 3.3. Faserverbundwerkstoffe

Für eine sinnvolle Optimierung ist Wissen um die Eigenschaften des zu verwendenden Materials sowie der dazugehörigen Fertigungsverfahren wichtig. Speziell beim Einsatz von Faserverbundwerkstoffen (FVW) sind aufgrund ihrer Zusammensetzung einige Besonderheiten zu beachten, weswegen nachfolgend relevantes Wissen zusammengefasst ist. Die Informationen sind anhand der Literatur **AVK 2014** und **Flemming 2003** aufbereitet.

#### 3.3.1. Allgemeines FVW

Verbundwerkstoffe entstehen, wenn unterschiedliche Materialien zu einem festen Verbund kombiniert werden. Die Intention hierbei ist, einen Werkstoff herzustellen, dessen Eigenschaften die der einzelnen Komponenten übersteigen. Durch Wahl und Einteilung der Eingangsmaterialien können die Eigenschaften des entstehenden Werkstoffs gesteuert werden. Wenn Materialien in Faserform in eine Kunststoff-Matrix eingebettet werden, werden die entstehenden Verbunde als *Faserverbundwerkstoffe* bezeichnet. Alternative Bezeichnungen sind *faserverstärkte Kunststoffe*, *Faserverbundstrukturen* und *Faserverbundkunststoffe*. Der große Vorteil der FVW sind die hohen spezifischen Kennwerte, aufgrund derer sie ideal für den Leichtbau geeignet sind. Durch die Gestaltungs- und Auslegungsmöglichkeiten der Verbundstruktur wird es zudem ermöglicht, den Werkstoff den Anforderungen anzupassen. Die Fasern dienen hierbei der Übertragung von Zugkräften und geben dem Verbund Festigkeit und Steifigkeit. Das umschließende Harz ermöglicht die Übertragung von Biege- und Druckbelastungen und ist zudem gestalt- und oberflächengebend. Die Verbindung (Haftung) zwischen Matrix und Fasern wird als *Interface* bezeichnet. Als Fasern kommen hauptsächlich Kohlenstoff-, Glas-, Aramid- und verschiedene Naturfasern zum Einsatz. Hierbei heben sich die Kohlenstofffasern hinsichtlich der Dichte- und Steifigkeitseigenschaften nochmals deutlich hervor, was sie für den Einsatz in hoch belasteten Strukturen prädestiniert. Die Matrix bildet in der Regel ein duromerer oder thermoplastischer Kunststoff.

#### 3.3.2. Faserverläufe

Bei der Herstellung von FVW ist es heutzutage üblich, Lamine mit Lagen aufzubauen, in denen die Fasern innerhalb der Lage geradlinig unter einem konstanten Winkel verlaufen. Die Variation der Eigenschaften entsteht ausschließlich über das Stacking, das heißt die Ausrichtung und Reihenfolge der Lagen. Übliche Winkelrichtungen sind  $0^\circ$ ,  $30^\circ$ ,  $45^\circ$ ,  $60^\circ$  und  $90^\circ$ , die sowohl positiv als auch negativ zur x-Achse des Bauteils definiert sein können. Auf diese Weise werden die Potenziale der FVW allerdings nicht vollständig ausgeschöpft. Hierzu muss die Faserrichtung auch innerhalb einer Lage den Belastungen angepasst werden. Mittels krummlinig drapierter Fasern wird es möglich, ihren Verlauf den Richtungen der Hauptspannungen in der Struktur anzunähern. Für diese krummlinigen Fasern existieren im Englischen viele unterschiedliche Bezeichnungen, z.B. *curvilinear fiber paths*, *curved tow paths* und *steered fibers*.

### 3.3.3. Herstellungsverfahren

Zur Herstellung von FVW gibt es unterschiedliche Verfahren. Die Anforderung, krummlinige Faserverläufe herstellen zu können, schließt jedoch die meisten dieser Fertigungsverfahren aus. Deshalb wird an dieser Stelle nur auf *automatisierte Legeverfahren* eingegangen, da diese die Herstellung von Strukturen mit krummlinigen Faserverläufen ermöglichen. Diese werden grundsätzlich in Automated Tape Layup (ATL) and Automated Fiber Placement (AFP) unterschieden. Kern dieser Technologien ist ein Ablegekopf, der eine richtungs- und positionsvariable Ablage unidirektional faserverstärkter Tapematerialien ermöglicht.[QUELE] Somit werden verbesserte Werkstoffeigenschaften und die Freiheit, gekrümmte Fasern zu drapieren, ermöglicht. Die Drapierfähigkeit und der prozessierbare Krümmungsradius ist maßgeblich vom verwendeten Halbzeug, also dem Material und der Tapebreite, abhängig. Durch Beweglichkeit des Tapelegers in allen sechs Freiheitsgraden können komplexe Strukturen gefertigt werden. Allerdings kann es hierbei zu sog. *Defekten* wie Lücken und Überlappungen kommen. Als Matrixsystem können sowohl Duro- als auch Thermoplaste eingesetzt werden, jedoch ist die Prozessreife bei Verwendung von Duroplatten aktuell fortgeschrittener.

Beim AFP-Verfahren mit Tapes maximaler Breiten von 15 mm sind laut **Lemaire** minimale Krümmungsradien von 500 mm möglich. Zudem können durch die schmalen Bänder die späteren Materialeigenschaften lokal sehr präzise gesteuert werden. Die ATL-Technik hingegen nutzt bis zu 300 mm breite Bänder, wodurch sie zwar eine höhere Produktivität, jedoch eine eingeschränkte Flexibilität aufweist. Daher eignet sich diese eher für große, ebene Strukturen. Hinsichtlich Fertigungsverfahren, die großflächige Halbzeuge einsetzen, ist bei den automatisierten Legeverfahren die insgesamt höhere Fertigungszeit nachteilhaft. Dieser Umstand entsteht aufgrund der geringen drapierten Fläche pro Durchlauf. Das ist besonders beim AFP-Verfahren der Fall, weswegen häufig sog. Multi-Tapeleger eingesetzt werden, die mehrere Bänder zeitgleich legen und so die Produktivität steigern. Nach Ablage sämtlicher Lagen muss das Laminat im Autoklav oder Ofen ausgehärtet werden.

Aus dem Herstellungsverfahren leiten sich Restriktionen für die Faserverläufe ab, die eine spätere Produzierbarkeit garantieren sollen. Hauptbeispiel ist die maximale Krümmung, die der Legekopf abfahren kann bzw. wie stark das Fasertape gekrümmt werden kann. Diese Restriktionen müssen im Optimierungsprozess mathematisch erfasst werden (vergl. Kapitel 3.2.4).

### 3.3.4. Mechanische Kennwerte

Im Gegensatz zu Metallen haben Faserverbundwerkstoffe anisotrope Materialeigenschaften. Ihre mechanischen Kennwerte in die einzelnen Koordinatenrichtungen hängen maßgeblich von den verwendeten Materialien und der Orientierung der Fasern ab. Die höchsten Materialeigenschaften entstehen in Längsrichtung der Fasern, weswegen die Kennwerte hauptsächlich durch die Faserorientierung und -lage bestimmt werden. Quer zur Faserrichtung sind sie größtenteils vom eingesetzten Harz und dem Interface abhängig. Diese Abhängigkeit der Kennwerte von der Faserausrichtung ist der Ansatzpunkt dieser Optimierungsuntersuchung.

Bei den Eigenschaften muss zwischen den Kennwerten unidirektionaler Fasertapes bzw. zweidimensionaler Faserhalbzeuge (z.B. Gewebe und Gelege) und denen dreidimensional geschichteter Laminats unterschieden werden. Im Allgemeinen müssen die Kennwerte für jedes Laminat entsprechend berechnet und geprüft werden. FE-Programme sind jedoch in der Lage, verschmierte Materialeigenschaften eines Laminatverbundes aus dem definierten Lagenaufbau und den verwendeten Materialien zu berechnen.

### 3.3.5. Festigkeitskriterien

Um aus den für Materialien bekannten einachsigen Festigkeitswerten die Belastbarkeit bei mehrachsigen Spannungszuständen berechnen zu können, wurden sog. *Festigkeitskriterien* entwickelt. Für metallische Werkstoffe ist die Vergleichsspannung nach von Mises weit verbreitet. Da FVW jedoch keine isotropen Werkstoffe sind, können die für Metalle üblichen Festigkeitskriterien nicht angewandt werden. Aus diesem Grund wurden unterschiedliche Festigkeitskriterien für die Berechnung des Bruchverhaltens von FVW entwickelt, die bei der Auslegung und Dimensionierung helfen. Diese basieren zumeist auf den in Tabelle 3.4 genannten fünf unabhängigen Materialparametern, auch Basisfestigkeiten genannt, in Verbindung mit den in den Materialhauptachsen auftretenden Normalspannungen  $\sigma_1$  und  $\sigma_2$  sowie der Schubspannung  $\tau_{12}$ .

Tabelle 3.4.: Materialparameter / Basisfestigkeiten

$X_t, X_c$	Zug-/Druckfestigkeit in Faserrichtung
$Y_t, Y_c$	Zug-/Druckfestigkeit quer zur Faserrichtung
$S$	Schubfestigkeit in der Schichtebene

Die aufgestellten Kriterien reichen von einfachen, allerdings auch verhältnismäßig unpräzisen Ansätzen (z.B. das *Kriterium der maximalen Spannungen*) bis hin zu aussagekräftigen Varianten, die relativ präzise Vorhersagen hinsichtlich der Versagenslast und auch der Versagensart ermöglichen. Hierzu zählen bspw. die im Flugzeugbau verbreiteten *Bruchtyp-Kriterien von Puck*. Um die Ergebnisse **Hondas** verifizieren zu können, wird in dieser Arbeit das *Festigkeitskriterium nach Tsai-Wu* betrachtet. Die Berechnung erfolgt nach Formel 3.5 mit den Faktoren aus den Formeln 3.6 - 3.11. Die sog. *Tsai-Wu-Indizes*  $\Phi$  können mit dem ZTL-Kriterium zur Berechnung von Reservefaktoren genutzt werden, was für das weitere Vorgehen jedoch nicht von Relevanz ist.

$$\Phi = F_1 \cdot \sigma_1 + F_2 \cdot \sigma_2 + F_{11} \cdot \sigma_1^2 + F_{22} \cdot \sigma_2^2 + 2 \cdot F_{12} \cdot \sigma_1 \cdot \sigma_2 + F_{66} \cdot \tau_{12}^2 \quad (3.5)$$

mit

$$F_1 = \frac{X_c - X_t}{X_c \cdot X_t} \quad (3.6)$$

$$F_2 = \frac{Y_c - Y_t}{Y_c \cdot Y_t} \quad (3.7)$$

$$F_{11} = \frac{1}{X_c \cdot X_t} \quad (3.8)$$

$$F_{22} = \frac{1}{Y_c \cdot Y_t} \quad (3.9)$$

$$F_{12} = \frac{1}{(X_c \cdot X_t \cdot Y_c \cdot Y_t)^2} \quad (3.10)$$

$$F_{66} = \frac{1}{S^2} \quad (3.11)$$

Bei der Formel 3.10 besteht die Möglichkeit je nach Auslegung einen Vorfaktor ( $-1 < x < +1$ ) einzubringen. Allerdings wird in **Flemming 2003** dem Faktor F12 die physikalische Sinnhaftigkeit abgesprochen, weswegen dieser auch komplett null gesetzt werden kann.

### 3.4. FEM

Ein wichtiger Teil des Optimierungsprozesses ist die Strukturberechnung der variierten Entwürfe. Diese Berechnungen sollen unter Zuhilfenahme der *Finite-Elemente-Methode* erfolgen, weswegen in diesem Kapitel ein allgemeiner Überblick sowie die notwendige Theorie zum

Verständnis der Untersuchungen gegeben werden. Für weiterreichende Informationen werden **Klein 2015** und **Rieg 2014** empfohlen, anhand derer auch diese Zusammenfassung entstanden ist.

#### 3.4.1. Allgemeines FEM

Die Finite-Elemente-Methode (FEM) ist ein numerisches Berechnungsverfahren für Strukturen, die nicht mehr (sinnvoll) analytisch berechnet werden können. Der Ansatz hierbei ist, eine Struktur in kleinere, endliche (*finite*) Elemente zu unterteilen, die deutlich leichter zu berechnen sind. Wichtig hierbei ist die Berücksichtigung von Rand- und Übergangsbedingungen zwischen den Elementen. Diese Unterteilung der Struktur erzeugt eine Gitterstruktur mit Zellen bzw. Elementen und Knoten.

Die meisten Gleichungslöser (Solver) arbeiten nach dem Verschiebungsgrößen-Verfahren, dem Formel 3.12 zugrunde liegt.  $K$  ist hierbei die Gesamtsteifigkeitsmatrix,  $U$  der Vektor der Verschiebungen und  $F$  der Vektor der äußeren Kräfte. Die Material- und Geometriedaten bilden die Grundlage zur Berechnung der Steifigkeitsmatrix und die gegebenen Belastungen stellen die äußeren Kräfte dar. Der Solver löst die Formel nach den Verschiebungen auf, aus denen er wiederum auf Spannungen, Dehnungen und Reaktionskräfte zurückrechnen kann. Bei der Nutzung der FEM muss stets bedacht werden, dass die Ergebnisse sich dem realen Bauteilverhalten nur annähern.

$$K \cdot U = F \quad (3.12)$$

#### 3.4.2. Art der Berechnung

Bei der Berechnung von Strukturen wird zwischen der *linearen* und der *nicht-linearen* FEM unterschieden. Der lineare Ansatz geht von kleinen Verschiebungen und einem linear-elastischen Materialverhalten aus. Hierdurch kann die benötigte Rechenzeit zwar signifikant reduziert werden, jedoch liefert der Ansatz nur unter bestimmten Voraussetzungen korrekte Ergebnisse. Solange der Anwender die Einschränkungen des linearen Rechenansatzes berücksichtigt und sie nicht im Bereich der nicht-linearen Werkstoffkennwerte oder Geometrien einsetzt, sind die erzielten Ergebnisse für die meisten Anwendungen hinreichend genau.

#### 3.4.3. FE-Programme

Grundsätzlich muss zwischen Programmen, die als Prä- und Postprozessor fungieren, und den FE-Solvern unterschieden werden. Mittels ersterer werden die zu untersuchenden Modelle erzeugt ("Prä"). Hierbei wird die Geometrie und ein Gitter aus finiten Elementen geschaffen. Weiterhin werden die Werkstoffkennwerte sowie die Lasten und Randbedingungen definiert. Nach Abschluss der Berechnung können die Ergebnisse visuell ausgewertet werden ("Post"). Prä- und Postprozessor dienen somit der einfacheren Handhabung. Die tatsächliche Berechnung der Modelle, also die Anwendung der FEM, erfolgt durch die Solver. Sie lösen die aufgestellten Differentialgleichungen unter Berücksichtigung aller gegebenen Informationen und stellen die gewünschten Ausgangsdaten bereit.

#### Verwendete Programme

Mit der Zeit haben sich auf dem Markt unterschiedliche Programme bzw. Programmkombinationen etabliert. Hier gibt es jedoch kein universell zu empfehlendes Programm, da je nach Problemstellung die einzelnen Solver Vor- und Nachteile haben. Durch die Aufgabenstellung ist die Verwendung von MSC Patran zur FE-Modell-Erstellung sowie von MSC Nastran zur Berechnung ebendieser Modelle vorgegeben. Durch die Wahl eines linearen Rechenansatzes, macht die Nutzung Nastrans Sinn, da dieses laut **Schumacher 2013** im Bereich der linearen Statik empfehlenswert ist. MSC Patran ist ein speziell für den Solver Nastran entwickelter Prä-Post-Prozessor und somit ideal auf die Ein- und Ausgabe Nastrans abgestimmt.

#### 3.4.4. Relevante Dateiformate

Im späteren Optimierungsprozess sind verschiedene Dateiformate Nastrans von wesentlicher Bedeutung, weswegen ihr Inhalt und ihre Funktion an dieser Stelle kurz erläutert werden. Die *bdf*- und *f06*-Dateien werden im ASCII-Code geschrieben, was ein einfaches Lesen und Modifizieren mit Textverarbeitungsprogrammen ermöglicht.

bdf-Format: Eine *bdf*-Datei fungiert als Eingabedatei für Nastran. In ihr sind alle für die Berechnung relevanten Daten hinterlegt. Hierzu zählen die Geometrie, die FE-Vernetzung, die Materialdaten, die Lasten, die Randbedingungen, der Lösungsansatz und die gewünschten Ausgabeparameter.

f06-Format: In einer *f06*-Datei werden von Nastran alle geforderten Ergebnisse der Berechnung ausgegeben. Je nach Einstellung sind sämtliche Informationen zu Spannungen,

Lagerkräften, Verschiebungen, Festigkeitskriterien, etc. aufgeführt.

xdb-Format: Zum Einlesen der Ergebnisse im Postprozessor werden diese zusätzlich im xdb-File abgespeichert. Im Gegensatz zum f06-Format erfolgt dies jedoch nicht im ASCII-Code. Die Datei dient im späteren Verlauf zur Darstellung optimierter Modelle.

#### 3.4.5. Parameter-Karten

Im bdf-Format werden einige Informationen in sogenannten *Karten* hinterlegt. Zur eindeutigen Zuordnung erhalten alle Karten eine Identifikationsnummer (ID). Die zwei nachfolgend beschriebenen Arten von Parameter-Karten sind für den späteren Optimierungsprozess von besonderer Bedeutung. Die Daten sind **QRG 2015** entnommen, welcher auch für detaillierte Informationen empfohlen wird.

PCOMP-Karte: In einer PCOMP-Karte ('P'roperties 'COMP'osite) werden die Eigenschaften eines Verbundwerkstoff-Laminats definiert. Mit ihrer ID können einer Zelle, einem Zellbereich oder der gesamten Struktur die hinterlegten Eigenschaften zugewiesen werden. In Tabelle 3.5 sind die relevanten Parameter der PCOMP-Karten beschrieben.

Tabelle 3.5.: Optionen PCOMP-Karte

<b>Feld</b>	<b>Beschreibung</b>
PID	ID der PCOMP-Karte
SB	Zulässige interlaminare Schubspannung. Nötig, wenn FT aktiv
FT	Die anzuwendende Fehlertheorie, bspw. Tsai-Wu-Theorie (TSAI)
LAM	Optionen bzgl. des Laminataufbaus, bspw. Symmetrie (SYM)
MIDi	ID der Materialkarte, die der jeweiligen Lage zugeordnet werden soll
Ti	Dicke der jeweiligen Lage
THETAi	Winkel zwischen der Längsrichtung der jeweiligen Lage und der Materialrichtung
SOUTi	Ausgabe der Spannungen und Dehnungen für die jeweilige Lage (YES/NO)

MAT8-Karte: In der MAT8-Karte werden sämtliche Materialkennwerte für ein orthotropes Material für Schalenelemente hinterlegt. Über die ID greifen die PCOMP-Karten auf die jeweilige MAT8-Karte zu. In Tabelle 3.6 sind die relevanten Parameter der MAT8-Karte beschrieben.

Tabelle 3.6.: Optionen MAT8-Karte

<b>Feld</b>	<b>Beschreibung</b>
MID	ID des Materials
E1, E2	E-Module in Längs- und Querrichtung
NU12	Querkontraktionszahl
G12,G1Z,G2Z	Schubmodule
Xt, Xc	Zug- (t) und Druck- (c) Festigkeit in Längsrichtung
Yt, Yc	Zug- (t) und Druck- (c) Festigkeit in Querrichtung
S	In-plane Schubfestigkeit
F12	Interaktionsfaktor zur Berechnung nach Tsai-Wu

### 3.5. Mathematik

Die Mathematik als Grundlage eines jeden Optimierungsprozesses wurde bereits in Kapitel 3.2 abgehandelt. In diesem Kapitel geht es nun um mathematische Hilfsmittel, die im Verlauf der Optimierung zusätzlich gebraucht werden.

#### 3.5.1. Bezierflächen

Das folgende Unterkapitel beruht auf Informationen von **Holldack 2011** und **Zhang 2006**. Bézierkurven ermöglichen eine Berechnung von flexiblen Kurven mit nur wenigen Parametern. Aus diesem Grund werden sie vielfach im Bereich der CAD-Anwendungen (Computer Aided Design) sowie für weitere Computergrafiken eingesetzt. Mit Hilfe von Bézierkurven ist es möglich, eine *Bézier-Fläche* aufzuspannen, welche ebenfalls trotz weniger Parameter sehr komplexe Formen abbilden kann. Die schematische Darstellung einer Bézier-Fläche mit ihren Kontrollpunkten ist in Abbildung 3.2 gegeben. Die Berechnung der Bézierkurven und -flächen erfolgt unter Zuhilfenahme von Bernstein-Polynomen, die hierfür entscheidende Eigenschaften (Rekursivität und Baseneigenschaft) aufweisen. Eine häufig angewandte Berechnungsvorschrift zur Lösung ist der *De-Casteljau-Algorithmus*.

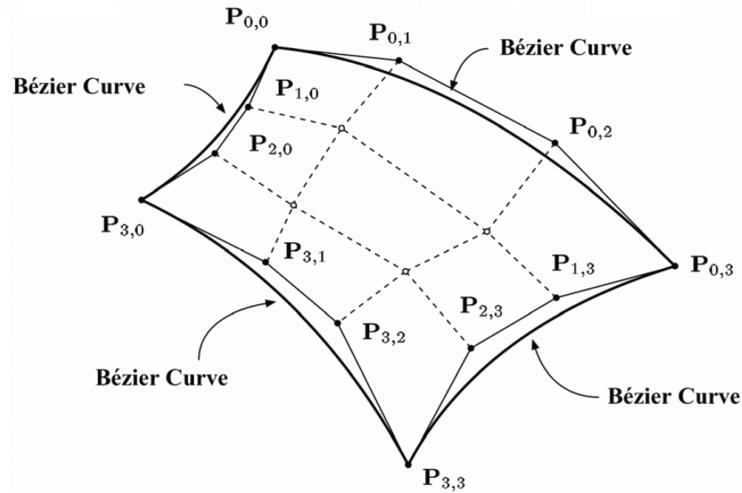


Abbildung 3.2.: Schematische Darstellung einer Bézier-Fläche [Zhang 2006]

Formel 3.13 zeigt die allgemeine Formulierung einer Bézier-Fläche, mit den Kontrollpunkten  $P_{i,j}$  und den Bernstein-Polynomen  $B_{i,n}(u)$  und  $B_{j,m}(v)$ . Die Parameter  $n$  und  $m$  sind der jeweilige Grad der Bernstein-Polynome, aus denen die Anzahl von  $(n + 1)(m + 1)$  Kontrollpunkten resultiert. Diese definieren die Form der Fläche, während die zwei Variablen  $u$  und  $v$  zur Positionsbestimmung auf der Fläche dienen. Aufgrund der Berechnungsform hat eine Verschiebung eines beliebigen Kontrollpunktes Einfluss auf die gesamte Fläche. Damit Änderungen nur noch einen lokalen Einfluss haben, können Formen aus mehreren Einzelflächen zusammengesetzt werden. Um eine kontinuierliche Fläche zu erzeugen, spielen hierbei verschiedene Übergangsbedingungen eine entscheidende Rolle.

$$P(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{i,j} B_{i,n}(u) B_{j,m}(v) \quad (3.13)$$

Für die programmiertechnische Umsetzung ist die Matrixschreibweise der Bézier-Fläche von besonderem Interesse. Diese ist in Formel 3.14 beispielhaft für die übliche Definition einer Bézier-Fläche mit 16 (4x4) Kontrollpunkten dargestellt. Die Kontrollpunkte sind in der mittleren Matrix  $P$  definiert und  $u$  und  $v$  sind weiterhin die Variablen. In Matrix  $M$  (s. Formel 3.15) stehen

die aus den Bernstein-Polynomen entstehenden Faktoren, weswegen die Matrixdimensionen vom Grad der Bernstein-Polynome abhängig sind.

$$x(u, v) = \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} M \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix} M^T \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix} \quad (3.14)$$

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \quad (3.15)$$

### 3.5.2. Bestimmung geometrischer Schwerpunkte

Im späteren Verlauf wird es notwendig sein, den geometrischen Schwerpunkt beliebiger Vielecke zu bestimmen. Hierzu dienen folgende, allgemeingültige Formeln. Die Gaußsche Dreiecksformel (Formel 3.16) dient zur Berechnung des Flächeninhalts des Polygons. Mit diesem können anschließend die x- und y-Koordinaten des Flächenschwerpunktes anhand der Formeln 3.17 und 3.18 bestimmt werden.

$$A = \frac{1}{2} \sum_{i=0}^{N-1} (x_i y_{i+1} - x_{i+1} y_i) \quad (3.16)$$

$$x_s = \frac{1}{6A} \sum_{i=0}^{N-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (3.17)$$

$$y_s = \frac{1}{6A} \sum_{i=0}^{N-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (3.18)$$

### 3.5.3. Funktionsableitungen

Ist eine Funktion  $f(x)$  gegeben, gibt ihre erste Ableitung  $f'(x)$  Informationen über die Steigung  $m_t$  der Kurventangente am Punkt  $x$  wieder. Zusätzlich bestimmt der Wert der zweiten Ableitung  $f''(x)$  das Krümmungsverhalten (Links- oder Rechtskrümmung) der Funktion am Punkt  $x$ . Liegt eine Funktion vor, können die Ableitungen per Differentialrechnung bestimmt und am gesuchten Punkt berechnet werden. Die notwendigen Zusammenhänge sind in Formel 3.19 aufgeführt. Der letzte Term der Gleichung dient zur analytischen Berechnung der Tangen-

tensteigung, wenn statt einer Funktion die Koordinaten der Punkte eines Funktionsverlaufes gegeben sind. Um hiermit verlässliche Werte zu erhalten, müssen die Koordinaten möglichst dicht beieinander liegen. Die zweite Ableitung wird durch erneutes Ableiten erhalten. Sind beide Ableitungen bekannt, kann mittels Formel 3.20  $\kappa$ , ein quantitatives Maß der Krümmungsstärke, berechnet werden. Sämtliche Informationen dieses Unterkapitels sind **Papula 2008** entnommen.

$$m_t = \tan(\alpha) = f'(x_0) = \lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = \frac{y - y_0}{x - x_0} \quad (3.19)$$

$$\kappa(x) = \frac{f''(x)}{(1 + f'(x)^2)^{3/2}} \quad (3.20)$$

### 3.5.4. Kreisselmeier-Steinhauser-Funktion

Mit der *Kreisselmeier-Steinhauser-Funktion* (KS-Funktion) können  $n$  Funktionen zu einer einzelnen Funktion kombiniert werden. Die KS-Funktion nähert sich von oben (alternativ von unten) an die gegebenen Funktionen an, hat jedoch die Eigenschaft, die eingehenden Funktionen nie zu erreichen. Mit dem Parameter  $\rho$  kann der Grad der Approximation festgelegt werden. **Wrenn 1989** gibt für  $\rho$  übliche Werte im Bereich 5-200 an. Je kleiner  $\rho$  gewählt wird, desto größer ist die Differenz zwischen den eingehenden Funktionen und der KS-Funktion, wodurch das Ergebnis konservativer ist. Im Optimierungsprozess kann die KS-Funktion für unterschiedliche Aufgaben genutzt werden, z.B. zur Umformulierung von Restriktion oder zur Zusammenfassung mehrerer Ausgabeparameter zu einem einzelnen Wert. Da es bei großen Werten in der Exponentialfunktion zu numerischen Problemen kommen kann, wird eine alternative Formulierung der KS-Funktion genutzt werden (s. Formel 3.21).

$$KS(g_j(x), \rho) = \max(g_j(x)) + \frac{1}{\rho} \ln \left[ \sum_{j=0}^m e^{\rho \cdot (g_j(x) - \max(g_j(x)))} \right] \quad (3.21)$$

## 4. FE-Modellbildung

Um die für den Optimierungsprozess notwendige Strukturantwort auf die veränderten Entwurfsvariablen zu berechnen, soll ein FE-Modell genutzt werden. Dieses muss so aufgebaut sein, dass es problemlos vom Optimierungsprogramm bei jeder Iteration verändert werden kann. Das gewählte Beispiel ist die in Abbildung 4.1 dargestellte Rechteckplatte mit mittigem Loch.

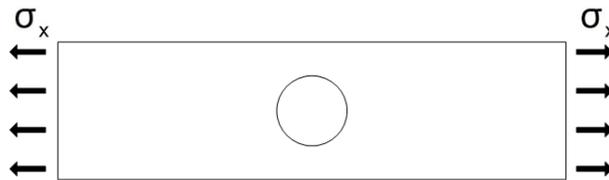


Abbildung 4.1.: Darstellung der Lochplatte

Welche Vorüberlegungen hierfür zu treffen sind und wie das Modell in die Praxis umgesetzt wird, sind die Hauptthemen dieses Kapitels. Zusätzlich werden Vergleichswerte für die spätere Optimierung erzeugt und die Güte der Ergebnisse des Modells geprüft.

### 4.1. Vorüberlegungen zum FE-Modell

Vor Erstellung des FE-Modells ist es wichtig, einige Vorüberlegungen zu treffen. So muss geklärt werden, wie krummlinige Faserverläufe mittels der FEM berechnet werden können und welche Besonderheiten bei einem Modell mit Faserverbundwerkstoffen zu beachten sind. Des Weiteren ist zu bestimmen, welcher Lösungsansatz der Gleichungen und welche Vernetzung für das vorliegende Problem genutzt werden soll. Von Interesse ist außerdem, welche Vereinfachungen für die Berechnung getroffen werden können. Ein weiterer Punkt ist die Klärung, wie das FE-Modell aufgebaut sein muss, um in den Optimierungsprozess integriert werden zu können.

#### 4.1.1. Ansatz der Umsetzung

Da die direkte Berechnung von krummlinigen Faserverläufen mit Nastran noch nicht möglich ist, muss ein Alternativansatz gewählt werden. Eine übliche Methode hierzu ist, die zu untersuchende Struktur zu vernetzen und jeder Zelle einen eigenen Faserwinkel zuzuweisen. In Bild 4.2 ist dieser Ansatz grafisch dargestellt. Auf diese Weise werden krummlinige Faserverläufe imitiert. Der Ansatz ermöglicht im gesamten Strukturbereich variable Faserverläufe, die sich lokalen Einflüssen ideal anpassen können (vgl. Kapitel 3.3.2).

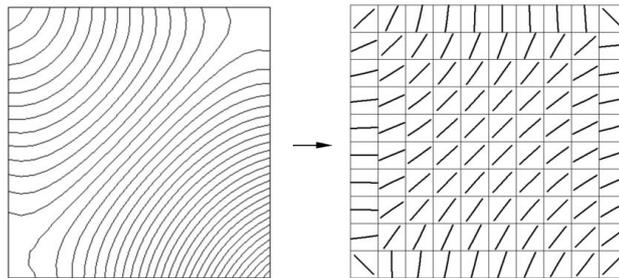


Abbildung 4.2.: Entstehung der Zellwinkel [Honda]

Grundsätzlich ist es dabei vorteilhaft, Bedingungen zu definieren, die einen kontinuierlichen Faserverlauf zwischen benachbarten Zellen gewährleisten. Dies würde allerdings die Designfreiheit zu stark einschränken, weshalb darauf verzichtet wird. Der Berechnungsansatz erzeugt somit eine grobe Vereinfachung der Faserverläufe, weshalb diese nur als schematische Verläufe angesehen werden dürfen. Daher ist eine anschließende Weiterverarbeitung der Ergebnisse erforderlich, um reale Faserverläufe fertigen zu können. Hierauf wird später in Kapitel 8.2 nochmals eingegangen.

#### 4.1.2. Unterschiede bei einem FVW-Modell

Bei der Berechnung von FE-Modellen mit FVW sind zwei wesentliche Unterschiede zur Verwendung von metallischen Materialien zu beachten.

Zum einen unterscheiden sich die Definitionen der *Materialeigenschaften*. FVW können auf unterschiedliche Weisen erzeugt werden. Daher ist es wichtig, vorab zu entscheiden, wie das Laminat modelliert werden soll. Die Geometrie kann mit dünnen Volumenelementen erzeugt werden, denen die jeweiligen Materialeigenschaften zugewiesen werden. Eine alternative Methode besteht darin, ein Laminat, dessen Lagenaufbau vorab definiert wird, als Material

zu erzeugen und dieses mittels PCOMP-Karten Platten- oder Schalenelementen zuzuordnen. Hierbei ergibt sich die Dicke der Geometrie aus den angegebenen Lagendicken und der Lagenanzahl. In dieser Arbeit wird der zweite Ansatz gewählt, da nur ein zweidimensionaler Lastfall betrachtet werden soll und er besser in den Optimierungsprozess integrierbar ist.

Wie in Kapitel 3.3.5 beschrieben, können die für Metalle üblichen Ansätze der Vergleichsspannungen zur Festigkeitsbetrachtung nicht herangezogen werden. Daher muss eines der beschriebenen *Festigkeitskriterien* (oder auch Versagenskriterien) für die Berechnung von FVW betrachtet werden. Soll Nastran eines dieser Kriterien automatisch berechnen, muss dieses festgelegt und die je nach Kriterium zusätzlich erforderlichen Materialkennwerte hinterlegt werden.

### 4.1.3. Rechenansatz und Vernetzung

Um ein möglichst reales Verformungsverhalten abbilden zu können, bieten FE-Solver nicht-lineare *Rechenansätze* (siehe Kapitel 3.4.2). Da diese jedoch wesentlich mehr Rechenkapazität benötigen und auch lineare Ansätze bereits aussagekräftige Ergebnisse liefern, die für den Zweck dieser Arbeit ausreichend sind, wird eine lineare Berechnung vorgezogen. Dies geschieht speziell in Anbetracht der vielen notwendigen Berechnungswiederholungen. Bei Nastran erfolgen lineare statische Berechnungen mit *Solution 101 - Linear Static*.

Da es sich um eine zweidimensionale Struktur handelt, soll die *Vernetzung* mittels viereckigen Plattenelementen durchgeführt werden. Aufgrund ihrer höheren Berechnungsgüte (**Klein 2015**) und der verhältnismäßig simplen Strukturgeometrie werden Rechteckelemente Dreieckselementen vorgezogen. Es wird der Typ *CQUAD4* gewählt, der an allen vier Ecken jeweils einen Knoten besitzt. Alternativ können zur Steigerung der Genauigkeit Rechteckelemente mit Mittelknoten, z.B. *CQUAD8*, verwendet werden. Die Anzahl und die Verteilung soll der Vernetzung **Hondas** entsprechen, wodurch die Struktur mit insgesamt 400 Elementen vernetzt wird. Die Zelldichte in der Region um das Loch ist deutlich erhöht, da in diesem Bereich die größten Spannungen erwartet werden. Auf diese Weise wird die Güte der Ergebnisse gegenüber einer gleichmäßigen Vernetzung verbessert.

### 4.1.4. Vereinfachungen des Modells

Da reine in-plane Spannungen untersucht werden sollen, wird die Betrachtung auf einen zweidimensionalen Lastfall reduziert. Dies geschieht, indem die Freiheitsgrade des Modells in

z-Richtung gesperrt werden.

Zudem ist es sinnvoll, mögliche Symmetrien zu prüfen, bevor die Lochplatte modelliert wird, um Rechenaufwand und damit verbundene Rechenzeit einzusparen. Da die Lochplatte sowohl zur x- als auch zur y-Achse symmetrisch ist, können zur Vereinfachung des Modells beide Symmetrien genutzt werden. Auf diese Weise muss nur ein Viertel der Lochplatte modelliert werden, wodurch die Anzahl der Knoten und Zellen ebenfalls auf ein Viertel reduziert wird. Die Viertelplatte mit den Symmetriekanten ist in Bild 4.3 dargestellt.

Der lineare Rechenansatz stellt auch eine Vereinfachung dar, was aber bereits im vorangegangenen Kapitel beschrieben wurde.

##### 4.1.5. Art der Einbindung

Essenziell ist die Frage, wie das Modell in den Optimierungsprozess eingebunden werden soll. Um einen möglichst effizienten Ablauf zu generieren, muss vor Erzeugung des Modells und der Schnittstelle zum UFO-Tool klar sein, wie die Schnittstelle der beiden Elemente funktionieren soll. Der Optimierungsalgorithmus muss auf der Eingabeseite den Aufbau bzw. die Veränderung der Struktur mit den Entwurfsvariablen steuern können. Das Modell muss dementsprechend parametrisch aufgebaut oder auf andere Art veränderbar sein. Auf der Ausgabeseite soll der Algorithmus die Ergebnisse möglichst einfach, z.B. als Zahlenwerte im ASCII-Code, auslesen können.

Eine Möglichkeit hierzu ist die Erzeugung einer Nastran-Eingabedatei (bdf-Format) mit dem Startentwurf, in dem alle Modellparameter hinterlegt sind. Mittels Matlab kann diese Datei in jeder Iteration entsprechend der Entwurfsvariablen angepasst werden. Der Einflussgrad auf das Modell ohne großen Programmieraufwand ist hierbei relativ gering. Um die Geometrie und die Vernetzung direkt im bdf-File zu ändern und einen korrekten Modellaufbau zu garantieren, wären komplexe Algorithmen nötig. Im Fall der zu untersuchenden Lochplatte ist aber sowohl die Geometrie als auch die Vernetzung invariabel, weswegen die Möglichkeiten dieser Variante ausreichend sind. Hinzu kommt, dass sie keinen Patran-Aufruf benötigt, was sich positiv auf die Dauer jedes Iterationsdurchlaufs auswirkt.

Um die Faserverläufe definieren zu können, muss geklärt werden, wie die Entwurfsvariablen am sinnvollsten in der bdf-Datei definiert werden. Das Ziel ist, jeder Zelle und jeder Lage einen variablen Winkel zuzuweisen. Die Zuordnung einer Orientierung kann auf unterschiedliche

Weisen erfolgen, bspw. indem das Koordinatensystem der Zelle gedreht wird. Da auf diese Weise jedoch alle Laminatlagen beeinflusst werden, ist dieser Ansatz ungeeignet. Bei Verwendung von PCOMP-Karten kann der Orientierungswinkel jeder Faserlage in der Karte direkt gesetzt werden. Wird nun jede Zelle mit einer eigenen PCOMP-Karte beschrieben, können allen Zellen und all ihren Lagen eine eigene Orientierung zugewiesen werden. Folglich müssen zum Einbringen der Entwurfsvariablen die Eigenschaftskarten in jeder Iteration vom Optimierungsprozess angepasst bzw. neu geschrieben werden. Diese Überlegung muss nun sowohl in die Erstellung des FE-Modells einfließen als auch die Entwurfsvariablen des Optimierungstools definieren.

Das Auslesen der Ergebnisse ist verhältnismäßig einfach, da Nastran mit der f06-Datei automatisch eine Ergebnisdatei im ASCII-Code erzeugt. Aus dieser können mit Matlab über entsprechende Befehle die benötigten Werte ausgelesen werden.

## 4.2. Realisierung FE-Modell

Da nun alle notwendigen Informationen gegeben sind, kann das FE-Modell mit Hilfe des Prä-Prozessors MSC Patran erstellt werden. Die Modellierung unterteilt sich in vier Hauptschritte, welche in den folgenden vier Unterkapiteln erläutert werden. Abschließend erfolgt noch eine Einstellung der gewünschten Ausgabeparameter. Die Beschreibung wird am Beispiel der zu untersuchenden Lochplatte durchgeführt, es können aber auch andere Strukturen für den Optimierungsprozess modelliert werden.

Zum Verständnis dieses Kapitels werden elementare Kenntnisse im Umgang mit MSC Patran vorausgesetzt. Es werden sämtliche Arbeitsschritte und alle relevanten Einstellungen genannt, auf grundlegende Befehle und nicht direkt relevante Einstellungsparameter wird jedoch nicht eingegangen.

### 4.2.1. Eingangsparameter

Bevor ein FE-Modell erstellt wird, muss festgelegt werden, in welchen Maßeinheiten die Eingabe der gegebenen Werte erfolgen soll. Hier werden die Dimensionen *Newton*, *Millimeter* und *Sekunde* gewählt. Für die Masse resultiert somit die Einheit *Tonne*.

Wie in Kapitel 1.2 erwähnt, sollen die Resultate von **Honda** als Verifikation der Ergebnisse dieser Arbeit dienen. Um die Vergleichbarkeit zu gewährleisten, wird das Modell geometrisch

identisch aufgebaut (siehe Bild 4.3) und die gleichen Materialeigenschaften und Lasten verwendet. Die Abmaße der Geometrie sind in Tabelle 4.1 und die Materialkonstanten in den entsprechenden Einheiten in Tabelle 4.2 aufgeführt. Lediglich bei der Bauteilvernetzung konnten die Zellgrößen nur optisch angenähert werden, da keine näheren Informationen vorliegen. Auf die verwendeten Einteilungen und weitere Informationen zur Vernetzung wird in Kapitel 4.2.3 eingegangen.

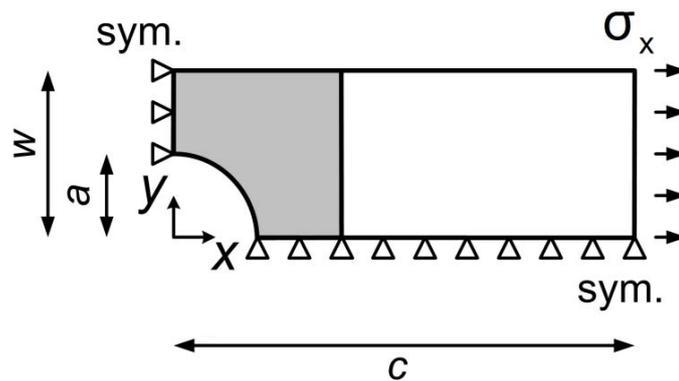


Abbildung 4.3.: Schematische Darstellung der Struktur [Honda]

Tabelle 4.1.: Abmaße der Lochplatte

Parameter	Wert
Höhe w	100 mm
Radius a	50 mm
Länge c	700 mm
Dicke t	1 mm

Tabelle 4.2.: Materialkonstanten

Parameter	Wert
E-Modul E1	138.000 $N/mm^2$
E-Modul E2	8.960 $N/mm^2$
Schubmodul G	7.100 $N/mm^2$
Querkontraktionszahl $\nu$	0.30
Zug-/Druckfestigkeit $X_t = X_c$	1448 $N/mm^2$
Zugfestigkeit $Y_t$	51.71 $N/mm^2$
Druckfestigkeit $Y_c$	206.85 $N/mm^2$
Schubfestigkeit S	90.08 $N/mm^2$
Vorfaktor F12	0

#### 4.2.2. Geometrie

Wie in Kapitel 4.1.4 beschrieben, muss aufgrund der Symmetrie lediglich ein Viertel der Lochplatte modelliert werden. In diesem Fall wird die rechte, obere Ecke verwendet, die als FE-Geometrie in Abbildung 4.4 dargestellt ist. Die Erstellung der Geometrie kann auf unterschiedliche Arten erfolgen und muss nicht zwingend der folgenden Beschreibung entsprechen. Zunächst werden alle notwendigen Punkte erzeugt und daraus die Geraden 1 bis 7 sowie der Kreisbogen gebildet. Um das Modell später besser vernetzen zu können, wird dieser in der Mitte aufgetrennt (Bögen 9 und 10) und zusätzlich die Gerade 11 erzeugt. Als letztes werden die drei Flächen I, II und III anhand ihrer jeweiligen Außenkanten definiert.



Abbildung 4.4.: Darstellung der Geometrie

#### 4.2.3. Vernetzung

Der nächste Schritt ist die Vernetzung der erzeugten Geometrie. Um die Zelldichte wie gewünscht beeinflussen zu können, werden auf allen Kanten sogenannte Mesh-Seeds erzeugt, anhand derer der Mesher das Netz erzeugt. Auf den Geraden 3,5,7,9 und 10 werden jeweils zehn

Mesh-Seeds mit dem Typ *Uniform* gleichmäßig verteilt. Auf den restlichen Kanten werden sie mit Hilfe einer verhältnismäßigen Einteilung (Typ *OneWayBias*) definiert. Die Einteilung ist in Tabelle 4.3 aufgeführt. Damit bei Verwendung der One Way Bias die Einteilung der Mesh-Seeds zur richtigen Seite erfolgt, kann es nötig sein, die Richtungen von Linien mittels *Geometry* → *Edit* → *Curve* → *Reverse* zu drehen.

Tabelle 4.3.: Einteilung Mesh-Seeds

Linie	Seed-Anzahl	Verhältnis
1	10	1:2
2	20	1:7
4	20	1:7
6	10	1:2
11	10	1:5

Sind alle Mesh-Seeds gesetzt, wird das FE-Netz mit dem *IsoMesh*-Vernetzer und *QUAD4*-Elementen erzeugt. Um von links nach rechts aufsteigende Zell-IDs zu generieren, werden die Flächen I bis III in aufsteigender Reihe vernetzt. Ideal für spätere Funktionen im Optimierungsprozess ist eine vertikal aufsteigende Nummerierung im hinteren Bereich, jedoch ist sie nicht zwingend erforderlich. Erzeugt Nastran eine horizontal aufsteigende Nummerierung kann diese per *Meshing* → *Renumber* → *Element* umgestaltet werden. Damit Nastran das Modell korrekt berechnen kann, müssen nach der Vernetzung doppelte Knoten gleichgesetzt werden. Nach der Äquivalenzprüfung hat das Modell 1335 Knoten und 400 Zellen. Davon liegt eine Hälfte im Bereich des Loches ( $0 < x < 100, 0 < y < 100$ ) und die andere im Außenbereich ( $100 < x < 700$ ). Somit ist die Zelldichte im relevanten Bereich deutlich höher. Abbildung 4.5 zeigt die gesamte Strukturvernetzung, Bild 4.6 die Vernetzung im Lochbereich mit den jeweiligen Zell-Nummern.

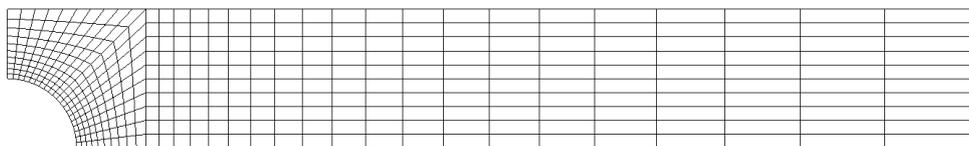


Abbildung 4.5.: Darstellung der FE-Vernetzung



bundmaterial mit *Materials* → *Create* → *Composite* → *Laminate* erzeugt wird. In dem sich öffnenden Fenster werden die einzelnen Lagen mit ihren Eigenschaften sowie der Lagenaufbau definiert. Durch Wahl von *StackingSequenceConvention* : *Symmetric* wird ein symmetrischer Laminataufbau gewählt und es muss nur die Hälfte der Lagen definiert werden. Das Lagenmaterial ist jeweils das erzeugte orthotrope Material. Die Lagendicke ergibt sich aus der Laminatdicke und der Lagenanzahl zu  $(1\text{mm}/8\text{Lagen}) = 0,125\text{mm}/\text{Lage}$  und ist ebenfalls für alle Lagen identisch. Da die Winkel später vom Optimierungsprozess direkt ins bdf-File geschrieben werden, ist der Eintrag der Orientierung ohne Relevanz und wird in allen Lagen null gesetzt. Zur Kontrolle stehen im unteren Bereich die Werte des definierten Laminats.

Damit das Modell berechenbar ist, muss das erstellte Laminat über *Properties* → *Create* → *2D* → *Shell* mit den Optionen *Thin*, *Laminate* und *StandardFormulation* noch den Zellen zugewiesen werden. Da Patran eine Warnung ausgibt, wenn Zellen keine Eigenschaften zugewiesen sind, wird der gesamten Struktur das definierte Laminat zugewiesen. Im späteren Optimierungsprozess wird jeder Zelle eine eigene Eigenschaften-Karte zugewiesen, in der der entsprechende Faserwinkel definiert ist.

#### 4.2.5. Äußere Kräfte und Randbedingungen

Die Struktur soll auf Zugbelastung untersucht werden. Hierzu wird die Zugspannung  $\sigma_t = 10\text{MPa}$  am rechten Bauteilrand (Linie 3) aufgebracht. Da nur eine 2D-Betrachtung durchgeführt wird, wird die Spannung über die Laminatdicke auf eine Linienlast umgerechnet (s. Formel 4.1) und mittels *Loads/BCs* → *Create* → *DistributedLoad* im Modell definiert. Die *DistributedLoad* verteilt die Last gleichmäßig auf sämtliche Knoten der Bauteilkante. Anhand der Lastpfeile muss kontrolliert werden, ob die Struktur tatsächlich auf Zug belastet wird. Andernfalls muss das Vorzeichen des Lastwertes geändert werden.

$$\sigma_{\text{Linienlast}} = \frac{\sigma_t}{t} = \frac{10\text{MPa}}{1\text{mm}} = 10 \frac{\text{N}}{\text{mm}} \quad (4.1)$$

Die besprochenen Symmetrien werden durch Unterdrückung der entsprechenden Freiheitsgrade an der linken und der unteren Bauteilkante simuliert und dienen gleichzeitig der Fixierung des Modells im Raum. An der unteren Kante (Linien 1 und 2) werden die FHG 2 und 4 gesperrt und somit die Verschiebung in y- Richtung und Rotation um die x-Achse unterdrückt. Analog dazu werden an der linken Modellkante (Linie 6) die FHG 1 und 5 zur Unterdrückung einer Translation in x-Richtung und einer Rotation um die y-Achse gesperrt. Die FHG der

z-Richtung (3 und 6) werden ebenfalls fixiert, da ein zweidimensionaler Lastfall betrachtet werden soll.

#### 4.2.6. Ausgangsparameter

Mit der Definition der Lasten ist die Modellbildung abgeschlossen. Bevor die Berechnung jedoch gestartet wird, müssen die geforderten Ausgabewerte eingestellt werden. Als Zielfunktion des Optimierungsprozess wurde der maximale Tsai-Wu-Index festgelegt. Die Tsai-Wu-Indizes können theoretisch aus den von Nastran berechneten Spannungen und den Materialkennwerten mittels der Formel 3.5 berechnet werden. Nastran bietet aber die Option, die Tsai-Wu-Indizes automatisch zu berechnen und auszugeben. Diese Lösung wird gewählt, da die Berechnung nach der selben Formel erfolgt und auf diese Weise nachträglicher Rechenaufwand erspart wird. Hierzu wurde bereits in Kapitel 4.2.4 das Versagenskriterium eingestellt und die entsprechenden Materialkennwerte eingetragen. Es werden für jede Zelle die Indizes in jeder Faserlage des Laminats berechnet und ausgegeben. Zusätzlich wird für jede Zelle auch der maximale Tsai-Wu-Index deklariert. Da für den Optimierungsprozess ausschließlich die Spannungen und die damit verbundenen Tsai-Wu-Indizes von Interesse sind, ist es sinnvoll, alle anderen Ausgabewerte unter *Analysis* → *Subcases...* → *OutputRequests...* abzuwählen. So wird Rechenzeit und Speicherplatz gespart.

### 4.3. Erzeugung von Vergleichswerten

Um die Ergebnisse des Optimierungsprozesses quantitativ bewerten zu können, müssen Vergleichswerte eines Laminats mit gradlinig verlaufenden Fasern vorliegen. Da im Optimierungsprozess zunächst ein Laminat mit acht Faserlagen untersucht werden soll, wird für die Vergleichsrechnung das FE-Modell mit einem  $[0^\circ \pm 45^\circ 90^\circ]_s$ - sowie einem reinen  $0^\circ$ -Lagen-Laminat versehen. Abgesehen von den Faserwinkeln sind alle Eingabewerte identisch mit denen des späteren Optimierungsmodells.

Die Nastran-Rechnung liefert für das  $[0^\circ \pm 45^\circ 90^\circ]_s$ -Laminat einen maximalen Tsai-Wu-Index von 0,0894. Dieser tritt in Zelle 91 (Oberseite des Loches, siehe Bild 4.6) in den  $90^\circ$ -Lagen auf. Dies resultiert aus Druckspannungen in Faserrichtung. Für das Laminat aus  $0^\circ$ -Lagen tritt der Maximalwert in allen Schichten der Zelle 191 (rechte Seite des Loches) auf und liegt bei 0,0475. Hauptursache sind Druckspannungen quer zum Faserverlauf. Da eine reine Zugbelastung in x-Richtung vorliegt, macht das deutlich bessere Ergebnis des  $0^\circ$ -Lagen-Laminats Sinn. Auf diese Informationen wird später in der Auswertung in Kapitel 8 zurückgekommen.

#### 4.4. Validierung des Modells

Die Werte aus Unterkapitel 4.3 bieten sich zusätzlich an, um die Berechnung der Tsai-Wu-Indizes zu validieren. Dies ist wichtig, um diesen Rechenschritt bei möglichen späteren Problemen als Fehlerquelle auszuschließen. Zusätzlich wird eine kleine Konvergenzstudie der FE-Vernetzung durchgeführt, um die Verlässlichkeit der Ergebnisse zu prüfen.

Zur Prüfung der von Nastran berechneten Tsai-Wu-Indizes werden in der folgenden Beispielrechnung die Spannungen der  $90^\circ$ -Lage in der Zelle 91 des  $[0^\circ \pm 45^\circ 90^\circ]_s$ -Laminats in die Tsai-Wu-Formel 3.5 eingesetzt und mit den Materialkennwerten berechnet. Die auftretenden Spannungen sind  $\sigma_1 = -23,26\text{N/mm}^2$ ,  $\sigma_2 = 5,916\text{N/mm}^2$  und  $\tau_{12} = -0,431\text{N/mm}^2$ . Zur Berechnung von  $F_{12}$  wird in Anlehnung an **Honda** kein Vorfaktor angenommen. Hinsichtlich des geringen  $F_{12}$ - Wertes hat eine Variation des Faktors auch nur minimale Auswirkungen auf den Tsai-Wu-Index. Bei der auf vier Nachkommastellen begrenzten Nastran-Ausgabe verursacht er keine Veränderung. Aus Gründen der Übersichtlichkeit wird auf Mitführung der Einheiten verzichtet.

$$F_1 = \frac{X_c - X_t}{X_c X_t} = \frac{1448 - 1448}{1448 \cdot 1448} = 0\text{N/mm}^2 \quad (4.2)$$

$$F_2 = \frac{Y_c - Y_t}{Y_c Y_t} = \frac{206.85 - 51.71}{206.85 \cdot 51.71} = \frac{1,450}{10^2\text{N/mm}^2} \quad (4.3)$$

$$F_{11} = \frac{1}{X_c X_t} = \frac{1}{1448 \cdot 1448} = \frac{4,769}{10^7\text{N}^2/\text{mm}^4} \quad (4.4)$$

$$F_{22} = \frac{1}{Y_c Y_t} = \frac{1}{206.85 \cdot 51.71} = \frac{9,349}{10^5\text{N}^2/\text{mm}^4} \quad (4.5)$$

$$F_{12} = \frac{1}{(X_c X_t Y_c Y_t)^2} = \frac{1}{(1448 \cdot 1448 \cdot 206.85 \cdot 51.71)^{1/2}} = \frac{6,677}{10^6\text{N}^4/\text{mm}^4} \quad (4.6)$$

$$F_{66} = \frac{1}{S^2} = \frac{1}{(90,08)^2} = \frac{1,232}{10^4\text{N}^2/\text{mm}^4} \quad (4.7)$$

$$\begin{aligned} \Phi = & 0 \cdot -23,26 + 1,450 \cdot 10^{-2} \cdot 5,916 + 4,769 \cdot 10^{-7} \cdot (-23,26)^2 \\ & + 9,349 \cdot 10^{-5} \cdot (5,916)^2 + 2 \cdot 6,677 \cdot 10^{-6} \cdot -23,26 \cdot 5,916 \\ & + 1,232 \cdot 10^{-4} \cdot -0,431^2 = 0,0875 \end{aligned} \quad (4.8)$$

Der von Nastran ausgegebene Index 0,0894 weicht von dem Ergebnis der Formel 4.8 um ungefähr 2 % ab, was als hinreichend genau erachtet wird.

Im Normalfall ist eine ausführliche Konvergenzstudie der Vernetzung durchzuführen, um festzustellen, ab welcher Netzdichte die Berechnungsergebnisse netzunabhängig sind. Erst wenn dies der Fall ist, sind die Ergebnisse verlässlich. Da die Elemententeilung sich an der **Hondas** orientiert, wird an dieser Stelle jedoch lediglich geprüft, ob diese bereits für unabhängige Ergebnisse ausreicht. Hierzu wird das Modell ( $[0^\circ \pm 45^\circ 90^\circ]_s$ -Laminat) mit vier weiteren FE-Netzen berechnet, wobei die Verteilung der Zellen gleich bleibt. Die Zellanzahl und die Ergebnisse sind in Tabelle 4.4 aufgeführt.

Tabelle 4.4.: Konvergenzstudie

Anzahl Zellen	$\Phi_{max}$
196	0,0853
<b>400</b>	<b>0,0894</b>
900	0,0925
1600	0,0941
2500	0,0951

Die Studie zeigt, dass, ausgehend von einem tatsächlichen maximalen Wert von 0,0951, die Ergebnisse der gewählten Vernetzung etwa 6 % zu niedrig sind. Eine höhere Zelldichte würde diese Diskrepanz verringern, jedoch mit einer höheren Rechenzeit einhergehen. Da die Abweichung keinen Einfluss auf die Vergleichbarkeit der Optimierungsergebnisse untereinander hat, wird die Vernetzung **Hondas** beibehalten. Allerdings darf dieser Umstand bei der Weiterverarbeitung der Ergebnisse nicht vernachlässigt werden, insbesondere da diese nicht konservativer Natur sind.

## 5. Bézier-Tool

Das Bézier-Tool stellt den Hauptbestandteil der Schnittstelle zwischen dem FE-Modell und dem UFO-Optimierungstool dar. Nötig wird das Hilfsprogramm durch den im vorangegangenen Kapitel zur FE-Modellierung der Faserverläufe gewählten Ansatz, dass jede Zelle der Struktur mit einem eigenen Winkel definiert wird. Da es bei großen Strukturen zu Tausenden Zellen kommen kann, würde eine Äquivalenz von Zell- und Entwurfsvariablenanzahl hohe Rechenzeiten verursachen. Daher ist der Zweck des Bézier-Tools, aus wenigen Variablen eine deutlich höhere Anzahl an Entwurfsvariablen zu generieren.

Nachfolgend werden zunächst alle relevanten Überlegungen und Informationen zum Tool dargelegt, die Erläuterung der programmiertechnischen Umsetzung erfolgt anschließend in Unterkapitel 5.6. Das letzte Unterkapitel befasst sich mit auftretenden Problematiken.

### 5.1. Ansatz Variablenerzeugung

Die Designvariablen im Optimierungsprozess sind Winkel und Krümmungswerte. Diese können theoretisch vom Algorithmus skalar variiert werden. Allerdings erfordert es, wie bereits erwähnt, hohe Rechenkapazitäten, alle Winkel als Variablen zu definieren und einzeln zu optimieren. Speziell bei Strukturen mit vielen Zellen oder begrenzten Rechenkapazitäten wäre diese Strategie unwirtschaftlich. Daher ist ein geeigneter Ansatz zur indirekten Bestimmung der Entwurfsvariablen sinnvoll. Dieser muss aus möglichst wenigen Variablen, im späteren Verlauf auch *Ersatzvariablen* genannt, viele Parameter generieren, die variabel und unabhängig voneinander sind. Um für den Einsatz in einem Optimierungsprozess geeignet zu sein, muss er zudem programmiertechnisch umsetzbar sein und geringe Rechenzeiten benötigen. Da es sich bei den Entwurfsvariablen um Winkel handelt, bietet es sich an, diese Werte aus Kurvenverläufen abzuleiten. Dies bringt im weiteren Verlauf zusätzlich die Möglichkeit, verhältnismäßig einfach einen Index für die Krümmung zu ermitteln (s. Kapitel 5.4.2). In der Literaturrecherche (s. Kapitel 2) sind bereits einige Methoden zur Kurvendefinition genannt worden. Der Umsetzungsansatz ist je nach angestrebter Flexibilität

zu wählen. Können oder sollen die Faserverläufe eine gewisse Ähnlichkeit aufweisen, ist die Verwendung einer Referenzkurve sowie von ihr abhängiger Kurven sinnvoll. Dies ist aus Sicht der Fertigung vorteilhaft, da auf diese Weise Probleme wie Unebenheiten, Harznester, etc. vermieden werden können. Allerdings ist dadurch der Optimierungsfreiraum deutlich eingeschränkt. Speziell bei lokalen Struktureinflüssen, wie z.B. Löchern, nutzt dieser Ansatz das Potential krummliniger Faserverläufe nicht aus. Sollen die Kurven daher möglichst unabhängig voneinander sein, können diese bspw. aus einer dreidimensionalen Hilfsfläche bestimmt werden. Nachteile solcher Ansätze sind in Kapitel 5.7.1 beschrieben.

Teil der Aufgabenstellung ist der Ansatz, dass die Kurvenverläufe aus der Form einer Bézier-Fläche resultieren sollen. Hierzu wird diese auf definierten Höhen mit horizontalen Ebenen verschnitten und die entstehenden Schnittkurven betrachtet. Wie bereits in Kapitel 3.5.1 beschrieben, ermöglichen Bézier-Flächen bereits mit wenigen Parametern (16-36) komplexe, dreidimensionale Formen zu beschreiben. Somit können entsprechend variable Kurven entstehen, die aufgrund ihres dreidimensionalen Ursprungs eine große Unabhängigkeit von Nebenpunkten aufweisen. Zudem können aus der Fläche theoretisch unendlich viele Winkel entnommen werden, da diese beliebig oft geschnitten werden kann. Die zu optimierenden Variablen bei diesem Ansatz sind die Koordinaten der Kontrollpunkte der Bézier-Fläche. Die Variablenzahl ist daher nur von der zur Flächenerzeugung gewählten Anzahl Kontrollpunkte abhängig und somit konstant. Die notwendigen Höhen (z-Koordinaten), an denen die Fläche geschnitten werden soll, leiten sich aus der Position der Abfragepunkte ab. Der schematische Verlauf ist in Abbildung 5.1 veranschaulicht. Wie die Umsetzung im Detail funktioniert, wird im folgenden Kapitel erläutert.

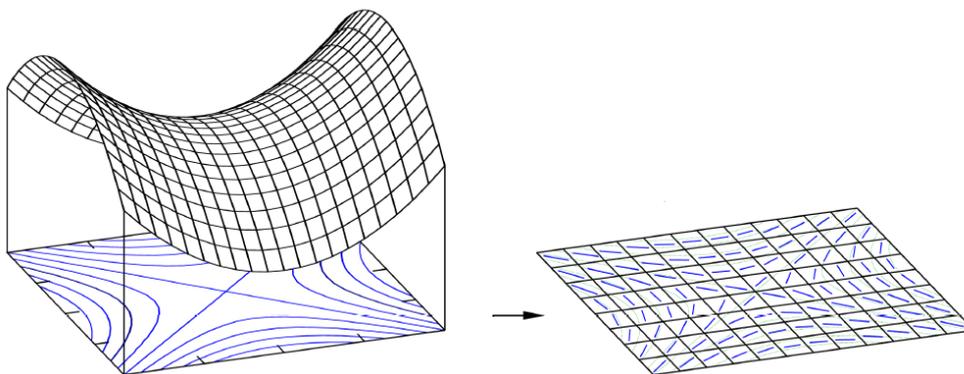


Abbildung 5.1.: Schematische Darstellung der Winkelbestimmung

Wichtig bei diesem Ansatz ist, zu verstehen, dass die erzeugten Kurven nicht die tatsächlichen Faserverläufe darstellen. Sie dienen lediglich dazu, den Winkel und die Krümmung des Faserverlaufes an den Zellmittelpunkten zu bestimmen.

### 5.2. Realisierung Kurvenerzeugung

Im vorangegangenen Kapitel wurde der Ansatz gewählt, die Kurven durch Verschneiden einer dreidimensionalen Fläche mit Ebenen auf unterschiedlichen Höhen zu erzeugen. Essenziell hierfür sind zum einen die Fläche selbst, zum anderen die Position des zu untersuchenden Punktes.

Zunächst wird die Fläche erzeugt. Im Gegensatz zur Methode **Hondas** geschieht dies nicht mit einer kubischen Polynomfunktion, sondern mit einer Bézier-Fläche. Diese benötigt zwar mehr beschreibende Parameter, bietet aber variabelere Formmöglichkeiten. Bei einfachen Untersuchungen ist der Rechenaufwand daher zwar größer, aber der Ansatz ist insgesamt flexibler einsetzbar. Die Grundlagen und die Mathematik der Bézier-Fläche wurden in Kapitel 3.5.1 dargestellt. Um die Fläche in Einklang mit der zu untersuchenden Struktur zu bringen, gibt es zwei Möglichkeiten. Einerseits kann die Bézier-Fläche in ihren üblichen, normierten Dimensionen von 0 bis 1 erzeugt und anschließend auf die Struktur projiziert werden. Hier können allerdings Probleme bei der Projektion, speziell bei der Äquidistanz der Werte, auftreten. Daher wird die zweite Variante genutzt. Hierbei wird vorab die Geometrie der Struktur abgefragt und die Bézier-Fläche direkt mit deren Koordinaten berechnet. Die Eckpunkte werden auf die Strukturecken gelegt und die restlichen Kontrollpunkte gleichmäßig dazwischen verteilt. Alternativ können die restlichen Kontrollpunkte auch manuell definiert werden, hierauf wird später bei der programmiertechnischen Umsetzung näher eingegangen (siehe Kapitel 5.6). Somit sind die Positionen der Kontrollpunkte in x- und y-Richtung festgelegt. Ihre jeweiligen z-Koordinaten bilden die Variablen, durch die die Form der Bézier-Fläche gesteuert wird. Bei 16 Kontrollpunkten entspricht das 16 Variablen.

Die zweite Voraussetzung ist ein Punkt innerhalb der Struktur. Dies werden später sämtliche Zellmittelpunkte (ZMP) des FE-Netzes sein. Da die Bézier-Fläche über die gesamte Struktur aufgespannt ist, sind die x- und y-Koordinaten des zu untersuchenden Punktes identisch mit denen der Bézier-Fläche. An ebendiesen Koordinaten kann nun die zugehörige z-Koordinate abgefragt werden. Wird nun eine zur xy-Ebene parallele Ebene an der ermittelten Höhe aufgespannt und diese mit der Bézier-Fläche verschnitten, entsteht die gesuchte Kurve

(im weiteren Verlauf *Höhenschnittlinie* genannt). Sie verläuft durch den gefragten Punkt und ihr Kurvenverlauf kann hinsichtlich des Winkels und der Krümmung untersucht werden.

Dieser Vorgang muss im späteren Optimierungsprozess für alle Zellmittelpunkte wiederholt werden. Auf diese Weise entstehen ebenso viele unterschiedliche Kurven, wie ZMP vorhanden sind. Die Bézier-Fläche selbst bleibt dabei unverändert. Auf diese Weise können aus den 16 Variablen der Bézier-Fläche theoretisch unendlich viele Kurvenverläufe erzeugt werden. Werden die z-Koordinaten der Kontrollpunkte variiert, verändert sich die Form der Bézier-Fläche und damit auch die Kurvenverläufe der Höhenschnittlinien an den ZMP. So entstehen bei jeder Iteration des Optimierungsalgorithmus neue Winkel und Krümmungen.

### 5.3. Anwendung Béziertool

Das Hauptziel des Bézier-Tools ist seine Einbindung in die Optimierungsprozedur. Hierbei erfolgt die Anwendung in Form eines Funktionsaufrufes im Optimierungstool, welches in Kapitel 6 beschrieben wird. Es werden die notwendigen Eingangsdaten per Funktionsaufruf übergeben und nach dem Durchlauf der Berechnung die geforderten Winkel sowie der Wert der durchschnittlichen Krümmung zurückgegeben. Alternativ kann das Tool auch unabhängig von der Optimierungsprozedur genutzt werden, um für bestimmte Kontrollpunkt-Koordinaten einen Überblick über die Bézier-Fläche, die Höhenschnittlinien und die Winkel zu erhalten. Hier werden alle Parameter manuell eingetragen und es erfolgt ausschließlich eine grafische Ausgabe. Zusätzlich können die Winkel und Krümmungswerte tabellarisch ausgegeben werden. Weitere Informationen zu den Eingabeparametern werden in Kapitel 5.6.1 beschrieben.

### 5.4. Ausgabewerte

Das Bézier-Tool erzeugt einen Vektor mit den Winkeln für alle FE-Zellen sowie einen gemittelten Krümmungswert der Kurven. Die Länge des Vektors entspricht der Zellanzahl der untersuchten Struktur. In den folgenden Unterkapiteln wird näher auf diese Ausgabewerte eingegangen.

#### 5.4.1. Winkel

Hauptziel des Tools ist die Gewinnung der Faserwinkel an den Zellmittelpunkten. Sie sind die relevanten Entwurfsinformationen für die FE-Modellierung, da sie die mechanischen Eigenschaften der Struktur bestimmen. Die Winkelinformation wird gewonnen, indem an

der jeweiligen Höhenschnittlinie im zu untersuchenden Punkt die Steigung der Tangente berechnet wird (vgl. Kapitel 5.2). Die programmiertechnische Umsetzung ist in Kapitel 5.6.4 beschrieben.

### 5.4.2. Krümmung

Um Aussagen über die Herstellbarkeit der Faserverläufe treffen zu können, muss ein entsprechender Parameter gewählt werden, dessen Wert diesbezüglich aussagekräftige Informationen liefert. Hierzu sollen die Werte der Krümmung der einzelnen Faserverläufe herangezogen werden. Da es schwierig ist, aus den Winkeln eine Krümmung der später realen Faserverläufe zu bestimmen, muss ein Alternativwert gesucht werden. Um die Vergleichbarkeit zu **Honda** zu gewährleisten, wird an dieser Stelle das dort verwendete Konzept übernommen. Hierbei wird die Krümmung der Kurven, die zur Gewinnung der Winkel genutzt werden, betrachtet. Somit entspricht die gewählte Krümmung nicht der tatsächlichen Faserkrümmung. Diese Problematik wird in Kapitel 5.7.1 (Unterpunkt *Krümmungswert*) gesondert betrachtet. Zur Berechnung der Krümmungen werden die Koordinaten der Höhenschnittlinien im Bereich des zu untersuchenden Punktes in Verbindung mit Formel 3.20 genutzt. Damit mathematisch die Krümmungswerte aller Zellen in die Betrachtung einfließen, werden diese gemittelt. Umso größer dieser Mittelwert  $\bar{\kappa}$  ist, desto krummliniger sind die Faserverläufe. Als Ausgabewert wird nur der Mittelwert zurückgegeben. Dieser geht nicht direkt in die FE-Modellierung ein, ist er jedoch höher als zulässig, ist die Auslegung ungültig. Auf diese Weise hat die Krümmung einen indirekten Einfluss auf den Entwurf. Die programmiertechnische Umsetzung ist in Kapitel 5.6.4 beschrieben.

## 5.5. Geometrische Restriktionen

Als Maß für die Prozessierbarkeit wurde in Kapitel 5.4.2 der Mittelwert der Krümmung  $\bar{\kappa}$  eingeführt. Da der später zum Einsatz kommende Algorithmus keine Mehrzieloptimierung unterstützt, wird eine *restriktionsorientierte Transformation* durchgeführt und die Zielfunktion, die Krümmung zu minimieren, in eine Restriktion umgewandelt.  $\bar{\kappa}$  darf somit einen zu Beginn der Optimierung definierten Wert nicht überschreiten. Verstößt  $\bar{\kappa}$  gegen diese Restriktion, ist der Entwurf unzulässig, was der Optimierungsalgorithmus in der nächsten Iteration beachtet.

Aufgrund der Erstellung der Winkel mittels der Bézier-Fläche, kann es zu Winkeln kommen, die nicht harmonisch mit den umliegenden Winkeln verlaufen. Daher ist unter Umständen eine Restriktion der berechneten Winkel in bestimmten Bereichen sinnvoll. Solche Restriktionen

können im Tool auf zwei Weisen erfolgen. Einerseits kann ein Maximalwert für Winkel vorgegeben werden, auf den alle berechneten Winkel restringiert werden, wenn sie diesen überschreiten. Andererseits können zu große Winkelsprünge zwischen benachbarten Zellen begrenzt werden, um einen geglätteten Verlauf zu erzeugen.

## 5.6. Programmiertechnische Umsetzung des Bézier-Tools

Zur Nutzung der vorab in diesem Kapitel besprochenen Informationen müssen diese programmiertechnisch umgesetzt werden. Nachfolgend werden die hierzu notwendigen Schritte beschrieben und kurz auf die jeweils verwendeten Funktionen eingegangen. Zusätzlich ist auch das Script-File *Bezier\_Tool.m* selbst umfangreich kommentiert. Für detaillierte Informationen zu den unterschiedlichen Funktionen kann bei Bedarf **MathWorks 2016** herangezogen werden.

### 5.6.1. Eingabeparameter

Um die geforderten Winkel und den Wert der durchschnittlichen Krümmung berechnen zu können, müssen zunächst diverse Parameter an das Modul *Bezier\_Tool.m* übergeben werden. Der Funktionsaufruf ist in Formel 5.1 dargestellt.

$$\begin{aligned} & \text{function}[Winkel, DSKappa] = \\ & \text{Bezier\_Tool}(KP, ZMP, mKP, nKP, Feinheit, Anzahl\_Elemente, Elementdaten, \\ & \quad \text{Bereich\_mW}, max\_Winkel, \text{Bereich\_mdW}, max\_Deltawinkel) \end{aligned} \tag{5.1}$$

Welche Informationen hinter den übergebenen Parameter stehen, wird in Tabelle 5.1 spezifiziert. Bei der manuellen Verwendung müssen diese Eingaben direkt ins m-File eingetragen werden.

Tabelle 5.1.: Eingabeparameter des Bézier-Tools

Paramter	Spezifikation
KP	Koordinaten in x-,y- und z-Richtung der Bézier-Kontrollpunkte
ZMP	Koordinaten in x- und y-Richtung der Zellmittelpunkte
mKP	Anzahl der Bézier-Kontrollpunkte in vertikaler Richtung
nKP	Anzahl der Bézier-Kontrollpunkte in horizontaler Richtung
Feinheit	Wert der Unterteilung der Bézierfläche
Anzahl_Elemente	Anzahl der FE-Zellen, mit denen die Struktur vernetzt ist
Elementdaten	ID der Zelle sowie der zugehörigen PCOMP-ID und aller Zellknoten-IDs
Bereich_mW	Bereich, in dem der Winkel absolut begrenzt werden soll
max_Winkel	Maximal zulässiger Winkel im Bereich_mW
Bereich_mdW	Bereich, in dem Winkeldifferenzen benachbarter Zellen begrenzt werden sollen
max_Deltawinkel	Maximal zulässige Winkeldifferenz benachbarter Zellen im Bereich_mdW

### 5.6.2. Berechnung der Bézierfläche

Der erste Schritt der Umsetzung ist die Berechnung der Bézier-Fläche. Hierzu werden zunächst drei Matrizen  $P_X$ ,  $P_Y$  und  $P_Z$  mit den jeweiligen Raumkoordinaten der Kontrollpunkte erstellt. Wie diese Koordinaten entstehen, ist in Kapitel 6.2.3 Unterpunkt *KuMP* beschrieben. Zu späteren Darstellungszwecken werden die Mini- und Maximalwerte der Koordinaten Variablen zugewiesen. Bevor die eigentliche Berechnung der Winkel beginnt, wird anhand  $P_Z$  geprüft, ob alle z-Koordinaten der KP identisch sind. In diesem Fall liegt eine ebene Bézier-Fläche vor und zur Zeitersparnis werden alle Winkel automatisch auf  $0^\circ$  gesetzt. Ist dies nicht der Fall, wird die Berechnung initiiert. Die Zielformel zur Bestimmung der Koordinaten der Bézier-Fläche lautet in Anlehnung an Formel 3.13:

$$Q = u_{(1xm)} \cdot M_{(m xm)} \cdot P_{(m xn)} \cdot M'_{(n xn)} \cdot v_{(nx1)} \quad (5.2)$$

Ihr Entstehen und die zu verwendenden Matrizen sind in Kapitel 3.5.1 nachzulesen. In Klammern sind die jeweiligen Dimensionen der Matrizen angegeben, die abhängig von der verwendeten Anzahl der KP  $n$  und  $m$  in x- und y-Richtung ist. Die Variablen  $u$  und  $v$  werden

mittels Schleifen unter Verwendung der Feinheit berechnet. Die Feinheit bestimmt, in wie viele Abschnitte die Bézier-Fläche unterteilt wird. Umso höher ihr Wert ist, umso feiner aufgelöst ist das entstehende Raster. Die aus den Bernsteinpolynomen entstehenden Matrizen sind statisch hinterlegt und werden je nach Anzahl der KP ausgewählt. Die Berechnung der Flächenkoordinaten in allen drei Koordinatenrichtungen x, y und z erfolgt mittels Schleifen und den zugehörigen Kontrollpunkt-Matrizen  $P_X$ ,  $P_Y$  und  $P_Z$ . Da die Koordinaten in x- und y-Richtung ein gleichmäßiges Raster aufspannen, wiederholen sich die Werte in jeder Spalte bzw. Reihe. Zur Zeitersparnis werden daher zunächst per Schleife nur Vektoren erzeugt, die nachträglich per Multiplikation mit Einsvektoren der entsprechenden Länge zu Matrizen erweitert werden. Ein erläuterndes Beispiel ist in Formel 5.3 gegeben. Die z-Koordinaten sind in x- und y-Richtung unabhängig und werden daher mit zwei Schleifen in beide Richtungen berechnet. Die Koordinaten werden in den Matrizen  $Q_X$ ,  $Q_Y$  und  $Q_Z$  gespeichert. Damit liegen alle Koordinaten vor und die Berechnung der Bézier-Fläche ist abgeschlossen.

$$Q_X = \text{ones}(i, 1) \cdot q_x(1, i) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{bmatrix} \quad (5.3)$$

Theoretisch besteht der alternative Berechnungsweg, Punkte der Bézier-Fläche direkt zu ermitteln, statt eine per Feinheit gerasterte Bézier-Fläche zu erstellen. Dies würde zwar genauere Werte ergeben, allerdings bietet diese Methode nicht die Möglichkeit, Höhenschnittlinien zu erzeugen.

### 5.6.3. Erzeugung der Höhenschnittlinien

Der nächste Schritt ist die Erzeugung der Höhenschnittlinien (HSL). Je nach Form der Fläche verlaufen diese theoretisch über die gesamte Geometrie. Zur Zeitersparnis werden die HSL jedoch nur in einem bestimmten Bereich um den jeweiligen Zellmittelpunkt erzeugt. Die Prozedur zur Berechnung des Winkels und der Krümmung einer Zelle wird nachfolgend beschrieben. Mittels einer Schleife wird diese für alle Zellen wiederholt. Programmiertechnisch umfasst diese Schleife auch die anschließende Winkelberechnung (siehe 5.6.4).

Zunächst muss der Bereich, in dem die HSL erzeugt werden soll, festgelegt werden. Hierzu müssen die dem jeweiligen ZMP nächsten Bézier-Flächenkoordinaten ermittelt werden. Dies geschieht über while-Schleifen in x- und y-Richtung. Nachdem der Mittelpunkt des Bereiches definiert ist, muss die Größe ebendieses bestimmt werden. Der Gesamtbereich ist über den

## 5. Bézier-Tool

Divisor 100 an den Wert der Feinheit gekoppelt. Um zu kleine Bereiche zu vermeiden ist ein Minimum von 10 gesetzt, wodurch in x- und y-Richtung jeweils mindestens fünf Koordinaten unter und fünf über dem Mittelpunkt betrachtet werden. Um abzusichern, dass der gewählte Bereich nicht über das berechnete Bézierraster (die Matrizen  $Q_X$  und  $Q_Y$ ) hinausreicht, erfolgt eine diesbezügliche Prüfung. Liegt ein Bereich zu nah an einer Flächengrenze, so wird der Bereich in diese Richtung entsprechend reduziert und die Differenz der entgegengesetzten Richtung hinzuaddiert. Auf diese Weise bleibt die gewünschte Gesamtgröße des Bereichs erhalten. Die Werte im definierten Bereich werden anschließend aus den  $Q_X$ ,  $Q_Y$  und  $Q_Z$  Matrizen entnommen und in die Matrizen  $S_X$ ,  $S_Y$  und  $S_Z$  geschrieben.

Zur Erzeugung einer HSL muss die z-Koordinate des ZMP bestimmt werden. Um diese möglichst exakt zu bestimmen, wird zunächst die Fläche  $F$  im ausgewählten Bereich mit der Funktion *scatteredInterpolant* interpoliert. Anschließend kann mit den x- und y-Koordinaten des ZMP die z-Koordinate abgefragt werden. Mit Hilfe der Funktion *contours* wird in dem vorab definierten Bereich der Schnittverlauf an ebendieser Höhe erzeugt. Die Funktion gibt die Koordinaten der Konturlinien in der Matrix  $C$  aus und erzeugt im Gegensatz zur Funktion *contour* keine zusätzliche visuelle Darstellung. Die erzeugten Koordinaten werden für eine einfachere Weiterverarbeitung in zwei getrennte Matrizen,  $CX$  und  $CY$ , geschrieben. Zusätzlich wird die jeweilige Anzahl der Koordinaten der HSL in  $Clen$  gespeichert, da diese von Zelle zu Zelle variieren kann. Bei der Ausgabematrix der *contour*-Funktion ist zu beachten, dass je nach Form der Bézier-Fläche in dem betrachteten Bereich mehrere HSL entstehen können. Zu Beginn jeder HSL, auch der ersten, steht ein Wertepaar, welches keine Koordinaten enthält. Der Wert in der ersten Zeile gibt die jeweilige Schnitthöhe an, der Wert in der zweiten Zeile die Anzahl der einzelnen Punkte der HSL. In Bild 5.2 ist das Format anhand eines Beispiels veranschaulicht. Diese Eigenschaft muss bei der weiteren Verwendung der Daten berücksichtigt werden.

```

C =

Columns 1 through 8
    -0.2000    -0.5504         0         0.5506         0    -2.9992         0         3.0000
     3.0000   -3.0000   -2.8900   -3.0000     3.0000   -3.0000   -2.4006   -2.9995

Columns 9 through 16
         0   -3.0000   -2.8924   -3.0000     0.2000   -0.9992   -2.3027         0
     3.0000     2.9974         0   -2.9945     5.0000     3.0000         0   -1.9112
  
```

Abbildung 5.2.: Format der Höhenschnittlinien-Koordinaten [MathWorks 2016]

### 5.6.4. Berechnung der Winkel und Krümmung

Durch die Ausgabe der Kurvenkoordinaten der HSL, gespeichert in den Matrizen  $CX$  und  $CY$ , ist theoretisch direkt eine analytische Berechnung der Winkel und Krümmungen möglich. Aufgrund der verhältnismäßig groben Verteilung der Koordinaten werden jedoch aus diesen vorab Kurven interpoliert, um präzisere Werte zur Berechnung der Krümmung und Winkel zu erhalten. Zur Approximation des HSL-Verlaufs soll die Funktion *spline* verwendet werden, die einen kubischen Spline interpoliert. Hierzu werden sowohl Punkte der HSL als auch ein Vektor mit x-Koordinaten, zu denen die entsprechenden y-Werte approximiert werden sollen, benötigt. Für die Punkte der HSL wird ein Bereich um den dem realen ZMP am nächsten liegenden Punkt der HSL ausgewählt. Hierbei wird geprüft und sichergestellt, dass nur tatsächliche Koordinaten der Kurven verwendet werden. Der Vektor  $x_{interp}$  wird mit x-Koordinaten definiert, zu denen die entsprechenden y-Koordinaten gesucht werden. Da zur Bestimmung der Krümmung drei Punkte nötig sind, wird die x-Koordinate des Mittelpunkts sowie zwei Nebepunkte gewählt, die mit einem Delta  $\Delta$  berechnet werden. Im Sinne einer infinitesimalen Ableitung wird  $\Delta < 10^{-4}$  gewählt. Anhand dieser Daten interpoliert die *spline*-Funktion anschließend die drei gesuchten y-Koordinaten und speichert diese in Vektor  $y_{interp}$ .

Durch Umstellung von Formel 3.19 und Verwendung des zentralen Differenzenquotienten (Formel 5.4) ergibt sich zur Berechnung des Winkels Formel 5.5. Der Index  $i$  ist die Laufvariable der Schleife und entspricht gleichzeitig der Identifikationsnummer der betrachteten FE-Zelle.

$$\frac{\Delta y}{\Delta x} = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} \quad (5.4)$$

$$Winkel(i) = \tan^{-1} \frac{Y_{diff}}{X_{diff}} = \tan^{-1} \frac{y_{interp}(3) - y_{interp}(1)}{2 \cdot \Delta} \quad (5.5)$$

Für die Berechnung der Krümmung müssen die Funktionswerte von  $x_{interp}$  und  $y_{interp}$  zweimal differenziert werden. Dies geschieht bei Matlab mit der Funktion *diff* und wird in den Formeln 5.6 und 5.7 exemplarisch für die x-Werte veranschaulicht. Die y-Differenzen werden analog dazu berechnet. Anschließend werden die jeweiligen Differenzen in Formel 5.8 eingesetzt und so der Wert der Krümmung errechnet.

$$x_{diff1} = diff(x_{interp}) = \begin{bmatrix} x_{interp}(2) - x_{interp}(1) \\ x_{interp}(3) - x_{interp}(2) \end{bmatrix} \quad (5.6)$$

$$x_{diff2} = diff(x_{diff1}) = x_{diff1}(2) - x_{diff1}(1) \quad (5.7)$$

$$Kappa(i) = \frac{(x_{diff1}(2) \cdot y_{diff2} - y_{diff1}(2) \cdot x_{diff2})}{[(x_{diff1}(2))^2 + (y_{diff1}(2))^2]^{3/2}} \quad (5.8)$$

An dieser Stelle endet die in Kapitel 5.6.3 erwähnte Prozedur zur Bestimmung des Winkels und der Krümmung. Nachdem dieser Ablauf durch die Schleife für jede Zelle der Struktur wiederholt wurde, entspricht die Länge der Vektoren *Winkel* und *Kappa* der Anzahl der FE-Zellen. Abschließend wird aus allen Einzelkrümmungen nach Formel 5.9 die Durchschnittskrümmung *dsk* berechnet.

$$dsk = \frac{1}{AnzahlZellen} \cdot \sum_{i=1}^n Kappa(i) \quad (5.9)$$

### 5.6.5. Methoden zur Begrenzung/Winkelrestriktionen

Um die in Kapitel 5.5 beschriebenen Restriktionen der Winkel und Winkelsprünge korrekt in den Optimierungsprozess einzubinden, müssen die Winkelinformationen berechnet und ein Rückgabeparameter an den Algorithmus übergeben werden. Anhand der Information prüft dieser die Gültigkeit des Entwurfs. Da dies aber bei vielen Entwürfen zur Ungültigkeit und somit zu einem starken Anstieg der Rechendauer führen würde, werden die Restriktionen auf andere Weise umgesetzt.

Wenn für die Eingangsparameter *maxWinkel* und/oder *maxDeltawinkel* Werte hinterlegt sind, werden die berechneten Winkel entsprechenden Winkelprüfungen unterzogen. Überschreiten die Winkel die zulässigen Werte, erfolgt eine direkte Anpassung der berechneten Winkel. Sind keine maximalen Werte gegeben, werden entsprechend keine Winkel-Restriktionen angewendet.

#### **Winkelbegrenzung**

Die *Winkelbegrenzung* wird mittels der gleichnamigen Funktion durchgeführt. Diese prüft die übergebenen Winkel in Bezug auf den vorgegebenen Maximalwert und passt sie gegebenenfalls dementsprechend an. Der Vorgang erfolgt entweder in der gesamten Struktur oder nur im definierten Bereich.

*function [Winkel] = Winkelbegrenzung (Winkel,Bereich\_mW,max\_Winkel)*

Im ersten Schritt werden die übergebenen Daten in die entsprechenden Formate umgewandelt. Ist kein Zellbereich definiert, wird die Prüfung auf alle Zellen angewandt. Mittels einer Schleife werden alle Winkel mit dem Maximalwinkel verglichen. Sind sie größer als zulässig, wird ihnen der Maximalwert zugewiesen. Dies gilt sowohl für positive als auch negative Winkel. Der überarbeitete Vektor der Winkel wird nach Vollendung der Prüfung dem aufrufenden Script zurückgegeben.

### **Winkelangleich**

Das Modul *Winkelangleich* dient zur Verlaufsglättung, indem die Winkel benachbarter Zellen betrachtet und zu große Abweichungen reguliert werden. Dies erfolgt in einem frei definierbaren Zellbereich auf ein zu wählendes Winkeldelta.

*function [Winkel] = Winkelangleich (Winkel,Elementdaten,Bereich\_mdW,max\_Winkeldelta)*

Zunächst wird wie auch bei der *Winkelbegrenzung* der zu betrachtende Bereich aufbereitet. Als Grundlage der Angleichfunktion wird eine Matrix erstellt, in der zu allen Zellknoten die zugehörigen Zellen aufgelistet sind. Wird nun eine Zelle betrachtet, können anhand ihrer Zellknoten aus dieser Matrix alle Zellen bestimmt werden, die an die Knoten und somit an die betrachtete Zelle angrenzen. Die Winkel dieser Nachbarzellen werden gemittelt und mit dem Winkel der aktuellen Zelle verglichen. Überschreitet die Differenz das vorgegebene Delta, wird dem betrachteten Winkel der Mittelwert zugewiesen. Um hierbei zu verhindern, dass stark abweichende Winkel in noch nicht untersuchten Zellen die Betrachtung des aktuellen Winkels verfälschen, wird vor der Mittelwertbildung geprüft, ob alle Nachbarwinkel innerhalb eines bestimmten Wertebereichs liegen. Der Vorgang wird durch eine Schleife solange wiederholt, bis keine Winkel mehr angeglichen werden.

### **5.6.6. Darstellung**

Der letzte Abschnitt des Moduls befasst sich mit der visuellen Darstellung der Ergebnisse. Standardmäßig sind hier nur zwei Ausgaben aktiv, die restlichen sind auskommentiert. Alle Diagramme starten mit dem Löschen des alten Diagramms (falls vorhanden), der Definition des Diagrammtitels und der -achsen sowie bei dreidimensionaler Darstellung der Festlegung des *Viewpoints*. Das erste Diagramm stellt die berechnete Bézier-Fläche visuell dar. Dies wird mit der Funktion *surf* realisiert, die den Verlauf farbige dargestellt,

wodurch die Höhenunterschiede deutlicher erkennbar sind. An die Funktion werden die Flächenkoordinaten-Matrizen  $Q_X$ ,  $Q_Y$  und  $Q_Z$  übergeben. Dieser Darstellung können keine Informationen bezüglich der Winkel oder Faserverläufe entnommen werden, aber sie zeigt, in welchen Bereichen der Algorithmus derzeit Veränderungen durchführt. Die zweite Ausgabe gibt die Anordnung der Winkel im Bereich des Loches wieder. Um nicht mit symbolischen Variablen rechnen zu müssen, werden zur Darstellung jeweils Linien mit dem Zellmittelpunkt und dem entsprechenden Winkel berechnet. Zusätzlich wird noch der zur y-Achse symmetrische Winkel gestrichelt in magenta dargestellt. Besonderheit bei beiden Ausgaben ist die Prüfung, ob eine ebene Bézier-Fläche vorliegt. In diesem Fall wird in beiden Diagrammen nur ein Textfeld mit ebendieser Information eingeblendet.

Zusätzlich sind standardmäßig deaktivierte Ausgaben programmiert, die analog zu den beschriebenen aufgebaut sind. Diese können je nach Bedarf aktiviert werden. Hierbei handelt es sich um eine Darstellung aller Höhengchnittlinien sowie eine Darstellung der Winkel im hinteren Bereich der Lochplatte.

### 5.7. Problematiken

Sowohl der Ansatz der indirekten Variablenerzeugung als auch die programmiertechnische Umsetzung führt zu verschiedenen Problematiken. Da diese nicht zu vernachlässigen sind, werden sie in diesem Kapitel gesondert diskutiert. Lösungs- oder Verbesserungsvorschläge werden später in Kapitel 8.2 besprochen.

#### 5.7.1. Probleme des Ansatzes

##### Bézier-Fläche

Grundsätzlich erzeugt der Ansatz der Bézier-Fläche zur indirekten Gewinnung der Winkel sehr freie Faserverläufe. Allerdings birgt das die Gefahr, dass diese entweder nicht sinnvoll oder nicht herstellbar sind. Ersteres Problem resultiert daraus, dass die Form der Bézier-Fläche häufig eine Anhöhe im mittleren Bereich aufweist, wodurch in diesem Gebiet konfuse Winkel entstehen. Dieses Phänomen wird später in Kapitel 7 betrachtet. Dieses Phänomen lässt sich bei Verwendung einer einzelnen Bézier-Fläche nicht vermeiden, der Algorithmus versucht aber automatisch, diesen Bereich in einer wenig belasteten Region zu positionieren. Auf die fertigungstechnischen Schwierigkeiten wurde bereits in Kapitel 4.1.1 bei der Überlegung, wie krummlinige Faserverläufe in der FEA umgesetzt werden können, hingewiesen. Die Problematik entsteht allgemein bei der Modellierung von krummlinigen Faserverläufen ohne

Kontinuitätsbedingungen zwischen den einzelnen Zellen. Da somit kein stetiger Faserverlauf durch Nachbarzellen garantiert ist, können Verzweigungen der Pfade oder endende Fasern resultieren. In der Fertigung führt dies zu Überlappungen von Fasern oder Lücken in den Lagen. Da die Einführung von Kontinuitätsbedingungen zwischen benachbarten FE-Zellen die Entwurfsvfreiheit aber stark einschränken würde, stellt dies keine sinnvolle Option dar. Konservativere Ansätze, bspw. die Nutzung einer Referenzkurve und zu ihr paralleler Kurven, böten hier Vorteile, jedoch würden diese das Potential der Spannungsoptimierung einschränken.

### **Ersatzvariablen**

Ein weiterer Nachteil beim Einsatz von Methoden zur indirekten Bestimmung der Entwurfsvariablen entsteht durch die verwendeten Ersatzvariablen. Der direkte Einfluss des Optimierungsalgorithmus auf die Auslegung wird durch ihre Verwendung gemindert, da nicht mehr jede Zelle explizit angesteuert werden kann, sondern Veränderungen der Ersatzvariablen mehrere Entwurfsvariablen gleichzeitig beeinflussen. Dies macht es für den Algorithmus schwieriger, den durchgeführten Parameteränderungen positive oder negative Auswirkungen zuzuordnen, wodurch das Konvergieren zum Optimum mehr Rechenzeit beansprucht. Hinzu kommt, dass ein nicht-linearer Zusammenhang zwischen Ersatz- und Entwurfsvariablen besteht, wodurch eine geringe Änderung eines Bézier-Kontrollpunktes zu großen oder kleinen Winkeländerungen führen kann. Dieser Umstand erschwert die Konvergenz zusätzlich.

### **Krümmungswert**

Bei der Wahl des Krümmungswertes als Indikator für die Prozessierbarkeit gibt es zwei Probleme. Einerseits werden bereits durch den Ansatz, die Krümmung der Höhengschnittlinie zu betrachten, keine realen Eigenschaften der Faserverläufe betrachtet. Daher ist der Krümmungswert mit Vorsicht und nur als richtungsweisender Hilfwert zu betrachten. Hinzu kommt, dass durch die Mittelwertbildung einzelne Krümmungen deutlich über dem Durchschnittswert liegen können. Dieser Umstand mindert die Aussagekraft des Krümmungswertes hinsichtlich der Fertigbarkeit zusätzlich. Dennoch wird der Wert für erste Untersuchungen als Richtwert herangezogen werden, da eine gewisse Korrelation zwischen stark gekrümmten Höhengschnittlinien und den resultierenden Faserverläufen besteht. Allerdings ist bei der Auswertung der Ergebnisse die Aussagekraft des Krümmungswertes zu prüfen.

### Winkelprüfungen

Die alternativ verwendete Methode der Winkel-Restriktionen ermöglicht zwar eine direktere und schnellere Umsetzung der Begrenzung, aber sie gibt Änderungen vor, die der Optimierungsalgorithmus nicht kennt. Diese nachträglichen Eingriffe haben eine erhebliche Auswirkung auf den Entwurf und dessen Ergebnis, allerdings ordnet der Optimierungsalgorithmus dieses den ursprünglichen Werten zu. Dadurch kann es passieren, dass der Algorithmus Variablenänderungen falsch beurteilt und der Optimierungsverlauf in eine falsche Richtung tendiert. Aus diesem Grund sind die Restriktionen in ihrer aktuellen Form nur optionale Ergänzungen und sollten mit Bedacht verwendet werden.

### 5.7.2. Probleme der programmiertechnischen Umsetzung

#### Feinheit

Die Feinheit hat einen großen Einfluss auf die Rechendauer, da sie quadratisch ((Feinheit · Länge) · (Feinheit · Breite)) in die Berechnung der Bézier-Fläche eingeht. Daher ist das Ziel, mit möglichst geringen Feinheits-Werten zu arbeiten, was allerdings ein grobes Flächenraster zur Folge hat. Diesem Problem wird durch Interpolation mit der Funktion *scatteredInterpolant* zwar entgegengewirkt, es beeinflusst die Genauigkeit der Ergebnisse aber dennoch. Feinheits-Werte im Bereich 100-800 erwiesen sich während der Untersuchungen als sinnvolle Einstellung. Größere Werte beanspruchen deutlich höhere Rechenzeiten und liefern keine besseren Ergebnisse. Der Einfluss der Feinheit auf die entstehenden Winkel wirkt sich des Weiteren auf die Konvergenz der Optimierungsprozedur aus. Bei unterschiedlichen Werten der Feinheit kann es zu schneller, langsamer oder gar keiner Konvergenz kommen. Diese Abhängigkeit ist nachteilhaft, da sie zusätzliche Optimierungsdurchgänge erfordert, aber nicht zu umgehen.

#### Interpolation

Die Interpolation der Kurvenwerte hat zur Folge, dass keine exakten 90°-Winkel erzeugt werden können, da bei diesen keine unterschiedlichen x-Koordinaten vorhanden sind, die zur Interpolation notwendig sind. Allerdings treten exakte 90°-Winkel nur bei einem entsprechenden Startentwurf auf, weswegen der Programmieraufwand, eine Prüfung für diesen Fall zu schaffen, gespart wird.

## 6. Optimierungstool

Mit dem UFO-Tool liegt bereits eine Optimierungsumgebung für MATLAB vor, auf die zurückgegriffen werden kann. Nun muss ein Optimierungstool programmiert werden, welches auf Grundlage des UFO-Tools eine Optimierung der Faserverläufe in einem FE-Modell durchführen kann. Ein Hauptbestandteil davon ist das in Kapitel 5 beschriebene Bezier-Tool, das zur Generierung der Entwurfsvariablen für jede Zelle der Struktur dient. Dieses muss zusammen mit diversen anderen Funktionen in das Optimierungstool eingebunden werden, um eine automatische Strukturoptimierung zu ermöglichen.

### 6.1. Ansatz der Optimierung

Das Ziel der Optimierung wurde in Kapitel 1.2 ausführlich beschrieben. Bevor die Optimierung programmiertechnisch umgesetzt werden kann, müssen jedoch gewisse Vorüberlegungen getroffen werden. Vier wesentliche Eckpfeiler des Optimierungsprozesses sind der *Lösungsalgorithmus*, die *Zielfunktion*, die *Restriktionen* und die *Entwurfsvariablen*. Diese Faktoren werden nachfolgend hinsichtlich des Optimierungsziels diskutiert und kurz die Einbindung des UFO-Tools erläutert.

#### 6.1.1. Einbindung des UFO-Tools

Da mit dem UFO-Tool bereits eine Grundlage zur Optimierung mit Matlab vorliegt, soll auf diese aufgebaut werden. Seine Funktionsweise wurde in Kapitel 3.2.9 beschrieben. Anhand dieser muss geklärt werden, wie das Programm am sinnvollsten und variabelsten in den Prozess eingebunden werden kann. Um in den Quellcode des Tool selbst nicht eingreifen zu müssen, wird sich dazu entschieden, die config-Datei als Schnittstelle zu wählen. Aus dieser bezieht das UFO-Tool sämtliche problembezogenen Daten und Informationen. Für nähere Informationen zum UFO-Tool und dessen Arbeitsweise siehe **Geisberger 2013**.

### 6.1.2. Entwurfsvariablen

Wie in den vorangegangenen Kapiteln beschrieben, sind die Entwurfsvariablen durch den gewählten Ansatz nicht die Winkel direkt, sondern die z-Koordinaten der Bezier-Kontrollpunkte. Die Variablenanzahl ist somit proportional zu der Anzahl der Kontrollpunkte. Um die Variabilität der Bezierfläche weiter zu erhöhen, sollen optional auch die x- und y-Koordinaten der Kontrollpunkte variabel sein. Im Idealfall sind alle Punkte unabhängig von einander innerhalb der Fläche frei positionierbar, die fixen Eckpunkten ausgenommen. Dies würde die Variablenanzahl jedoch deutlich erhöhen. Daher wird vorerst der Kompromiss gewählt, die Kontrollpunkte in horizontalen Zeilen und vertikalen Spalten zusammenzufassen. Diese Zeilen und Spalten sind in x- respektive y-Richtung frei positionierbar (siehe Bild 6.1). Hieraus resultiert pro Reihe bzw. Spalte eine zusätzliche Variable, was beispielhaft bei 16 Kontrollpunkten zu insgesamt 20 Variablen führt. Alternativ soll die Positionierung der Spalten und Zeilen auch manuell vorgenommen werden können.

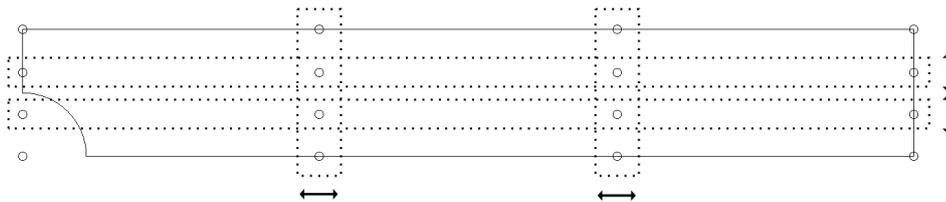


Abbildung 6.1.: Schematische Darstellung der variablen Kontrollpunkte

### 6.1.3. Algorithmus

Da die Gradienten analytisch bestimmbar sind, sind laut **Schumacher 2013** Verfahren der mathematischen Programmierung zu empfehlen. Daher wird an dieser Stelle die *Sequentielle Quadratische Programmierung (SQP)* als Algorithmus gewählt. Dieser ist stabil und für die Zwecke dieser Arbeit ausreichend. Im Vergleich zur Nutzung eines generischen Algorithmus ist allerdings die Wahrscheinlichkeit, ein lokales statt dem globalen Optimum zu finden, größer. Zudem unterstützt der Algorithmus keine Mehrzieloptimierung. In Kapitel 8.2 werden mögliche Alternativen und Kombinationen diskutiert.

Die Einbindung des Algorithmus ist Teil des vorgegebenen UFO-Tools und wurde bereits in Kapitel 3.2.9 beschrieben. Um einen Algorithmus zu wählen, muss dieser im config-File definiert werden.

### 6.1.4. Zielfunktion

In der Arbeit **Hondas** wurden zwei Zielfunktionen definiert. Zum einen die Maximierung der Festigkeits- und Stabilitätskennwerte, zum anderen die Minimierung der Faserkrümmung. Somit wird der ideale Kompromiss aus hohen Festigkeitswerten bei möglichst geringer Krümmung gesucht. Da der SQP-Algorithmus lediglich die Optimierung einer Zielfunktion unterstützt, wird die zweite als Restriktion in den Optimierungsprozess integriert. Dieser Vorgang wird im nachfolgenden Unterkapitel detaillierter beschrieben. Somit wird in dieser Arbeit als Zielfunktion die Minimierung der Spannungen im Bereich des Loches definiert. Diese sollen anhand des Tsai-Wu-Kriteriums ausgewertet werden.

Die Zielfunktion wird im UFO-Tool in den Matlab-Solver *fmincon* (wenn keine Randbedingungen gegeben sind *fminunc*) eingebunden. Zum Lösen des Optimierungsproblems muss daher in der config-Datei die zu optimierende Variable *maxTsaiWu* erstellt und ihr der zugehörige Rückgabewert zugewiesen werden. Dieser Wert wird aus den numerischen Berechnungen gewonnen. Nastran gibt die Tsai-Wu-Koeffizienten für jede Zelle in jeder Lage aus. Bei acht Lagen und 400 Zellen ergeben sich 3200 Koeffizienten. Zusätzlich wird in der Ausgabedatei direkt der größte Koeffizient pro Zelle bereitgestellt. Um die hohe Anzahl an Rückgabeparametern auf einen einzigen Wert für die Zielfunktion zu bringen, sollen die jeweilig größten Koeffizienten der 400 Zellen ausgelesen und in die Kreisselmeier-Steinhauser-Funktion übergeben werden. Der daraus resultierende Wert stellt zwar eine Vereinfachung der Ergebnisse dar, bietet aber dennoch eine aussagekräftige Information über die Güte des Entwurfs. Daher dient er als Ergebnis für die Zielfunktion.

Hierbei sollte darauf geachtet werden, wo der maximale Tsai-Wu-Index auftritt. Werden alle Indizes der Struktur betrachtet, kann es passieren, dass fehlerhafte Winkel (siehe Kapitel 5.7.1 Unterpunkt Bezier-Fläche) im hinteren Bereich der Struktur große Werte erzeugen, wodurch das Ergebnis des Entwurfs verfälscht wird. Um dem entgegenzuwirken, kann versucht werden, die Winkelsprünge mit Hilfe der Winkel-Restriktionen so zu begrenzen, dass diese Fehlwinkel nicht entstehen. Alternativ können die Tsai-Wu-Indizes des hinteren Bereichs aus der Berechnung der Kreisselmeier-Steinhauser-Funktion schwächer gewichtet oder ganz entfernt werden. Eine dritte, indirekte Option besteht darin, die Kontrollpunkte

durch Veränderung ihrer x-Koordinaten in den vorderen Bereich zu verschieben. Diese Methode behebt die Problematik nicht, mindert das Problem allerdings, da durch weniger KP im hinteren Bereich dort die Verformung der Bezier-Fläche abgeschwächt wird.

In Anlehnung an die Vorgehensweise **Hondas** wird sich dazu entschieden, bei der Auswertung nur die Tsai-Wu-Indizes im Lochbereich ( $0 < x < 100\text{mm}$ ,  $0 < y < 100\text{mm}$ ) in die KMS-Funktion zu übergeben. Dies hilft zwar, den optimalen Wert im Lochbereich zu finden, ist bezogen auf die gesamte Struktur aber mathematisch nicht korrekt, da nun Winkeländerungen im hinteren Bauteilbereich nicht mehr in die Betrachtung miteinbezogen werden. In Kapitel 8.2 werden Lösungsansätze für dieses Problem besprochen.

### 6.1.5. Restriktionen

Direkte Restriktionen sind in der Aufgabenstellung nicht vorgegeben. Allerdings wurde im vorangegangenen Unterkapitel die Problematik genannt, dass im eingesetzten Algorithmus nur eine Zielfunktion definiert werden kann. Daher wird die Methode der *restriktionsorientierten Transformation* angewandt (vergl. 6.1.4). Hierbei wird die zweite Zielfunktion, in diesem Falle die durchschnittliche Krümmung, in eine Restriktion umgewandelt. Die Krümmung wird auf einen konstanten Wert restringiert und mit diesem die Optimierungsrechnung durchgeführt. Somit wird aus dem Ziel, ein möglichst gutes Verhältnis zwischen Festigkeit und Fertigbarkeit zu schaffen, das Ziel, das Optimum an Festigkeit unter Einhaltung der Fertigbarkeit zu erreichen. Umso mehr Durchläufe mit unterschiedlichen Krümmungswerten durchgeführt werden, umso genauer kann die optimale Kombination aus mechanischen Eigenschaften und Prozessierbarkeit angenähert werden.

Um den Entwurfsraum der Optimierung sinnvoll zu beschränken, werden die z-Variablen mittels *expliziter Restriktionen* begrenzt. Hierfür soll ein Gesamtbereich von zehn bei mittigem Startwert gewählt werden. Da diese Werte nicht vorgegeben sind, können sie bei Bedarf angepasst werden. Wie in Kapitel 5 beschrieben, werden die Winkel-Restriktionen zur Einhaltung eines harmonischen Verlaufs in Form externer Funktionen und nicht als Restriktionen des Algorithmus eingebracht. Daher werden sie hier nicht betrachtet.

Die programmiertechnische Einbindung der Restriktionen findet im UFO-Tool statt, weswegen die entsprechenden Werte lediglich über das config-File übergeben werden müssen. Hierzu muss dort die entsprechende Variable *dsk* definiert und ihr der zugehörige Rückgabeparameter

zugewiesen werden. Diesen Rückgabewert ermittelt das Bezier-Tool (siehe Kapitel 5) anhand der Höhengsnittlinien.

## 6.2. Programmiertechnische Umsetzung

In diesem Kapitel wird die Programmiertechnische Umsetzung des Optimierungstools beschrieben. Im Unterkapitel 6.2.1 wird zunächst der Aufbau des Optimierungstools anhand eines Flussdiagramms aufgezeigt. Der Zweck und die Umsetzung der dabei genannten Module werden im anschließenden Unterkapitel beschrieben. Zusätzlich wird für jedes Modul der Funktionsaufruf angegeben, aus dem zu entnehmen ist, welche Parameter an das Modul übergeben werden und welche es zurückgibt.

### 6.2.1. Aufbau des Optimierungstools

Um das gesamte Optimierungstool schnell auf unterschiedliche Problemstellungen anpassen zu können, ist es sinnvoll, es möglichst variabel und dynamisch zu halten. Diese Fähigkeit kann mittels eines modularen Aufbaus gewährleistet werden. Hierbei werden nicht alle Abläufe in ein einziges Script geschrieben, sondern ein Hauptsript programmiert, welches auf verschiedene Funktionen/Module zurückgreift. In diese Module werden Eingangsdaten übergeben, wo diese mit den programmierten Prozeduren verarbeitet werden. Anschließend übergibt das jeweilige Modul seine Ergebnisse als Rückgabewerte an das aufrufende Script. Alternativ ist es auch möglich, dass ein Modul keine Parameter zurückgibt, sondern diese stattdessen in eine externe Datei schreibt. Auf diese Datei kann anschließend von einem anderen Modul zugegriffen werden. Ändert sich in einem neuen Projekt eine Anforderung, kann diese verhältnismäßig einfach durch den Austausch des entsprechenden Moduls umgesetzt werden. Um den Aufbau und die Abläufe des Optimierungstools zu veranschaulichen, werden nachfolgend der schematische Programmablauf und der Datenfluss innerhalb der Prozedur dargestellt. Zusätzlich werden die Module *Config erstellen* und *BDF erstellen* detaillierter betrachtet. Die Funktionsweisen und die programmiertechnische Umsetzung der einzelnen Module werden in den anschließenden Kapiteln im Detail betrachtet.

In Diagramm 6.2 ist zunächst der Optimierungsablauf mit den wichtigsten Funktionen schematisch dargestellt. Kleinere Hilfsfunktionen sind aus Gründen der Übersichtlichkeit nicht aufgeführt. Mittels der *GUI* erfolgt die manuelle Eingabe aller relevanter Informationen und der Start des Optimierungsprozesses. Hierbei wird zunächst die Funktion *Config erstellen* aufgerufen, welche die Datei *config.m* schreibt. Diese wird anschließend automatisch mit der

Funktion *Run* ausgeführt. Im ersten Schritt ruft *Config* das Modul *BDF erstellen* auf, welches wiederum mit dem *Bezier-Tool* die Winkel erzeugt und in die *bdf*-Datei schreibt. Darauffolgend startet *Config* die Berechnung des *bdf*-Files mit *Nastran* und lässt *MaxTsai* anschließend die *f06*-Ergebnisdatei auswerten. Die gewonnenen Daten werden von *Config* bewertet und nach Abschluss der Optimierung über die *Run*-Funktion an die *GUI* zur visuellen Ausgabe übergeben.

Das Flussdiagramm 6.3 veranschaulicht den relevanten Datenfluss des Optimierungsprozesses. Dieser beginnt mit der Hinterlegung der Startwerte für die z-Koordinaten der Kontrollpunkte im Modul *Config erstellen*. Bei Aufruf der Funktion schreibt dieses die Koordinaten in die *config*-Datei, welche die Werte bei ihrer Ausführung wiederum an die Funktion *BDF erstellen* übergibt. Diese erzeugt mithilfe des *Bezier-Tools* die gesuchten Winkel und schreibt sie unter Verwendung des Moduls *PCOMPwrite* in die *bdf*-Datei. *Nastran* berechnet diese und erzeugt die *f06*-Datei mit den zu betrachtenden Tsai-Wu-Indizes. Die Funktion *MaxTsai* sucht aus der *f06*-Datei die maximalen Indizes der einzelnen Zellen und fasst diese mit der Kreisselmeier-Steinhauser-Funktion zusammen. Der resultierende Parameter wird an *config* übergeben und ausgewertet. Liegt ein Minimum vor, wird das Ergebnis an die *GUI* übermittelt, wenn nicht, werden die Variablen verändert und mit diesen erneut die Funktion *BDF erstellen* aufgerufen. Diese Schleife wiederholt sich solange, bis ein Optimum gefunden wurde bzw. andere Abbruchkriterien erfüllt sind. Der Ablauf erfolgt analog für die x- und y-Koordinaten, wenn diese als variabel gesetzt sind. Die Eingabe der Startwerte erfolgt hier jedoch über die *GUI*.

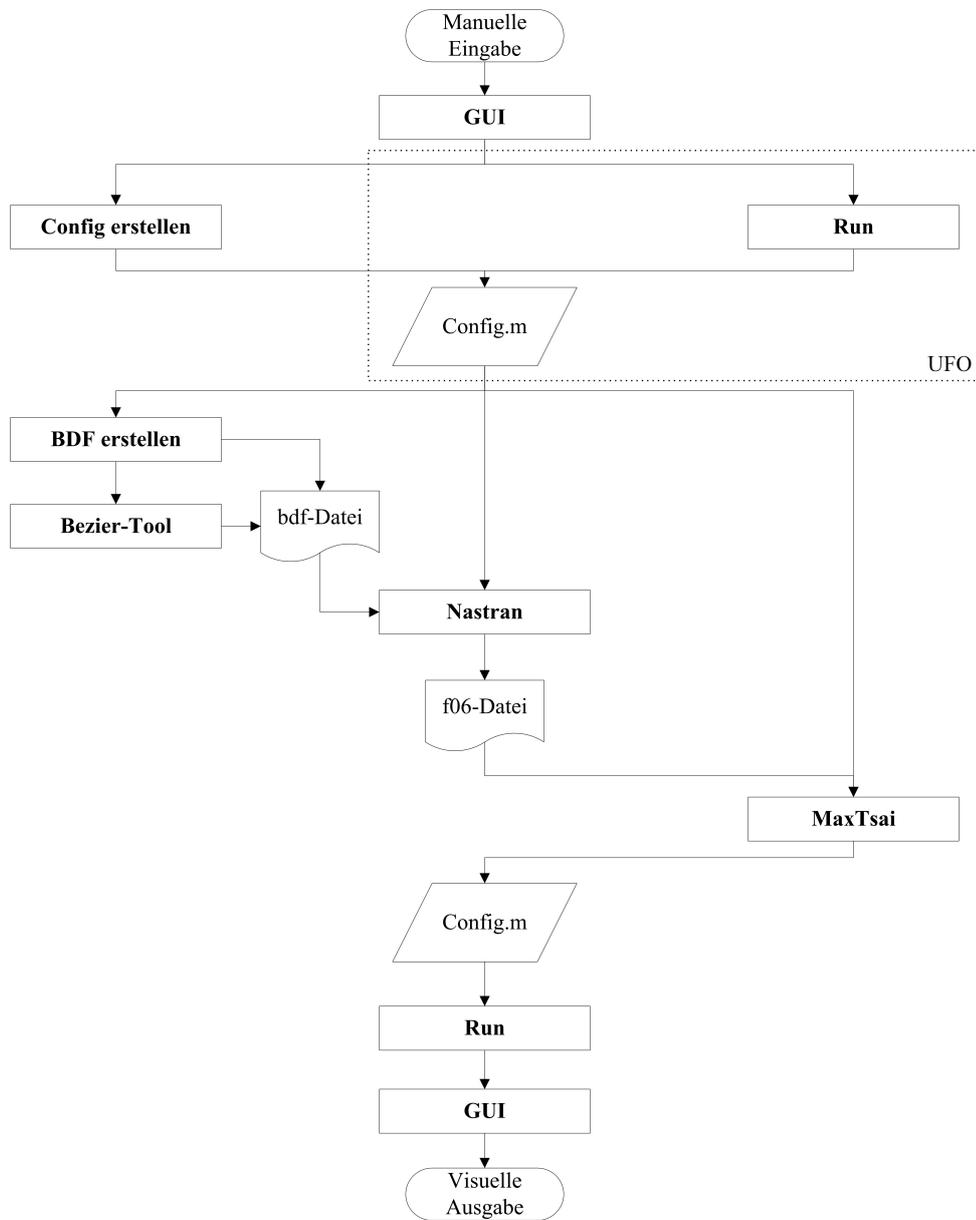


Abbildung 6.2.: Flussdiagramm Optimierungsablauf

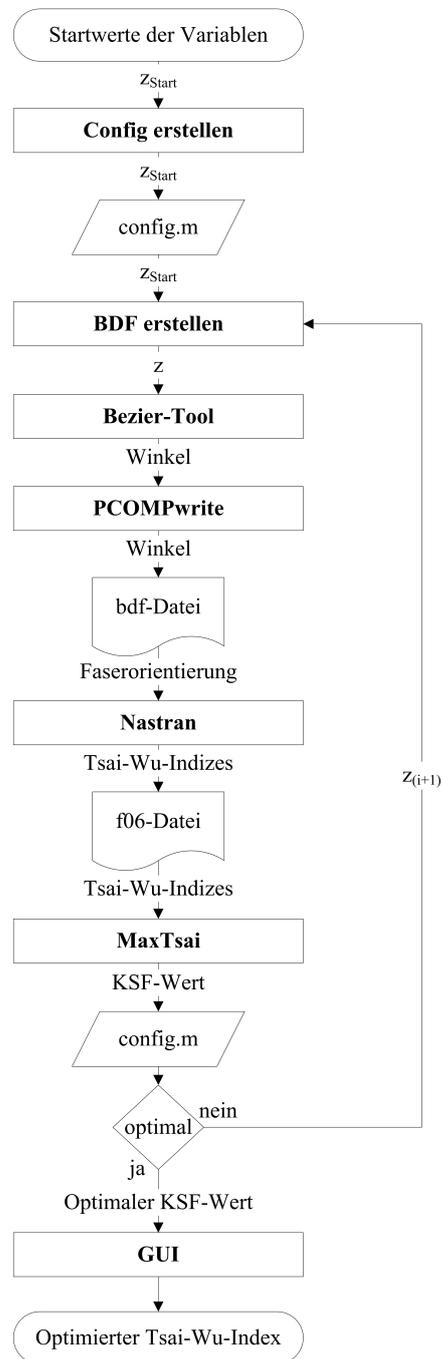


Abbildung 6.3.: Datenfluss des Optimierungsprozesses

Diagramm 6.4 visualisiert den Entstehungsablauf der für den Optimierungsprozess relevanten Konfigurationsdatei. Hierzu werden als erstes die in der *GUI* eingetragenen Informationen übergeben. Beispiele dieser Informationen sind die maximal zulässige Durchschnittskrümmung, das Delta zur Gradientenbildung und die Anzahl der Bezier-Kontrollpunkte. Als z-Startkoordinaten werden entweder die hinterlegten oder manuell eingetragene Werte verwendet. Das Modul *AbschnitteConfig* liest aus der originalen Konfigurationsdatei die invariablen Abschnitte aus und übergibt diese. Anhand all dieser Eingangsdaten schreibt das Modul abschließend die neue *config*-Datei.

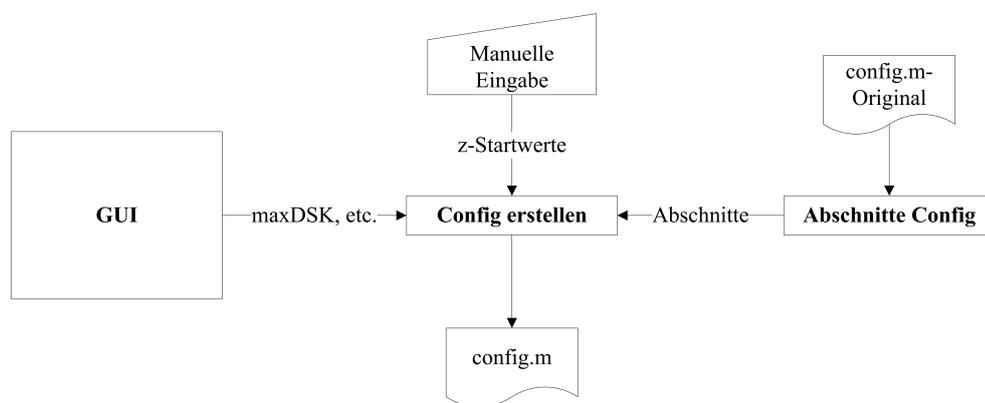


Abbildung 6.4.: Flussdiagramm Erstellung Konfigurationsdatei

Die Erstellung der *bdf*-Datei ist ebenfalls ein wesentlicher Prozess und wird daher in Diagramm 6.5 abgebildet. Die Konfigurationsdatei ruft das Modul *BDF erstellen* auf und übergibt die z-Koordinaten. Die Funktion *KuMP* ermittelt aus den Geometriedaten der *bdf*-Datei die x- und y-Koordinaten der Bezier-Kontrollpunkte und die Zellmittelpunkte der FE-Vernetzung. Diese Daten werden an das Modul *BDF erstellen* übergeben, welches sie zusammen mit den z-Koordinaten an das *Bezier-Tool* weiterleitet. Dieses erzeugt daraus die gesuchten Winkel an den Zellmittelpunkten und gibt diese zurück. Die Winkel werden mittels *PCOMPwrite* in definiertem Format in die externe Textdatei *output.txt* geschrieben. Im letzten Schritt fügt das Modul die von *AbschnitteBDF* übermittelten invariablen Bereiche der *bdf*-Datei und die aus *output.txt* eingelesenen Zeilen zur neuen *bdf*-Datei zusammen.

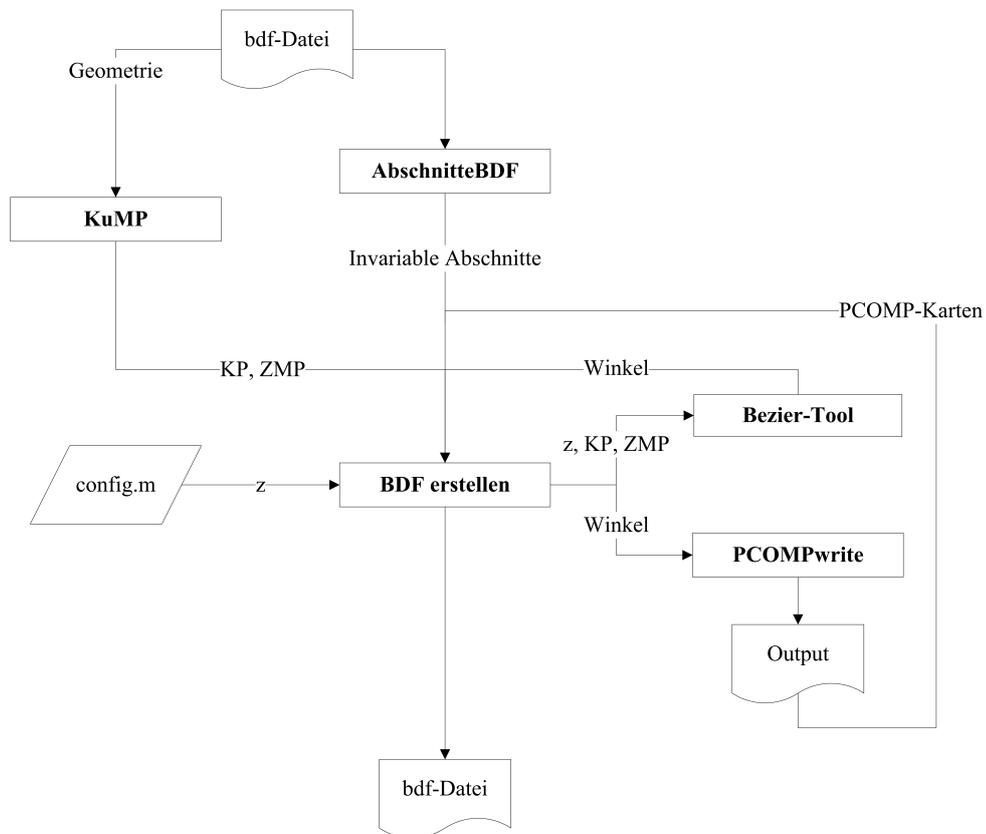


Abbildung 6.5.: Flussdiagramm Erstellung bdf-Datei

### 6.2.2. Hauptmodule

Zunächst werden die drei Hauptmodule des Optimierungstools beschrieben. Diese nehmen jeweils eine übergeordnete Rolle ein, da sie andere Module ansteuern und aus den zurück erhaltenen Informationen eine abschließende Ausgabe erzeugen.

#### Opti\_GUI

Das Modul *Opti\_GUI* bietet eine grafische Oberfläche für unterschiedliche Ein- und Ausgaben des Optimierungstools und dient damit der Vereinfachung des Umgangs mit diesem. Es übernimmt selbst keine für den Optimierungsprozess notwendigen Aufgaben. Da es der Startpunkt der Optimierung ist, besitzt es keinen Funktionsaufruf. Der Umgang mit der

Oberfläche wird in Kapitel 6.3.1 näher erläutert.

Die GUI wurde mithilfe des Matlab-eigenen Editors *GUIDE* geschrieben und funktioniert nur im Zusammenspiel mit der *Opti\_GUI.fig*, in der die grafischen Informationen der Eingabeoberfläche hinterlegt sind. Da die grafische Oberfläche kein essenzieller Bestandteil des Optimierungstools ist, wird die Beschreibung der programmiertechnischen Umsetzung auf die wesentlichen Punkte beschränkt. Alle relevanten Funktionen und Abläufe liegen im sog. callback des *Start*-Buttons und werden gestartet, sobald der Nutzer sämtliche Informationen eingetragen hat und diesen anklickt. Zunächst werden alle eingetragenen Eingabeparameter ausgelesen und internen Variablen zugewiesen. Bei variablen Kontrollpunkten werden die Werte der zugehörigen Vektoren mithilfe des *KP*-Moduls bearbeitet. Manuell kann eine Matrix mit Delta-Werten definiert werden, die mittels einer Schleife vom Tool nacheinander automatisch in die config-Datei eingesetzt werden und mit denen jeweils ein Optimierungsprozess durchgeführt wird. Wenn *Neue Rechnung* gewählt wurde, werden die Daten des vorherigen Durchlaufs gelöscht und mit der Funktion *Config erstellen* eine neue config-Datei erzeugt. Bei *Weiterrechnen* entfällt dieser Schritt, wodurch die config-Datei mit den alten Einstellungen beibehalten wird. Die Berechnung wird im nächsten Schritt durch Ausführen der *Run*-Funktion gestartet, die nach Abschluss des Optimierungsprozesses den optimierten Tsai-Wu-Index und die dazugehörigen Variablen zurückgibt. Zusätzlich erfolgt die Berechnung eines Vergleichswertes mit dem Modul *Rechnung 04590*. Zur visuellen Ausgabe der optimierten Auslegung wird das Modul *Darstellung Optimum* ausgeführt, welches zudem den tatsächlich maximalen Tsai-Wu-Index, die Zelle, in der dieser auftritt und die durchschnittliche Krümmung bereitstellt. Alle gewonnenen Werte bzgl. der optimierten Auslegung sowie die berechnete prozentuale Verbesserung des Tsai-Wu-Index werden in der GUI angezeigt. Diese wird als Bild gespeichert und eine Excel-Tabelle mit den optimierten Variablen erzeugt. Im letzten Schritt werden die Ergebnisse mit der Funktion *Ergebnisse speichern* in einen gemeinsamen Ordner verschoben. Unabhängig von dieser Prozedur speichert der *Merken*-Button alle Verwaltungsparameter in einer externen Datei, damit diese beim nächsten Aufruf wieder zur Verfügung stehen. Es sei darauf hingewiesen, dass das Modul keine Schutzfunktionen beinhaltet, die bei falscher Nutzung aussagekräftige Fehlermeldungen zurückgeben.

### **Config\_erstellen**

Bei manueller Nutzung des UFO-Tools müssen alle relevanten Parameter per Hand in die config-Datei eingetragen werden. Da die im vorangegangenen Kapitel beschriebenen grafischen Oberfläche dazu dienen soll, dies zu vereinfachen und übersichtlicher zu machen,

müssen die eingetragenen Daten entsprechend automatisch an das UFO-Tool übergeben werden. Wie in Kapitel 6.1.1 beschrieben, soll hierzu die config-Datei als Schnittstelle fungieren. Mit dem Modul *Config erstellen* wird daher maschinell eine config-Datei mit den getätigten Eingaben der GUI erzeugt. Diese wird vom UFO-Tool eingelesen und übernimmt auf diese Weise alle eingestellten Parameter.

```
Config_erstellen(nastran_exe,nastran_para,bdf_file,f06_ordner,Form,mKP,nKP,KPx,KPy,
Feinheit,Lagen,delta_zahl,delta_exp,max_DSKappa,Bereich_mW,max_Winkel,Bereich_mdW,
max_Winkeldelta,minmax,Faktor)
```

Im Dateikopf können die Startwerte und Grenzen der z-Koordinaten der Bezier-Kontrollpunkte manuell eingegeben werden. Als erstes wird eine Sicherheitskopie der originalen config-Datei gemacht, für den Fall, dass während des Schreibens der neuen Datei ein Fehler auftritt und eine unvollständige Datei entsteht. Es gibt unterschiedliche Ansätze, die Informationen in die config-Datei zu schreiben. Entweder kann die entsprechende Stelle in der vorhandenen Datei gesucht und durch den neuen Wert ersetzt werden oder es werden alle unveränderten Abschnitte extrahiert und die neuen Werte mit der entsprechenden Formatierung zwischen diese Abschnitte geschrieben. Letztere Variante wird gewählt und daher werden zunächst mittels des Moduls *Config\_Abschnitte* die invariablen Abschnitte der config-Datei eingelesen. Um die eingegebenen Informationen in die neue Datei schreiben zu können, werden verschiedene programmiertechnische Vorbereitungen, wie z. B. Umwandlungen von Datentypen und Formatdefinitionen für die Ausgabe, durchgeführt. Da die x- und y-Koordinaten der Kontrollpunkte als Variablen einen wesentlich größeren Bereich als die z-Koordinaten abdecken, fallen bei Änderungen mit dem vorgegebenen Delta Veränderungen am Ergebnis sehr gering aus (siehe Kapitel 3.2.2). Um dieses Problem zu umgehen, werden die Koordinaten beim Schreiben der config-Datei durch den Wert *Faktor* dividiert und später im Modul *KuMP* wieder in die ursprüngliche Dimension gebracht. Hierbei ist darauf zu achten, dass ein sinnvoller Faktor gewählt wird, damit alle Variablen einen ähnlichen Einfluss auf den Entwurf haben. Nach dieser Skalierung werden verschiedene Platzhalter für die Variablen erzeugt, die später zum Schreiben der config notwendig sind. Der letzte Abschnitt des Moduls dient dazu, die neue config-Datei aus all diesen Daten und den eingelesenen Abschnitten in entsprechender Formatierung zusammenzufügen und zu schreiben.

### **BDF\_erstellen**

Um die vom Bezier-Tool erzeugten Winkel in das Rechenmodell zu integrieren, wird in jeder Iteration mit dem Modul *BDF erstellen* ein neues bdf-File geschrieben, welches die Materialbeschreibung sowie die Elementinformationen beinhaltet. Zudem gibt das Modul die Anzahl der Elemente und den Wert der durchschnittlichen Krümmung an das aufrufende Script zurück.

$[Anzahl\_Elemente, DSKappa] = BDF\_erstellen(z, Dateiname, mKP, nKP, Feinheit, Lagen, Faktor, Form, KP_x, KP_y, Bereich\_mW, max\_Winkel, Bereich\_mdW, max\_Deltawinkel)$

Der Aufbau des Moduls besteht hauptsächlich aus Aufrufen verschiedener Untermodule. Mithilfe des *KuMP*-Moduls werden zunächst die Bezier-Kontrollpunkte und alle relevanten Informationen des zu untersuchenden Modells aus der bdf-Datei gewonnen. Anschließend startet die Berechnung der Winkel und durchschnittlichen Krümmung mit dem *BezierTool*. Wie auch beim Erstellen der config-Datei wird der Ansatz gewählt, die bdf-Datei komplett neu zu schreiben. Hierfür werden die invariablen Abschnitte des bdf-Files mit *Abschnitte BDF* eingelesen und die Winkel- und Elementinformationen per *PCOMPwrite* im entsprechenden Format zusammengefasst. Im letzten Schritt werden alle Abschnitte in die neue bdf-Datei geschrieben.

### **6.2.3. Untermodule**

Die Untermodule erfüllen eine direkte Aufgabe, um Daten für die Hauptmodule aufzubereiten oder eine bestimmte Ausgabe zu erzeugen. Das in Kapitel 5 behandelte Bezier-Tool stellt ebenfalls ein Untermodul dar, wird an dieser Stelle aber nicht erneut betrachtet.

### **Abschnitte BDF**

Um die Variablen in der bdf-Datei zu definieren, wird der Ansatz gewählt, die Datei für jede Iteration neu zu schreiben. Alternativ könnte die bestehende Datei modifiziert werden. Das Neuschreiben erspart das Suchen und Ersetzen der jeweiligen Werte, macht jedoch das Zwischenspeichern der unveränderten Dateiabschnitte notwendig. Diese Aufgabe übernimmt die Funktion *Abschnitte BDF*.

$[AnfangBDF, EndeBDF] = Abschnitte\_BDF (Dateiname)$

Als erstes wird die übergebene Datei geöffnet. Um das Auslesen und spätere Schreiben der Textdatei möglichst variabel und schnell zu gestalten, werden hierfür sog. *low-level* Ein- und Ausgaben verwendet. Der strukturelle Aufbau der bdf-Datei ermöglicht eine Unterteilung in zwei unabhängige Abschnitte und den zu verändernden Bereich der Materialdefinition. Mittels zweier Schleifen wird anhand von charakteristischen Zeichenfolgen (Begriffen) zuerst der Anfang der Materialbeschreibung und anschließend das Ende ebendieser gesucht. Die Abschnitte vor und nach der Materialbeschreibung werden jeweils als Vektoren reiner Textzeichen (*uchar*) gespeichert und an das aufrufende Modul übergeben.

### **Abschnitte Config**

Wie im Modul *Config\_erstellen* beschrieben, wird die zu verwendende config-Datei aus den unveränderten Abschnitten der Originaldatei und neu geschriebenen Zeilen gebildet. Diese feststehenden Abschnitte werden mittels des Moduls *Abschnitte Config* ausgelesen und bereitgestellt. Das Modul *Abschnitte Config* hat daher grundsätzlich die gleiche Funktion wie *Abschnitte BDF*.

$[AnfangConfig, MitteConfig1, MitteConfig2, MitteConfig3, EndeConfig] = Abschnitte\_Config$

Die programmiertechnische Umsetzung erfolgt analog zu der in *Abschnitte BDF*. Allerdings ist es für die Bestimmung der dazugehörigen Anfänge und Enden der mittleren Bereiche notwendig, jeweils zwei Suchen durchzuführen.

### **Darstellung\_Optimum**

Um nach Beendigung des Optimierungsprozesses die Ergebnisse visuell darzustellen und für spätere Betrachtungen zu speichern, wird das Modul *Darstellung Optimum* genutzt. Unter Verwendung der optimierten z-Koordinaten der KP berechnet es wie das Bezier-Tool die Faserwinkel. Diese werden in zwei getrennten Grafiken zum einen im gesamten Strukturbereich, zum anderen nur im Lochbereich dargestellt. Zwei weitere Diagramme zeigen die dreidimensionale Bezier-Fläche und die daraus hergeleiteten Höhengschnittlinien. Zusätzlich dient das Tool zur Bestimmung des tatsächlich maximalen Tsai-Wu-Wertes (nicht des KS-Funktionswertes) sowie der Zelle, in der dieser vorliegt. Da das UFO-Tool den Wert der Restriktion nicht zurückgibt, wird auch die durchschnittliche Krümmung bei optimierter Auslegung als Rückgabewert bereitgestellt.

$[maxTsai,maxZelle,DSKappa] = Darstellung\_Optimum(nastran\_exe,nastran\_para, f06\_ordner,Dateiname,z,mKP,nKP,Feinheit,Form,Lagen, KP_x,KP_y,Bereich\_mW,max\_Winkel, Bereich\_mdW,max\_Deltawinkel,Faktor)$

In das Modul wird eine große Menge Parameter übergeben, da es sowohl eine Winkelberechnung als auch eine Nastran-Berechnung durchführt. Nach Datentyp-Umwandlung einiger übergebener Variablen werden weitere benötigte Strukturdaten aus der bdf-Datei eingelesen. Der darauffolgende Abschnitt berechnet die Winkel sowie die durchschnittliche Krümmung und erzeugt die visuellen Ausgaben. Die programmiertechnische Umsetzung entspricht der des Bezier-Tools (siehe Kapitel 5.6), es werden lediglich die zwei oben genannten zusätzlichen Diagramme erstellt. Sämtliche Grafiken werden anschließend als Bilddateien gespeichert. Die berechneten Winkel werden in ein neues bdf-File geschrieben und dieses mit Nastran ausgeführt. Im letzten Schritt werden die gefragten Tsai-Wu-Informationen aus der f06-Datei ausgelesen und als Rückgabewerte definiert.

### KP

Die *KP*-Funktion wird notwendig, um die Kontrollpunkte auch in x- bzw. y-Richtung als Variablen definieren zu können. Hierzu muss die Geometrie vorab ausgelesen werden, um die variablen Kontrollpunkte anhand der mini- und maximalen Strukturpunkte eingrenzen zu können. Zusätzlich dient das Modul der Umrechnung oder alternativ der automatischen Belegung der Kontrollpunkt-Startwerte. Der Einfachheit halber wird der Vektor *minmax* auch zur Übermittlung von Informationen darüber, welche Kontrollpunkte variabel sind und ob eine ebene oder freie Start-Bezierfläche definiert ist, genutzt.

$[KP_x,KP_y,minmax] = KP (Dateiname,mKP,nKP,Form,KP_x,KP_y)$

Die bdf-Datei wird geöffnet und sämtliche durch Leerzeichen getrennte Zeichenfolgen als Strings ausgelesen. Anhand dieser werden alle Zellknoten der vernetzten Geometrie gesucht und inklusive ihrer Koordinaten gespeichert. Aus den Koordinatenwerten werden die mini- und maximalen Geometriepunkte gewählt und in Vektorform zur Rückgabe bereitgestellt. Zudem wird die äquidistante Distanz zwischen den KP bei automatischer Einteilung berechnet. Für die Belegung der Vektoren der x- und y-Koordinaten der inneren Kontrollpunkte werden diese zunächst auf ihren Inhalt geprüft. Wenn absolute Werte hinterlegt sind, bleiben diese unverändert. Prozentuale Angaben rechnet das Modul mit den Maximalwerten der Geometrie

zu absoluten Werten um und bei keinen Angaben erfolgt eine automatische Verteilung der Koordinaten. Die erstellten Vektoren werden an die aufrufende Funktion zurückgegeben.

### **KuMP**

Das *KuMP*-Modul (Kontroll- und Mittelpunkte) hat das Ziel, alle relevanten Informationen aus dem bdf-File auszulesen und für die weitere Verarbeitung bereitzustellen. Hierzu zählen die Koordinaten der Bezier-Kontrollpunkte sowie die der Zellmittelpunkte, die Anzahl der Elemente, Elementdaten, die maximal zulässige Schubspannung und die Dicke der einzelnen Faserlagen.

*[KP, MP, Anzahl\_Elemente, Elementdaten, S, Lagendicke] =  
KuMP (Dateiname,z,mKP,nKP,Faktor,Form,KPx,KPy)*

Analog zum *KP*-Modul werden sämtliche Zeichenfolgen aus der bdf-Datei eingelesen und die Daten der Zellknoten aufbereitet. Des Weiteren werden auf die gleiche Weise die Informationen der Zellelemente ausgelesen, wobei das Modul lediglich CQUAD4 und TRIA3-Elementen verarbeiten kann. Wird die Geometrie mit anderen Elementen vernetzt, muss das Modul angepasst werden. Die gespeicherten Informationen bestehen aus der Zell-ID, der ID der zugehörigen PCOMP-Karte sowie den IDs der drei bzw. vier Zellknoten. Zur Bereitstellung der zulässigen Schubspannung und der Lagendicke werden diese ebenfalls ausgelesen. Anschließend werden aus den Geometriedaten und den z-Koordinaten im *z*-Vektor die Kontrollpunkte definiert. Sind in *KPx* und *KPy* manuelle Werte oder Optimierungsvariablen hinterlegt, werden die automatisch generierten Werte mit diesen überschrieben. Da variable x- und y-Koordinaten der Kontrollpunkte beim Schreiben der config-Datei aus Skalierungsgründen mit dem Wert *Faktor* dividiert wurden, werden sie an dieser Stelle durch Multiplikation mit diesem wieder auf einen absoluten Wert gebracht. Zuletzt werden die Zellmittelpunkte, an denen später die Winkel und Krümmungen berechnet werden sollen, mittels der Formel zur Bestimmung des Schwerpunkts von allgemeinen Polygonen (siehe Kaptitel 3.5.2) berechnet. Diese ist zwar komplexer als die Formeln für Drei- oder Vierecke, macht die Berechnung aber unabhängig von der verwendeten Elementart. In die Formel fließen die Koordinaten der Zellknoten ein, die anhand der Knoten-ID ausgewählt werden. Damit sind alle geforderten Rückgabeparameter bestimmt. Theoretisch brauchen die Kontroll- und Zellmittelpunkte nur ein Mal ausgelesen und in einer Excel-Datei hinterlegt werden, um sie von dort in allen weiteren Durchläufen ohne Suchen einlesen zu können. In der Realität dauert dieser Vorgang jedoch länger, was wahrscheinlich auf die schnellere Verarbeitung von low level Ein- und Ausgaben zurückzuführen ist.

### MaxTsai

Die Funktion *MaxTsai* wertet die Ergebnisse der Nastran-Berechnungen aus und gibt einen kombinierten Rückgabewert für die Zielfunktion aus. Diese Kombination ermöglicht nur einen Ergebniswert betrachten zu müssen, aber dennoch alle Tsai-Wu-Indizes mathematisch zu berücksichtigen. Dies wird mithilfe der Kreisselmaier-Steinhauser-Funktion (siehe Kapitel 3.5.4) realisiert. Zu Informationszwecken während der Optimierungsdurchläufe werden zudem der tatsächliche maximale Tsai-Wu-Index sowie die Zelle, in der er auftritt, ausgegeben.

$$[KSF\_Wert, maxTsai, maxZelle] = MaxTsai(Dateiname, Anzahl\_Elemente, Lagen)$$

Zu Beginn des Moduls können zwei wesentliche Faktoren der Berechnung definiert werden. Zum einen ist das der Roh-Faktor, der bestimmt, wie nah sich die KS-Funktion den tatsächlichen Werten annähert. Normalerweise sind Faktoren von 5-10 üblich, laut [NASA] je nach Anwendung auch 5-200. Da die einfließenden Werte zum Teil jedoch verhältnismäßig klein sind (ca. 0,01), muss Roh sehr groß (>1000) gewählt werden, um eine präzise Annäherung aller Tsai-Wu-Indizes zu gewährleisten. Der zweite wichtige Einstellparameter ist der Bereich der Zellen, die in die KS-Funktion einfließen sollen. Da das f06-Format die Ergebnisse zu Seiten zusammenfasst, wird die Ausgabe der Tsai-Wu-Indizes in regelmäßigen Abständen unterbrochen. Um das Auslesen der maximalen Indizes zu vereinfachen, werden daher zunächst die reinen Tsai-Wu-Linien extrahiert. Diese werden in eine Ausgabedatei geschrieben, in der anschließend die Maximalwerte der einzelnen Elemente gesucht werden. Aus ebendiesen wird im Anschluss der größte Wert und die dazugehörige Zelle identifiziert. Die Indizes der zu betrachtenden Zellen werden im letzten Schritt in die KS-Funktion eingesetzt und so der dritte geforderte Rückgabeparameter berechnet.

### PCOMPwrite

Essenzieller Bestandteil des Optimierungsprozesses ist die Einbringung der Entwurfsvariablen, in diesem Fall der Winkel, in das zu optimierende FE-Modell. Zu diesem Zweck schreibt das Modul *PCOMPwrite* die Winkel in das zur Übergabe an Nastran erforderliche Format der PCOMP-Karten (vgl. Kapitel 3.4.5). Des Weiteren schreibt es die einzelnen Elemente der Vernetzung neu und weist ihnen die entsprechende PCOMP-Karte zu. Als Variante existiert das Modul *PCOMPwrite 04590*, welches ausschließlich ein symmetrisches Laminat ( $[0^\circ \pm 45^\circ 90^\circ]_s$  bzw.  $[0^\circ 90^\circ]_s$ ) erzeugt. Dies wird benötigt, um einen Referenzwert eines konventionellen Laminats zu erzeugen.

*PCOMPwrite(Winkel,Elementdaten,Lagen,S,Lagendicke)*

Anhand der Anzahl an Zellknoten in den Elementdaten wird geprüft, ob Drei- oder Viereckselemente vorliegen. Bevor die Karten geschrieben werden, werden durch Multiplikation mit  $-1$  die Symmetriewinkel erzeugt. Die Ausgabe erfolgt in einer Textdatei (Output.txt), die dazu erstellt wird. In diese werden alle Zeilen der PCOMP-Karte sowie die Definition des dazugehörigen Elementes geschrieben. Je nach Anzahl der Lagen erfolgt eine unterschiedliche Ausgabe. Der Vorgang wird mittels einer Schleife für alle Elemente wiederholt. Diese Datei kann später eingelesen und so als kompletter Abschnitt weiterverarbeitet werden. Die Einstellungen zur Lagendefinition bieten nur die Möglichkeit eines symmetrischen Laminataufbaus mit entweder insgesamt 4 oder 8 Lagen. Sollen andere Lamine betrachtet werden, muss das Modul angepasst werden.

### **Rechnung\_04590**

Mit dem Ziel einen Vergleichswert für den maximalen Tsai-Wu-Index anhand eines konventionellen Laminats bereitzustellen, wird die Funktion *Rechnung\_04590* ausgeführt. Diese betrachtet die gleiche Struktur wie der Optimierungsprozess, jedoch mit einem fest definierten Laminataufbau aus gradlinigen Faserverläufen. Das gewählte Stacking ist ein konventionelles  $[0^\circ \pm 45^\circ 90^\circ]_s$ -Laminat.

*[maxTsai04590] = Rechnung\_04590(bdf\_file,nastran\_exe,  
nastran\_para,f06\_ordner,Lagen,mKP,nKP,Form)*

Zunächst werden für die Funktion sowohl relevante Strukturdaten als auch die invariablen Abschnitte aus der bdf-Datei gelesen. Um die für die Strukturdaten nötige *KP*-Funktion aufrufen zu können, müssen leere Platzhalter erstellt werden. Mit diesen Informationen wird anschließend ein neues bdf-File mit dem  $[0^\circ \pm 45^\circ 90^\circ]_s$ -Laminat und entsprechendem Namen erstellt. Mittels Systemaufrufs wird dieses von Nastran berechnet und aus der Ergebnisdatei der maximale Tsai-Wu-Index, dessen Wert der gesuchte Rückgabeparameter ist, extrahiert.

### **Run**

Die Run-Funktion übernimmt die Aufgabe, das UFO-Tool mit der erzeugten config-Datei aufzurufen und somit die Optimierung zu starten. Nach Beendigung übergibt sie die Optimierungsergebnisse an die grafische Oberfläche (GUI). Zum manuellen Start der Optimierung ohne Verwendung der GUI dient *Run\_manuell.m*, die den selben Aufbau hat, jedoch keine

Matlab-Funktion ist.

```
[variablen, max_tsai] = Run
```

Die Funktion basiert auf dem run-File des UFO-Tools und führt dieses mit dem UFO-Aufruf aus. Hierbei kann festgelegt werden, welche Optimierungsinformationen zurückgegeben werden sollen. Standardmäßig sind als Ausgabe die optimierten Variablenwerte und der dazugehörige Wert der Zielfunktion definiert. Weitere Auswahlmöglichkeiten sind der *UFO.m* zu entnehmen.

### 6.2.4. Hilfsmodule

Zusätzlich zu den für den Optimierungsprozess relevanten Modulen wurden zwei Hilfsmodule programmiert, die der Datenverwaltung dienen.

#### **Ergebnisse\_speichern**

Die Funktion *Ergebnisse speichern* dient dazu, sämtliche relevanten Dateien eines Optimierungsprozesses in einem Ordner zusammenzufassen. Sie erstellt einen neuen Ordner mit Nummer, Ergebniswert und dem verwendeten Delta, in den alle wichtigen Daten verschoben werden.

```
function Ergebnisse_speichern(max_tsai,delta_zahl,delta_exp)
```

Als erstes prüft das Modul, wie viele Rechnungen bereits im aktuellen Ordner vorhanden sind. Hierbei wird nach der größten Rechnungszahl gesucht und nicht die Anzahl der Rechnungsordner geprüft, wodurch problemlos alte Rechnungen gelöscht werden können, ohne die Nummerierung zu stören. Anschließend wird ein neuer Ordner erzeugt, dessen Bezeichnung die entsprechende Rechnungszahl, den übergebenen Ergebniswert sowie das verwendete Delta beinhaltet. Diese Informationen dienen der Übersichtlichkeit bei mehreren Optimierungsrechnungen. In diesen Ordner werden sämtliche Ausgabedateien (siehe Kapitel 6.3.2) sowie eine Kopie der config-Datei verschoben.

#### **Refresh**

Um Speicherplatz zu sparen und eine problemlose Optimierung zu ermöglichen, löscht das Modul *Refresh* Dateien von vorherigen Optimierungsdurchgängen. Soll eine bereits begonnene

Rechnung fortgeführt werden (Einstellung in der GUI), wird die Refresh-Funktion nicht ausgeführt.

Refresh(f06\_ordner)

Das Modul prüft, ob relevante Dateien und Ordner vorhanden sind und entfernt diese dann gegebenenfalls. Anschließend erzeugt es den Ordner für die Ausgabedateien Nastrans neu.

### 6.3. Anwendung des Optimierungstools

Nachdem nun alle Module und ihr Zusammenspiel im Optimierungstool beschrieben sind, erfolgt nun die Anwendung. Hierzu wird zunächst die Eingabe aller Informationen geschildert. Das zweite Unterkapitel befasst sich mit der Ausgabe des Optimierungstools.

#### 6.3.1. Eingabe

Zur Vereinfachung der Eingabe wurde mittels Matlab eine rudimentäre Oberfläche programmiert. Da die GUI (Graphical User Interface - Grafische Benutzeroberfläche) nur ein Hilfsmittel darstellt, können über sie nur die wichtigsten Werte eingegeben werden. Diese werden im nächsten Unterpunkt anhand Bild 6.6 erklärt. Im darauffolgenden Abschnitt wird auf die Parameter eingegangen, die bei Bedarf manuell in den einzelnen Modulen bzw. im UFO-Config-File geändert werden müssen. Wird auf die Verwendung der GUI verzichtet, müssen sämtliche Werte manuell in der Config-Datei eingetragen werden.

## 6. Optimierungstool

**Optimierungstool**

**Verwaltung**

Pfad der Nastran.exe: C:\MSC.Software\MSC\_Nastran\_Student\_Edition\20141\Nastran\bin\nastran.exe Parameter: msgbell=no scr=yes

Vollständiger Name des bdf.files: Lochplatte.bdf

Ordnerpfad für Nastran-Dateien: E:\Optimierungstool\Ergebnisdateien\

**Berechnungsparameter**

Delta für Gradientenbildung: 1 e<sup>-</sup> 3

Delta-Bereich durchlaufen

Feinheit (Integer 100-400): 100

Start-Bezierfläche:  eben  frei\*  
\* Startwerte in "Config\_erstellen" definieren

max. Durchschnittskrümmung: 0.05

**Kontrollpunkte (vertikal)**

4  5  6

Verteilung:  automatisch  manuell  variabel

KP 2:  KP 3:   
KP 4:  KP 5:

**Kontrollpunkte (horizontal)**

4  5  6

Verteilung:  automatisch  manuell  variabel

KP 2:  KP 3:   
KP 4:  KP 5:

(KP 1 und KP 4,5,6 (je nach Anzahl der KP) liegen an den Bauteilgrenzen)

**Winkelmodifikation (optional)**

Winkelbegrenzung  
Zellbereich:   
Maximaler Winkel (°):

Maximale Winkeländerung  
Zellbereich:   
Maximale Winkeldifferenz (°):

**Ergebnisse**

	z11	z12	z13	z14	z21	z22	z23	z24	z31	z32	z3:
z-Koordinat:	5.0598	4.9456	4.9514	4.9569	4.9933	4.9943	5.0842	4.9464	4.9698	5.0787	4

Maximaler Tsai-Wu-Wert bei optimierter Auslegung: **0.0125** Zelle mit max. Tsai-Wu-Index: **71**

Maximaler Tsai-Wu-Wert bei [0/+45/90]s Laminat: **0.0893** Prozentual: **14**

Durchschnittliche Krümmung bei optimierter Auslegung: **0.00776132**

Anzahl Faserlagen:  4  8

Grundfläche:  Rechteck  Trapez\*\*  
\*\* Trapezberechnung nur eingeschränkt

Vergleichswerte berechnen

Neue Rechnung  Weiterrechnen

**Optimierung erfolgreich**

Abbildung 6.6.: Darstellung der grafischen Oberfläche

### GUI-Parameter

In das Feld *Pfad der Nastran.exe* wird der vollständige Pfad inkl. der gewünschten Nastran.exe. Hier muss darauf geachtet werden, welche exe. genutzt wird. Bei Nastran-Anwendungen mit *w* am Ende der Bezeichnung (z.B. *nastranw.exe*) kann es zu Problemen kommen, da das Matlab-Script nicht wartet, bis die Rechnung abgeschlossen ist und Ergebnisse vorliegen. Unter *Parameter* können optionale Zusatzparameter, mit denen Nastran gestartet werden soll, definiert werden. Beispiele dieser Parameter sind in Tabelle 6.1 aufgelistet. Weitere mögliche Parameter sind bei CAE 2015 zu finden. Während bei *Name des bdf.files* die Bezeichnung des zu optimierenden bdf-Files hinterlegt wird, dient *Ordnerpfad für Nastran-Dateien* dazu, den Speicherort der Berechnungsdateien von Nastran zu bestimmen. Dies ist sinnvoll, damit der Hauptordner übersichtlich bleibt. Wird hier ein Ordner gewählt, der noch nicht existiert, wird dieser automatisch erstellt. Da in der Regel die genannten Einstellungen für ein Projekt nicht wechseln, können diese Einträge mit der *Merken*-Schaltfläche gespeichert werden, damit sie beim nächsten Aufruf der GUI direkt wieder zur Verfügung stehen.

Parameter	Beschreibung
memory	Größe des zugewiesenen Arbeitsspeichers [Example: memory=4000Mb]
parallel	Anzahl zugewiesener Prozessorkerne [Example: parallel=4]
scratch	Bestimmung der zu löschenden Rechen- und Ausgabedateien. Optionen: <i>yes</i> erhält nur die Ausgabedateien und Protokolldateien, <i>mini</i> komprimierte Daten und <i>no</i> sämtliche Dateien
out= <i>Pfad</i>	gewünschter Zielordner für die Ausgabedateien (Im Script wird dieser automatisch über das Feld <i>Ordnerpfad für Nastran-Dateien</i> festgelegt)

Tabelle 6.1.: Optionale Parameter für Nastran

Im darunterliegenden Bereich *Berechnungsparameter* werden Informationen für den Optimierungsalgorithmus, die Erstellung und Berechnung der Bezierfläche sowie zur optionalen Winkelbegrenzung eingegeben. Die maximale Durchschnittskrümmung sowie die Mantisse und der Exponent des Deltas der Gradientenbildung sind Informationen für den Algorithmus. Zur Findung eines idealen Deltas kann optional die Funktion *Delta-Bereich durchlaufen* aktiviert werden. Hierdurch wird die Optimierungsprozedur automatisch mittels einer Schleife mit unterschiedlichen Deltas wiederholt. Diese sind im Modul *Opti\_GUI* hinterlegt und können bei Bedarf manuell angepasst werden. Zur Steuerung der Bezier-Fläche kann der Wert der Feinheit sowie die Anzahl, Verteilung und Position der Kontrollpunkte eingegeben werden. Bei der manuellen Eingabe der KP-Koordinaten können absolute oder prozentuale Werte (0.2 = 20%) angegeben werden. Zudem wird über die Wahl der *Start-Bezierfläche* der Startentwurf definiert. Bei der Wahl *frei* müssen die zu verwendenden Startwerte in *Config erstellen* definiert werden. Im Bereich *Winkelmodifikation* können die optionalen Winkelrestriktionen zugeschaltet und die Maximalwerte sowie die zu prüfenden Zellbereiche eingetragen werden. Die Schaltflächen bezüglich der Faserlagen und der Grundfläche sind lediglich Informationen für den Optimierungsprozess und keine Einstellmöglichkeiten.

### Weitere Parameter

Alle Parameter des Optimierungsprozesses, die nicht über die GUI geändert werden können, sind in den jeweiligen .m-Files im Dokumentenkopf definiert, um eine schnellere manuelle Anpassung zu ermöglichen. In Tabelle 6.2 sind die Parameter mit einer Beschreibung und dem zugehörigen .m-File aufgeführt. Sämtliche Parameter sind mit sinnvollen Konstanten

vordefiniert. In den Modulen sind zum Teil weitere Faktoren, Bereiche und Parameter, bspw. zur grafischen Ausgabe, definiert. Da diese hinsichtlich des Optimierungsprozesses von untergeordneter Bedeutung sind, wird an dieser Stelle nicht auf sie eingegangen.

Tabelle 6.2.: Weitere Eingabeparameter

<b>Paramter</b>	<b>Beschreibung</b>	<b>Modul</b>
skala_unten	Mindestwert der z-Koordinaten	Config erstellen
skala_oben	Maximalwert der z-Koordinaten	Config erstellen
z_Start_eben	Startwerte der z-Koordinaten bei ebener Bezierfläche	Config erstellen
z_Start_frei	Startwerte der z-Koordinaten bei freier Bezierfläche	Config erstellen
Roh	Roh-Faktor für die Berechnung der Kreisselmaier-Steinhauser-Funktion	MaxTsai
Zellen_Anfang	ID der ersten Zelle, die in die KSF-Betrachtung einfließen soll	MaxTsai
Zellen_Ende	ID der letzten Zelle, die in die KSF-Betrachtung einfließen soll	MaxTsai
range_delta	Verschiedene Deltas (Mantissen und Exponenten) zur Gradientenbildung	OptiGUI
HSL_Anzahl	Anzahl der auszugebenen Höhengschnittlinien	BezierTool / Darstellung Optimum

### 6.3.2. Ausgabe

Während und nach Beendigung des Optimierungsprozesses erzeugt das Optimierungstool verschiedene Ausgaben, um dem Anwender Informationen über die aktuelle Iteration und das Ergebnis der Optimierung bereitzustellen. Dies erfolgt durch Ausgabedateien, Diagramme sowie durch Anzeigen in der GUI und im Matlab-Command-Window.

#### Diagramme

Bei jeder Iteration werden visuelle Darstellungen der aktuellen Bezier-Fläche und der Winkel im Lochbereich erzeugt. Nach erfolgreicher Optimierung werden diese Diagramme sowie zwei weitere mit den optimierten Werten dargestellt. Die zusätzlichen Diagramme zeigen einerseits die Höhengschnittlinien, andererseits die daraus resultieren Winkel im gesamten

Strukturbereich. Wenn die Diagramme modifiziert oder weitere Diagramme erstellt werden sollen, kann dies in den Modulen *Bezier-Tool* und *Darstellung Optimum* realisiert werden.

### **Ausgabedateien**

Die Ausgabedateien umfassen die bdf- und f06-Dateien aller Iterationen, eine UFO-Protokolldatei, eine Excel-Tabelle mit den optimalen Kontrollpunktvariablen und ein Bild der GUI als Übersicht der Ein- und Ausgabeparameter. Zusätzlich werden die vier erzeugten grafischen Darstellungen (siehe Unterpunkt *Diagramme*) als Bild-Dateien gespeichert. Wie im Modul *Ergebnisse\_speichern* beschrieben, wird für jede Optimierung ein Ordner angelegt, in dem alle genannten Dateien gespeichert werden.

### **GUI-Parameter**

Die wichtigsten Informationen werden nach Abschluss der Optimierung direkt in der grafischen Oberfläche angezeigt. Hierzu zählen der maximale Tsai-Wu-Wert, die Zelle, in der er auftritt sowie die durchschnittliche Krümmung. Alle drei Werte beziehen sich auf die optimierte Auslegung. Zusätzlich wird ein Vergleichswert einer konventionellen Auslegung sowie die darauf bezogene prozentuale Größe des optimierten Tsai-Wu-Wertes angegeben. Des Weiteren werden tabellarisch die optimierten Variablenwerte dargestellt.

### **Command Window-Parameter**

Die Ausgabe im Command Window von Matlab wird maßgeblich vom UFO-Tool bestimmt. Hier werden während des Prozesses die aktuellen Informationen der Iterationen und nach Beendigung eine Zusammenstellung der Ergebnisse ausgegeben. Die Abschlussinformationen sind abhängig von den im Run-File geforderten Rückgabewerten. Standardmäßig sind hierfür das Ergebnis der Zielfunktion sowie die zum Optimum gehörenden Variablenwerte definiert. Falls weitere Ausgaben, wie bspw. die Hesse-Matrix, die Abbruchbedingung des Optimierers oder die Anzahl der benötigten Iterationen, von Interesse sind, kann die Ausgabe dieser im Modul *Run* in Nastran eingestellt werden.

## **6.4. Sensitivitätsanalyse**

Ein passendes Delta für die Gradientenbildung des SQP-Algorithmus zu finden ist wichtig, damit der Optimierungsprozess sinnvolle Ergebnisse findet und dies im Idealfall mit wenigen Iterationen und somit schnell. Wird es zu klein gewählt, haben die Änderungen der Variablen

zu wenig Einfluss auf die Ausgabewerte, sind sie zu groß, kann es passieren, dass der Algorithmus Funktionsabstiege zu einem Minimum nicht erkennt. Um daher sinnvolle Werte abschätzen zu können, wird üblicherweise vorab eine Sensitivitätsanalyse (SA) (siehe Kapitel 3.2.7) des zu optimierenden Modells durchgeführt. An dieser Stelle wird jedoch auf eine umfangreiche Analyse der Sensitivitäten verzichtet und lediglich eine Einflussprüfung der einzelnen Variablen auf die Strukturantwort durchgeführt. Hierzu werden alle Variablen mit einer Reihe von zu testenden Deltas verändert und der Einfluss dieser Änderungen auf die Ergebnisse betrachtet. Gerechtfertigt wird diese Vorgehensweise dadurch, dass hinsichtlich der geringen Größe des Optimierungsproblems die Rechendauer auch ohne SA überschaubar ist und dass für jede Änderung der Bézier-Flächen-Eigenschaften eine erneute SA notwendig wäre, wodurch der Aufwand den Nutzen übersteigen würde.

Zur Umsetzung der Einflussprüfung könnte theoretisch ein separates Modul programmiert werden, jedoch kann sich auch zu Nutze gemacht werden, dass die ersten Rechnungen beim Optimierungsprozess genau dem gewünschten Schema folgen. Wird der Prozess nun jeweils mit vorher festgelegten Deltas gestartet, können die gesuchten Einflüsse der einzelnen Variablen den Ergebnissen entnommen werden. Dies ist für vier Deltas bei einer ebenen Startfläche durchgeführt worden. Hierbei ergab sich, dass die Strukturantwort bei nur einer veränderten Variablen nahezu unabhängig von der Größe des Deltas ist. Lediglich beim Kontrollpunkt 1 (Ecke am Loch) entstehen teilweise leichte Unterschiede. Die somit identischen Differenzen der Strukturantwort bezogen auf den Ausgangswert ohne Variablenänderung sind in Tabelle 6.3 dokumentiert. Hierbei entspricht die Tabelleneinteilung der Kontrollpunktverteilung. Hier ist ersichtlich, dass die z-Koordinate des KP1 die einzige Variable ist, deren Änderung eine Verbesserung der Strukturantwort nach sich zieht. Veränderungen an den Punkten 5, 9 und 13, alle am linken Bauteilrand, haben minimal verschlechternde Auswirkungen, während die Punkte 14, 15 und 16 (obere Bauteilkante) die Strukturantwort stark negativ beeinflussen. Die restlichen Punkte beeinträchtigen die Strukturantwort etwa halb so stark wie die letztgenannten Punkte. Da jedoch angesichts der mathematischen Formulierung der Bézier-Fläche starke Wechselwirkungen zwischen den Kontrollpunkt-Variablen erwartet werden, macht es auf Grundlage der gewonnenen Informationen keinen Sinn, Variablen zu deaktivieren. Die Differenzen der Kontrollpunkte am linken Bauteilrand (1, 5, 9 und 13) lassen schlussfolgern, dass diese das größte Potential haben, die Auslegung positiv zu beeinflussen. Dies gilt speziell für Kontrollpunkt 1.

Tabelle 6.3.: Differenzen der Strukturantwort

-0,0026	-0,6131	-0,6502	-0,6513
-0,0154	-0,2589	-0,2365	-0,2859
-0,0073	-0,2329	-0,1861	-0,1689
0,0052	-0,3055	-0,2387	-0,2136

## 7. Demonstration des Optimierungstools

Nachdem im letzten Kapitel die Entwicklung des Optimierungsprogramms abgeschlossen wurde, wird dieses nun praktisch angewandt. Zunächst wird die Festigkeitsoptimierung der gelochten Platte mit verschiedenen Krümmungsrestriktionen durchgeführt. Im Anschluss erfolgt eine Parameterstudie zur Prüfung, welchen Einfluss die Anzahl und Positionierung der Kontrollpunkte auf das Optimierungsergebnis haben. Um die dargestellten Ergebnisse einordnen zu können, sollen die in Kapitel 4.3 gewonnenen Vergleichswerte helfen. Ein  $[0^\circ \pm 45^\circ 90^\circ]_s$ -Laminat weist einen maximalen Tsai-Wu-Index von 0,0894 auf und der eines reinen  $0^\circ$ -Lagen-Laminats liegt bei 0,0475. Eine abschließende Bewertung der Ergebnisse erfolgt in Kapitel 8. Alle relevanten Diagramme der Untersuchungen sind in Anhang X aufgeführt.

### 7.1. Festigkeitsbetrachtung einer gelochten Platte

Das Ziel ist, die Festigkeit einer gelochten Platte unter Berücksichtigung der Herstellbarkeit zu optimieren. Hierzu wird eine Mehrzieloptimierung mit der Methode der restriktionsorientierten Transformation simuliert. Diese erfordert eine Festlegung unterschiedlicher Krümmungsrestriktionen und getrennte Optimierungen mit diesen. Es werden 13 Werte festgelegt und eine 14. Untersuchung ohne Restriktion durchgeführt. Dies erfolgt mit dem Ziel einen PARETO-optimalen Rand darstellen zu können. Die einzelnen Optimierungen erfolgen mit dem entwickelten Tool. Hierbei dient zur Bestimmung der Winkel eine Bézier-Fläche mit 16 Kontrollpunkten und ebener Startfläche. Die Fläche wird mit einem Feinheitwert von 100 unterteilt und, um bestmögliche Ergebnisse zu erhalten, wird jede Optimierung mit verschiedenen Gradienten-Deltas wiederholt.

#### 7.1.1. Unrestringierte Optimierung

Als Ausgangspunkt wird die unrestringierte Optimierung durchgeführt. Die Optimierung ergab nach 400 Iterationen optimierte Faserverläufe, bei denen der maximale Tsai-Wu-Index 0,0164 beträgt. Dieser tritt in Zelle 105 auf, die mit etwas Abstand zum Lochrand auf der  $45^\circ$ -Achse des Kreisbogens liegt (siehe Abbildung 4.6). Der Wert  $\Phi_{max} = 0,0164$  wird als

## 7. Demonstration des Optimierungstools

Referenzwert gewählt, auf den im Folgenden bei weiteren Optimierungen Bezug genommen wird. Die Optimierungsergebnisse werden zur beispielhaften Beschreibung der einzelnen visuellen Ausgaben genutzt. In Bild 7.1 sind die optimierten Winkel im Lochbereich dargestellt. Die Darstellung beinhaltet der Übersichtlichkeit halber nur die Winkel jeder zweiten Zelle. Es ist erkennbar, dass die Fasern im oberen Bereich horizontal verlaufen, während die Fasern auf Lochhöhe in ein Laminat mit gekreuzten Winkeln ( $\pm 30^\circ < \text{Winkel} < \pm 60^\circ$ ) übergehen. Dies ist logisch nachzuvollziehen, da im ungestörten Bereich Zugspannungen vorherrschen, während hinter dem Loch auch relevante Druckspannungen entstehen.

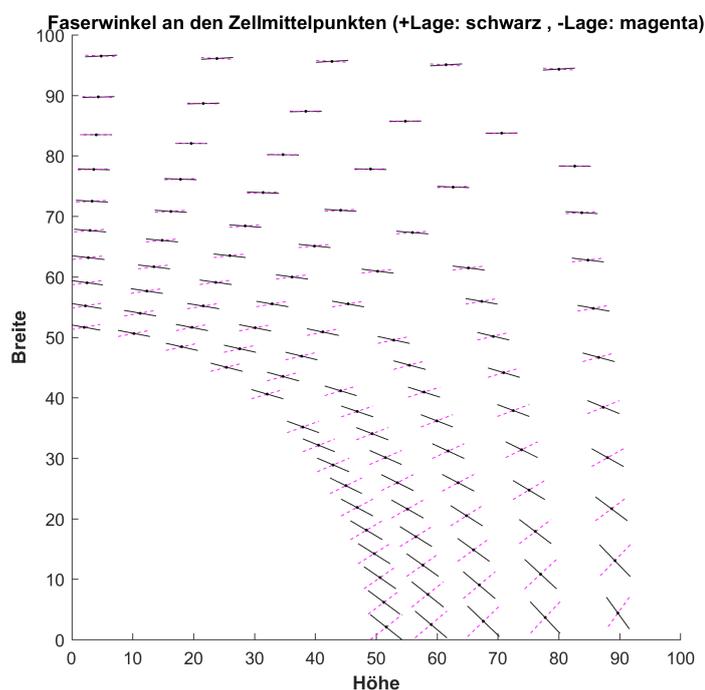


Abbildung 7.1.: Faserwinkel im Lochbereich bei unrestringierter Optimierung

In Abbildung 7.2 sind die Höhenschnittlinien, aus denen die Faserwinkel und Krümmungswerte resultieren, visualisiert. Die Darstellung muss mit dem Wissen betrachtet werden, dass aus ihr nicht direkt auf entstehende Winkel geschlossen werden kann, da nicht ersichtlich ist, welche HSL zu welchem Zellmittelpunkt gehört. Dennoch geben die Verläufe einen Hinweis auf die allgemeinen Verlaufsrichtungen. So sind die Rundung um den Lochbereich sowie die geraden Verläufe oberhalb des Loches gut erkennbar. Durch den gewählten Bereich der KS-Funktion versucht der Algorithmus gezielt dieses Gebiet zu optimieren. Im mittleren Bereich

## 7. Demonstration des Optimierungstools

---

sind die Kurvenverläufe ersichtlich, an denen die Problematik der konfusen Winkel entsteht (vgl. Kapitel 5.7.1 Unterpunkt *Bézier-Fläche*). Im hinteren Bereich verlaufen die Linien wieder horizontal, was aus logischer Sicht sinnvoll, für den Optimierungsablauf aber irrelevant ist, da die Winkel in diesem Bereich nicht in die Ergebnisse einfließen.

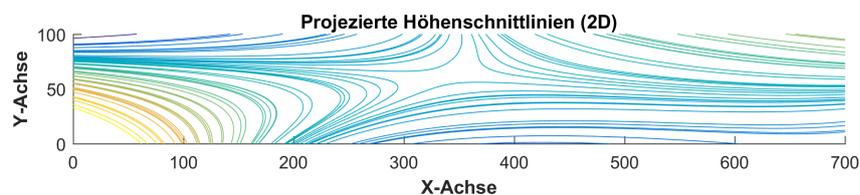


Abbildung 7.2.: Höhenschnittlinien bei unrestringierter Optimierung

Das dritte Bild veranschaulicht die dreidimensionale Bézier-Fläche, aus der die eben besprochenen Höhenschnittlinien resultieren. Während des Optimierungsprozesses ist diese Darstellung interessant, um zu überblicken, in welchem Bereich der Algorithmus aktuell Änderungen vornimmt. Nach Beendigung der Optimierung gibt sie Informationen darüber, wo die größten Modifikationen vorgenommen wurden. Daraus lässt sich ableiten, welche Kontrollpunkte für die Optimierung besonders wichtig sind. Wie zu erwarten sind es bei der untersuchten Struktur der Bereich der Aussparung und somit die dem Gebiet naheliegenden Kontrollpunkte. Zudem ist die zu Problemen führende Anhöhe dezent erkennbar. Sie verläuft diagonal von der Anhöhe im Lochbereich (0/0) in die obere Geometriemitte (350/100).

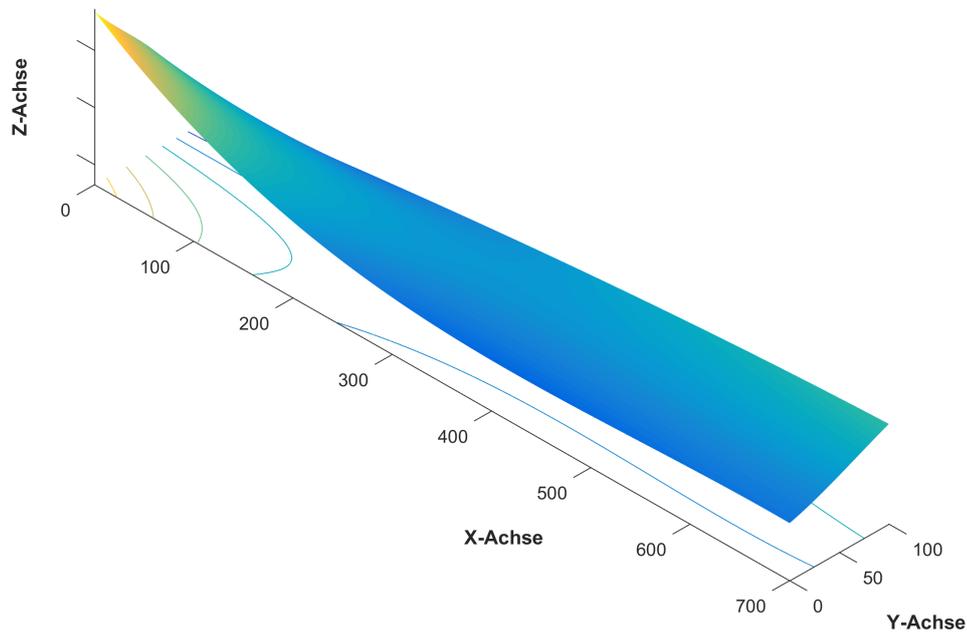


Abbildung 7.3.: Bézier-Fläche bei unrestringierter Optimierung

Das letzte Darstellung (7.4) zeigt die Winkel im hinteren, lochfernen Bereich. Hier sind die erwähnten konfusen Winkel zu erkennen, die aus der Anhöhe der Bézier-Fläche resultieren. Aus logischer Sicht müssten die Fasern im oberen und im Einspannungsbereich reine  $0^\circ$ -Lagen aufweisen. Das kreuzförmig angeordnete Laminat im Nachlaufbereich des Lochs müsste somit in Richtung der Einspannung kontinuierlich in  $0^\circ$ -Lagen übergehen.

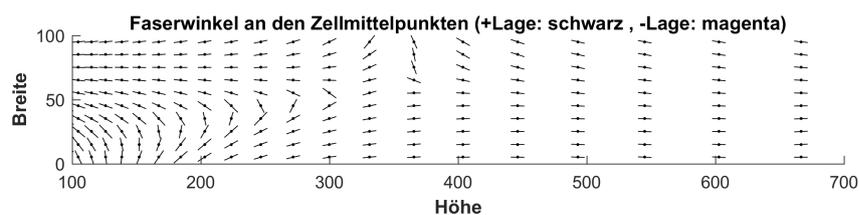


Abbildung 7.4.: Faserwinkel im hinteren Bereich bei unrestringierter Optimierung

Um zu veranschaulichen wie die Spannungsverläufe sich durch die veränderten Faserwinkel verändern, werden die optimierten Spannungsverläufe denen des  $0^\circ$ -Lagen-Laminats gegenübergestellt. In Abbildung 7.5 sind die Spannungsverläufe in Faserrichtung (x-Richtung) im

## 7. Demonstration des Optimierungstools

Vergleichslaminat (links) und im optimierten Entwurf (rechts) dargestellt. Analog dazu sind in Bild 7.6 die Spannungen quer zur Faserrichtung (y-Richtung) aufgezeigt.

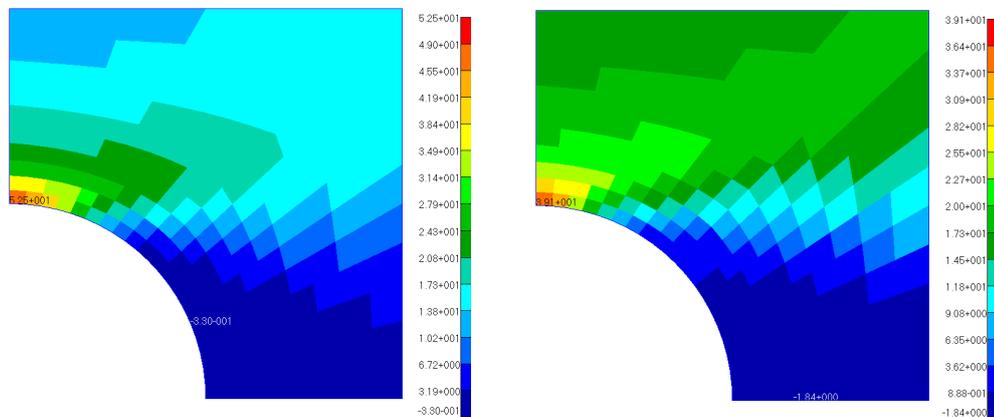


Abbildung 7.5.: Spannungsverläufe in Faserrichtung

Der Spannungsverlauf in Faserrichtung ändert sich nicht wesentlich, allerdings verändern sich die Spannungswerte, was zur veränderten Farbgebung führt (Einstellungsfehler). Die Druckspannungen in Faserrichtung variieren nicht wesentlich, die Zugspannungen nehmen hingegen um etwa ein Viertel ab.

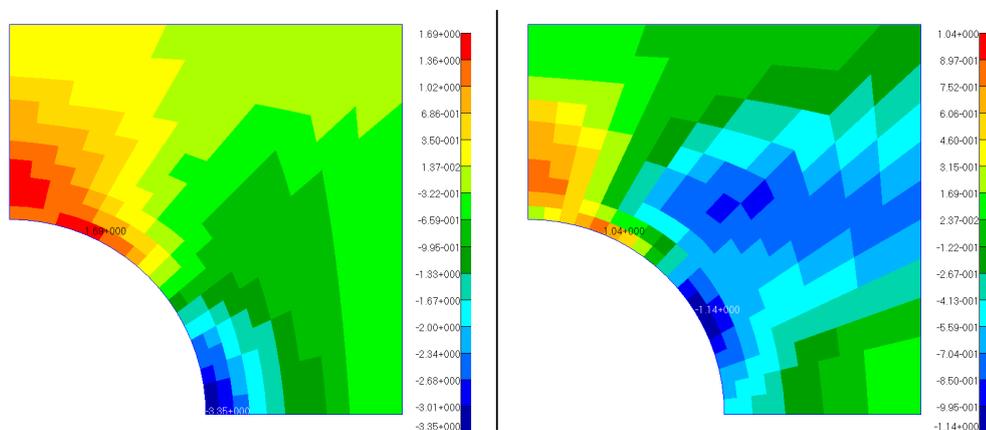


Abbildung 7.6.: Spannungsverläufe quer zur Faserrichtung

Quer zur Faserrichtung treten mehr Veränderungen auf, was hinsichtlich der deutlich schwächeren Materialeigenschaften in dieser Richtung logisch ist. Während die Zugspannungen um etwa 40 Prozent reduziert werden, erfolgt bei den Druckspannungen eine Minderung um

rund 66 Prozent. Der Punkt der höchsten Druckspannung verschiebt sich zudem vom rechten Kerbgrund entlang des Lochrands nach oben. Zusätzlich findet im Nachlaufbereich des Loches ein Wechsel von geringen Druckspannungen zu minimalen Zugspannungen statt.

### 7.1.2. Restringierte Optimierung

Der gleiche Optimierungsvorgang wird mit den 13 definierten Krümmungsrestriktionen durchgeführt. Die Ergebnisse der Berechnungen sind in Tabelle 7.1 zusammengefasst. Hierbei sind zu den Restriktionswerten  $\kappa$  jeweils der optimierte maximale Tsai-Wi-Index, die dazugehörige Zelle, die tatsächliche Durchschnittskrümmung und die Anzahl der benötigten Iterationen aufgeführt. Die Werte der maximalen Tsai-Wu-Indizes sind die tatsächlichen Werte und nicht die mittels Kreisselmeier-Steinhauser-Funktion zusammengefassten Werte.

Tabelle 7.1.: Vergleich restringierter Optimierungen

Optimierung	1	2	3	4	5	6	7
$\kappa_{restr.} [\frac{1}{mm}]$	1	0,07	0,06	0,05	0,04	0,03	0,025
$\Phi_{max}$	0,0164	0,0208	0,0169	0,0111	0,0129	0,0123	0,0201
Zelle $\Phi_{max}$	105	60	91	71	69	69	60
$\kappa_{real} [\frac{1}{mm}]$	0,0094	0,0415	0,0329	0,0168	0,0330	0,0210	0,0199
Iterationen	400	196	423	623	779	339	123
Optimierung	8	9	10	11	12	13	14
$\kappa_{restr.} [\frac{1}{mm}]$	0,022	0,02	0,018	0,01	0,008	0,005	0,003
$\Phi_{max}$	0,0155	0,0343	0,0152	0,0209	0,0203	0,0222	0,0293
Zelle $\Phi_{max}$	70	171	150	191	95	95	161
$\kappa_{real} [\frac{1}{mm}]$	0,0127	0,0040	0,0158	0,0122	0,0072	0,0082	0,0055
Iterationen	566	67	392	132	140	65	214

Die Daten zeigen, dass die Restriktionen großen Einfluss auf die Ergebnisse haben. Allerdings ist dieser Einfluss wider Erwarten bis zu einem gewissen Wert positiv. So wird das beste Ergebnis ( $\Phi_{max} = 0,0111$ ) bei  $\kappa_{restr.} = 0,05$  erreicht und selbst bei einer Restriktion von  $\kappa_{restr.} = 0,018$  wird bezogen auf den Referenzwert ein kleineres Optimum gefunden ( $\Phi_{max} = 0,0155$ ). Das liegt daran, dass bei diesen Werten die Restriktionen noch nicht erreicht werden und keine geradlinigeren Faserverläufe gewählt werden müssen. Dieser Umstand ist an den teils deutlichen Abständen zwischen dem Restriktionswert und dem tatsächlichen Krümmungswert erkennbar. Die Intention, die Krümmung so weit zu restringieren, dass andere Entwürfe entstehen, wird erst bei darunterliegenden Werten erreicht. Durch starke Sprünge

der durchschnittlichen Krümmung findet der Algorithmus vom Startentwurf aus bei kleinen Restriktionen keine Entwürfe im zulässigen Bereich und bricht daher ab. Diesem Problem kann mittels eines vorgegebenen Startentwurfs entgegengewirkt werden. Hierbei wird der Startentwurf verwendet, der auch später in Kapitel 7.2.4 für Kombination 3 genutzt wird. Dort sind auch weitere Informationen zu dem Thema beschrieben. Bei der maximalen Restriktion wird nur noch ein  $\Phi_{max} = 0,0293$  erreicht. Die dazugehörigen Winkel sind in Bild 7.7 dargestellt. Im Vergleich mit Bild 7.1 ist die deutlich geminderte Variabilität der Faserwinkel zu erkennen. In den Optimierungen 11, 13 und 14 werden die vorgegebenen Restriktionen überschritten.

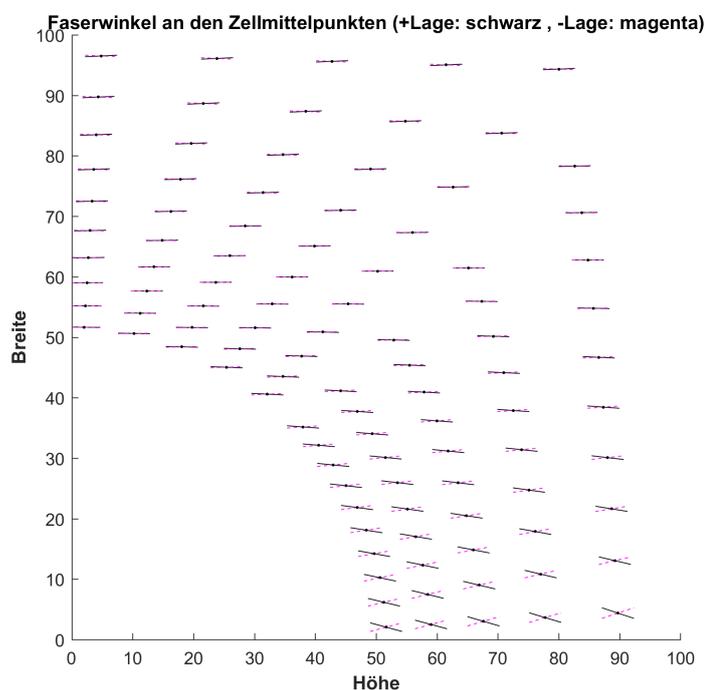


Abbildung 7.7.: Faserwinkel im Lochbereich bei restringierter Optimierung

Die Zahl der Iterationen schwanken stark und es ist kein Zusammenhang zwischen Restriktion und Optimierungsdauer erkennbar. Für den besten Entwurf braucht der Optimierungsprozess mit 623 Iterationen etwa 50% länger (Optimierung 4) als bei der unrestringierten Suche (Optimierung 1). Das zweitbeste Ergebnis (Optimierung 6) wird in 339 Iterationen ermittelt, womit die Rechnung schneller als die unrestringierte ist. In Optimierung 7 wird bereits nach deutlich weniger Iterationen (123) ein verhältnismäßig gutes Optimum gefunden. Dieses reicht jedoch nicht an die unrestringierte Lösung heran. Optimierung 13 ist mit 65

Iterationen die schnellste, überschreitet dabei aber die vorgegebene Restriktion. Die geringste durchschnittliche Krümmung erzeugt Optimierung 9, allerdings ist das gefundene Minimum verglichen mit dem optimalen Ergebnis etwa um den Faktor drei größer.

Die Zellen, in denen der jeweils maximale Tsai-Wu-Index auftritt, variieren leicht. Die meisten liegen in dem gleichen Gebiet wie bei der unrestringierten Lösung. Bei den Optimierungen 3, 4, 12 und 13 tritt der Maximalwert hingegen oberhalb des Loches und bei den Optimierungen 9,11 und 14 am rechten Lochrand auf. Einziger Ausreißer ist Optimierung 10, bei der die kritischste Zelle in der vertikalen Strukturmitte am rechten Rand des Lochbereiches liegt. Der Wert hierbei ist jedoch nur unwesentlich größer als die Indizes nahe des Loches.

### 7.1.3. Auswertung der Mehrzieloptimierung

Das ursprüngliche Ziel, einen kontinuierlichen PARETO-Rand mit definierten Krümmungswerten zu erzeugen, wird mit den gewonnenen Ergebnissen nur bedingt erreicht, da bei den meisten Optima die Restriktionen noch nicht aktiv sind. In Diagramm 7.8 sind die maximalen Tsai-Wu-Werte aller Optimierungen über der zugehörigen Durchschnittskrümmung aufgetragen. Aus den Ergebnissen der Optimierungen x,y, und z entsteht ein dünn besetzter PARETO-Rand. Um den Rand definierter auszuarbeiten, müssen deutlich mehr Optimierungen durchgeführt werden.

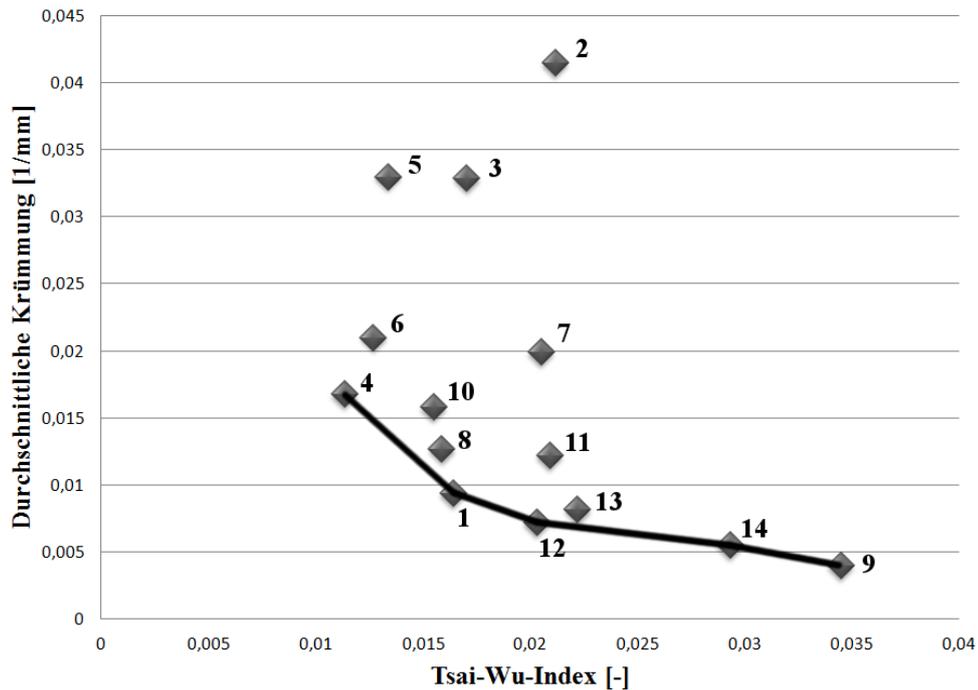


Abbildung 7.8.: Ergebnisse der restringierten Optimierungen

## 7.2. Parameterstudie zu Kontrollpunkten

Die bisherigen Optimierungen wurden mit den zu Beginn des Kapitels 7.1 beschriebenen Einstellungen durchgeführt. Mittels einer Parameterstudie soll nun geprüft werden, ob veränderte Einstellungen allgemein das Auffinden besserer Optima ermöglichen. Diese Studie unterteilt sich in vier Bereiche, die sich mit drei unterschiedlichen Einstellungsmöglichkeiten sowie der Kombination ebendieser befassen. Die zu untersuchenden Einstellungsmöglichkeiten sind die Startwerte der z-Variablen, die Anzahl sowie die x-y-Positionierung der Bézier-Kontrollpunkte.

### 7.2.1. z-Koordinaten der Kontrollpunkte

In Kapitel 3.2.6 wurde darauf eingegangen, dass Zielfunktionen mehrere Minima haben können, wenn ein nicht-konvexes Problem vorliegt. Die Konvexität des vorliegenden Optimierungsproblems ist nicht zu prüfen, jedoch wird aufgrund des indirekten Ansatzes der Variablenerzeugung angenommen, dass es nicht-konvex ist. Aus diesem Grund werden mit dem Optimierungsprozess verschiedene Startentwürfe, die von der bisher verwendeten Ebene abweichen, getestet. Mit vier verschiedenen Ansätzen werden hierzu die Startwerte der z-Koordinaten gezielt ver-

ändert. Die daraus resultierenden optimierten Tsai-Wu-Indizes sowie die dafür notwendigen Iterationen sind in Tabelle 7.2 zusammengefasst. In Anhang A befinden sich die entsprechenden Höhenschnittlinien- und Winkeldarstellungen.

Tabelle 7.2.: Startvariationen der z-Koordinaten

<b>Ansatz</b>	1	2	3	4
$\Phi_{max}$	0,019	0,0185	0,026	0,0261
<b>Iterationen</b>	141	436	115	97

Die Aussage der Sensitivitätsanalyse in Kapitel 6.4 ist, dass speziell der Kontrollpunkt 1 positiven Einfluss auf die Strukturantwort hat. Daher wird zunächst eine Startfläche, bei der nur dieser KP von der Ebene abweicht, untersucht (Ansatz 1). Der Algorithmus findet bereits nach rund 50 Iterationen Auslegungen mit einem maximalen Tsai-Wu-Wert um 0,02. Nach insgesamt 141 Iterationen liegt das Optimum bei 0,019. Bzgl. der Rechendauer ist die Sensitivitätsanalyse und der daraus entstandene Ansatz somit hilfreich, das gefundene Optimum erreicht jedoch nicht den gesetzten Referenzwert. Der zweite Ansatz verfolgt die gleiche Idee, jedoch ist der Startentwurf eine ebene Fläche, die kontinuierlich in Richtung KP1 ansteigt. Hierbei ist bereits der Startentwurf mit einem  $\Phi_{max} = 0,0373$  besser als der Startentwurf mit  $0^\circ$ -Lagen. Allerdings braucht die Optimierung mit 436 Iterationen deutlich länger als der erste Ansatz und erzielt dabei nur ein unwesentlich besseres Ergebnis ( $\Phi_{max} = 0,0185$ ). Beide Ansätze helfen folglich nicht, ein verbessertes Optimum zu finden.

Die dritte Startflächen-Variation sieht eine Anhebung der beiden mittleren KP-Reihen vor, woraus eine in Längsrichtung nach oben gewölbte Fläche entsteht. Zwar resultieren im ersten Entwurf aus dieser ebenfalls  $0^\circ$ -Lagen, jedoch haben die Variablenänderungen in den folgenden Iterationen andere Auswirkungen. Mit einem  $\Phi_{max} = 0,026$  ist das gefundene Optimum jedoch verhältnismäßig groß und dabei mit 115 Iterationen nur unwesentlich schneller als der erste Ansatz. Erwähnenswert ist, dass eine stark verformte Fläche mit einer durchschnittlichen Krümmung von 0,935 entsteht. Somit bringt diese Startfläche keine Vorteile.

Als letzte Variation (Ansatz 4) wird eine zufällig generierte Matrix (siehe Formel 7.1) für die z-Koordinaten genutzt. Dies dient nicht dem Ziel, ein verbessertes Optimum zu finden, sondern zu prüfen, ob das Programm auch aus einer ungeordneten Fläche zu einem sinnvollen Entwurf findet. Die Startfläche ist in Bild 7.9 dargestellt. Um trotz der signifikanten Unter-

schiede der Startwerte Tendenzen zu finden, wird mit  $1e^{-2}$  ein entsprechend großes Delta zur Gradientenbestimmung gewählt. Mit diesem ist es dem Optimierungsalgorithmus in nur 97 Iterationen möglich, eine deutliche Verbesserung des maximal auftretenden Tsai-Wu-Indexes zu erreichen. Zwar ist das Optimum  $\Phi_{max} = 0,0261$  verglichen mit dem Referenzwert groß, hinsichtlich des Ausgangswertes bei Optimierungsstart ( $\Phi_{max} = 0,1208$ ) aber eine deutliche Verbesserung. Dass der Wert nicht feiner ausgearbeitet wird, liegt am großen Gradienten-Delta. Ein anschließender, zweiter Optimierungsprozess mit entsprechend kleinerem Delta könnte das Ergebnis verfeinern.

$$z\_Start\_frei = \begin{bmatrix} 5.18 & 5.21 & 5.75 & 5.13 \\ 5.36 & 5.91 & 5.74 & 5.21 \\ 5.06 & 5.68 & 5.56 & 5.89 \\ 5.52 & 5.47 & 5.18 & 5.07 \end{bmatrix} \begin{pmatrix} KP1 & KP2 & KP3 & KP4 \\ KP5 & KP6 & KP7 & KP8 \\ KP9 & KP10 & KP11 & KP12 \\ KP13 & KP14 & KP15 & KP16 \end{pmatrix} \quad (7.1)$$

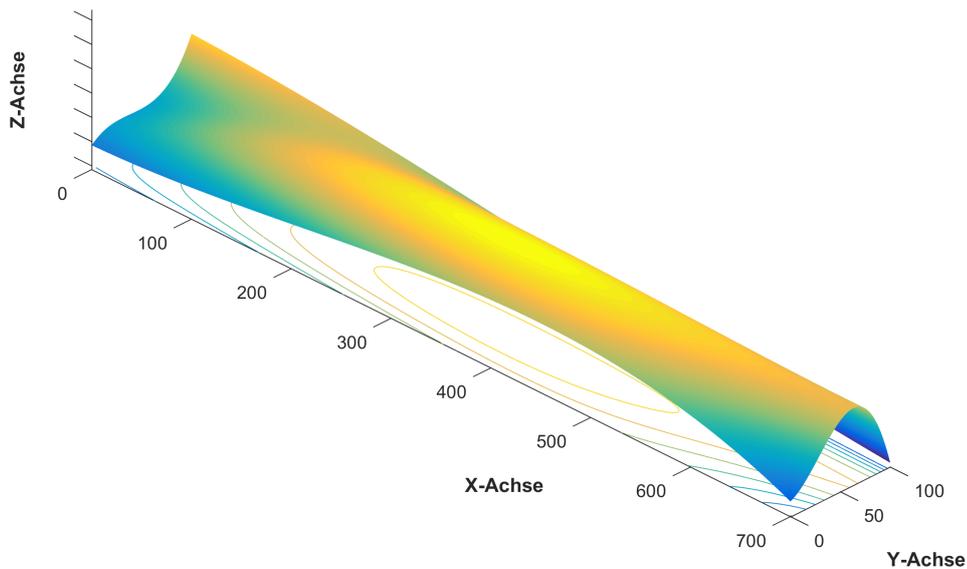


Abbildung 7.9.: Bézier-Fläche bei zufälligen Startwerten

### 7.2.2. Anzahl der Kontrollpunkte

Da die Bézier-Fläche mit mehr Kontrollpunkten noch variabler gesteuert werden kann, soll geprüft werden, ob sich diese zusätzliche Variabilität positiv auf die Optimierungsergebnisse

auswirkt. Zu diesem Zweck wird eine Parameterstudie zu unterschiedlichen Kontrollpunktanzahlen durchgeführt. Aus drei unterschiedlichen Möglichkeiten (4, 5 oder 6 KP) pro Richtung ergeben sich neun verschiedene Konstellationen. Die Optimierungen erfolgen hierbei unrestringiert und mit einer ebenen Startfläche. Bei Kombinationen mit fünf Kontrollpunkten pro Seite wird die Feinheit auf 200-400 angehoben, während zur Optimierung mit sechs Kontrollpunkten im Idealfall 800 als Einstellung gewählt wird. Bei zu kleinen Feinheits-Werten kann es hier zu Prozessabbrüchen aufgrund zu weniger Punkte für die Interpolation kommen. Die Ergebnisse sind in Tabelle 7.3 dargestellt. Die erste Zahl gibt die Anzahl der Kontrollpunkte in vertikaler Richtung an, die zweite in horizontaler.

Tabelle 7.3.: Ergebnisvergleich unterschiedlicher KP-Anzahlen

<b>Kombination</b>	4x4	4x5	5x4	5x5	5x6	6x6
<b>Variablenanzahl</b>	16	20	20	25	30	36
$\Phi_{max}$	0,0164	0,0243	0,0246	0,218	0,0187	0,0338
<b>Iterationen</b>	400	98	309	182	518	221

Die Ergebnisse zeigen, dass die Nutzung von mehr Kontrollpunkten kontraproduktiv ist. Die gefundenen Minima sind teils deutlich größer als der Referenzwert. Die Optimierungen benötigen zwar wider Erwarten weniger Iterationen, allerdings dauern diese aufgrund der feineren Auflösung der Bézier-Fläche länger. Speziell bei der Untersuchung mit 6x6 Kontrollpunkten fällt dieser Umstand ins Gewicht, da durch die Feinheit von 800 etwa die vierfache Rechendauer pro Iteration benötigt wird. Somit dauert die gesamte Optimierung etwa doppelt so lange wie die der 4x4-Kombination. Lediglich die Konstellation 5x6 liefert mit einem  $\Phi_{max} = 0,0187$  ein ansatzweise vergleichbares Resultat. Hierzu benötigt der Optimierungsprozess jedoch 518 Iterationen, wodurch die Konstellation in Anbetracht der beschriebenen Rechendauer pro Iteration ebenfalls unpraktikabel ist. Zu den Konstellationen 4x6, 6x4 und 6x5 findet das Optimierungstool mit den genutzten Einstellungen keine sinnvollen Entwürfe.

Theoretisch erhöhen zusätzliche Kontrollpunkte die Flexibilität der Bézier-Fläche. Allerdings erschwert diese Flexibilität dem Optimierungsalgorithmus, die Auswirkungen einzelner Kontrollpunkte zu erkennen. Dadurch erhöht sich die Gefahr gegen lokale Minima zu konvergieren. Dieses Phänomen wird anhand Abbildung 7.10, in der die mit 6x6 Kontrollpunkten optimierten Winkel dargestellt sind, verdeutlicht. Der Algorithmus stellt den positiven Effekt gekreuzten Lagen im unteren Bereich sowie den der horizontal verlaufenden Fasern im oberen und middle-

ren Bereich fest. Zu deren Erzeugung hebt er die entsprechenden Kontrollpunkte an, was eine Verbesserung der Strukturantwort nach sich zieht. Allerdings entstehen auf diese Weise in der Bézier-Fläche Täler zwischen den Punkten, aus denen wiederum die beiden Orientierungswechsel im Bereich um  $y = 15$  und  $y = 75$  resultieren. Der gradientenbasierte Algorithmus erkennt nicht die Notwendigkeit eines Richtungswechsels der mittleren Kontrollpunkte und optimiert die Konstellation so gut wie möglich weiter. Allerdings sind die Winkel im unteren Bereich durch den notwendigen Orientierungswechsel begrenzt, wodurch im relevanten Gebiet am rechten Kerbgrund (Zelle 191) der Maximalwert  $\Phi_{max} = 0,0338$  entsteht und der Entwurf nicht weiter verbessert werden kann. Auf diese Thematik und den Umstand, dass bei manchen Kombinationen keine Verbesserung eintritt, wird in Kapitel 8.1 nochmals eingegangen.

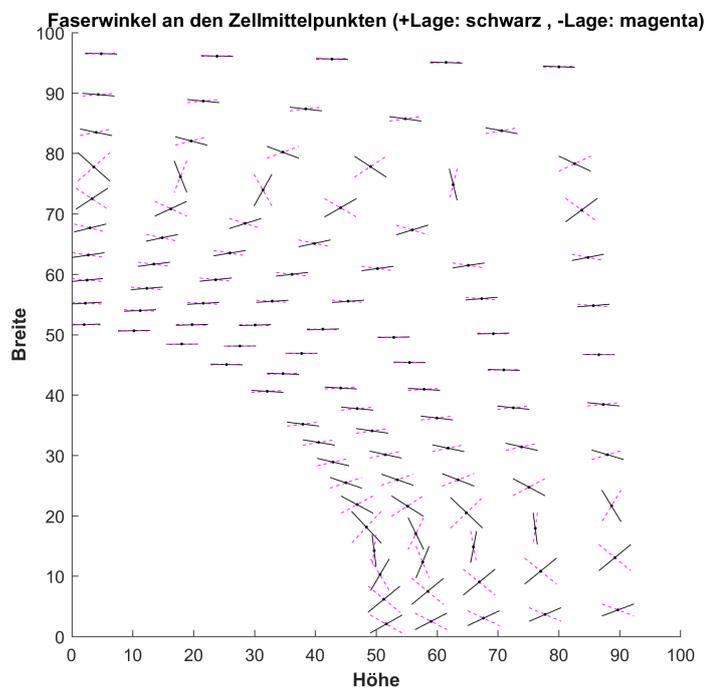


Abbildung 7.10.: Optimierte Winkel im Lochbereich bei 6x6 Kontrollpunkten

### 7.2.3. Position der Kontrollpunkte

Nachdem in Kapitel 7.2.1 bereits der Einfluss unterschiedlicher z-Startkoordinaten getestet wurde, wird nun geprüft, ob die Verschiebung der KP-Spalten und Reihen in x- bzw. y-Richtung

Vorteile bringt. Dies kann einerseits in Form von zusätzlichen Optimierungsvariablen erfolgen, andererseits durch manuelle Positionierung.

### Variable KP

Die Option des Optimierungstools, auch die x- und y-Koordinaten einiger Kontrollpunkte als Variablen deklarieren zu können, schafft die Möglichkeit, zu prüfen, wo die Kontrollpunkte am sinnvollsten sind. Hierzu werden unter Verwendung einer Bézier-Fläche mit 16 Kontrollpunkten die Variabilität zuerst getrennt in x- und y-Richtung getestet, anschließend die Kombination beider Optionen. Die Ergebnisse inklusive der optimierten Kontrollpunkt-Positionen sind in Tabelle 7.4 dargestellt. Die dazugehörigen Höhenschnittlinien- und Winkeldarstellungen befinden sich in Anhang B.

Tabelle 7.4.: Ergebnisse variabler Kontrollpunkte

Rechnung	1	2	3
<b>Variablen</b>	KP x	KP y	KP x & KP y
$\Phi_{max}$	0,0175	0,0130	0,0145
<b>Positionen in x [mm]</b>	367; 700	automatisch	345; 700
<b>Positionen in y [mm]</b>	automatisch	50,3; 100	81,3; 100
<b>Iterationen</b>	299	544	254

Beide Optimierungen, in denen die Kontrollpunkte in y-Richtung variabel sind, finden Optima, die den Referenzwert von 0,164 unterbieten. Aber auch die erste Rechnung findet ein dem Referenzwert ähnliches Minimum und das mit einem 25 % geringeren Rechenaufwand. Mit einem ( $\Phi_{max} = 0,0130$ ) erzeugt Rechnung 2 die beste Auslegung, indem der Algorithmus die zweite Kontrollpunktreihe in die Strukturmitte legt, während die dritte auf die vierte verschoben wird. Somit wird die Bézier-Fläche in y-Richtung effektiv nur noch mit drei Kontrollpunkten aufgespannt. In horizontaler Richtung entsteht ein ähnlicher Ansatz. Die zweite Kontrollpunktspalte wird nahe der Bauteilmitte definiert und die dritte zur vierten auf den Bauteilrand verlegt. Auch wenn beide Richtungen variabel sind, verschiebt der Algorithmus eine Reihe und eine Spalte auf die Flächengrenzen. Während in horizontaler Richtung die verbleibende Spalte wieder in der Strukturmitte positioniert wird, liegt die verbleibende Kontrollpunktreihe in diesem Fall deutlich über der Mitte. Trotz der erhöhten Variablenanzahl sinkt die Zahl der Iterationen. Im Vergleich benötigt die zweite Rechnung mehr als die doppelte Zeit zur Findung des Optimums.

Der Vergleich der Rechnungen 2 und 3 ist ein anschauliches Beispiel für die Problematik lokaler Minima. Logisch betrachtet müsste Rechnung 3, indem die Kontrollpunkte in horizontaler Richtung nicht verschoben werden, mindestens das Ergebnis der zweiten Optimierung erreichen. Der positive Effekt auf die Strukturantwort durch Verschieben der Kontrollpunktspalten ist jedoch zunächst größer, wodurch der Algorithmus diese verlagert. Eine anschließende Variation der y-Koordinaten führt nun allerdings zu einem anderen Minimum, aus welchem der gradientenbasierte SQP-Algorithmus nicht mehr zurückfindet.

### Manuelle Positionierung

Nach Prüfung mittels variabler Kontrollpunkte können die Koordinaten der Bézierpunkte in x- und y-Richtung für weitere Rechnungen auf statische Werte festgelegt werden. Alternativ kann dies auch anhand logischer Vorüberlegungen, in welchem Bereich eine erhöhte Kontrollpunktzahl tendenziell sinnvoll ist, geschehen. Auf diese Weise entfällt die äquidistante Verteilung und es können andere Bézier-Flächen entstehen. Anhand zweier Rechnungen soll untersucht werden, ob mit manueller Positionierung optimierungstechnische Vorteile entstehen. Da vermutet wird, dass die Kontrollpunkte im Lochbereich eine variabelere Gestaltung der Faserverläufe ermöglichen, werden diese manuell dorthin verschoben. Die Resultate sind in Tabelle 7.5 aufgeführt.

Tabelle 7.5.: Ergebnisse manueller Positionierung

<b>Kombination</b>	KPx definiert	KPy definiert
$\Phi_{max}$	0,0252	0,0161
<b>Positionen in x [mm]</b>	50; 100	<i>automatisch</i>
<b>Positionen in y [mm]</b>	<i>automatisch</i>	25; 50
<b>Iterationen</b>	132	199

Die manuelle Positionierung in vertikaler Richtung auf Höhen von 25 mm und 50 mm führt zu guten Ergebnissen. Mit einem  $\Phi_{max} = 0,0161$  wird der Referenzwert marginal unterboten und dazu nur die Hälfte an Iterationen benötigt. Vorüberlegungen hinsichtlich der y-Positionen haben dementsprechend Potential, den Optimierungsprozess zu verbessern. Die manuelle Veränderung des Kontrollpunktrasters in horizontaler Richtung hat hingegen negative Folgen, denn das gefundene Optimum ist mit 0,0252 signifikant größer als der Referenzwert. Eine zusätzliche Betrachtung mit Kontrollpunktspalten bei 50 mm und 350 mm ergibt keine Ergebnisse. Eine zusätzliche Prüfung mit sowohl in x- als auch in y-Richtung definierten Kontrollpunkten (bei jeweils 50 mm und 75 mm) ergibt ebenfalls kein Optimierungsergebnis.

Dies liegt vermutlich darin begründet, dass durch die Enge der Kontrollpunkte der Algorithmus keine eindeutigen Tendenzen erkennt.

#### 7.2.4. Kombination

Die beschriebenen Möglichkeiten, die Bézier-Fläche und Startentwürfe variabler zu gestalten, können kombiniert werden. Da sich hierbei unzählige Konstellationen ergeben, werden vier ausgewählt, die als besonders sinnvoll oder prüfenswert erachtet werden. Diese werden nachfolgend beschrieben und die dazugehörigen Ergebnisse in Tabelle 7.6 präsentiert. In Anhang C sind die zugehörigen Höhenschnittlinien- und Winkeldarstellungen angefügt.

Tabelle 7.6.: Ergebnisse Kombinationen

<b>Kombination</b>	1	2	3	4
$\Phi_{max}$	0,0223	0,0125	0,014	0,0132
<b>Positionen in x [mm]</b>	141; 281; 422; 563	<i>autom.</i>	50; 75	<i>autom.</i>
<b>Positionen in y [mm]</b>	20,3; 40,5; 60,4; 80,4	96,7; 100	50; 75	<i>autom.</i>
<b>Iterationen</b>	316	203	209	1019

Die erste Untersuchung gilt einer Bézier-Fläche mit der Maximalzahl an Kontrollpunkten (36), die zusätzlich in x- und y-Richtung variabel sind (Kombination 1). Somit sind insgesamt 40 Variablen zu optimieren. Da hierbei eine Feinheit von 800 genutzt wird, ist die Optimierung die insgesamt rechenintensivste aller durchgeführten Untersuchungen. Die optimierten Positionen der Kontrollpunktspalten und -reihen sind in Tabelle 7.6 aufgeführt. In Anbetracht einer Schrittweite von 140 mm in x-Richtung und 20 mm in y-Richtung bei äquidistanter Verteilung, ist erkennbar, dass die x- und y-Positionen nur minimal verändert sind. Die Optimierung führt zu einem  $\Phi_{max} = 0,0223$  und ist somit in Anbetracht des notwendigen Rechenaufwandes deutlich schlechter als die Referenzoptimierung.

Als zweites wird eine Kombination einiger Parameter, die in den unterschiedlichen Studien zu den besten Ergebnissen geführt haben, gewählt (Kombination 2). Somit wird eine Bézier-Fläche mit 4x4 Kontrollpunkten, variable Kontrollpunkte in y-Richtung und eine Krümmungsrestriktion von 0.05 verwendet. Diese Konstellation führt mit 203 Iterationen überdurchschnittlich schnell zu einem Optimum, das signifikant besser als der Referenzwert ist ( $\Phi_{max} = 0,0125$ ). Wie bereits bei der Untersuchung zu variablen Kontrollpunkten in Kapitel 7.2.3 positioniert der Algorithmus eine der Kontrollpunktzeilen auf der Flächenkante. Die zweite verschiebt er bei dieser Optimierung auf eine Höhe von 96,7 mm und somit ebenfalls

nah an den Rand gegenüber des Loches. Der Optimierungsprozess dieser Kombination ist gemessen am Ergebnis und der Rechenzeit überdurchschnittlich gut. Jedoch muss einkalkuliert werden, dass für den Startentwurf sämtliche im Vorfeld durchgeführten Untersuchungen notwendig sind.

Für die dritte Optimierung (Kombination 3) wird ebenfalls ein Startentwurf gewählt, der aus den Betrachtungen der bisherigen Ergebnisse resultiert. In diesem Fall wird eine gezielt definierte Startfläche verwendet und die freien Kontrollpunkte im Lochbereich positioniert. Die z-Startkoordinaten werden hierbei in Anlehnung an die Anhebung des KP1 (vgl. Kapitel 7.2.1) gewählt und sind in Formel 7.2 in Matrixform dargestellt. Durch diese Startwerte liegt ein leichtes Gefälle der Fläche in Richtung des Kontrollpunkts 1 vor, welches durch den Kontrollpunkt selbst noch verstärkt wird. In x- und y-Richtung werden die Kontrollpunkte direkt auf den Lochrand und in 75 mm Abstand zum Koordinatenursprung definiert. Somit entspricht die Verteilung der manuellen Einteilung im vorangegangenen Unterkapitel. Nun ist durch die veränderten z-Koordinaten jedoch bereits eine Tendenz vorgegeben. Dieser Startentwurf erzeugt um das Loch gekrümmte Kurvenverläufe, woraus schon zu Optimierungsbeginn ein maximaler Tsai-Wu-Index von lediglich 0,0181 resultiert. In 209 Iterationen wird dieser noch auf  $\Phi_{max} = 0,014$  gesenkt. Der Startentwurf ist ein gutes Beispiel für die gewinnbringende Nutzung der bereits gewonnenen Ergebnisse. Wie schon bei der vorangegangenen Optimierung darf der Aufwand der Voruntersuchungen bei der Bewertung der Ergebnisse nicht vernachlässigt werden.

$$z\_Start\_frei = \begin{bmatrix} 4,6 & 4,8 & 4,9 & 5 \\ 4,8 & 4,8 & 4,9 & 5 \\ 4,9 & 4,9 & 4,9 & 5 \\ 5 & 5 & 5 & 5 \end{bmatrix} \begin{pmatrix} KP1 & KP2 & KP3 & KP4 \\ KP5 & KP6 & KP7 & KP8 \\ KP9 & KP10 & KP11 & KP12 \\ KP13 & KP14 & KP15 & KP16 \end{pmatrix} \quad (7.2)$$

In der letzten Kombination wird ebenfalls der Ansatz einer vordefinierten Startfläche verfolgt. Statt variablen Kontrollpunkten ist nun die Kontrollpunktanzahl auf 36 erhöht, wofür die Matrix der Startwerte entsprechend erweitert wird. Im Gegensatz zu der Optimierung mit ebener Startfläche und 6x6 Kontrollpunkten erkennt der Algorithmus durch die vorgegebene Form die korrekte Tendenz und konvergiert zum gewünschten Optimum. Hierbei wird ein  $\Phi_{max} = 0,0132$  erreicht. Die Optimierung dauert mit über 1000 Iterationen jedoch überdurchschnittlich lange.

## 8. Auswertung und Fazit

In diesem Kapitel sollen zunächst die gewonnenen Ergebnisse geprüft und bewertet werden. Bevor ein abschließendes Fazit gezogen wird, erfolgt eine Auflistung möglicher Verbesserungen und Weiterentwicklungen des entwickelten Optimierungsprogramms. Im letzten Kapitel erfolgt ein Ausblick, welche Inhalte in nachfolgenden Arbeiten betrachtenswert sind, um das Thema der optimalen Strukturauslegung mit krummlinigen Faserverläufen voranzutreiben.

### 8.1. Auswertung

In Kapitel 7 wurden diverse Berechnungen durchgeführt und deren Ergebnisse dargestellt. Um diese auszuwerten, werden die gewonnenen Daten in diesem Kapitel zunächst zusammengefasst und mit den in Kapitel 4.3 ermittelten Werten konventioneller Laminats verglichen. Anschließend wird anhand der Ergebnisse **Hondas** geprüft, ob die ermittelten Optima verifiziert werden können.

#### 8.1.1. Betrachtung der Optimierungsergebnisse

In Tabelle 8.1 sind die sechs besten Optimierungsergebnisse aufgelistet. Ihre Daten werden sowohl an den Vergleichswerten als auch gegenseitig gemessen. Zu dem absoluten Ergebnis wird auch eine prozentuale Verbesserung, bezogen auf das konventionelle  $[0^\circ \pm 45^\circ 90^\circ]_s$ -Laminat, angegeben.

Tabelle 8.1.: Top 6 Optimierungsergebnisse

	$\Phi_{max}$	Iter.	Verbes.[%]	Beschreibung
1	0,0111	623	87,6	Restringierte Optimierung $\kappa = 0,05$
2	0,0123	339	86,2	Restringierte Optimierung $\kappa = 0,03$
3	0,0125	203	86,0	Variable Kontrollpunkte in vertikaler Richtung und $\kappa = 0,05$
4	0,0129	779	85,6	Restringierte Optimierung $\kappa = 0,04$
5	0,013	544	85,5	Variable Kontrollpunkte in vertikaler Richtung
6	0,0132	1019	85,2	6x6 Kontrollpunkte und definierte Startfläche

Die Ergebnisse zeigen deutliche Verbesserungen durch den Einsatz krummliniger Faserverläufe. Die sechs besten Entwürfe legen die Fasern so, dass der maximal auftretende Tsai-Wu-Index um rund 87 Prozent vermindert wird. Dadurch betragen die absoluten Werte nur noch rund ein Siebtel des Vergleichswerts der konventionellen Auslegung. Bei der Bewertung der absoluten und prozentualen Verbesserungen muss allerdings bedacht werden, dass es sich um theoretische Werte handelt und diese in der Realität aufgrund der getroffenen Modellvereinfachungen und fertigungstechnischer Bedingungen wahrscheinlich nicht erreicht werden. Der aufgezeigte PARETO-Rand ist aufgrund der wenigen Optimierungen mit aktiven Restriktionen verhältnismäßig dünn besetzt. Dennoch lässt sich daraus bereits abschätzen, mit welchen Krümmungswerten welche Festigkeiten erreicht werden können.

Bei der Größe des untersuchten Problems ist die Rechendauer von geringer Bedeutung. Bei großen Untersuchungen kann diese jedoch ein entscheidender Faktor sein. Daher muss das gefundene Optimum auch im Verhältnis zur benötigten Iterationszahl betrachtet werden. Hier ist erkennbar, dass das beste Ergebnis hinsichtlich des Vergleichslaminats nur um 1,4 Prozent besser als die zweite Optimierung ist, sie dafür allerdings fast die doppelte Rechenzeit benötigt. Das dritte Ergebnis ist zwar bei vergleichbarem Ergebnis noch schneller, jedoch muss bei diesem einbezogen werden, dass diese Optimierung auf Voruntersuchungen basiert. Das sechste Ergebnis liefert zwar ein ähnliches Ergebnis, ist hinsichtlich der Rechendauer jedoch weit abgeschlagen, da über 1000 Iterationen nötig sind, die zudem mit erhöhter Feinheit berechnet werden. Die insgesamt stark variierenden und teilweise hohen Iterationszahlen sind auf den Ansatz der Bézier-Fläche und die damit verbundenen Ersatzvariablen zurückzuführen. Diese hindern den Algorithmus sichtlich

am schnellen Konvergieren, da einzelne Variablenänderungen Auswirkungen auf viele Winkel haben. Aufgrund der Spannungskonzentration im Lochbereich haben bereits geringe Winkeländerungen in diesem Gebiet große Auswirkungen auf den Tsai-Wu-Wert. Dies erschwert es dem Algorithmus zusätzlich, ein Optimum auszumachen. Die Krümmungswerte reagieren ebenfalls sensibel auf Veränderungen, was es dem Algorithmus bei restringierten Optimierungen schwer macht, gegen eine Restriktion zu konvergieren.

Auch wenn die Ergebnisse aus der einfachen Festigkeitsbetrachtung nicht verbessert werden können, zeigen alle in der Parameterstudie getesteten Einstellungsmöglichkeiten positive Effekte, wenn sie richtig eingesetzt und kombiniert werden. Besonders die variablen Kontrollpunkte führen schnell zu deutlichen Verbesserungen. Hierbei ist auffällig, dass der Algorithmus Kontrollpunkte auf den Bauteilrand legt und die Fläche bzw. die Verläufe somit effektiv nur noch mit 3 Kontrollpunkten je Richtung optimiert. Hierbei werden sehr gute Ergebnisse erreicht, woraus erkennbar ist, dass bereits mit wenigen Kontrollpunkten sehr variable Formen und Verläufe möglich sind. Eine Steigerung der Kontrollpunkte führt daher quantitativ nicht zu besseren Ergebnissen, bildet aber geradlinigere Verläufe im oberen Strukturbereich ab. Dieser Umstand resultiert aus dem feineren Kontrollpunktraaster, wodurch die unteren Punkte weniger Einfluss auf den oberen Bereich haben. Die Optimierungen mit mehr Kontrollpunkten benötigen jedoch einen Startentwurf, da ansonsten die Wahrscheinlichkeit groß ist, dass der Lösungsalgorithmus sich aufgrund der hohen Variablenanzahl und deren starken Wechselwirkungen in einem lokalen Minimum verfängt oder keine Lösung findet. Bei der Studie zur Kontrollpunktanzahl ist dies die Ursache dafür, dass die Kombinationen 4x6, 6x4 und 6x5 zu keinen Ergebnissen führen. Bei 16 Kontrollpunkten und großen Restriktionswerten schafft der Algorithmus es aus einer ebenen Startfläche gute bis sehr gute Ergebnisse zu erzeugen. Für eine Betrachtung mit starken Restriktionen ( $k_{restr.} < 0,018$ ) wird jedoch ebenfalls die Vorgabe eines Startentwurfs notwendig. Um zu prüfen, wie ein förderlicher Startentwurf aussieht, hilft die Sensitivitätsanalyse. An den durchgeführten Optimierungen ist erkennbar, dass die z-Startwerte einen wesentlichen Einfluss auf den Optimierungsverlauf haben, indem sie dem Algorithmus eine Tendenz vorgeben. Die Ergebnisse der kombinierten Parameteränderungen zeigen des Weiteren, dass Voruntersuchungen und die sinnvolle Kombination der gewonnenen Ergebnisse bei einer erfolgreichen Optimierung helfen.

### 8.1.2. Verifikation mit Referenzwerten

Wie zu Beginn der Arbeit beschrieben, besteht nicht die Möglichkeit einer realen Verifikation der Ergebnisse. Daher erfolgt an dieser Stelle der angestrebte Vergleich mit dem optimierten

Entwurf **Hondas**. Da die unterschiedlichen Ansätze zur Flächenerzeugung aller Wahrscheinlichkeit nach keine vollständig identischen Faserverläufe generieren können, werden leichte Abweichungen erwartet. Zusätzlich ist die FE-Vernetzung der **Hondas** nur angenähert und es liegen keine Informationen vor, mit welchen FE-Programmen **Honda** rechnet. Da diese sich in ihren Berechnungsweisen teilweise unterscheiden, können hier speziell bei den kleinen betrachteten Werten zusätzliche Abweichungen entstehen. Um mehrere Ergebnisse vergleichen zu können, werden neben dem optimalen Ergebnis zwei zusätzlichen Wertepaare einem Diagramm von **Honda** entnommen. Damit eine Vergleichbarkeit gegeben ist, müssen die Krümmungsindizes komparabel sein. Ob dies gegeben ist, kann aufgrund der unterschiedlichen Ansätze, die Flächen zu erzeugen, nicht mit Gewissheit gesagt werden. Die Werte müssen aber in der gleichen Größenordnung liegen.

Für den Vergleich sind in Tabelle 8.2 die drei erwähnten Optima mit den dazugehörigen Krümmungen aus **Honda** sowie die Ergebnisse entsprechender Vergleichsrechnungen aufgelistet. **Honda** erreicht in seiner Untersuchung ein  $\Phi_{max} = 0,0134$ . Dieses Ergebnis können die durchgeführten Optimierungen sowohl erreichen als auch leicht überbieten (siehe Tabelle 8.1). Für den direkten Vergleich wird die sechste Optimierung herangezogen, da diese mit einem  $\Phi_{max} = 0,0132$  nahezu das gleiche Ergebnis hat. Die dazugehörige Krümmung weicht leicht ab, liegt aber in der gleichen Größenordnung. Die Werte des zweiten Vergleichs sind nahezu identisch, während beim dritten die Krümmung der durchgeführten Optimierung wieder größer als das Resultat **Hondas** ist. Die Abweichungen liegen im Rahmen der besprochenen Toleranzen und die qualitativen Ergebnisse gelten somit als verifiziert. Aus ungeklärter Ursache weist **Honda** für ein Laminat ohne Krümmung (ausschließlich  $0^\circ$ -Faserlagen) ein  $\Phi_{max} = 0,038$  aus. Dieser Wert ist deutlich niedriger, als der in dieser Arbeit berechnete Wert für ein reines Laminat aus  $0^\circ$ -Lagen ( $\Phi_{max} = 0,475$ ).

Tabelle 8.2.: Vergleich mit Ergebnissen von **Honda**

<b>Honda</b>		<b>Untersuchung</b>	
$\Phi_{max}$	$\kappa[\frac{1}{mm}]$	$\Phi_{max}$	$\kappa[\frac{1}{mm}]$
<b>0,0134</b>	0,0145	<b>0,0132</b>	0,019
0,02	0,0075	0,0203	0,072
0,029	0,0025	0,0293	0,0055

Zusätzlich sollen die entstehenden Faserverläufe verglichen werden. Hierzu ist die Auslegung **Hondas** in Abbildung 8.1 aufgeführt. Sie ist unterteilt in die + Lage (rechts), die

- Lagen (Mitte) sowie die Überlagerung beider Schichten (rechts). Zum Vergleich werden die Winkel der Optimierung mit 6x6 Kontrollpunkten und angepasster Startfläche genutzt, da diese mit einem  $\Phi_{max} = 0,0132$  nahezu das gleiche Ergebnis erzeugt. Die Verläufe sind in Abbildung 8.2 dargestellt, wobei die Winkel direkt überlagert sind. Die + Lage ist in schwarz, die - Lage in magenta gezeichnet. Die optimierten Verläufe anderer durchgeführter Rechnungen mit vergleichbaren Ergebnissen sind relativ ähnlich, lediglich die Winkel im Bereich  $y < 15$  schwanken leicht.

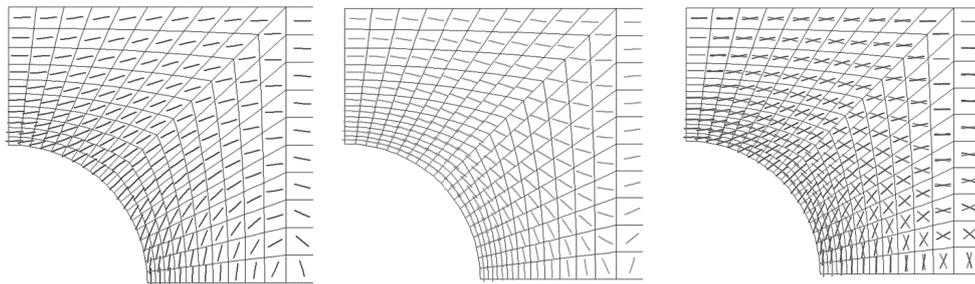


Abbildung 8.1.: Optimierte Winkel im Lochbereich bei 6x6 Kontrollpunkten

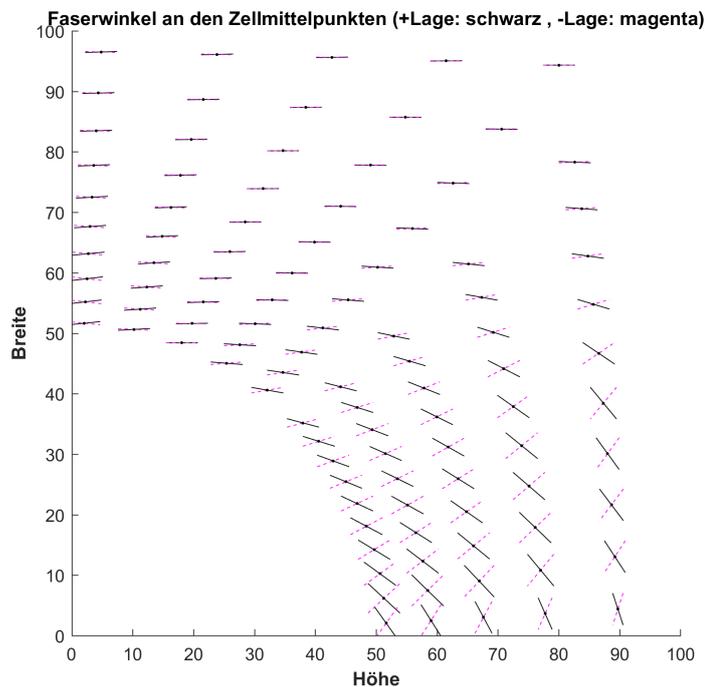


Abbildung 8.2.: Optimierte Winkel im Lochbereich bei 6x6 Kontrollpunkten

Auffällig bei dem Vergleich ist, dass die + und - Lagen vertauscht sind. Da eine reine Zugbelastung aufgebracht wird, hat die Lagenordnung keinen relevanten Einfluss auf das betrachtete Strukturverhalten, solange sie symmetrisch ist. Daher ist es vom Algorithmus abhängig, in welche Richtung er die + Lage orientiert. Die Faserverläufe selbst weisen sichtbare Unterschiede auf. Während die durchgeführten Optimierungen die Fasern hinter dem Loch langsam in ein Laminat mit gekreuzten Winkeln ( $\pm 30^\circ < \text{Winkel} < \pm 60^\circ$ ) überführt, folgen die Fasern bei **Honda** vollständig der Lochkontur und führen somit zu  $90^\circ$ -Winkeln. Im Bereich der Strukturmitte ähneln sich die Verläufe, im oberen Bereich hingegen entstehen wieder Unterschiede. Bei **Honda** entstehen wieder kleine Winkel, während bei der eigenen Optimierung die Fasern horizontal verlaufen. Das trotz dieser teils signifikanten Unterschiede, das quantitative Ergebnis identisch ist, liegt vermutlich an der ausschließlichen Betrachtung des Maximalwerts. Dieser liegt im Falle der genannten Rechnung in Zelle 94 und somit am Lochrand im mittleren Strukturbereich. In diesem relevanten Gebiet liegt eine Übereinstimmung der Winkel mit denen **Hondas** vor.

## 8.2. Verbesserungsmöglichkeiten

Das Optimierungstool erzeugt zwar bereits aussagekräftige und verifizierbare Ergebnisse, dennoch gibt es einige Bereiche, in denen aus verschiedenen Gründen Modifikationen und Erweiterungen das Programm weiter verbessern würden. Beispielsweise sind im Laufe der Arbeit für manche Funktionen Vereinfachungen gemacht worden, die nun, nachdem der Funktionsnachweis des Tools erbracht wurde, wieder verallgemeinert werden müssen. Des Weiteren war es im Rahmen der Arbeit nicht möglich, alle entstandenen Ideen umzusetzen und einzuarbeiten. Auf die wichtigsten dieser Punkte wird nachfolgend eingegangen.

### Verwendung mehrerer Bézier-Flächen

Anstatt die Anzahl der Kontrollpunkte einer Bézier-Fläche zu erhöhen, besteht die alternative Möglichkeit, einen Verbund aus mehreren Bézier-Flächen zu bilden. Auf diese Weise erhöht sich ebenfalls die Flexibilität der Form, jedoch mit dem Vorteil, dass nicht mehr jeder Kontrollpunkt Auswirkungen auf die gesamte Fläche hat. So werden Verlaufsänderungen der Fasern in lokalen Bereichen ermöglicht, ohne die Verläufe im gesamten Strukturbereich zu beeinflussen. Beispielsweise könnten mit diesem Ansatz sowohl die Fasern im hinteren als auch im Lochbereich gemeinsam optimiert werden. Die Anzahl der zusätzlichen Optimierungsvariablen richtet sich hierbei nach der Anzahl der verwendeten Bézier-Flächen und ihrer gewählten Kontrollpunktanzahl. An einer Seite teilen sich die Flächen jeweils eine Kontrollpunktreihe. Um einen korrekten Übergang zu gewährleisten, müssen Kontinuitätsbedingungen zwischen den einzelnen Bézier-Flächen definiert werden, was zusätzlichen Programmieraufwand erfordert.

### Wahl des Optimierungsalgorithmus

Um die Wahrscheinlichkeit, dass das globale Optimum gefunden wird, zu erhöhen, kann statt des SQP- ein generischer Algorithmus eingesetzt werden. Da diese Algorithmen in der Regel jedoch mehr Rechenzeit benötigen, um ein Optimum auszuarbeiten, können beide Algorithmen kombiniert werden. Zunächst wird mit einem Evolutionsalgorithmus das globale Optimum gesucht und dieses anschließend mit dem SQP-Algorithmus präzisiert. Da in dieser Arbeit ausschließlich mit dem SQP-Algorithmus gearbeitet wird, ist auch das Optimierungsverhalten der weiteren Algorithmen, die die *fmincon*-Funktion seitens Matlab anbietet, zu testen. Zusätzlich könnte eine umfangreichere Sensitivitätsanalyse hinsichtlich der Optimierungsdauer weitere Vorteile bringen.

### **Wahl des Index der Prozessierbarkeit**

Der gewählte Index für die Prozessierbarkeit ist mit der Wahl der Krümmung der Höhenschnittlinien, die zudem noch gemittelt werden, physikalisch wenig aussagekräftig. Er beruht nicht auf realen Faserverläufen und durch die Mittelwertbildung kann es passieren, dass unerkannt in der Struktur lokal deutlich größere Krümmungswerte entstehen. Daher sollte dieser von **Honda** übernommene Ansatz in weiteren Untersuchungen nicht verwendet werden. Ist kein sinnvoller Alternativwert ermittelbar, ist zumindest eine Zusammenfassung der Krümmungswerte mit der Kreisselmeier-Steinhauser-Funktion statt einer Mittelwertbildung zu empfehlen. Auf diese Weise werden die maximal vorliegenden Krümmungen berücksichtigt.

### **Berechnung des Rückgabeparameters**

In Anlehnung an **Honda** werden während der Optimierung nur die Tsai-Wu-Indizes im Lochbereich betrachtet. Dies ist nicht im Sinne der Gesamtoptimierung, da die Winkel im hinteren Bereich somit keinen Einfluss auf die Strukturantwort haben und daher nicht optimiert werden. Um die Ergebnisse verifizieren zu können, wurde dieser Ansatz übernommen, jedoch ist für weitere Betrachtungen die Verwendung sämtlicher Tsai-Wu-Indizes zu empfehlen.

### **Optimierung aller Laminatlagen**

Zur Demonstration der Funktionsweise wurde sich darauf beschränkt, die Fasern nur in einer Lage zu optimieren und daraus ein symmetrisches Laminat zu erzeugen. Ein Ziel der Arbeit ist jedoch, die Fasern in allen Lagen optimieren zu können. Dies kann ermöglicht werden, indem für jede Lage ein getrennter Aufruf des Bézier-Tools und kleine Modifikationen beim Schreiben der PCOMP-Karten durchgeführt werden. Allerdings erhöht sich somit die Anzahl an Entwurfsvariablen pro unabhängiger Lage um mindestens 16. Der Rechenaufwand insgesamt erhöht sich noch zusätzlich, da die Lagen sich gegenseitig beeinflussen.

### **Einbeziehung der Fertigungsfaktoren**

Wie in Kapitel 4.1.1 beschrieben, sind die erzeugten Faserverläufe nur schematische Abbildungen und in der dargestellten Form nicht herstellbar. Da keine Kontinuität besteht, müssen Fasern mitten in der Struktur starten oder enden. Auf diese Weise entstehen bei der Fertigung Lücken und Überlappungen, welche einen relevanten Einfluss auf die mechanischen Eigenschaften des Laminats haben. Eine Option zur Begrenzung dieses Problems ist ein weiteres Programm, welches nach der Optimierung aus den einzelnen Winkeln möglichst kontinuierliche und herstellbare Faserverläufe erzeugt. Da auf diese Weise nachträglich der optimierte

Entwurf verändert wird, ist es vorteilhafter, die Effekte der Fertigung auf das mechanische Verhalten der Struktur zu untersuchen und diese mathematisch direkt in die Optimierung zu integrieren.

### **Anpassung der FE-Vernetzung**

Wie in Kapitel 4.4 ermittelt, ist die FE-Vernetzung der Geometrie mit 400 Zellen nur bedingt ausreichend. Nachdem die Ergebnisse **Hondas** mit dieser Vernetzung verifiziert sind, sollte für reale Anwendungen in weiteren Untersuchungen eine FE-Vernetzung gewählt werden, die netzunabhängige Ergebnisse erzeugt.

### **Erweiterungen des Optimierungstools**

Da die grundsätzliche Funktionalität des Optimierungstools sowie der Nachweis über korrekte Ergebnisse im Vordergrund der Arbeit steht, sind die entwickelten Funktionen und Module auf die Nachbildung der Untersuchung **Hondas** ausgelegt. Um das Tool universell einsetzbar zu machen und komplexere Strukturen betrachten zu können, bedarf es daher einiger Modifikationen. Beispielsweise ist eine Erweiterung zur Betrachtung anderer Zielfunktionen bzw. Strukturantworten wichtig. Hinsichtlich des FE-Modells sind des Weiteren die Verwendbarkeit anderer Elementarten sowie die Erzeugung variablerer Laminatstrukturen von Interesse. Hier sind aktuell lediglich TRIA3 und CQUAD4-Elemente sowie Lamine mit 4 oder 8 Lagen möglich.

### **Erweiterungen der GUI**

Die Entwicklung der grafischen Oberfläche ist kein geforderter Bestandteil der Arbeit und dient hauptsächlich der Vereinfachung der Handhabung des Optimierungstools. Dementsprechend wurde sie verhältnismäßig rudimentär programmiert und es bestehen einige Verbesserungsmöglichkeiten. Wesentlich ist hier die Integration von Sicherheitsfunktionen zur Verhinderung falscher Eingaben sowie die Programmierung aussagekräftiger Fehlermeldungen. Weitere sinnvolle Erweiterungen sind die Eingabemöglichkeit der Startwerte der z-Variablen sowie eine Auswahlmöglichkeit der zu untersuchenden Zielfunktion. Entsprechend dieser müssen dann ebenfalls die Ausgabefelder angepasst werden. Des Weiteren kann die grafische Darstellung der optimierten Winkel direkt in die Oberfläche eingebunden werden.

### 8.3. Fazit

Anhand der eingangs der Arbeit aufgestellten Ziele wird abschließend resümiert, ob und in wie weit diese erreicht worden sind.

Das zu erschaffene Optimierungsprogramm auf Basis der UFO-Umgebung liegt vor und ermöglicht die automatische Optimierung von Faserverbundstrukturen mit krummlinigen Faserverläufen. Hierbei ist die angestrebte Modularität gegeben, wodurch das Tool mit geringem Aufwand an andere Aufgabenstellungen angepasst werden kann. Die Güte der Optimierungsergebnisse und die notwendige Rechendauer sind stark abhängig von den vorgenommenen Einstellungen. Dieser Umstand ist auf den indirekten Ansatz der Variablenbestimmung und den verwendeten, gradientenbasierten Optimierungsalgorithmus zurückzuführen. Die zusätzlich programmierte grafische Benutzeroberfläche vereinfacht die Ein- und Ausgabe der relevanten Optimierungsparameter und ermöglicht teilweise automatisierte Optimierungsabläufe.

Das zweite Ziel, mit dem entwickelten Optimierungsprogramm eine Struktur hinsichtlich der Festigkeit zu optimieren, ist ebenfalls erfüllt. Die gewonnenen Ergebnisse konnten mit den Referenzwerten **Hondas** verifiziert werden und weisen große Potentiale zur Verbesserung mechanischer Eigenschaften auf. Die als Festigkeits-Kriterium betrachteten Tsai-Wu-Indizes konnten bei unrestrictierten Verläufen um bis zu 87 Prozent gegenüber einer konventionellen Auslegung gesenkt werden. Somit ist nachgewiesen, dass Bézier-Flächen zur Bestimmung und Optimierung von krummlinigen Faserverläufen erfolgreich einsetzbar sind. Die Intention, einen PARETO-optimalen Rand mit bestmöglichen Kompromissen aus Prozessierbarkeit und Festigkeit zu erzeugen, wurde bedingt erreicht. Hier bedarf es weiterer Rechnungen, um den PARETO-Rand zu präzisieren.

Die Einflüsse der unterschiedlichen Einstellungsparameter des Optimierungstools sind umfangreich in der durchgeführten Parameterstudie getestet worden. Aus dieser Studie konnten viele Erkenntnisse hinsichtlich der Kontrollpunktanzahl, deren Position und der Startkonfiguration der Entwurfsvariablen gewonnen werden. Zusätzlich wurde das Verständnis der Relevanz gezielter Überlegungen und Voruntersuchungen zur Steigerung der Effizienz einer Optimierung geschult. Demgegenüber wurden auch Erfahrungen zu möglichen Problembereichen und Hindernissen in einem Optimierungsprozess gemacht.

Abschließend bleibt zu sagen, dass alle gesetzten Ziele erreicht wurden. Lediglich kleinere Intentionen und Ideen konnten nicht umgesetzt werden. Diese sind in Kapitel 8.2 als Verbesserungsmöglichkeiten aufgeführt, damit sie gegebenenfalls in folgenden Arbeiten implementiert werden können.

### 8.4. Ausblick

Da die gewonnenen Ergebnisse große Verbesserungspotentiale aufzeigen, sollten weitere Arbeiten zum Thema *Optimale Auslegung von Luftfahrtstrukturen aus Faserverbundwerkstoffen mit krummlinig verlaufenden Faserorientierungen* durchgeführt werden. Dabei sollte speziell das übergeordnete Ziel, die Methode für die Auslegung von Luftfahrtstrukturbauteilen einzusetzen, im Fokus stehen. Das in dieser Arbeit geschaffene Optimierungstool sowie die gewonnenen Erfahrungen ermöglichen nachfolgenden Untersuchungen den direkten Einstieg in die Bearbeitung komplexerer Aufgabenstellungen.

Zudem ist es sinnvoll, hinsichtlich der in Kapitel 8.2 aufgeführten Verbesserungsmöglichkeiten weitere Betrachtungen anzustellen. Speziell in Bezug auf die Fertigung und der Einbindung ihrer Eigenschaften sind noch Probleme zu betrachten und zu lösen. Hinzu kommen konstruktive Verbesserungen des Tools selbst, beispielsweise die Einbindung generischer Algorithmen, der Einsatz mehrerer Bézier-Flächen zur Abbildung unabhängiger Faserverläufe oder die Erweiterung der grafischen Oberfläche, um das Programm benutzerfreundlicher zu gestalten.

# Literaturverzeichnis

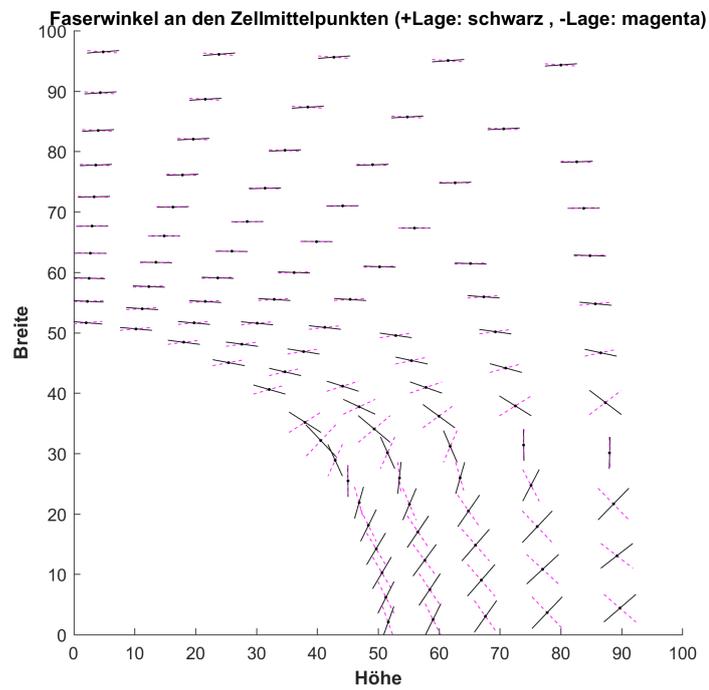
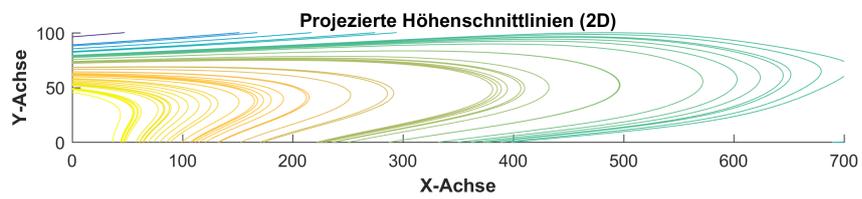
- AVK 2014** AVK-INDUSTRIEVEREINIGUNG VERSTÄRKTE KUNSTSTOFFE E.V.: *Handbuch Faserverbundkunststoffe/Composites*. Springer Vieweg, 2014
- Blom** BLOM, ADRIANA W.; ABDALLA, MOSTAFA M.; GÜRDAL, ZAFER: *Optimization of tow-placed, tailored composite laminates*. Delft University of Technology, 2007
- Brampton** BRAMPTON, CHRISTOPHER J.; KIM, H. ALICIA: *Optimization of Tow Steered Fiber Orientation Using the Level Set Method*. University of Bath, 2013
- CAE 2015** CAE SIMULATION SOLUTIONS: *Wie und mit welchen Parametern kann Nastran gestartet werden?*. <http://www.cae-sim-sol.com/de/content/wie-und-mit-welchen-parametern-kann-nastran-gestartet-werden> (abgerufen am 24.05.2016)
- Charette** CHARETTE, R.F.; HYE, M.W.: *Innovative design of composite structures: The use of curvilinear fiber format in structural design of composites*. University of Maryland, 1990
- Flemming 2003** FLEMMING, MANFRED ; ROTH, SIEGFRIED: *Faserverbundbauweisen: Eigenschaften*. Springer Verlag, 2003
- Geisberger 2013** GEISBERGER, MORITZ: *Optimale Auslegung von Faserverbundstrukturen unter Einsatz der FEM-Strukturanalyse*. HAW Hamburg, 2013
- Güldü** GÜLDÜ, SEDAT: *Analysis and Optimization of cylindrical Structures manufactured by automated Fiber Placement Technique*. Middle East Technical University, 2014

- Harzheim 2014** HARZHEIM, LOTHAR: *Strukturoptimierung: Grundlagen und Anwendungen*. Verlag Europa-Lehrmittel, 2014
- Holldack 2011** HOLLDAK, MARIO; IGARASHI, TERUKI; NARITA, YOSHIHIRO: *Bezierkurven*. Goethe-Universität Frankfurt am Main, 2011
- Honda** HONDA, SHINYA; IGARASHI, TERUKI; NARITA, YOSHIHIRO: *Multi-Objective Optimization of Curvilinear Fiber Shapes for Laminated Composite Plates by Using NSGA-II*. Hokkaido University, 2013
- Klein 2015** KLEIN, BERND: *FEM: Grundlagen und Anwendungen der Finite-Elemente-Methode im Maschinen- und Fahrzeugbau*. Springer Vieweg, 2015
- Lemaire** LEMAIRE, ETIENNE; ZEIN, SEMIH; BRUYNEEL, MICHAEL: *Optimization of composite structures with curved fiber trajectories*. SAMTECH s.a.; University of Liège, 2015
- MathWorks 2016** THE MATHWORKS, INC.: *MATLAB Documentation R2016a*. <http://de.mathworks.com/help/> (abgerufen am 24.05.2016)
- Papula 2008** PAPULA, LOTHAR: *Mathematik für Ingenieure und Naturwissenschaftler: Band 1*. Vieweg+Teubner Verlag, 2008
- Parnas** PARNAS, LEVEND; ORAL, SÜHA; CEYHAN, ÜMIT: *Optimum design of composite structures with curved fiber courses*. Middle East Technical University, 2003
- Prabhakar** PRABHAKAR, SUHAS: *A Methodology for Finite Element Analysis of Curvilinear Fiber Laminates with Defects, Fabricated by Automated Fiber Placement Technique*. McGill University, 2011
- QRG 2015** MSC SOFTWARE COOPERATION: *Quick Reference Guide: MSC Nastran 2014.1.*, 2015
- Rieg 2014** RIEG, FRANK; HACKENSCHMIDT, REINHARD; ALBER-LAUKANT, BETTINA: *Finite Elemente Analyse für Ingenieure*. Carl Hanser Verlag, 2014
- Schumacher 2013** SCHUMACHER, AXEL: *Optimierung mechanischer Strukturen*. Springer Vieweg, 2013

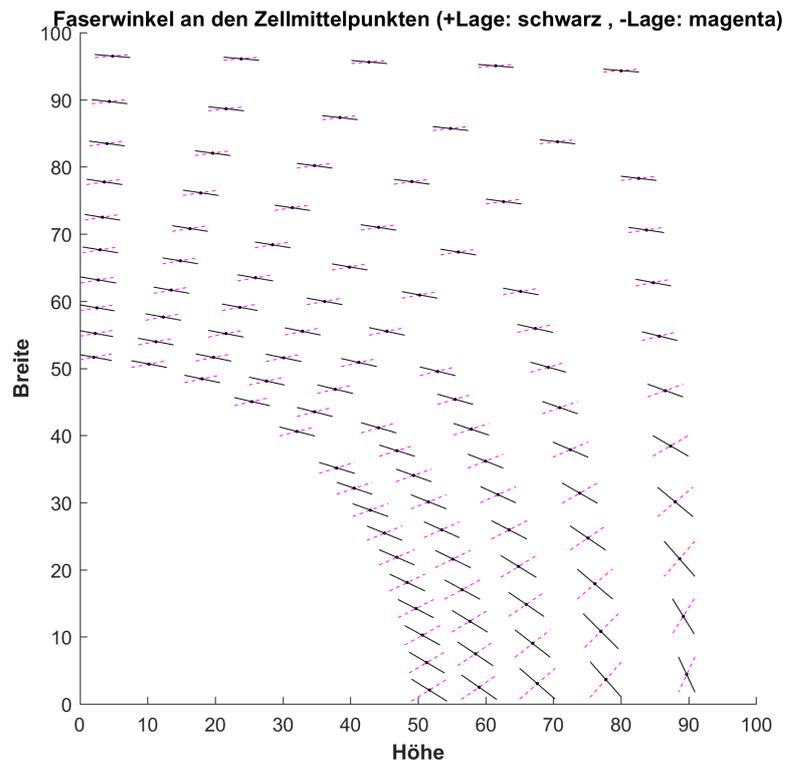
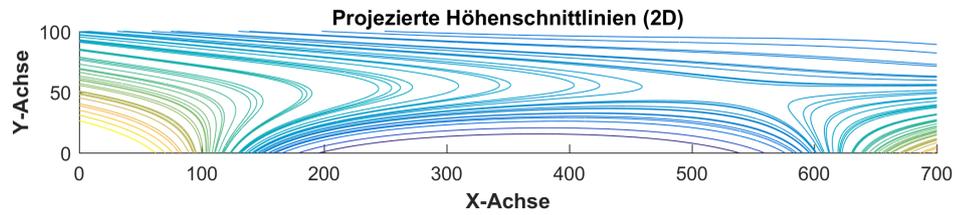
- Setoodeh** SETOODEH, SHAHRIAR; GÜRDAL, ZAFER: *Design of composite layers with Curvilinear Fiber Paths using Cellular Automata*. Virginia Polytechnic Institute and State University, 2003
- Ungwattanapanit** UNGWATTANAPANIT, TANUT; BAIER, HORST: *Postbuckling Analysis and Optimization of stiffened Fuselage Panels utilizing Variable-Stiffness Laminates*. Technische Universität München, 2014
- Wrenn 1989** WRENN, GREGORY A.: *An indirect Method for Numerical Optimization Using the Kreisselmeier-Steinhauser Function*. NASA Contractor Report, 1989
- Wu** WU, ZHANGMING; WEAVER, PAUL M.; RAJU, GANGADHARAN: *Postbuckling Optimization of Variable Angle Tow Composite Plates*. Advanced Composite Centre for Innovation and Science UK, 2013
- Zhang 2006** ZHANG, HANGXIN; FENG, JIEQING: *Bézier Curves and Surfaces(2)*. Präsentation, 2006

# A. Anhang Darstellungen zu 7.2.1

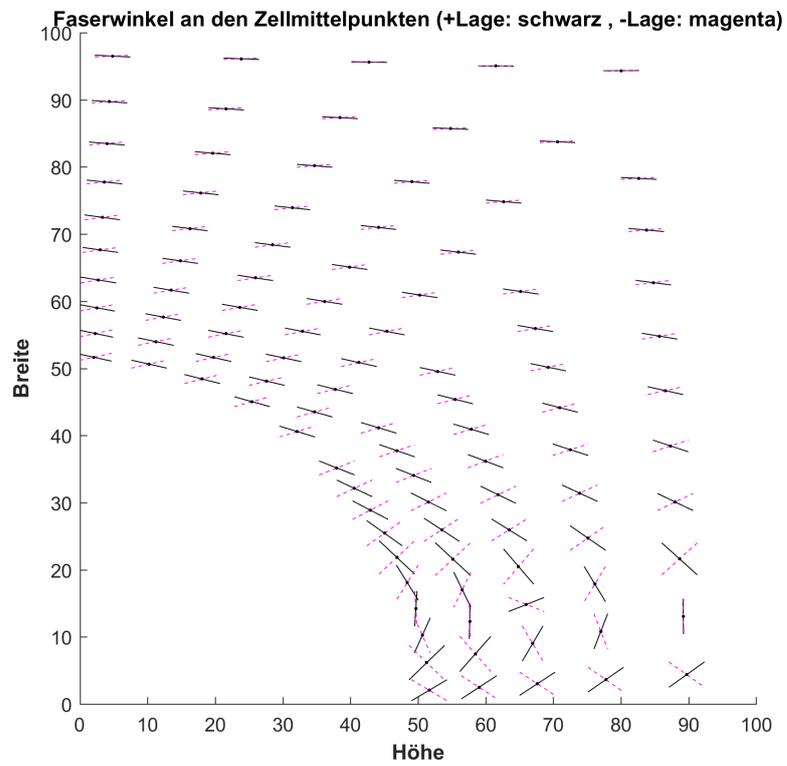
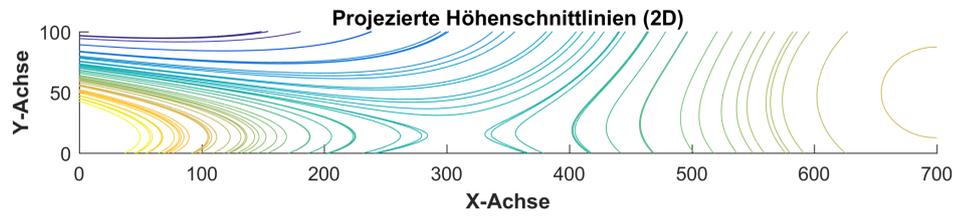
## Ansatz 1



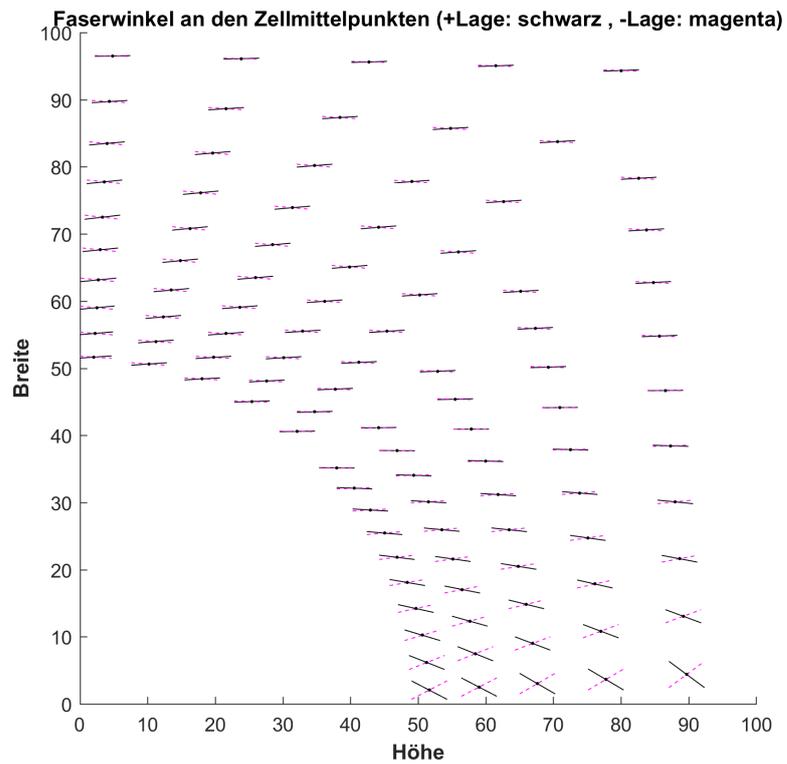
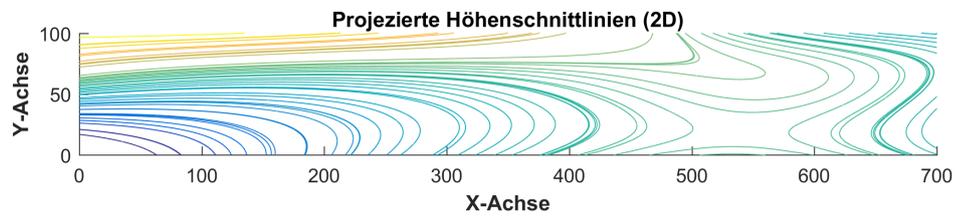
## Ansatz 2



**Ansatz 3**

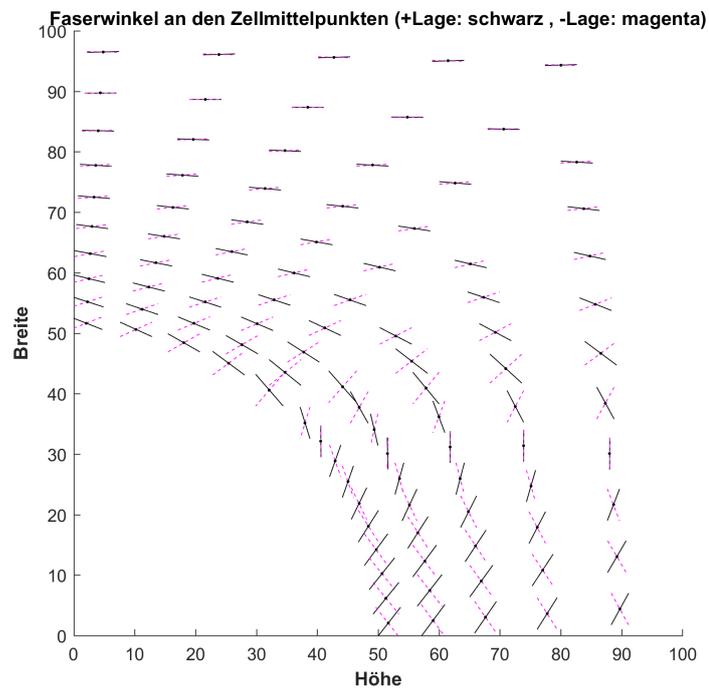
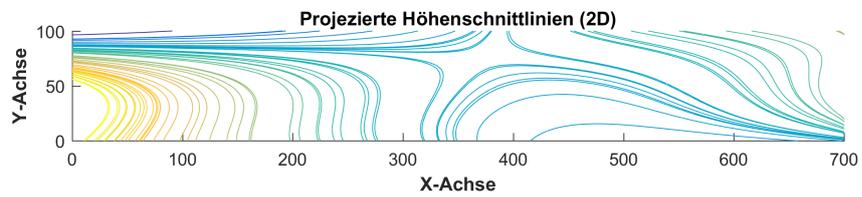


### Ansatz 4

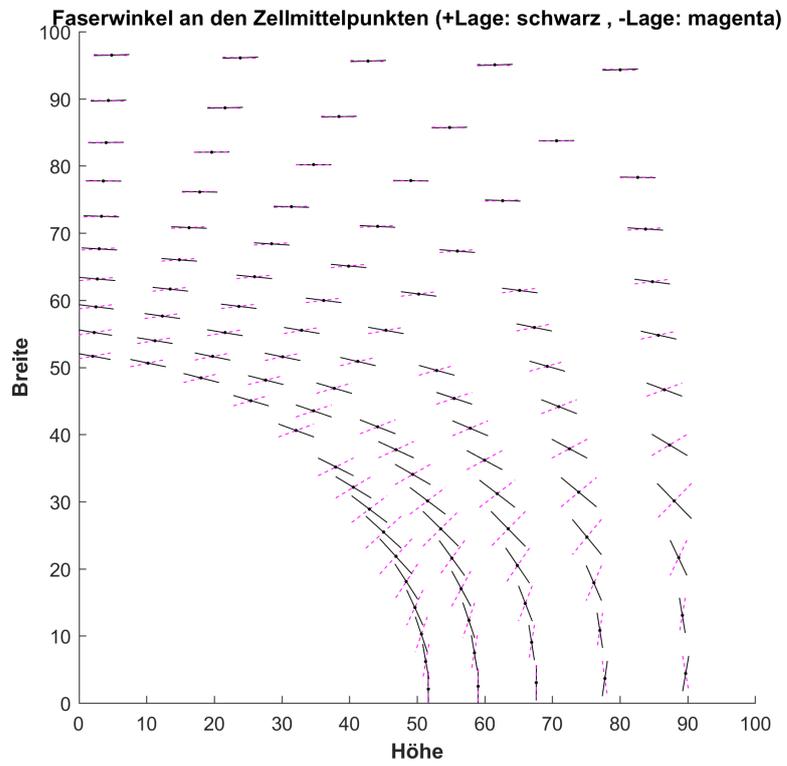
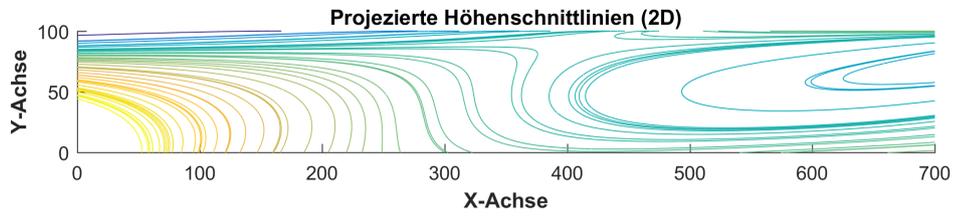


## B. Anhang Darstellungen zu 7.2.3

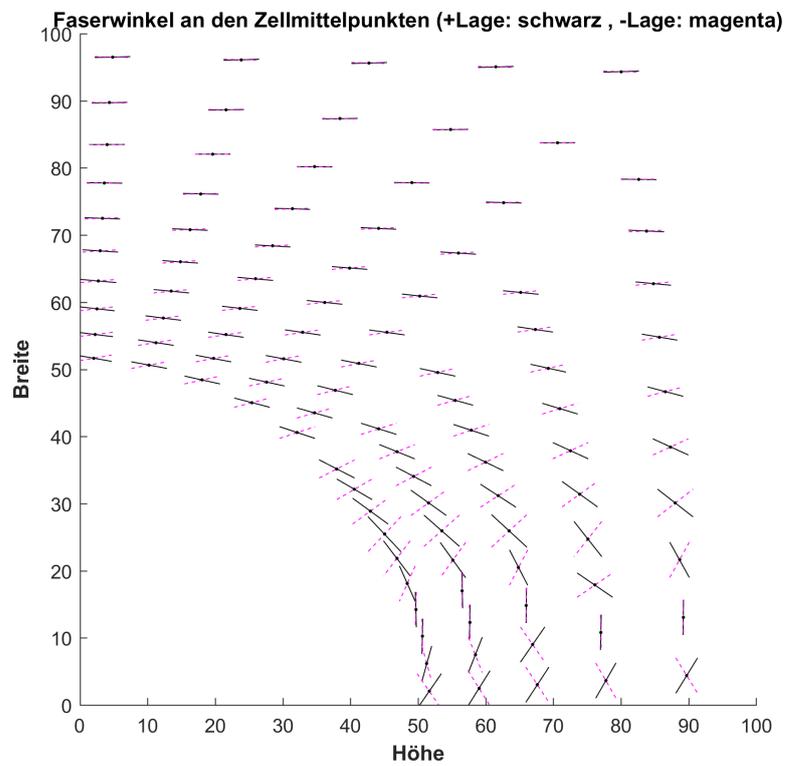
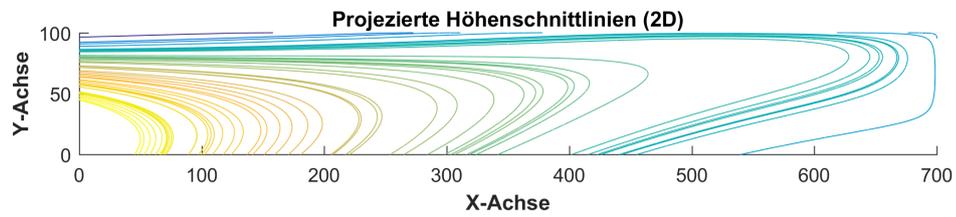
KP in x-Richtung variabel



### KP in y-Richtung variabel

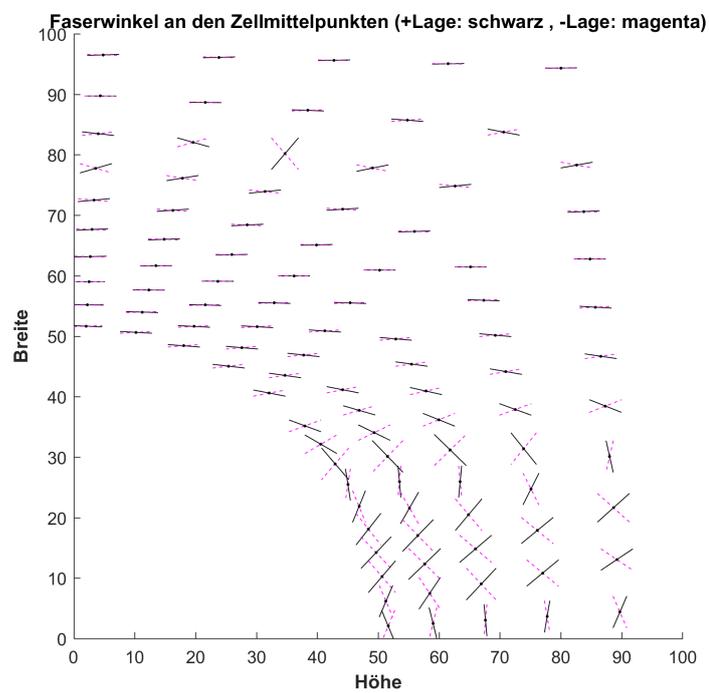
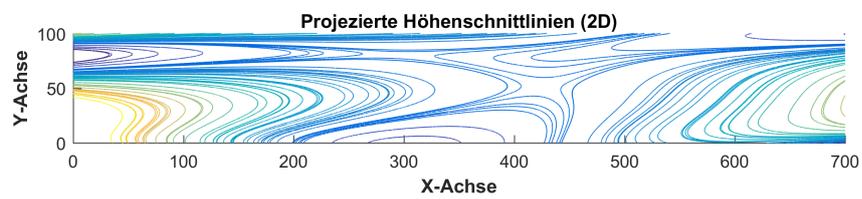


### KP in x- und y-Richtung variabel

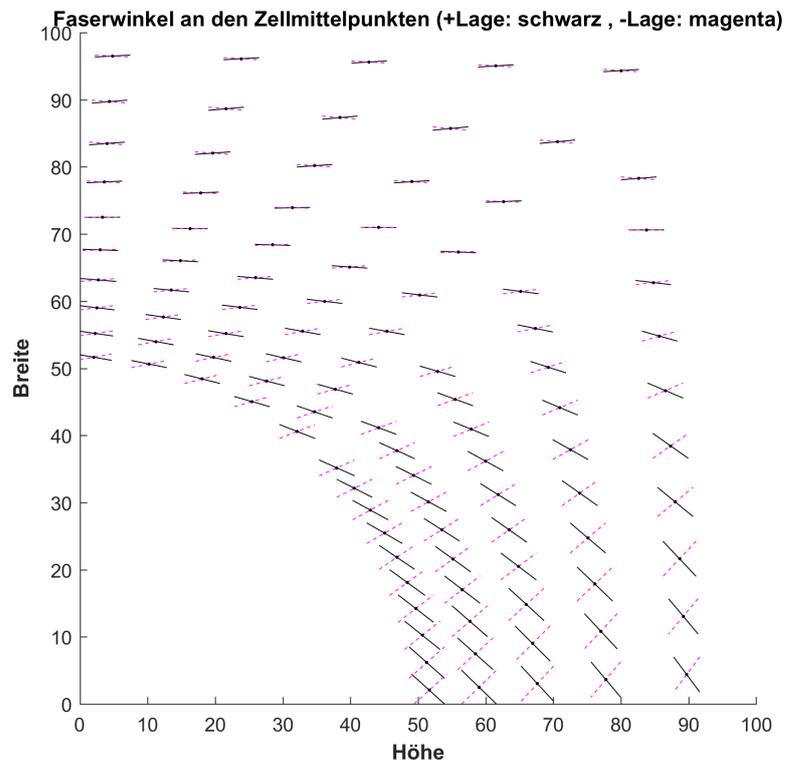
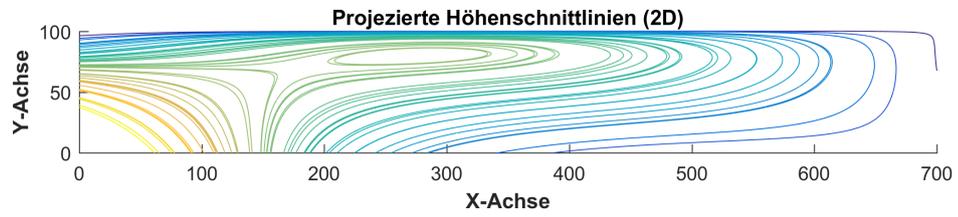


## C. Anhang Darstellungen zu 7.2.4

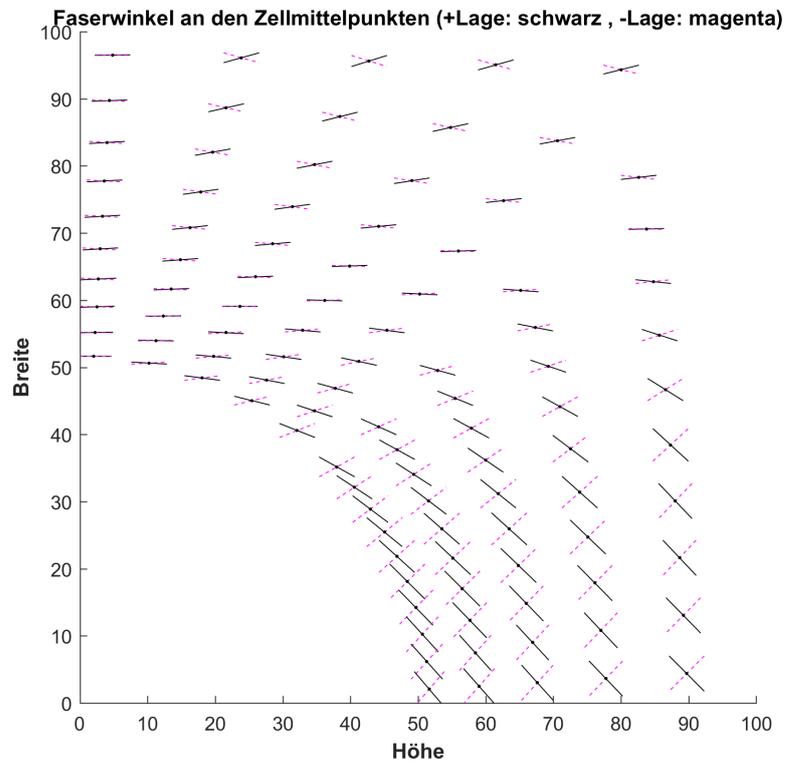
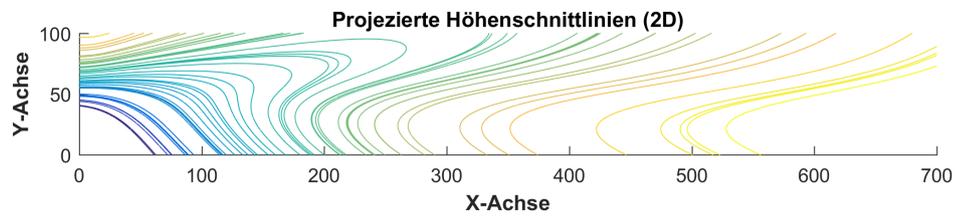
### Kombination 1



### Kombination 2



### Kombination 3



### Kombination 4

