

# **Analyse und Erzeugung von glatten Flächenübergängen für das CNC-Fräsen**

zur Erlangung des akademischen Grades  
„Doktor der Naturwissenschaften“  
(Dr. rer. nat.)  
an der  
Fakultät für Informatik und Mathematik  
der  
Universität Passau

eingereicht von

**Florian Lorenz**

geboren am 18.07.1988

in Regensburg

Gutachter: Prof. Dr. Tomas Sauer  
Prof. Dr. Hartmut Prautzsch

Tag des Rigorosums: 16.11.2017

Dezember 2017



# INHALTSVERZEICHNIS

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Prozesskette einer Fertigung mit numerisch gesteuerten Werkzeugmaschinen . . . . .	1
1.3	Problemstellung und Lösungsansatz . . . . .	4
1.4	Gliederung der Arbeit . . . . .	7
<b>2</b>	<b>Grundlagen</b>	<b>9</b>
2.1	Differentialgeometrie . . . . .	9
2.2	B-Splines . . . . .	24
2.3	Schnittstelle NXOpen . . . . .	42
<b>3</b>	<b><math>G</math>-Stetigkeiten</b>	<b>47</b>
3.1	$C^n/G^n$ -stetige Kurvenübergänge . . . . .	47
3.2	$C^n/G^n$ -stetige Flächenübergänge . . . . .	57
3.3	$G^{0,1,2}$ -Stetigkeitstest . . . . .	64
<b>4</b>	<b><math>G^2</math>-Patcher</b>	<b>83</b>
4.1	Problemstellung . . . . .	83
4.2	Algorithmus - Überblick und Lösungsstrategie . . . . .	85
4.3	Minimierung mithilfe der $G^2$ -Freiheitsgrade . . . . .	106
<b>5</b>	<b>Erweiterungs-Flächen</b>	<b>117</b>
5.1	Problemstellung . . . . .	117
5.2	Trim and Extend in NX . . . . .	119
5.3	CAM Verfahrensmethoden . . . . .	121
5.4	Lokale $C^2$ -stetige Flächenerweiterung in einer Parameterrichtung . . . . .	122
5.5	Maschinentest an einer 5-Achs-Fräsmaschine . . . . .	132
<b>6</b>	<b>Schlussbemerkung</b>	<b>137</b>
	<b>Literaturverzeichnis</b>	<b>142</b>
	<b>Symbolverzeichnis</b>	<b>143</b>
	<b>Liste aller Algorithmen</b>	<b>144</b>
	<b>Index</b>	<b>145</b>
	<b>Erklärung der Selbständigkeit</b>	<b>149</b>



## 1.1 Motivation

Das Produktportfolio und der Einsatzbereich der industriellen Fertigung mit numerisch gesteuerten Werkzeugmaschinen reichen von Spritzgussformen für Lebensmittel bis hin zu künstlichen Kniegelenken und sind in der heutigen Zeit aus dem Alltag nicht mehr wegzudenken. Jedem dieser Bauteile liegt ein geometrisches Modell zugrunde, das mittlerweile computergestützt erzeugt und digital gespeichert wird. Aufgrund der enormen Entwicklung von Rechnerleistung und Speicherkapazitäten innerhalb der letzten Jahre, erschließen sich für einen Designer immer neue Möglichkeiten noch komplexere Modelle zu konstruieren. Dies führt dazu, dass die Detailtreue immer tiefer und die Bauteile immer filigraner werden und die Anzahl an verwendeten Freiformflächen stark zunimmt. Die Aufgabe, aus einem solchen Modell mithilfe einer numerischen Steuerung und einer Werkzeugmaschine Bauteile in höchster Qualität zu fertigen, wird damit immer anspruchsvoller. Bevor das Werkstück an einer Werkzeugmaschine gefertigt werden kann, muss eine Bearbeitungsstrategie ermittelt werden. Während dieser Phase, die zwischen der Konstruktion des Modells und der Werkzeugmaschine einzuordnen ist, kommt es in der Praxis an Flächenübergängen des Konstruktions-Modells in einigen Fällen zu unerwünschten Effekten, die dazu führen, dass bei der Bearbeitung durch eine Werkzeugmaschine markante Stellen auftreten (vgl. Abbildung 1.3).

In vielen Fällen lässt sich nachweisen, dass diese Problematik durch den Verlust von Informationen entlang der Datenverarbeitungskette und unbemerkt durch den Konstrukteur (vgl. Abbildung 1.1) entsteht. Es ist davon auszugehen, dass in Zukunft die numerische Steuerung Zugriff auf Geometrieinformationen des ursprünglichen Modells besitzt. Anstatt allerdings sämtliche Informationen als Datenflut der numerischen Steuerung zu übergeben, liefern die Ergebnisse dieser Arbeit eine Möglichkeit die Werkstückqualität, durch intelligentes Anreichern und Verarbeiten der verlorenen Informationen aus dem Modell, zu verbessern.

## 1.2 Prozesskette einer Fertigung mit numerisch gesteuerten Werkzeugmaschinen

Der Begriff *Werkzeugmaschine* umfasst ein großes Spektrum an verschiedenen Maschinen, die mithilfe eines Werkzeugs Veränderungen an einem Werkstück vornehmen. Ein Überblick über die verschiedenen Maschinenarten sowie deren Anwendungsbereich ist in [13] nachzulesen. Obwohl die folgende Prozesskette allgemeingültig für viele Maschinentypen ist, wird in dieser Arbeit speziell ein sehr häufig verwendeter und weit verbreiteter Typ, nämlich die Fräsmaschine, betrachtet. Die Funktionsweise ist unabhängig von der jeweiligen Achs-Konfiguration der Maschine. In der Regel besitzen Fräsmaschinen bis zu drei Linearachsen und wahlweise zusätzliche Rundachsen. Durch entsprechende Achsbewegungen fährt das Werkzeug (der *Fräser*) einen vordefinierten Weg ab und trägt dabei Material eines Bauteils ab (*spanende Bearbeitung*).

Wie in Abbildung 1.1 dargestellt, steht zu Beginn der Fertigung mit numerisch gesteuerten Werkzeugmaschinen ein Modell der Bauteilgeometrie, das mit einem *CAD-System (Computer Aided Design)* konstruiert wird und aus einer Menge von Körpern und Flächen, die in analytischer Form vorliegen, besteht. Im nächsten Schritt wird die Geometrie des CAD-Systems mit einer Vielzahl von gewählten Parametern diskretisiert und eine sogenannte Werkzeugbahn erzeugt. Diese Aufgabe übernimmt das sogenannte *CAM-System (Computer Aided Manufacturing)*, das aufgrund seiner Komplexität und seines Spektrums an Einstellungsmöglichkeiten ein Thema für sich darstellt. Die Werkzeugbahn wird in einem allgemeinen, maschinenunabhängigen Format dargestellt. Allerdings sind gewisse Grundkenntnisse der Fertigungsmethode im CAM-System notwendig, da die Wahl verschiedener Parameter (wie z. B. Verfahrmethode, Aufspannsituation, Werkzeugausprägung, Toleranzen, etc.) Einfluss auf die Bearbeitung haben. Der *Postprozessor (PP)* übersetzt dann die Werkzeugbahn in steuerungsabhängigen Code, das sogenannte *NC-Programm*. Dieser Code besitzt üblicherweise das in der Norm DIN66025 festgelegte Format (*G-Code*, siehe [31, Kapitel 6]) und besteht in der Regel aus einer Folge von Positionsvektoren  $(x_i, y_i, z_i)^T$  und ggf. Orientierungen, die auf unterschiedliche Art und Weise dargestellt werden können (Vektoren, Winkel, etc.). Zu den Aufgaben der *numerischen Steuerung (CNC - Computerized Numerical Control)* gehört es, diese Folge von Positionen/Orientierungen für die verwendete Kinematik zu übersetzen (Transformationen, etc.) und daraus eine möglichst glatte, nach der Zeit parametrisierte Bahnplanung durchzuführen, um die entsprechenden Sollwerte für die Antriebe bestimmen zu können. Das Aufgabenspektrum und die interne Umsetzung der numerischen Steuerung sind deutlich umfangreicher als hier vorgestellt. Weiterführende Informationen diesbezüglich können [42] entnommen werden.

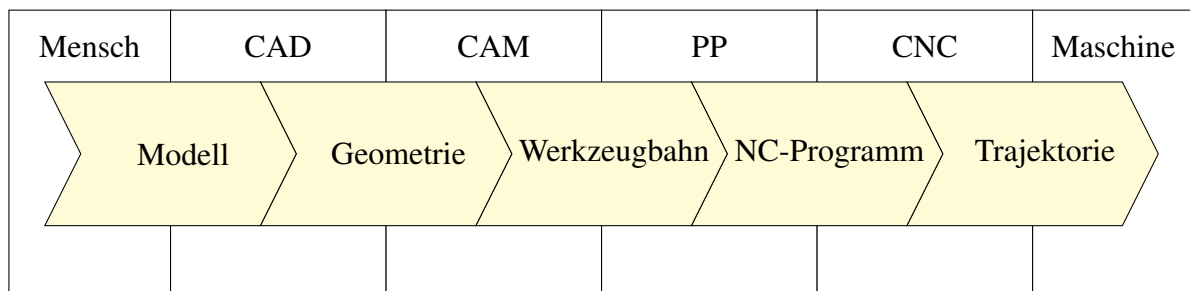


Abb. 1.1: Die schematische Darstellung der Prozesskette zur Fertigung mit numerisch gesteuerten Werkzeugmaschinen beginnt mit einem digitalen Modell, das der Mensch mit einer CAD-Software konstruiert. Durch das CAM-System und den Postprozessor wird daraus ein NC-Programm erzeugt, das von einer numerischen Steuerung verarbeitet wird.

Im Optimalfall gelingt es der numerischen Steuerung, die Dynamik der Maschine möglichst gut zu nutzen. Diese Aufgabe ist aufgrund der Komplexität der dynamischen Belastungen in einer Werkzeugmaschine, die z. B. in [14, Kapitel 2.4] beschrieben sind sehr anspruchsvoll. Da bei Werkzeugmaschinen teilweise Komponenten mit nicht unerheblicher Masse bewegt werden, gibt es hier, abhängig vom Maschinentyp, physikalische Grenzen für Position, Geschwindigkeit, Beschleunigung und Variation der Beschleunigung (Ruck). Eine weitere Messgröße zur qualitativen Bewertung des Bearbeitungsprozesses ist eine möglichst konstante Vorschubgeschwindigkeit, da auch hier Schwankungen zu Artefakten auf der Oberfläche führen. Zuletzt sei hier noch exemplarisch auf eine weitere geometrische Problematik hingewiesen, nämlich möglichen

Singularitäten der realen Maschinenkinematik. Die Vermeidung von Bewegungen, die in solchen Singularitäten enden, gehört ebenfalls zu den Aufgaben einer numerischen Steuerung. Anhand dieses groben Überblicks wird bereits ersichtlich, dass dieser Bereich eine äußerst kritische und wichtige Rolle im Rahmen der Fertigungsqualität einnimmt. In der Praxis ist eine Bahnplanung zu Lasten obiger Kriterien allerdings unumgänglich, wie nachfolgendes Beispiel zeigen soll. In Abbildung 1.2 ist ein NC-Programm in Form von G-Code

```
N1000 X 0 Y 0 Z 0 A3 -0.3162 B3 0.9487 C3 0
N1010 X 3 Y 1 Z 0 A3 -0.3162 B3 0.9487 C3 0
N1020 X 3 Y 1 Z 0 A3 0.3162 B3 0.9487 C3 0
N1030 X 6 Y 0 Z 0 A3 0.3162 B3 0.9487 C3 0
```

für eine mögliche Bewegungsführung abgebildet. Die Interpretation des Programms ist relativ intuitiv. Der Satz *N1000* liefert in dieser Betrachtung den Startwert für die Linearachsen an der Position  $(0, 0, 0)^T$  mit einer normierten Werkzeuganstellung  $(-0.3162, 0.9487, 0)^T$ , die beispielsweise durch zwei Rundachsen realisiert wird. Im nächsten Satz *N1010* verändert sich die Position auf  $(3, 1, 0)^T$  bei gleichbleibender Werkzeuganstellung, bevor anschließend im Satz *N1020* die Orientierung verändert wird, wobei die Position identisch bleibt. Im letzten Satz *N1030* wird dann das Werkzeug auf die Endposition  $(6, 0, 0)^T$  bewegt, ohne die Werkzeuganstellung zu verändern.

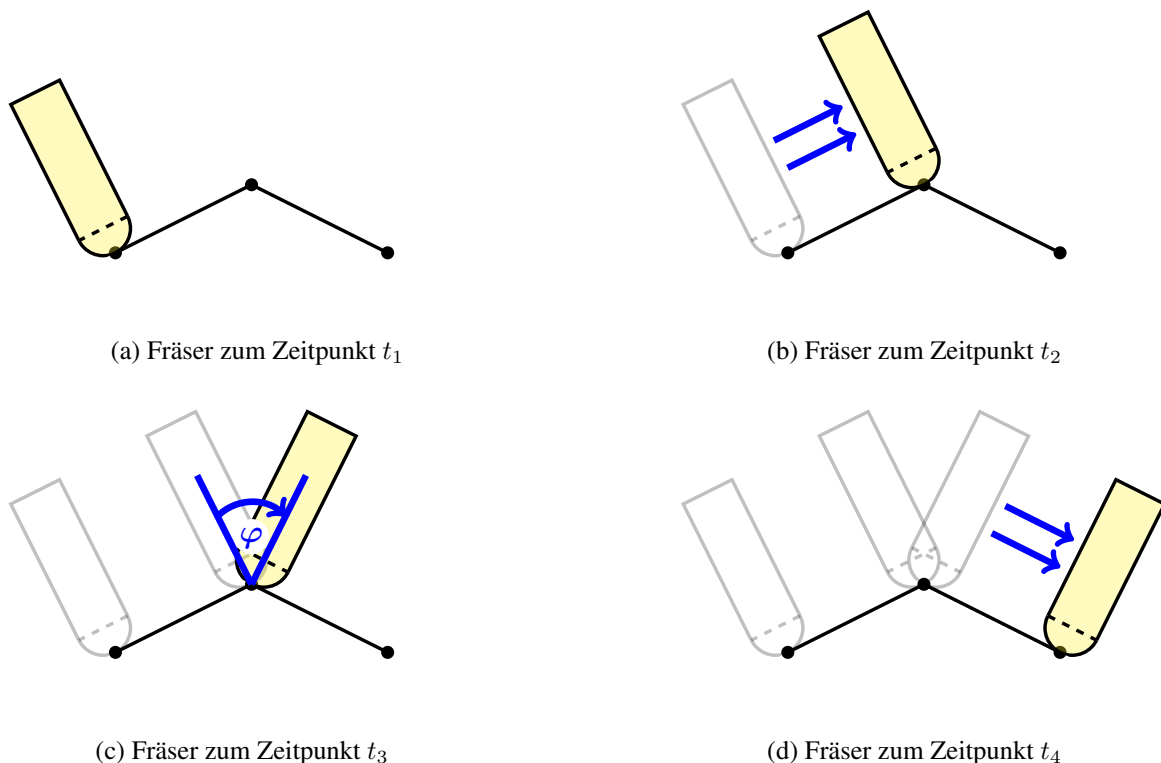


Abb. 1.2: Graphische Darstellung eines NC-Programms bestehend aus vier Sätzen. Vom Zeitpunkt  $t_1$  bis  $t_2$  wird das Werkzeug linear bewegt, ohne die Anstellung zu verändern. Sobald die Spitze im Zeitpunkt  $t_2$  erreicht ist, muss die Vorschubgeschwindigkeit reduziert werden, um die neue Anstellung zum Zeitpunkt  $t_3$  zu ermöglichen. Hierbei müssen die Rotationsachsen, die für die Anstellung zuständig sind, vom Stillstand zum Zeitpunkt  $t_2$  maximal beschleunigt und anschließend zum Zeitpunkt  $t_3$  wieder abgebremst werden.

An diesem stark vereinfachten Beispiel ist bereits zu erkennen, dass zu den Zeitpunkten  $t_2$  und  $t_3$  Sprünge in den Achs-Geschwindigkeiten auftreten, da die Werkzeuganstellung bis  $t_2$  konstant bleibt und zwischen  $t_2$  und  $t_3$  der Fräser mit einer bestimmten (möglichst hohen) Winkelgeschwindigkeit umorientiert wird. Dies hat zur Folge, dass die Vorschubgeschwindigkeit an dieser Ecke so lange 0 ist, wie die Rundachse für die Umorientierung benötigt. Dieser Sachverhalt ist nicht zwangsläufig problematisch, da es in Ecken z. B. explizit gefordert sein kann, eine solche Umorientierung durchzuführen. Unerwünscht ist dieser Effekt allerdings inmitten einer Freiformfläche oder an einem glatt konstruierten Flächenübergang. An solchen Stellen werden Marken, die durch solche Werkzeugbewegungen auf der zu bearbeitenden Oberfläche entstehen, als qualitativer Mangel klassifiziert.

Da die Ergebnisse jedes einzelnen Moduls in Abbildung 1.1 aus den Informationen der Vorgänger abgeleitet werden, liegt es nahe, dass Einflüsse am Anfang der Kette große Wirkungen auf alle nachgelagerten Prozesse sowohl an der numerischen Steuerung als auch der Maschine hervorrufen.

### 1.3 Problemstellung und Lösungsansatz

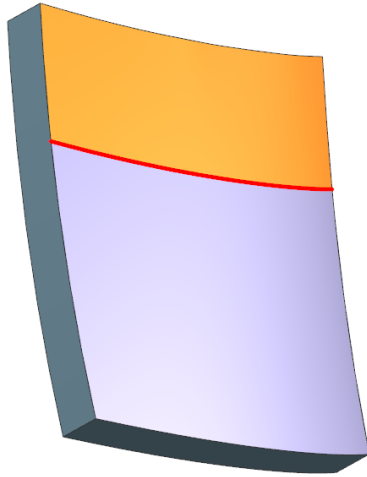
In [29] wurde ein Verfahren entwickelt, dass durch die Analyse des NC-Programms problematische Bahnabschnitte identifiziert. Diese kritischen Stellen können unterschiedliche Ausprägungen haben und entstehen z. B. durch

- wechselnde Vorzeichen der Werkzeugorientierung,
- doppelte Kontaktpunkte des Werkzeugs (mit divergenten Werkzeuganstellungen),
- inhomogene Verteilung der Punkte im NC-Programm,
- ungewollte Umorientierung des Werkzeugs ohne Bahnweg (vgl. Abbildung 1.2).

Obwohl diese und viele weitere Ausprägungen prinzipiell während der gesamten Prozesskette entstehen können, liegt die Ursache häufig bereits im CAD-Modell. In der Praxis lässt sich der Ursprung der hier genannten Ausprägungen sogar in den meisten Fällen noch genauer identifizieren. Inhomogene Abtastungen, doppelte Kontaktpunkte und ungewollte Umorientierungen auf (teils extrem) kurzen Bahnabschnitten treten oftmals an Flächenübergängen des CAD-Modells auf. Die dadurch entstehenden Effekte reichen von sichtbaren Artefakten auf der zu bearbeitenden Oberfläche bis hin zu Beschädigungen (z. B. durch Kollisionen) der Maschine. Abbildung 1.3 zeigt einen im CAD-System maximal glatt konstruierten Flächenübergang zweier Freiformflächen und das Resultat von dennoch verbleibenden, minimalen Umorientierungen im Bereich der Kante.

Thematisch einzuordnen ist diese Arbeit als weiterführender Schritt zur tiefgreifenden Analyse und Korrektur kritischer Stellen auf Basis einer Rückfrage an das CAD-System, nachdem diese durch Anwendung des in [29] vorgestellten Verfahrens im NC-Programm lokalisiert wurden. In dieser Arbeit werden neue Methoden zur Prüfung der Glattheit von Übergängen vor-





(a) CAD-Modell eines glatten Flächenübergangs



(b) Marke auf der Oberfläche im Bereich des Flächenübergangs

Abb. 1.3: Beispiel für eine ungewollte Umorientierung des Werkzeugs ohne Bahnweg, die im CAD-System an einem Flächenübergang lokalisiert werden kann. Rechts ist ein Foto eines gefrästen Bauteils abgebildet.

gestellt, die auf der Basis einer Formulierung als lineares Ausgleichsproblem numerisch stabil gelöst werden können und trotzdem geometrisch interpretierbar sind. Unter bestimmten Voraussetzungen lassen sich diese Verfahren auf gängige Begriffe zur Klassifizierung der Glattheit von Flächenübergängen zurückführen. Der Benutzer hat dadurch die Möglichkeit, im ersten Schritt die Glattheit der Kante, an der die kritische Stelle detektiert wurde, zu erfassen. Im Anschluss daran wird ein Algorithmus entwickelt, der es ermöglicht, ein NC-Programm zu korrigieren, welches bei der Abarbeitung dort kritische Stellen verursacht, wo im geometrischen Modell Flächenübergänge sind. Für die Korrektur einer oder mehrerer Bahnen wird dazu ein Bereich um die Kante ausgeschnitten und mit einer neuen, nach festgelegten Vorschriften, konstruierten Fläche, die an Start- und Endkante gewissen Glattheitsbedingungen genügt, gefüllt. Durch erneute Bahngenerierung im CAM-System kann unabhängig von der zugrundeliegenden Maschinenkinematik ein neues NC-Programm erzeugt und die entsprechende(n) Bahn(en) ersetzt werden. In Abbildung 1.4 ist das unveränderte CAD-Modell vor Anwendung des Algorithmus abgebildet.

In Abbildung 1.5 ist das CAD-Modell nach Anwendung des Algorithmus zu sehen. In einem Vorverarbeitungsschritt wurde zunächst ein Bereich um die Kante entfernt. Anschließend wurde eine neue Fläche (grün) erzeugt, die an Start- und Endkante eine definierte Glattheit besitzt, damit der Fräser beim Auf- und Abfahren der Fläche keine unerwünschten Orientierungsänderungen aufweist. Die problematischen Werkzeugbahnen können anschließend mit den neu erzeugten Bahnen ersetzt und korrigiert werden.

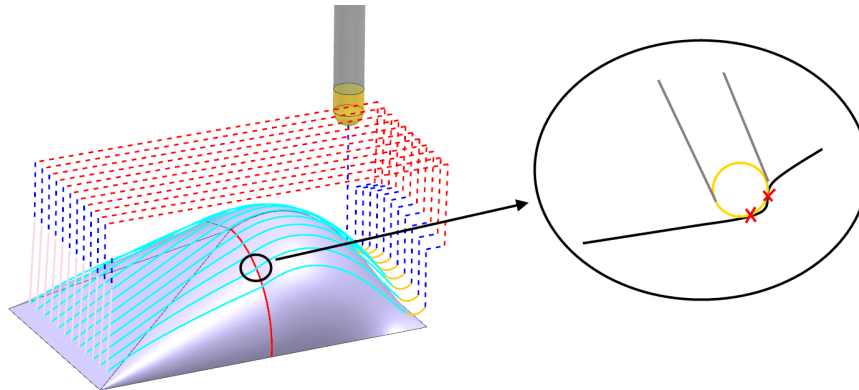


Abb. 1.4: Ausgangslage für den Algorithmus zur Korrektur von Bahnen im NC-Programm an einem Flächenübergang im CAD. Obwohl die Flächen glatt aneinanderstoßen, entsteht ein doppelter Kontaktpunkt der zu einem Einbruch der Vorschubgeschwindigkeit und einer Marke auf der Oberfläche führt.

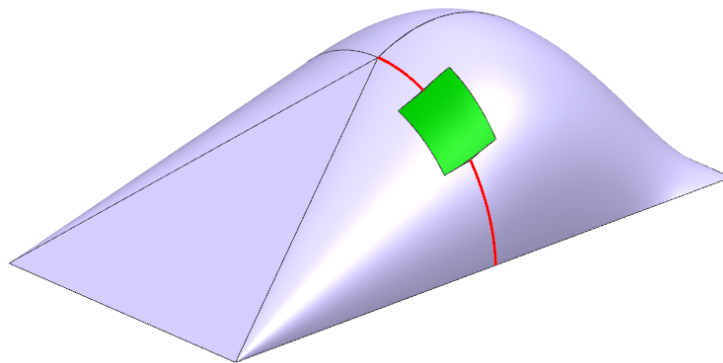


Abb. 1.5: CAD-Modell nach der Anwendung des Algorithmus. Es wurde ein Bereich um die Kante entfernt und mit einer Fläche (grün) gefüllt. Aufgrund der Glattheit an der Start- und Endkante sind sowohl das Auf- als auch das Abfahren des neuen Patches durch das Werkzeug krümmungsstetig. Alle Bahnen des NC-Programms innerhalb dieses Patches können durch das erneute Erzeugen der Werkzeugbahnen im CAM-System korrigiert werden.

Auf dieser Grundlage wurde ein weiterer Algorithmus entworfen, der durch entsprechende Anpassungen als Ausgabe eine glatte Anschlussfläche liefert. Unter einer Anschlussfläche ist in diesem Zusammenhang eine Fläche zu verstehen, die eine bereits existierende Geometrie z. B. entlang einer Kante glatt fortführt. Im Fall von kubischen Spline-Flächen wird damit eine Erweiterungsfläche erzeugt, die ohne Geometrieverlust mit der ursprünglichen Fläche vereinigt werden kann. Obwohl eine zweite Fläche im CAD-System erzeugt wurde, vermeidet man durch das Zusammenfügen beider Flächen einen Flächenübergang. Wie ein im Rahmen dieser Arbeit durchgeführter Maschinentest zeigt, ist es damit nicht nur möglich, Werkzeugbahnen zu reparieren, sondern den Prozess der Bahnplanung merklich zu verbessern. Die Erweiterungsfläche erlaubt das Fortsetzen einer glatten Werkzeugbahn über die Fläche hinaus. Damit können in der Praxis An- und Abfahrbewegungen erzeugt werden, die keine Sprünge in Geschwindigkeit, Beschleunigung oder Ruck aufweisen.

Die für diese Arbeit notwendige Kommunikation mit dem verwendeten CAD-System NX wurde über eine Schnittstelle (NXOpen) realisiert. Alle Simulationen und der Maschinentest wurden mit der numerischen Steuerung Sinumerik 840D durchgeführt. Für weitere Informationen zu aktuellen Aspekten bzgl. der Prozessoptimierung und Simulation von Zerspanprozessen sei auf [12] verwiesen.

## 1.4 Gliederung der Arbeit

In Kapitel 2 werden bekannte Resultate aus den Bereichen Differentialgeometrie und Splinetheorie sowie die verwendete Notation eingeführt. Darüber hinaus werden die Anwendung der Schnittstelle *NXOpen* sowie einige praktische Aspekte der Implementierung im Zusammenhang mit MATLAB erläutert.

Kapitel 3 beginnt zunächst mit der Betrachtung eines Glattheitsbegriffs für Kurven. In dieser Arbeit bezieht sich dieser Begriff zumeist auf die sogenannte  $G^n$ -Stetigkeit. Es werden Sätze aus der Standardliteratur behandelt und in Bezug zu den Kontrollpunkten von Splinekurven gesetzt. Die Freiheitsgrade bei der Betrachtung der  $G^1$ - und  $G^2$ -Stetigkeit werden visuell veranschaulicht und dazu verwendet, die Bestimmung einer Splinekurve mit  $G^2$ -Anschlussbedingungen, die ein Funktional minimiert, herzuleiten. Anschließend wird die  $G^n$ -Stetigkeit bezogen auf parametrische Flächen betrachtet. Dabei wird im ersten Teil eine entsprechende Notation eingeführt, um danach numerisch stabile Tests zur Prüfung auf  $G^1$ - und  $G^2$ -Stetigkeit als lineare Ausgleichsprobleme formulieren zu können. Der Zusammenhang zwischen diesen Tests und den üblichen Standardgrößen zur Bestimmung von  $G^1$ - und  $G^2$ -Stetigkeit wird hergestellt, bevor am Ende dieses Kapitels eine Anwendung zur Berechnung des Sprungs der Normalenkrümmung in einer Tangentialrichtung zweier Flächen an einem  $G^1$ -stetigen Übergang exemplarisch mit den hier vorgestellten Methoden veranschaulicht wird.

In Kapitel 4 wird zunächst eine Problemstellung aus der Praxis vorgestellt, und erläutert, warum es beim Design zweier Freiformflächen mit Glattheitsbedingungen an ihrem Übergang in CAD-Systemen zu Fertigungsproblemen an einer Werkzeugmaschine kommen kann. Anschließend wird ein konzeptionelles Ablaufschema vorgestellt, eine solche Kante im CAD-System derart anzupassen, dass durch erneute Anwendung des CAM-Systems Werkzeugbahnen repariert werden können. Darauf aufbauend wird in einzelnen Teilschritten die Erzeugung einer Fläche mit  $G^2$ -Stetigkeitsbedingungen an Start- und Endkante hergeleitet. Den Abschluss dieses Kapitels bildet die Formulierung eines linearen Gleichungssystems, dessen Lösung ein quadratisches Energiefunktional mit den aus der  $G^2$ -Stetigkeit resultierenden linearen Freiheitsgraden minimiert. Die Ergebnisse des Algorithmus sowie die Reparatur innerhalb der möglichen Gegebenheiten des CAD-Systems werden anhand von Simulationen verglichen.

In Kapitel 5 wird das Verfahren aus Kapitel 4 für eine weitere praktische Anwendung angepasst und weiterentwickelt. Dazu wird zu Beginn des Kapitels ein Überblick über die Schwierigkeiten, die bei der Planung von glatten An- und Abfahrbewegungen auftreten können, gegeben. Anschließend wird eine Funktionalität des CAD-Systems vorgestellt, deren Lösung allerdings aufgrund der gewählten Verfahrensmethode unzureichende Ergebnisse liefert. Der anschließend vor-

gestellte Algorithmus erzeugt eine glatte Erweiterungsfläche entlang einer Parameterrichtung, die im Folgeschritt durch entsprechendes Knotenentfernen mit der Ursprungsfläche verbunden werden kann, sodass kein zusätzlicher Flächenübergang entsteht. Die Vorteile dieser Variante gegenüber bisherigen Ansätzen wird im letzten Abschnitt anhand von Ergebnissen eines im Rahmen dieser Arbeit durchgeführten Maschinentests an einer 5-Achs-Fräsmaschine dargelegt.

Im nachfolgenden Kapitel werden im ersten Teil elementare Grundlagen der Differentialgeometrie sowie der Spline-Theorie behandelt, auf denen diese Arbeit basiert. Zu Beginn werden wichtige Größen von parametrischen Kurven und Flächen hergeleitet und veranschaulicht. Ziel ist es insbesondere, Aspekte, die im Zusammenhang mit  $G^{0,1,2}$ -Stetigkeit stehen, näher zu beleuchten. Dazu gehören Kurven und Flächen im Allgemeinen, deren Tangentialräume sowie Krümmungen und Flächennormalen. Der Fokus des zweiten Teils dieses Grundlagenkapitels liegt auf der speziellen Darstellung von Kurven und Flächen in heutigen CAD-Softwaresystemen als Linearkombination von B(asis)-Splines. Es werden Themen, wie die Bestimmung von Ableitungen, das Verfeinern von Knotenfolgen sowie die Interpolation und Approximation mit Kurven und Flächen behandelt. Eine Zusammenfassung wichtiger geometrischer Eigenschaften von Kurven und Flächen mit Bezug zu B-Splines ist in [43, 47] zu finden. Der letzte Abschnitt dieses Kapitels widmet sich einigen Aspekten der praktischen Umsetzung der in dieser Arbeit vorgestellten Algorithmen unter Zuhilfenahme der Schnittstelle NXOpen, sowie der Kopplung des CAD-Systems NX mit der Software MATLAB zur Berechnung numerischer Problemstellungen durch einen COM Automation Server.

## 2.1 Differentialgeometrie

Um Flächen nach qualitativen Kriterien bewerten und untersuchen zu können, ist es sinnvoll, im ersten Schritt mit der Kurventheorie zu beginnen. Wie später im Kapitel deutlich wird, sind viele Konzepte und Eigenschaften von Kurven auf Flächen übertragbar oder zumindest erweiterbar. Dieser Abschnitt gibt einen knappen Überblick, für eine vollständige Übersicht über die allgemeine Differentialgeometrie von Kurven und Flächen kann auf [33, 34, 52] zurückgegriffen werden.

### 2.1.1 Kurventheorie im euklidischen Raum

Es gibt verschiedene Möglichkeiten Kurven im Raum darzustellen. In modernen CAD-Systemen werden Kurven meistens als Splines repräsentiert und besitzen daher eine *parametrische Darstellungsform*  $f(u) : \mathbb{R} \rightarrow \mathbb{R}^3$ .

**Definition 2.1.** Eine Parametrisierung ist definiert als vektorwertige, stetig differenzierbare Funktion

$$f(u) = \begin{pmatrix} x(u) \\ y(u) \\ z(u) \end{pmatrix}, \quad a \leq u \leq b,$$

wobei jedem Parameter  $u$  ein Punkt im  $\mathbb{R}^3$  zugeordnet wird. Das Bild von  $f$  heißt Kurve.

In dieser Arbeit werden, wie im CAGD-Umfeld oft gebräuchlich, die Begriffe Parametrisierung, Kurvenparametrisierung und Kurve synonym verwendet.

**Bemerkung 2.2.** Für diese Arbeit werden darüber hinaus folgende Einschränkungen getroffen:

- Alle behandelten Splines sind mindestens zweimal stetig differenzierbar.
- Die Kurve ist regulär, das heißt die parametrische erste Ableitung von  $f$  ist an jeder Stelle  $u$  ungleich 0.

Der Parameter  $u$  muss nicht zwangsläufig ein Skalar sein, sondern kann selbst wiederum eine Funktion sein. In so einem Fall nennt man  $u$  eine gültige oder *reguläre Reparametrisierung* von  $f$ , sofern sie folgender Definition genügt.

**Definition 2.3.** Sei eine Kurve wie in Definition 2.1 gegeben. Weiter sei  $u = \varphi(\tilde{u})$  mit  $\tilde{a} \leq \tilde{u} \leq \tilde{b}$  und  $\varphi$  besitze folgende Eigenschaften:

1.  $\varphi : [\tilde{a}, \tilde{b}] \rightarrow [a, b]$  ist surjektiv.
2. Die Funktion  $\varphi$  ist stetig differenzierbar.
3.  $\varphi'(\tilde{u}) \neq 0$  für alle  $\tilde{u} \in [\tilde{a}, \tilde{b}]$ .

Dann heißt  $u = \varphi(\tilde{u})$  reguläre Reparametrisierung von  $f$ .

Für viele weitere Untersuchungen ist es notwendig, die Länge einer Kurve  $f$  zu kennen. Diese Problemstellung kann im Allgemeinen numerisch nicht exakt berechnet werden, allerdings kann ihre Lösung approximativ angenähert werden, indem man die Kurve in  $n$  lineare Segmente unterteilt, wobei

$$a = u_0 \leq u_1 \leq \dots \leq u_n = b$$

gilt.

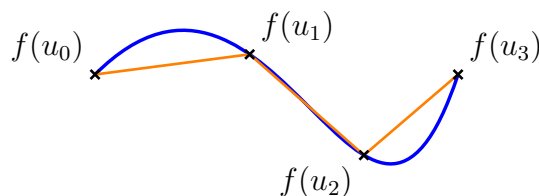


Abb. 2.1: Lineare Teilstücke einer Kurve zur Approximation der Bogenlänge

Die Länge des  $i$ -ten Segments ist dann durch

$$\ell_i = \|f(u_i) - f(u_{i-1})\|_2, \quad 1 \leq i \leq n,$$

gegeben. Sei  $Z$  die stückweise Unterteilung der Kurve in Teilstücke und bezeichne weiter  $\ell(Z) = \sum_{i=1}^n \ell_i$  die Summe aller Teillängen. Nun vergrößert man  $n$  beliebig lange, bis der Term  $\max_{1 \leq i \leq n} (u_i - u_{i-1})$  gegen 0 geht. Existiert dann ein Grenzwert für  $\ell(Z)$ , so heißt die Kurve *rektifizierbar*. Ein Beweis für das folgende, grundlegende Resultat ist wohlbekannt und z.B. in [33, Seite 26] zu finden.

**Satz 2.4.** *Sei  $f$  eine Kurve wie in Definition 2.1. Die Länge des Kurvenstücks von  $f(t_0)$  bis  $f(t_1)$  ist dann gegeben durch*

$$s(t_0, t_1) := \int_{t_0}^{t_1} \left\| \frac{df}{du} \right\|_2 du$$

und ist unabhängig von der aktuellen Parametrisierung von  $f$ . Die Funktion  $s(t_0, t_1)$  heißt *Bogenlänge des Stücks von  $f(t_0)$  bis  $f(t_1)$* .

Ersetzt man die variablen Grenzen durch den minimalen und maximalen Parameterwert  $a$  und  $b$ , erhält man die Länge der gesamten Kurve

$$s := \int_a^b \left\| \frac{df}{du} \right\|_2 du. \quad (2.1)$$

**Bemerkung 2.5.** *Ist eine Kurve nach ihrer Bogenlänge  $s(t_0, t_1)$  parametrisiert, spricht man von einer sogenannten Bogenlängenparametrisierung. Diese spezielle Parametrisierung ist für die weiteren theoretischen Herleitungen und Betrachtungen von großer Bedeutung. Da allerdings im Kontext der  $G^{0,1,2}$ -Stetigkeiten Ableitungen von lediglich regulär parametrisierten Kurven betrachtet werden, wird folgende Notation eingeführt, die nicht der Standardliteratur entspricht:*

Bogenlängenparametrisierung $f(s)$	Freie Parametrisierung $f(u)$
$\dot{f} = \frac{df}{ds}$	$f' = \frac{df}{du}$
$\ddot{f} = \frac{d^2f}{ds^2}$	$f'' = \frac{d^2f}{du^2}$
$\vdots$	$\vdots$

Um einen beliebigen Vektor  $v \neq 0$  als normierten Richtungsvektor darzustellen, wird die Notation  $\hat{v} = \frac{v}{\|v\|_2}$  verwendet.

**Definition 2.6.** *Sei eine nach der Bogenlänge parametrisierte Kurve  $f$  gegeben. Der Vektor*

$$t(s) := \lim_{h \rightarrow 0} \frac{f(s+h) - f(s)}{h} = \dot{f}(s)$$

heißt *Einheitstangentenvektor der Kurve  $f$  im Punkt  $f(s)$* .

Für den Fall, dass  $f$  beliebig parametrisiert ist, gilt

$$\dot{f}(u) = \frac{df(u)}{du} \frac{du}{ds} = \frac{f'(u)}{\|f'(u)\|_2}.$$

Mit der Kenntnis eines einzelnen Punktes und der zugehörigen ersten Ableitung  $f'(u)$  ist es bereits möglich, eine Ebene aufzuspannen:

**Definition 2.7.** Die affine Ebene

$$E = \{x \in \mathbb{R}^3 : \langle x - f(u), f'(u) \rangle = 0\}$$

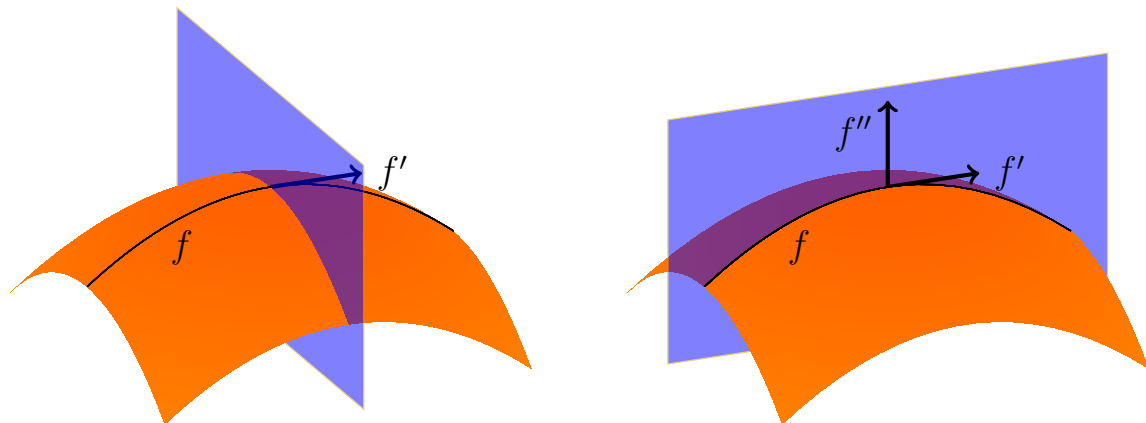
heißt Normalenebene.

Wie in Abbildung 2.2a zu sehen, ist eine solche Ebene ohne Zusatzinformationen wenig aussagekräftig. Kennt man allerdings neben der Tangente auch noch die zweite Ableitung der Kurve an der entsprechenden Stelle, lassen sich bereits einige Größen zur Analyse der Kurve ableiten.

**Definition 2.8.** Seien die beiden Ableitungen  $f'(u)$  und  $f''(u)$  linear unabhängig. Die affine Ebene

$$E(u) = f(u) + (f'(u) \quad f''(u)) \cdot \mathbb{R}^2$$

heißt Schmiegeebene.



(a) Normalenebene einer Kurve im Punkt  $f(u)$

(b) Schmiegeebene einer Kurve im Punkt  $f(u)$

Abb. 2.2: Links ist die Normalenebene einer Kurve abgebildet. Auf der rechten Seite ist die Schmiegeebene abgebildet, die von der ersten und zweiten Ableitung aufgespannt wird.

Während die Tangente einer Kurve im Punkt  $f(u)$  dadurch entsteht, dass die Gerade zwischen  $f(u)$  und einem weiteren Punkt  $P_1$  auf  $f$  betrachtet wird (wobei  $P_1$  gegen  $f(u)$  läuft), entsteht



die Schmiegeebene durch Hinzunahme eines weiteren Punktes  $P_2$  auf  $f$  der ebenfalls gegen  $f(u)$  läuft. Die Ebene, die alle drei Punkte  $f(u)$ ,  $P_1$  und  $P_2$  beinhaltet, ist dann die gesuchte Schmiegeebene. Aus dieser Anschauung heraus wird schnell ersichtlich, warum die beiden Vektoren  $f'(u)$  und  $f''(u)$  linear unabhängig sein müssen. Ansonsten ist die zugrundeliegende Kurve eine Gerade und die Schmiegeebene nicht eindeutig definiert.

Die erste und zweite Ableitung einer gekrümmten Kurve ermöglicht es also, an einem Punkt  $f(u)$  die Normalen- und Schmiegeebene darzustellen. Diese reichen bereits aus, um das Konzept der *Kurvenkrümmung* einzuführen:

**Definition 2.9.** Sei eine Kurve  $f$  gegeben, die an einem Punkt  $f(u)$  eine Normalen- und eine Schmiegeebene besitzt. Der Schnitt dieser Ebenen heißt *Kurvennormale* und ihre Richtung ist gegeben durch

$$n(s) := \ddot{f}(s).$$

Der Zusammenhang, dass die Normale sowohl in der Normalenebene als auch in der Schmiegeebene liegen muss, wird deutlich, wenn man die Eigenschaft der Einheitstangente näher betrachtet. Aus der Relation

$$\langle t(s), t(s) \rangle = 1,$$

wobei  $\langle \cdot, \cdot \rangle$  das Skalarprodukt ist, folgt durch erneutes Ableiten nach  $s$  und der Produktregel

$$\langle t(s), \dot{t}(s) \rangle = 0 \quad \Leftrightarrow \quad \langle \dot{f}(s), \ddot{f}(s) \rangle = 0.$$

Damit ist klar, dass  $\ddot{f}(s)$  orthogonal zur Tangente sein muss. Nach der Konstruktionsvorschrift der Ebene in Definition 2.8 liegt der Vektor darüber hinaus in der Schmiegeebene und ergibt sich somit aus dem Schnitt beider Ebenen. Die Länge dieses Vektors besitzt eine besondere Bedeutung:

**Definition 2.10.** Die Krümmung  $\kappa(s)$  einer nach der Bogenlänge parametrisierten Kurve an einem Punkt  $f(s)$  ist definiert durch

$$\kappa(s) := \left\| \ddot{f}(s) \right\|_2.$$

Ihr Reziprokwert heißt Krümmungsradius und wird mit  $\rho(s)$  bezeichnet.

Bevor die Krümmung für allgemeingültige Parametrisierungen hergeleitet wird, ist es hilfreich, die bereits vorhandenen Informationen kurz zusammenzufassen. Für eine Kurve  $f$  kann man folgende Größen zur Analyse des Kurvenverlaufs bestimmen (die Parameter werden aufgrund der besseren Lesbarkeit weggelassen):

Tangente $\frac{df}{du}$	$f'$
Einheitstangente $\frac{df}{ds}$	$t = \dot{f} = \frac{f'}{\ f'\ _2}$
2. Ableitung von $f : \frac{d^2f}{du^2}$	$f''$
1. Ableitung von $t : \frac{dt}{ds}$	$\dot{t} = \ddot{f}$
Krümmung der Kurve	$\kappa = \ \ddot{f}\ _2 = \ \dot{t}\ _2$

Um die Krümmung einer regulär parametrisierten Kurve zu bestimmen, fehlt noch der Zusammenhang zwischen den parametrischen Ableitungen  $f'$ ,  $f''$  und  $\kappa$ . Hierfür sei eine Kurve  $f$  und eine reguläre Bogenlängenreparametrisierung  $\varphi$  gegeben. Dann gilt für die Einheitstangente

$$(f \circ \varphi)'(u) = f'(\varphi(u))\varphi'(u)$$

$$\frac{df}{du} = t(u) \left( \frac{ds}{du} \right),$$

wobei das Symbol  $\circ$  für die Komposition zweier Funktionen verwendet wird. Betrachtet man nun die Ableitung zweiter Ordnung, muss sowohl die Ketten- als auch die Produktregel angewandt werden und es ergibt sich

$$((f \circ \varphi)'(u)\varphi'(u))' = f''(\varphi(u))(\varphi'(u))^2 + f'(\varphi(u))\varphi''(u)$$

$$\frac{d^2f}{du^2} = n \left( \frac{ds}{du} \right)^2 + t \frac{d^2s}{du^2} = \kappa \hat{n} \left( \frac{ds}{du} \right)^2 + t \frac{d^2s}{du^2}.$$

Aus obiger Formel ist es nicht möglich,  $\kappa$  zu berechnen, da sowohl die Reparametrisierung  $\frac{d^2s}{du^2}$  als auch  $\frac{ds}{du}$  unbekannt sind. Man kann allerdings mithilfe des Kreuzprodukts den Zusammenhang zwischen Einheitstangente und normierter Normalen wie folgt darstellen und anschließend auf beiden Seiten die euklidische Norm betrachten:

$$\frac{df}{du} \times \frac{d^2f}{du^2} = \frac{ds}{du} \frac{d^2s}{du^2} (t \times t) + \kappa \left( \frac{ds}{du} \right)^3 (t \times \hat{n}) = \kappa \left( \frac{ds}{du} \right)^3 (t \times \hat{n}) \quad (2.2)$$

$$\left\| \frac{df}{du} \times \frac{d^2f}{du^2} \right\|_2 = \kappa \left\| \frac{ds}{du} \right\|_2^3 = \kappa \left\| \frac{df}{du} \right\|_2^3$$

$$\kappa = \frac{\|f' \times f''\|_2}{\|f'\|_2^3} \quad (2.3)$$

Im letzten Schritt nutzt man die Tatsache, dass die Norm der Reparametrisierung  $\frac{ds}{dt}$  bekannt ist. Diese ist exakt die Länge der ersten Ableitung, also  $\left\| \frac{ds}{dt} \right\|_2 = \left\| \frac{df}{dt} \right\|_2$ . Dieser in der Differentialgeometrie elementare Satz kann in der Standardliteratur gefunden werden [33].

**Satz 2.11.** Die Krümmung einer regulär parametrisierten Kurve  $f$  an einem Parameterwert  $u$  ist gegeben durch

$$\kappa(u) = \frac{\|f'(u) \times f''(u)\|_2}{\|f'(u)\|_2^3}.$$

Eine anschauliche Vorstellung des abstrakten Krümmungsbegriffs liefert der sogenannte *Schmiegekreis*. Dieser Kreis liegt in der Schmiegeebene und ist wie folgt definiert.

**Definition 2.12.** Sei eine Kurve  $f(u)$  sowie die Ableitungen erster und zweiter Ordnung bekannt. Der Kreis mit Mittelpunkt  $M = f(u) + \rho(u)\hat{n}(u)$  und Radius  $\rho(u)$ , der in der Schmiegeebene liegt, wird *Schmiegekreis* genannt.

In Abbildung 2.3 ist exemplarisch in orange ein Schmiegekreis am Scheitelpunkt einer Parabel  $f$  dargestellt. Der Radius  $\rho$  des Kreises ist exakt der Reziprokwert der Länge der zweiten Ableitung bezüglich der Bogenlänge. Zu beachten ist, dass die Ableitung  $f''$  nicht senkrecht auf die Tangente stehen muss, sondern nur die Differenz  $\ddot{f} - f''$  ein Vielfaches der Tangente sein muss. Dieser Sachverhalt wird im Kapitel 3.1 (vgl. Beispiel 3.11) anhand der  $G^2$ -Stetigkeit näher erläutert.

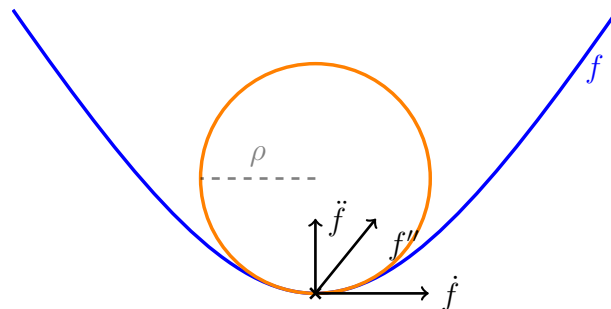


Abb. 2.3: Schmiegekreis einer Parabel  $f$  am Scheitelpunkt mit Ableitungen

An dieser Stelle taucht zwangsläufig die Frage auf, welche Bedeutung die verschiedenen theoretischen Größen (Ableitungen, Krümmungen, Bogenlänge, etc.) im praktischen Umfeld der industriellen Fertigung mit Werkzeugmaschinen besitzen. Zu den Aufgaben einer Steuerung gehört es, anhand dieser Daten eine Kurve optimal (z.B. im Sinne der Geschwindigkeitsführung) abzufahren. In diesem Zusammenhang besitzt die Parametrisierung nach der Zeit  $t$  eine besondere Bedeutung:

- $f(t)$  = Aktuelle Position des Werkzeugs
- $\dot{f}(t)$  = Geschwindigkeit des Werkzeugs
- $\ddot{f}(t)$  = Beschleunigung des Werkzeugs

Der Krümmungsverlauf einer Kurve lässt sich im CAD-System NX durch sogenannte *Krümmungskämme* darstellen. Eine solche Kurvenanalyse ist bei der Bewertung von Kurven sehr hilfreich und exemplarisch in Abbildung 2.4 zu sehen.

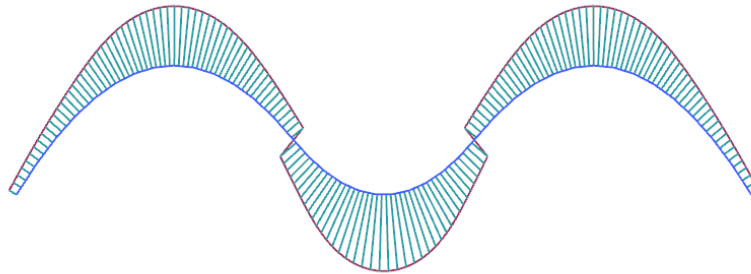


Abb. 2.4: Krümmungsanalyse in NX durch Darstellung des Krümmungskamms. Die zugrundeliegende blaue Kurve wird im ersten Schritt mit einer durch den Benutzer definierten Anzahl an Punkten abgetastet. Anschließend wird für jeden dieser Abtastpunkte die Krümmung (bzw. der Krümmungsradius) berechnet und entgegengesetzt der Kurvennormalen aufgetragen. Umso weiter der approximative Krümmungsverlauf (lila) von der Kurve entfernt ist, umso stärker ist die Kurve an dieser Stelle gekrümmt.

Die Theorie von parametrischen Kurven im  $\mathbb{R}^3$  ist an dieser Stelle noch deutlich umfassender, aber die bisher behandelten Konzepte reichen aus, um die für diese Arbeit relevanten Eigenschaften, wie z.B. Krümmungsbegriffe, auf Flächen zu definieren.

## Flächen im euklidischen Raum

Wie bereits bei den Kurven werden auch Flächen in CAD-Systemen als vektorwertige Funktionen, also mit einer parametrischen Darstellung, beschrieben.

**Definition 2.13.** *Seien  $U, V$  zwei reelle Intervalle. Die Parametrisierung einer Fläche ist definiert als stetig differenzierbare Funktion über dem rechteckigen Parameterbereich  $\Omega = U \times V = \{(u, v), u \in U, v \in V\}$  mit*

$$F(u, v) = \begin{pmatrix} F_x(u, v) \\ F_y(u, v) \\ F_z(u, v) \end{pmatrix}.$$

**Bemerkung 2.14.** *Die Begriffe Flächenparametrisierung, Parametrisierung einer Fläche und Fläche werden in der folgenden Arbeit synonym verwendet, wie es in der CAGD-Literatur zu meist üblich ist.*

Im Gegensatz zu Kurven besitzen Flächen zwei Parameter, nach denen man die Fläche ableiten kann. Aus diesem Grund bietet es sich an, die einzelnen Ableitungen spaltenweise in Matrizen zu

ordnen. Für die partiellen Ableitungen einer parametrischen Fläche wird aufgrund der besseren Lesbarkeit die Schreibweise

$$F_u(u, v) = \frac{\partial}{\partial u} F(u, v), \quad F_v(u, v) = \frac{\partial}{\partial v} F(u, v), \quad F_{uu}(u, v) = \frac{\partial^2}{\partial u^2} F(u, v), \quad \dots$$

verwendet. Die  $n$ -te Ableitung wird mit der Schreibweise

$$D^n F(u, v) = \left( \frac{\partial^n}{\partial u^n} F(u, v) \quad \frac{\partial^n}{\partial u^{n-1} \partial v} F(u, v) \quad \dots \right)$$

dargestellt. Bei einigen Betrachtungen mit geometrischem Hintergrund wird anstatt der Notation für die Ableitungen erster und zweiter Ordnung die Jakobi- und Hesse-Matrix verwendet.

**Definition 2.15.** Die Jakobi-Matrix einer Fläche  $F$  besteht aus den partiellen Ableitungen erster Ordnung und besitzt die Form

$$\mathbf{J}_F(u, v) := \begin{pmatrix} F_u(u, v) & F_v(u, v) \end{pmatrix} = \begin{pmatrix} F_{u,x}(u, v) & F_{v,x}(u, v) \\ F_{u,y}(u, v) & F_{v,y}(u, v) \\ F_{u,z}(u, v) & F_{v,z}(u, v) \end{pmatrix} = D^1 F(u, v).$$

Analog dazu heißt der Tensor, der als Matrix bestehend aus den partiellen Ableitungen zweiter Ordnung angeordnet werden kann, Hesse-Matrix und wird durch

$$\mathbf{H}_F(u, v) := \begin{pmatrix} F_{uu}(u, v) & F_{uv}(u, v) \\ F_{vu}(u, v) & F_{vv}(u, v) \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} F_{uu,x}(u, v) \\ F_{uu,y}(u, v) \\ F_{uu,z}(u, v) \end{pmatrix} & \begin{pmatrix} F_{uv,x}(u, v) \\ F_{uv,y}(u, v) \\ F_{uv,z}(u, v) \end{pmatrix} \\ \begin{pmatrix} F_{vu,x}(u, v) \\ F_{vu,y}(u, v) \\ F_{vu,z}(u, v) \end{pmatrix} & \begin{pmatrix} F_{vv,x}(u, v) \\ F_{vv,y}(u, v) \\ F_{vv,z}(u, v) \end{pmatrix} \end{pmatrix}$$

repräsentiert. Bei der Betrachtung der  $G^2$ -Stetigkeit an Flächenübergängen tauchen solche Tensoren in verschiedener Form auf. Um trotzdem einen Formalismus mithilfe von Matrix-Produkten einführen zu können, wird an dieser Stelle die Notation

$$\mathbf{H}_F^\dagger(u, v) := \begin{pmatrix} F_{uu,x}(u, v) & F_{uv,x}(u, v) \\ F_{uv,x}(u, v) & F_{vv,x}(u, v) \\ F_{uu,y}(u, v) & F_{uv,y}(u, v) \\ F_{uv,y}(u, v) & F_{vv,y}(u, v) \\ F_{uu,z}(u, v) & F_{uv,z}(u, v) \\ F_{uv,z}(u, v) & F_{vv,z}(u, v) \end{pmatrix}$$

eingeführt. Eine ausführliche Herleitung und Begründung für diese Anordnung der Hesse-Matrix ist in Kapitel 3.2 zu finden.

**Bemerkung 2.16.** Ähnlich dem Kurvenfall liegen diesem Kapitel folgende Annahmen zugrunde:

1. Die parametrische Fläche  $F$  ist mindestens zweimal stetig differenzierbar.
2. Der Rang von  $\mathbf{J}_F$  ist im gesamten Parametergebiet maximal.

3. Der Parameterbereich besitzt die Form  $\Omega = U \times V$ .

Wie bereits bei den Kurven ist es auch bei Flächen möglich, eine Koordinatentransformation, im Folgenden auch (*Flächen-*)*Reparametrisierung*  $\varphi : \tilde{\Omega} \rightarrow \Omega$  genannt, einzuführen, wobei  $\tilde{\Omega}$  ebenfalls ein kartesisches Produkt  $\tilde{U} \times \tilde{V}$  darstellt. Eine solche Anpassung der Parameterwerte ist von zentraler Bedeutung für diese Arbeit und bietet die Möglichkeit, die Ableitungen zweier aneinanderstoßender Flächen mit unterschiedlicher Parametrisierung zu vergleichen.

**Definition 2.17.** Sei eine Fläche  $F(u, v)$  mit Parameterbereich  $\Omega = U \times V$  gegeben. Eine Funktion  $\varphi : \tilde{\Omega} \rightarrow \Omega$  mit  $(\tilde{u}, \tilde{v}) \rightarrow (u, v)$  heißt reguläre Reparametrisierung, wenn folgende Eigenschaften erfüllt sind:

1.  $\varphi$  ist eine surjektive Abbildung auf  $\Omega$ .
2. Die Determinante von  $D^1\varphi(\tilde{u}, \tilde{v})$  ist ungleich 0 für alle  $(\tilde{u}, \tilde{v}) \in \tilde{\Omega}$ .

In Abbildung 2.5 ist eine Reparametrisierung  $\varphi$  veranschaulicht. Im ersten Schritt werden die Parameterwerte  $(\tilde{u}, \tilde{v})$  auf den ursprünglichen Parameterbereich  $(u, v)$  abgebildet und anschließend die Funktion  $F$  ausgewertet.

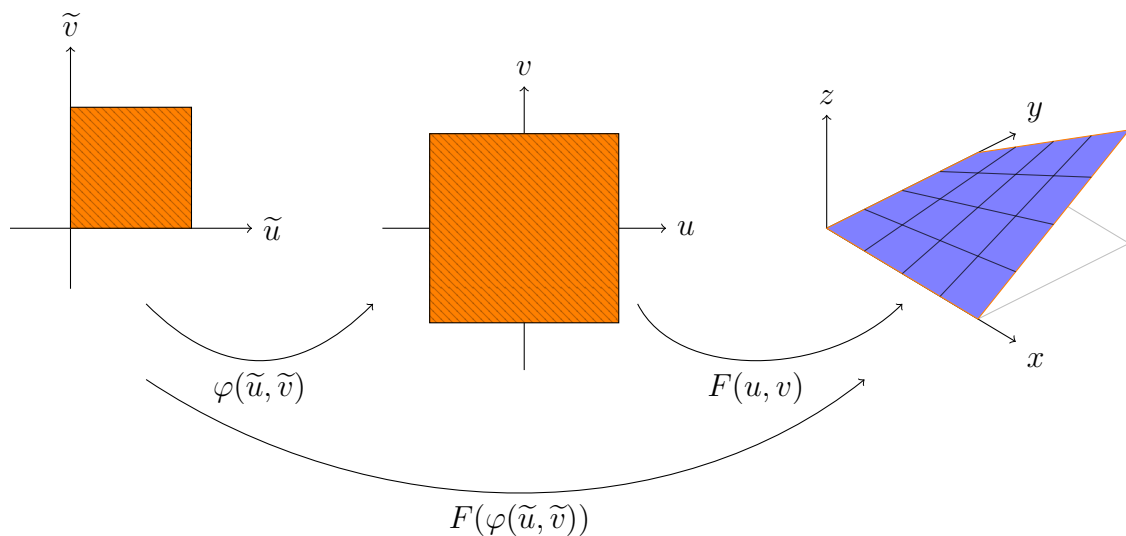


Abb. 2.5: Reparametrisierung  $\varphi$  des Parameterbereichs und anschließende Auswertung der Fläche  $F$ .

Ein erstes wichtiges Hilfsmittel zur Untersuchung einer Fläche ist die Tatsache, dass für jeden Punkt  $F(u, v)$  durch Kenntnis der beiden partiellen Ableitungen  $F_u(u, v)$  und  $F_v(u, v)$  eine Ebene aufgespannt werden kann, die sogenannte *Tangentialebene*.

**Definition 2.18.** Sei eine Fläche  $F$  wie in Definition 2.13 gegeben. Die im Punkt  $F(u, v)$  durch die beiden partiellen Ableitungen  $F_u(u, v)$  und  $F_v(u, v)$  aufgespannte affine Ebene

$$T_F(u, v) = F(u, v) + \mathbf{J}_F(u, v) \cdot \mathbb{R}^2$$

heißt Tangentialebene und der zugehörige Vektor

$$n_F(u, v) := F_u(u, v) \times F_v(u, v)$$

$$\hat{n}_F(u, v) := \frac{F_u(u, v) \times F_v(u, v)}{\|F_u(u, v) \times F_v(u, v)\|_2}$$

(Flächen-)Normalenvektor von  $F$  an der Stelle  $(u, v)$ .

Für viele weitere Eigenschaften einer Fläche reicht es nicht aus, nur einen Punkt  $F(u, v)$  zu betrachten. Untersucht man beispielsweise einen beliebigen Punkt auf der Mantelfläche eines Zylinders, so gibt es von diesem Punkt aus Richtungen, in denen die Fläche keine Krümmung besitzt und andere, in denen der Krümmungsradius genau dem Radius des Zylinders entspricht (vgl. Abbildung 2.6).

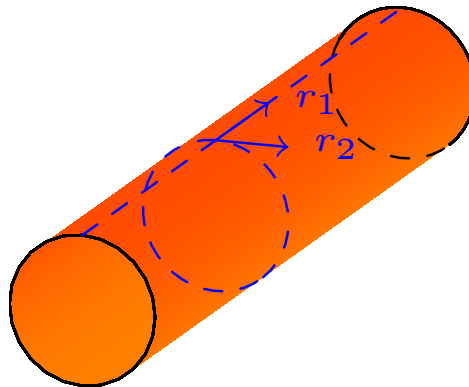


Abb. 2.6: Ein beliebiger Punkt auf einer Mantelfläche besitzt je nach gewählter Richtung verschiedene Krümmungen. Während die Krümmung in Richtung  $r_1$  stets 0 ist, so ist in Richtung  $r_2$  der Krümmungsradius gleich dem Kreisradius, mit dem der Zylinder erzeugt wurde.

Um die Krümmung einer Fläche in der Umgebung eines Punktes zu bestimmen, muss man Kurven auf der Fläche, die durch den entsprechenden Punkt verlaufen, näher untersuchen. Eine solche Kurve, die ausschließlich auf einer Fläche  $F(u, v)$  existiert, kann mit

$$u = u(t), \quad v = v(t), \quad F(u(t), v(t))$$

dargestellt werden. Leitet man eine solche Kurve ab, erhält man für  $F(u(t), v(t))'$

$$\frac{dF(u(t), v(t))}{dt} = \frac{dF(u(t), v(t))}{du} \frac{du}{dt} + \frac{dF(u(t), v(t))}{dv} \frac{dv}{dt} = F_u(u(t), v(t)) \frac{du}{dt} + F_v(u(t), v(t)) \frac{dv}{dt}.$$

Es ist sofort zu erkennen, dass die Ableitung jeder Kurve durch diesen Punkt immer in der Tangentialebene  $T_F(u(t), v(t))$  liegt. Um die bekannte Kurventheorie auch auf Flächen anwenden

zu können, ist es notwendig, die Bogenlänge einer Kurve auf einer Fläche berechnen zu können.

$$\begin{aligned}
 s(t^*) &= \int_a^{t^*} \left\| \frac{dF(u(t), v(t))}{dt} \right\|_2 dt \\
 &= \int_a^{t^*} \left\| F_u(u(t), v(t)) \frac{du(t)}{dt} + F_v(u(t), v(t)) \frac{dv(t)}{dt} \right\|_2 dt \\
 &= \int_a^{t^*} \sqrt{\left( F_u(u(t), v(t)) \frac{du(t)}{dt} + F_v(u(t), v(t)) \frac{dv(t)}{dt} \right)^2} dt. \\
 (s')^2 &= \left( F_u(u(t), v(t)) \frac{du(t)}{dt} + F_v(u(t), v(t)) \frac{dv(t)}{dt} \right)^2 \\
 &= (F_u(u(t), v(t))\dot{u}(t) + F_v(u(t), v(t))\dot{v}(t))^2.
 \end{aligned}$$

In der Literatur wird meistens für die Skalarprodukte der partiellen Ableitungen

$$\begin{aligned}
 \mathcal{E} &= \langle F_u(u(t), v(t)), F_u(u(t), v(t)) \rangle \\
 \mathcal{F} &= \langle F_u(u(t), v(t)), F_v(u(t), v(t)) \rangle \\
 \mathcal{G} &= \langle F_v(u(t), v(t)), F_v(u(t), v(t)) \rangle
 \end{aligned}$$

geschrieben.

**Definition 2.19.** Die quadratische Form

$$ds^2 = \mathcal{E} du^2 + 2\mathcal{F} dudv + \mathcal{G} dv^2$$

heißt erste Fundamentalform von  $F$ .

**Definition 2.20.** Anhand der Größen  $\mathcal{E}$ ,  $\mathcal{F}$  und  $\mathcal{G}$  werden folgende Begriffe über die zugrundeliegende Parametrisierung eingeführt:

- Gilt  $\mathcal{F} = 0$ , so heißt die Parametrisierung orthogonal, d.h. die beiden partiellen Ableitungen  $F_u$  und  $F_v$  sind orthogonal zueinander.
- Gilt für eine orthogonale Parametrisierung zusätzlich  $\mathcal{E} = \mathcal{G}$ , so wird diese konform genannt.
- Wenn  $\mathcal{E} = \mathcal{G} = 1$  und  $\mathcal{F} = 0$  gilt, so liegt eine isometrische Parametrisierung zugrunde.

Eine isometrische Parametrisierung hat den Vorteil, dass sowohl Längen- als auch Winkeltreue zwischen dem Parameter- und Bildbereich herrscht. Um das Verhalten einer Fläche in der Umgebung eines Punktes zu analysieren, reicht es jedoch nicht aus, eine einzelne Kurve zu betrachten, wie nachfolgendes Beispiel zeigen soll.



**Beispiel 2.21.** Die Schnittkurve eines Kreiskegels mit einer Ebene parallel zur seiner Grundfläche liefert einen Kreis. Die erste Ableitung  $\dot{f}$  liegt tangential an, während  $\hat{n}_f = \frac{\dot{f}}{\kappa}$  in Richtung des Mittelpunkts des Kreises zeigt. Zieht man in diese Betrachtung zusätzlich die Flächennormale  $\hat{n}_F$  mit ein, so liegt diese zwar ebenfalls in der Normalenebene von  $f$ , aber unterscheidet sich von  $\hat{n}_f$ .

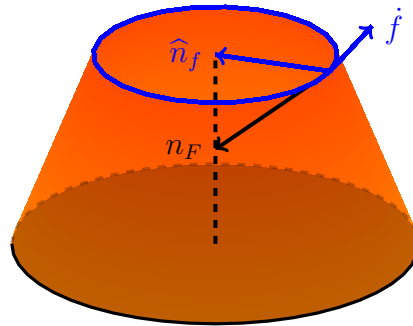


Abb. 2.7: Vergleich zwischen Normalenvektor der Schnittkurve  $f$  und der Flächennormale  $n_F$

Da sowohl  $\hat{n}_f$  als auch  $\hat{n}_F$  normierte Vektoren sind, gilt für den Winkel  $\alpha$  zwischen ihnen

$$\begin{aligned} \cos(\alpha) &= \langle \hat{n}_f, \hat{n}_F \rangle \\ \kappa \cos(\alpha) &= \langle \dot{f}, \hat{n}_F \rangle. \end{aligned} \quad (2.4)$$

Betrachtet man die allgemeine Ableitung unter einer gültigen Reparametrisierung von  $F$ , so gilt:

$$\begin{aligned} \dot{F} &= F_u \frac{du}{ds} + F_v \frac{dv}{ds}, \\ \ddot{F} &= F_{uu} \left( \frac{du}{ds} \right)^2 + 2F_{uv} \frac{du}{ds} \frac{dv}{ds} + F_{vv} \left( \frac{dv}{ds} \right)^2 + F_u \frac{d^2u}{ds^2} + F_v \frac{d^2v}{ds^2}. \end{aligned}$$

Nun multipliziert man an die Gleichung der zweiten Ableitung die normierte Flächennormale und nützt dabei aus, dass die partiellen ersten Ableitungen  $F_u$  und  $F_v$  jeweils senkrecht auf  $\hat{n}_F$  stehen:

**Definition 2.22.** Der Ausdruck

$$\langle \ddot{F}, \hat{n}_F \rangle = \underbrace{\langle F_{uu}, \hat{n}_F \rangle}_{\mathcal{L}} \left( \frac{du}{ds} \right)^2 + 2 \underbrace{\langle F_{uv}, \hat{n}_F \rangle}_{\mathcal{M}} \frac{du}{ds} \frac{dv}{ds} + \underbrace{\langle F_{vv}, \hat{n}_F \rangle}_{\mathcal{N}} \left( \frac{dv}{ds} \right)^2$$

heißt zweite Fundamentalform von  $F$  mit den Koeffizienten  $\mathcal{L}, \mathcal{M}, \mathcal{N}$ .

**Bemerkung 2.23.** Der Grund für die Multiplikation mit  $\hat{n}_F$  liegt darin, dass in die Betrachtung nur der Anteil in Richtung der Flächennormale eingehen soll. Untersucht man z.B. eine beliebig

gekrümmte Kurve auf der  $xy$ -Ebene, so wird mithilfe der Flächennormalen sichergestellt, dass bei der Berechnung der Krümmung der Fläche nur der Normalenanteil eine Rolle spielt und das Ergebnis durch eine „ungünstige“ Wahl der Kurve nicht verfälscht wird. Die Krümmung in der Tangentialebene eines Untersuchungspunktes, die orthogonal zur Flächennormale  $\hat{n}_F$  ist, heißt in der Literatur geodätische Krümmung (vgl. [33, Kapitel 5]).

Für eine regulär parametrisierte Fläche  $F(u, v)$  kann nun mit der zweiten Fundamentalform und dem Zusammenhang

$$\dot{u} = \frac{du}{ds} = du \frac{1}{s'}$$

die Gleichung (2.4) umformuliert werden:

$$\begin{aligned} \kappa \cos(\alpha) &= \frac{\langle F_{uu}, \hat{n}_F \rangle (du)^2 + 2 \langle F_{uv}, \hat{n}_F \rangle du dv + \langle F_{vv}, \hat{n}_F \rangle (dv)^2}{\mathcal{E}(du)^2 + 2\mathcal{F} du dv + \mathcal{G}(dv)^2} \\ \kappa \cos(\alpha) &= \frac{\mathcal{L}(du)^2 + 2\mathcal{M} du dv + \mathcal{N}(dv)^2}{\mathcal{E}(du)^2 + 2\mathcal{F} du dv + \mathcal{G}(dv)^2}. \end{aligned}$$

Fixiert man eine Tangentialrichtung (also das Verhältnis von  $du$  und  $dv$ ) an einem Punkt, so ist die rechte Seite obiger Gleichung konstant. Geometrisch bedeutet das, dass die Krümmung  $\kappa$  nur von zwei Faktoren abhängt:

- Einheitstangente der Kurve auf  $F$  (= Tangentialrichtung)
- Flächennormale  $\hat{n}_F$

Der geometrische Zusammenhang und die Bedeutung dieser Aussage war bereits Meusnier [33, Seite 121] bekannt.

**Bemerkung 2.24.** Die Theorie der Krümmung von Kurven auf einer Fläche  $F$  durch einen Punkt  $F(u, v)$  für eine gegebene Tangentialrichtung  $(x, y)$  kann auf ein deutlich überschaubareres Problem zurückgeführt werden. Möchte man die Krümmung einer solchen Kurvenschar berechnen, reicht es aus, die Krümmung derjenigen Kurve zu bestimmen, die durch den Schnitt der Fläche  $F$  und der durch die Tangentialrichtung und der Flächennormalen  $n_F$  aufgespannten Ebene zu bestimmen. Diese daraus resultierende Kurve wird üblicherweise als Normalschnittkurve bezeichnet. Die Krümmung dieser ebenen Kurve am Punkt  $F(u, v)$  stimmt mit der Krümmung aller Kurven auf  $F$  durch den Punkt  $F(u, v)$  mit gleicher Tangentialrichtung und Schmiegeebene überein.

**Definition 2.25.** Die Krümmung der in Bemerkung 2.24 beschriebenen Kurve  $f$  (siehe Abbildung 2.8) heißt Normalenkrümmung und ist durch

$$\kappa_n(x, y) = \frac{\mathcal{L}x^2 + 2\mathcal{M}xy + \mathcal{N}y^2}{\mathcal{E}x^2 + 2\mathcal{F}xy + \mathcal{G}y^2} \quad (2.5)$$

gegeben.

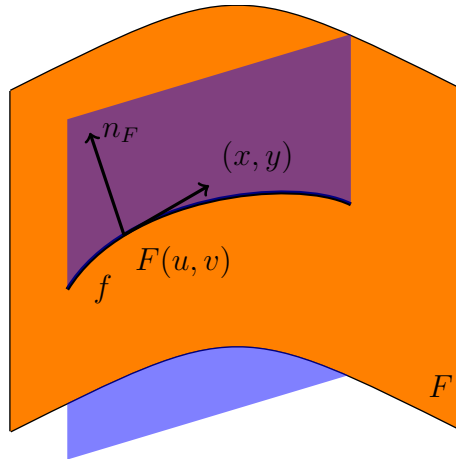


Abb. 2.8: Jede Kurve durch den Punkt  $F(u, v)$  mit identischer Tangentialrichtung und gleicher Schmiegeebene (hier in blau dargestellt) besitzt dieselbe Krümmung  $\kappa$ . Zu diesen Kurven gehört insbesondere auch diejenige (ebene) Kurve  $f$ , die durch den Schnitt von  $F$  und der durch die Tangentialrichtung und  $n_F$  aufgespannten Ebene entsteht.

Die Normalenkrümmung ermöglicht es, das Krümmungsverhalten einer Fläche  $F$  in einer Umgebung des Punktes  $F(u, v)$  zu klassifizieren und in zwei Gruppierungen zu unterteilen. In der ersten Gruppe ist die Normalenkrümmung unabhängig von der gewählten Tangentialrichtung, das gilt z.B. für Punkte auf einer Ebene oder einer Kugel. Im zweiten Fall besteht eine Abhängigkeit zwischen der gewählten Tangentialrichtung und der Normalenkrümmung der Fläche, d.h. es gibt Richtungen  $(x, y)$ , an denen  $\kappa_n$  eindeutig maximal bzw. minimal wird. Die beiden Richtungen  $r_1$  und  $r_2$ , für die  $\kappa_n$  einen Extremalwert annimmt, heißen *Hauptkrümmungsrichtungen* und können durch die Nullstellen der partiellen Ableitungen von (2.5) berechnet werden.

$$\begin{aligned} \frac{d}{dx} \kappa_n &= x^2(\mathcal{F}\mathcal{L} - \mathcal{E}\mathcal{M}) + xy(\mathcal{G}\mathcal{L} - \mathcal{E}\mathcal{N}) + y^2(\mathcal{G}\mathcal{M} - \mathcal{F}\mathcal{N}), \\ \frac{d}{dy} \kappa_n &= x^2(\mathcal{E}\mathcal{M} - \mathcal{F}\mathcal{L}) + xy(\mathcal{E}\mathcal{N} - \mathcal{G}\mathcal{L}) + y^2(\mathcal{F}\mathcal{N} - \mathcal{G}\mathcal{M}), \\ 0 &= \begin{vmatrix} x^2 & -xy & y^2 \\ \mathcal{E} & \mathcal{F} & \mathcal{G} \\ \mathcal{L} & \mathcal{M} & \mathcal{N} \end{vmatrix}. \end{aligned}$$

Die Lösungspaare dieser quadratischen Gleichung liefern die Richtungen in der Tangentialebene, für die  $\kappa_1$  minimal und  $\kappa_2$  maximal wird. Die beiden zugehörigen Werte der Normalenkrümmung heißen *Hauptkrümmungen*.

**Definition 2.26.** Das Produkt beider Hauptkrümmungen ist durch

$$\kappa_G := \kappa_1 \kappa_2$$

definiert und heißt Gauß-Krümmung. Das arithmetische Mittel

$$\kappa_M := \frac{1}{2} (\kappa_1 + \kappa_2)$$

heißt mittlere Krümmung.

**Beispiel 2.27.** Gegeben ist eine Kurve, die in Richtung eines festgelegten Vektors im Raum verschoben wird. Diese Operation erzeugt Schiebeflächen und wird im CAD-Umfeld extrudieren genannt. Sie wird häufig dazu verwendet, um eine zweidimensionale Zeichnung in den dreidimensionalen Raum zu erweitern. Aufgrund der linearen Erweiterung besitzt die Fläche in einer der beiden Parameterrichtungen keine Krümmung, d.h.  $\kappa_1 = 0$ . Damit ist die Gauß-Krümmung auf der gesamten Fläche stets 0, während die mittlere Krümmung auf der Fläche variiert.

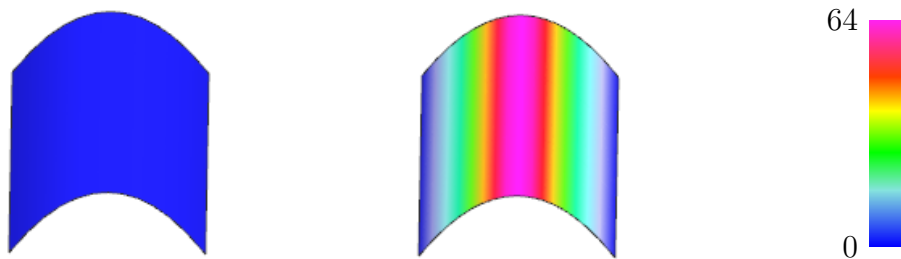


Abb. 2.9: Krümmungsplots einer parametrischen Fläche mit Gauß-Krümmung (links) und mittlerer Krümmung (rechts) generiert in NX. Während die Gauß-Krümmung für diese Fläche stets 0 ist, variiert die mittlere Krümmung entlang der gekrümmten Richtung und ist am Scheitelpunkt maximal.

In modernen CAD-Systemen hat sich die Repräsentation von Kurven und Flächen mit B-Splines durchgesetzt. Aufgrund ihrer Flexibilität und effizienten Speicherhaltung eignen sich Splines sehr gut zur Interpolation und Approximation verschiedener Datensätze und sind im digitalen Design-Umfeld nicht mehr wegzudenken. Die bisher behandelte allgemeine Kurven- und Flächentheorie lässt sich mit den folgenden Grundlagen uneingeschränkt auf Splines anwenden.

## 2.2 B-Splines

Dieser Abschnitt beschränkt sich ebenfalls auf die für diese Arbeit wesentlichen Grundlagen der Spline-Theorie und gibt einen Überblick über die verwendeten Algorithmen und wichtige Eigenschaften. Analog zu Kapitel 2.1 werden zunächst Kurven behandelt und anschließend die Verfahren auf Flächen erweitert. Ein ausführlicher Einblick sowohl in die theoretischen Aspekte der Spline-Theorie als auch in anwendungsbezogene Beispiele findet sich unter anderem in [10, 22]. Ein Großteil der Notation deckt sich mit der aus [49].

## 2.2.1 Splinekurven

Splinekurven bestehen aus abschnittsweise definierten Polynomen, die an ihren Endpunkten, auch *Knoten* genannt, zusammengefügt werden.

**Definition 2.28.** Eine Menge  $T_{m,n} = \{t_1, \dots, t_{m+n+1}\} \in \mathbb{R}^{m+n+1}$  heißt Knotenfolge der Ordnung  $m$ , wenn

$$t_1 \leq \dots \leq t_{m+n+1}, \quad t_i < t_{i+m+1}, \quad i = 1, \dots, n$$

gilt. Im Folgenden werden die Indizes  $m, n$  bei der Bezeichnung der Knotenfolge weggelassen, sofern aus dem Kontext die Wahl beider Werte klar ist.

**Bemerkung 2.29.** Die Vielfachheit eines Knotens  $t_i \in T$  gibt die Häufigkeit eines identischen Knotens an

$$t_1 \leq \dots < t_i = \dots = t_{i+\mu} < t_{i+\mu+1} \leq \dots \leq t_{m+n+1}$$

und wird mit  $\mu$  bezeichnet.

Auf einer monoton steigenden Knotenfolge  $T$  kann man die B-Splines  $N_j^k$  definieren:

**Definition 2.30.** Für einen gegebenen Knotenvektor  $T$  sind die B-Splines  $N_j^k$  der Ordnung  $k+1$  rekursiv definiert durch

$$N_j^0(\cdot | T) = \chi[t_j, t_{j+1}),$$

$$N_j^k(\cdot | T) = \frac{\cdot - t_j}{t_{j+k} - t_j} N_j^{k-1}(\cdot | T) + \frac{t_{j+k+1} - \cdot}{t_{j+k+1} - t_{j+1}} N_{j+1}^{k-1}(\cdot | T),$$

wobei  $j = 1, \dots, n + m - k$  gilt und  $\chi$  die charakteristische Funktion darstellt.

Einer großer Vorteil von B-Splines ist die Tatsache, dass der Träger eines B-Splines immer kompakt ist, obwohl der zugrundeliegende Knotenvektor  $T$  nicht zwangsläufig endlich sein muss (z.B.  $T = \mathbb{Z}$ ). Eine rekursive Auflösung ist beispielhaft in Abbildung 2.10 dargestellt.

An einem  $(k+1)$ -fachen Knoten gilt für die Knotendifferenz  $t_{j+k+1} - t_j = 0$ . Wie aus dem Algorithmus von de Boor (Algorithmus 2.1) hervorgeht, wird der Term nicht benötigt und deshalb ignoriert und nicht in die Rekursionsgleichung mit einbezogen.

**Bemerkung 2.31.** In dieser Arbeit besitzen alle Knotenvektoren  $T$ , soweit nicht explizit anders angegeben, stets  $(m+1)$ -fache Randknoten und innere Knoten der Vielfachheit 1. Die hier vorgestellten Ergebnisse und Algorithmen sind ohne Weiteres auf Splinekurven und Tensorprodukt-Flächen beliebigen Grades anwendbar, werden hier aber aus Gründen der Einheitlichkeit mit  $m = 3$  vorgestellt.

**Definition 2.32.** Zu einer gegebenen Knotenfolge  $T$  und Grad  $m$  bezeichnet der Spline-Raum  $\mathbb{S}_{m,T}$  die Menge aller stückweisen Polynome vom Maximalgrad  $m$ , die an ihren Übergangsstellen bzw. Knoten  $(m-\mu)$ -mal stetig differenzierbar sind und außerhalb der konvexen Hülle von  $T$  verschwinden.

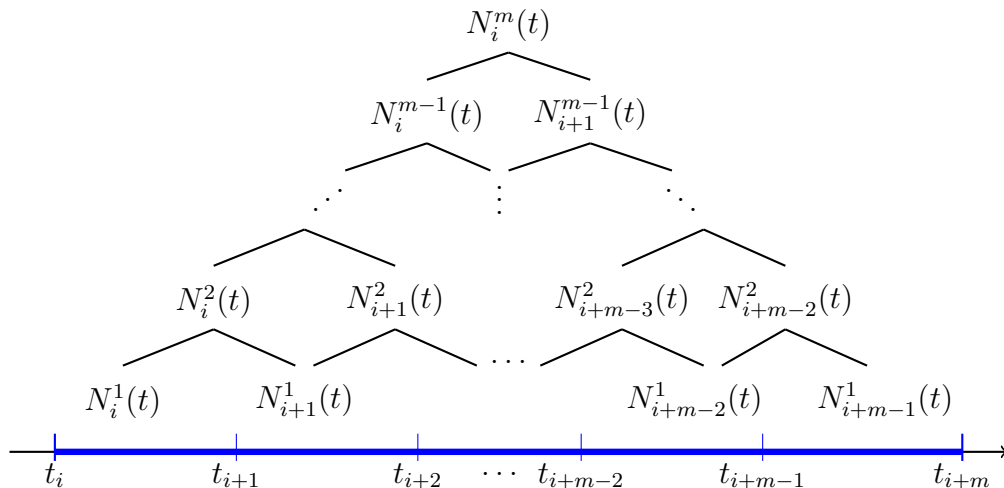


Abb. 2.10: Rekursion des  $i$ -ten B-Splines vom Grad  $m$

Ausgehend von obiger Definition wurde folgende Aussage, die als Satz von Curry und Schoenberg bekannt ist, bewiesen [10]:

**Satz 2.33.** *Sei der Knotenvektor  $T$  sowie der Grad  $m$  gegeben. Dann stellen die B-Splines  $N_j^m(\cdot | T)$  eine Basis von  $\mathbb{S}_{m,T}$  dar.*

Mit dieser wichtigen Eigenschaft ist es möglich, alle stückweisen Polynomkurven (mit einer globalen Differenzierbarkeit abhängig von  $T$ ) als B-Splines darzustellen.

**Definition 2.34.** *Eine Splinekurve im  $\mathbb{R}^3$  vom Grad  $m$  und Knotenvektor  $T$  ist definiert durch*

$$S_m \mathbf{d} = \sum_{j=1}^n d_j N_j^m(\cdot | T),$$

wobei  $\mathbf{d} \in \mathbb{R}^{3 \times n}$  das Kontrollpolygon und  $d_j$  den  $j$ -ten Kontrollpunkt bezeichnet.

**Bemerkung 2.35.** *Für die in dieser Arbeit verwendeten Splinekurven wird folgende Notation festgelegt:*

$f$	Splinekurve
$f.\mathbf{d}$	Kontrollpolygon der Kurve $f$
$f.T$	Knotenvektor der Kurve $f$
$f.m$	Grad der Kurve $f$
$[f.T]$	Intervall $[t_{m+1}, t_{n+1}]$

An dieser Stelle sei darauf hingewiesen, dass das Knotenintervall  $T$  einer Splinekurve soweit nicht anders angegeben am Anfang und Ende jeweils einen Knoten der Vielfachheit  $m + 1$  besitzt und  $[T]$  daher das Intervall  $[t_{m+1}, t_{n+1}]$  darstellt. Sofern aus dem Kontext ersichtlich ist, um

welche Kurve es sich handelt, wird das Präfix „f.“ zugunsten der besseren Lesbarkeit weggelassen. Insbesondere in Kapitel 3.1 bei der Behandlung von  $G^{0,1,2}$ -stetigen Übergängen wird diese Notation Verwendung finden.

Wie im vorigen Abschnitt dargelegt, beginnt die Analyse einer Kurve  $f$  mit der Auswertung  $f(u)$  an einer Stelle  $u \in [T]$ . Für Splinekurven bietet der Algorithmus von de Boor [8] eine effiziente Möglichkeit zur Berechnung des Wertes  $f(u)$ :

— Algorithmus (de Boor) —

Eingabe :

Knotenfolge  $T_{m,n}$

Kontrollpunkte  $\mathbf{d} = d_1, \dots, d_n$

$u \in [t_{m+1}, t_{n+1}]$

Prozedur :

*Setze*  $l \in \{m+1, \dots, n\}$ , sodass  $u \in [t_l, t_{l+1})$

$d_j^0(u) = d_j, j = l-m, \dots, l$

*For*  $k = 1, \dots, m$

*For*  $j = l-m+k, \dots, l$

$\alpha_j(u) = \frac{t_{j+m-k+1}-u}{t_{j+m-k+1}-t_j}$

$d_j^k(u) = \alpha_j(u)d_{j-1}^{k-1}(u) + (1-\alpha_j(u))d_j^{k-1}(u)$

*End*

*End*

Ausgabe :

$S_m \mathbf{d}(u) = d_l^m(u)$

Algorithmus 2.1: Algorithmus von de Boor

**Bemerkung 2.36.** Die Summe über die Produkte der einzelnen Basisfunktionen und den entsprechenden Kontrollpunkten aus Definition 2.34 kann auch als Produkt zweier Matrizen geschrieben werden. Dazu kann die sogenannte Vandermonde-Matrix verwendet werden.

**Definition 2.37.** Sei ein Knotenvektor  $T$  und eine Menge an Punkten im Parameterbereich  $U = \{u_1, u_2, \dots, u_q\}, u_i \in [T], 1 \leq i \leq q$ , gegeben. Die Matrix

$$\mathbf{V}(U) = \begin{pmatrix} N_1^m(u_1 | T) & N_1^m(u_2 | T) & \dots & N_1^m(u_q | T) \\ N_2^m(u_1 | T) & N_2^m(u_2 | T) & \dots & N_2^m(u_q | T) \\ \vdots & \vdots & \ddots & \vdots \\ N_n^m(u_1 | T) & N_n^m(u_2 | T) & \dots & N_n^m(u_q | T) \end{pmatrix}$$

heißt (verallgemeinerte) Vandermonde-Matrix oder Kollokationsmatrix.

Die Struktur einer solchen Matrix besteht aus einer Basis (in diesem Fall die der B-Splines  $N_j^m$ ) und einer Menge von Auswertestellen  $U$ . Sollte die Menge von Auswertestellen  $U$  aus einem einzelnen Punkt bestehen, besitzt die Vandermonde-Matrix die Struktur eines  $(n \times 1)$ -Vektors. Aufgrund der besseren Lesbarkeit wird in solchen Fällen auf die geschweiften Mengenklammern verzichtet. Für eine einelementige Menge  $\{u\}$  wird im Folgenden also  $\mathbf{V}(u)$  statt  $\mathbf{V}(\{u\})$  verwendet. Mithilfe der Vandermonde-Matrix ist es nun möglich, eine kompakte Schreibweise für die Auswertung einer Splinekurve an einer oder mehrerer Stellen einzuführen:

$$\begin{pmatrix} \sum_{j=1}^n d_j N_j^m(u_1 | T) \\ \sum_{j=1}^n d_j N_j^m(u_2 | T) \\ \dots \\ \sum_{j=1}^n d_j N_j^m(u_q | T) \end{pmatrix}^T = \underbrace{\mathbf{d}}_{3 \times n} \underbrace{\mathbf{V}(U)}_{n \times q}, \quad U = \{u_1, u_2, \dots, u_q\}.$$

Obwohl die Vandermonde-Matrix sehr häufig im Kontext von Interpolationsproblemen (also der Auswertung an mehreren Stellen) auftritt, wird diese aufgrund der Lesbarkeit im Folgenden auch für punktweise Auswertungen einer Splinekurve verwendet.

Die nächste grundlegende Operation auf Splinekurven ist die Bestimmung der Ableitung an einer Stelle  $u \in [T]$ . Dazu müssen im ersten Schritt die Ableitungen der B-Splines berechnet werden und anschließend mit dem Kontrollpolygon  $\mathbf{d}$  auf die Splinekurve übertragen werden. Die Formel zur Bestimmung der Ableitung von B-Splines ist allgemein bekannt und findet sich z.B. in [49, Kapitel 8].

**Lemma 2.38.** Die erste Ableitung eines B-Splines  $N_j^m(u | T)$  ist gegeben durch

$$\frac{d}{du} N_j^m(u | T) = \frac{m}{t_{j+m} - t_j} N_j^{m-1} - \frac{m}{t_{j+m+1} - t_{j+1}} N_{j+1}^{m-1}(u | T).$$

Wendet man die Ableitungsformel für B-Splines nun auf Kurven an, erhält man

$$\frac{d}{du} f(u) = \sum_{j=1}^{n+1} \frac{m(d_j - d_{j-1})}{t_{j+m} - t_j} N_j^{m-1}(u | T), \quad (2.6)$$

wobei  $d_0 = d_{n+1} = 0$  gilt. Bei der praktischen Implementierung einer solchen Methode zur Berechnung der ersten Ableitung einer Splinekurve ist zu beachten, dass für  $(m + 1)$ -fache Knoten im Nenner eine Division durch 0 entsteht. Auch hier können analog zur rekursiven Auswertung der B-Splines solche Terme ignoriert werden. Wendet man selbiges Schema ein weiteres Mal auf



die B-Splines an, erhält man die zweite Ableitung

$$\begin{aligned} \frac{d^2}{du^2} N_j^m(u | T) &= \frac{m}{t_{j+m} - t_j} \left( \frac{m-1}{t_{j+m+1} - t_j} N_j^{m-2}(u | T) - \frac{m-1}{t_{j+m} - t_{j+1}} N_{j+1}^{m-2}(u | T) \right) \\ &\quad - \frac{m}{t_{j+m+1} - t_{j+1}} \left( \frac{m-1}{t_{j+m} - t_{j+1}} N_{j+1}^{m-2}(u | T) - \frac{m-1}{t_{j+m+1} - t_{j+2}} N_{j+2}^{m-2}(u | T) \right). \end{aligned}$$

**Lemma 2.39.** Für eine Splinekurve  $f$  lässt sich die zweite Ableitung darstellen als

$$\frac{d^2}{du^2} f(u) = \sum_{j=1}^{n+2} \frac{m(m-1)}{t_{j+m-1} - t_j} \left( \frac{d_j - d_{j-1}}{t_{j+m} - t_j} - \frac{d_{j-1} - d_{j-2}}{t_{j+m-1} - t_{j-1}} \right) N_j^{m-2}(u | T), \quad (2.7)$$

wobei  $d_{-1} = d_0 = d_{n+1} = d_{n+2} = 0$  gilt.

Die beiden Formeln (2.6) und (2.7) ermöglichen es, alle behandelten theoretischen Aspekte parametrischer Kurven in Kapitel 2.1 in der Praxis umzusetzen. Um sowohl die Notation übersichtlicher als auch eine Implementierung effizienter zu gestalten, werden die Konzepte der Ableitungen von Splinekurven als Matrixmultiplikation dargestellt:

**Lemma 2.40.** Die Matrix  $\mathbf{D}_n$  der Form

$$\mathbf{D}_n = \begin{pmatrix} 1 & -1 & & \mathbf{0} \\ & \ddots & \ddots & \\ \mathbf{0} & & 1 & -1 \end{pmatrix} \in \mathbb{R}^{n \times n+1}.$$

heißt Differenzenmatrix. Weiter bestehe die Matrix  $\Delta_m T$

$$\Delta_m T = \begin{pmatrix} t_{m+1} - t_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & t_{n+m+1} - t_{n+1} \end{pmatrix}^{-1} \in \mathbb{R}^{(n+1) \times (n+1)}$$

aus den Reziprokwerten von jeweils zwei Knoten aus  $T$  mit Abstand  $m$ . Dann lässt sich die  $k$ -te Ableitung des Kontrollpolygons einer Splinekurve kompakt als

$$\frac{d^k}{du^k} \mathbf{dV}(U) = \left( \mathbf{d} \frac{m!}{m-k} \prod_{j=0}^{k-1} \mathbf{D}_{n+j} \Delta_{m-j} T \right) \mathbf{V}(U) =: \mathbf{dG}_k \mathbf{V}(U) \quad (2.8)$$

darstellen und es gilt für beliebige  $U = \{u_1, u_2, \dots, u_q\}$  aus dem Parameterbereich

$$\mathbf{dV}_m^{(k)}(U) = \mathbf{d}^k \mathbf{V}_{m-k}(U) = \mathbf{dG}_k \mathbf{V}_{m-k}(U). \quad (2.9)$$

Im Folgenden wird für die Ableitung einer Splinekurve die Notation

$$\frac{d}{du} (\mathbf{dV}(U)) = \mathbf{dV}'(U), \quad U = \{u_1, u_2, \dots, u_q\}$$

verwendet. Wie bereits im ersten Teil dieses Kapitels deutlich wurde, ist die Wahl der Parametrisierung (des Knotenvektors) ein sehr einflussreicher Faktor für z.B. die Längen der Ableitungen, wie nachfolgendes Beispiel veranschaulicht.

**Beispiel 2.41.** Sei eine kubische Splinekurve  $f(u)$  mit Kontrollpolygon

$$d = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 0 & 2 & -2 & 2 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

gegeben. In Abbildung 2.11 ist die Auswertung der 1. und 2. Ableitung mit unterschiedlichen Parametrisierungen dargestellt. Im ersten Fall wird der ursprüngliche Knotenvektor  $T = [0, 0, 0, 0, 1, 1, 1, 1]$  verwendet, während im zweiten Fall die Kurve mit einer regulären Re-parametrisierung  $\varphi : u \rightarrow 2u$  ausgewertet wird. Damit entsteht aus dem ursprünglichen  $T$  der neue Knotenvektor  $\tilde{T} = [0, 0, 0, 0, 2, 2, 2, 2]$ . Die Bilder der Kurven sind sowohl für  $u \in [T]$  als auch  $\tilde{u} \in [\tilde{T}]$  identisch.

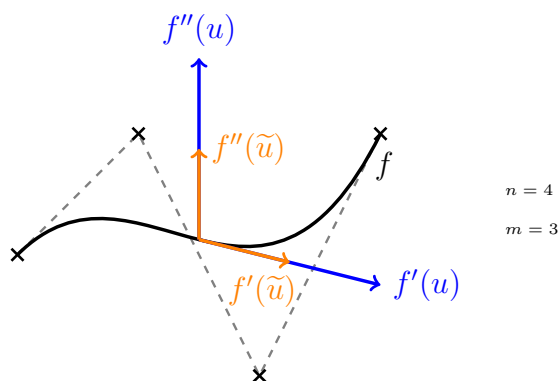


Abb. 2.11: Die Längen der Ableitung variieren je nach Wahl des Knotenvektors. Während die blauen Ableitungen dem Knotenvektor  $T = [0, 0, 0, 0, 1, 1, 1, 1]$  zugeordnet werden können, sind die orangenen Ableitungen mit  $\tilde{T} = [0, 0, 0, 0, 2, 2, 2, 2]$  entstanden.

Einen weiteren Grund für die hohe Flexibilität von Splines liefert die Möglichkeit, an beliebigen Stellen des Knotenvektors einzelne (oder mehrere) Knoten einzufügen, ohne dabei das Bild der Parametrisierung zu verändern. Dabei entsteht kein Fehler, denn anschaulich wird durch die Hinzunahme eines Knotens der Spline-Raum um eine Basis-Funktion erweitert und damit vergrößert. Mit dieser Grundidee kann man folgenden Satz formulieren und beweisen [49]:

**Satz 2.42.** Für gegebene Knotenfolgen  $\tilde{T} \subset T$  gilt  $\mathbb{S}_{m, \tilde{T}} \subset \mathbb{S}_{m, T}$ .

Es gibt zwei gängige Verfahren, um eine Knotenfolge  $T$  zu verfeinern. Die erste Variante ist der sogenannte *Boehm-Algorithmus* [6], der in der Lage ist, pro Iterationsschritt einen neuen Knoten in die alte Knotenfolge  $T$  einzufügen. Im Gegensatz dazu steht der *Oslo-Algorithmus* [16], der in einem Schritt mehrere Knoten gleichzeitig einfügt. Da durch mehrfache Anwendung des Boehm-Algorithmus eine identische Ausgabe erreicht werden kann und in diesem Grundlagenkapitel

nur das grundsätzliche Konzept des Knoteneinfügens näher betrachtet werden soll, wird darauf verzichtet, beide Algorithmen vorzustellen. Seien der Grad  $m$ , das Kontrollpolygon  $\mathbf{d}$  sowie ein Knotenvektor  $T$  einer Spline-Kurve gegeben. Um nun einen Knoten  $t_1 \leq t \leq t_{m+n+1}$  in  $T$  einzufügen, müssen folgende Schritte sukzessive abgearbeitet werden.

— Algorithmus (Boehm) —

Eingabe :

Knotenfolge  $T_{m,n}$

Neuer Knoten  $t$  mit  $t_j \leq t \leq t_{j+1}$

Prozedur :

*For*  $k = 1, \dots, j - m$

$$\tilde{d}_k = d_k$$

*End*

*For*  $k = j - m + 1, \dots, j$

$$\tilde{d}_k = \frac{t_{k+m} - t}{t_{k+m} - t_k} \cdot d_{k-1} + \left(1 - \frac{t_{k+m} - t}{t_{k+m} - t_k}\right) \cdot d_k$$

*End*

*For*  $k = j + 1, \dots, n + 1$

$$\tilde{d}_k = d_{k-1}$$

*End*

Ausgabe :

$$S_m \cdot \tilde{\mathbf{d}}(\cdot | \tilde{T}) = S_m \cdot \mathbf{d}(\cdot | T)$$

Algorithmus 2.2: Boehm-Algorithmus

Analog zu den Ableitungen ist es auch hier möglich, mithilfe der linearen Algebra das Verfahren als Multiplikation von Matrizen darzustellen:

**Definition 2.43.** Sei ein Knotenvektor  $T$  sowie ein Kontrollpolygon  $\mathbf{d}$  einer B-Spline Kurve gegeben. Weiter bezeichne  $t$  den einzufügenden Knoten. Dann heißt die Matrix  $\mathbf{A}(t)$  der Form

$$\mathbf{A}(t) = \begin{pmatrix} 1 & & & & & & \mathbf{0} & & & \\ & \ddots & & & & & & & & \\ & & 1 & & \alpha_{j-m+1} & & & & & \\ & & & (1 - \alpha_{j-m+1}) & & \ddots & & & & \\ & & & & & \ddots & & & & \\ & & & & & & \alpha_j & & & \\ \mathbf{0} & & & & & & (1 - \alpha_j) & 1 & & \\ & & & & & & & & \ddots & \\ & & & & & & & & & 1 \end{pmatrix} \in \mathbb{R}^{n \times n+1}$$

Knoteneinfügematrix *und es gilt*

$$\tilde{\mathbf{d}} = \mathbf{dA}(t),$$

wobei  $\alpha_k = \frac{t_{k+m}-t}{t_{k+m}-t_k}, k = j - m + 1, \dots, j$  gilt.

In Abbildung 2.12 ist die Splinekurve aus Beispiel 2.11 zu sehen. Während auf der linken Seite die Originalkurve mit Knotenvektor  $T = [0, 0, 0, 0, 1, 1, 1, 1]$  abgebildet ist, wurde mit der Knoteneinfügematrix auf der rechten Seite der Knoten  $t = 0.5$  in  $T$  eingefügt. Die Bilder beider Kurven sind identisch.

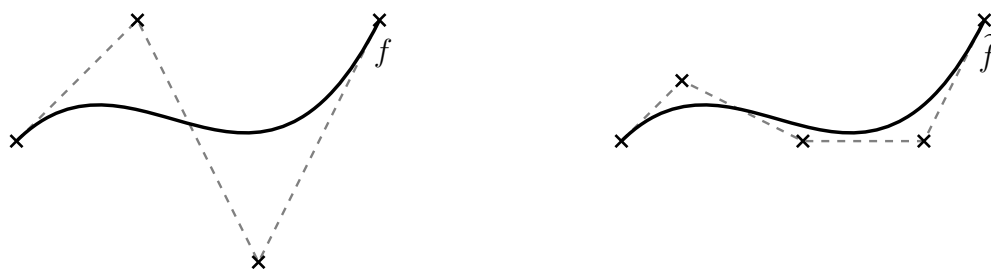


Abb. 2.12: Splinekurve vor (links) und nach (rechts) dem Einfügen eines neuen Knotens

Genauso wie man eine Knotenfolge verfeinern kann, ist es auch möglich, Knoten aus einem gegebenen Knotenvektor zu entfernen. Im Allgemeinen ist diese Anwendung allerdings nicht fehlerfrei durchführbar, da man den Spline-Raum verkleinert. Um das neue Kontrollpolygon  $\tilde{\mathbf{d}}$  zu berechnen, beginnt man mit dem bereits existierenden  $\tilde{\mathbf{d}}$  und minimiert

$$\begin{aligned} \left\| \mathbf{dA}(t) - \tilde{\mathbf{d}} \right\|_2^2 &= \mathbf{dA}(t)\mathbf{A}(t)^T \mathbf{d} - 2\tilde{\mathbf{d}}\mathbf{A}(t)^T \mathbf{d} + \tilde{\mathbf{d}}\tilde{\mathbf{d}}^T \\ \mathbf{dA}(t)\mathbf{A}(t)^T &= \tilde{\mathbf{d}}\mathbf{A}(t)^T, \end{aligned}$$

wobei  $t$  der zu entfernende Knoten ist. Weiterführende Informationen und Strategien zum Knotenentfernen finden sich z. B. in [27, 36, 37].

## 2.2.2 Interpolation und Approximation mit Splinekurven

Der Grund, warum CAD-Systeme standardmäßig auf B-Splines zur Darstellung von Kurven und Freiformflächen zurückgreifen, zeigt sich vor allem bei der Interpolation und Approximation von Datenpunkten. Sei  $\mathbf{P} \in \mathbb{R}^{3 \times q}$  eine Menge von Punkten  $\mathbf{P}_{:,i}$  in einer Matrix angeordnet. Damit eine Splinekurve  $f$  die Punkte  $\mathbf{P}$  an Stützstellen  $U = \{u_1, u_2, \dots, u_q\}$  interpoliert, muss

$$\mathbf{dV}(U) = \mathbf{P}, \tag{2.10}$$

gelten, wobei  $\mathbf{d}$  das Kontrollpolygon und  $\mathbf{V}(U)$  die Vandermonde-Matrix von  $f$  ist. An dieser Stelle wird ebenfalls die Notation  $f.\mathbf{V}$  verwendet, falls aus dem Kontext nicht ersichtlich ist,

zu welcher Kurve die Vandermonde-Matrix gehört. Hier stellt sich zwangsläufig die Frage nach Bedingungen für die Invertierbarkeit von  $\mathbf{V}(U)$  und der damit verbundenen Eindeutigkeit des Interpolationsproblems. Dieser Zusammenhang wurde in [50] formuliert und bewiesen und ist als Satz von Schoenberg und Whitney bekannt.

**Satz 2.44.** *Die Gleichung (2.10) hat genau dann eine eindeutige Lösung in  $\mathbb{S}_{m,T}$ , wenn für die  $q$  Interpolationsstellen  $u_i$*

$$N_i^m(u_i) > 0, \quad 1 \leq i \leq q, \quad (2.11)$$

*gilt. Für einfache Knoten ist diese Bedingung äquivalent zu*

$$t_j < u_j < t_{j+m+1}, \quad j = 1, \dots, m.$$

Falls eine innerer Knoten die Vielfachheit  $(m + 1)$  besitzt, ist die Kurve an dieser Stelle nicht stetig differenzierbar und damit ist dieser Fall hier nicht von Bedeutung. Aus diesem Satz ergeben sich kanonische Interpolationsstellen  $u_i$ , die (2.11) erfüllen und *Greville-Abszissen* heißen:

**Definition 2.45.** *Die zu einem Knotenvektor  $T$  zugehörigen Werte  $\tilde{t}_j$  mit der Eigenschaft*

$$\tilde{t}_j = \frac{1}{m} \sum_{k=1}^m t_{j+k}, \quad 1 \leq j \leq n,$$

*heißen Greville-Abszissen.*

Einer der Vorteile von Splinekurven gegenüber Polynomfunktionen liegt darin, dass es möglich ist, bei gleichbleibendem Grad (z. B.  $m = 3$ ) eine beliebige Menge an Punkten zu interpolieren. Mit steigender Anzahl an Punkten wächst aber auch die Zahl der Basisfunktionen und damit die Dimension des zu lösenden linearen Gleichungssystems. Obwohl das Oszillationsverhalten von Splinekurven verglichen mit konventionellen Interpolationsschemata vergleichsweise gutmütig ist, können hier insbesondere am Rand Effekte wie das „Überschwingen“ einer Kurve auftreten. Aus diesem Grund wird in der Praxis oftmals ein Approximationsansatz vorgezogen, der neben den zu interpolierenden Punkten zusätzlich die globale Glätte der Kurve in Form von „Fairness“-Funktionalen miteinbezieht (siehe [40]). Ein häufig verwendeter Ansatz im Kontext der Approximation von Datenpunkten ist z.B. das folgende quadratische Minimierungsproblem mit Gleichheitsrandbedingungen:

Funktional	Nebenbedingung
$\min_f \int_{[T]} \ f''(u)\ _2^2 \, du,$	$f(u_i) = \mathbf{P}_{:,i}, \quad 1 \leq i \leq q$

An dieser Stelle gibt es zwei Dinge zu beachten. Zum einen muss das Gleichungssystem lösbar sein. Zum anderen handelt es sich im Funktional um die parametrische Ableitung  $f''$  und nicht

die Ableitung bezüglich der Bogenlänge  $f$ . Funktionale, die Krümmungen beinhalten, sind nicht-linear und aus Gründen der Konvergenz häufig nicht wünschenswert.

Obwohl  $f''$  nur indirekten Einfluss auf die Krümmung besitzt (vgl. Satz 2.11), ist diese Ableitung aus Gründen der Effizienz eine gute Wahl, da im Nenner keinerlei Normen auftreten und die Ableitung des Problems linear ist. Für eine Splinekurve ergibt sich damit folgendes Problem

$$\min_{\mathbf{d}} \int_{[T]} \mathbf{dV}''(u) \mathbf{V}''(u)^T \mathbf{d}^T \, du = \mathbf{dAd}^T, \quad \mathbf{dV}(U) = \mathbf{P},$$

dass man durch Lösen des LGS

$$\begin{pmatrix} 2\mathbf{A} & \mathbf{V}(U) \\ \mathbf{V}(U)^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{d}^T \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{P}^T \end{pmatrix}$$

erhält. In obiger Darstellung steht  $\lambda$  für die Lagrangemultiplikatoren [41] und  $\mathbf{V}''(u)$  für die 2. Ableitung der Vandermonde-Matrix nach dem Parameter  $u$ .

Um etwas mehr Freiheit bei der Wahl des Spline-Raums  $\mathbb{S}_{m,T}$  zu erhalten und um den Kompromiss aus Datentreue und zu minimierendem Funktional mithilfe eines Parameters stärker beeinflussen zu können, bietet sich die Verwendung eines sogenannten *Smoothing Splines* an:

**Definition 2.46.** *Die Lösung des Optimierungsproblems*

$$\mathcal{F} = \min_f \left( \sum_{u_i} w_i \|f(u_i) - \mathbf{P}_i\|_2^2 + \lambda \int_{[T]} \|f^{(k)}(u)\|_2^2 \, du, \quad w_i, \lambda > 0, \quad k = \mathbb{N} \right)$$

heißt gewichteter Smoothing Spline.

Gesucht ist das Kontrollpolygon einer Splinekurve für einen gegebenen Spline-Raum  $\mathbb{S}_{m,T}$ . Dazu kann man die Gleichung aus Definition 2.46 wie folgt umformulieren

$$\mathcal{F} = (\mathbf{dV}(U) - \mathbf{P}) \mathbf{W} (\mathbf{dV}(U) - \mathbf{P})^T + \lambda \mathbf{d} \int_{[T]} \mathbf{V}''(u) \mathbf{V}''(u)^T \, du \mathbf{d}^T,$$

wobei  $\mathbf{W}$  eine Matrix ist, deren Diagonaleinträge genau die Gewichte  $w_i$  sind. Damit kann der zu minimierende Term  $\mathcal{F}$  folgendermaßen in quadratischer Form dargestellt werden:

$$\mathcal{F} = \mathbf{d} \left( \mathbf{V}(U) \mathbf{W} \mathbf{V}(U)^T + \lambda \int_{[T]} \mathbf{V}''(u) \mathbf{V}''(u)^T \, du \right) \mathbf{d}^T - 2 \mathbf{dV}(u_i) \mathbf{W} \mathbf{P}^T + \mathbf{P} \mathbf{W} \mathbf{P}^T.$$

Durch Nullsetzen der ersten Ableitung erhält man die positiv semidefinite, quadratische Form

$$\left( \mathbf{V}(U)\mathbf{W}\mathbf{V}(U)^T + \lambda \int_{[T]} \mathbf{V}''(u)\mathbf{V}''(u)^T \, du \right) \mathbf{d}^T = \mathbf{V}(U)\mathbf{W}\mathbf{P}^T \quad (2.12)$$

und kann damit das Minimum berechnen. Folgendes Beispiel veranschaulicht den Kompromiss zwischen Oszillationen und Datentreue anhand unterschiedlich gewählter Glättungsparameter  $\lambda$ .

**Beispiel 2.47.** Seien Punkte  $\mathbf{P} \in \mathbb{R}^{3 \times 6}$  sowie ein Knotenvektor  $T = [0, 0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1]$  gegeben. Abbildung 2.13 zeigt die resultierenden Splinekurven als Lösung des linearen Gleichungssystems (2.12). Für die Wahl  $\lambda = 0$  erhält man das bekannte Interpolationsschema. Je höher der Wert  $\lambda$  gesetzt wird, desto glatter ist die Kurve auf Kosten der Datentreue.

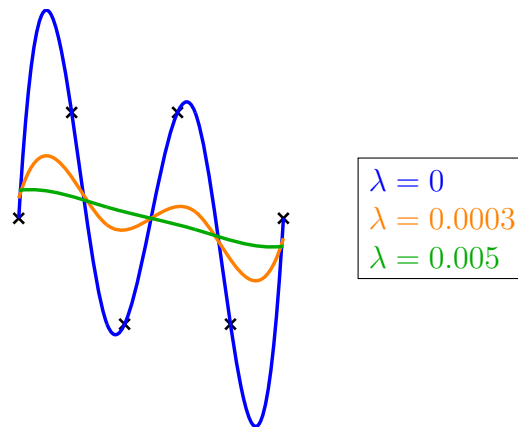


Abb. 2.13: Verschiedene Smoothing-Splines mit entsprechender Wahl von  $\lambda$

Im nächsten Teil des Grundlagenkapitels werden diese Themenfelder auf Tensorprodukt-Flächen erweitert und Vor- bzw. Nachteile diskutiert.

## 2.2.3 Tensorprodukt-Flächen

Eine geometrisch sehr anschauliche Einführung in die Idee hinter Tensorprodukt-Flächen liefert folgende Überlegung:

**Beispiel 2.48.** Sei eine parametrische Fläche im  $\mathbb{R}^3$  wie in Abbildung 2.14 gegeben. Betrachtet man nun die zwei Randkurven  $f_1$  und  $f_n$ , so „wandert“ die erste Splinekurve entlang vieler Kurven auf der Fläche und verändert sich dabei stetig. Aus der Ausgangsparabel  $f_1$  in diesem Beispiel wird die gerade Strecke  $f_n$ .

Mit dieser Überlegung ist es möglich, univariate Splinekurven auf Tensorprodukt-Flächen zu

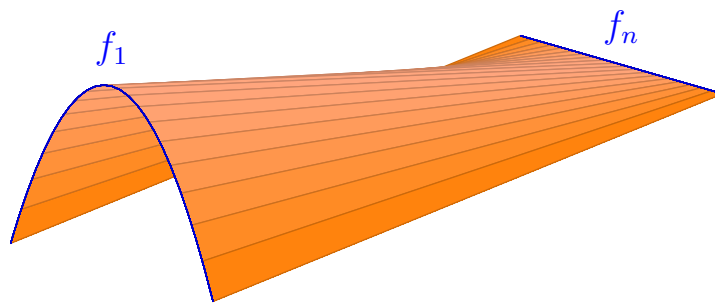


Abb. 2.14: Tensorprodukt-Fläche:  $f_1$  verläuft entlang von Kurven und verändert sich stetig zu  $f_n$

erweitern. Dazu beginnt man mit einer einparametrischen Familie von Kurven

$$f_v(u) = \sum_{j=1}^n d_j(v) N_j^{m_u}(u | T_u),$$

wobei  $d_j(i)$  selbst wieder eine Splinekurve der Form

$$d_j(v) = \sum_{k=1}^{\tilde{n}} d_{jk} N_k^{m_v}(v | T_v)$$

ist. Während die Splinekurven  $f_v$  alle demselben Knotenvektor  $T_u$  zugeordnet sind, ist die entsprechende Knotenfolge der Kurven  $d_j$  stets  $T_v$ . Dadurch entsteht eine Tensorprodukt-Gitterstruktur, die in Abbildung 2.15 exemplarisch dargestellt ist.

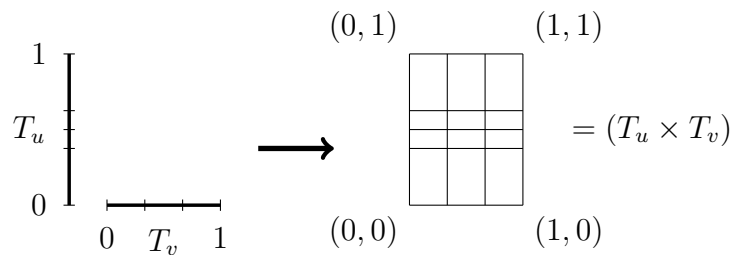


Abb. 2.15: Durch das Kreuzprodukt zweier univariater Knotenfolgen  $T_u$  und  $T_v$  entsteht eine rechteckige Gitterstruktur.

Anhand dieser Struktur wird sowohl der größte Vorteil als auch der damit zusammenhängende Nachteil deutlich. Die Gitterstruktur ermöglicht es, eine Fläche durch die univariate Spline-Theorie zu beschreiben. Die bereits vorgestellten Verfahren und Eigenschaften können ohne weitere Anpassung auf Tensorprodukt-Flächen angewandt werden. Zur Interpolation oder Approximation beliebig komplexer Punktwolken eignen sie sich allerdings nur bedingt, da durch das Tensorprodukt der Knotenfolgen implizit eine Struktur auf den zu interpolierenden Datensatz gelegt wird. In diesem Zusammenhang wurden spezielle Splines, sogenannte THB-Splines [15],



entwickelt, die lokale Verfeinerungen auf dem Knotengitter vornehmen. Eine praxisnahe Anwendung im CAD-Umfeld kann z.B. in [2] nachgelesen werden. Bevor man den Begriff einer Tensorprodukt-Fläche definieren kann, ist es sinnvoll, kurz eine passende Notation für das Kontrollpolygon (das im Gegensatz zum Kurvenfall nun ein Gitter ist) einzuführen. Flächen werden in den weiteren Kapiteln im Gegensatz zu den Kurven  $f, g, h$  mit den Großbuchstaben  $F, G, H$  notiert.

**Bemerkung 2.49.** *Im bivariaten Fall handelt es sich bei einem Kontrollpolygon  $\mathbf{d}$  um einen Tensor der Form*

$$\mathbf{d} \in \mathbb{R}^{n \times \tilde{n} \times 3},$$

wobei  $n$  die Anzahl an Kontrollpunkten in  $u$ -Richtung ( $\tilde{n}$  entsprechend in  $v$ -Richtung) darstellt. Das heißt, man erhält für jede festgelegte Position  $(j, k)$  im Gitter einen dreidimensionalen Vektor. Um eine übersichtliche Notation für Tensorprodukt-Flächen mit Matrixprodukten zu ermöglichen, wird folgende an MATLAB orientierte Notation für Tensoren

$$\mathbf{d}_{(a,b,c)} = \begin{cases} a = \text{Reihen in } (u) - \text{Richtung} \\ b = \text{Spalten in } (v) - \text{Richtung} \\ c = \text{Komponenten des Bildraums } x, y, z \end{cases},$$

eingeführt. Fixiert man eine der Dimensionen  $a, b, c$  wird festgelegt, dass sich der Tensor um die entsprechende Dimension reduziert und zu einer Matrix bzw. einem Vektor wird. Für den ersten Kontrollpunkt einer Tensorprodukt-Fläche gilt also z.B.

$$\mathbf{d}_{(1,1,:)} \in \mathbb{R}^3.$$

**Definition 2.50.** *Eine Tensorprodukt-Fläche ist definiert durch zwei Knotenfolgen  $T_u$  und  $T_v$  sowie dem zweidimensionalen Gitter von Kontrollpunkten  $\mathbf{d}$  und besitzt die Form*

$$F(u, v) = \sum_{j=1}^n \sum_{k=1}^{\tilde{n}} \mathbf{d}_{(j,k,:)} N_k^{m_v}(v | T_v) N_j^{m_u}(u | T_u).$$

**Definition 2.51.** *Sei eine Matrix  $\mathbf{A} \in \mathbb{R}^{p \times q}$  gegeben und bezeichne  $a_{(:,i)}$  die  $i$ -te Spalte von  $\mathbf{A}$ . Der Operator  $\text{vec}(\cdot)$  ist definiert durch*

$$\text{vec}(\mathbf{A}) = \begin{pmatrix} a_{(:,1)} \\ \vdots \\ a_{(:,q)} \end{pmatrix}.$$

**Definition 2.52.** *Für zwei Matrizen  $\mathbf{A} \in \mathbb{R}^{p \times q}$  und  $\mathbf{B} \in \mathbb{R}^{r \times s}$  ist das Kronecker-Produkt  $\otimes$*

definiert durch

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{(1,1)}\mathbf{B} & \dots & a_{(1,q)}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{(p,1)}\mathbf{B} & \dots & a_{(p,q)}\mathbf{B} \end{pmatrix} \in \mathbb{R}^{pr \times qs}.$$

Wie bereits im Kurvenfall ist eine kompakte Darstellung zur Auswertung von  $F$  an Parameterstellen  $U = \{u_1, u_2, \dots, u_q\}$ ,  $V = \{v_1, v_2, \dots, v_r\}$  sehr hilfreich. Da  $\mathbf{d}$  allerdings ein Tensor ist, muss die dritte Komponente  $(x, y, z)$  abgekoppelt werden, um alle Berechnungen als Matrixmultiplikation formulieren zu können:

$$\text{vec}(F_j(U, V)) = \text{vec}(\mathbf{V}_u(U)^T \mathbf{d}_{(:, :, j)}^T \mathbf{V}_v(V)), \quad j = 1, 2, 3.$$

An dieser Stelle ist die Struktur des Kronecker-Produkts der Vandermonde-Matrizen nicht ohne Weiteres ersichtlich. Durch die folgende Umformung, deren Beweis in der gängigen Standardliteratur (z.B. [38]) zu finden ist, wird der Zusammenhang allerdings deutlich.

**Satz 2.53.** Für Matrizen  $\mathbf{A} \in \mathbb{R}^{p \times q}$ ,  $\mathbf{B} \in \mathbb{R}^{r \times s}$  und  $\mathbf{X} \in \mathbb{R}^{q \times r}$  gilt folgender Zusammenhang

$$\text{vec}(\mathbf{AXB}) = (\mathbf{B}^T \otimes \mathbf{A}) \text{vec}(\mathbf{X}),$$

wobei  $\text{vec}(\cdot)$  die Vektorisierung einer Matrix in einen Zeilenvektor darstellt.

Betrachtet man nun die Auswertung einer Tensorprodukt-Fläche an den Parameterstellen  $(U, V)$  in einer einzelnen Komponente (z.B.  $x$ ) ergibt sich:

$$\begin{aligned} \text{vec}(F_x(U, V))^T &= \text{vec}(\mathbf{d}_{(:, :, 1)}^T)^T (\mathbf{V}_u(U) \otimes \mathbf{V}_v(V)) \\ &= \text{vec}(\mathbf{V}_u(U)^T \mathbf{d}_{(:, :, 1)}^T \mathbf{V}_v(V)). \end{aligned}$$

Mit diesem Verfahren ist es möglich, eine Tensorprodukt-Fläche auszuwerten, ohne ein ggf. großes und speicherintensives Kronecker-Produkt der Form  $(\mathbf{V}_u(U) \otimes \mathbf{V}_v(V))$  ausrechnen zu müssen. Wie bereits zu Beginn dieses Abschnittes gesehen, können alle Eigenschaften der Splincurven übernommen werden. Die Jacobi-Matrix  $\mathbf{J}_F(u, v)$  sowie die Hesse-Matrix  $\mathbf{H}_F(u, v)$  ergeben sich damit analog zu Gleichung (2.9):

**Lemma 2.54.** Die  $j$ -ten Komponenten der Ableitungen erster und zweiter Ordnung einer Tensorprodukt-Fläche an Parameterstellen  $U, V$  sind gegeben durch

$$\begin{aligned} \text{vec}(F_{u,j}(U, V))^T &= \text{vec}(\mathbf{d}_{(:, :, j)}^T)^T (\mathbf{V}'_u(U) \otimes \mathbf{V}_v(V)), \\ \text{vec}(F_{v,j}(U, V))^T &= \text{vec}(\mathbf{d}_{(:, :, j)}^T)^T (\mathbf{V}_u(U) \otimes \mathbf{V}'_v(V)), \\ \text{vec}(F_{uu,j}(U, V))^T &= \text{vec}(\mathbf{d}_{(:, :, j)}^T)^T (\mathbf{V}''_u(U) \otimes \mathbf{V}_v(V)), \\ &\vdots \end{aligned}$$

Das Einfügen und Entfernen von Knoten ist aufgrund der Tensorstruktur beider Knotenfolgen

nur reihen- bzw. spaltenweise möglich, allerdings mit den bereits bekannten Algorithmen für Splinekurven, wie in Abbildung 2.16 zu sehen:

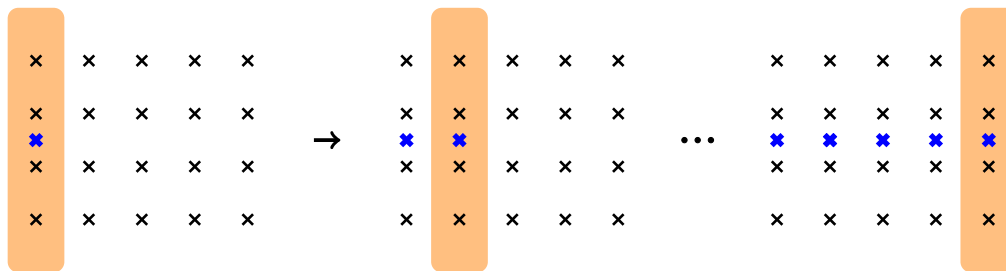
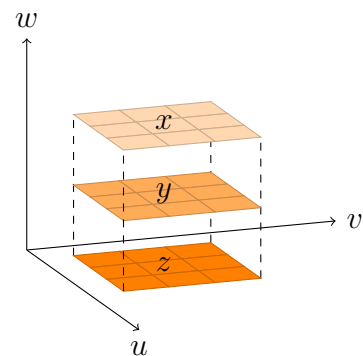


Abb. 2.16: Um eine neue Knotenreihe einzufügen/zu entfernen, muss spaltenweise über das Tensorprodukt iteriert werden und in jedem univariaten Spline der entsprechende Knoten eingefügt/entfernt werden.

Bevor die verschiedenen Möglichkeiten der Interpolation und Approximation von Punktwolken mit Tensorprodukt-Flächen erörtert werden, wird an dieser Stelle kurz folgende Notation eingeführt.

**Bemerkung 2.55.** Für dreidimensionale Punktwolken, die in Spalten und Reihen strukturiert werden können, wird vereinbart, dass die drei Dimensionen jeweils mit  $x, y$  und  $z$  bezeichnet werden. Die Schreibweise für solche Tensoren und Tensorprodukt-Flächen orientiert sich aufgrund der Praxisnähe an MATLAB:

$F$	Splinekurve
$F.d$	Kontrollpolygon (Gitter) der Tensorprodukt-Fläche $F$
$F.T_u$	Knotenvektor in $u$ -Richtung der Fläche $F$
$F.T_v$	Knotenvektor in $v$ -Richtung der Fläche $F$
$F.m_u$	Grad der Fläche $F$ in $u$
$F.m_v$	Grad der Fläche $F$ in $v$
$F.d_x$	$x$ -Komponente des Kontrollpolygons $F.d(:, :, 1)$
$\vdots$	$\vdots$



## 2.2.4 Interpolation und Approximation mit Tensorprodukt-Flächen

Wie bereits erwähnt, sind Tensorprodukt-Flächen ein probates Mittel zur Interpolation von Punktwolken, die reihen- und spaltenweise angeordnet werden können. Seien also Punkte  $\mathbf{P} \in \mathbb{R}^{p \times q \times 3}$  gegeben. Weiter seien die Knotenfolgen  $T_u$  und  $T_v$  sowie ihre Greville Abszissen  $U = \{u_i, i = 1, \dots, q\}$  und  $V = \{v_j, j = 1, \dots, r\}$ , die ja den Schoenberg-Whitney Bedingungen 2.44 genügen, als Stützstellen gegeben. Um einen geeigneten Interpolanten zu finden,

müssen die drei Gleichungssysteme der Form

$$\begin{aligned}\text{vec}^T(F.d_x)(\mathbf{V}_u(U) \otimes \mathbf{V}_v(V)) &= \text{vec}^T(\mathbf{P}_x), \\ \text{vec}^T(F.d_y)(\mathbf{V}_u(U) \otimes \mathbf{V}_v(V)) &= \text{vec}^T(\mathbf{P}_y), \\ \text{vec}^T(F.d_z)(\mathbf{V}_u(U) \otimes \mathbf{V}_v(V)) &= \text{vec}^T(\mathbf{P}_z)\end{aligned}$$

nach  $F.d$  gelöst werden. Dabei fällt auf, dass das Kroneckerprodukt nur einmal berechnet werden muss und anschließend dann für alle Komponenten verwendet werden kann. Auch hier ist es möglich, das Kroneckerprodukt als Matrixmultiplikation zu schreiben, um nicht die Inverse von  $(\mathbf{V}_u(U) \otimes \mathbf{V}_v(V))$  berechnen zu müssen. Es ergibt sich das folgende Gleichungssystem:

$$\begin{aligned}\text{vec}^T(\mathbf{V}_v(V)F.d_x\mathbf{V}_u(U)) &= \text{vec}^T(\mathbf{P}_x), \\ \text{vec}^T(\mathbf{V}_v(V)F.d_y\mathbf{V}_u(U)) &= \text{vec}^T(\mathbf{P}_y), \\ \text{vec}^T(\mathbf{V}_v(V)F.d_z\mathbf{V}_u(U)) &= \text{vec}^T(\mathbf{P}_z),\end{aligned}$$

das ebenfalls komponentenunabhängig gelöst werden kann. Wie bereits in Definition 2.50 von Tensorprodukt-Flächen gesehen, kann man sich auch bei der Interpolation von Punktwolken die univariaten Eigenschaften zunutze machen, indem man, ähnlich dem Knoteneinfügen, einzelne Splinekurven verwendet, um die Reihen (bzw. Spalten) zu interpolieren und anschließend über die Spalten (bzw. Reihen) iteriert und selbiges Verfahren anwendet. Der nachfolgende Pseudo-Code skizziert dieses Verfahren:

— **Algorithmus (Interpolation einer Punktwolke)** —

Eingabe:

- Punkte  $\mathbf{P}$
- Knotenfolgen  $T_u, T_v$
- Greville-Abszissen  $u_i, v_j$
- Grad in  $u$ -Richtung und  $v$ -Richtung  $m_u, m_v$

Prozedur:

```

foreach 'Row in P'
    Löse Interpolationsproblem:  $\tilde{\mathbf{d}}(j, :) \mathbf{V}_u(U) = \mathbf{P}(:, j, :)^T$ 
end
foreach 'Column in P'
    Löse Interpolationsproblem:  $\mathbf{d}(:, i)^T \mathbf{V}_v(V) = \tilde{\mathbf{d}}(:, i)^T$ 
end

```

Algorithmus 2.3: Interpolation einer Punktwolke

Die Hinzunahme eines Minimierungsfunktionals zur Glättung des Interpolanten gestaltet sich bei Flächen etwas komplizierter als bei Kurven, da man von Haus aus zwei Ableitungen erster Ordnung und vier Ableitungen zweiter Ordnung besitzt. Die erste wichtige Erkenntnis ist, dass bei reiner Interpolation mit Tensorprodukt-Flächen die Komponenten  $x, y$  und  $z$  unabhängig

voneinander betrachtet werden können. Nimmt man nun z.B. ein Funktional der Form

$$\mathcal{F} = \int_{[T_u]} \int_{[T_v]} \|F_{uu}(u, v)\|_2^2 + 2 \|F_{uv}(u, v)\|_2^2 + \|F_{vv}(u, v)\|_2^2 \, \mathbf{d}v \, \mathbf{d}u$$

zur reinen Interpolation hinzu, so verliert man die Unabhängigkeit der einzelnen Komponenten und koppelt diese aneinander. Um beide Bedingungen in einem Gleichungssystem lösen zu können, muss man im ersten Schritt die Datentreue umformulieren. Aus

$$\begin{aligned} \text{vec}^T(F.d_x) (\mathbf{V}_u(U) \otimes \mathbf{V}_v(V)) &= \text{vec}^T(\mathbf{P}_x) \\ \text{vec}^T(F.d_y) (\mathbf{V}_u(U) \otimes \mathbf{V}_v(V)) &= \text{vec}^T(\mathbf{P}_y) \\ \text{vec}^T(F.d_z) (\mathbf{V}_u(U) \otimes \mathbf{V}_v(V)) &= \text{vec}^T(\mathbf{P}_z) \end{aligned}$$

wird dann

$$\underbrace{\begin{pmatrix} \text{vec}(F.d_x) \\ \text{vec}(F.d_y) \\ \text{vec}(F.d_z) \end{pmatrix}^T}_{\text{vec}^T(F.d)} (\mathbf{V}_u(U) \otimes \mathbf{V}_v(V) \otimes \mathbf{I}_3) = \underbrace{\begin{pmatrix} \text{vec}(\mathbf{P}_x) \\ \text{vec}(\mathbf{P}_y) \\ \text{vec}(\mathbf{P}_z) \end{pmatrix}^T}_{\text{vec}^T(\mathbf{P})}.$$

Das Gleichungssystem vergrößert sich in jeder Dimension um den Faktor 3. Im nächsten Schritt muss das Funktional entsprechend angepasst werden:

$$\begin{aligned} \mathcal{F} &= \int_{[T_u]} \int_{[T_v]} \|F_{uu}(u, v)\|_2^2 + 2 \|F_{uv}(u, v)\|_2^2 + \|F_{vv}(u, v)\|_2^2 \, \mathbf{d}v \, \mathbf{d}u \\ &= \text{vec}(F.d)^T \mathbf{A} \text{vec}(F.d) + 2 \text{vec}(F.d)^T \mathbf{B} \text{vec}(F.d) + \text{vec}(F.d)^T \mathbf{C} \text{vec}(F.d), \end{aligned}$$

wobei

$$\begin{aligned} \mathbf{A} &= \int_{[T_u]} \int_{[T_v]} (\mathbf{V}_u''(u, v) \otimes \mathbf{V}_v(u, v)) (\mathbf{V}_u''(u, v) \otimes \mathbf{V}_v(u, v))^T \otimes \mathbf{I}_3 \, \mathbf{d}v \, \mathbf{d}u \\ \mathbf{B} &= \int_{[T_u]} \int_{[T_v]} (\mathbf{V}_u'(u, v) \otimes \mathbf{V}_v'(u, v)) (\mathbf{V}_u'(u, v) \otimes \mathbf{V}_v'(u, v))^T \otimes \mathbf{I}_3 \, \mathbf{d}v \, \mathbf{d}u \\ \mathbf{C} &= \int_{[T_u]} \int_{[T_v]} (\mathbf{V}_u(u, v) \otimes \mathbf{V}_v''(u, v)) (\mathbf{V}_u(u, v) \otimes \mathbf{V}_v''(u, v))^T \otimes \mathbf{I}_3 \, \mathbf{d}v \, \mathbf{d}u \end{aligned}$$

gilt. Die Ableitung dieser quadratischen Form ist dann durch

$$\frac{d}{d(F.d)} \mathcal{F} = 2 \text{vec}(F.d) \mathbf{A} + 4 \text{vec}(F.d) \mathbf{B} + 2 \text{vec}(F.d) \mathbf{C} = 2 \text{vec}(F.d) (\mathbf{A} + 2\mathbf{B} + \mathbf{C})$$

gegeben. Das zu lösende Gleichungssystem ergibt sich damit zu

$$\begin{pmatrix} \text{vec}(F.d) \\ \lambda \end{pmatrix}^T \begin{pmatrix} 2(\mathbf{A} + 2\mathbf{B} + \mathbf{C}) & \mathbf{V}_u(U) \otimes \mathbf{V}_v(V) \otimes \mathbf{I}_3 \\ (\mathbf{V}_u(U) \otimes \mathbf{V}_v(V) \otimes \mathbf{I}_3)^T & 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \text{vec}(\mathbf{P}) \end{pmatrix}^T.$$

Für spezielle Anwendungen mit großen Punktwolken kann es nützlich sein, auf den univariaten Fall zurückzugreifen. In diesem Fall ist es zwar nicht möglich, alle Bestandteile des Funktionals  $\mathcal{F}$  gleichzeitig zu minimieren, allerdings könnte man für die Splinekurven in  $u$ -Richtung z.B. einen Smoothing Spline verwenden. Das zu lösende Gleichungssystem ist deutlich kleiner und damit effizienter berechenbar. Die Tatsache, dass es mit dem univariaten Ansatz nicht möglich ist, sowohl  $\mathbf{A}$ ,  $\mathbf{B}$  und  $\mathbf{C}$  gleichzeitig zu minimieren, ergibt sich bereits aus der Tatsache, dass die Summe von Tensorprodukten  $(\mathbf{A} + 2\mathbf{B} + \mathbf{C})$  selbst nicht als Tensorprodukt formuliert werden kann. Weitere Informationen zum effizienten Umgang mit Tensorproduktflächen und deren Summen sind in [9, 35] nachzulesen.

Ein wichtiger Bestandteil dieser Arbeit ist neben den theoretischen Resultaten auch die praktische Umsetzbarkeit aller Algorithmen und Verfahren. Aus diesem Grund wird im folgenden Abschnitt näher auf spezifische Aspekte der Informatik, die im Zusammenhang mit der Implementierung stehen, eingegangen.

## 2.3 Schnittstelle NXOpen

Um einen Überblick über die programmiertechnischen Komponenten dieser Arbeit zu erhalten, wird im ersten Schritt auf die Struktur der im Zuge dieser Arbeit entwickelten Bibliotheken und die verwendeten Softwarepakete eingegangen. Anschließend werden einige Funktionalitäten der Schnittstelle NXOpen, die es dem Programmierer ermöglicht, in das laufende CAD-System aktiv einzugreifen, vorgestellt.

### 2.3.1 Grundlegender Aufbau

Das Softwaresystem NX ist sehr umfangreich und besteht aus drei großen Hauptkomponenten, die oftmals als  $CAx$  zusammengefasst werden:

- **CAD: Computer Aided Design**  
Das CAD-Modul ermöglicht dem Konstrukteur, komplexe Geometrien, Bauteile und komplette Bauteilgruppen anzufertigen. Mögliche Anwendungsfälle reichen von einzelnen Zahnflankengeometrien bis hin zu Automodellen, die aus einer Vielzahl einzelner Bauteilelemente zusammengesetzt sind.
- **CAM: Computer Aided Manufacturing**  
Mithilfe des CAM-Moduls kann aus einem fertigen CAD-Modell eine NC-Programmierung erstellt werden, die z.B. an Werkzeugmaschinen (meist Dreh- und Fräsmaschinen) Ver-

wendung findet. Dieses Modul unterstützt darüber hinaus die Bearbeitung mit komplexen Multi-Achs-Systemen.

- **CAE: Computer Aided Engineering**

Das CAE-Modul ermöglicht konstruktionsbegleitende Simulationen in den Bereichen Festigkeit, Wärmestruktur, Fluid und Akustik. Auf Basis von Finite Elemente Methoden (FEM) können physikalische Modelle simuliert und Vorhersagen bezüglich Steifigkeit und anderer Kenngrößen getroffen werden.

In der industriellen Fertigung tauchen oftmals wiederkehrende Anforderungen auf, die im Optimalfall automatisiert werden können. So ist es beispielsweise wünschenswert, Aussparungen für Schrauben einer in einem CAD-System konstruierten Schrankwand abhängig von der Länge und Breite zu platzieren und bei dynamischen Änderungen durch den Benutzer automatisiert anzupassen. Abhängig von der Komplexität einer solchen Anforderung ist es in einigen Fällen notwendig bzw. sinnvoll, Berechnungen oder Vorverarbeitungsschritte in andere Programme, wie z.B. MATLAB auszulagern. Für solche Anwendungsfälle bietet NX eine programmierbare Schnittstelle, die NXOpen heißt und in verschiedenen Programmiersprachen (C++, C#, Java, Python) bedient werden kann. Dadurch ist es möglich, in allen drei Modulen (CAD, CAM und CAE) Funktionen zu nutzen, Informationen zu extrahieren und direkten Einfluss auf den laufenden Prozess zu nehmen. Dazu wird eine *Dynamic Link Library (DLL)* programmiert, die zur Laufzeit von NX geladen wird und sich an den laufenden Prozess andockt. Die Applikation verbindet sich zusätzlich mithilfe des COM Automation Servers mit einer (ggf. bereits ausgeführten) MATLAB-Instanz. Für nähere Informationen bzgl. der Verwendung von MATLAB über externe Schnittstellen und die technische Umsetzung sei auf [39] verwiesen. Mit dieser Architektur ist es möglich, die mathematischen Algorithmen und numerischen Berechnungen in MATLAB auszuführen und über die DLL-Applikation an NX weiter zu geben.

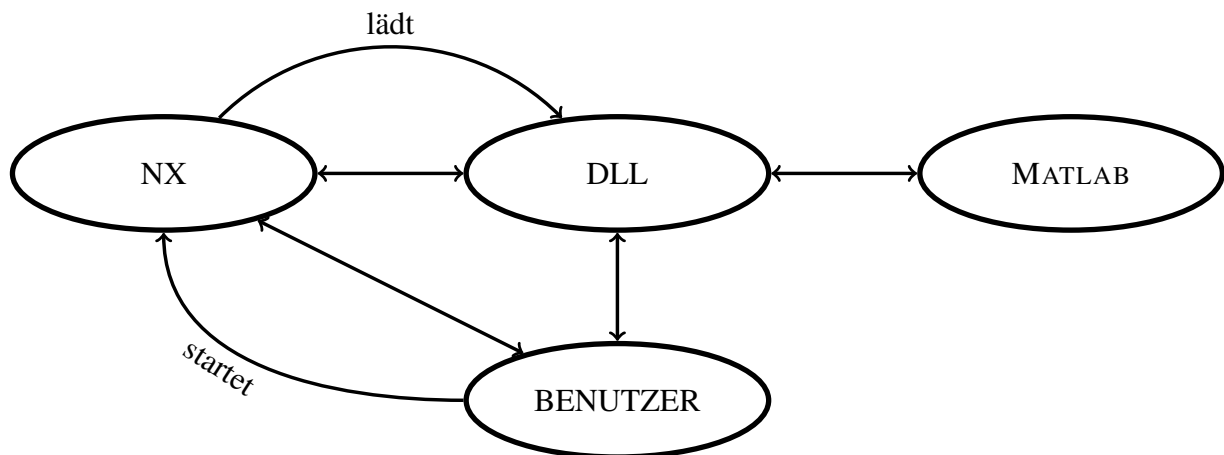


Abb. 2.17: Schematisches Ablaufschema: Der Benutzer startet und arbeitet mit NX und lädt anschließend zur Laufzeit eine programmierte DLL. Diese Bibliothek verbindet sich sowohl mit NX als auch mit MATLAB und ermöglicht den Datenaustausch zwischen allen Komponenten. Durch entsprechende Austauschformate und Synchronisierung aller Prozesse kann eine solche DLL entweder vollautomatisiert im Hintergrund laufen oder mit einer graphischen Benutzeroberfläche interaktiv verwendet werden.

Für diese Arbeit wurde folgendes Szenario realisiert:

Der Benutzer lädt nach dem Start der Software NX eine DLL, die es ihm ermöglicht, Flächeninformationen aus einem geladenen CAD-Modell zu extrahieren und diese anschließend an MATLAB zu übergeben. In MATLAB selbst können implementierte Algorithmen verwendet werden, die eine Tensorprodukt-Fläche erzeugen, die dann ihrerseits wiederum durch die DLL in das CAD-Modell der NX-Instanz geladen werden können. Die DLL selbst kann neben der graphischen Oberfläche von NX ein eigenes Benutzerinterface besitzen und somit an die entsprechenden Bedürfnisse der jeweiligen Applikation angepasst werden. Damit besitzt man eine praktische und generische Testumgebung für mathematische Algorithmen, die unter realen Bedingungen im Einsatz eines CAD-Systems getestet und qualitativ bewertet werden können. Für die Umsetzung der Algorithmen der folgenden Kapitel wurde ein derartiges Framework implementiert und genutzt.

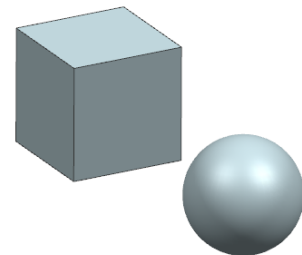
Kombiniert man diese modulare Anordnung der einzelnen Teilkomponenten mit Programmierparadigmen wie z.B. dem Model View Controller Pattern und legt geeignete Austauschformate zwischen NX und MATLAB in Form von spezifizierten \*.m-Dateien fest, wird eine maximale Flexibilität und Austauschbarkeit gewährleistet. Der Vollständigkeit halber sei auf die entsprechende Literatur zum Thema Freiformflächen in NX [1], Programmieren mit MATLAB [5] und Programmieren mit C++ [55] verwiesen. Da die Schnittstelle NXOpen mehr als 10000 Klassen, Funktionen und Methoden umfasst und damit einen schier unüberschaubaren Funktionsumfang besitzt, werden im folgenden Abschnitt nur die für diese Arbeit zentralen Funktionalitäten kurz vorgestellt.

### 2.3.2 Implementierte Funktionalitäten

Folgende Funktionalitäten wurden mithilfe der Schnittstelle NXOpen im Bereich CAD umgesetzt:

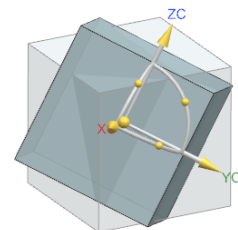
- **Erzeugung von primitiven Körpern**

Mit dieser Funktionalität ist es möglich, automatisiert primitive Objekte wie z.B. Quader, Kugeln oder Kegel mit benutzerdefinierter Größe in das CAD-Modell zu laden.



- **Translation und Rotation von geometrischen Objekten**

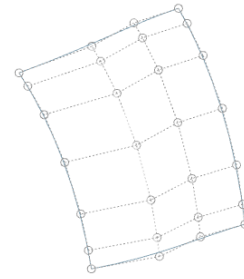
Geometrische Objekte aller Art (Körper, Flächen, Kurven, etc.) können durch Angabe eines Translationsvektors/einer Rotationsachse sowie einem Versatz/einem Winkel transliert/rotiert werden.





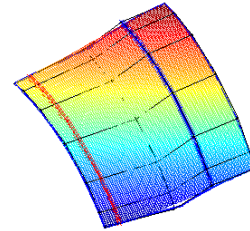
- **Extraktion von Kurven und Flächen**

Der Benutzer kann durch die Auswahl entsprechender Kurven/Flächen im geladenen CAD-Modell die Geometrieinformation als Splinekurve/Tensorprodukt-Fläche extrahieren und in eine Textdatei schreiben.



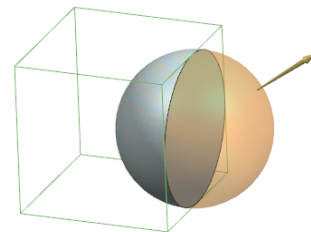
- **Import von Kurven und Flächen**

Kurven und Flächen, die in einer Textdatei vorliegen, können in ein bestehendes CAD-Modell importiert oder neu erstellt werden.



- **Erzeugung von Schnittkurven und -flächen**

Zwei (oder mehr) geometrische Objekte können mit Funktionen des CAD-Systems miteinander geschnitten werden.



Um durch Anwendung des in Kapitel 4 vorgestellten Algorithmus einzelne Werkzeugbahnen zu reparieren, wurden im Bereich CAM folgende Funktionalitäten implementiert:

- Extraktion bereits generierter Werkzeugbahnen  
Es ist über die Schnittstelle NXOpen möglich, bereits erzeugte Werkzeugbahnen einer zugeordneten Geometrie zu extrahieren. Es kann sich abhängig von den eingestellten Parametern hierbei sowohl um Punkte (mit Flächennormale) als auch um Splines handeln.
- Erzeugung neuer Werkzeugbahnen basierend auf der Eingabe von Parametern bzgl. Werkzeugeigenschaften und Verfahrgometrie  
Der Benutzer kann durch die Verwendung der Schnittstelle eigene Werkzeugbahnen erzeugen oder auf Basis bereits vorhandener Methoden die Bahnen neu generieren.
- Anpassung einzelner Punkte einer Werkzeugbahn (Einfügen, Ändern, Entfernen)  
In NX besteht darüber hinaus die Möglichkeit, einzelne Punkte einer Werkzeugbahn zu extrahieren, zu verändern, einzufügen und zu entfernen.

Um die implementierten Funktionalitäten der Schnittstelle NXOpen in Verbindung mit MATLAB optimal nutzen zu können, wurde im Zuge dieser Arbeit ein Austauschformat mit entsprechenden Parsern entwickelt, das den reibungslosen Datenaustausch von Splinekurven und Tensorprodukt-

Flächen ermöglicht.

Aufbauend auf den hier vorgestellten mathematischen Grundlagen sowie den Strukturen der Software-Testumgebung für Algorithmen und CAD-Anwendungen werden in den folgenden Kapiteln die Stetigkeit zwischen parametrischen Flächen sowie die Erzeugung  $G^2$ -stetiger Übergänge behandelt. Alle in dieser Arbeit vorgestellten Verfahren wurden mit C# (Visual Studio 2010 Professional) implementiert und mit NX 10 und MATLAB R2012b verwendet und getestet.

Die Glattheit von Kurven und Flächen nimmt in heutigen CAD-Systemen aus verschiedensten Gründen eine tragende Rolle ein. Beginnend bei der aerodynamischen Konstruktion eines Flugzeugs bis hin zu der Modellbildung visuell ästhetischer Spritzgußformen oder anspruchsvollen Zahnradgeometrien sind „glatte“ Übergänge von essentieller Bedeutung. Insbesondere in Kombination mit CAM-Systemen und der damit verbundenen Generierung von Werkzeugbahnen sind solche Übergänge oftmals auch technisch motiviert, da z.B. kleine Abweichungen von Flächennormalen im CAD-Modell zu unbeabsichtigten Ruckeffekten an einer Werkzeugmaschine führen können, die letztendlich die Qualität der Oberfläche des Werkstücks merklich verschlechtern. In Kapitel 2.1 wurden sowohl parameterabhängige als auch parameterunabhängige Eigenschaften und Kenngrößen von Kurven und Flächen vorgestellt. Der Begriff der Stetigkeit lässt sich ebenfalls mit Bezug auf diese extrinsischen und intrinsischen Eigenschaften einführen. Ist von einer parameterabhängigen Stetigkeit die Rede, spricht man in der Literatur von der sogenannten  $C$ -Stetigkeit, während man den Ausdruck  $G$ -Stetigkeit verwendet, wenn zur Betrachtung parametrisierungsunabhängige Größen, wie z.B. die Flächennormale, verwendet werden. Zunächst wird der Kurvenfall [4] näher betrachtet, bevor anschließend die Konzepte auf Flächen erweitert werden (vgl. hierfür [17, 23, 30, 53]).

## 3.1 $C^n/G^n$ -stetige Kurvenübergänge

In diesem Abschnitt wird die Stetigkeit von Kurvenübergängen näher untersucht und der Unterschied zwischen  $C$ -Stetigkeit und  $G$ -Stetigkeit erläutert. Diese Stetigkeiten werden in der Literatur auch oftmals als *parametrische* bzw. *geometrische* Stetigkeit bezeichnet. Unter dem Begriff Stetigkeit höherer Ordnung ist die Stetigkeit von Ableitungen zu verstehen. Es werden Methoden vorgestellt, die es ermöglichen, die Stetigkeit von Flächenübergängen mithilfe von Minimierungsproblemen numerisch stabil zu prüfen. Den Abschluss dieses Kapitels bildet die punktweise Untersuchung der Normalenkrümmungen zweier  $G^1$ -stetiger Flächen  $F, G$  in einer Tangentialrichtung.

### 3.1.1 Parametrische Kurven

Eine erste Möglichkeit, die Stetigkeit zweier Kurven an einem gemeinsamen Übergang zu definieren, bietet die  $C^n$ -Stetigkeit.

**Definition 3.1.** Zwei Parametrisierungen  $f, g$  sind an einem Punkt  $C^n$ -stetig, wenn nach entsprechender Verschiebung der Knotenvektoren alle Ableitungen an der Stelle  $u$  bis zur Ordnung  $n$  übereinstimmen

$$f^{(k)}(u) = g^{(k)}(u), \quad k = 0, \dots, n.$$

**Bemerkung 3.2.** Die in Definition 3.1 angesprochene Verschiebung stellt keine Einschränkung dar und sorgt lediglich für eine übersichtliche Notation. Seien zwei Splinekurven  $f$  und  $g$  gegeben, die jeweils normierte Knotenvektoren besitzen und sich an ihren Endpunkten ( $f(1) = g(0)$ ) berühren. In dieser Arbeit wird implizit eine Verschiebung des Knotenvektors von  $g$  vorausgesetzt, sodass  $f$  und  $g$  ihren gemeinsamen Punkt beim selben Parameterwert (hier z. B.  $u = 1$ ) annehmen. Selbige Annahme lässt sich ohne Weiteres auch auf Flächen anwenden.

Durch eine reguläre Reparametrisierung  $\varphi$  lässt sich die Definition der  $C^n$ -Stetigkeit allgemeiner formulieren und man erhält die sogenannte  $G^n$ -Stetigkeit.

**Definition 3.3.** Seien zwei Parametrisierungen  $f, g$  gegeben. Der Kurvenübergang heißt  $G^n$ -stetig, wenn eine reguläre Reparametrisierung  $\varphi$  (vgl. Definition 2.3) existiert, sodass

$$f^{(k)}(u) = (g \circ \varphi)^{(k)}(u), \quad k = 0, \dots, n$$

gilt.

Der Vorteil eines Stetigkeitsbegriffes unter Zuhilfenahme einer Reparametrisierung  $\varphi$  im Gegensatz zur reinen Betrachtung der jeweiligen Ableitungen wird in nachfolgendem Beispiel (vgl. [22, Kapitel 12]) verdeutlicht:

**Beispiel 3.4.** Seien zwei zueinander symmetrisch angeordnete Parabeln wie in Abbildung 3.1 gegeben. Betrachtet man beide Kurven am gemeinsamen Punkt  $p$ , so lassen sich folgende Aussagen treffen:

- Die zwei Parabeln besitzen einen gemeinsamen Punkt  $p$  und sind in diesem sowohl tangential als auch krümmungsstetig.

Anhand der Symmetrie und der daraus resultierenden Anordnung der Kontrollpunkte folgen sowohl die Tangentenstetigkeit als auch die Krümmungsstetigkeit sofort.

- Die Kombination beider Kurven liefert keine  $C^2$ -Kurve.

Obwohl die beiden Parabeln in Abbildung 3.1 im gemeinsamen Punkt sogar  $C^2$ -stetig sind, existiert keine Darstellungsform als eine einzelne quadratische Kurve. Anschaulich lässt sich dieser Zusammenhang damit erklären, dass keine Parabel existiert, die an zwei unterschiedlichen Punkten dieselbe Tangentenrichtung besitzen.

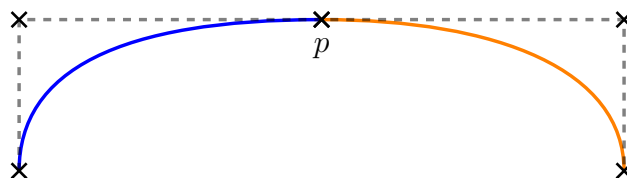


Abb. 3.1: Zwei Parabeln, deren Krümmung am gemeinsamen Punkt übereinstimmt

Ein sehr ähnlicher Sachverhalt lässt sich ebenfalls mit einer quadratischen Funktion darstellen. In Abbildung 3.2 ist eine Parabel (blau) abgebildet, die mithilfe der orangen Splinekurven  $G^2$ -stetig fortgesetzt wird. Am Übergang  $p$  sind alle Kombinationen  $C^1$ -stetig und sogar krümmungsstetig, aber nur eine dieser Kurven liefert eine  $C^2$ -stetige Parabel.

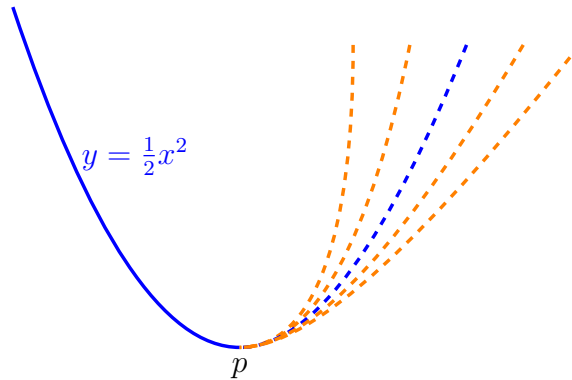
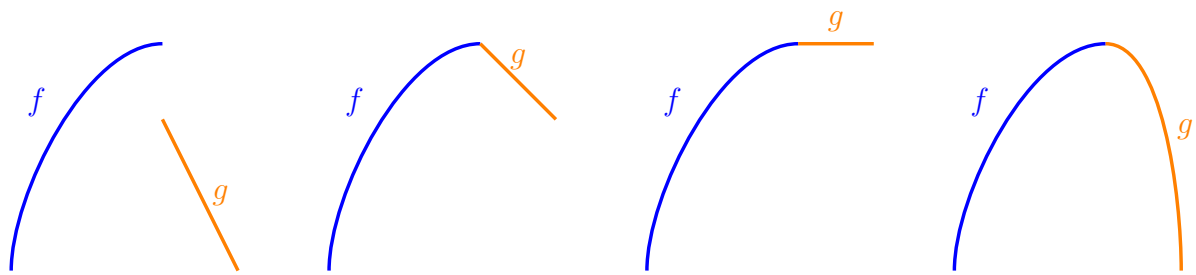


Abb. 3.2: Parabel, die  $G^2$ -stetig fortgesetzt wird und nur mit einer Splinekurve zu einer  $C^2$ -stetigen Parabel zusammengefügt werden kann.

**Bemerkung 3.5.** In dieser Arbeit wird vereinbart, dass ein „Kurvenübergang“ stets nur am Start- bzw. Endpunkt zweier parametrischer Kurven auftreten kann. Insbesondere bei der Anwendung mit Splinekurven und -flächen lassen sich die vereinbarten  $(m + 1)$ -fachen Randknoten ausnutzen. Diese Annahme ist von zentraler Bedeutung, da im praktischen Umfeld von CAD-Anwendungen nur derartige Übergänge eine Rolle spielen.

**Beispiel 3.6.** In Abbildung 3.3 sind alle für diese Arbeit relevanten Stetigkeiten dargestellt. Während die beiden Kurven in Abbildung 3.3a keinen gemeinsamen Punkt besitzen und damit unstetig sind, teilen sie sich in 3.3 (b) einen gemeinsamen Punkt. Im dritten Fall zeigen die Tangenten beider Parametrisierungen  $f, g$  in dieselbe Richtung und zusätzlich liegt kein Umkehrpunkt vor. Ein solcher Übergang heißt dann  $G^1$ -stetig. In der letzten Abbildung rechts stimmen sogar die Krümmungen im gemeinsamen Punkt überein.



(a) Unstetiger Übergang    (b)  $G^0$ -stetiger Übergang    (c)  $G^1$ -stetiger Übergang    (d)  $G^2$ -stetiger Übergang

Abb. 3.3: Beispiele für verschiedene  $G$ -stetige Kurvenübergänge

### 3.1.2 Splinekurven

Besonders interessant bei der Betrachtung der Stetigkeit eines Übergangs zweier Splinekurven mit vorgegebenen Knotenfolgen ist die Lage der verschiedenen Kontrollpunkte zueinander. Die nachfolgenden Kriterien sind bekannt und z.B. in [22, 30, 46] zu finden. Die folgenden Sätze werden aufgrund der Notation und besseren Übersicht gezeigt und bewiesen. Konkret lautet die Fragestellung, wo die Kontrollpunkte liegen müssen, damit ein Kurvenübergang  $G^{0,1,2}$ -stetig ist und welche Freiheitsgrade zur Verfügung stehen, um ggf. Einfluss auf den Kurvenverlauf zu nehmen.

#### $G^0$ -stetiger Übergang

Die Basis für alle weiteren Untersuchungen eines Übergangs zweier Splinekurven bildet die  $G^0$ -Stetigkeit. Diese ist sehr intuitiv und aufgrund der Endpunktinterpolation durch  $(m + 1)$ -fache Randknoten sofort ersichtlich:

**Satz 3.7.** *Sei die Splinekurve  $f$  und die Knotenfolge  $g.T$  mit  $(m + 1)$ -fachen Randknoten (vgl. Bemerkung 2.31) gegeben. Die Splinekurve  $g$  schließt  $G^0$ -stetig an  $f$  an, wenn der erste Kontrollpunkt von  $g$  mit dem letzten Kontrollpunkt von  $f$  übereinstimmt.*

**Beweis:** Aufgrund der  $(m+1)$ -fachen Randknoten wird sowohl der erste als auch letzte Punkt des Kontrollpolygons interpoliert. Sei  $u = \max(f.T)$  und  $\varphi(x) : x \rightarrow x + (\min(g.T) - \max(f.T))$ . Damit folgt sofort die Bedingung für  $G^0$ -Stetigkeit

$$\begin{aligned} f(u) &= (g \circ \varphi)(u) \\ f.d \mathbf{V}_f(u) &= g.d \mathbf{V}_g(\varphi(u)) \end{aligned}$$

$$f.d \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} = g.d \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

□

Die Reparametrisierung  $\varphi$  ist in diesem Fall eine Translation des Parameterbereichs von  $f$  nach  $g$ . Zwischen  $G^0$  und  $C^0$  existiert damit kein Unterschied, beide Kurven besitzen einen gemeinsamen Endpunkt.

#### $G^1$ -stetiger Übergang

Sind zwei Splinekurven  $f, g$   $C^0$ -stetig, kann man durch die Kenntnis der ersten Ableitung von  $f$  eine Bedingung an den zweiten Kontrollpunkt  $(g.d)_2$  herleiten, sodass der Übergang  $G^1$ -stetig

ist. Dazu ist es sinnvoll, mit der Betrachtung der Ableitung  $(g \circ \varphi)'(u)$  zu beginnen:

$$(g \circ \varphi)'(u) = \varphi'(u)g'(\varphi(u)) = \varphi'(u) g.d \mathbf{V}'_g(\varphi(u))$$

Mit dieser Vorüberlegung lässt sich für einen  $G^0$ -stetigen Übergang zweier Splinekurven  $f, g$  die Bedingung für den nächsten Kontrollpunkt von  $g$  herleiten, sodass beide Parametrisierungen  $G^1$ -stetig aneinanderschließen.

**Satz 3.8.** Sei ein  $G^0$ -stetiger Übergang zweier Splinekurven  $f$  und  $g$  an der Parameterstelle  $u$  gegeben. Der Übergang ist  $G^1$ -stetig, wenn für den zweiten Kontrollpunkt von  $g$

$$(g.d)_2 = \frac{\alpha f'(u) + \beta(g.d)_1}{\beta}, \quad \alpha \neq 0,$$

gilt, wobei  $\beta$  durch die Berechnung von  $\mathbf{V}'_g(\varphi(u)) = \begin{pmatrix} -\beta \\ \beta \\ 0 \\ \vdots \end{pmatrix}$  festgelegt ist.

**Beweis:** Damit ein Kurvenübergang  $G^1$ -stetig ist, muss gezeigt werden, dass die beiden Bedingungen

$$f(u) = (g \circ \varphi)(u) \quad (G^0\text{-Stetigkeit})$$

$$f'(u) = (g \circ \varphi)'(u) \quad (G^1\text{-Stetigkeit})$$

erfüllt sind. Da der Übergang nach Voraussetzung bereits  $G^0$ -stetig ist, muss also nur noch Bedingung 2 gezeigt werden. Die Vorüberlegung und die Tatsache, dass  $\varphi(u) = \min(g.T)$  ist, liefern

$$f'(u) = \varphi'(u) g.d \mathbf{V}'_g(\varphi(u)) = \varphi'(u) g.d \begin{pmatrix} -\beta \\ \beta \\ 0 \\ \vdots \end{pmatrix} = \varphi'(u) \beta ((g.d)_2 - (g.d)_1)$$

und damit

$$(g.d)_2 = \frac{\alpha f'(u) + \beta(g.d)_1}{\beta}, \quad \alpha = \frac{1}{\varphi'(u)}$$

$$f'(u) = \frac{\beta}{\alpha} ((g.d)_2 - (g.d)_1) = \tilde{\alpha} ((g.d)_2 - (g.d)_1).$$

□

Für einen gegebenen Knotenvektor  $g.T$  ist der Parameter  $\beta$  implizit festgelegt und es verbleibt ein frei wählbarer Parameter  $\tilde{\alpha}$ . Der zur Verfügung stehende Freiheitsgrad ist damit die Ableitung der Reparametrisierung, also ein linearer Faktor, der an die Ableitung von  $f$  multipliziert wird. Die Ableitung einer regulären Reparametrisierung  $\varphi$  ist definitionsgemäß stets ungleich 0. Geometrisch betrachtet liegt der Kontrollpunkt  $(g.d)_2$  damit auf einer Geraden durch den vor-

letzten und letzten Kontrollpunkt von  $f$  (vgl. Abbildung 3.4), wobei der Punkt  $(g.d)_1$  aufgrund von  $\alpha \neq 0$  ausgeschlossen ist.

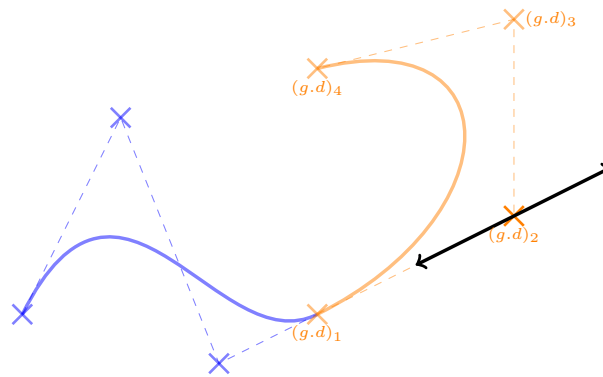


Abb. 3.4: Visualisierung des Freiheitsgrades bei der Bestimmung der  $G^1$ -Stetigkeit. Die orange Kurve  $g$  schließt an  $f$   $G^1$ -stetig an, solange der zweite Kontrollpunkt  $(g.d)_2$  nur in Richtung der schwarz markierten Pfeile verschoben wird (und ungleich  $(g.d)_1$  ist).

**Bemerkung 3.9.** *Gibt es eine Reparametrisierung  $\varphi$ , sodass an einem gemeinsamen Punkt zweier Kurven  $f, g$  der Zusammenhang*

$$f'(u) = (g \circ \varphi)'(u),$$

mit  $\varphi'(u) = 1$  gilt, so ist der Übergang  $C^1$ -stetig.

### $G^2$ -stetiger Übergang

Setzt man die Taylorentwicklung der Reparametrisierung  $\varphi$  fort, so erhält man die Bedingungen für die  $G^2$ -Stetigkeit. Analog zur Herleitung der  $G^1$ -Stetigkeit wird zunächst mit der Betrachtung der Ableitung, also von  $(g \circ \varphi)''(u)$ , begonnen:

$$\begin{aligned} (g \circ \varphi)''(u) &= (\varphi'(u)(g' \circ \varphi)(u))' \\ &= \varphi'(u)g''(\varphi(u))\varphi'(u) + g'(\varphi(u))\varphi''(u). \end{aligned} \quad (3.1)$$

Sind nun die ersten beiden Kontrollpunkte derart gewählt, dass der Kurvenübergang  $G^1$ -stetig ist, so kann die Bedingung an den dritten Kontrollpunkt  $(g.d)_3$ , sodass darüber hinaus die Krümmungen beider Splinekurven  $f, g$  am Übergang identisch sind, wie folgt formuliert werden.

**Satz 3.10.** *Sei ein  $G^1$ -stetiger Kurvenübergang zweier Splinekurven  $f, g$  gegeben und  $\alpha$  durch die Reparametrisierung  $\varphi$  festgelegt. Der Übergang ist  $G^2$ -stetig, wenn für den dritten Kontrollpunkt*

$$(g.d)_3 = \frac{\alpha^2 f''(u) - \lambda f'(u) + \delta_2 (g.d)_2 - \delta_1 (g.d)_1}{\delta_3}, \quad \lambda \in \mathbb{R},$$



$$\text{wobei } \mathbf{V}_g''(\varphi(u)) = \begin{pmatrix} \delta_1 \\ -\delta_2 \\ \delta_3 \\ 0 \\ \vdots \end{pmatrix} \text{ gilt.}$$

**Beweis:** Um die  $G^2$ -Stetigkeit eines Kurvenübergangs nachzuweisen, müssen die drei Bedingungen

$$\begin{aligned} f(u) &= (g \circ \varphi)(u) && (G^0\text{-Stetigkeit}) \\ f'(u) &= (g \circ \varphi)'(u) && (G^1\text{-Stetigkeit}) \\ f''(u) &= (g \circ \varphi)''(u) && (G^2\text{-Stetigkeit}) \end{aligned}$$

erfüllt sein. Die erste und zweite Bedingung sind nach Voraussetzung bereits erfüllt, es gilt also noch

$f''(u) = (g \circ \varphi)''(u)$  nachzuweisen:

$$\begin{aligned} f''(u) &= (g \circ \varphi)''(u) \\ &= \varphi'(u)g''(\varphi(u))\varphi'(u) + \varphi''(u)g'(\varphi(u)) \\ &= g.d \mathbf{V}_g''(\varphi(u))\varphi'(u)^2 + g.d \mathbf{V}_g'(\varphi(u))\varphi''(u) \end{aligned}$$

$$f''(u) = g.d \begin{pmatrix} \delta_1 \\ -\delta_2 \\ \delta_3 \\ 0 \\ \vdots \end{pmatrix} \varphi'(u)^2 + g.d \begin{pmatrix} -\beta \\ \beta \\ 0 \\ \vdots \end{pmatrix} \varphi''(u)$$

$$\frac{f''(u)}{\varphi'(u)^2} = \delta_1(g.d)_1 - \delta_2(g.d)_2 + \delta_3(g.d)_3 + \frac{f'(u)}{\varphi'(u)^3} \varphi''(u)$$

$$(g.d)_3 = \frac{\alpha^2 f''(u) - \lambda f'(u) + \delta_2(g.d)_2 - \delta_1(g.d)_1}{\delta_3}, \quad \lambda = \frac{\varphi''(u)}{\varphi'(u)^3}.$$

□

Die Gleichung für  $(g.d)_3$  lässt sich noch in drei separate Anteile aufspalten, sodass ihre geometrische Bedeutung klarer wird

$$(g.d)_3 = \alpha^2 \frac{f''(u)}{\delta_3} - \lambda \frac{f'(u)}{\delta_3} + \frac{\delta_2(g.d)_2 - \delta_1(g.d)_1}{\delta_3},$$

wie nachfolgendes Beispiel veranschaulichen soll.

**Beispiel 3.11.** Seien zwei Splinekurven  $f, g$  wie in Abbildung 3.5 gegeben. Verändert man den Wert des Freiheitsgrades  $\alpha$  und setzt  $\lambda = 0$ , so wandert der Kontrollpunkt  $(g.d)_3$  entlang der Parabel, die links abgebildet ist. Der Parameter  $\lambda$  gibt die Verschiebung des Kontrollpunktes in Richtung der ersten Ableitung an. Dieser Sachverhalt ist in der rechten Abbildung zu erkennen.

**Bemerkung 3.12.** Analog zur Bemerkung 3.9 lässt sich auch für die  $G^2$ -Stetigkeit ein Zusammen-

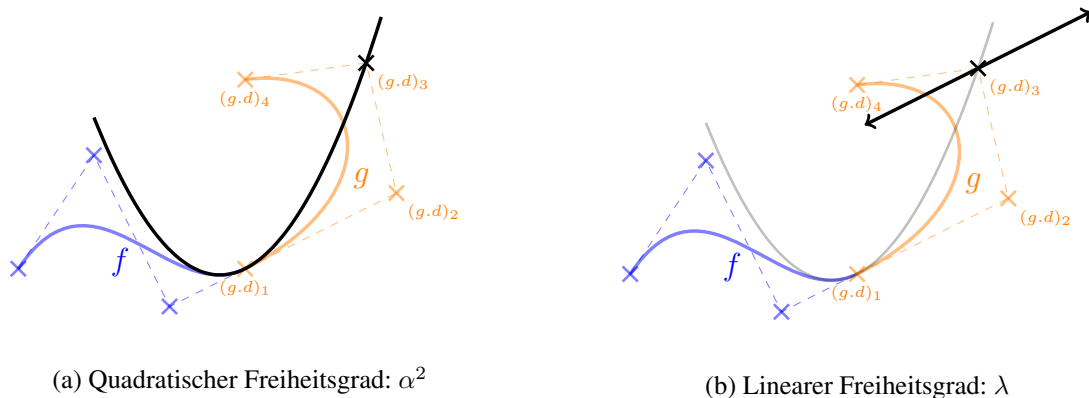


Abb. 3.5: Aufspaltung und Visualisierung des quadratischen und linearen Anteils der Freiheitsgrade  $\alpha$  und  $\lambda$ . Verschiebt man den zweiten Kontrollpunkt  $(g.d)_2$  linear, wandert  $(g.d)_3$  entlang der Parabel (Freiheitsgrad  $\alpha$ ). Darüber hinaus kann  $(g.d)_3$  selbst linear mit  $\lambda$  verschoben werden, ohne die Krümmung am gemeinsamen Punkt zu verändern.

menhang zwischen  $C^2$ -Stetigkeit und der Reparametrisierung  $\varphi$  herstellen. Ein  $G^2$ -stetiger Übergang zweier Kurven mit

$$\begin{aligned} f(u) &= (g \circ \varphi)(u) \\ f'(u) &= (g \circ \varphi)'(u) \\ f''(u) &= (g \circ \varphi)''(u) \end{aligned}$$

ist  $C^2$ -stetig, wenn am gemeinsamen Punkt zusätzlich

$$\varphi'(u) = 1, \quad \varphi''(u) = 0$$

gilt.

Im nächsten Abschnitt wird ähnlich zum Prinzip der Hermite-Interpolation die Möglichkeit untersucht, eine Splinekurve zu erzeugen, die  $G^2$ -Stetigkeitsbedingungen am Start- und Endpunkt erfüllt und darüber hinaus die Freiheitsgrade nutzt, um ein Glättungsfunktional zu minimieren. Aufgrund der Übersichtlichkeit wird die Vorgehensweise anhand eines Beispiels erklärt, das Konzept ist allerdings allgemein anwendbar und z.B. mit der Interpolation gegebener Punkte kombinierbar.

### 3.1.3 Minimierung mit $G^2$ -Randbedingungen

Gegeben seien zwei Punkte  $p_1, p_2 \in \mathbb{R}^3$  sowie Ableitungen erster ( $p'_1, p'_2 \in \mathbb{R}^3$ ) und zweiter ( $p''_1, p''_2 \in \mathbb{R}^3$ ) Ordnung. Weiter sei der Grad  $m = 3$  und eine Knotenfolge

$$f.T = [0, 0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1, 1]$$

gegeben. Gesucht ist das Kontrollpolygon  $d$ , bestehend aus 6 Kontrollpunkten, das das Funktional

$$\mathcal{F} = \int_{[f.T]} \|f''(u)\|_2^2 \, d u$$

unter den Nebenbedingungen

$$\begin{aligned} f(\varphi(0)) &= p_1 & f(\varphi(1)) &= p_2 \\ f'(\varphi(0)) &= p'_1 & f'(\varphi(1)) &= p'_2 \\ f''(\varphi(0)) &= p''_1 & f''(\varphi(1)) &= p''_2 \end{aligned}$$

minimiert. In diesem Fall erhält man für jeden  $G^2$ -stetigen Übergang jeweils zwei Freiheitsgrade, nämlich  $\alpha_1, \lambda_1$  und  $\alpha_2, \lambda_2$ . Um zu entscheiden, welche Freiheitsgrade zur Minimierung von  $\mathcal{F}$  besser oder schlechter geeignet sind, wird zunächst das Funktional näher betrachtet:

$$\begin{aligned} \mathcal{F} &= \int_{[f.T]} \|f''(u)\|_2^2 \, d u \\ &= \int_{[f.T]} \text{trace}(f.d \mathbf{V}''(u) \mathbf{V}''(u)^T f.d^T) \, d u \\ &= \text{trace}(f.d \mathbf{A} f.d^T). \end{aligned}$$

Mithilfe der hergeleiteten Stetigkeitsbedingungen lässt sich das Kontrollpolygon als Funktion in den Variablen  $\alpha_1, \alpha_2, \lambda_1$  und  $\lambda_2$  darstellen.

$$f.d(\alpha_1, \alpha_2, \lambda_1, \lambda_2) = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{pmatrix}^T = \begin{pmatrix} p_1 \\ \frac{\alpha_1 p'_1 + \beta d_1}{\beta} \\ \frac{\alpha_1^2 p''_1 - \lambda_1 p'_1 + \delta_2 d_2 - \delta_1 d_1}{\delta_3} \\ \frac{\alpha_2^2 p''_2 - \lambda_2 p'_2 + \delta_2 d_5 - \delta_1 d_6}{\delta_3} \\ \frac{\alpha_2 p'_2 - \beta d_6}{-\beta} \\ p_2 \end{pmatrix}^T, \quad \mathbf{V}'(0) = \begin{pmatrix} -\beta \\ \beta \\ 0 \\ \vdots \end{pmatrix}, \quad \mathbf{V}''(0) = \begin{pmatrix} \delta_1 \\ -\delta_2 \\ \delta_3 \\ 0 \\ \vdots \end{pmatrix}$$

Durch die Betrachtung des Quadrats  $\|f''(u)\|_2^2$  ergibt sich in der Gesamtheit damit der Höchstgrad vier für  $\alpha_1, \alpha_2$  und zwei für  $\lambda_1, \lambda_2$ . Optimiert man diese Problemstellung für alle Parameter zur selben Zeit ergibt sich ein quartisches Problem, das mit einem nicht-linearen Lösungsalgorithmus (z.B. mehrdimensionales Newton-Verfahren [25]) näherungsweise gelöst werden muss. Fixiert man allerdings die Parameter  $\alpha_1, \alpha_2$  ist das Minimierungsproblem quadratischer Natur und mit Standardverfahren der linearen Algebra lösbar. Bevor die partiellen Ableitungen  $\frac{\partial}{\partial \lambda_1} \mathcal{F}$  und  $\frac{\partial}{\partial \lambda_2} \mathcal{F}$  berechnet werden, wird zur besseren Lesbarkeit das Kontrollpolygon nur in Abhängig-

keit von  $\lambda_1, \lambda_2$  vereinfacht dargestellt

$$f.d(\lambda_1, \lambda_2) = \begin{pmatrix} \tilde{d}_1 \\ \tilde{d}_2 \\ \tilde{d}_3 - \lambda_1 \frac{p'_1}{\delta_3} \\ \tilde{d}_4 - \lambda_2 \frac{p'_2}{\delta_3} \\ \tilde{d}_5 \\ \tilde{d}_6 \end{pmatrix}^T.$$

Diese Darstellung taucht in der Literatur meistens als  $\gamma$ -Spline auf, für weiterführende Informationen sei auf [7] verwiesen. Da bei den partiellen Ableitungen nach  $\lambda_1, \lambda_2$  alle konstanten Terme wegfallen, werden im nächsten Schritt nur Terme von  $f.d \mathbf{A} f.d^T$  betrachtet, die nicht konstant sind:

$$\begin{aligned} \tilde{\mathcal{F}}(\lambda_1, \lambda_2) &= \text{trace}(f.d(\lambda_1, \lambda_2) \mathbf{A} f.d(\lambda_1, \lambda_2)^T) \\ &= \text{trace}\left(\sum_{j=1}^6 \sum_{i=1}^6 (\tilde{d})_i a_{i,j} (\tilde{d})_j^T\right) \\ &= 2 \sum_{i=1}^6 -\lambda_1 \frac{p'_1}{\delta_3} a_{3,i} \tilde{d}_i^T + \lambda_1^2 \frac{p'_1}{\delta_3} a_{3,3} \frac{p'_1}{\delta_3} + 2\lambda_1 \lambda_2 \frac{p'_1}{\delta_3} a_{3,4} \frac{p'_2}{\delta_3} \\ &\quad + 2 \sum_{i=1}^6 -\lambda_2 \frac{p'_2}{\delta_3} a_{4,i} \tilde{d}_i^T + \lambda_2^2 \frac{p'_2}{\delta_3} a_{4,4} \frac{p'_2}{\delta_3}. \end{aligned}$$

Durch diese Darstellung können die partiellen Ableitungen angegeben werden:

$$\begin{aligned} \frac{\partial}{\partial \lambda_1} \mathcal{F} &= 2 \sum_{i=1}^6 -\frac{p'_1}{\delta_3} a_{3,i} \tilde{d}_i^T + 2\lambda_1 \frac{p'_1}{\delta_3} a_{3,3} \frac{p'_1}{\delta_3} + 2\lambda_2 \frac{p'_1}{\delta_3} a_{3,4} \frac{p'_2}{\delta_3} \\ \frac{\partial}{\partial \lambda_2} \mathcal{F} &= 2 \sum_{i=1}^6 -\frac{p'_2}{\delta_3} a_{4,i} \tilde{d}_i^T + 2\lambda_2 \frac{p'_2}{\delta_3} a_{4,4} \frac{p'_2}{\delta_3} + 2\lambda_1 \frac{p'_1}{\delta_3} a_{3,4} \frac{p'_2}{\delta_3} \end{aligned}$$

Im letzten Schritt wird das lineare Gleichungssystem aufgestellt, um mithilfe der Nullstellen der partiellen Ableitungen das Minimum von  $\mathcal{F}$  zu finden

$$\begin{pmatrix} \frac{p'_1}{\delta_3} a_{3,3} \frac{p'_1}{\delta_3} & \frac{p'_1}{\delta_3} a_{3,4} \frac{p'_2}{\delta_3} \\ \frac{p'_1}{\delta_3} a_{3,4} \frac{p'_2}{\delta_3} & \frac{p'_2}{\delta_3} a_{4,4} \frac{p'_2}{\delta_3} \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^6 \frac{p'_1}{\delta_3} a_{3,i} \tilde{d}_i^T \\ \sum_{i=1}^6 \frac{p'_2}{\delta_3} a_{4,i} \tilde{d}_i^T \end{pmatrix}.$$

In der nachfolgenden Abbildung 3.6 ist eine Problemstellung dieser Art dargestellt:

**Beispiel 3.13.** *Gesucht ist eine Splinekurve, die sowohl am Punkt  $p_1$  als auch am Punkt  $p_2$   $G^2$ -stetig an die blauen Splinekurven anschließt. Für die fixierten Parameter wurde in diesem Bei-*

spiel  $\alpha_1 = \alpha_2 = 1$  gewählt. Während die grüne Kurve ohne Optimierung mit  $\lambda_1 = \lambda_2 = 0$  entstanden ist, wurde für die orange Splinekurve  $f$  das obige Minimierungsproblem gelöst.

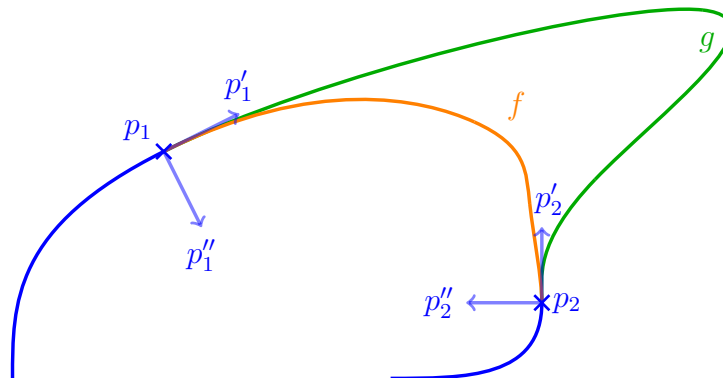


Abb. 3.6: Zwei unterschiedliche Lösungen für eine Splinekurve mit  $G^2$ -Randbedingungen. Die grüne Splinekurve wurde mit  $\lambda_1 = \lambda_2 = 0$  erzeugt, die orange Splinekurve ist die Lösung des Minimierungsproblems mit  $\lambda_1 = 0.82$  und  $\lambda_2 = 0.41$ .

Obwohl die Krümmung an beiden Punkten  $p_1, p_2$  in beiden Fällen  $f, g$  übereinstimmt, ist deutlich zu erkennen, dass die Lösung des Minimierungsproblems eine glattere und optisch ansprechendere Lösung liefert.

Die Interpolation  $C^2$ -stetiger Daten mithilfe kubischer Kurven, die unter bestimmten Voraussetzungen an Start- und Endpunkt krümmungsstetig anschließen, ist z. B. in [11] nach zu lesen.

### 3.2 $C^n/G^n$ -stetige Flächenübergänge

Die hier behandelten  $C/G$ -Stetigkeitskonzepte für Kurven lassen sich auf Flächen erweitern. Bei der Betrachtung und Klassifizierung von Stetigkeiten an Flächenübergängen muss man allerdings im Gegensatz zu den Kurven nicht einen einzelnen Punkt sondern eine ganze Kante betrachten. Wie bei den Kurven macht die Betrachtung von Flächenübergängen nur unter der Voraussetzung Sinn, dass Übergänge ausschließlich am Rand, also den Kanten von Flächen, auftreten.

#### Definition der $C^n/G^n$ -Stetigkeit für Flächenparametrisierungen

Für zwei parametrische Flächen  $F, G$  lässt sich die  $C^n$ -Stetigkeit wie folgt definieren.

**Definition 3.14.** Zwei parametrische Flächen  $F, G$  sind in einem Punkt  $C^n$ -stetig, wenn

$$D^k F(u, v) = D^k G(u, v), \quad k = 0, \dots, n,$$

gilt. Ein Flächenübergang heißt  $C^n$ -stetig, wenn die Flächen entlang einer gemeinsamen Kante punktweise  $C^n$ -stetig sind.

Auch im Flächenfall lässt sich unter der Zuhilfenahme einer Reparametrisierung der Begriff der  $G$ -Stetigkeit definieren. Der Unterschied zum Kurvenfall liegt allerdings darin, dass die Reparametrisierung  $\varphi$  nun anstatt einer univariaten Funktion eine bivariate Koordinatentransformation ist, die entlang der gemeinsamen Kante nicht konstant sein darf. Bevor die  $G^0$ -,  $G^1$ - und  $G^2$ -Stetigkeit für Flächen eingeführt und hergeleitet wird, ist es sinnvoll, die Ableitungen multivariater verketteter Funktionen im allgemeinen Fall zu betrachten und eine passende Notation einzuführen.

## Ableitungen multivariater verketteter Funktionen

Im folgenden Abschnitt werden die bekannten Ableitungsregeln auf multivariate verkettete Funktionen angewandt und als Matrixmultiplikation formuliert. In der Standardliteratur (z.B. [20, 24]) wird häufig nur die erste Ableitung zweier verketteter Funktionen

$$f : \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad g : \mathbb{R}^n \rightarrow \mathbb{R}^n \quad (3.2)$$

betrachtet, da sich dieser Fall noch sehr anschaulich als Multiplikation von Jakobi-Matrizen darstellen lässt und damit die Funktionsweise der Kettenregel erklärt werden kann. Da die Dimension des Bildraums hier allerdings größer als 1 ist, ergibt sich für die Hesse-„Matrix“ ein Tensor, der erst nach einer entsprechenden Umordnung wieder als Produkt von Matrizen dargestellt werden kann. Seien für diesen Abschnitt die zwei Funktionen  $f, g$  keine Splinekurven, sondern allgemeine Funktionen wie in (3.2) definiert. Weiter sei  $u \in \mathbb{R}^n$  ein entsprechender Parameterwert mit Komponenten  $u = (u_1, u_2, \dots)^T$ , an dem die Ableitung gesucht ist. Für die erste Ableitung von  $(f \circ g)$  gilt

$$\begin{aligned} D(f \circ g)(u) &= \left[ \sum_{r=1}^n \frac{\partial f}{\partial u_r}(g(u)) \frac{\partial g_r}{\partial u_k}(u) : k = 1, \dots, n \right] \\ &= \mathbf{J}_f(g(u)) \mathbf{J}_g(u). \end{aligned}$$

Sei  $y \in \mathbb{R}^n$  eine Richtung mit  $\|y\|_2 = 1$ . Dann ist die Richtungsableitung  $D_y(f \circ g)$  an einer Stelle  $u$  gegeben durch

$$\begin{aligned} D_y(f \circ g)(u) &= \sum_{k=1}^n \sum_{r=1}^n \frac{\partial f}{\partial u_r}(g(u)) \frac{\partial g_r}{\partial u_k}(u) y_k \\ &= \mathbf{J}_f(g(u)) \mathbf{J}_g(u) \cdot y. \end{aligned}$$

Für die zweite Ableitung von  $(f \circ g)(u)$  gilt durch die Anwendung von Ketten- und Produktregel

$$D^2(f \circ g)(u) = \left[ \underbrace{\sum_{r=1}^n \sum_{s=1}^n \frac{\partial^2 f}{\partial u_r \partial u_s}(g(u)) \frac{\partial g_s}{\partial u_k}(u) \frac{\partial g_r}{\partial u_j}(u)}_{(I)} + \underbrace{\sum_{r=1}^n \frac{\partial f}{\partial u_r}(g(u)) \frac{\partial^2 g_r}{\partial u_k \partial u_j}(u)}_{(II)} : k, j = 1, \dots, n \right].$$

Die Hesse-Matrix  $\mathbf{H}_f(g(x))$  ist ein Element aus  $\mathbb{R}^{n \times n \times m}$  und lässt sich als die folgende Matrixmultiplikation

$$(I) = \left[ \sum_{r=1}^n \sum_{s=1}^n \frac{\partial^2 f}{\partial u_r \partial u_s}(g(u)) \frac{\partial g_s}{\partial u_k}(u) \frac{\partial g_r}{\partial u_j}(u) : k, j = 1, \dots, n \right] \\ = [\mathbf{J}_g(u)^T \mathbf{H}_{f,(:, :, l)}(g(u)) \mathbf{J}_g(u) : l = 1, \dots, m]$$

in Abhängigkeit von  $l = 1, \dots, m$  darstellen und wie folgt formalisieren. Zu einem Tensor  $\mathbf{T} \in \mathbb{R}^{n \times m \times p}$  wird die Matrix  $\mathbf{T}^\dagger$  definiert, sodass

$$\mathbf{T} = \begin{pmatrix} \begin{pmatrix} T_{1,1,1} \\ \vdots \\ T_{1,1,o} \end{pmatrix} & \begin{pmatrix} T_{1,2,1} \\ \vdots \\ T_{1,2,o} \end{pmatrix} & \dots & \begin{pmatrix} T_{1,m,1} \\ \vdots \\ T_{1,m,o} \end{pmatrix} \\ \begin{pmatrix} T_{2,1,1} \\ \vdots \\ T_{2,1,o} \end{pmatrix} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \begin{pmatrix} T_{n,1,1} \\ \vdots \\ T_{n,1,o} \end{pmatrix} & \dots & \dots & \begin{pmatrix} T_{n,m,1} \\ \vdots \\ T_{n,m,o} \end{pmatrix} \end{pmatrix} \Rightarrow \mathbf{T}^\dagger = \begin{pmatrix} T_{1,1,1} & \dots & T_{1,m,1} \\ \vdots & \ddots & \vdots \\ T_{n,m,1} & \dots & T_{n,m,1} \\ T_{1,1,2} & \dots & T_{1,m,2} \\ \vdots & \ddots & \vdots \\ T_{n,m,2} & \dots & T_{n,m,2} \\ \vdots & \ddots & \vdots \\ T_{1,1,o} & \dots & T_{1,m,o} \\ \vdots & \ddots & \vdots \\ T_{n,m,o} & \dots & T_{n,m,o} \end{pmatrix}$$

gilt.  $\dagger$  ist damit ein Operator, der Tensoren matrifiziert. Für weiterführende Informationen bzgl. der Umsortierung von Tensoren in Matrizen sei auf [32] verwiesen. Durch die entsprechende Anwendung des Kronecker-Produkts kann man nun den Ausdruck (I) als reines Matrix-Produkt

$$(I)^\dagger = (\mathbf{I}_m \otimes \mathbf{J}_g(x)^T) \mathbf{H}_f^\dagger(g(x)) \mathbf{J}_g(x)$$

darstellen.

Um den zweiten Summanden

$$(II) = \left[ \sum_{r=1}^n \frac{\partial f}{\partial u_r}(g(u)) \frac{\partial^2 g_r}{\partial u_k \partial u_j}(u) : k, j = 1, \dots, n \right]$$

ebenfalls als Produkt von Matrizen darstellen zu können, bedarf es zunächst einer Darstellung des Tensors  $\frac{\partial^2 g}{\partial x_k \partial x_j}$  als Matrix. Hierzu wird ebenfalls der Operator  $\dagger$  verwendet. An dieser Stelle

steht die Einheitsmatrix  $\mathbf{I}_n$  des Kronecker-Produkts mit der Jakobi-Matrix  $\mathbf{J}_g$  allerdings auf der rechten Seite, um analog zur Umformung von (I) die Dimensionen des Bildraums  $f$  (hier  $m$ ) übereinander zu stapeln. Damit kann man (II) mit

$$(II)^\dagger = (\mathbf{J}_f(x) \otimes \mathbf{I}_n) \mathbf{H}_g^\dagger(x)$$

umformen. Der Schlüssel zu einer Darstellung von  $D^2(f \circ g)(x)$  als Matrixmultiplikation liegt in der Wahl eines geeigneten Operators, der die Tensoren in entsprechende Matrizen stapelt, und der Anwendung von Kronecker-Produkten in Verbindung mit Einheitsmatrizen der jeweiligen Dimension. Damit ergibt sich die Darstellung

$$\begin{aligned} D^2(f \circ g)(x) &= \left[ \underbrace{\sum_{r=1}^n \sum_{s=1}^n \frac{\partial^2 f}{\partial x_r \partial x_s}(g(x)) \frac{\partial g_s}{\partial x_k}(x) \frac{\partial g_r}{\partial x_j}(x)}_{(I)} + \underbrace{\sum_{r=1}^n \frac{\partial f}{\partial x_r}(g(x)) \frac{\partial^2 g_r}{\partial x_k \partial x_j}(x)}_{(II)} : k, j = 1, \dots, n \right] \\ &= (\mathbf{I}_m \otimes \mathbf{J}_g(x)^T) \mathbf{H}_f^\dagger(g(x)) \mathbf{J}_g(x) + (\mathbf{J}_f(g(x)) \otimes \mathbf{I}_n) \mathbf{H}_g^\dagger(x). \end{aligned}$$

Möchte man die zweite Richtungsableitung für zwei Richtungen  $y, z \in \mathbb{R}^n, \|y\|_2 = \|z\|_2 = 1$  berechnen, ergibt sich mit der Herleitung obiger Formel

$$(D_z(D_y(f \circ g))(x))^\dagger = (\mathbf{I}_m \otimes z^T \mathbf{J}_g(x)^T) \mathbf{H}_f^\dagger(g(x)) \mathbf{J}_g(x)y + \mathbf{J}_f(g(x)) (z^T \otimes \mathbf{I}_n) \mathbf{H}_g^\dagger(x)y. \quad (3.3)$$

Diese allgemeine Betrachtung und die entsprechend eingeführte Notation ermöglichen es nun, Ableitungen der Form  $D(G \circ \varphi)(u, v)$  und  $D^2(G \circ \varphi)(u, v)$  kompakt darzustellen und damit die Stetigkeit von Flächenübergängen zu definieren.

**Definition 3.15.** Zwei parametrische Flächen  $F, G$  sind in einem Punkt  $G^n$ -stetig, falls eine Reparametrisierung  $\varphi$  (vgl. Definition 2.17) existiert, sodass

$$D^k F(u, v) = D^k(G \circ \varphi)(u, v), \quad k = 0, \dots, n$$

gilt. Existiert für jeden Punkt einer gemeinsamen Kante eine gültige Reparametrisierung, so heißt der gesamte Flächenübergang  $G^n$ -stetig.

Das nächste Beispiel verdeutlicht den Unterschied zwischen  $C^1$ - und  $G^1$ -Stetigkeit an einem Punkt eines Flächenübergangs.

**Beispiel 3.16.** Sei ein  $G^0$ -stetiger Flächenübergang zweier Parametrisierungen  $F, G$  gegeben, beide Flächen besitzen also eine gemeinsame Kante. Betrachtet man nun die  $C^1/G^1$ -Stetigkeit an einem einzelnen Punkt der gemeinsamen Kante, so ergeben sich die in Abbildung 3.7. dargestellten zwei Forderungen.

Anhand dieses einfachen Beispiels wird bereits deutlich, dass die Forderung nach  $G$ -Stetigkeit im Gegensatz zur  $C$ -Stetigkeit gewisse Freiheitsgrade zulässt. In diesem Fall müssen die parti-



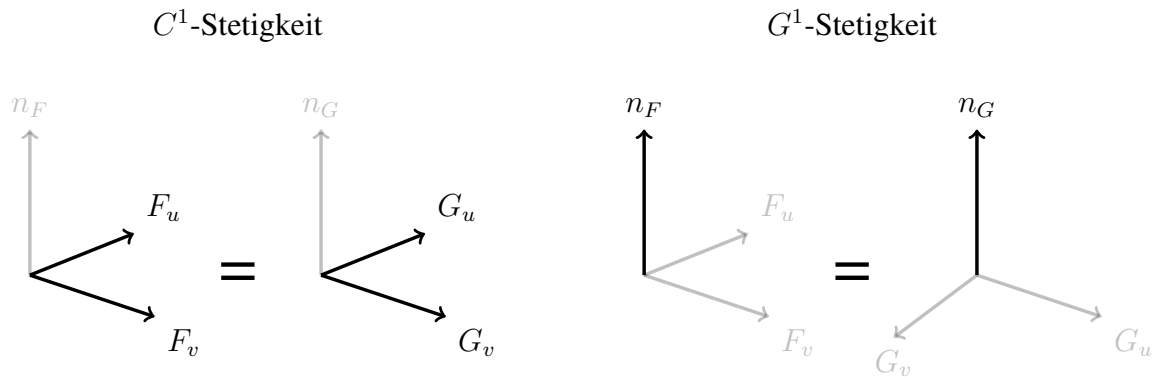


Abb. 3.7: Während auf der linken Seite für  $C^1$ -Stetigkeit in einem Punkt sowohl Länge als auch Richtung der beiden partiellen Ableitungsvektoren übereinstimmen müssen, ist rechts zu erkennen, dass es für die  $G^1$ -Stetigkeit ausreicht, wenn die beiden Flächennormalen an einem gemeinsamen Punkt kollinear sind, sofern ein  $C^0$ -Kontakt besteht.

ellen Ableitungen erster Ordnung  $F_u, F_v$  und  $G_u, G_v$  paarweise linear unabhängig sein und in einer Ebene liegen, während im Falle der parametrischen Stetigkeit explizit die Gleichheit der partiellen Ableitungen gefordert ist.

### 3.2.1 $G^0$ -Stetigkeit für Flächenparametrisierungen

Ist ein Flächenübergang  $G^0$ -stetig, so besitzen beide parametrischen Flächen eine gemeinsame Kante und es existiert eine Reparametrisierung  $\varphi$ , sodass für jeden Punkt auf dieser Kante

$$F(u, v) = (G \circ \varphi)(u, v)$$

gilt. Eine solche Reparametrisierung ist im Gegensatz zum Kurvenfall nicht univariat sondern bivariat. Im Flächenfall wird nicht explizit zwischen  $G^0$ - und  $C^0$ -Stetigkeit unterschieden, da dies optisch keinen Unterschied macht und den Parameterbereich nur transliert. Deutlich interessanter ist die Entwicklung im  $G^1$ -Fall, die im folgenden Abschnitt behandelt wird.

### 3.2.2 $G^1$ -Stetigkeit für Flächenparametrisierungen

Besitzen zwei parametrische Flächen  $F, G$  eine gemeinsame Randkurve ( $C^0$ -Stetigkeit) und soll der Flächenübergang darüber hinaus  $G^1$ -stetig sein, muss für eine Umgebung der gemeinsamen Randkurve eine Reparametrisierung  $\varphi$  existieren, sodass

$$D^1 F(u, v) = D^1 (G \circ \varphi)(u, v)$$

$$\mathbf{J}_F(u, v) = \mathbf{J}_G(\varphi(u, v)) \underbrace{\mathbf{X}(u, v)}_{D^1 \varphi(u, v)}$$

gilt. Die erste Ableitung der Reparametrisierung  $\varphi$  ist in diesem Fall (im Gegensatz zu den Kurvenübergängen) kein Skalar mehr, sondern eine Jakobi-Matrix. Mithilfe der eingeführten Notation der allgemeinen Ableitungsregeln für multivariate verkettete Funktionen, lässt sich im Folgenden die  $G^2$ -Stetigkeit für Flächen übersichtlich betrachten.

### 3.2.3 $G^2$ -Stetigkeit für Flächenparametrisierungen

Sei ein  $G^1$ -stetiger Flächenübergang von  $F, G$  gegeben. Damit die Flächen entlang der gemeinsamen Kante  $G^2$ -stetig sind, muss zusätzlich für jeden Punkt auf der Kante noch

$D^2F(u, v) = D^2(G \circ \varphi)(u, v)$  oder

$$\mathbf{H}_F^\dagger(u, v) = \left( \mathbf{I}_3 \otimes \underbrace{\mathbf{X}(u, v)^T}_{D^1\varphi(u, v)} \right) \underbrace{\mathbf{H}_G^\dagger(\varphi(u, v))}_{D^2G(\varphi(u, v))} \underbrace{\mathbf{X}(u, v)}_{D^1\varphi(u, v)} - \left( \underbrace{\mathbf{J}_G(\varphi(u, v))}_{D^1G(\varphi(u, v))} \otimes \mathbf{I}_2 \right) \underbrace{\mathbf{Y}^\dagger(u, v)}_{D^2\varphi(u, v)} \quad (3.4)$$

gelten, wobei der Tensor  $D^2\varphi(u, v) \in \mathbb{R}^{2 \times 2 \times 2}$  in obiger Gleichung die Matrix-Form

$$\mathbf{Y}^\dagger(u, v) = \begin{pmatrix} y_{uu,u} & y_{uv,u} \\ y_{uv,u} & y_{vv,u} \\ y_{uu,v} & y_{uv,v} \\ y_{uv,v} & y_{vv,v} \end{pmatrix}$$

besitzt. Die Struktur der Formel lässt sich mit den gegebenen Dimensionen geometrisch anschaulich erklären und visualisieren. Betrachtet wird zunächst der erste Ausdruck

$$(\mathbf{I}_3 \otimes \mathbf{X}(u, v)^T) \mathbf{H}_G^\dagger(\varphi(u, v)) \mathbf{X}(u, v). \quad (3.5)$$

Der  $\dagger$ -Operator stapelt die  $x$ -,  $y$ - und  $z$ -Komponenten von  $\mathbf{H}_F(\varphi(u, v))$  in einer Matrix  $\mathbf{H}_F^\dagger(\varphi(u, v))$  übereinander, während das Kronecker-Produkt  $(\mathbf{I}_3 \otimes \mathbf{X}(u, v)^T)$  dafür sorgt, dass die Reparametrisierung mit  $D^1\varphi(u, v) = \mathbf{X}(u, v)$  auf jede einzelne Komponente angewandt wird (siehe Abbildung 3.8).

Der zweite Teil der Gleichung (3.4)

$$(\mathbf{J}_G(\varphi(u, v)) \otimes \mathbf{I}_2) \mathbf{Y}^\dagger(u, v)$$

wird zunächst aus reiner Tensorsicht betrachtet. Seien hierzu die Jakobi-Matrix  $\mathbf{J}_G(\varphi(u, v))$  und  $D^2\varphi(u, v) = \mathbf{Y}(u, v)$  für die folgende Veranschaulichung als Tensoren mit den Dimensionen

$$\begin{aligned} \mathbf{J}_G(\varphi(u, v)) &\in \mathbb{R}^{1 \times 2 \times 3} \\ \mathbf{Y}(u, v) &\in \mathbb{R}^{2 \times 2 \times 2} \end{aligned}$$

dargestellt. Das Produkt beider Terme lässt sich graphisch wie folgt darstellen, wobei immer eine Zeile der Jakobi-Matrix mit zwei untereinanderliegenden Werten von  $\mathbf{Y}(u, v)$  multipliziert wird,

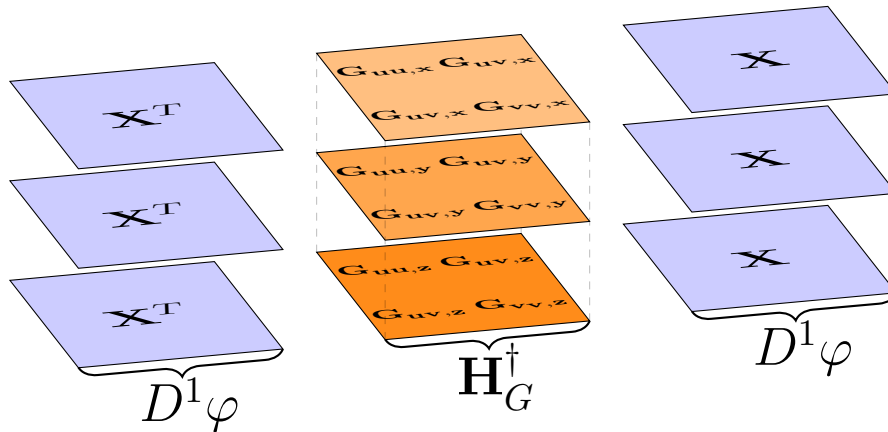


Abb. 3.8: Graphische Darstellung von  $(\mathbf{I}_3 \otimes \mathbf{X}(u, v)^T) \mathbf{H}_G^\dagger(\varphi(u, v)) \mathbf{X}(u, v)$

also  $x_u \cdot y_1 + x_v \cdot y_2$ , usw.:

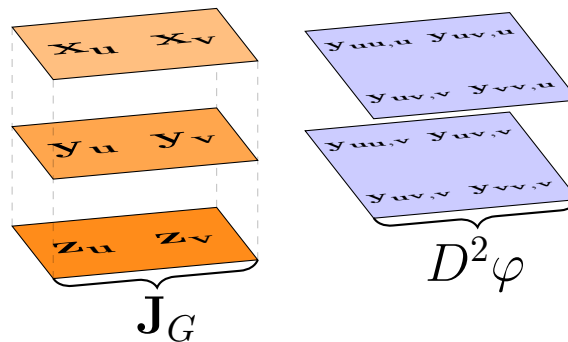


Abb. 3.9: Graphische Darstellung von  $(\mathbf{J}_G(\varphi(u, v)) \otimes \mathbf{I}_2) \mathbf{Y}^\dagger(u, v)$

Durch die Kopie und Umordnung der Jacobi-Matrix mit  $(\mathbf{J}_G(\varphi(u, v)) \otimes \mathbf{I}_2)$  und der Betrachtung von  $D^2\varphi(u, v) = \mathbf{Y}(u, v)$  als  $(4 \times 2)$ -Matrix, lässt sich der Term als reine Matrixmultiplikation

$$(\mathbf{J}_G(\varphi(u, v)) \otimes \mathbf{I}_2) \mathbf{Y}^\dagger(u, v) = \begin{pmatrix} F_{u,x} & 0 & F_{v,x} & 0 \\ 0 & F_{u,x} & 0 & F_{v,x} \\ F_{u,y} & 0 & F_{v,y} & 0 \\ 0 & F_{u,y} & 0 & F_{v,y} \\ F_{u,z} & 0 & F_{v,z} & 0 \\ 0 & F_{u,z} & 0 & F_{v,z} \end{pmatrix} \begin{pmatrix} y_{uu,u} & y_{uv,u} \\ y_{uv,v} & y_{vv,u} \\ y_{uu,v} & y_{uv,v} \\ y_{uv,v} & y_{vv,v} \end{pmatrix}$$

realisieren. Damit ergibt sich für beide Teile von Gleichung (3.4) als Resultat der Matrixmulti-

pplikationen eine  $(6 \times 2)$ -Matrix der Form wie sie auch  $\mathbf{H}_G^\dagger(\varphi(u, v))$  besitzt

$$\begin{pmatrix} \tilde{x}_1 & \tilde{x}_2 \\ \tilde{x}_3 & \tilde{x}_4 \\ \tilde{y}_1 & \tilde{y}_2 \\ \tilde{y}_3 & \tilde{y}_4 \\ \tilde{z}_1 & \tilde{z}_2 \\ \tilde{z}_3 & \tilde{z}_4 \end{pmatrix}.$$

Mit diesen Informationen ist es nun möglich, ein Maß für die Abweichung eines Flächenübergangs bezüglich der  $G^{0,1,2}$ -Stetigkeit zu definieren, das numerisch stabil berechnet werden kann.

### 3.3 $G^{0,1,2}$ -Stetigkeitstest

Das Prüfen von Stetigkeiten an einem Punkt eines Flächenübergangs ist ein iteratives Verfahren beginnend mit der  $G^0$ -Stetigkeit. Der Grund hierfür liegt in der geometrischen Bedeutung der einzelnen Stetigkeiten, denn die Tangentialebenen zweier Flächen an zwei Punkten zu untersuchen, die sich nicht treffen, gibt keinerlei Mehrwert. Ebenso macht die Betrachtung von Krümmungen in verschiedenen Tangentialrichtungen keinen Sinn, wenn die zwei betrachteten Flächenpunkte keine gemeinsame Tangentialebene besitzen. Alle im Folgenden vorgestellten Verfahren sind punktweise anzuwenden, d.h. um einen Flächenübergang zu prüfen, müssen alle Tests für jedes Paar an Betrachtungspunkten der gemeinsamen Kante wiederholt werden.

#### 3.3.1 $G^0$ -Stetigkeit

Im ersten Schritt muss getestet werden, ob die Betrachtungspunkte beider parametrischer Flächen  $F, G$  übereinstimmen, also ob

$$F(u, v) = G(\varphi(u, v))$$

gilt. Ein Fehlermaß erhält man durch die Anwendung der euklidischen Norm

$$\|F(u, v) - G(\varphi(u, v))\|_2, \quad (3.6)$$

welches gegen die Toleranz  $\tau$  getestet werden soll. Obwohl die Gleichung (3.6) auf den ersten Blick keine große Bedeutung zu haben scheint, wird ihre Wichtigkeit mit nachfolgendem Beispiel klarer.

**Beispiel 3.17.** *An die Mantelfläche eines Zylinders wurde in einem CAD-System mit einer polynomialen Tensorprodukt-Fläche (vgl. Definition 2.50) eine Anschlussfläche mit  $G^0$ -Bedingung erzeugt. Ohne die explizite Verwendung von rationalen Tensorprodukt-Flächen (NURBS) [48] ist es nicht möglich, einen Kreis fehlerfrei zu interpolieren oder, mit anderen Worten, in diesem Fall einen fehlerfreien Zylinder-Anschluss zu erzeugen. In Abbildung 3.10 ist ein solcher Über-*

gang zu sehen. Die orange Fläche ist eine Tensorprodukt-Fläche, die an den blauen Zylinder (konstruiert als NURBS-Fläche)  $G^0$ -stetig anschließt. Die schwarzen Kämmen visualisieren den  $G^0$ -Fehler am gemeinsamen Rand.

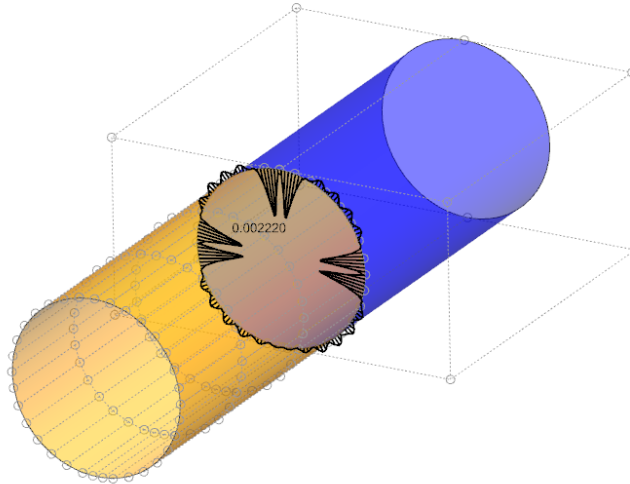


Abb. 3.10: Ein in NX konstruierter  $G^0$ -Übergang einer Tensorprodukt-Fläche (orange) und einem Zylinder (blau) mit Fehlerplot der  $G^0$ -Stetigkeit. Der maximale Fehler beträgt in diesem Fall 0.00222mm.

Ist ein solcher Flächenübergang innerhalb einer gewissen Toleranzschranke  $\tau$  als  $G^0$ -stetig klassifiziert, kann im nächsten Schritt die Tangentenstetigkeit ( $G^1$ -Stetigkeit) geprüft werden.

### 3.3.2 $G^1$ -Stetigkeit

Üblicherweise geben CAD-Systeme die  $G^1$ -Stetigkeit an, indem der Winkel zwischen den Flächennormalen berechnet wird. Ist ein Flächenübergang an einer Stelle  $G^1$ -stetig, so ist der Winkel  $0^\circ$ , da die partiellen Ableitungen dieselbe Tangentialebene aufspannen. Obwohl man durch diese Betrachtung ein sehr anschauliches Fehlermaß erhält, ist die Berechnung

$$\cos(\theta) = \frac{\langle (F_u \times F_v), (G_u \times G_v) \rangle}{\|(F_u \times F_v)\|_2 \cdot \|(G_u \times G_v)\|_2} = \frac{\langle n_F, n_G \rangle}{\|n_F\|_2 \cdot \|n_G\|_2} \quad (3.7)$$

aus numerischer Sicht gegenüber Rundungsfehlern und anderen Störungen (z.B. an fast singulären Punkten) anfällig. In dieser Arbeit wird ein neuer Ansatz zur Prüfung der Stetigkeit an einem Flächenübergang vorgestellt. Anstatt die  $G^1$ -Stetigkeit über den Winkel der Flächennormalen zu bestimmen, kann man auch versuchen, eine Reparametrisierung  $\varphi$  zu finden, sodass  $D^1\varphi(u, v) = \mathbf{X}(u, v)$  die Frobeniusnorm

$$\|\mathbf{J}_F(u, v) - \mathbf{J}_G(\varphi(u, v))\mathbf{X}(u, v)\|_F \quad (3.8)$$

minimiert. Da in dieser Arbeit parametrische Flächen stets regulär sind, besitzen die Jakobi-Matrizen  $\mathbf{J}_F(u, v)$  und  $\mathbf{J}_G(\varphi(u, v))$  beide den maximalen Rang 2. Wählt man die Matrix  $\mathbf{X}(u, v)$  als

$$\mathbf{X}(u, v) = \mathbf{J}_G(\varphi(u, v))^+ \mathbf{J}_F(u, v),$$

wobei  $\mathbf{J}_G(\varphi(u, v))^+$  die *Moore-Penrose Pseudo-Inverse* ist, so minimiert sie den Term aus Gleichung (3.8) (vgl. [28, Kapitel 5]). Die Matrix  $\mathbf{X}(u, v)$  besitzt in diesem Zusammenhang eine wichtige geometrische Bedeutung, da sie für den Fall, dass der Übergang  $G^1$ -stetig ist, bereits die erste Ableitung der Reparametrisierung  $\varphi$  liefert, die notwendig ist, um einen  $C^1$ -stetigen Übergang zu erzeugen.

**Bemerkung 3.18.** *Die Berechnung der Matrix  $\mathbf{X}(u, v)$  durch die Pseudo-Inverse ist nicht symmetrisch. Um die Aussagekraft des Fehlers zu verbessern, ist es nützlich das Verfahren einmal mit  $F$  und einmal mit  $G$  als Ausgangsfläche durchzuführen und das Minimum beider Ergebnisse zu verwenden. Ein weiterer wichtiger Aspekt ist der Quotient der Singulärwerte von  $\mathbf{X}(u, v)$ . Mit dieser numerischen Rangbestimmung lässt sich beurteilen, wie gut das Problem konditioniert ist.*

Bei der Verwendung von modernen CAD-Systemen taucht allerdings häufig das Problem auf, dass der Benutzer an eine bereits existierende (und gutmütig parametrisierte) Fläche eine weitere, durch die Software automatisch generierte, Fläche mit gewissen Stetigkeitsbedingungen „anheften“ möchte. Dadurch entsteht entlang der gemeinsamen Randkurve auf der neuen Fläche eine fast singuläre Parametrisierung, wohingegen der ursprünglichen Fläche eine gute Parametrisierung zugrundeliegt. Es kann also nicht garantiert oder vorausgesetzt werden, dass beide Flächen sinnvoll und miteinander konsistent parametrisiert sind. Um an solchen Übergängen trotzdem mit numerisch stabilen Methoden Aussagen zu treffen, kann folgender Satz verwendet werden: Ist eine der beiden Flächen orthogonal parametrisiert und liegen ihre partiellen Ableitungen normiert vor, so lassen sich mithilfe der Pseudo-Inversen sogar Aussagen über den Winkel zwischen den Flächennormalen treffen, ohne dabei das Kreuzprodukt der partiellen Ableitungen berechnen zu müssen, wie nachfolgender Satz zeigt.

**Satz 3.19.** *Sei ein Flächenübergang zweier parametrischer Flächen  $F, G$  gegeben, die eine gemeinsame Randkurve besitzen, die mindestens  $C^1$ -stetig ist. Sei weiter die Jakobi-Matrix  $\mathbf{J}_F(u^*, v^*)$  ein Orthonormalsystem der Tangentialebene im Punkt  $(u^*, v^*)$  und sei  $(G \circ \varphi)(u^*, v^*)$  am gemeinsamen Punkt nicht degeneriert. Dann ist der Winkel  $\theta$  zwischen den Flächennormalen von  $F$  und  $G$  am Untersuchungspunkt  $(u^*, v^*)$  gegeben durch*

$$\sin(\theta) = \min_{\mathbf{X}} \|\mathbf{J}_F(u^*, v^*) - \mathbf{J}_G(\varphi(u^*, v^*))\mathbf{X}(u^*, v^*)\|_F, \quad \mathbf{X}(u^*, v^*) = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix}. \quad (3.9)$$

**Beweis:** Für den Spezialfall von getrimmten Flächen kann durch die Multiplikation von  $\mathbf{J}_F(u^*, v^*)$  mit einer Orthogonalmatrix  $\mathbf{Q}$  (die invariant bezüglich  $\|\cdot\|_F$  ist) stets dafür gesorgt werden, dass  $F_u(u^*, v^*)$  nach der Multiplikation mit  $\mathbf{Q}$  kollinear zur Ableitung der gemeinsamen Kurve am Untersuchungspunkt ist. In allen anderen Fällen ist die Randkurve bereits eine der beiden Parameterlinien. Da beide Flächen nach Voraussetzung dieselbe Randkurve besitzen und keine der Flächen degeneriert ist, interpolieren sowohl  $F$  als auch  $G$  die gemeinsame  $C^1$ -Kante (und damit

auch ihre Ableitung) und es gibt eine Linearkombination, sodass

$$\mathbf{Q}F_u(u^*, v^*) = \mathbf{Q}\mathbf{J}_G(\varphi(u, v)) \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \quad \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \in \mathbb{R}^2,$$

gilt. Die Lösungen der beiden Minimierungsprobleme

$$\min_{\mathbf{X}} \|\mathbf{J}_F(u^*, v^*) - \mathbf{J}_G(\varphi(u^*, v^*))\mathbf{X}(u^*, v^*)\|_F = \min_{x_{12}, x_{22}} \left\| \mathbf{Q}F_v(u^*, v^*) - \mathbf{Q}\mathbf{J}_G(\varphi(u^*, v^*)) \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix} \right\|_2$$

sind damit äquivalent. Aufgrund der besseren Lesbarkeit werden die Parameter  $(u^*, v^*)$  und  $\varphi(u^*, v^*)$  in den folgenden Beweisschritten verkürzt als  $F^* = F(u^*, v^*)$ ,  $G^* = G(\varphi(u^*, v^*))$ , usw. geschrieben. Der Fehler der Least-Squares Lösung ist orthogonal zum Vektorraum  $V = \mathbf{J}_{\mathbf{Q}G}^* \cdot \mathbb{R}^2$  (siehe [25, Kapitel 2]). Es gilt

$$0 = \left\langle \mathbf{Q}F_v^* - \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix}, \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \right\rangle, \quad \alpha, \beta \in \mathbb{R},$$

und insbesondere für  $\alpha = x_{12}$ ,  $\beta = x_{22}$

$$0 = \left\langle \mathbf{Q}F_v^* - \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix}, \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix} \right\rangle. \quad (3.10)$$

Der Winkel zwischen den beiden Vektoren  $\mathbf{Q}F_v^*$  und  $\mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix}$  lässt sich aufgrund von  $\|F_v^*\|_2 = \|\mathbf{Q}F_v^*\|_2 = 1$  mithilfe des Sinussatzes wie folgt berechnen

$$\sin(\gamma) = \frac{\left\| \mathbf{Q}F_v^* - \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix} \right\|_2}{\|\mathbf{Q}F_v^*\|_2} = \left\| \mathbf{Q}F_v^* - \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix} \right\|_2.$$

Es bleibt zu zeigen, dass  $\gamma$  und der Winkel  $\theta$  zwischen den Flächennormalen identisch sind.

Da  $F_u^*$  und  $F_v^*$  ein Orthonormalsystem von  $T_F^*$  bilden, stehen sie aufeinander senkrecht und es gilt

$$0 = \langle F_u^*, F_v^* \rangle = \langle \mathbf{Q}F_u^*, \mathbf{Q}F_v^* \rangle.$$

Die Projektion von  $\mathbf{Q}F_v^*$  durch Lösen des Least-Squares Problems ist orthogonal zu  $\mathbf{Q}F_u^*$ , denn

es gilt

$$\begin{aligned}
0 &= \langle F_u^*, F_v^* \rangle \\
&= \langle \mathbf{Q}F_u^*, \mathbf{Q}F_v^* \rangle \\
&= \left\langle \mathbf{Q}F_u^*, \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix} + \mathbf{Q}F_v^* - \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix} \right\rangle \\
&= \left\langle \mathbf{Q}F_u^*, \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix} \right\rangle + \underbrace{\left\langle \mathbf{Q}F_u^*, \mathbf{Q}F_v^* - \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix} \right\rangle}_{=0},
\end{aligned}$$

nach (vgl. (3.10)). Für den Winkel  $\gamma$  gilt

$$\cos(\gamma) = \frac{\left\langle \mathbf{Q}F_v^*, \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix} \right\rangle}{\|\mathbf{Q}F_v^*\|_2 \cdot \left\| \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix} \right\|_2}.$$

Der Winkel  $\theta$  zwischen den Flächennormalen kann analog berechnet werden:

$$\cos(\theta) = \frac{\langle n_F^*, n_G^* \rangle}{\|n_F^*\|_2 \cdot \|n_G^*\|_2}.$$

Die beiden Flächennormalen von  $F$  und  $G$  lassen sich durch die Kreuzprodukte

$$\begin{aligned}
n_F^* &= (\mathbf{Q}F_u^* \times \mathbf{Q}F_v^*), \\
n_G^* &= \left( \mathbf{Q}F_u^* \times \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix} \right)
\end{aligned}$$

darstellen und in obiger Formel einsetzen und vereinfachen ([26, Seite 77])

$$\begin{aligned}
\cos(\theta) &= \frac{\left\langle (\mathbf{Q}F_u^* \times \mathbf{Q}F_v^*), \left( \mathbf{Q}F_u^* \times \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix} \right) \right\rangle}{\|\mathbf{Q}F_u^* \times \mathbf{Q}F_v^*\|_2 \cdot \left\| \mathbf{Q}F_u^* \times \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix} \right\|_2} \\
&= \frac{\langle \mathbf{Q}F_u^*, \mathbf{Q}F_u^* \rangle \cdot \left\langle \mathbf{Q}F_v^*, \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix} \right\rangle - \langle \mathbf{Q}F_v^*, \mathbf{Q}F_u^* \rangle \cdot \left\langle \mathbf{Q}F_u^*, \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix} \right\rangle}{\|\mathbf{Q}F_u^*\|_2 \cdot \|\mathbf{Q}F_v^*\|_2 \cdot \|\mathbf{Q}F_u^*\|_2 \cdot \left\| \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix} \right\|_2} \\
&= \frac{\left\langle \mathbf{Q}F_v^*, \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix} \right\rangle}{\|\mathbf{Q}F_u^*\|_2 \cdot \left\| \mathbf{J}_{\mathbf{Q}G}^* \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix} \right\|_2} = \cos(\gamma).
\end{aligned}$$



Für den Fall

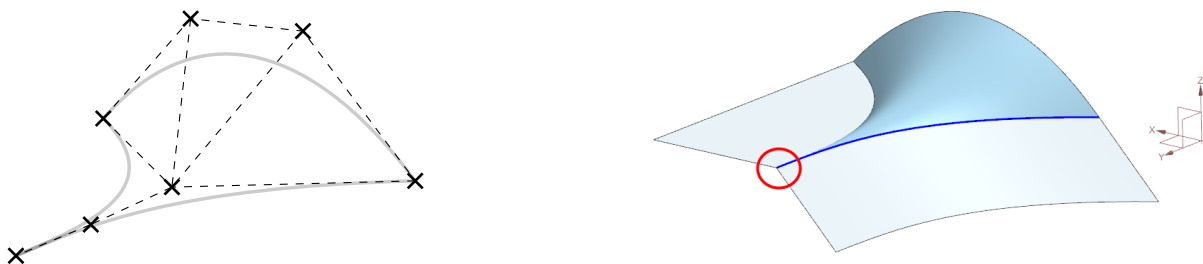
$$0 \leq \theta, \gamma \leq \pi$$

folgt damit die Gleichheit beider Winkel. Das stellt keine Einschränkung dar, da bei der Betrachtung des  $G^1$ -Fehlers stets der kleinere Winkel zwischen den Flächennormalen betrachtet wird.

□

Die Angabe des Winkels zwischen zwei Flächennormalen zur Prüfung und Klassifizierung eines Flächenübergangs bzgl. der  $G^1$ -Stetigkeit hat sich in modernen CAD-Systemen etabliert. Obwohl die Berechnungsmethoden effizient implementiert sind, kommt es insbesondere in der Nähe von Singularitäten (oder sehr schlechten Parametrisierungen) allerdings manchmal zu unerwünschten Effekten. Im nachfolgenden Beispiel wird der Vorteil der  $G^1$ -Stetigkeitsprüfung mithilfe des Least-Squares Ansatzes im Gegensatz zu einem konventionellen CAD-Verfahren veranschaulicht.

**Beispiel 3.20.** Gegeben sei eine kubische Tensorprodukt-Fläche  $F$  mit einem wie in Abbildung 3.11 dargestellten  $(4 \times 4)$ -Kontrollpolygon. Um die numerische Stabilität beider  $G^1$ -Testverfahren zu prüfen, wurde eine fast singuläre Situation erzeugt, in dem die Punkte der Reihen 2,3 und 4 jeweils sehr dicht ( $\approx 1\mu\text{m}$ ) liegen (zum Vergleich: Die dunkelblaue Kurve in Abbildung 3.11 besitzt ca. eine Länge von 36mm).



(a) Kontrollpolygon der Tensorprodukt-Fläche  $F$  mit fast singulär zulaufender Spitze.

(b) CAD-Modell mit blau markierter Kante, entlang derer die  $G^1$ -Stetigkeit überprüft werden soll.

Abb. 3.11: Die Tensorprodukt-Fläche  $F$  wurde derart konstruiert, dass bis auf zwei Punkte des Kontrollpolygons alle in der  $xy$ -Ebene liegen und die gebogene Spline-Kante (blau) fast auf einen Punkt zusammenfällt. Der fast singuläre Punkt, an dem die  $G^1$ -Stetigkeit getestet werden soll, ist rot hervorgehoben.

Die zweite Fläche  $G$  wurde erzeugt, in dem die blaue Randkurve um 10mm linear in Richtung des Vektors  $v = (-1, 0, -1)^T$  extrudiert wurde. Aufgrund der Konstruktionsvorschriften von  $F$  ergibt sich an der singulären Spitze ( $u^*$ ) die normierte Flächennormale

$$\hat{n}_F(u^*) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

Zusammen mit dem Extrusionsvektor ergibt sich damit am fast singulären Punkt ein Winkel von exakt  $45^\circ$  zwischen den beiden Flächennormalen. In der Abbildung 3.12 links ist das aktuelle, in

*NX* implementierte Standardverfahren zur Prüfung auf  $G^1$ -Stetigkeit entlang eines Flächenübergangs dargestellt. Bei näherer Betrachtung fällt auf, dass der  $G^1$ -Fehler nicht monoton fällt und sogar stellenweise leicht verrauscht ist. Am Endpunkt selbst wird der Winkel mit  $45.0017^\circ$  und damit einem Fehler in der dritten Nachkommastelle angegeben. Rechts wurde zur Bestimmung des Winkels zwischen den Flächennormalen der hier vorgestellte Ansatz verwendet. Wie zu erwarten reagiert die Berechnung mit der Pseudo-Inversen stabiler und liefert selbst bei fast singulären Punkten die korrekte Lösung.

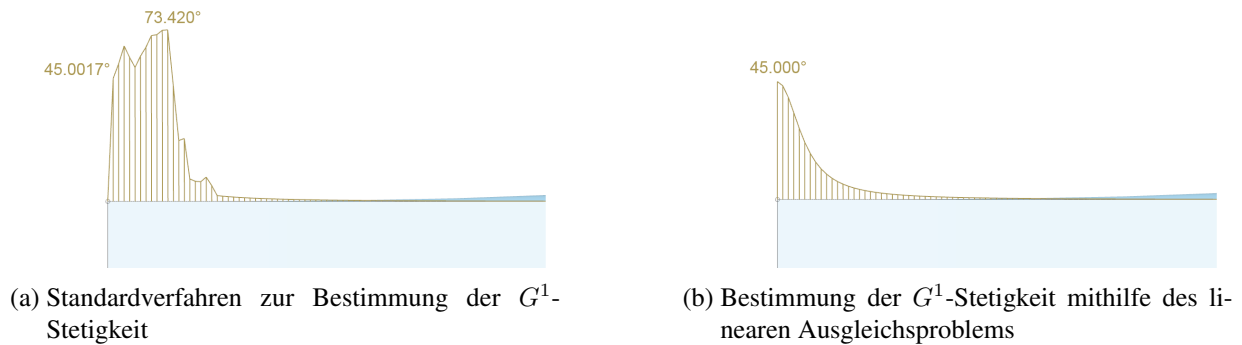


Abb. 3.12: Vergleich der  $G^1$ -Stetigkeitsprüfung im Standardfall (NX) und als Least Squares Lösung (MATLAB) am fast singulären Punkt.

Ist der Übergang an einem Punkt der gemeinsamen Randkurve  $G^1$ -stetig, so gibt es dafür eine Reparametrisierung  $\varphi$  mit  $D\varphi(u, v) = \mathbf{X}(u, v)$ , sodass die Bedingung

$$\|\mathbf{J}_F(u, v) - \mathbf{J}_G(\varphi(u, v))\mathbf{X}(u, v)\|_F \leq \tau,$$

innerhalb einer gewählten Toleranzgrenze  $\tau$ , erfüllt ist.

**Bemerkung 3.21.** *Stimmen die Normalen beider Flächen an einem Punkt überein, existiert eine Reparametrisierung  $\mathbf{X}(u, v)$ , die obige Bedingung erfüllt. Besitzt  $\mathbf{X}(u, v)$  zusätzlich die Form*

$$\mathbf{X}(u, v) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

so ist der Übergang sogar  $C^1$ -stetig.

War die Prüfung auf  $G^1$ -Stetigkeit erfolgreich und hat man eine Reparametrisierung  $\mathbf{X}(u, v)$  gefunden, kann man mit der Prüfung der  $G^2$ -Stetigkeit fortfahren.

### 3.3.3 $G^2$ -Stetigkeit

Sowohl der euklidische Abstand bei der  $G^0$ -Betrachtung als auch der Winkel zwischen den Flächennormalen bei der  $G^1$ -Betrachtung liefern sehr anschauliche Fehlermaße. Im letzten Abschnitt über die Tangentenstetigkeit wurde allerdings schon deutlich, dass die Flächennormale

durch das Kreuzprodukt beider partieller Ableitungen erster Ordnung entsteht und die Problematik im Vergleich zum univariaten Kurvenfall deutlich komplexer ist. Dieser Sachverhalt spiegelt sich auch in der Tatsache wider, dass die Hesse-Matrix ein Tensor ist und ad hoc kein greifbares Fehlermaß existiert. Der Grund dafür ist in Kapitel 2.1, Abbildung 2.6, gut zu erkennen, denn die Krümmung von Kurven auf Flächen hängt von der gewählten Tangentialrichtung ab und wird im nächsten Beispiel veranschaulicht.

**Beispiel 3.22.** Seien zwei parametrische Flächen der Form

$$F(u, v) = \begin{pmatrix} u \\ v \\ -uv^2 \end{pmatrix}, \quad u \in [-1, 0], v \in [-1, 1],$$

$$G(u, v) = \begin{pmatrix} u \\ v \\ uv \end{pmatrix}, \quad u \in [0, 1], v \in [-1, 1],$$

wie in Abbildung 3.13 dargestellt, gegeben.

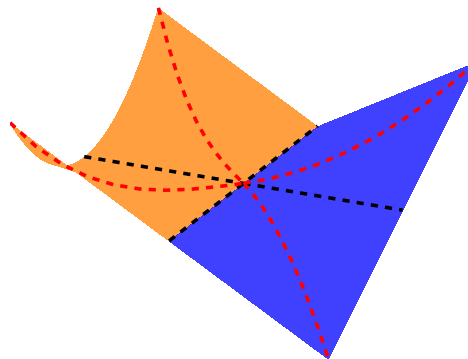


Abb. 3.13: Flächenübergang, der am Untersuchungspunkt zwar  $G^1$ -stetig ist, aber nur in zwei Tangentialrichtungen (schwarz)  $G^2$ -stetig ist und in allen anderen Richtung nicht krümmungstetig ist.

Offensichtlich sind beide Flächen im Punkt  $(0, 0)$   $C^0$ -stetig

$$F(0, 0) = G(0, 0).$$

Um die  $G^1$ -Stetigkeit zu prüfen, werden zunächst die Jakobi-Matrizen aufgestellt

$$\mathbf{J}_F(u, v) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -v^2 & -2uv \end{pmatrix}, \quad \mathbf{J}_G(u, v) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ v & u \end{pmatrix}$$

und anschließend im Punkt  $(0, 0)$  mit dem  $G^1$ -Test geprüft:

$$\begin{aligned} \|\mathbf{J}_F(0, 0) - \mathbf{J}_G(0, 0)\mathbf{X}\|_F &\leq \tau \\ \left\| \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \mathbf{X} \right\|_F &\leq \tau. \end{aligned}$$

Die gesuchte Ableitung der Reparametrisierung ist in diesem Fall die Einheitsmatrix selbst, der Übergang ist also an dieser Stelle sogar  $C^1$ -stetig. Um die  $G^2$ -Stetigkeit zu prüfen, werden zunächst die Hesse-Matrizen beider Flächen berechnet

$$\mathbf{H}_F(u, v) = \begin{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -2v \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ -2v \\ 0 \\ 0 \end{pmatrix} \end{pmatrix}, \quad \mathbf{H}_G(u, v) = \begin{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} & \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \end{pmatrix}.$$

Anhand der Hesse-Matrizen ist zu erkennen, dass beide Flächen entlang ihrer Parameterrichtung  $u$  keinerlei Krümmung besitzen. Lediglich in der gemischten partiellen Ableitung zweiter Ordnung und in  $F_{vv}$  treten parameterabhängige Vektoren ungleich 0 auf. Die Situation im Punkt  $(0, 0)$  ist also richtungsabhängig.

Betrachtet man die Krümmung am Übergang in eine der beiden Parameterrichtungen ist der Übergang offensichtlich  $G^2$ -stetig, wählt man hingegen eine Kurve, die nicht auf den Parameterlinien verläuft, ist der Übergang nicht  $G^2$ -stetig. In Abbildung 3.13 sind die zwei Richtungen, in denen der Übergang  $G^2$ -stetig ist, schwarz dargestellt. In allen anderen Richtungen sind die Flächen am gemeinsamen Punkt nur tangentialstetig. Zwei Kurven, die in 0 nicht  $G^2$ -stetig sind, sind exemplarisch rot dargestellt.

Wie das Beispiel 3.22 gezeigt hat, ist es schwierig den  $G^2$ -Fehler als skalaren Wert auszudrücken. Heutige CAD-Systeme nutzen hierfür im Kurvenfall z.B.

$$\kappa_{G^2Error} = 2 \frac{\|\kappa_f - \kappa_g\|_2}{\|\kappa_f\|_2 + \|\kappa_g\|_2}.$$

Um auch im Flächenfall eine Aussage machen zu können, werden die Werte  $\kappa_f$  und  $\kappa_g$  mit den entsprechenden Krümmungsbegriffen für Flächen (Gauß-Krümmung, mittlere Krümmung, vgl. 2.26) ersetzt. Diese Werte sind allerdings nur bedingt aussagekräftig, wie nachfolgende Abbildung 3.14 zeigt:

Der  $G^2$ -Fehler bezogen auf die mittlere Krümmung ist entlang der gesamten Kante stets 0, während der Fehler in den beiden anderen Fällen an exakt der Stelle maximal ist, an der es sogar zwei  $G^2$ -stetige Richtungen gibt. Ähnlich zum  $G^1$ -Stetigkeitstest ist es auch hier möglich mit numerisch stabilen Methoden eine passende Reparametrisierung zu finden, anstatt die Hauptkrümmungsrichtungen zu bestimmen. Wie bereits bei der allgemeinen Betrachtung von  $C^n/G^n$ -stetigen Flächenübergängen (Kapitel 3.2) erwähnt, handelt es sich bei Hesse-Matrizen in diesem Fall bereits um Tensoren aus  $\mathbb{R}^{2 \times 2 \times 3}$ , die allerdings als Matrizen dargestellt werden können (vgl.

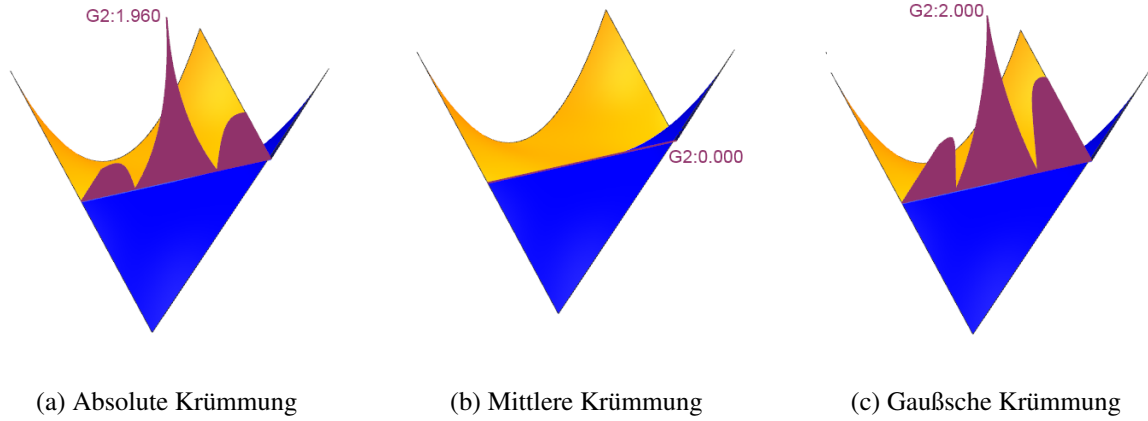


Abb. 3.14: Für zwei gegebene Flächen (gelb und blau) wurden in NX verschiedene  $G^2$ -Krümmungsplots erstellt, die unterschiedliche Fehlerausprägungen zeigen. Der jeweilige Fehler entlang der gesamten Kante ist in lila dargestellt. Es ist zu erkennen, dass sowohl die Absolute Krümmung als auch die Gaußsche Krümmung am Mittelpunkt der Kante maximal sind obwohl es dort sogar zwei Tangentialrichtungen gibt, in denen der Übergang  $G^2$ -stetig ist. Weiter fällt auf, dass der Fehler der Mittleren Krümmung entlang der gesamten Kante 0 ist.

Definition 2.15). Die bereits hergeleitete  $G^2$ -Stetigkeitsbedingung ((3.4))

$$\min_{\mathbf{Y}} \left\| \mathbf{H}_F^\dagger(u, v) - (\mathbf{I}_3 \otimes \mathbf{X}(u, v)^T) \mathbf{H}_G^\dagger(\varphi(u, v)) \mathbf{X}(u, v) - (\mathbf{J}_G(\varphi(u, v)) \otimes \mathbf{I}_2) \mathbf{Y}^\dagger(u, v) \right\|_F \leq \tau$$

kann wieder mithilfe der Least-Squares Methode überprüft werden. Sei dafür

$$\tilde{\mathbf{H}}_{F-G}^\dagger(u, v) := \mathbf{H}_F^\dagger(u, v) - (\mathbf{I}_3 \otimes \mathbf{X}(u, v)^T) \mathbf{H}_G^\dagger(\varphi(u, v)) \mathbf{X}(u, v) \in \mathbb{R}^{6 \times 2}$$

die Differenz der Hesse-Matrix von  $F$  und der reparametrisierten Hesse-Matrix von  $G$ . Bezeichne weiter

$$\tilde{\mathbf{J}}_G(\varphi(u, v)) := (\mathbf{J}_G(\varphi(u, v)) \otimes \mathbf{I}_2) \in \mathbb{R}^{6 \times 4}$$

das Kronecker-Produkt aus Jakobi-Matrix von  $G$  und der Einheitsmatrix  $\mathbf{I}_2$ . Das zu lösende Minimierungsproblem besitzt dann die Form

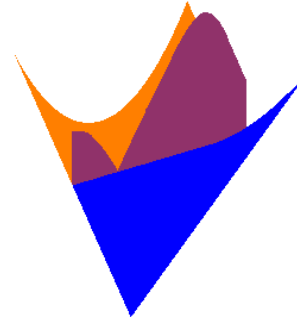
$$\min_{\mathbf{Y}} \left\| \tilde{\mathbf{H}}_{F-G}^\dagger(u, v) - \tilde{\mathbf{J}}_G(\varphi(u, v)) \mathbf{Y}^\dagger(u, v) \right\|_F.$$

Dieser Term lässt sich analog zum  $G^1$ -Stetigkeitstest mit der Pseudo-Inversen minimieren. Durch diese Berechnung erhält man also ein Maß für die Ähnlichkeit beider Hesse-Matrizen unter Berücksichtigung einer regulären Parametrisierung, das in Abbildung 3.15 dargestellt ist.

**Bemerkung 3.23.** Wie bereits im Kurvenfall (vgl. Bemerkung 3.12) lässt sich auch hier der Zusammenhang zwischen  $C^2$ -Stetigkeit und der Reparametrisierung  $\varphi$  herstellen. Ein Flächenübergang heißt an einem Punkt  $C^2$ -stetig, wenn die Bedingungen für  $G^1$ - und  $G^2$ -Stetigkeit erfüllt



(a)  $G^1$ -Stetigkeitsanalyse in MATLAB



(b)  $G^2$ -Stetigkeitsanalyse in MATLAB

Abb. 3.15: Visualisierung der  $G^1$ - und  $G^2$ -Stetigkeitsprüfung in MATLAB durch Lösen der hier vorgestellten Least Squares Ansätze

sind und darüber hinaus

$$D\varphi(u, v) = \mathbf{X}(u, v) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad D^2\varphi(u, v) = \mathbf{Y}^\dagger(u, v) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

gilt.

Der  $G^2$ -Stetigkeitstest mithilfe der Frobeniusnorm zeigt bereits deutliche Unterschiede zu den Ergebnissen aus NX. Allerdings wird auch ersichtlich, dass diese Betrachtung für einen nicht  $G^2$ -stetigen Übergang in manchen Fällen noch unzureichend ist. Bei der Wahl einer möglichst günstigen Abfahrstrategie der Oberfläche mit einem Werkzeug beispielsweise tritt die Frage nach Vorzugsrichtungen auf. Aus diesem Grund wird der Begriff Richtungs- $G^2$ -Stetigkeit eingeführt.

### 3.3.4 Richtungs- $G^2$ -Stetigkeit

Die Motivation einen Flächenübergang entlang einer oder mehrerer Tangentialrichtungen auf  $G^2$ -Stetigkeit zu prüfen, entspringt direkt aus dem Beispiel 3.22 zu Beginn des vorangegangenen Kapitels. Ziel ist es, an einem Untersuchungspunkt Vorzugsrichtungen zu finden. Sei hierzu  $z \in \mathbb{R}^2$  ein normierter Vektor in der Tangentialebene von  $F$

$$z = \alpha F_u(u, v) + \beta F_v(u, v), \quad \alpha, \beta \in \mathbb{R}, \|z\|_2 = 1.$$

Um die Richtungs- $G^2$ -Stetigkeit eines gemeinsamen Punktes zweier parametrischer Flächen nachzuweisen, wird die Herleitung von (3.3) verwendet und die 2-Norm des folgenden Mini-

mierungsproblems auf einen Schwellwert  $\tau$  geprüft

$$\min_{\mathbf{Y}} \left\| (\mathbf{I}_3 \otimes z^T) \tilde{\mathbf{H}}_{F-G}^\dagger(u, v) z - \mathbf{J}_G(\varphi(u, v)) (z^T \otimes \mathbf{I}_2) \mathbf{Y}^\dagger(u, v) z \right\|_2. \quad (3.11)$$

Durch Ausmultiplizieren des Term  $(z^T \otimes \mathbf{I}_2) \mathbf{Y}^\dagger(u, v) z$  kann man zeigen, dass das Minimierungsproblem (3.11) äquivalent zu

$$\min_y \left\| (\mathbf{I}_3 \otimes z^T) \tilde{\mathbf{H}}_{F-G}^\dagger(u, v) z - \mathbf{J}_G(\varphi(u, v)) y \right\|_2, \quad y \in \mathbb{R}^2,$$

ist, da für  $z \neq 0$  der Ausdruck

$$\begin{aligned} (z^T \otimes \mathbf{I}_2) \mathbf{Y}^\dagger(u, v) z &= \left( (z_1 \ z_2) \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right) \begin{pmatrix} y_{uu,u} & y_{uv,u} \\ y_{uv,u} & y_{vv,u} \\ y_{uu,v} & y_{uv,v} \\ y_{uv,v} & y_{vv,v} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \\ &= \begin{pmatrix} z_1 (z_1 y_{uu,u} + z_2 y_{uu,v}) + z_2 (z_1 y_{uv,u} + z_2 y_{uv,v}) \\ z_1 (z_1 y_{uv,u} + z_2 y_{uv,v}) + z_2 (z_1 y_{vv,u} + z_2 y_{vv,v}) \end{pmatrix} \\ &= \begin{pmatrix} z_1^2 y_{uu,u} + z_1 z_2 (y_{uv,v} + y_{uu,v}) + z_2^2 y_{uv,v} \\ z_1^2 y_{uv,u} + z_1 z_2 (y_{uv,v} + y_{vv,u}) + z_2^2 y_{vv,v} \end{pmatrix} \end{aligned}$$

eine Basis des  $\mathbb{R}^2$  ergibt, wenn  $\mathbf{Y}^\dagger(u, v)$  variiert. Berechnet man die Lösung des obigen Minimierungsproblems für beliebig viele  $z \in \mathbb{R}^2$  auf dem Einheitskreis, so lässt sich die Richtungs- $G^2$ -Stetigkeit graphisch sehr anschaulich darstellen. In Abbildung 3.16 sind „gutmütige“ Richtungen (also kleine  $G^2$ -Stetigkeitsfehler) in schwarz und unstetige Richtungen in rot abgebildet. Es ist zu erkennen, dass der Übergang sowohl entlang der Randkurve als auch orthogonal dazu richtungs- $G^2$ -stetig ist.

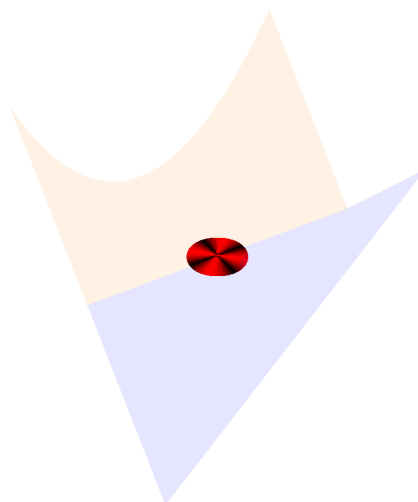


Abb. 3.16: Richtungs- $G^2$ -Stetigkeit an einem Punkt für variierende  $y$ . Der Farbindikator gibt diejenigen Richtungen an, die  $G^2$ -stetig (schwarz) bzw. nicht  $G^2$ -stetig (rot) sind.

**Satz 3.24.** Sei ein  $G^1$ -stetiger Flächenübergang zweier nicht degenerierter Flächen  $F, G$  gegeben und sei  $F$  an einem Punkt  $(u^*, v^*)$  der gemeinsamen Randkurve isometrisch parametrisiert. Für eine normierte Tangentialrichtung im Parameterbereich  $t = (t_u \ t_v)^T \in \mathbb{R}^2$  liefert das Residuum des linearen Ausgleichsproblems

$$\min_{\mathbf{Y}} \left\| \left( \mathbf{I}_3 \otimes t^T \right) \left( \mathbf{H}_F^\dagger(\cdot) - \left( \mathbf{I}_3 \otimes \mathbf{X}(\cdot)^T \right) \mathbf{H}_G^\dagger(\cdot) \mathbf{X}(\cdot) \right) t + \mathbf{J}_G(\cdot) \left( t^T \otimes \mathbf{I}_2 \right) \mathbf{Y}^\dagger(\cdot) t \right\|_2$$

den Krümmungssprung der Normalenkrümmung beider Flächen, wobei  $\mathbf{X}(u^*, v^*)$  die lokale  $C^1$ -Reparametrisierung an der Übergangsstelle  $(u^*, v^*)$  ist.

**Beweis:** Da der Übergang nach Voraussetzung  $G^1$ -stetig ist, gibt es eine Reparametrisierung  $\varphi$  mit  $\varphi'(u^*, v^*) = \mathbf{X}(u^*, v^*)$ , sodass

$$\| \mathbf{J}_F(u^*, v^*) - \mathbf{J}(\varphi(u^*, v^*)) \mathbf{X}(u^*, v^*) \|_F = 0$$

gilt. Damit folgt  $D^1 F(u^*, v^*) = D^1 (G \circ \varphi)(u^*, v^*)$  und durch die isometrische Parametrisierung von  $F$  am Punkt  $(u^*, v^*)$  ergeben sich folgende drei Eigenschaften:

1) Gleichheit der partiellen Ableitungen von  $F$  und  $G \circ \varphi$

$$\frac{dF}{du}(u^*, v^*) = \frac{d(G \circ \varphi)}{du}(u^*, v^*), \quad \frac{dF}{dv}(u^*, v^*) = \frac{d(G \circ \varphi)}{dv}(u^*, v^*),$$

2) Normiertheit aller partiellen Ableitungen

$$\left\| \frac{dF}{du}(u^*, v^*) \right\|_2 = \left\| \frac{d(G \circ \varphi)}{du}(u^*, v^*) \right\|_2 = \left\| \frac{dF}{dv}(u^*, v^*) \right\|_2 = \left\| \frac{d(G \circ \varphi)}{dv}(u^*, v^*) \right\|_2 = 1,$$

3) Orthogonalität der partiellen Ableitungen

$$\left\langle \frac{dF}{du}(u^*, v^*), \frac{dF}{dv}(u^*, v^*) \right\rangle = \left\langle \frac{d(G \circ \varphi)}{du}(u^*, v^*), \frac{d(G \circ \varphi)}{dv}(u^*, v^*) \right\rangle = 0.$$

Bezeichne im Folgenden  $\tilde{G} = (G \circ \varphi)(u^*, v^*)$  und entsprechend  $\tilde{G}_u = \frac{\partial(G \circ \varphi)}{\partial u}(u^*, v^*)$ , usw. die Ableitungen am Punkt  $(u^*, v^*)$ . Aus obigen Eigenschaften sind die Koeffizienten der ersten Fundamentalform von  $F$  und  $\tilde{G}$  damit bereits implizit festgelegt durch

$$\mathcal{E}_{\tilde{G}} = \langle \tilde{G}_u, \tilde{G}_u \rangle = \mathcal{E}_F = \langle F_u, F_u \rangle = 1$$

$$\mathcal{F}_{\tilde{G}} = \langle \tilde{G}_u, \tilde{G}_v \rangle = \mathcal{F}_F = \langle F_u, F_v \rangle = 0$$

$$\mathcal{G}_{\tilde{G}} = \langle \tilde{G}_v, \tilde{G}_v \rangle = \mathcal{G}_F = \langle F_v, F_v \rangle = 1.$$

Durch die Voraussetzung, dass  $F$  im Punkt  $(u^*, v^*)$  lokal isometrisch parametrisiert und die Kante  $G^1$ -stetig ist, vereinfacht sich die Berechnung der Normalenkrümmungen durch den Quo-



tienten der ersten und zweiten Fundamentalform

$$\begin{aligned}\kappa_F &= \frac{\mathcal{L}_F t_u^2 + 2\mathcal{M}_F t_u t_v + \mathcal{N}_F t_v^2}{\mathcal{E}_F t_u^2 + 2\mathcal{F}_F t_u t_v + \mathcal{G}_F t_v^2} = \frac{\mathcal{L}_F t_u^2 + 2\mathcal{M}_F t_u t_v + \mathcal{N}_F t_v^2}{t_u^2 + t_v^2} \stackrel{\|t\|=1}{=} \mathcal{L}_F t_u^2 + 2\mathcal{M}_F t_u t_v + \mathcal{N}_F t_v^2, \\ \kappa_{\tilde{G}} &= \frac{\mathcal{L}_{\tilde{G}} t_u^2 + 2\mathcal{M}_{\tilde{G}} t_u t_v + \mathcal{N}_{\tilde{G}} t_v^2}{\mathcal{E}_{\tilde{G}} t_u^2 + 2\mathcal{F}_{\tilde{G}} t_u t_v + \mathcal{G}_{\tilde{G}} t_v^2} = \frac{\mathcal{L}_{\tilde{G}} t_u^2 + 2\mathcal{M}_{\tilde{G}} t_u t_v + \mathcal{N}_{\tilde{G}} t_v^2}{t_u^2 + t_v^2} \stackrel{\|t\|=1}{=} \mathcal{L}_{\tilde{G}} t_u^2 + 2\mathcal{M}_{\tilde{G}} t_u t_v + \mathcal{N}_{\tilde{G}} t_v^2.\end{aligned}$$

Betrachtet man nun die Differenz der beiden Normalenkrümmung in der Tangentialrichtung  $t$  und ersetzt die entsprechenden Koeffizienten durch ihre Definition 2.22, erhält man

$$\begin{aligned}|\kappa_F - \kappa_{\tilde{G}}| &= |\mathcal{L}_F t_u^2 + 2\mathcal{M}_F t_u t_v + \mathcal{N}_F t_v^2 - \mathcal{L}_{\tilde{G}} t_u^2 + 2\mathcal{M}_{\tilde{G}} t_u t_v + \mathcal{N}_{\tilde{G}} t_v^2| \\ &= |(\mathcal{L}_F - \mathcal{L}_{\tilde{G}}) t_u^2 + 2(\mathcal{M}_F - \mathcal{M}_{\tilde{G}}) t_u t_v + (\mathcal{N}_F - \mathcal{N}_{\tilde{G}}) t_v^2| \\ &= \left| \langle F_{uu} - \tilde{G}_{uu}, \hat{n} \rangle t_u^2 + 2 \langle F_{uv} - \tilde{G}_{uv}, \hat{n} \rangle t_u t_v + \langle F_{vv} - \tilde{G}_{vv}, \hat{n} \rangle t_v^2 \right|\end{aligned}$$

Seien  $\nu_1, \nu_2$  und  $\nu_3$  drei Vektoren in der Tangentialebene und damit der Form

$$\nu_1 = \mathbf{J}_G \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \quad \nu_2 = \mathbf{J}_G \begin{pmatrix} y_3 \\ y_4 \end{pmatrix}, \quad \nu_3 = \mathbf{J}_G \begin{pmatrix} y_5 \\ y_6 \end{pmatrix}, \quad y_i \in \mathbb{R}, i = 1, \dots, 6,$$

gegeben. Aufgrund ihrer Konstruktion sind alle drei Vektoren  $\nu_1, \nu_2, \nu_3$  orthogonal zur (normierten) Flächennormale  $\hat{n}$ , d.h. es gilt

$$\langle \nu_1, \hat{n} \rangle = \langle \nu_2, \hat{n} \rangle = \langle \nu_3, \hat{n} \rangle = 0.$$

Die Differenz der Normalenkrümmungen lässt sich mit der Cauchy-Schwarzschen Ungleichung für frei wählbare  $y_i, i = 1, \dots, 6$  abschätzen und anschließend kompakt als Matrixprodukt darstellen:

$$\begin{aligned}|\kappa_F - \kappa_{\tilde{G}}| &= \left| \langle F_{uu} - \tilde{G}_{uu}, \hat{n} \rangle t_u^2 + 2 \langle F_{uv} - \tilde{G}_{uv}, \hat{n} \rangle t_u t_v + \langle F_{vv} - \tilde{G}_{vv}, \hat{n} \rangle t_v^2 \right| \\ &= \left| \langle F_{uu} - \tilde{G}_{uu}, \hat{n} \rangle t_u^2 + \langle \nu_1, \hat{n} \rangle t_u^2 + 2 \langle F_{uv} - \tilde{G}_{uv}, \hat{n} \rangle t_u t_v + 2 \langle \nu_2, \hat{n} \rangle t_u t_v \right. \\ &\quad \left. + \langle F_{vv} - \tilde{G}_{vv}, \hat{n} \rangle t_v^2 + \langle \nu_3, \hat{n} \rangle t_v^2 \right| \\ &= \left| \left\langle \left( F_{uu} - \tilde{G}_{uu} + \nu_1 \right) t_u^2 + 2 \left( F_{uv} - \tilde{G}_{uv} + \nu_2 \right) t_u t_v + \left( F_{vv} - \tilde{G}_{vv} + \nu_3 \right) t_v^2, \hat{n} \right\rangle \right| \\ &\leq \left\| \left( F_{uu} - \tilde{G}_{uu} + \nu_1 \right) t_u^2 + 2 \left( F_{uv} - \tilde{G}_{uv} + \nu_2 \right) t_u t_v + \left( F_{vv} - \tilde{G}_{vv} + \nu_3 \right) t_v^2 \right\|_2 \|\hat{n}\|_2 \\ &= \left\| \left( \mathbf{I}_3 \otimes t^T \right) \left( \mathbf{H}_F^\dagger - \left( \mathbf{I}_3 \otimes \mathbf{X}^T \right) \mathbf{H}_G^\dagger \mathbf{X} \right) t + \nu_1 t_u^2 + \nu_2 t_u t_v + \nu_3 t_v^2 \right\|_2 \\ &= \left\| \left( \mathbf{I}_3 \otimes t^T \right) \left( \mathbf{H}_F^\dagger - \left( \mathbf{I}_3 \otimes \mathbf{X}^T \right) \mathbf{H}_G^\dagger \mathbf{X} \right) t + \mathbf{J}_G \left( t^T \otimes \mathbf{I}_2 \right) \begin{pmatrix} y_1 & y_3 \\ y_2 & y_4 \\ y_3 & y_5 \\ y_4 & y_6 \end{pmatrix} t \right\|_2.\end{aligned}$$

Da obiger Zusammenhang für beliebige  $\mathbf{Y}^\dagger$  gilt, folgt

$$|\kappa_F - \kappa_{\tilde{G}}| \leq \min_{\mathbf{Y}} \left\| (\mathbf{I}_3 \otimes t^T) \left( \mathbf{H}_F^\dagger - (\mathbf{I}_3 \otimes \mathbf{X}^T) \mathbf{H}_G^\dagger \mathbf{X} \right) t + \mathbf{J}_G (t^T \otimes \mathbf{I}_2) \mathbf{Y}^\dagger t \right\|_2. \quad (3.12)$$

Wähle nun  $\nu_1, \nu_2$  und  $\nu_3$  folgendermaßen

$$\begin{aligned} \tilde{\nu}_1 &= \mathbf{J}_G \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, & \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} &= \arg \min_{\mu_1, \mu_2} \left( \left\| F_{uu} - \tilde{G}_{uu} + \mathbf{J}_G \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \right\|_2 \right), \\ \tilde{\nu}_2 &= \mathbf{J}_G \begin{pmatrix} y_3 \\ y_4 \end{pmatrix}, & \begin{pmatrix} y_3 \\ y_4 \end{pmatrix} &= \arg \min_{\mu_1, \mu_2} \left( \left\| F_{uv} - \tilde{G}_{uv} + \mathbf{J}_G \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \right\|_2 \right), \\ \tilde{\nu}_3 &= \mathbf{J}_G \begin{pmatrix} y_5 \\ y_6 \end{pmatrix}, & \begin{pmatrix} y_5 \\ y_6 \end{pmatrix} &= \arg \min_{\mu_1, \mu_2} \left( \left\| F_{vv} - \tilde{G}_{vv} + \mathbf{J}_G \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \right\|_2 \right). \end{aligned}$$

Da die Residuen obiger Minimierungsprobleme senkrecht auf der Tangentialebene (aufgespannt von  $\mathbf{J}_G$ ) stehen, lassen sie sich als Vielfaches von  $\hat{n}$  darstellen

$$\begin{aligned} F_{uu} - \tilde{G}_{uu} + \tilde{\nu}_1 &= \lambda_1 \hat{n}, \\ F_{uv} - \tilde{G}_{uv} + \tilde{\nu}_2 &= \lambda_2 \hat{n}, \\ F_{vv} - \tilde{G}_{vv} + \tilde{\nu}_3 &= \lambda_3 \hat{n} \end{aligned}$$

und damit insbesondere

$$\left( F_{uu} - \tilde{G}_{uu} + \tilde{\nu}_1 \right) t_u^2 + \left( F_{uv} - \tilde{G}_{uv} + \tilde{\nu}_2 \right) t_u t_v + \left( F_{vv} - \tilde{G}_{vv} + \tilde{\nu}_3 \right) t_v^2 = (\lambda_1 t_u^2 + \lambda_2 t_u t_v + \lambda_3 t_v^2) \hat{n}.$$

Daraus folgt für die Cauchy-Schwarzschen Ungleichung auf beiden Seiten Gleichheit und es gilt

$$\begin{aligned} |\kappa_F - \kappa_{\tilde{G}}| &= \left| \left\langle \left( F_{uu} - \tilde{G}_{uu} + \tilde{\nu}_1 \right) t_u^2 + 2 \left( F_{uv} - \tilde{G}_{uv} + \tilde{\nu}_2 \right) t_u t_v + \left( F_{vv} - \tilde{G}_{vv} + \tilde{\nu}_3 \right) t_v^2, \hat{n} \right\rangle \right| \\ &= \left\| \left( F_{uu} - \tilde{G}_{uu} + \tilde{\nu}_1 \right) t_u^2 + 2 \left( F_{uv} - \tilde{G}_{uv} + \tilde{\nu}_2 \right) t_u t_v + \left( F_{vv} - \tilde{G}_{vv} + \tilde{\nu}_3 \right) t_v^2 \right\|_2 \|\hat{n}\|_2 \\ &= \left\| (\mathbf{I}_3 \otimes t^T) \left( \mathbf{H}_F^\dagger - (\mathbf{I}_3 \otimes \mathbf{X}^T) \mathbf{H}_G^\dagger \mathbf{X} \right) t + \tilde{\nu}_1 t_u^2 + \tilde{\nu}_2 t_u t_v + \tilde{\nu}_3 t_v^2 \right\|_2 \end{aligned}$$

und mit (3.12) ergibt sich

$$|\kappa_F - \kappa_{\tilde{G}}| = \min_{\mathbf{Y}} \left\| (\mathbf{I}_3 \otimes t^T) \left( \mathbf{H}_F^\dagger - (\mathbf{I}_3 \otimes \mathbf{X}^T) \mathbf{H}_G^\dagger \mathbf{X} \right) t + \mathbf{J}_G (t^T \otimes \mathbf{I}_2) \mathbf{Y}^\dagger t \right\|_2,$$

da das gesuchte  $\mathbf{Y}$  obigen Term genau dann minimiert, wenn  $\tilde{\nu}_1, \tilde{\nu}_2$  und  $\tilde{\nu}_3$  angenommen werden.  $\square$

Solange die Fläche am Untersuchungspunkt nicht singulär ist, kann die für Satz 3.24 notwendige punktweise Isometrie der Fläche  $F$  durch eine lineare Transformation stets erzwungen werden. Das nachfolgende Beispiel zeigt die Anwendung des obigen Satzes anhand zweier in NX konstruierter Flächenparametrisierungen unter Zuhilfenahme von MATLAB.

### Beispiel (Krümmungssprung in Tangentialrichtung)

Seien zwei Flächenparametrisierungen  $F, G$  mit einem gemeinsamen  $G^1$ -stetigen Übergang (siehe Abbildung 3.17) gegeben.

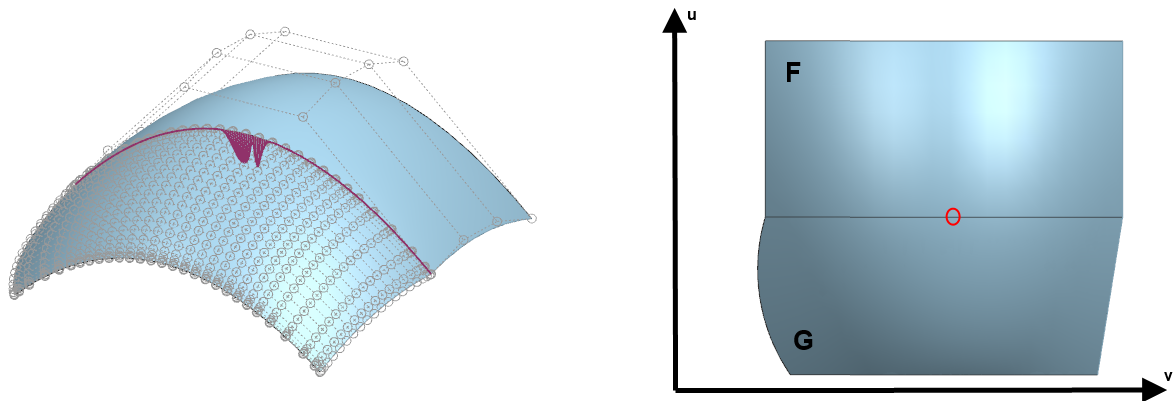


Abb. 3.17:  $G^1$ -stetiger Übergang zweier Flächenparametrisierungen  $F, G$  und ihre Kontrollpolygone (in grau dargestellt); Der lila Farbverlauf gibt den  $G^2$ -Fehler entlang der Kante mit einer NX-Funktion an. Auf der rechten Seite ist der Untersuchungspunkt hervorgehoben.

Anhand der Analysefunktion zur Überprüfung der Gauß-Krümmung eines Übergangs in NX ist bereits zu erkennen, dass im mittleren Bereich der gemeinsamen Kante Unstetigkeiten in den zweiten Ableitungen auftreten. In grau sind darüber hinaus die Kontrollpolygone von  $F$  und  $G$  dargestellt. Auf der rechten Seite sind die Parametergebiete (in der  $uv$ -Ebene) beider Flächen abgebildet. Gesucht ist der Sprung der Normalenkrümmung am Mittelpunkt der Randkurve in der Tangentialrichtung

$$t_F = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Hierzu werden die Parametrisierungen  $F, G$  und zwei Kurven auf  $F$  bzw.  $G$  wie folgt definiert: Seien  $F$  und  $G$  kubische Tensorprodukt-Flächen mit normierten Knotenvektoren, d.h.

$$F.T_u = [0, 0, 0, 0, \dots, 1, 1, 1, 1], \text{ usw.}$$

und seien die Splinekurven definiert durch

$$\begin{aligned} s_F(t) &= F(u(t), 0.5), \\ s_G(t) &= G(u(t), 0.5), \end{aligned}$$

wobei  $u(t) = t, 0 \leq t \leq 1$  gilt. In Abbildung 3.18 sind beide Kurven  $s_F$  und  $s_G$  dargestellt. Auf der rechten Seite ist der Übergang mit Sicht auf die  $xy$ -Ebene, die Tangentialebene im Untersuchungspunkt, zu sehen. Die Projektion von  $s_F$  in die Tangentialebene am Mittelpunkt liefert eine Kurve mit Krümmung 0.

Das bedeutet, dass  $s_F$  am Untersuchungspunkt geodätische Krümmung 0 hat und die Krümmung  $\kappa_{s_F}$  der Kurve  $s_F$  genau der Normalenkrümmung von  $F$  in Tangentialrichtung  $t_F$  entspricht. Die

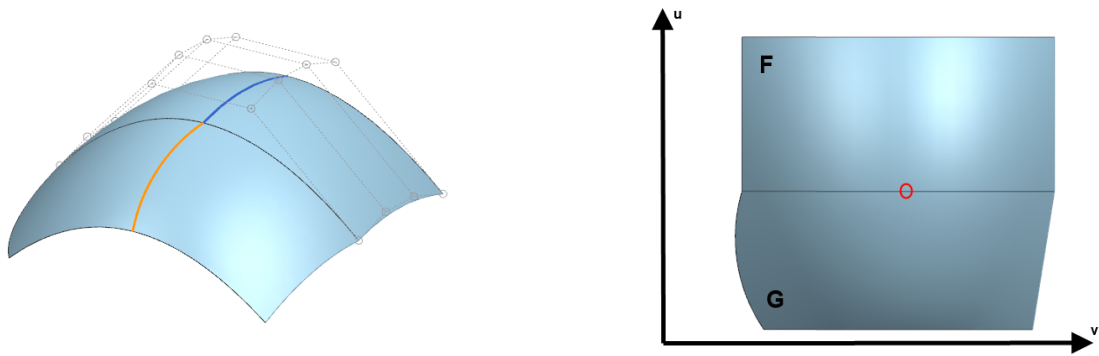


Abb. 3.18: Die beiden Splinekurven  $s_F$  und  $s_G$  sind auf  $F$  und  $G$  visualisiert. Auf der rechten Seite ist deutlich erkennbar, dass die Projektion von  $s_F$  in die Tangentialebene eine Gerade ohne Krümmung liefert und damit eine Geodäte ist.

zweite Ableitung  $s_F''$  ist damit kollinear zur Flächennormale  $n_F$  am Untersuchungspunkt. Damit ist  $F$  am Untersuchungspunkt in Richtung  $s_F$  „gutmütig“ parametrisiert und wir erhalten die Ausgangslage von Satz 3.24, da für die Richtung  $t_F$  eine punktweise isometrische Parametrisierung zugrunde liegt und damit die Normalenkrümmung in  $t_F$  der Flächenparametrisierung  $F$  gegeben ist.

Im ersten Schritt muss  $\varphi'$  gefunden werden, um die Parametrisierung der ersten Ableitung der Fläche  $G$  an die von  $F$  anzugleichen. Dazu wird das Optimierungsproblem

$$\min_{\mathbf{X}} \|\mathbf{J}_F(u, v) - \mathbf{J}_G(\varphi(u, v))\mathbf{X}(u, v)\|_F$$

gelöst. Da der Übergang  $G^1$ -stetig ist, gibt es eine Matrix  $\mathbf{X}(u, v)$ , sodass der Fehler des obigen Problems verschwindet. In diesem Beispiel gilt

$$\mathbf{X}(u, v) = \begin{pmatrix} 0.1649 & 0 \\ -0.0142 & 0.0625 \end{pmatrix}.$$

Nach Anwendung dieser Reparametrisierung ist der Übergang  $C^1$ -stetig und es gilt für die Richtungsableitungen in Richtung  $t_F$

$$\mathbf{J}_F(u, v)t_F = \mathbf{J}_G(\varphi(u, v))\mathbf{X}(u, v)t_F.$$

Da  $\mathbf{J}_F$  nach Voraussetzung (isometrische Parametrisierung) und somit auch  $\mathbf{J}_G\mathbf{X}$  eine Orthonormalbasis der Tangentialebene darstellen und  $t_F$  normiert ist, gilt insbesondere

$$\|\mathbf{J}_F(u, v)t_F\|_2 = \|\mathbf{J}_G(\varphi(u, v))\mathbf{X}(u, v)t_F\|_2 = \|t_F\|_2 = 1. \quad (3.13)$$

Diese Eigenschaft wird später bei der Berechnung der Normalenkrümmung eine essentielle Rolle spielen und sei deshalb hier nochmal erwähnt.

Im nächsten Schritt wird die Normalenkrümmung in Richtung  $t_F$  bestimmt, welche wegen der isometrischen Parametrisierung der Norm der zweiten Richtungsableitung von  $F$  in  $t_F$  ent-

spricht. In diesem Beispiel ergibt sich

$$v_1 := (\mathbf{I}_3 \otimes t_F^T) \mathbf{H}_F^\dagger(u, v) t_F = \begin{pmatrix} 0 \\ 0 \\ 0.0833 \end{pmatrix}, \quad \text{also } \kappa_{n,F} = 0.0833.$$

Entsprechend wird nun die zweite Richtungsableitung von  $(G \circ \varphi)$  in Richtung  $t_F$  berechnet (vgl. (3.11)):

$$\begin{aligned} D_{t_F} (D_{t_F} (G \circ \varphi)) (u, v) &= (\mathbf{I}_3 \otimes (t_F^T \mathbf{X}(u, v)^T)) \mathbf{H}_G^\dagger(\varphi(u, v)) \mathbf{X}(u, v) t_F \\ &\quad + \mathbf{J}_G(\varphi(u, v)) \mathbf{X}(u, v) (t_F^T \otimes \mathbf{I}_2) \mathbf{Y}^\dagger(u, v) t_F \\ &= \begin{pmatrix} 0.5588 \\ -0.0372 \\ -0.2947 \end{pmatrix} + \mathbf{J}_G(\varphi(u, v)) y. \end{aligned}$$

Das heißt die zweite Ableitung der Reparametrisierung sorgt für eine Verschiebung von  $v^1$  in der Tangentialebene mit dem Vektor  $v_2 := (0.5588 \quad -0.0372 \quad -0.2947)^T$ .

In Abbildung 3.19 links sind beide Vektoren an der entsprechenden Stelle abgebildet. Auf der rechten Seite ist das Optimierungsproblem

$$\min_y \left\| (\mathbf{I}_3 \otimes t_F^T) \left( \mathbf{H}_F^\dagger(u, v) - (\mathbf{I}_3 \otimes \mathbf{X}(u, v)^T) \mathbf{H}_F^\dagger(\varphi(u, v)) \mathbf{X}(u, v) \right) t_F + \mathbf{J}_G(\varphi(u, v)) y \right\|_2$$

abgebildet. Die Fragestellung lautet konkret, mit welchem Vektor (in Abbildung 3.19 grün dargestellt) aus der Ebene (rot) parallel zur Tangentialebene die Differenz beider Richtungsableitungen minimiert werden kann.

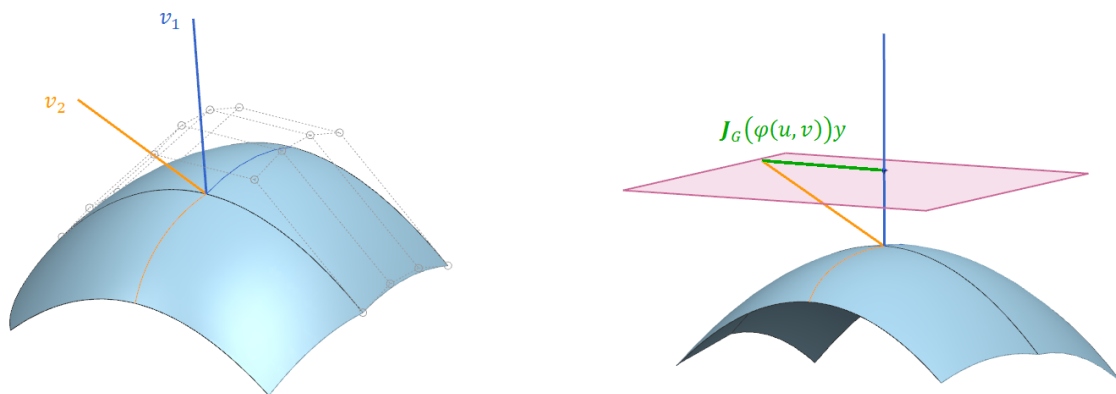


Abb. 3.19: Zweite Richtungsableitungen der Flächenparametrisierungen  $F$  und  $G$ . Rechts ist die Tangentialebene mit dem grünen Lösungsvektor  $\mathbf{J}_G(\varphi(u, v))y$  des Optimierungsproblems dargestellt.

Die Lösung  $\mathbf{J}_G(\varphi(u, v))y$  findet also die zweite Ableitung von  $\varphi(u, v)$ , die dafür sorgt, dass die zweite Richtungsableitung (orange) im Spann der ersten Ableitung und des Normalenvektors liegt. Mit anderen Worten ist es nun möglich, die Kurve  $s_G$  am Untersuchungspunkt derart zu

reparametrisieren, dass

$$s_G'' \parallel n_F$$

gilt. Damit liefert die Krümmung von  $s_G$  die Normalenkrümmung der Fläche  $G$  in Richtung  $s_F$ . An dieser Stelle verwendet man die Eigenschaft aus (3.13):

$$\kappa_{s_G} = \frac{\|s_G' \times s_G''\|_2}{\|s_G'\|_2^3} = \|s_G' \times s_G''\|_2 = \|s_G'\|_2 \cdot \|s_G''\|_2 \sin(\theta) = \|s_G''\|_2.$$

Das bedeutet, dass die Norm der zweiten Ableitung von  $s_G$  die Normalenkrümmung von  $G$  in Richtung  $t_F$  ist und damit der Fehler des Optimierungsproblems genau den Krümmungssprung zwischen beiden Normalenkrümmungen von  $F$  und  $G$  in Richtung  $t_F$  darstellt.

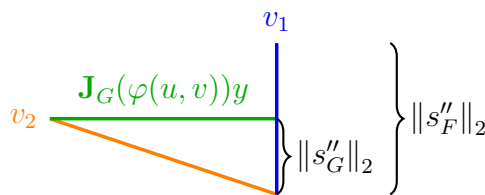


Abb. 3.20: Durch die entsprechende Reparametrisierung  $\mathbf{J}_G(\varphi(u, v))y$  sind die Normen beider Ableitungen  $s_G''$  und  $s_F''$  genau die Normalenkrümmung in Richtung  $t_F$  und der Fehler des linearen Ausgleichsproblems gibt den Krümmungssprung an.

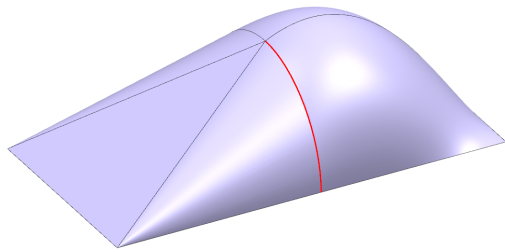
Im nachfolgenden Kapitel werden auf Basis der Definitionen von  $G^{0,1,2}$ -stetigen Flächenübergangen Bedingungen an das Kontrollpolygon von Tensorprodukt-Flächen hergeleitet, damit ein  $G^{0,1,2}$ -stetiger Flächenübergang entlang einer Kante erzeugt werden kann.

Die Erzeugung einer Tensorprodukt-Fläche, die an ihren Rändern bestimmten Stetigkeitsforderungen genügt, gewann seit dem Aufkommen der ersten CAD-Systeme enorm an Bedeutung. In [30, 53] finden sich sowohl Definitionen der  $G^1$ - und  $G^2$ -Stetigkeit zweier Flächen mit einer gemeinsamen Randkurve, als auch Verfahren zur Erzeugung  $G^{1,2}$ -stetiger Übergänge. Darüber hinaus sind in [21, 54] Methoden für die Erzeugung  $G^1$ -stetiger stückweise polynomialer und rationaler Flächen zu finden. Glatte Flächenübergänge sehen nicht nur optisch ansprechend aus, sondern sind von essentieller Bedeutung für die Erzeugung von Werkzeugbahnen, die eine nachgelagerte numerische Steuerung verarbeiten muss. Unglücklicherweise pflanzen sich Unzulänglichkeiten zu Beginn eines Fertigungsprozesses im CAD-Modell während der weiteren Bearbeitungskette fort und führen teilweise zu unerwünschtem Verhalten an der Werkzeugmaschine. Von diesem Standpunkt aus betrachtet ist es sinnvoll, das Kapitel mit einer konkreten Problemstellung zu beginnen, anhand derer die Stetigkeitsbedingungen an das Kontrollpolygon der Tensorprodukt-Fläche stückweise hergeleitet werden können. Nach der Betrachtung der  $G^0$ - und  $G^1$ -Stetigkeit wird auf der Basis des Linkage Curve Theorems [45] die  $G^2$ -Stetigkeitsbedingungen erläutert. Den Abschluss dieses Kapitels bildet die Formulierung eines quadratischen Minimierungsproblems zur Minimierung eines Energiefunktionals mithilfe der Freiheitsgrade der  $G^2$ -Stetigkeit, die in der Literatur oftmals auch *shape parameter* [3] genannt werden.

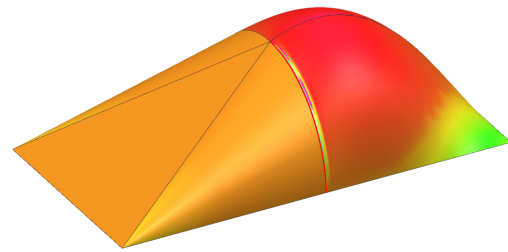
## 4.1 Problemstellung

In Abbildung 4.1 ist ein CAD-Modell abgebildet, das aus verschiedenen Freiformflächen besteht. Bei der Konstruktion wurde darauf geachtet, dass zwei benachbarte Flächen stets mindestens  $G^1$ -stetig miteinander verbunden sind. Betrachtet man speziell die rote Randkurve, stellt man bei näherer Betrachtung der Krümmungsradien auf den benachbarten Flächen fest, dass es im Bereich der rot markierten Kurve zu Unstetigkeiten in den Krümmungen kommt.

Erzeugt man anschließend auf Basis dieser CAD-Daten Werkzeugbahnen (vgl. Abbildung 4.2) für einen Kugelfräser, springt der Kontaktpunkt des Werkzeugs aufgrund der starken Krümmung direkt hinter dem Flächenübergang. Die ursprüngliche Intention des Designers, dass die Flächen „glatt“ (im Sinne von  $G^1$ -Stetigkeit) miteinander verbunden sind, spiegelt sich damit nicht im Bahncode wider. Ein weiteres Problem, das bezogen auf die Erzeugung von Werkzeugbahnen ähnliche Auswirkungen besitzt, entsteht durch Singularitäten und numerische Ungenauigkeiten. So kann es vorkommen, dass obwohl der Benutzer  $G^2$ -Stetigkeit an einem Flächenübergang gefordert hat, an einigen Stellen der Randkurve die Normalenkrümmung nur innerhalb einer gewissen Toleranz übereinstimmt. An einem solchen Übergangspunkt besitzt das Werkzeug unter Umständen nicht nur zwei (oder sogar unendlich viele) Kontaktpunkte, sondern darüber hinaus keine eindeutige oder eventuell sogar gar keine Werkzeuganstellung bezüglich der Flächengeometrie aufgrund divergenter Flächennormalen. Solche Umorientierungen auf extrem kurzen Bahnabschnitten führt dazu, dass die Steuerung die programmierte Vorschubgeschwindigkeit



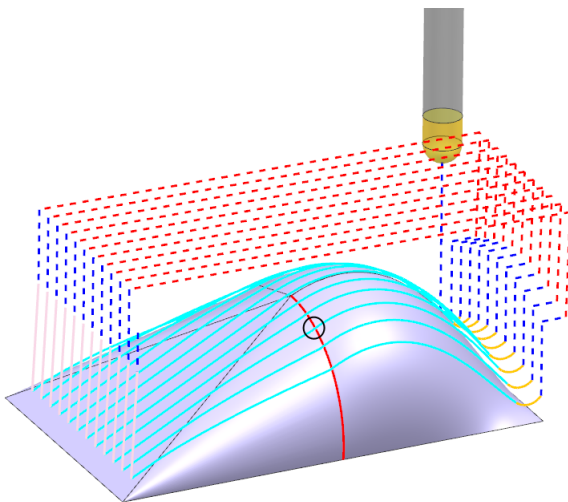
(a) CAD-Modell, das aus mehreren Freiformflächen besteht, mit einer kritischen gemeinsamen Kurve (in rot dargestellt), die bei der Erzeugung von Werkzeugbahnen ungenügende Resultate verursacht.



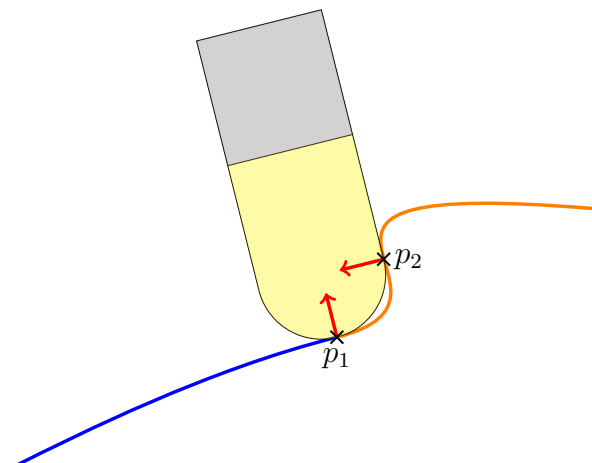
(b) Gauß-Krümmungsanalyse in NX, anhand derer deutlich zu sehen ist, dass im Bereich der Randkurve starke Krümmungssprünge auftreten, die als starke Farbänderungen erkennbar sind.

Abb. 4.1: CAD-Modell in NX mit verschiedenen Ansichten

aufgrund von physikalischen Grenzen (z.B. Ruckbegrenzung der Rundachsen) möglicherweise sogar fast bis zum kurzzeitigen Stillstand der Maschine drosselt.



(a) Aus den CAD-Daten generierte Werkzeugbahnen, die bei näherer Betrachtung am rot markierten Flächenübergang starke Umorientierungen auf kurzen Wegstrecken aufweisen



(b) Querschnitt des CAD-Modells an einer Werkzeugbahn: Doppelter Kontaktpunkt  $p_1, p_2$  mit unterschiedlichen Flächennormalen an einem  $G^1$ -stetigen Übergang

Abb. 4.2: Die aus den CAD-Daten generierten Werkzeugbahnen sehen auf den ersten Blick glatt und „gutmütig“ aus. Betrachtet man einzelne Punkte im Bereich der rot markierten Kurve näher sind starke Krümmungsänderungen sichtbar.

Auf der linken Seite (Abbildung 4.2a) sind exemplarisch erzeugte CAM-Bahnen zu sehen. Der Kugelfräser fährt das Bauteil mit dem sogenannten ZIG-Muster ab und ist zum Bauteil in Richtung der Flächennormalen angestellt. Verfeinert man die grobe Betrachtung und untersucht eine einzelne Bahn (Querschnitt des CAD-Modells) stellt man fest, dass der Übergang zwar aus mathematischer Sicht wie gefordert  $G^1$ -stetig ist, allerdings direkt nach dem gemeinsamen Punkt eine stufenartige Ausprägung entstanden ist, die zu ungewollten Fräserbewegungen führt. Die Simulation mit realen Maschinendaten einer 5-Achs-Fräsmaschine liefert die bereits vermuteten Ergebnisse. In Abbildung 4.3 ist die reale Vorschubgeschwindigkeit durch eine Maschinensi-



mulation berechnet worden. Wird ein Satz mit der möglichen Maximalgeschwindigkeit abgefahren, ist dieser in rot dargestellt. Ein blauer Abschnitt weist auf einen kompletten Stillstand der Maschinenachsen hin und bedeutet keinen Vorschub. In diesen Simulationsergebnissen ist zu erkennen, dass der Fräser entlang der Kurve auf fünf Bahnen im Bereich des rot markierten Flächenübergangs die Vorschubgeschwindigkeit drosselt.

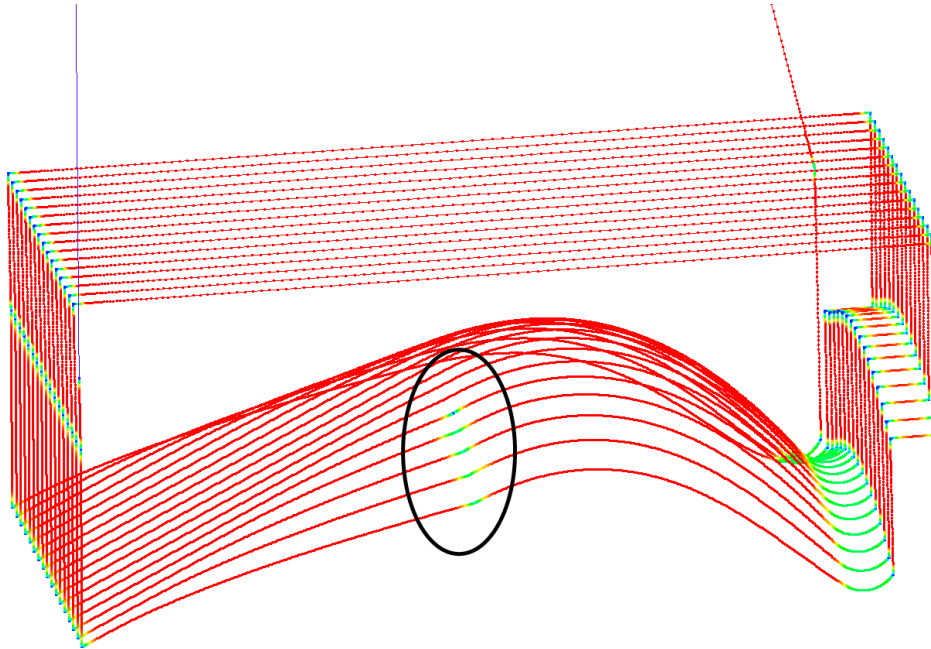


Abb. 4.3: Die Darstellung des Vorschubs anhand der simulierten Achsgeschwindigkeiten zeigt einen deutlichen Bahngeschwindigkeitseinbruch im Bereich des Flächenübergangs durch dunklere Farben.

Die Detektion solcher kritischer Stellen im Teileprogramm ist sehr aufwändig und komplex. Ein Verfahren zur Analyse von Teileprogrammen basierend auf krümmungsabhängiger Approximation des Bahnweges und anschließender Wavelet-Zerlegung mithilfe eines Spline Multiskalen Ansatzes ist in [29] zu finden. Im folgenden Algorithmus liegt eine solche Analyse bereits zugrunde, d.h. die Randkurve inklusive ihrer kritischer Stellen ist bekannt und soll nun im CAD-Modell repariert werden, um daraus bessere Werkzeugbahnen generieren zu können.

## 4.2 Algorithmus - Überblick und Lösungsstrategie

Die Ausgangslage für den Algorithmus zur Korrektur von Fräsbahnen an einem Flächenübergang bilden zwei Tensorprodukt-Flächen mit einer gemeinsamen Randkurve, in dessen Umgebung Auffälligkeiten im Teileprogramm aufgetreten sind. In einem ersten Vorverarbeitungsschritt wird mit einem primitiven Körper (z.B. einem Quader) ein ausreichend großes Loch um den kritischen Kurvenbereich ausgeschnitten/*getrimmt* (vgl. [1, Kapitel 4 und 5]) und die Informationen der getrimmten Flächen aus NX extrahiert und in MATLAB übertragen. Abbildung 4.4 zeigt die Vorverarbeitungsschritte der Eingangsdaten im CAD-System, bevor der eigentliche

Algorithmus in MATLAB die  $G^2$ -stetige Fläche erzeugt. Bevor der Algorithmus zur Erzeugung

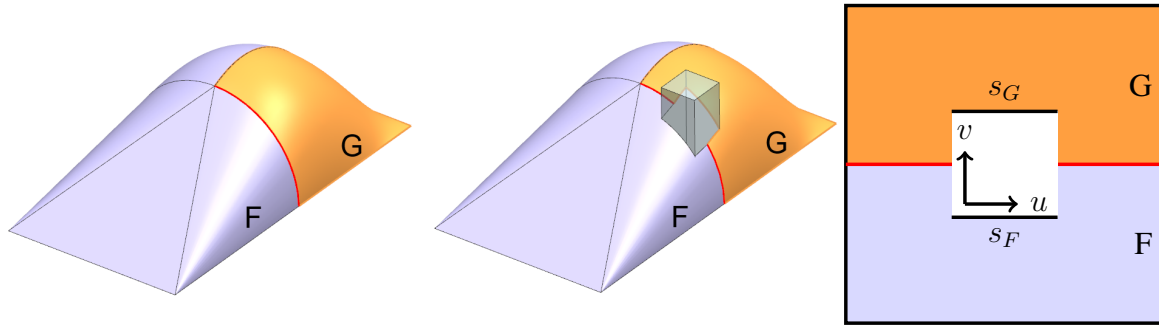
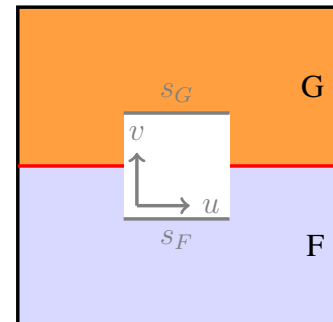


Abb. 4.4: Schema zur Vorverarbeitung der Daten im CAD System zur Erzeugung eines  $G^2$ -stetigen Patches. Im ersten Schritt wird ein Quader an der richtigen Stelle platziert, um anschließend einen Bereich um die Kurve auszuschneiden. Rechts ist der Flächenübergang schematisch abgebildet. In Richtung der beiden Schnittkurven  $s_F$  und  $s_G$  verläuft die  $u$ -Richtung des neuen Patches. In  $v$ -Richtung muss der Patch entlang beider Kurven  $G^2$ -stetig anschließen.

eines  $G^2$ -stetigen Patches konzeptionell erarbeitet und anschließend detailliert auf die konkrete Umsetzung eingegangen wird, ist es notwendig einen Überblick über Konventionen und die verwendete Notation zu geben:

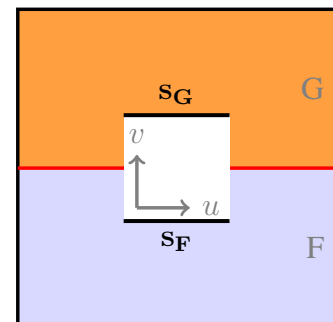
### Flächenparametrisierungen $F, G, H$

$F$  und  $G$  sind zwei bikubische Tensorprodukt-Flächen, die aus dem CAD-System NX exportiert werden. Sie besitzen eine gemeinsame Kurve, die innerhalb eines bekannten Schnittquaders repariert werden soll (siehe Abbildung 4.4). Der zu erzeugende und an Start- und Endkurve  $G^2$ -stetige Patch wird mit  $H$  bezeichnet und ist ebenfalls in beiden Parameterrichtungen kubisch.



### Splinekurven $s_F, s_G$

Die beiden Kurven  $s_F$  und  $s_G$ , die auf den Flächen  $F, G$  liegen, stellen die Randkurven der mit einem Quader erzeugten Schnittfläche dar. Sie seien als kubische  $C^2$ -stetige Splinekurven ohne Singularitäten gegeben. Entlang dieser Kurven soll die neue Fläche  $H$  an die bereits existierenden Flächen  $F$  und  $G$  jeweils  $G^2$ -stetig anschließen.

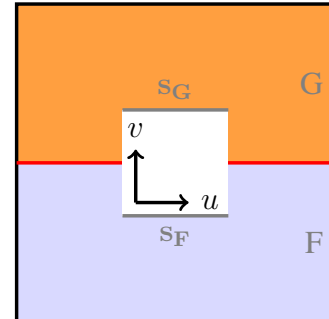


## *Parameterrichtungen des neuen Patches $H$*

Die Parameterrichtung  $u$  des neuen Patches verläuft entlang der Splinekurven  $s_F$  und  $s_G$ . Der Knotenvektor  $H.T_u$  ergibt sich damit aus den Knotenvektoren der beiden Splinekurven  $s_F$  und  $s_G$ . Die Parameterrichtung  $v$  verläuft quer zu den Splinekurven von  $s_F$  nach  $s_G$  und besitzt einen Knotenvektor der Form

$$H.T_v = [0, 0, 0, 0, a, b, 1, 1, 1, 1], \quad 0 < a < b < 1.$$

Durch diese spezielle Wahl des Knotenvektors ergeben sich insgesamt sechs Kontrollpunktzeilen für den neuen kubischen Patch  $H$ . Unter der Annahme von „gutmütigen“ Eingangsdaten [siehe [11]] reichen jeweils 3 Knotenreihen aus, um an beiden Randkurve  $G^0$ -,  $G^1$ - und  $G^2$ -Stetigkeit zu gewährleisten, da für jede Bedingung jeweils eine Kontrollpunktzeile benötigt wird.



### *Notation der Abtastung*

Im Folgenden wird unter anderem die Wahl einer geeigneten Abtastung der beiden Splinekurven  $s_F$  und  $s_G$  diskutiert. Da die beiden Kurven komplett auf den Flächen  $F$  bzw.  $G$  liegen, gibt es zu jedem Parameterwert der entsprechenden Splinekurve auch ein Paar an Parameterwerten der zugehörigen Flächenparametrisierung. Für die Abtastung einer Splinekurve  $s_F$  auf einer Fläche  $F$  (bzw.  $s_G$  auf  $G$ ) mit  $n$  Punkten wird die folgende Notation bzgl. ihrer Parameterwerte

$$\begin{aligned} s_F(u_{i,F}) &= F(\tilde{u}_{i,F}, \tilde{v}_{i,F}), & U_{s_F} &= \{u_{i,F}\}, & i &= 1, 2, \dots, n, \\ s_G(u_{i,G}) &= G(\tilde{u}_{i,G}, \tilde{v}_{i,G}), & U_{s_G} &= \{u_{i,G}\}, & i &= 1, 2, \dots, n, \end{aligned}$$

festgelegt. Ein Parameterwert  $u_{i,F} \in U_{s_F}$  einer Splinekurve besitzt mit anderen Worten also immer ein zugehöriges Parameterpaar  $(\tilde{u}_{i,F}, \tilde{v}_{i,F})$  der Fläche, auf der die Kurve existiert, sodass ihr Bild identisch ist. Parameterwerte der Flächenparametrisierungen  $F, G$  werden in diesem Kapitel stets mit dem Symbol  $\tilde{\phantom{x}}$  gekennzeichnet, während für die Parameter beider Splinekurven  $u_{i,F}$  bzw.  $u_{i,G}$  verwendet wird. Der Index  $i$  gibt an, um welchen Punkt es sich bei einer gegebenen Abtastung handelt.

### **Konzeptionelles Ablaufschema**

Alle notwendigen Operationen im CAD-System NX (Flächenextraktion, Trim, Schnittkurven, etc.) können durch die Verwendung der Schnittstelle NXOpen (siehe Kapitel 2.3) durchgeführt und die Ergebnisse durch eine entsprechende Applikation an MATLAB weiter gegeben werden. Der MATLAB-Algorithmus beginnt mit der Homogenisierung der Knotenvektoren beider extrahierter Kurven  $s_F$  und  $s_G$ , damit im Folgeschritt eine Tensorprodukt-Fläche erzeugt werden kann, die an Start- und Endkurve  $G^0$ -stetig anschließt. Durch die Generierung weiterer Reihen

des Kontrollpolygons werden die  $G^1$ - und  $G^2$ -Stetigkeit an den Rändern sichergestellt. Im letzten Schritt werden die Freiheitsgrade der  $G^2$ -Stetigkeit dazu verwendet, ein Energiefunktional  $\mathcal{F}$  zu minimieren. Der in diesem Algorithmus erzeugte neue Patch  $H$  ist sowohl in  $u$ - als auch  $v$ -Richtung kubisch. In den nachfolgenden Abschnitten werden die sechs Kontrollpunktzeilen stückweise hergeleitet. Ein konzeptionelles Ablauf-Schema ist in Abbildung 4.5 abgebildet.

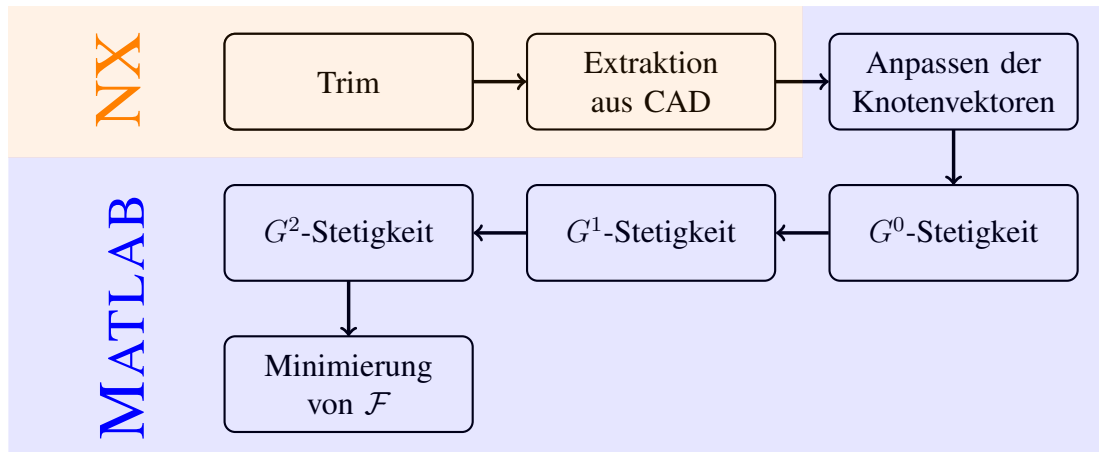


Abb. 4.5: Ablaufschema des Algorithmus zur Erzeugung eines  $G^2$ -stetigen Patches; in orange ist der Aufgabenbereich der CAD-Software NX gekennzeichnet, die auszuführenden Schritte in MATLAB sind blau hinterlegt. Um sowohl Aufgaben in NX als auch in MATLAB ausführen zu können, wurde im Zuge dieser Arbeit eine DLL programmiert, die über die Schnittstellen NXOpen und COM Automation Server beide Softwarepakete anspricht und darüber hinaus den Datenaustausch ermöglicht.

In den folgenden Abschnitten wird auf die konkrete und praxisrelevante Umsetzung der einzelnen Schritte in MATLAB näher eingegangen.

### Vorverarbeitung: Anpassung der Knotenvektoren

Der erste Schritt besteht darin, die beiden Knotenvektoren der aus NX extrahierten Splinekurven  $s_F$  und  $s_G$  zu homogenisieren. Das ist notwendig, um eine geeignete Tensorprodukt-Struktur für den neuen Patch zu erhalten. Es wird vereinbart, dass sowohl  $s_F$  als auch  $s_G$  normierte Knotenvektoren besitzen. Dies stellt keine Einschränkung dar, da durch geeignete Translation und Skalierung ein beliebiger Knotenvektor normiert werden kann. Alle Knoten, die in  $s_F$  enthalten sind und in  $s_G$  fehlen, werden entsprechend eingefügt (siehe Abbildung 4.6). Selbiger Schritt wird umgekehrt auch für alle Knoten in  $s_G$ , die nicht in  $s_F$  auftauchen, durchgeführt. Liegen zwei Knoten auf  $s_F$  und  $s_G$  dicht nebeneinander, so kann innerhalb einer gewissen Toleranz das arithmetische Mittel beider Knoten eingefügt werden, und die jeweiligen Ursprungsknoten im Anschluss daran entfernt werden. Der nachfolgende Algorithmus wird zweimal mit jeweils vertauschten Rollen von  $s_F.T$  und  $s_G.T$  aufgerufen, um die Knotenvektoren zu homogenisieren, wobei  $A(t)$  die Knoteneinfügematrix aus Kapitel 2.43 darstellt.

— Algorithmus (Homogenisierung zweier Knotenvektoren) —

Eingabe :

Knotenfolgen  $s_F.T$  und  $s_G.T$   
 Kontrollpolygone  $s_F.d$  und  $s_G.d$   
 Schwellwert  $\tau$

Prozedur :

```

ForEach  $t$  in  $s_F.T$ 
    ForEach  $\tilde{t}$  in  $s_G.T$ 
        If  $\|t - \tilde{t}\|_2 \geq \tau$ 
             $\tilde{s}_G.T = \tilde{s}_G.T \cup t$ 
             $\tilde{s}_G.d = \tilde{s}_G.d \cdot \mathbf{A}(t)$ 
        ElseIf  $\|t - \tilde{t}\|_2 > 0$ 
             $\tilde{s}_G.T = \left( \tilde{s}_G.T \cup \frac{t + \tilde{t}}{2} \right) \setminus \tilde{t}$ 
             $\tilde{s}_G.d = \tilde{s}_G.d \cdot \mathbf{A}\left(\frac{t + \tilde{t}}{2}\right)$ 
             $\tilde{s}_G.d = \tilde{s}_G.d \cdot \left( \mathbf{A}(\tilde{t})^T (\mathbf{A}(\tilde{t}) \mathbf{A}(\tilde{t})^T)^{-1} \right)$ 
        End
    End
End
    
```

Ausgabe :

Neue Knotenfolge  $\tilde{s}_G.T$   
 Neues Kontrollpolygon  $\tilde{s}_G.d$

Algorithmus 4.1: Homogenisierung zweier Knotenvektoren

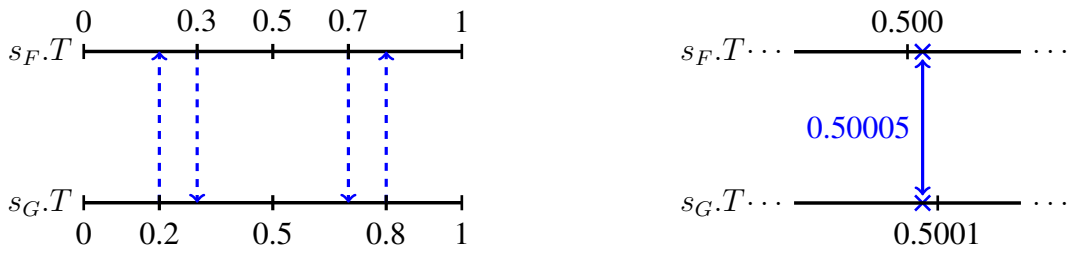
Für weitere Informationen bezüglich der Fehlerabschätzung des Knotenentfernens und der Wahl einer sinnvollen Strategie sei auf [19, 27, 37] verwiesen.

Mit diesem aus  $s_F.T$  und  $s_G.T$  zusammengesetzten Knotenvektor sind bereits die Knoten in  $H.T_u$  der Tensorprodukt-Fläche  $H$  sowie der vorab festgelegte Grad  $H.m_u$  in der  $u$ -Parameterrichtung gegeben:

$$H.m_u = 3$$

$$H.T_u = [0, 0, 0, 0, \dots, 1, 1, 1, 1].$$

Diese Informationen reichen bereits aus, um mit der Bedingung für die  $G^0$ -Stetigkeit fortzufahren.



- (a) Die jeweils fehlenden Knoten des ersten Knotenvektors  $s_F.T$  werden in  $s_G.T$  eingefügt und analog dazu die Rückrichtung des Einfügeprozesses von  $s_G.T$  nach  $s_F.T$ .
- (b) Bei dicht liegenden Knoten in  $s_F.T$  und  $s_G.T$  wird in beide Knotenvektoren das arithmetische Mittel (hier bei  $t = 0.50005$ ) eingefügt und anschließend der entsprechende Knoten in  $s_F.T$  und  $s_G.T$  (hier 0.5 und 0.5001) entfernt.

Abb. 4.6: Veranschaulichung der Homogenisierung zweier Knotenvektoren mit beiden Fällen: einfaches Knoteneinfügen links und Knotenzusammenführen rechts

## 4.2.1 $G^0$ -stetiger Übergang

Die allgemeine Bedingung für einen  $G^0$ -stetigen Übergang zweier Flächenparametrisierungen  $F, H$  lautet für die Parameter der Randkurve

$$D^0 F(u, v) = D^0 (H \circ \varphi) (u, v),$$

d.h. sie müssen beide dieselbe Randkurve besitzen. Durch die Homogenisierung der beiden Knotenvektoren der jeweiligen Randkurve kann also das Kontrollpolygon der Splinekurven  $s_F$  und  $s_G$  direkt übernommen werden:

$$\begin{aligned} H.d_{(1, :, :)} &= s_F.d \\ H.d_{(6, :, :)} &= s_G.d \end{aligned} \quad (G^0\text{-Bedingung})$$

Durch die  $(m + 1)$ -fachen Randknoten in  $H.T_v$  interpoliert die Tensorprodukt-Fläche die beiden Randkurven und der Übergang ist automatisch  $G^0$ -stetig. Um eine Tangentenstetigkeit in jedem Punkt der gemeinsamen Randkurven zu erzwingen, bedarf es der Berechnung zusätzlicher Größen wie im folgenden Abschnitt näher erläutert wird.

## 4.2.2 $G^1$ -stetiger Übergang

Bei der Betrachtung der  $G^1$ -Stetigkeit wird zunächst nur der Übergang entlang der Splinekurve  $s_F$  betrachtet. Damit der neue Patch an beiden Randkurven  $G^1$ -stetig ist, müssen alle folgenden Schritte entlang von  $s_G$  analog durchgeführt werden. Um eine Bedingung an das Kontrollpolygon eines  $G^0$ -stetigen Flächenübergangs herzuleiten, sodass die  $G^1$ -Stetigkeit

$$D^1 F(u, v) = D^1 (H \circ \varphi) (u, v)$$

entlang der gesamten Kurve  $s_F$  gilt, wird zunächst der Begriff der *Randquerableitung* (engl. cross boundary derivative) eingeführt.

**Definition 4.1.** Sei eine Flächenparametrisierung  $F$  mit einer Randkurve  $s_F$  und einer Abtastung  $U_{s_F}$ , sowie den entsprechenden Parameterwerten auf  $F$  gegeben. Eine Linearkombination

$$cbd_{s_F}(u_{i,F}) := \mu_1 s'_F(u_{i,F}) + \mu_2 (s'_F(u_{i,F}) \times n_F(\tilde{u}_{i,F}, \tilde{v}_{i,F})), \quad \mu_1, \mu_2 \in \mathbb{R}, \mu_2 \neq 0,$$

bestehend aus der Ableitung der Randkurve  $s_F$  sowie dem Kreuzprodukt aus  $s'_F$  und der Flächennormalen  $n_F$  am entsprechenden Punkt, heißt *Randquerableitung* [44].

**Bemerkung 4.2.** Eine Randquerableitung  $cbd_{s_F}(u_{i,F})$  besitzt aufgrund ihrer Definition offensichtlich die Eigenschaft in der Tangentialebene des korrespondierenden Punktes  $(\tilde{u}_{i,F}, \tilde{v}_{i,F})$  der Flächenparametrisierung  $F$  zu liegen.

Da diese Ableitungsvektoren für endlich viele Punkte berechnet werden können, ist es zunächst notwendig, eine geeignete Abtastung der Randkurven zu finden. Deshalb ist der nächste Schritt bei der Erzeugung eines  $G^1$ -stetigen Übergangs die Berechnung einer sinnvollen Abtastung beider Splinekurven  $s_F$  und  $s_G$ , bevor anschließend genauer auf die  $cbd_{s_F}$ -Punkte eingegangen wird. Hierbei bieten sich im Wesentlichen zwei Varianten besonders gut an. Während die Greville-Abszissen (vgl. Definition 2.45) einen sehr intuitiven Ansatz darstellen und darüber hinaus ohne größeren Rechenaufwand zu bestimmen sind, spielt eine äquidistante Abtastung bezüglich der Bogenlänge von  $s_F$  eine geometrisch bedeutende Rolle und liefert in praktischen Tests bessere Ergebnisse.

## Gleichmäßige Abtastung einer Splinekurve bzgl. ihrer Bogenlänge

Wie bereits in Satz 2.4 formuliert ist die Bogenlänge einer Splinekurve  $s_F$  gegeben durch

$$s(0, u^*) = \int_0^{u^*} \|s'_F(u)\|_2 \, du.$$

Eine Approximation dieses Integrals kann mit der Gauß-Quadraturformel [18, Kapitel 9]

$$\int_a^b f(u) \, du \approx (b-a) \sum_{i=0}^n \alpha_i f(u_i)$$

gelöst werden, wobei  $\alpha_i$  die Gewichte und  $u_i$  die Stützstellen bezeichnen. Sei dazu  $\delta$  ein Index auf dem Knotenvektor  $s_F.T$ , sodass  $t_\delta < t_{\delta+1}$  gilt und sei  $\alpha$  der höchste Index mit  $t_\alpha < u^*$ . Dann kann obiges Integral folgendermaßen aufgespalten werden

$$s(0, u^*) = \sum_{\delta < \alpha} \int_{t_\delta}^{t_{\delta+1}} \|s'_F(u)\|_2 \, du + \int_{t_\alpha}^{u^*} \|s'_F(u)\|_2 \, du. \quad (4.1)$$

Die einzelnen Teilintegrale können nun z.B. durch Anwendung der Gauß-Legendre-Integration approximativ gelöst werden. Hierbei ist zu beachten, dass das Standardintervall  $[-1; 1]$  durch entsprechende Variablentransformationen auf beliebige Intervalle  $[a; b]$  erweitert werden muss.

— Algorithmus (Bogenlängenapproximation einer Splinekurve) —

Eingabe :

Knotenfolge  $s_F.T$

Start- und Endknoten  $t_{start}, t_{end}$

Gewichte  $\alpha$  und Stützstellen  $x$  der Gauß-Quadratur mit Ordnung  $n$

Prozedur :

$\ell = 0;$

*ForEach*  $t_i$  in  $s_F.T \setminus \{t_{end}\}$

*If*  $t_{start} \geq t_i \ \&\& \ t_{start} \leq t_{i+1} \ \&\& \ t_{end} \leq t_{i+1}$

$\ell \ += \sum_{j=1}^n \alpha_j \left\| s'_F \left( \frac{1}{2}(t_{end} - t_{start})x(j) + \frac{1}{2}(t_{end} + t_{start}) \right) \right\|_2$

*ElseIf*  $t_{start} \geq t_i \ \&\& \ t_{start} \leq t_{i+1} \ \&\& \ t_{end} \geq t_{i+1}$

$\ell \ += \sum_{j=1}^n \alpha_j \left\| s'_F \left( \frac{1}{2}(t_{i+1} - t_{start})x(j) + \frac{1}{2}(t_{i+1} + t_{start}) \right) \right\|_2$

*ElseIf*  $t_{start} \leq t_i \ \&\& \ t_{end} \geq t_{i+1}$

$\ell \ += \sum_{j=1}^n \alpha_j \left\| s'_F \left( \frac{1}{2}(t_{i+1} - t_i)x(j) + \frac{1}{2}(t_{i+1} + t_i) \right) \right\|_2$

*ElseIf*  $t_{start} \leq t_i \ \&\& \ t_{end} \leq t_{i+1}$

$\ell \ += \sum_{j=1}^n \alpha_j \left\| s'_F \left( \frac{1}{2}(t_{end} - t_i)x(j) + \frac{1}{2}(t_{end} + t_i) \right) \right\|_2$

*End*

*End*

Ausgabe :

Approximierte Bogenlänge  $\ell$

Algorithmus 4.2: Bogenlängenapproximation einer Splinekurve

Kombiniert man diesen Algorithmus zur numerischen Bestimmung der Bogenlänge einer Splinekurve mit einem Newton-Verfahren der Form

$$u_{neu} = u_{alt} - \frac{f(u_{alt})}{f'(u_{alt})},$$

(zu finden in der Standardliteratur, z.B. [18, Kapitel 4]), dann ist es möglich, eine Abtastung im Parameterbereich derart zu approximieren, dass die Punkte auf der Splinekurve bzgl. ihrer Bogenlänge auf der Kurve gleichmäßigen Abstand besitzen. Im Folgenden werden als Eingangsvariablen die Gesamtlänge der Kurve  $\ell$  sowie die Längen der Kurven-Teilstücke  $\ell_i$ , für die die



Parameterwerte  $u_i$  gesucht werden, benötigt. Für einen geeigneten Startwert wird der Quotient

$$u_{i,start} = \frac{\ell_i}{\ell}$$

verwendet. Als Funktion des Newton-Verfahren wird  $f(\tilde{u}) = \int_0^{\tilde{u}} \|s'_F(u)\|_2 \, du$  und entsprechend  $f'(\tilde{u}) = \|s'_F(\tilde{u})\|_2$  verwendet. Der Algorithmus bricht ab, wenn entweder die maximale Anzahl an Iterationen  $maxIt$  erreicht wird, oder  $\|u_{neu} - u_{alt}\|_2 < \varepsilon$  gilt.

— Algorithmus (Äquidistante Abtastung einer Splinekurve) —

Eingabe :

Splinekurve  $s_F$

Gesamtlänge  $\ell$

Teillängen  $\tilde{\ell}_i, \quad 1 \leq i \leq count$

Prozedur :

*For*  $i = 1 : count$

$u_{alt} = \frac{\tilde{\ell}_i}{\ell}$  % Startwert

*for*  $j = 1 : maxIt$

$$u_{neu} = u_{alt} - \frac{\int_0^{u_{alt}} \|s'_F(u)\|_2 \, du - \tilde{\ell}_i}{\|s'_F(u_{alt})\|_2}$$

*If*  $\|u_{alt} - u_{neu}\|_2 < \varepsilon$

**break**

*Else*

$$u_{alt} = u_{neu}$$

*End*

*If*  $u_{alt} < \min(f.T)$

$$u_{alt} = \min(f.T)$$

*End*

*If*  $u_{alt} > \max(f.T)$

$$u_{alt} = \max(f.T)$$

*End*

*End*

$$u(i) = u_{alt}$$

*End*

Ausgabe :

Parameterwerte  $u_i$

Algorithmus 4.3: Äquidistante Abtastung einer Splinekurve

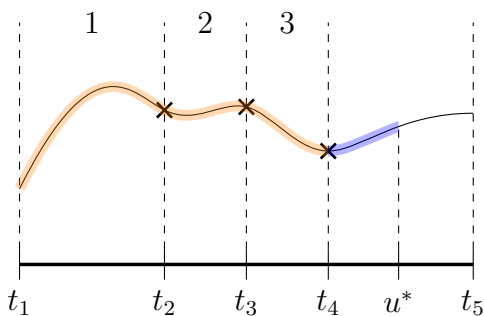
Das nachfolgende Beispiel zeigt die Ergebnisse einer solchen Berechnung durch die Anwendung beider Algorithmen.

**Beispiel 4.3.** Sei eine Splinekurve  $s_F(u)$  mit einem Knotenvektor der Form

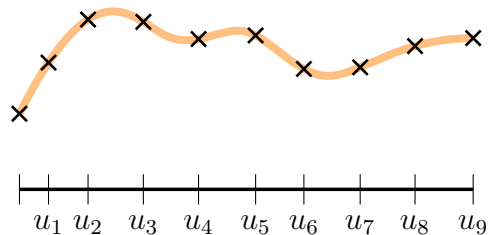
$$s_F.T = [ 0, 0, 0, \underbrace{0}_{t_1}, \underbrace{0.25}_{t_2}, \underbrace{0.5}_{t_3}, \underbrace{0.75}_{t_4}, \underbrace{1}_{t_5}, 1, 1, 1 ]$$

gegeben. Im ersten Teil dieses Beispiels wird die Bogenlänge  $s(0, u^*)$  der Splinekurve  $s_F$  berechnet, wobei  $t_4 < u^* < t_5$  gelten soll. In Abbildung 4.7a ist die Berechnung der Bogenlänge durch die Aufspaltung in einzelne Teilintegrale abgebildet (vgl. (4.1)):

$$s(0, u^*) = \underbrace{\int_{t_1}^{t_2} \|s'_F(u)\|_2 du}_1 + \underbrace{\int_{t_2}^{t_3} \|s'_F(u)\|_2 du}_2 + \underbrace{\int_{t_3}^{t_4} \|s'_F(u)\|_2 du}_3 + \int_{t_4}^{u^*} \|s'_F(u)\|_2 du$$



(a) Darstellung der einzelnen Teilintegrale 1-3, um die Gesamtlänge der Kurve bis  $u^*$  zu bestimmen.



(b) Äquidistante Abtastung der Splinekurve  $s_F$  in neun gleichlange Teilstücke.

Abb. 4.7: Links ist die Darstellung der einzelnen vollständigen Knotenintervalle in orange visualisiert. Das letzte Teilstück bis  $u^*$  ist in blau dargestellt und umfasst kein ganzes Knotenintervall mehr. Rechts ist das Ergebnis der Kombination aus den Algorithmen 4.2 und 4.3 zur Berechnung einer äquidistanten Abtastung bzgl. der Bogenlänge zu sehen.

In Abbildung 4.7b wurde die Kombination aus Bogenlängenberechnung und Newton-Verfahren verwendet, um die Splinekurve in neun gleichlange Teilstücke zu separieren. Als Eingabeparameter für das Newton-Verfahren wurden folgende Werte verwendet, wobei  $\ell$  die approximative Bogenlänge der Splinekurve darstellt:

$$\ell \approx 7.0043$$

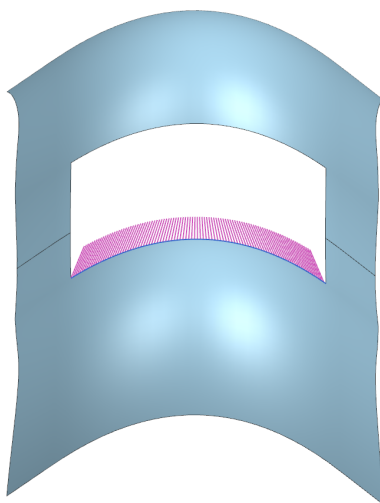
$$\tilde{\ell}_1 \approx \frac{1 \cdot \ell}{9}, \tilde{\ell}_2 \approx \frac{2 \cdot \ell}{9}, \tilde{\ell}_3 \approx \frac{3 \cdot \ell}{9}, \dots$$

Der Algorithmus liefert dann die Parameter  $u_i$  zu den entsprechenden Punkten auf  $f$ , sodass das  $i$ -te Kurvenstück die Länge  $\ell_i$  besitzt.

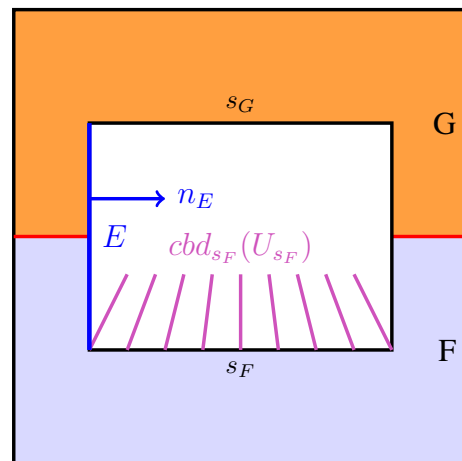
Durch das Zusammenfügen (ggf. mit Verfeinerung durch Knoteneinfügeoperationen) der Knotenvektoren und durch die Möglichkeit eine Splinekurve in gleichlange Teilstücke zu zerlegen, kann man auch komplexere Bereiche bearbeiten. Stellt man fest, dass der resultierende Übergang den numerischen Randbedingungen nicht genügt, kann durch wiederholtes Knoteneinfügen und feineres Abtasten der Randkurven die Genauigkeit erhöht werden. Um nun die Kontrollpunkt-reihen zu finden, die an beiden Randkurven der  $G^1$ -Stetigkeitsbedingung zu genügen, müssen im nächsten Schritt die  $cbd$ -Punkte (Randquerableitung) berechnet werden.

## Berechnung der Randquerableitung

Die allgemeine Definition 4.1 der Randquerableitung entlang einer Splinekurve  $s_F$  mit gegebener Abtastung  $U_{s_F}$  legt zugrunde, dass  $cbd_{s_F}(u_{i,F})$  an einem Punkt  $s_F(u_{i,F})$  lediglich in der entsprechenden Tangentialebene liegen muss und nicht kollinear zu  $s'_F(u_{i,F})$  sein darf. Sowohl die Länge als auch die Richtung des Vektors sind Freiheitsgrade, die dazu genutzt werden können, Einfluss auf die Ausdehnung und die Form des neuen Patches  $H$  zu nehmen. Eine Herangehensweise wäre z.B.  $\pm(s'_F(u_{i,F}) \times n_F(\tilde{u}_{i,F}, \tilde{v}_{i,F}))$  direkt als  $cbd_{s_F}(u_{i,F})$  zu verwenden und mit einer entsprechenden Länge zu versehen. Dies führt allerdings in vielen Fällen zu Artefakten, die den neu erzeugten Patch qualitativ unbrauchbar machen. In Abbildung 4.8 sind die einzelnen  $cbd_{s_F}$ -Vektoren entlang einer Randkurve abgebildet. Aufgrund der Flächenparametrisierung wölben sich die Vektoren in diesem Fall nach innen.



(a) In violett sind die  $cbd_{s,F}$ -Punkte entlang einer gleichmäßig abgetasteten Randkurve dargestellt. Es fällt auf, dass sich die  $cbd_{s,F}$ -Punkte nach innen wölben und somit unzureichend ist, um einen füllenden Patch zu erzeugen.



(b) Schematische Darstellung nach innen laufender  $cbd_{s_F}$ s an einem Flächenübergang entlang des Randsplines  $s_F$ . Die  $cbd_{s_F}$ -Vektoren sind durch das Kreuzprodukt zwischen der jeweiligen Flächennormalen und der entsprechenden Ableitung  $s'_F$  entstanden.

Abb. 4.8: In violett sind die Randeffekte der  $cbd_{s,F}$ -Punkte durch Berechnung mittels Kreuzprodukt abgebildet.

Um solche Randeffekte zu vermeiden, kann man Informationen des verwendeten (primitiven)

Schnittkörpers dazu verwenden, die  $cbd_{s,F}$ -Punkte durch geeignete Projektionen zu homogenisieren. Im Falle des Quaders als Schnittkörper bietet sich diejenige Ebene  $E$  an, die am Rand des Patches verläuft und die Startpunkte beider Randkurven miteinander verbindet (vgl. Abbildung 4.8b). Diese Ebene unterteilt den Patch ähnlich dem Prinzip eines Eierschneiders und ermöglicht es damit jeden  $cbd$ -Vektor durch geeignete Projektion zu begradigen. Seien also eine Abtastung der Splinekurve  $s_F$  sowie der lineare Unterraum  $E$

$$E = \{x \in \mathbb{R}^3 : \langle x - s_F(u_{i,F}), n_E \rangle = 0\},$$

gegeben. Seien weiter die Flächennormalen  $n_F$  an den Abtastpunkten von  $s_F$  bekannt. Dann liefert der folgende Algorithmus eine Projektion der  $cbd$ -Punkte auf  $E$  und damit gleichmäßig verteilte Punkte auf dem Patch:

— **Algorithmus (Projektion der  $cbd_{s_F}$ )** —

Eingabe :

Abtastung der Splinekurve  $s_F(u_{i,F}) : u_{i,F} \quad 1 \leq i \leq count$

Flächennormalen an den Stellen der Splinekurve  $s_F : n_F(\tilde{u}_{i,F}, \tilde{v}_{i,F})$

Normalenvektor der Ebene  $E : n_E$

Prozedur :

*For*  $i = 1 : count$

$incbd_{s_F}(u_{i,F}) = (s'_F(u_{i,F}) \times n_F(\tilde{u}_{i,F}, \tilde{v}_{i,F}))$

$cbd_{s_F}(u_{i,F}) = incbd_{s_F}(u_{i,F}) - \arg \min_{\psi} \|s_F(u_{i,F}) - (incbd_{s_F}(u_{i,F}) - \psi \cdot n_E)\|_2 n_E$

*End*

Ausgabe :

Randquerableitung  $cdb_{s_F}(u_{i,F})$

Algorithmus 4.4: Projektion der  $cbd$ -Punkte

Durch diese Projektion bleibt für jeden  $cbd_{s,F}$ -Punkt lediglich ein Skalar als Freiheitsgrad übrig. Dieser gibt die Entfernung des jeweiligen Punktes zum Randspline an und wird im Folgenden mit  $\alpha_i$  für  $s_F$  und  $\beta_i$  für  $s_G$  bezeichnet. Damit lassen sich abhängig von der gewählten Abtastung eine endliche Anzahl an Punkten  $p$  berechnen

$$\begin{aligned} p_{(2,i,:)} &= \alpha_i cbd_{s_F}(u_{i,F}) + s_F(u_{i,F}) \\ p_{(5,i,:)} &= \beta_i cbd_{s_G}(u_{i,G}) + s_G(u_{i,G}). \end{aligned} \tag{4.2}$$

Mithilfe der Bogenlängenapproximation der Randkurven  $s_F$  und  $s_G$ , einer darauf basierenden äquidistanten Abtastung  $U_{s_F}$  und  $U_{s_G}$  und der Möglichkeit zur Berechnung und Projektion der Randquerableitung kann die  $G^1$ -Stetigkeitsbedingung an die nächste Reihe des Kontrollpolygons als Interpolationsproblem der Form

$$\begin{aligned} \mathbf{V}_u(U_{s_F})^T H \cdot d_{(2,,:)}^T &= (p_{(2,1,:)} \quad p_{(2,2,:)} \quad \dots)^T \\ \mathbf{V}_u(U_{s_G})^T H \cdot d_{(5,,:)}^T &= (p_{(5,1,:)} \quad p_{(5,2,:)} \quad \dots)^T \end{aligned} \tag{G^1-Bedingung}$$

formuliert werden, wobei  $U_{s_F}$  die Parameterwerte der entsprechenden Abtastung darstellen. An dieser Stelle wird der Zusammenhang mit der ursprünglichen Abtastung der Randsplines durch ihre Greville-Abszissen deutlich. Wählt man eine feinere Abtastung wird aus obigem eindeutig lösbares Interpolationsproblem ggf. ein Approximationsproblem, welches wiederum durch Verfeinerung der Knotenvektoren angepasst werden kann. Nach der Anwendung aller Teilschritte erhält man ein Kontrollpolygon mit vier Reihen, dass sowohl an Start- als auch an Endkurve bereits (numerisch)  $G^1$ -stetig anschließt.

### 4.2.3 $G^2$ -stetiger Übergang

Im Folgenden werden auf dieser Basis die Bedingungen an zwei weitere mittlere Kontrollpunkt-reihen hergeleitet, damit eine  $G^2$ -Stetigkeit gewährleistet ist. Wie im vorherigen Abschnitt wird die Situation entlang einer von beiden Randkurven (hier  $s_F$ ) betrachtet. Um sowohl entlang von  $s_F$  als auch von  $s_G$  die  $G^2$ -Stetigkeit bzgl.  $H$  zu gewährleisten, müssen auch hier alle Schritte für  $s_G$  in analoger Form durchgeführt werden. Damit ein  $G^1$ -stetiger Flächenübergang zweier Flächenparametrisierungen  $F, H$  entlang einer Randkurve  $G^2$ -stetig ist, muss für jeden gemeinsamen Punkt

$$D^2 H(u, v) = D^2 (F \circ \varphi) (u, v)$$

gelten. Die folgenden Äquivalenzen sind in der Standardliteratur hinreichend bekannt und aufgrund der Konstruktion der zweiten Fundamentalform offensichtlich, allerdings bei der Erzeugung eines  $G^2$ -stetigen Übergangs von essentieller Bedeutung und aus diesem Grund an dieser Stelle aufgeführt. Eine Auflistung einiger solcher Kriterien sind z.B. in [45] zu finden.

**Bemerkung 4.4.** *Damit ein  $G^1$ -stetiger Flächenübergang an einem gemeinsamen Punkt  $(u, v)$   $G^2$ -stetig ist, sind folgende Kriterien äquivalent:*

- (1)  $D^2 H(u, v) = D^2 (F \circ \varphi) (u, v)$
- (2)  $\kappa_{n,H} = \kappa_{n,F}$

Mit anderen Worten kann ein  $G^1$ -stetiger Übergang an einem Punkt  $(u, v)$  als  $G^2$ -stetig klassifiziert werden, wenn dort die Normalenkrümmung in allen Tangentialrichtungen übereinstimmt. Eine Erweiterung dieser Aussage liefert das Linkage Curve Theorem, das einen engen Bezug zur Dupin Indikatrix aufweist und in [45] bewiesen wurde:

**Satz 4.5.** *Stimmen die Normalenkrümmungen an einem gemeinsamen Punkt für einen  $G^1$ -stetigen Flächenübergang entlang einer  $C^1$ -Kurve  $f$  in einer Tangentialrichtung unabhängig von  $f'(x)$  überein, so stimmen sie entlang aller Tangentialrichtungen überein.*

Dieser Satz ist von entscheidender Bedeutung, da nach Anwendung der bisher vorgestellten Arbeitsschritte alle Voraussetzungen erfüllt sind und im Folgenden nur noch sichergestellt werden muss, dass die Normalenkrümmung beider Flächen in jedem Punkt der gemeinsamen Kurve in mindestens einer Richtung übereinstimmt. Da die erste und zweite Kontrollpunktreihe von  $H$

bereits bekannt ist, kann die Jacobi-Matrix an den gegebenen Abtastpunkten entlang von  $s_F$  berechnet werden. In Kombination mit den berechneten  $cbd_{s_F}$ -Punkten (die ja in der Tangentialebene liegen) kann damit die Gleichung der Richtungsableitung nach der Tangentialrichtung  $(q_{i,u}, q_{i,v})^T$  eindeutig gelöst werden

$$\mathbf{J}_H(\tilde{u}_{i,H}, \tilde{v}_{i,H}) \cdot \begin{pmatrix} q_{i,u} \\ q_{i,v} \end{pmatrix} = cbd_{s_F}(u_{i,F}). \quad (4.3)$$

Bei der Berechnung der  $cbd_{s_F}$ -Punkte legt man implizit eine lokale Reparametrisierung fest, die sich in den Koeffizienten der Linearkombination (Definition 4.1) widerspiegelt. Konkret liefert der Ansatz zur Prüfung der  $G^1$ -Stetigkeit

$$\min_{\mathbf{X}} \|\mathbf{J}_H(\tilde{u}_{i,H}, \tilde{v}_{i,H}) - \mathbf{J}_F(\tilde{u}_{i,F}, \tilde{v}_{i,F})\mathbf{X}(\tilde{u}_{i,F}, \tilde{v}_{i,F})\|_F \quad (4.4)$$

eine reguläre Reparametrisierung. Mithilfe von  $\mathbf{X}$  gilt für die Richtungsableitungen folgender Zusammenhang

$$\mathbf{J}_H(\tilde{u}_{i,H}, \tilde{v}_{i,H}) \begin{pmatrix} q_{i,u} \\ q_{i,v} \end{pmatrix} = \mathbf{J}_F(\tilde{u}_{i,F}, \tilde{v}_{i,F})\mathbf{X}(\tilde{u}_{i,F}, \tilde{v}_{i,F}) \begin{pmatrix} q_{i,u} \\ q_{i,v} \end{pmatrix}.$$

Bildet man die zweite Richtungsableitung von  $F$ , erhält man die  $cdb_{s_F}$ -Punkte zweiter Ordnung:

**Definition 4.6.** Sei eine Flächenparametrisierung  $F$  mit einer Randkurve  $s_F$  und einer Abtastung  $U_{s_F}$ , sowie den entsprechenden Parameterwerten auf  $F$ , gegeben. Seien weiter die Tangentialrichtungen  $(q_{i,u} \ q_{i,v})^T$  sowie die entsprechenden Reparametrisierungen (vgl. (4.3) und (4.4)) durch die bereits bekannten  $cbd_{s_F}$ -Punkte gegeben. Die  $cbd_{s_F}^2$ -Punkte zweiter Ordnung sind dann durch

$$cdb_{s_F}^2(\cdot) := \left( \mathbf{I}_3 \otimes \begin{pmatrix} q_{i,u} \\ q_{i,v} \end{pmatrix}^T \right) \tilde{\mathbf{H}}_F^\dagger(\cdot) \begin{pmatrix} q_{i,u} \\ q_{i,v} \end{pmatrix} + \mathbf{J}_F(\cdot) \left( \begin{pmatrix} q_{i,u} \\ q_{i,v} \end{pmatrix}^T \otimes \mathbf{I}_2 \right) \mathbf{Y}^\dagger(\cdot) \begin{pmatrix} q_{i,u} \\ q_{i,v} \end{pmatrix} \quad (4.5)$$

definiert, wobei  $\tilde{\mathbf{H}}_F^\dagger(\cdot)$  die reparametrisierte Hessematrix  $\tilde{\mathbf{H}}_F^\dagger(\cdot) = (\mathbf{I}_3 \otimes \mathbf{X}(\cdot)^T) \mathbf{H}_F^\dagger(\cdot) \mathbf{X}(\cdot)$  ist.

**Bemerkung 4.7.** In Definition 4.6 ist der Einfluss der  $cbd_{s_F}$ -Punkte erster Ordnung gut zu erkennen. Hat man bereits einen  $G^1$ -stetigen Übergang erzeugt, so ist der entsprechende Freiheitsgrad, also die Tangentialrichtung  $(q_{i,u} \ q_{i,v})^T$ , fixiert. Damit ergibt sich der erste Teil

$$\left( \mathbf{I}_3 \otimes \begin{pmatrix} q_{i,u} \\ q_{i,v} \end{pmatrix}^T \right) \tilde{\mathbf{H}}_F^\dagger(\cdot) \begin{pmatrix} q_{i,u} \\ q_{i,v} \end{pmatrix}$$

bereits aus der ersten Ableitung  $\varphi'(\cdot)$  und der Hesse-Matrix  $\tilde{\mathbf{H}}_F^\dagger(\cdot)$  (vgl. (3.4)). Der zweite Teil beinhaltet nun einen weiteren Freiheitsgrad, nämlich  $\mathbf{Y}^\dagger$ . Wie bereits in (3.11) gezeigt, lässt sich dieser Freiheitsgrad, für den Fall dass die Tangentialrichtung ungleich 0 ist, vereinfacht anstatt

einer  $(4 \times 2)$ -Matrix als  $(2 \times 1)$ -Vektor darstellen

$$\mathbf{J}_F(\cdot) \begin{pmatrix} \gamma_{i,1} \\ \gamma_{i,2} \end{pmatrix}.$$

Das bedeutet, dass jeder  $cbd_{s_F}^2$ -Punkt als Freiheitsgrad eine Verschiebung um einen Vektor in der Tangentialebene besitzt, der von der bereits bekannten Jacobi-Matrix und dem Vektor  $(\gamma_{i,1} \ \gamma_{i,2})^T$  erzeugt wird. Analog wird der Vektor am Übergang entlang von  $s_G$  entsprechend mit  $(\eta_{i,1} \ \eta_{i,2})^T$  bezeichnet.

Nachfolgender Algorithmus zeigt exemplarisch die Berechnung der  $cbd_{s_F}^2$ -Punkte, wobei  $\gamma_{i,1}, \gamma_{i,2} \in \mathbb{R}$  frei wählbare Freiheitsgrade darstellen:

— **Algorithmus (Berechnung der  $cbd_{s_F}^2$ )** —

Eingabe :

Abtastung der Splinekurve  $s_F(u_i)$ ,  $1 \leq i \leq count$

Hesse-Matrizen an den Abtaststellen  $\mathbf{H}_F^\dagger(\tilde{u}_{i,F}, \tilde{v}_{i,F})$

Tangentialrichtung  $(q_{i,u} \ q_{i,v})^T$

Prozedur :

*For*  $i = 1 : count$

$$cbd_{s_F}^2(i) = \left( \mathbf{I}_3 \otimes \begin{pmatrix} q_{i,u} \\ q_{i,v} \end{pmatrix}^T \right) \tilde{\mathbf{H}}_F^\dagger(\tilde{u}_{i,F}, \tilde{v}_{i,F}) \begin{pmatrix} q_{i,u} \\ q_{i,v} \end{pmatrix} + \mathbf{J}_F(\tilde{u}_{i,F}, \tilde{v}_{i,F}) \begin{pmatrix} \gamma_{i,1} \\ \gamma_{i,2} \end{pmatrix}$$

*End*

Ausgabe :

Randquerableitung  $cbd_{s_F}^2(i)$

Algorithmus 4.5: Berechnung der  $cbd_{s_F}^2$

Aufgrund von Satz 4.5 reicht es aus, dass die zweite Ableitung entlang einer einzigen Tangentialrichtung übereinstimmt, wenn der Übergang bereits  $G^1$ -stetig ist. Damit kann man die  $G^2$ -Stetigkeitsbedingung für einen einzelnen Punkt eines  $G^1$ -stetigen Flächenübergangs wie folgt formulieren:

$$\left( \mathbf{I}_3 \otimes \begin{pmatrix} q_{i,u} \\ q_{i,v} \end{pmatrix}^T \right) \mathbf{H}_H^\dagger(\tilde{u}_{i,H}, \tilde{v}_{i,H}) \begin{pmatrix} q_{i,u} \\ q_{i,v} \end{pmatrix} = cbd_{s_F}^2(u_{i,F}). \quad (4.6)$$

Die Gleichheit der zweiten Ableitung impliziert, dass die Normalenkrümmungen von  $F$  und  $H$  in dieser Richtung identisch sind und damit der Flächenübergang an diesem Punkt  $G^2$ -stetig ist. Die zweite Ableitung der Reparametrisierung bewirkt eine Verschiebung in der Tangentialebene und stellt wie bereits in Kapitel 3.2.3 beschrieben, einen linearen Freiheitsgrad dar, der keinen Einfluss auf die Normalenkrümmung hat. Dieser wird im letzten Abschnitt dieses Kapitels nochmals aufgegriffen und dazu verwendet ein quadratisches Funktional zu minimieren. Die rechte Seite der Gleichung (4.6) kann durch die Bestimmung der Hesse-Matrix  $\tilde{\mathbf{H}}_F^\dagger(\cdot)$  und mit den Gleichungen (4.3) und (4.4) berechnet werden. Auf eine geeignete Wahl der Parameter  $\gamma_{i,1}$

und  $\gamma_{i,2}$  wird am Ende dieses Kapitels näher eingegangen. An dieser Stelle soll nur das Konzept der Erzeugung eines  $G^2$ -stetigen Übergangs erläutert werden. Aus diesem Grund gelte an dieser Stelle  $\gamma_{i,1} = \gamma_{i,2} = 0$ , sodass die rechte Seite bereits vollständig bestimmt ist.

**Bemerkung 4.8.** Aufgrund der Konstruktion der zweiten Kontrollpunktreihe gilt die  $G^1$ -Bedingung

$$\mathbf{V}_u(U_{s_F})^T H \cdot d_{(2,;,)}^T = \left( p_{(2,1,;)} \quad p_{(2,2,;)} \quad \dots \right)^T,$$

mit  $p_{(2,;,)} = \alpha_i \text{cbd}_{s_F}(u_{i,F}) + s_F(u_{i,F})$ . Damit ist die partielle Ableitung in  $v$ -Richtung des neuen Patches  $H$  an der gemeinsamen Kurve ein Vielfaches von  $(p_{(2,;,)} - s_F(u_{i,F})) = \tilde{\alpha}_i \text{cbd}_{s_F}(u_{i,F})$ . Damit folgt, dass die Tangentialrichtung aus Gleichung (4.3) die Form

$$\begin{pmatrix} q_{i,u} \\ q_{i,v} \end{pmatrix} = \begin{pmatrix} 0 \\ q_{i,v} \end{pmatrix}$$

besitzt. Damit vereinfacht sich die linke Seite von Gleichung (4.6) und es ergibt sich

$$\begin{aligned} \left( \mathbf{I}_3 \otimes \begin{pmatrix} 0 \\ q_{i,v} \end{pmatrix}^T \right) \mathbf{H}_H^\dagger(\cdot) \begin{pmatrix} 0 \\ q_{i,v} \end{pmatrix} &= \text{cbd}_{s_F}^2(\cdot) \\ q_{i,v}^2 H_{vv}(\cdot) &= \text{cbd}_{s_F}^2(\cdot) \\ H_{vv}(\cdot) &= \frac{\text{cbd}_{s_F}^2(\cdot)}{q_{i,v}^2}. \end{aligned} \tag{4.7}$$

An dieser Stelle ist zu beachten, dass die Rückführung auf die zweite partielle Ableitung in  $v$ -Richtung für die Fläche  $H$  gilt, **nicht** aber für  $F$ , denn bei der Berechnung von  $\text{cbd}_{s_F}^2(\cdot)$

$$\text{cbd}_{s_F}^2(\cdot) = \underbrace{\left( \mathbf{I}_3 \otimes \begin{pmatrix} 0 \\ q_{i,v} \end{pmatrix}^T \right) \tilde{\mathbf{H}}_F^\dagger(\cdot) \begin{pmatrix} 0 \\ q_{i,v} \end{pmatrix}}_{\neq q_{i,v}^2 F_{vv}(\cdot)} + \mathbf{J}_F(\cdot) \begin{pmatrix} \gamma_{i,1} \\ \gamma_{i,2} \end{pmatrix}$$

wird  $F$  lokal mit  $\mathbf{X}$  reparametrisiert

$$\tilde{\mathbf{H}}_F^\dagger(\cdot) = (\mathbf{I}_3 \otimes \mathbf{X}(\cdot)^T) \mathbf{H}_F^\dagger(\cdot) \mathbf{X}(\cdot).$$

Dass die Tangentialrichtung eine derartige Form besitzt, liegt ausschließlich an der Konstruktion der  $G^1$ -Stetigkeit und hat keinerlei Einfluss auf die Lösbarkeit der Gleichung (4.3), da sowohl  $H_{uu}(\cdot)$  als auch  $H_{uv}(\cdot)$  bereits ohne Kenntnis der dritten Kontrollpunktreihe berechnet werden können. Damit ist die Lösung (also die Ableitung  $H_{vv}(\cdot)$ ) unabhängig von der Wahl der Tangentialrichtung.

Es muss für jeden  $\text{cbd}_{s_F}^2$ -Punkt der entsprechende dritte Kontrollpunkt berechnet werden, sodass die zweite partielle Ableitung  $H_{vv}(\cdot)$  Gleichung (4.7) erfüllt. Dazu wird im ersten Schritt die



Struktur von  $H_{vv}(\cdot)$  näher betrachtet. Nach Satz 2.53 gilt der Zusammenhang

$$H_{vv}(u, v) = H.d(\mathbf{V}_u(u) \otimes \mathbf{V}_v''(v))$$

$$\text{vec}(H_{vv}(u, v)^T) = \text{vec}(\mathbf{V}_u(u)^T H.d^T \mathbf{V}_v''(v)).$$

Anhand dieser Umformung und der Struktur von Tensorprodukt-Flächen (siehe Kapitel 2.2.3) wird ersichtlich, dass das Kontrollpolygon im ersten Schritt reihenweise (also in  $u$ -Richtung) an den einzelnen Stellen  $u_{i,F} \in U_{s_F}$  ausgewertet wird. Durch die Konstruktion des  $G^1$ -stetigen Übergangs sind die ersten zwei Kontrollpunktreihen bereits bekannt. Eine exemplarische Auswertung an der Parameterstelle  $u_{i,F}$  ist in Abbildung 4.9 zu sehen. Der Punkt  $p_{(3,i,:)}$  ist noch unbekannt und wird erst durch Hinzunahme der  $cbd_{s_F}^2(u_{i,F})$ -Bedingung festgelegt.

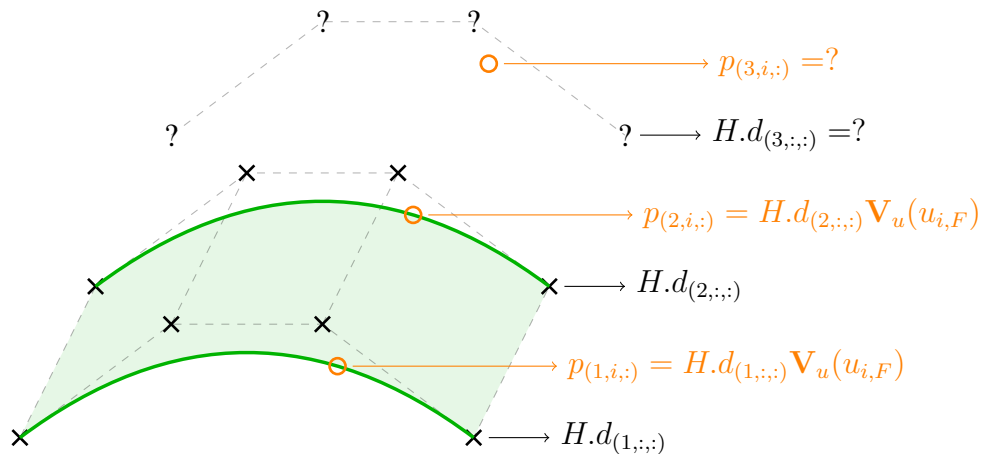


Abb. 4.9: Exemplarische Auswertung der ersten zwei Kontrollpunktreihen einer Tensorprodukt-Fläche. Da die dritte Reihe erst durch die  $G^2$ -Stetigkeitsbedingung festgelegt wird, ist sie an dieser Stelle noch unbekannt und kann damit nicht ausgewertet werden.

Das Ergebnis der ausgewerteten Reihen wird anschließend spaltenweise (also in  $v$ -Richtung) mit der entsprechenden abgeleiteten Vandermonde-Matrix  $\mathbf{V}_v''$  multipliziert. Da die Vandermonde-

Matrix  $\mathbf{V}_v''$  nur an der Start- und Endkurve berechnet werden muss, ergibt sich folgende Struktur:

$$\begin{pmatrix} p(1,i,:) & p(2,i,:) & p(3,i,:) & p(4,i,:) & p(5,i,:) & p(6,i,:) \end{pmatrix} \underbrace{\begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ 0 \\ 0 \\ 0 \end{pmatrix}}_{\mathbf{V}_v''(0)} = \frac{cbd_{s_F}^2(u_{i,F})}{q_{i,v}^2}$$

$$\begin{pmatrix} p(1,i,:) & p(2,i,:) & p(3,i,:) & p(4,i,:) & p(5,i,:) & p(6,i,:) \end{pmatrix} \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ \delta_4 \\ \delta_5 \\ \delta_6 \end{pmatrix}}_{\mathbf{V}_v''(1)} = -\frac{cbd_{s_G}^2(u_{i,G})}{q_{i,v}^2}$$

Hierbei nutzt man aus, dass die Vandermonde-Matrix  $\mathbf{V}_v''$  an Start- und Endkurve genau drei von 0 verschiedene Einträge besitzt und zwar die ersten bzw. letzten Ableitungen der Basisfunktionen. Im kubischen Fall sind zur Berechnung also genau drei Kontrollpunkte notwendig. Damit ist der Punkt  $p(3,i,:)$  eindeutig durch die Wahl des  $cbd_{s_F}^2(u_{i,F})$ -Punktes in Abhängigkeit der linearen Freiheitsgrade  $\gamma_{i,:}$  festgelegt (analog hierzu  $p(4,i,:)$  durch die Wahl von  $cbd_{s_G}^2(u_{i,G})$  mit  $\eta_{i,:}$ ) und kann mit

$$p(3,i,:) = \frac{cbd_{s_F}^2(u_{i,F})}{q_{i,v}^2} - \frac{\delta_1 p(1,i,:) - \delta_2 p(2,i,:)}{\delta_3}, \quad \mathbf{V}_v''(0) = (\delta_1 \ \delta_2 \ \delta_3 \ 0 \ 0 \ 0)^T$$

$$p(4,i,:) = -\frac{cbd_{s_G}^2(u_{i,G})}{q_{i,v}^2} - \frac{\delta_6 p(6,i,:) - \delta_5 p(5,i,:)}{\delta_4}, \quad \mathbf{V}_v''(1) = (0 \ 0 \ 0 \ \delta_4 \ \delta_5 \ \delta_6)^T$$
(4.8)

bestimmt werden. Mit diesen Vorüberlegungen lässt sich die Bedingung an die dritte und vierte Reihe des Kontrollpolygons eines bereits  $G^1$ -stetigen Übergangs formulieren, sodass der Übergang  $G^2$ -stetig ist:

$$\begin{aligned} H.d_{(3,i,:)} \mathbf{V}_u(U_{s_F}) &= (p(3,1,:) \ p(3,2,:) \ \dots) \\ H.d_{(4,i,:)} \mathbf{V}_u(U_{s_G}) &= (p(4,1,:) \ p(4,2,:) \ \dots) \end{aligned} \quad (G^2\text{-Bedingung})$$

Der zusätzliche  $G^2$ -Freiheitsgrad  $D^2\varphi(u, v)$  (vgl. Kapitel 3.2.3) ist an dieser Stelle nicht mehr so offensichtlich in der Bedingung an das Kontrollpolygon sichtbar, wie noch bei der Betrachtung der  $G^1$ -Stetigkeit. Die lineare Verschiebung in der Tangentialebene taucht allerdings analog zur  $G^1$ -Stetigkeit in den einzelnen  $cdb^2$ -Punkten (siehe Gleichung (4.8)) auf. Damit sind nun für jede der sechs Kontrollpunktzeilen die entsprechenden Bedingungen hergeleitet, sodass mit folgendem Gleichungssystem eine neue Fläche  $H$  erzeugt werden kann, die sowohl entlang der Start-

als auch der Endkurve  $G^2$ -stetig ist

$$\begin{pmatrix} H.d_{(1,::)} \\ H.d_{(2,::)} \mathbf{V}_u(U_{s_F}) \\ H.d_{(3,::)} \mathbf{V}_u(U_{s_F}) \\ H.d_{(4,::)} \mathbf{V}_u(U_{s_F}) \\ H.d_{(5,::)} \mathbf{V}_u(U_{s_F}) \\ H.d_{(6,::)} \end{pmatrix} = \begin{pmatrix} s_F \cdot d \\ p_{2,1} & p_{2,2} & \dots \\ p_{3,1} & p_{3,2} & \dots \\ p_{4,1} & p_{4,2} & \dots \\ p_{5,1} & p_{5,2} & \dots \\ s_G \cdot d \end{pmatrix}.$$

Da nun die Bedingungen an das Kontrollpolygon einer Tensorprodukt-Fläche hergeleitet sind, damit diese an Start- und Endkurve  $G^2$ -stetig ist, werden nun nochmal kurz alle verfügbaren Freiheitsgrade der  $G^1$ - und  $G^2$ -Stetigkeit betrachtet.

## Freiheitsgrade

In der zweiten und fünften Kontrollpunktreihe spielen die *cbd*-Punkte aus Definition 4.1 die entscheidende Rolle. Die Anforderung an einen *cbd*-Punkt ist, dass er in einer entsprechenden Tangentialebene liegen muss, das bedeutet jeder Punkt besitzt zwei Freiheitsgrade ( $\mu_1, \mu_2$  in Definition 4.1). Durch die Projektion in eine geeignete Ebene (abhängig vom Schnittkörper, hier ein Quader) wird ein Freiheitsgrad eliminiert und es bleibt exakt ein Freiheitsgrad für jeden *cbd*-Punkt übrig, die für die zweite Kontrollpunktreihe mit  $\alpha_i$  und für die fünfte Kontrollpunktreihe mit  $\beta_i$  bezeichnet werden (vgl. (4.2)). Diese Projektion ist notwendig, um Randeffekte des neuen Patches zu vermeiden. Besonders zu beachten ist hierbei, dass durch die Projektion und die Wahl von  $\alpha_i$  bzw.  $\beta_i$  die Reparametrisierung  $D^1\varphi(\cdot)$  an der entsprechenden Stelle festgelegt wird. Wie in Abbildung 4.10 dargestellt, wird eine gemeinsame Kurve zunächst abgetastet und anschließend die *cbd*-Punkte berechnet. Die Länge der Ableitungen ist der bei der Erzeugung eines  $G^1$ -stetigen Übergangs zur Verfügung stehende Freiheitsgrad. Dadurch ist implizit für jeden gemeinsamen Punkt auf der Kurve die Reparametrisierung  $D\varphi(u, v) = \mathbf{X}(u, v)$  festgelegt.

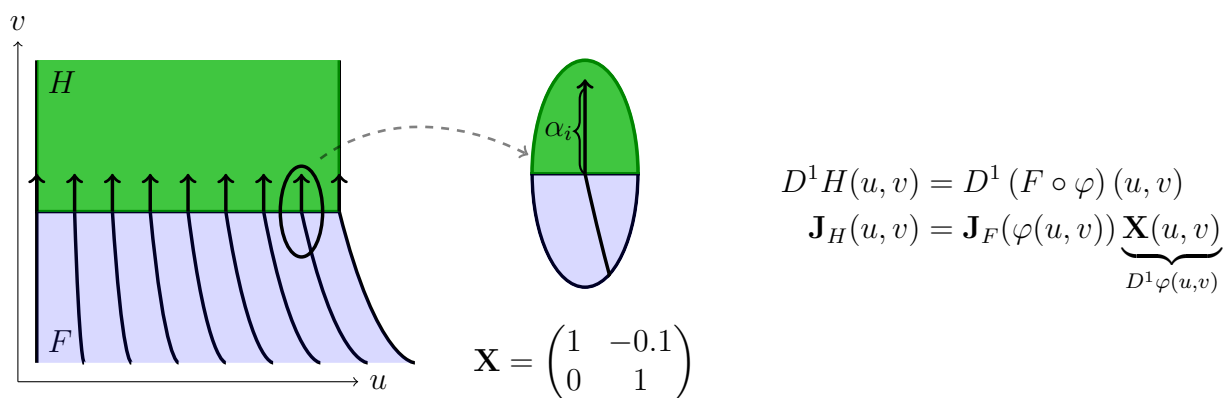


Abb. 4.10: Auf der linken Seite sind die Isolinien in  $v$ -Richtung der Fläche  $F$  dargestellt. An einem vergrößerten gemeinsamen Punkt wird  $D^1\varphi(u, v)$  einerseits durch die Projektion und andererseits durch die entsprechende Entfernung des *cbd*-Punktes zur Kurve festgelegt. Rechts ist die  $G^1$ -Bedingung durch Anwendung der Kettenregel formuliert.

Betrachtet man unter diesem Gesichtspunkt die Bedingung für  $G^2$ -Stetigkeit, stellt man fest, dass

die Reparametrisierung  $D^1\varphi(u, v) = \mathbf{X}(u, v)$  (und damit  $\alpha_i$  bzw.  $\beta_i$ ) quadratisch in die zweite Ableitung einfließt:

$$D^2H(\cdot) = D^2(F \circ \varphi)(\cdot)$$

$$\mathbf{H}_H^\dagger(\cdot) = (\mathbf{I}_3 \otimes \mathbf{X}(\cdot)^T) \mathbf{H}_F^\dagger(\varphi(\cdot)) \mathbf{X}(\cdot) + (\mathbf{J}_F(\cdot) \otimes \mathbf{I}_2) \mathbf{Y}^\dagger(\cdot).$$

Das bedeutet, dass durch die Erzeugung eines  $G^1$ -stetigen Übergangs die ersten Freiheitsgrade  $\alpha_i, \beta_i$  gewählt und fixiert werden müssen, um anschließend das Kontrollpolygon für die  $G^2$ -Stetigkeit berechnen zu können. Die dritte bzw. vierte Kontrollpunktreihe haben dann ihrerseits wie im ersten Fall lineare Freiheitsgrade, die eine Verschiebung in der Tangentialebene darstellen. Um auch in den mittleren beiden Reihen Randeffekte wie bei der Berechnung der *cbd*-Punkte zu vermeiden, wird auch hier keine beliebige Verschiebung in der Tangentialebene zugelassen, sondern nur eine Verschiebung in Richtung der *cbd*-Punkte aus der Erzeugung des  $G^1$ -stetigen Übergangs. Damit halbiert sich die Anzahl an Freiheitsgraden und die Terme

$$\mathbf{J}_F(\tilde{u}_{i,F}, \tilde{v}_{i,F}) \begin{pmatrix} \gamma_{i,1} \\ \gamma_{i,2} \end{pmatrix}, \quad \mathbf{J}_G(\tilde{u}_{i,G}, \tilde{v}_{i,G}) \begin{pmatrix} \eta_{i,1} \\ \eta_{i,2} \end{pmatrix}$$

vereinfachen sich zu

$$\gamma_i \text{cbd}_{s_F}(u_{i,F}),$$

$$\eta_i \text{cbd}_{s_G}(u_{i,G}).$$

Jeder  $i$ -te Punkt der Abtastung besitzt mit dieser Einschränkung für die  $G^2$ -Stetigkeit einen Freiheitsgrad, nämlich  $\gamma_i$  bzw.  $\eta_i$ . Bezeichne im Folgenden

$$\gamma = \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_n \end{pmatrix}, \quad \eta = \begin{pmatrix} \eta_1 \\ \vdots \\ \eta_n \end{pmatrix}.$$

Man verwendet also auch hier die in die Ebene projizierte Tangente und vermeidet damit ungewollte Ausprägungen der mittleren Kontrollpunktfolgen, da diese nicht beliebig nach außen oder innen wandern können.

Dieser lineare Freiheitsgrad (vgl. Abbildung 4.11 links), also die Verschiebung des dritten bzw. vierten Kontrollpunkts, kann dazu verwendet werden ein Funktional zu minimieren. Im Folgenden wird der Algorithmus nochmals knapp zusammengefasst, bevor im nächsten Abschnitt die aufgrund ihrer Freiheitsgrade im Allgemeinen nicht eindeutig lösbare Problemstellung der  $G^2$ -Anschlussbedingungen als quadratisches Minimierungsproblem mithilfe eines entsprechenden Energiefunktional formuliert wird.

## Zusammenfassung des Algorithmus

Nachdem alle notwendigen Geometrieinformationen aus dem CAD-Modell extrahiert sind, muss zunächst eine passende Abtastung der beiden Randkurven, entlang derer der neue Patch  $G^2$ -

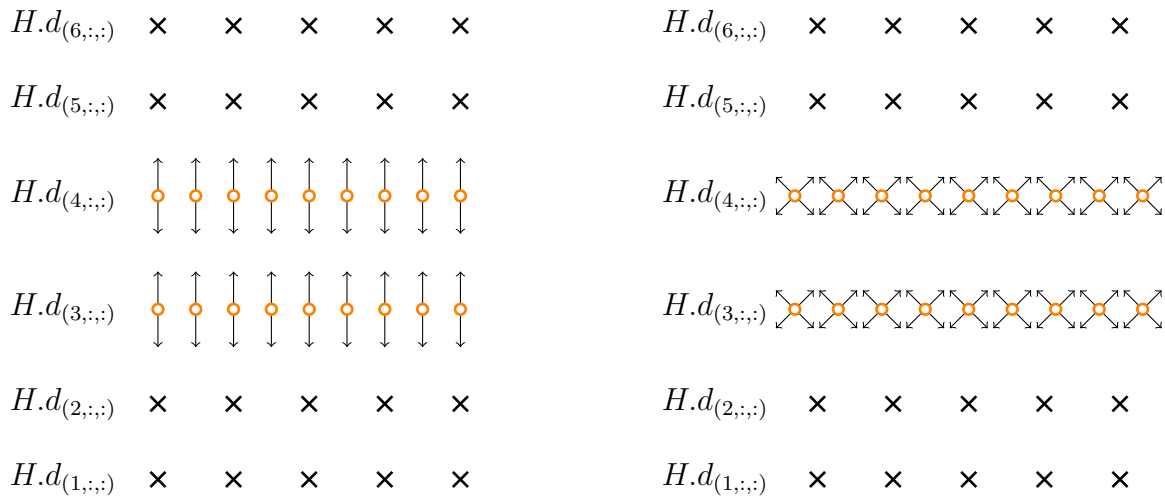


Abb. 4.11: Links sind die Freiheitsgrade der  $G^2$ -Stetigkeit in Tangentenrichtung ( $cbd$ -Punkte) dargestellt. Rechts sind beide Freiheitsgrade der mittleren Kontrollpunktreihe ohne Restriktionen abgebildet. Es ist zu erkennen, dass sich die Fläche in diesem Fall nach außen (oder innen) biegen kann.

stetig sein soll, gefunden werden. Hierbei kann man z.B. die Greville-Abszissen, eine äquidistante Abtastung oder eine Kombination aus beiden Varianten (z.B. unter Berücksichtigung der Schoenberg-Whitney Bedingung 2.44) verwenden. Die Angleichung der Knotenvektoren beider Splinekurven durch entsprechendes Knoteneinfügen liefert dann bereits die erste und letzte Kontrollpunktreihe des neuen Patches. Um die Stetigkeitsordnung des  $G^0$ -Übergangs zu erhöhen, werden im Folgeschritt für jede Abtastung der beiden Randkurven  $cbd$ -Vektoren berechnet, die jeweils in der entsprechenden Tangentialebene des Abtastpunktes liegen. Die Interpolation bzw. Approximation jener Punkte liefert dann die Reihen zwei und fünf, sodass der neue Patch numerisch  $G^1$ -stetig ist. Die schwache  $G^1$ -Restriktion, dass die  $cbd$ -Vektoren lediglich in der jeweiligen Tangentialebene liegen müssen, lässt für jeden Punkt zwei Freiheitsgrade zu, die zur Formgebung des neuen Patches verwendet werden können. Durch eine geeignete Projektion ist es möglich, die Anzahl der Freiheitsgrade je Punkt auf eins zu reduzieren, indem eine konkrete Richtung in der Tangentialebene vorgegeben wird und nur noch die Länge variabel ist. Unter Ausnutzung der differentialgeometrischen Eigenschaft, dass eine Fläche an einem Punkt eines  $G^1$ -stetigen Flächenübergangs  $G^2$ -stetig ist, wenn die Normalenkrümmung in einer Tangentialrichtung übereinstimmt, können  $cbd^2$ -Vektoren entlang der gesamten Randkurve für die gewählte Abtastung berechnet werden und damit  $G^2$ -Stetigkeit entlang des Flächenübergangs erzwingen. Diese Punkte erhält man durch die Berechnung der Richtungsableitung beider gegebener Flächen in  $u$ -Richtung des neu erzeugten (und bereits  $G^1$ -stetigen) Patches. Auch hier erhält man die Kontrollpunktfolgen durch Interpolation bzw. Approximation der  $cbd^2$ -Vektoren. Jeder dieser Punkte kann parallel zur entsprechenden Tangentialebene verschoben werden, ohne dabei die Normalenkrümmung in Tangentialrichtung zu beeinflussen. Wie bereits bei der Erzeugung der  $G^1$ -Stetigkeit wird hier allerdings lediglich die Verschiebung in einer Tangentialrichtung betrachtet, um die Anzahl der Freiheitsgrade pro Punkt auf eins zu beschränken. Im folgenden Abschnitt wird genau diese Verschiebung der  $cbd^2$ -Vektoren in der Tangentialebene (entlang des Richtungsvektors der durch den entsprechenden  $cbd$ -Vektor gegeben ist) dazu verwendet, ein quadratisches Energiefunktional zu minimieren.

### 4.3 Minimierung mithilfe der $G^2$ -Freiheitsgrade

Wie bereits in Definition 2.5 beschrieben, ist die Normalenkrümmung einer parametrischen Fläche  $F$  an einem Punkt  $(u, v)$  gegeben durch

$$\kappa_n(du, dv) = \frac{\mathcal{L}(du)^2 + 2\mathcal{M} du dv + \mathcal{N}(dv)^2}{\mathcal{E}(du)^2 + 2\mathcal{F} du dv + \mathcal{G}(dv)^2},$$

wobei  $(du \ dv)^T$  eine Tangentialrichtung darstellt. Die Minimierung solcher Krümmungsgrößen ist offensichtlich nichtlinear und damit deutlich aufwändiger als das Lösen linearer Gleichungssysteme. In praktischen Tests und Auswertungen hat sich das folgende Funktional

$$\mathcal{F} = \int_{[F_u]} \int_{[F_v]} \|F_{uu}(u, v)\|_2^2 + 2 \|F_{uv}(u, v)\|_2^2 + \|F_{vv}(u, v)\|_2^2 \, dv \, du$$

als sehr nützlich erwiesen, da es einerseits optisch ansprechende Flächen erzeugt und andererseits „nur“ quadratische Terme auftreten. Ein weiterer Vorteil offenbart sich bei näherer Betrachtung der Zerlegung der zweiten Ableitungen in Koeffizienten (Kontrollpolygon) und Basisfunktionen. Anstatt allerdings die Komponenten wie in Lemma 2.54 getrennt voneinander zu betrachten, kann man diese auch mit einem weiteren Kroneckerprodukt zusammenfassen. Die zweite Ableitung in  $u$ -Richtung einer Fläche  $F$  lässt sich damit z.B. wie folgt darstellen

$$\text{vec}(F_{uu}(\tilde{u}_{i,F}, \tilde{v}_{i,F}))^T = \text{vec}(F.d)^T (\mathbf{I}_3 \otimes \mathbf{V}_u''(\tilde{u}_{i,F}) \otimes \mathbf{V}_v(\tilde{v}_{i,F})).$$

Hierbei muss allerdings darauf geachtet werden, dass die Vektorisierung von  $F.d$  komponentenweise durchgeführt wird, d.h. die Form

$$\text{vec}(F.d) = \begin{pmatrix} \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{pmatrix} \begin{matrix} \left. \vphantom{\begin{matrix} \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{matrix}} \right\} x - \text{Komponenten} \\ \left. \vphantom{\begin{matrix} \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{matrix}} \right\} y - \text{Komponenten} \\ \left. \vphantom{\begin{matrix} \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{matrix}} \right\} z - \text{Komponenten} \end{matrix}$$

besitzt. Ordnet man die Komponenten nicht getrennt voneinander an, muss man die Einheitsmatrix  $\mathbf{I}_3$  von rechts mithilfe des Kroneckerprodukts multiplizieren. Da das Kontrollpolygon  $F.d$  in dem die Freiheitsgrade auftauchen, unabhängig von den Integrationsvariablen  $u$  und  $v$  ist, kann das Funktional in einigen Schritten vorteilhaft umformuliert werden. Im ersten Schritt wird das Integral in seine einzelnen Komponenten zerlegt, um die einzelnen Ableitungen separiert

voneinander betrachten zu können:

$$\begin{aligned}\mathcal{F} &= \int_{[F_u]} \int_{[F_v]} \|F_{uu}(u, v)\|_2^2 + 2 \|F_{uv}(u, v)\|_2^2 + \|F_{vv}(u, v)\|_2^2 \, \mathbf{d}v \, \mathbf{d}u \\ &= \int_{[F_u]} \int_{[F_v]} \|F_{uu}(u, v)\|_2^2 \, \mathbf{d}v \, \mathbf{d}u + 2 \int_{[F_u]} \int_{[F_v]} \|F_{uv}(u, v)\|_2^2 \, \mathbf{d}v \, \mathbf{d}u + \int_{[F_u]} \int_{[F_v]} \|F_{vv}(u, v)\|_2^2 \, \mathbf{d}v \, \mathbf{d}u.\end{aligned}$$

In dieser Darstellungsform zeigt sich die Nützlichkeit der MATLAB-Notation, denn es gilt (analog mit angepassten Vandermonde-Matrizen für die anderen Ableitungen zweiter Ordnung)

$$\|F_{uu}(u, v)\|_2^2 = \sum_{j=1}^3 \left( \text{vec} (F.d_{(:, :, j)})^T (\mathbf{V}_u''(u) \otimes \mathbf{V}_v(v)) \right)^2.$$

Mit dieser Zerlegung können die einzelnen Teilintegrale des Funktionals weiter vereinfacht werden. Der nächste Schritt ist wieder exemplarisch für die zweite Ableitung in  $u$ -Richtung formuliert und kann analog für die weiteren Terme durchgeführt werden:

$$\begin{aligned}\int_{[F_u]} \int_{[F_v]} \|F_{uu}(u, v)\|_2^2 \, \mathbf{d}v \, \mathbf{d}u &= \\ &= \int_{[F_u]} \int_{[F_v]} \sum_{j=1}^3 \text{vec} (F.d_{(:, :, j)})^T (\mathbf{V}_u''(u) \otimes \mathbf{V}_v(v)) (\mathbf{V}_u''(u) \otimes \mathbf{V}_v(v))^T \text{vec} (F.d_{(:, :, j)}) \, \mathbf{d}v \, \mathbf{d}u \\ &= \sum_{j=1}^3 \text{vec} (F.d_{(:, :, j)})^T \left( \int_{[F_u]} \int_{[F_v]} (\mathbf{V}_u''(u) \otimes \mathbf{V}_v(v)) (\mathbf{V}_u''(u) \otimes \mathbf{V}_v(v))^T \, \mathbf{d}v \, \mathbf{d}u \right) \text{vec} (F.d_{(:, :, j)}) \\ &= \sum_{j=1}^3 \text{vec} (F.d_{(:, :, j)})^T \left( \int_{[F_u]} \mathbf{V}_u''(u) \mathbf{V}_u''(u)^T \, \mathbf{d}u \otimes \int_{[F_v]} \mathbf{V}_v(v) \mathbf{V}_v(v)^T \, \mathbf{d}v \right) \text{vec} (F.d_{(:, :, j)})\end{aligned}$$

Durch die Umformulierung (siehe [35, Kapitel 2])

$$(\mathbf{V}_u(u) \otimes \mathbf{V}_v(v)) (\mathbf{V}_u(u) \otimes \mathbf{V}_v(v))^T = (\mathbf{V}_u(u) \mathbf{V}_u(u)^T) \otimes (\mathbf{V}_v(v) \mathbf{V}_v(v)^T)$$

vermeidet man die Notwendigkeit der Berechnung eines Flächenintegrals mit einer mehrdimensionalen Gauß-Quadratur und vereinfacht damit den Rechen- und Speicheraufwand enorm. Das Matrixprodukt der einzelnen Teilintegrale besitzt die Form einer Gram-Matrix und kann effizient (siehe z.B. [49, Kapitel 5]) berechnet werden. Das Integral lässt sich wie schon bei der Approximation der Bogenlänge (siehe Kapitel 4.2.2) mit der Gauß-Legendre-Quadratur berechnen. Seien

also

$$\begin{aligned}\mathbf{A} &= \left( \int_{[F_u]} \mathbf{V}_u''(u) \mathbf{V}_u''(u)^T \mathrm{d}u \otimes \int_{[F_v]} \mathbf{V}_v(v) \mathbf{V}_v(v)^T \mathrm{d}v \right), \\ \mathbf{B} &= \left( \int_{[F_u]} \mathbf{V}_u'(u) \mathbf{V}_u'(u)^T \mathrm{d}u \otimes \int_{[F_v]} \mathbf{V}_v'(v) \mathbf{V}_v'(v)^T \mathrm{d}v \right), \\ \mathbf{C} &= \left( \int_{[F_u]} \mathbf{V}_u(u) \mathbf{V}_u(u)^T \mathrm{d}u \otimes \int_{[F_v]} \mathbf{V}_v''(v) \mathbf{V}_v''(v)^T \mathrm{d}v \right),\end{aligned}$$

die Integrale der Gram-Matrizen der jeweiligen Ableitung. Dann kann das Funktional  $\mathcal{F}$  vereinfacht dargestellt werden, als

$$\mathcal{F} = \sum_{j=1}^3 \operatorname{vec} (F.d_{(:, :, j)})^T (\mathbf{A} + 2\mathbf{B} + \mathbf{C}) \operatorname{vec} (F.d_{(:, :, j)}).$$

Bezieht man die linearen Freiheitsgrade, die sich durch die  $G^2$ -Stetigkeiten ergeben, mit in diese Betrachtung ein, lässt sich das Kontrollpolygon  $F.d$  in Abhängigkeit von  $\gamma$  und  $\eta$  schreiben und folgendes Minimierungsproblem formulieren:

$$\min_{\gamma, \eta} \mathcal{F}(\gamma, \eta) = \min_{\gamma, \eta} \sum_{j=1}^3 \operatorname{vec} (F.d_{(:, :, j)}(\gamma, \eta))^T (\mathbf{A} + 2\mathbf{B} + \mathbf{C}) \operatorname{vec} (F.d_{(:, :, j)}(\gamma, \eta)).$$

Bevor die konkreten Formeln der einzelnen partiellen Ableitungen nach  $\gamma_i$  und  $\eta_i$  angegeben werden können, bedarf es Vorverarbeitungsschritte zur Bestimmung der ersten und letzten zwei Kontrollpolygonreihen. Zunächst müssen, wie in Algorithmus 4.1 beschrieben, durch entsprechendes Knoteneinfügen die Knotenvektoren angepasst werden. Die daraus resultierenden Kontrollpolygone bilden dann die erste und letzte Reihe des neuen Patches. Damit in der zweiten Ableitung nur lineare Freiheitsgrade auftauchen, müssen die Freiheitsgrade zur Bestimmung der  $G^1$ -Stetigkeit, also  $\alpha_i$  und  $\beta_i$  (siehe Gleichung (4.2)), festgelegt werden. Ist der Schnittquader derart konstruiert, dass alle Punkte von  $s_F$  (bzw.  $s_G$ ) in etwa denselben Abstand zur problematischen Randkurve besitzen, haben praktische Tests und Auswertungen gezeigt, dass eine gleichmäßige Länge  $\alpha_i$  (bzw.  $\beta_i$ ) für alle  $i = 1, \dots, n$  optisch sehr ansprechende Ergebnisse liefert. Es werden also im Vorfeld die zwei Freiheitsgrade

$$\begin{aligned}\tilde{\alpha} &= \alpha_i, & i = 1, \dots, n, \\ \tilde{\beta} &= \beta_i, & i = 1, \dots, n,\end{aligned}$$



festgelegt und somit die zweite und fünfte Kontrollpunktreihe bestimmt. Damit lässt sich das Funktional in Abhängigkeit von  $\gamma$  und  $\eta$  weiter aufspalten

$$\begin{aligned} \mathcal{F}(\gamma, \eta) &= \sum_{j=1}^3 \text{vec} \left( F.d_{(:, :, j)}(\gamma, \eta) \right)^T (\mathbf{A} + 2\mathbf{B} + \mathbf{C}) \text{vec} \left( F.d_{(:, :, j)}(\gamma, \eta) \right) \\ &= \sum_{j=1}^3 \begin{pmatrix} \text{vec} \left( H.d_{(1, :, j)} \right) \\ \text{vec} \left( H.d_{(2, :, j)} \right) \\ (\mathbf{V}_u^+(U_{s_F}))^T p_{(3, :, j)}(\gamma)^T \\ (\mathbf{V}_u^+(U_{s_G}))^T p_{(4, :, j)}(\eta)^T \\ \text{vec} \left( H.d_{(5, :, j)} \right) \\ \text{vec} \left( H.d_{(6, :, j)} \right) \end{pmatrix}^T (\mathbf{A} + 2\mathbf{B} + \mathbf{C}) \begin{pmatrix} \text{vec} \left( H.d_{(1, :, j)} \right) \\ \text{vec} \left( H.d_{(2, :, j)} \right) \\ (\mathbf{V}_u^+(U_{s_F}))^T p_{(3, :, j)}(\gamma)^T \\ (\mathbf{V}_u^+(U_{s_G}))^T p_{(4, :, j)}(\eta)^T \\ \text{vec} \left( H.d_{(5, :, j)} \right) \\ \text{vec} \left( H.d_{(6, :, j)} \right) \end{pmatrix}. \end{aligned}$$

Es ist zu erkennen, dass sowohl die ersten als auch die letzten beiden Kontrollpunktfolgen unabhängig von den Freiheitsgraden  $\gamma$  bzw.  $\eta$  sind. Weiter sind die Berechnungen der  $cbd^2$ -Punkte aufgrund der Konstruktionsvorschriften nur linear. Damit ist obige Gleichung quadratisch in  $\gamma_i$  und  $\eta_i$  und die partiellen Ableitungen ihrerseits wieder linear. Um diese jedoch formal herleiten zu können, ist es sinnvoll die Notation anzupassen. Seien  $r, s, t$  gegeben durch

$$r_j = \begin{pmatrix} \text{vec} \left( H.d_{(1, :, j)} \right) \\ \text{vec} \left( H.d_{(2, :, j)} \right) \end{pmatrix}, \quad s_j(\gamma, \eta) = \begin{pmatrix} (\mathbf{V}_u^+(U_{s_F}))^T p_{(3, :, j)}(\gamma)^T \\ (\mathbf{V}_u^+(U_{s_G}))^T p_{(4, :, j)}(\eta)^T \end{pmatrix}, \quad t_j = \begin{pmatrix} \text{vec} \left( H.d_{(5, :, j)} \right) \\ \text{vec} \left( H.d_{(6, :, j)} \right) \end{pmatrix}$$

und sei weiter  $\mathbf{D}$  eine symmetrische Blockmatrix der Form

$$\begin{aligned} \mathbf{D} &= (\mathbf{A} + 2\mathbf{B} + \mathbf{C}) \\ \mathbf{D} &= \begin{pmatrix} \mathbf{D}_1 & \mathbf{D}_2 & \mathbf{D}_3 \\ \mathbf{D}_2^T & \mathbf{D}_4 & \mathbf{D}_5 \\ \mathbf{D}_3^T & \mathbf{D}_5^T & \mathbf{D}_6 \end{pmatrix}. \end{aligned}$$

Die Symmetrie von  $\mathbf{D}$  ist eine direkte Folge der Konstruktion von  $\mathbf{A}, \mathbf{B}$  und  $\mathbf{C}$  als Gram-Matrizen der B-Spline Basisfunktionen. Dann lässt sich das Funktional  $\mathcal{F}(\gamma, \eta)$  kompakt schreiben als

$$\begin{aligned} \mathcal{F}(\gamma, \eta) &= \sum_{j=1}^3 \begin{pmatrix} r_j \\ s_j(\gamma, \eta) \\ t_j \end{pmatrix}^T \begin{pmatrix} \mathbf{D}_1 & \mathbf{D}_2 & \mathbf{D}_3 \\ \mathbf{D}_2^T & \mathbf{D}_4 & \mathbf{D}_5 \\ \mathbf{D}_3^T & \mathbf{D}_5^T & \mathbf{D}_6 \end{pmatrix} \begin{pmatrix} r_j \\ s_j(\gamma, \eta) \\ t_j \end{pmatrix} \\ &= \sum_{j=1}^3 \left( (r_j^T \mathbf{D}_1 + s_j(\gamma, \eta)^T \mathbf{D}_2^T + t_j^T \mathbf{D}_3^T) r_j + (r_j^T \mathbf{D}_2 + s_j(\gamma, \eta)^T \mathbf{D}_4 + t_j^T \mathbf{D}_5^T) s_j(\gamma, \eta) \right. \\ &\quad \left. + (r_j^T \mathbf{D}_3 + s_j(\gamma, \eta)^T \mathbf{D}_5 + t_j^T \mathbf{D}_6) t_j \right). \end{aligned}$$

Da  $r_j$  und  $t_j$  konstant sind, verschwindet ihre Ableitung und es ergeben sich folgende partielle Ableitungen nach  $\gamma_k$  bzw.  $\eta_k$ :

$$\begin{aligned}\frac{\partial \mathcal{F}}{\partial \gamma_k}(\gamma, \eta) &= \sum_{j=1}^3 2r_j^T \mathbf{D}_2 \frac{\partial s_j}{\partial \gamma_k}(\gamma, \eta) + 2s_j(\gamma, \eta)^T \mathbf{D}_4 \frac{\partial s_j}{\partial \gamma_k}(\gamma, \eta) + 2t_j^T \mathbf{D}_5^T \frac{\partial s_j}{\partial \gamma_k}(\gamma, \eta), \\ \frac{\partial \mathcal{F}}{\partial \eta_k}(\gamma, \eta) &= \sum_{j=1}^3 2r_j^T \mathbf{D}_2 \frac{\partial s_j}{\partial \eta_k}(\gamma, \eta) + 2s_j(\gamma, \eta)^T \mathbf{D}_4 \frac{\partial s_j}{\partial \eta_k}(\gamma, \eta) + 2t_j^T \mathbf{D}_5^T \frac{\partial s_j}{\partial \eta_k}(\gamma, \eta).\end{aligned}\tag{4.9}$$

Da die Matrix  $\mathbf{D}$  positiv semidefinit ist, ist der Extremwert der quadratischen Form  $\mathcal{F}(\gamma, \eta)$  ein Minimum, das durch Nullsetzen aller partieller Ableitungen gefunden werden kann. Ist  $\mathbf{D}$  positiv definit ist die Lösung eindeutig, andernfalls gibt es unendlich viele Lösungen. Sei  $q$  die Anzahl an Kontrollpunkten in einer Reihe. Dann sind die partiellen Ableitungen von  $s_j(\gamma, \eta)$  gegeben durch

$$\begin{aligned}\frac{\partial s_j(\gamma, \eta)}{\partial \gamma_k} &= \begin{pmatrix} \frac{\partial}{\partial \gamma_k} \left( \sum_{i=1}^n \left( \mathbf{V}_{u,(1,i)}^+(U_{s_F}) \right)^T p_{3,::j}(\gamma_i)^T \right) \\ \vdots \\ \frac{\partial}{\partial \gamma_k} \left( \sum_{i=1}^n \left( \mathbf{V}_{u,(q,i)}^+(U_{s_F}) \right)^T p_{3,::j}(\gamma_i)^T \right) \\ \frac{\partial}{\partial \gamma_k} \left( \sum_{i=1}^n \left( \mathbf{V}_{u,(1,i)}^+(U_{s_G}) \right)^T p_{4,::j}(\eta_i)^T \right) \\ \vdots \\ \frac{\partial}{\partial \gamma_k} \left( \sum_{i=1}^n \left( \mathbf{V}_{u,(q,i)}^+(U_{s_G}) \right)^T p_{4,::j}(\eta_i)^T \right) \end{pmatrix} = \begin{pmatrix} \left( \mathbf{V}_{u,(1,k)}^+(U_{s_F}) \right)^T c b d_{s_F}(u_{k,F}) \\ \vdots \\ \left( \mathbf{V}_{u,(w,k)}^+(U_{s_F}) \right)^T c b d_{s_F}(u_{k,F}) \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\ \\ \frac{\partial s_j(\gamma, \eta)}{\partial \eta_k} &= \begin{pmatrix} \frac{\partial}{\partial \eta_k} \left( \sum_{i=1}^n \left( \mathbf{V}_{u,(1,i)}^+(U_{s_F}) \right)^T p_{3,::j}(\gamma_i)^T \right) \\ \vdots \\ \frac{\partial}{\partial \eta_k} \left( \sum_{i=1}^n \left( \mathbf{V}_{u,(q,i)}^+(U_{s_F}) \right)^T p_{3,::j}(\gamma_i)^T \right) \\ \frac{\partial}{\partial \eta_k} \left( \sum_{i=1}^n \left( \mathbf{V}_{u,(1,i)}^+(U_{s_G}) \right)^T p_{4,::j}(\eta_i)^T \right) \\ \vdots \\ \frac{\partial}{\partial \eta_k} \left( \sum_{i=1}^n \left( \mathbf{V}_{u,(q,i)}^+(U_{s_G}) \right)^T p_{4,::j}(\eta_i)^T \right) \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \left( \mathbf{V}_{u,(1,k)}^+(U_{s_G}) \right)^T c b d_{s_G}(u_{k,G}) \\ \vdots \\ \left( \mathbf{V}_{u,(w,k)}^+(U_{s_G}) \right)^T c b d_{s_G}(u_{k,G}) \end{pmatrix}\end{aligned}$$

Damit lassen sich bereits vier der obigen sechs Terme aus der Gleichung (4.9) für die partiellen Ableitungen erster Ordnung berechnen:

$$\begin{aligned} \frac{\partial \mathcal{F}}{\partial \gamma_k}(\gamma, \eta) &= \sum_{j=1}^3 2r_j^T \mathbf{D}_2 \frac{\partial s_j}{\partial \gamma_k}(\gamma, \eta) + 2s_j(\gamma, \eta)^T \mathbf{D}_4 \frac{\partial s_j}{\partial \gamma_k}(\gamma, \eta) + 2t_j^T \mathbf{D}_5^T \frac{\partial s_j}{\partial \gamma_k}(\gamma, \eta) \\ &= 2 \sum_{j=1}^3 (r_j^T \mathbf{D}_2 + t_j^T \mathbf{D}_5^T) \begin{pmatrix} \left( \mathbf{V}_{u,(1,k)}^+(U_{s_F}) \right)^T \text{cbd}_{s_F}(u_{k,F}) \\ \vdots \\ \left( \mathbf{V}_{u,(q,k)}^+(U_{s_F}) \right)^T \text{cbd}_{s_F}(u_{k,F}) \\ 0 \\ \vdots \\ 0 \end{pmatrix} + s_j(\gamma, \eta)^T \mathbf{D}_4 \frac{\partial s_j}{\partial \gamma_k}(\gamma, \eta) \end{aligned}$$

$$\begin{aligned} \frac{\partial \mathcal{F}}{\partial \eta_k}(\gamma, \eta) &= \sum_{j=1}^3 2r_j^T \mathbf{D}_2 \frac{\partial s_j}{\partial \eta_k}(\gamma, \eta) + 2s_j(\gamma, \eta)^T \mathbf{D}_4 \frac{\partial s_j}{\partial \eta_k}(\gamma, \eta) + 2t_j^T \mathbf{D}_5^T \frac{\partial s_j}{\partial \eta_k}(\gamma, \eta) \\ &= 2 \sum_{j=1}^3 (r_j^T \mathbf{D}_2 + t_j^T \mathbf{D}_5^T) \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \left( \mathbf{V}_{u,(1,k)}^+(U_{s_G}) \right)^T \text{cbd}_{s_G}(u_{k,G}) \\ \vdots \\ \left( \mathbf{V}_{u,(q,k)}^+(U_{s_G}) \right)^T \text{cbd}_{s_G}(u_{k,G}) \end{pmatrix} + s_j(\gamma, \eta)^T \mathbf{D}_4 \frac{\partial s_j}{\partial \eta_k}(\gamma, \eta) \end{aligned}$$

Um die Nullstelle der  $k$ -ten partiellen Ableitungen zu bestimmen, fehlt noch die Berechnung der einzelnen  $\gamma_i, \eta_i$ , die in den Termen  $s_j(\gamma, \eta)^T \mathbf{D}_4 \frac{\partial s_j}{\partial \gamma_k}(\gamma, \eta)$  und  $s_j(\gamma, \eta)^T \mathbf{D}_4 \frac{\partial s_j}{\partial \eta_k}(\gamma, \eta)$  auftauchen. Deshalb wird im nächsten Schritt zunächst den Term  $s_j(\gamma, \eta)^T$  näher betrachtet und umformuliert.

$$s_j(\gamma, \eta) = \begin{pmatrix} \left( \mathbf{V}_u^+(U_{s_F}) \right)^T \begin{pmatrix} p(3,1,j) \\ \vdots \\ p(3,n,j) \end{pmatrix} \\ \left( \mathbf{V}_u^+(U_{s_G}) \right)^T \begin{pmatrix} p(4,1,j) \\ \vdots \\ p(4,n,j) \end{pmatrix} \end{pmatrix}$$

Im nächsten Schritt werden die Punkte  $p_{(3,i,:)}$  und  $p_{(4,i,:)}$  ausformuliert und in zwei Anteile gespalten, einem konstanten und einem linearen Term:

$$p_{(3,i,:)} = \underbrace{\frac{\left( \mathbf{I}_3 \otimes \begin{pmatrix} 0 \\ q_{i,v} \end{pmatrix}^T \right) \tilde{\mathbf{H}}_F^\dagger(\tilde{u}_{i,F}, \tilde{v}_{i,F}) \begin{pmatrix} 0 \\ q_{i,v} \end{pmatrix} - q_{i,v}^2 (\delta_1 p_{(1,i,:)} - \delta_2 p_{(2,i,:)})}{q_{i,v}^2 \delta_3}}_{\tilde{w}_{F,:}} - \frac{cbd_{s_F}(u_{i,F})}{q_{i,v}^2 \delta_3} \gamma_i,$$

$$p_{(4,i,:)} = \underbrace{\frac{\left( \mathbf{I}_3 \otimes \begin{pmatrix} 0 \\ q_{i,v} \end{pmatrix}^T \right) \tilde{\mathbf{H}}_G^\dagger(\tilde{u}_{i,G}, \tilde{v}_{i,G}) \begin{pmatrix} 0 \\ q_{i,v} \end{pmatrix} - q_{i,v}^2 (\delta_6 p_{(6,i,:)} - \delta_5 p_{(5,i,:)})}{q_{i,v}^2 \delta_4}}_{\tilde{w}_{G,:}} - \frac{cbd_{s_G}(u_{i,G})}{q_{i,v}^2 \delta_4} \eta_i.$$

Damit lässt sich der Ausdruck  $s_j(\gamma, \eta)$  in zwei Teile aufspalten:

$$s_j(\gamma, \eta) = \begin{pmatrix} (\mathbf{V}^+(U_{s_F}))^T \tilde{w}_{F,j} - (\mathbf{V}^+(U_{s_F}))^T \begin{pmatrix} \frac{cbd_{s_F}(u_{1,F})}{q_{1,v}^2 \delta_3} \gamma_1 \\ \vdots \\ \frac{cbd_{s_F}(u_{n,F})}{q_{n,v}^2 \delta_3} \gamma_n \end{pmatrix} \\ (\mathbf{V}^+(U_{s_G}))^T \tilde{w}_{G,j} - (\mathbf{V}^+(U_{s_G}))^T \begin{pmatrix} \frac{cbd_{s_G}(u_{1,G})}{q_{1,v}^2 \delta_6} \eta_1 \\ \vdots \\ \frac{cbd_{s_G}(u_{n,G})}{q_{n,v}^2 \delta_6} \eta_n \end{pmatrix} \end{pmatrix}.$$

Sei nun

$$w_j = \begin{pmatrix} (\mathbf{V}^+(U_{s_F}))^T \tilde{w}_{F,j} \\ (\mathbf{V}^+(U_{s_G}))^T \tilde{w}_{G,j} \end{pmatrix},$$

dann lässt sich die partielle Ableitung darstellen als

$$\frac{\partial \mathcal{F}}{\partial \gamma_k}(\gamma, \eta) = 2 \sum_{j=1}^3 (r_j^T \mathbf{D}_2 + w_j^T \mathbf{D}_4 + t_j^T \mathbf{D}_5^T) \begin{pmatrix} \left( \mathbf{V}_{u,(1,k)}^+(U_{s_F}) \right)^T \text{cbd}_{s_F}(u_{k,F}) \\ \vdots \\ \left( \mathbf{V}_{u,(q,k)}^+(U_{s_F}) \right)^T \text{cbd}_{s_F}(u_{k,F}) \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\ - \begin{pmatrix} \left( \mathbf{V}^+(U_{s_F}) \right)^T \begin{pmatrix} \frac{\text{cbd}_{s_F,j}(u_{1,F})}{q_{1,v}^2 \delta_3} \gamma_1 \\ \dots \\ \frac{\text{cbd}_{s_F,j}(u_{n,F})}{q_{n,v}^2 \delta_3} \gamma_n \end{pmatrix} \\ \left( \mathbf{V}^+(U_{s_G}) \right)^T \begin{pmatrix} \frac{\text{cbd}_{s_G,j}(u_{1,G})}{q_{1,v}^2 \delta_6} \eta_1 \\ \dots \\ \frac{\text{cbd}_{s_F,j}(u_{n,F})}{q_{n,v}^2 \delta_6} \eta_n \end{pmatrix} \end{pmatrix}^T \mathbf{D}_4 \begin{pmatrix} \left( \mathbf{V}_{u,(1,k)}^+(U_{s_F}) \right)^T \text{cbd}_{s_F}(u_{k,F}) \\ \vdots \\ \left( \mathbf{V}_{u,(q,k)}^+(U_{s_F}) \right)^T \text{cbd}_{s_F}(u_{k,F}) \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Mit dieser Umformulierung ist es nun möglich, ein lineares Gleichungssystem aufzustellen und alle  $2n$  partiellen Ableitungen gleich 0 zu setzen. Man erhält damit  $2n$  Gleichung für genau  $2n$  Unbekannte ( $\gamma_i, \eta_i, 1 \leq i \leq n$ ). Im nächsten Abschnitt werden die Ergebnisse miteinander verglichen und qualitativ bewertet. Die Testreihe besteht aus einer NX-generierter Lösung, der Lösung des in MATLAB implementierten Algorithmus zum Erzeugen  $G^2$ -stetiger Flächenübergänge und der erweiterten Lösung mit zusätzlicher Minimierung von  $\mathcal{F}$ .

### 4.3.1 Vergleich: Algorithmus und CAD-System

In heutigen CAD-Systemen gibt es eine Vielzahl an Möglichkeiten, Freiformflächen zu erzeugen, die an ihren Rändern gewissen Stetigkeitsbedingungen genügen. Für die folgenden Tests wurde in NX mithilfe der sehr gängigen und weit verbreiteten Funktion *Through Curves* [1, Kapitel 4.2] eine  $G^2$ -stetige Anschlussfläche erzeugt. An dieser Stelle sei nochmals explizit darauf hingewiesen, dass der nachfolgende Test nur eine einzelne, allerdings weit verbreitete und gängige Methode aus dem CAD-System NX mit dem hier vorgestellten und in MATLAB implementierten Algorithmus vergleicht. Zur qualitativen Einordnung und Vergleichbarkeit werden zunächst die maximalen Krümmungen der drei Flächen verglichen. Anschließend werden die Geschwindigkeitsprofile einzelner Maschinensimulationen aus den generierten Werkzeugbahnen der entsprechenden Flächen untersucht. Eine derartige Darstellung der Bahngeschwindigkeit ohne Reparatur der Bahnwege ist bereits in Abbildung 4.3 vorgestellt worden.

## Beschreibung der verschiedenen Varianten

Für die Gegenüberstellung der verschiedenen Möglichkeiten dient wie bereits zuvor in diesem Kapitel das in Abbildung 4.12 dargestellte CAD-Modell.

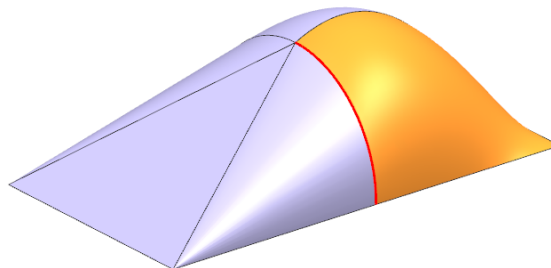
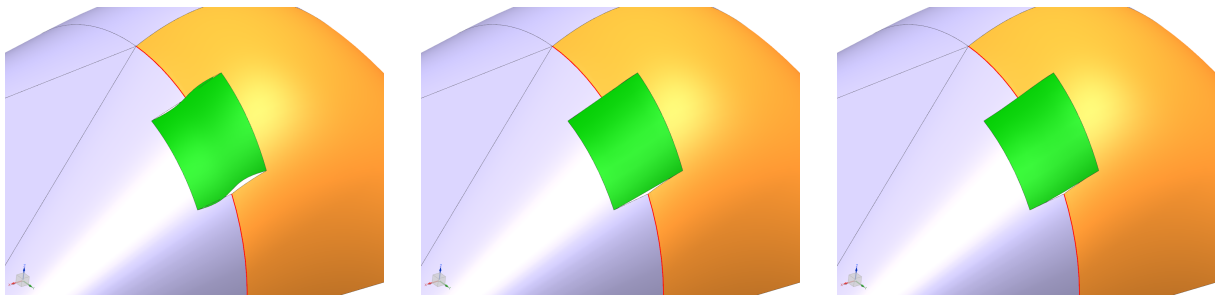


Abb. 4.12: CAD-Modell mit rot markierter  $G^1$ -stetiger Kurve, an der es zu ungewollten Einbrüchen in der Vorschubgeschwindigkeit kommt. Dieses Modell dient als Ausgangslage, um verschiedene Verfahren zur Reparatur dieser Randkurve zu vergleichen.

Die erste Variante wurde mit der Funktion *Through Curves* im CAD-System NX erstellt. Bei der Konstruktion wurde sowohl an der Start- als auch der Endkurve  $G^2$ -Stetigkeit gefordert. Bei näherer Betrachtung der erzeugten Fläche (siehe Abbildung 4.13a) fällt auf, dass diese leicht geschwungen ist und nicht den Schnittkurven folgt. Der Randeffect ist bei dieser Fläche derart ausgeprägt, dass die Fläche an einer der beiden Seiten sogar in die alte, bereits existierende Geometrie läuft. Vergleicht man die Ergebnisse des  $G^2$ -Algorithmus mit (siehe Abbildung 4.13b) und ohne (siehe Abbildung 4.13c) Minimierung stellt man visuell im CAD-Modell kaum Unterschiede fest.



(a) Variante 1: NX-erzeugte Fläche (b) Variante 2:  $G^2$ -Algorithmus ohne Minimierung (c) Variante 3:  $G^2$ -Algorithmus mit Minimierung

Abb. 4.13: Überblick über die zu vergleichenden Varianten zur Erzeugung eines  $G^2$ -stetigen Flächenpatches. Links ist das Resultat der Methode *Through Curves* aus NX zu sehen, während in der Mitte und auf der rechten Seite die Ergebnisse des in MATLAB implementierten Algorithmus zur Erzeugung  $G^2$ -stetiger Flächen ohne und mit Minimierung eines Energiefunktional zu sehen ist.

Im nächsten Schritt werden die maximalen Krümmungen, die auf den Flächen auftauchen, mit-

einander verglichen. Hierbei ist zu beachten, dass das Funktional

$$\mathcal{F} = \int_{[F_u]} \int_{[F_v]} \|F_{uu}(u, v)\|_2^2 + 2\|F_{uv}(u, v)\|_2^2 + \|F_{vv}(u, v)\|_2^2 \, du \, dv$$

nicht zwangsläufig die Krümmungen in den jeweiligen Parameterrichtungen minimiert. Je näher die zugrundeliegende Parametrisierung einer isometrischen Parametrisierung gleicht, umso stärker korrelieren auch die Normen der zweiten Ableitung mit den Krümmungen in die entsprechende Richtung. Liegt der Patch wie in diesem Beispiel mit gleichmäßigen Abständen der Start- und Endkurve zur problematischen Kurve vor und besitzt darüber hinaus keine Singularitäten, liefert die Minimierung von  $\mathcal{F}$  im Allgemeinen sehr häufig gute Resultate. In Abbildung 4.14 ist die maximale Krümmung farblich für alle drei Varianten dargestellt. Zur Visualisierung wurde die Funktion *Radius* [1, Kapitel 6.2] verwendet, mit der man Flächenkrümmungen durch Farbverläufe charakterisieren und beurteilen kann. Eine solche Analyse im CAD-System dient keineswegs der Bestimmung exakter Krümmungswerte, sondern vielmehr einer visuellen Kontrolle der Flächenbeschaffenheit.

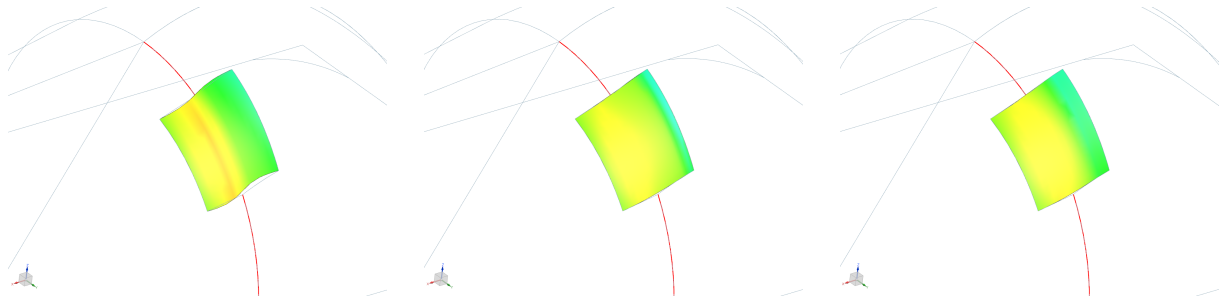


Abb. 4.14: Visualisierung der maximalen Krümmung auf den drei verschiedenen Flächen. Der Farbverlauf beginnt bei blau für geringe Krümmungen und endet bei rot für große Krümmungen. Die in NX erzeugte Fläche weist einen schmalen rötlichen Bereich in der Mitte des Patches auf, indem die Krümmung plötzlich stark ansteigt. In der mittleren Abbildung fällt auf, dass der Übergang rechts schnell von gelblich auf türkis wechselt, was auf rasche Änderungen im Krümmungsverhalten hindeutet. Durch die Optimierung auf der rechten Seite ist der Übergang im Gesamtbild homogener und glatter im Krümmungsverlauf.

Während die mit NX erstellte Lösung einen „Krümmungsknick“ aufweist, der in Abbildung 4.14 links als leichte Rotschattierung zu sehen ist, sind die beiden Resultate des in MATLAB implementierten  $G^2$ -Algorithmus deutlich homogener. Mit der zusätzlichen Minimierung von  $\mathcal{F}$  sind die Übergänge am rechten Rand etwas weicher. Das ist daran zu erkennen, dass der grüne Farbverlauf weiter in das Flächeninnere verläuft und nicht so abrupt in gelb wechselt, wie in der Mitte von Abbildung 4.14.

Die vorherigen Betrachtungen der Geometrien im CAD-System lassen bereits erahnen, dass sich die durch das CAM-System generierten Werkzeugbahnen qualitativ unterscheiden. Um dies zu überprüfen, wurden für alle drei Varianten die realen Vorschubgeschwindigkeiten mithilfe einer Maschinensimulation berechnet. Im Falle der in NX erzeugten Freiformfläche ist der Randeffekt deutlich sichtbar, wie die nachfolgenden Abbildungen 4.15 und 4.16 zeigen.

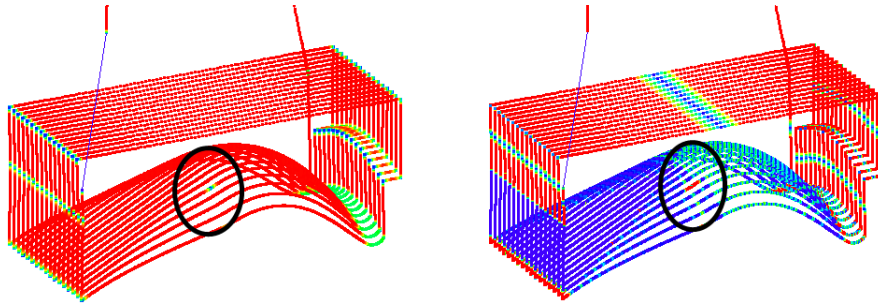


Abb. 4.15: Links ist die reale Vorschubgeschwindigkeit einer Fräsmaschine beim Abfahren der Werkzeugbahnen mithilfe einer Software visualisiert. Sowohl an der ersten als auch der letzten Bahn, die gerade noch auf dem Patch liegen, ist ein Geschwindigkeitseinbruch durch dunklere Farbsegmente sichtbar. Auf der rechten Seite ist die Gesamtbeschleunigung aller Achsen der Werkzeugmaschine zu sehen.

Bei der visuellen Bewertung der Vorschubgeschwindigkeiten aus den anderen beiden Varianten ist mit dieser Darstellung bereits kein Unterschied mehr erkennbar. Der Fräser fährt mit maximaler Vorschubgeschwindigkeit über den Flächenpatch hinweg. Bei genauer Betrachtung der Beschleunigung (rechts in Abbildung 4.16) fällt auf, dass im Bereich des Patches weniger helle Stellen zu sehen sind. Das ist ein Indikator dafür, dass die Beschleunigung in diesem Bereich deutlich geringer (nahe 0) ist und die Bahngeschwindigkeit damit gleichmäßiger wird.

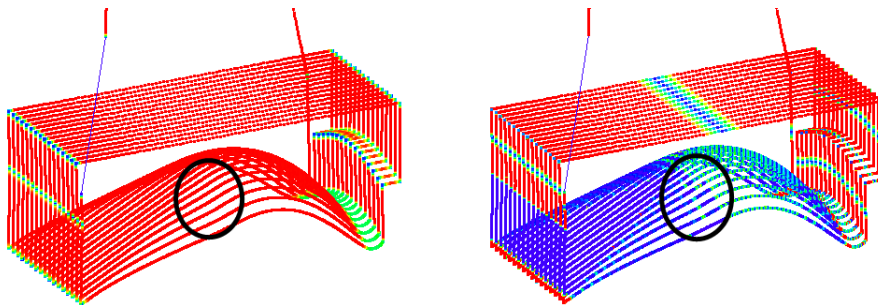


Abb. 4.16: Links ist die reale Vorschubgeschwindigkeit einer Fräsmaschine beim Abfahren der Werkzeugbahnen zu sehen. Über die  $G^2$ -stetige Fläche hinweg bleibt der Vorschub maximal und konstant. Rechts ist die Beschleunigung des Bahnwegs abgebildet. Im Gegensatz zur ersten Variante ist hier im Falle des  $G^2$ -Patches mit Optimierung im Bereich des Übergangs weniger helle Stellen erkennbar. Das bedeutet, dass die Beschleunigung konstant bei fast  $0 \frac{mm}{s^2}$  (= blau) lag.

Vergleicht man die Resultate des hier vorgestellten Algorithmus mit einer herkömmlichen Konstruktionsmethode aus dem CAD-System NX, so fällt vor allem auf, dass die Vermeidung von Randeffekten durch geeignete Projektionen nicht nur visuell ansprechende Flächen erzeugt, sondern auch bezüglich der Korrektur bereits erzeugter Werkzeugbahnen deutlich stabilere Ergebnisse liefert. Die zusätzliche Minimierung von  $\mathcal{F}$  liefert an dieser Stelle ein spezielles Feintuning, das nur durch tiefgreifendere Analysen der Achspositionen und -geschwindigkeiten in MATLAB sichtbar gemacht werden kann.



Im praktischen Umfeld der industriellen Fertigung mit Multiachs-Kinematiken wie z.B. 5-Achs-Fräsmaschinen stellt sich häufig die Frage nach möglichen An- bzw. Abfahrstrategien, d.h. wie der Fräser aus der Luft auf das Werkstück auf- bzw. abfährt. Betrachtet man diese Problemstellung im Kontext der spanenden Oberflächenbearbeitung, ist das Ziel, möglichst stetige Bewegungen zum An- und Abfahren der zu bearbeitenden Fläche zu finden. Diese Anforderung ist in der Praxis äußerst komplex und wird im ersten Teil dieses Kapitels kurz beschrieben. Anschließend wird eine Methode aus dem CAD-System NX vorgestellt, die es dem Benutzer ermöglicht, Flächen nachträglich zu verlängern, um damit An- und Abfahrbewegung zu erzeugen. Diese Operation verändert allerdings die zugrundeliegende Parametrisierung der entsprechenden Fläche und ist damit keine lokale, sondern vielmehr eine globale Methode. Durch wenige Anpassungen des in Kapitel 4 vorgestellten Algorithmus ist es möglich, eine lokale Lösung für diese Problemstellung zu erzeugen. Den Abschluss dieses Kapitels bilden die Auswertung und Gegenüberstellung der Resultate der verschiedenen Methoden eines im Rahmen dieser Arbeit durchgeführten Maschinentests mit einer 5-Achs-Fräsmaschine.

## 5.1 Problemstellung

In Abbildung 5.1 ist ein exemplarisch aus einem CAD-Modell erzeugtes NC-Programm zu sehen. Um insbesondere am Rand der zu fertigenden Oberfläche eine hohe Abtragsqualität zu erreichen, ist es in der Praxis wünschenswert, einen kurzen Anfahrweg einzuplanen, sodass das Werkzeug „glatt“ auf die Fläche auffährt. Die etwas unpräzise Formulierung „glatt“ meint in diesem Fall, dass der Verlauf von Geschwindigkeit, Beschleunigung und Ruck der tatsächlichen Werkzeugmaschine (bzw. ihrer einzelnen Achsen) möglichst stetig ist. Das hier berechnete NC-Programm, z.B. bestehend aus einer Folge von Positionen  $(x_i, y_i, z_i)^T$  und Orientierungen  $(a_i, b_i, c_i)^T$ , wird von einer numerischen Steuerung verarbeitet. Mithilfe einer kinematischen Transformation und weiterer Verarbeitungsschritte werden aus den Positionen und Orientierungen des CAM-Systems Bahnkurven und -verläufe abgeleitet, die die Achspositionen der realen Maschine festlegen. Obwohl an dieser Stelle für einen stetigen Verlauf der Achspositionen weitaus mehr zu beachten ist (z.B. Singularitäten der Maschinenachsen, Achslimits, etc.), ist es naheliegend, dass sich die qualitative Güte des NC-Programms aus dem CAM-System in den realen Achsbewegungen widerspiegelt.

Da das vom CAM-System erzeugte NC-Programm aus Positionen und Orientierungen besteht, lässt sich in diesem Zusammenhang nicht von stetigen bzw. unstetigen Bahnen sprechen. Weisen benachbarte Punkte allerdings bereits starke Umorientierungen auf kurzen Abschnitten auf oder besitzt ein NC-Programm eine stark inhomogene Abtastung, ist im Allgemeinen davon auszugehen, dass die Profile für die Maschinenachsen an diesen Stellen Dynamiksprünge aufweisen. Damit an der Werkzeugmaschine insbesondere zum Zeitpunkt des ersten Kontakts zwischen Fräser und Oberfläche Unstetigkeiten in Geschwindigkeit, Beschleunigung oder Ruck möglichst vermieden werden, wurde auf der rechten Seite von Abbildung 5.1 eine tangentielle Anfahrbewegung durch das CAM-System erzeugt.

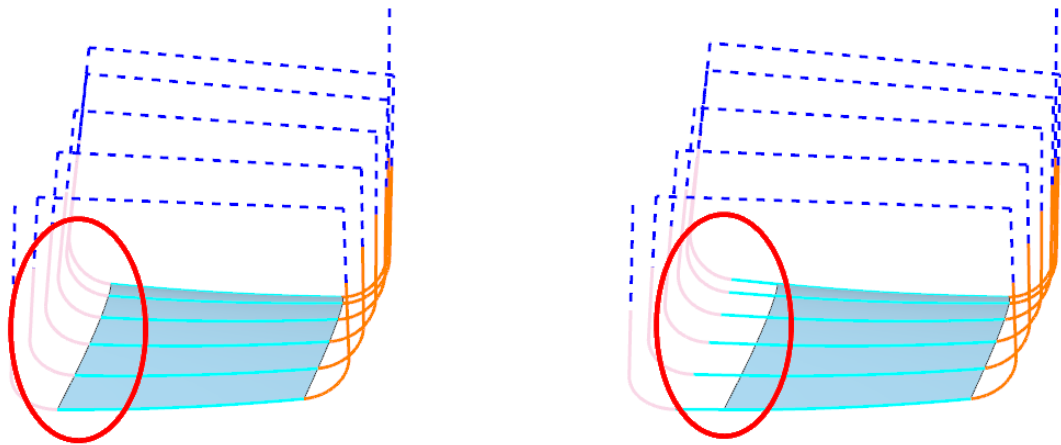


Abb. 5.1: Mit dem CAM-System NX exemplarisch erzeugtes NC-Programm: In der linken Graphik taucht das Werkzeug direkt in die zu bearbeitende Fläche ein, während auf der rechten Seite ein kurzer, tangentialer Anfahrweg eingeplant ist, damit die Geschwindigkeiten der Maschinenachsen beim Auftreffen auf die Fläche möglichst stetig sind.

Eine solche tangentiale Anfahrt liefert zwar  $G^1$ -stetige Positionen  $(x_i, y_i, z_i)^T$ , allerdings auch einen Sprung in der Winkelgeschwindigkeit der Werkzeuganstellungen  $(a_i, b_i, c_i)^T$ . Um diesen Sachverhalt besser zu visualisieren, wurden obige Werkzeugbahnen mit tangentialer Anfahrt simuliert. Als Maschinenmodell wurde eine 5-Achs-Fräsmaschine, wie sie in Abbildung 5.2 zu sehen ist, verwendet. Diese besitzt neben den drei Geoachsen  $X, Y$  und  $Z$  einen drehbaren Werkstücktisch (lila) der auf einem Schwenktisch (grün) aufgebracht ist. Oftmals werden die Rundachsen bezogen auf ihre Drehachsen mit den Buchstaben  $A, B$  und  $C$  bezeichnet, wobei  $A$  einer Drehung um die  $X$ -Achse,  $B$  einer Drehung um die  $Y$ -Achse und  $C$  einer Drehung um die  $Z$ -Achse entspricht.

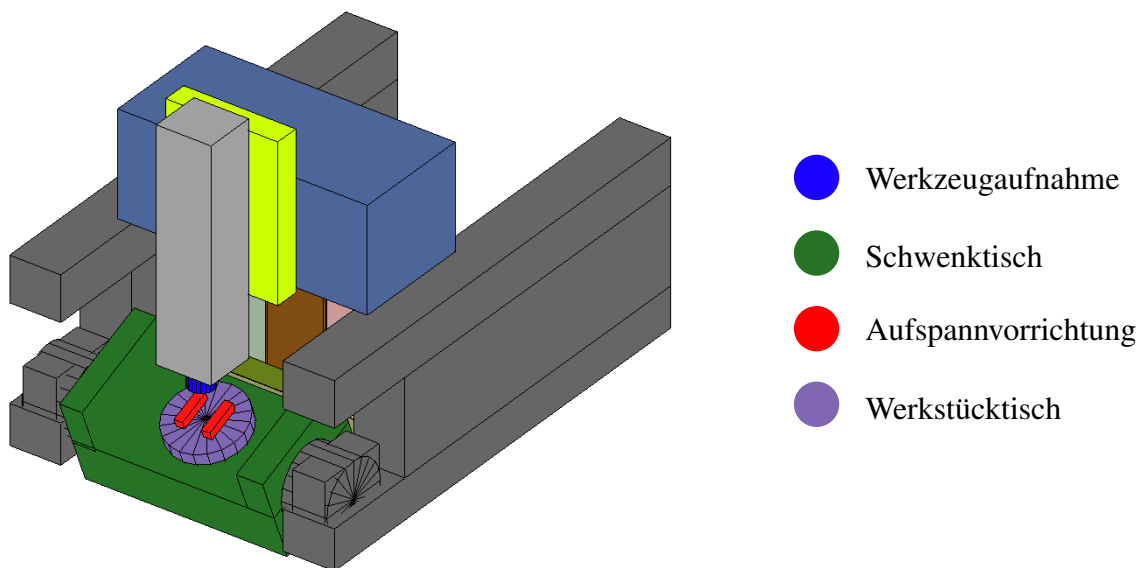


Abb. 5.2: Maschinenmodell einer 5-Achs-Fräsmaschine mit AC-Kinematik zur Verfügung gestellt von Lisa Buchner, Universität Passau/SIEMENS AG.

In Abbildung 5.3 ist links die Beschleunigung in der Rundachse  $C$  in blau abgebildet. In grün ist die physikalische Grenze der Beschleunigung dargestellt. Betrachtet man die Werte mit dem NC-Programm auf der rechten Seite, stellt man fest, dass es genau am tangentialen Übergang zu einer Maximalbelastung (bzgl. der Beschleunigung) der  $C$ -Achse kommt. Links vom Übergang ist sowohl die Geschwindigkeit als auch die Beschleunigung der Rundachse 0. Am Übergangspunkt springt dann die Geschwindigkeit auf einen konstanten Wert, was eine unendliche Beschleunigung zur Folge hat.

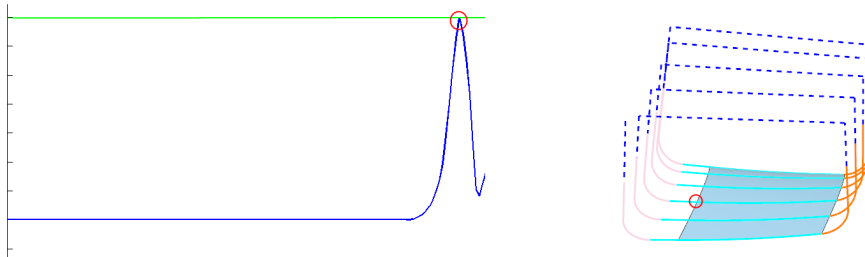


Abb. 5.3: Die Bahnpunkte rechts wurden mit einer 5-Achs-Kinematik simuliert. Links sind die Beschleunigungswerte der Rundachse  $C$  abgebildet. Es fällt auf, dass am rot markierten (tangentialen) Übergangspunkt der maximal zugelassene Beschleunigungswert angenommen wird, d.h. die Maschinen-Achse ist an diesem Punkt einer maximalen Belastung ausgesetzt. Der Grund hierfür liegt darin, dass die Werkzeuganstellung während des gesamten tangentialen Anfahrwegs konstant bleibt und beim Auftreffen auf die Fläche in eine konstante Bewegung überführt wird. Die Folge dieser Unstetigkeit ist, dass die Maschine an dieser Stelle stark abbremsen wird.

Solche Unstetigkeiten führen in den meisten Fällen zu einem kurzzeitigen Einbruch der Vorschubgeschwindigkeit, der auf der zu bearbeitenden Oberfläche sichtbar wird. Ein in der Praxis häufig genutztes Hilfsmittel zur Erzeugung von besseren NC-Programmen ist die künstliche Erweiterung der entsprechenden Fläche im CAD-System vor der Bahnplanung. Dies hat den Vorteil, dass die zugrundeliegende Geometrie, aus der das CAM-System Positionen und Orientierungen erzeugt, eine hohe Stetigkeitsordnung besitzt. Hierbei bietet das CAD-System NX z.B. die Funktionalität *Trim and Extend*, die im Folgenden näher beschrieben wird.

## 5.2 Trim and Extend in NX

Das Prinzip der Funktionalität *Trim and Extend* (siehe [1, Kapitel 4.5]) ist trotz unterschiedlicher Umsetzungen in verschiedenen CAD-Systemen weitestgehend identisch. Ausgehend von einer bereits gegebenen Fläche wird eine Kante ausgewählt, zu der die Fläche durch eine Quertangente oder durch ein Krümmungskreissegment fortgesetzt werden soll. Die Funktionalität *Trim* erlaubt es, Flächen beliebig komplex zuzuschneiden, indem Kurven im Parameterbereich definiert werden, die einen (oder mehrere) Bereiche umranden, die dann im Bildbereich angezeigt werden. Damit ist es z.B. möglich, eine Dreiecksfläche mit einer Tensorprodukt-Fläche, deren Parameterbereich rechteckig ist, darzustellen, ohne dabei Singularitäten durch degenerierende Kanten zu erzeugen. In Abbildung 5.4 ist eine getrimmte Fläche exemplarisch dargestellt. Auf der linken

Seite wurde der Parameterbereich mit drei begrenzenden Kurven (orange) getrimmt. Das Resultat im Bildraum ist rechts dargestellt. Bei einer solchen Operation würde man im CAD-Modell die zugrundeliegende rechteckige Fläche nicht mehr sehen und es würde nur die Dreiecksfläche verbleiben.

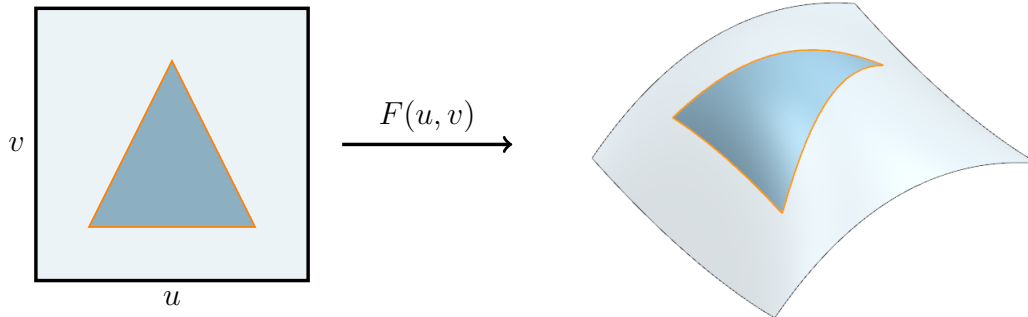


Abb. 5.4: Links ist der rechteckige Parameterbereich mit orange dargestellten Trim-Kurven dargestellt. Im CAD-Modell auf der rechten Seite kann damit eine dreidimensionale Dreiecksfläche dargestellt werden, ohne an den Ecken Singularitäten zu erhalten. In diesem Beispiel ist rechts aus Gründen der Übersichtlichkeit die ursprüngliche Fläche noch halbdurchsichtig abgebildet.

Wie der Name bereits vermuten lässt, ist es nicht nur möglich, Flächen zu verkleinern, sondern auch zu vergrößern. Hierbei tritt allerdings häufig ein zur Trim-Operation gegenteiliger Effekt auf. Intuitiv würde man erwarten, dass bei der Erweiterung einer Kante der korrespondierende Teil im Parameterbereich erweitert wird. Tatsächlich allerdings wird der gesamte Parameterbereich in jeder Richtung erweitert. In Abbildung 5.5 ist auf der linken Seite eine Fläche mit Gesamtlänge  $25\text{mm}$  abgebildet, die entlang der orangen Kante  $5\text{mm}$  krümmungsstetig fortgeführt werden soll. Während das entsprechende, in grau hinterlegte Kontrollpolygon auf der linken Seite exakt zur Fläche passt, ist deutlich zu erkennen, dass nach Anwendung der Extend-Funktionalität der Parameterbereich in allen Richtungen erweitert wurde. Neben den entstandenen Trim-Kurven, die den Parameterbereich einschränken, wurde auch die zugrundeliegende Parametrisierung verändert.

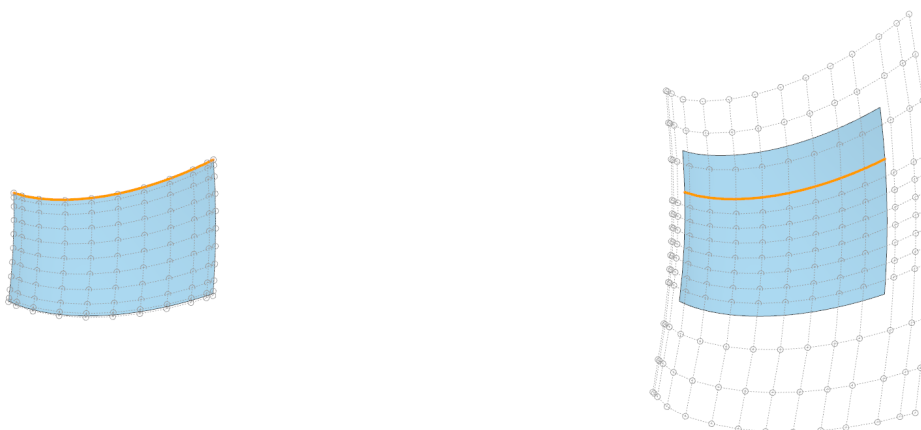


Abb. 5.5: Kontrollpolygon einer Fläche im CAD vor und nach der Extend-Operation: Es ist deutlich zu erkennen, dass nach durchgeführter Operation die Fläche zwar wie gewollt erweitert wurde, allerdings nun approximativ als Trim-Fläche vorliegt.

Eine solche grundlegende Veränderung kann, abhängig von der verwendeten Verfahrenmethode, Auswirkungen auf sämtliche Werkzeugbahnen des nachgelagerten CAM-Prozesses haben.

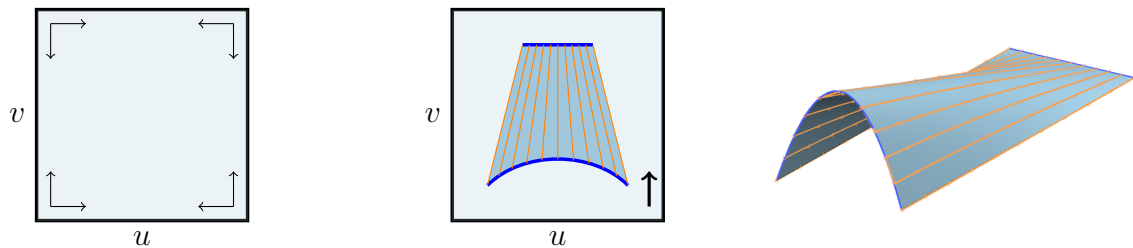
## 5.3 CAM Verfahrenmethoden

Es gibt sehr viele Möglichkeiten, um NC-Programme ausgehend von Flächengeometrien zu erzeugen. Eine sehr intuitive und anschauliche Variante ist die Projektion ausgewählter Randkurven auf eine frei wählbare Bezugsebene. Anschließend werden (zweidimensionale) Bahnen nach einem benutzerdefinierten Muster (ZIG, ZIGZAG, etc.) auf dieser Ebene erzeugt, die dann ihrerseits wieder auf die Fläche(n) projiziert werden. Je nach Flächenbeschaffenheit und CAM-Einstellungen (Flächenkrümmungen, Bezugsebene, Projektionsvektor) variieren die resultierenden Bahnen in Qualität und Homogenität. Da die Freiformfläche aus Abbildung 5.5 Teil einer komplexeren Geometrie ist, eignet sich die Erzeugung von Bahnpunkten durch Projektionen in diesem Falle nicht. Eine, in einem solchen Fall gängige, weitere Abfahrstrategie erhält man durch die Verfahrenmethode *Surface Area*.

### Surface Area

Diese Verfahrenmethode ist insbesondere bei komplexen Geometrien in Verbindung mit 5-Achs-Werkzeugmaschinen sinnvoll, da die Projektion auf eine Bezugsebene eben jene Vorteile und Flexibilität (wie z.B. Erreichbarkeit, Werkzeuganstellung, etc.) wieder einschränkt. Der Benutzer beginnt mit der Auswahl einer oder mehrerer an ihren Flächenübergängen mindestens  $G^0$ -stetigen Flächen des CAD-Modells. Intern werden die Kanten des Bildbereichs im Parameterbereich lokalisiert und acht Richtungen zur Auswahl angeboten. Nach entsprechender Bestätigung werden dann die Kanten im Parameterbereich miteinander verbunden, mit vorgegebenen Schwellwerten abgetastet und im Bildbereich ausgewertet. Für ungetrimmte Flächen bedeutet das, dass die Ränder des Parameterbereichs linear miteinander verbunden werden. Ist die Fläche allerdings getrimmt, sind sowohl Start- als auch Endkante bereits selbst approximierende Trimkurven im Parameterbereich. In Abbildung 5.6 ist auf der linken Seite ein rechteckiger Parameterbereich mit allen möglichen Verfahrrichtungen dargestellt. In der Mitte ist bereits eine Richtung ausgewählt und die Kanten der getrimmten Fläche werden linear miteinander verbunden. Auf der rechten Seite sind die resultierenden Fräsbahnen im CAD-Modell abgebildet.

Offensichtlich ist diese Art der Bahngenerierung in höchstem Maße abhängig von der zugrundeliegenden Parametrisierung. Eine isometrische Parametrisierung würde aufgrund ihrer Winkel- und Längentreue in diesem Fall homogen verteilte Punkte liefern. Betrachtet man nun die eingangs erklärte Problemstellung der stetigen An- und Abfahrbewegungen, stellt man fest, dass die lokale Operation *Extend* durch die Veränderung der globalen Parametrisierung Einfluss auf alle Werkzeugbahnen hat. Das Ziel einer solchen Flächenerweiterung ist es allerdings nicht innerhalb einer gewissen Toleranz sämtliche Bahnen zu verändern, sondern den Einfluss auf die Fläche und damit auf die Bahnen möglichst lokal zu halten.



(a) Acht Parameterrichtungen      (b) CAM-Bahnen Parameterbereich      (c) CAM-Bahnen CAD-Modell

Abb. 5.6: In obiger Abbildung sind links die acht möglichen Richtungen der CAM-Bahnen im Parameterbereich abgebildet. In der Mitte ist die Fläche im Parameterbereich inklusive der CAM-Bahnen und der entsprechenden Richtung dargestellt und rechts ist die Fläche im Bildbereich, wie sie im CAD-Modell erscheint, zu sehen.

## 5.4 Lokale $C^2$ -stetige Flächenerweiterung in einer Parameterrichtung

Um eine lokale Lösung zur Erweiterung einer ungetrimmten Tensorprodukt-Fläche zu erhalten, reicht es aus, den im Kapitel 4 vorgestellten Algorithmus an wenigen Stellen entsprechend anzupassen. Als Grundlage dient das in [51] vorgestellte Verfahren. Bevor die einzelnen Verfahrensschritte erläutert werden, gibt die nachfolgende Auflistung einen kurzen Überblick über alle notwendigen Änderungen.

### 1. Randbedingungen

Anstatt wie bisher entlang zweier Splinekurven  $s_F$  und  $s_G$   $G^2$ -Stetigkeit zu erzwingen, genügt es in diesem Fall, die Fläche entlang einer einzigen Kante krümmungsstetig fortzusetzen.

### 2. Anzahl an Kontrollpunktzeilen

Im Gegensatz zum vorherigen Kapitel werden keine inneren Knoten des Knotenvektors  $H.T_v$  benötigt, um pro Übergang drei Kontrollpunktzeilen zur Erzeugung eines  $G^2$ -stetigen Übergangs zur Verfügung zu haben. In diesem Fall kann die minimale Anzahl von vier Kontrollpunktzeilen für kubische Tensorprodukt-Flächen verwendet werden.

### 3. $cbd$ -Punkte sind vorgegeben

Da die ungetrimmte Fläche entlang einer Kante in einer Parameterrichtung erweitert werden soll, ergeben sich die  $cbd$ -Punkte bereits durch die Berechnung der Ableitung in derjenigen Parameterrichtung, die nicht in Kantenrichtung verläuft. Durch erneutes Ableiten erhält man ebenso die  $cbd^2$ -Punkte.

### 4. Keine Freiheitsgrade

Mit der entsprechenden Konstruktion ist es möglich, den Übergang auf Kosten der Freiheitsgrade  $C^2$ -stetig zu erzeugen. Dies ist ein wesentlicher Unterschied zum bereits vorgestellten Algorithmus und wird im späteren Verlauf von großer Bedeutung sein.

## 5.4.1 Algorithmus - Überblick und Lösungsstrategie

Analog zu Kapitel 4 werden hier zunächst ein allgemeines Ablaufschema vorgestellt sowie Konventionen bzgl. der Notation festgelegt. Ausgangslage ist eine ungetrimmte, kubische Tensorprodukt-Fläche  $F$ , die entlang einer Kante  $s_F$  in einer Parameterrichtung  $v$  erweitert werden soll.

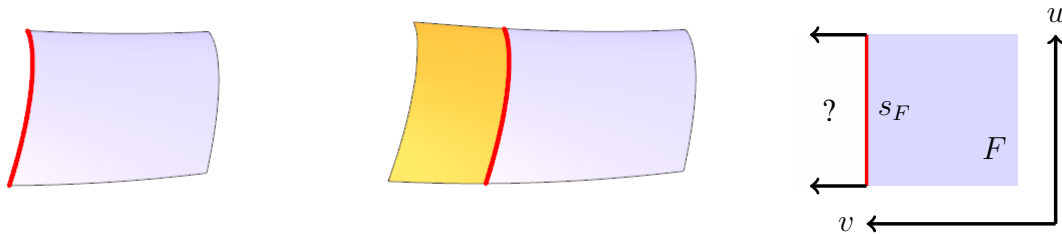


Abb. 5.7: Schema einer  $C^2$ -stetigen Erweiterungsfläche entlang einer Kante. Links ist die Ausgangsgeometrie zu sehen, die entlang der rot markierten Kante krümmungsstetig erweitert werden soll. In der Mitte ist die gewünschte Zielfläche in orange dargestellt und auf der rechten Seite ist die Fläche im Parameterbereich mit Namenskonventionen dargestellt.

Es wird vereinbart, dass die Fläche immer derart parametrisiert ist, dass die Erweiterungsfläche entlang der  $v$ -Richtung verläuft und die zugrundeliegende Geometrie am maximalen Wert des Parameterbereichs in  $v$ -Richtung fortgesetzt werden soll. Diese Annahme stellt keine Einschränkung dar, da durch Translation und Skalierung des Parameterbereichs die Fläche immer auf diese Form gebracht werden kann. Der neue, ebenfalls kubische, Patch wird analog zum vorherigen Kapitel mit  $H$  bezeichnet. Die Splinekurve, entlang derer die Fläche  $F$  fortgesetzt werden soll, wird mit  $s_F$  bezeichnet. Da es sich dabei nach obigen Konventionen um eine spezielle Randkurve von  $F$  handelt, hat  $s_F$  folgende Darstellung

$$s_F(u_{i,F}) = F(u_{i,F}, \max(F.T_v)), \quad 1 \leq i \leq n.$$

Die Notation für eine geeignete Abtastung entlang von  $s_F$  aus Kapitel 4.2 vereinfacht sich deshalb.

### Konzeptionelles Ablaufschema

Bevor die einzelnen Schritte in MATLAB im Detail erläutert werden, wird im Folgenden zunächst das Konzept des Algorithmus erarbeitet, mit dem man eine Fläche lokal erweitern kann. Die erste Aufgabe, nämlich die Extraktion der erforderlichen Geometrieinformation aus dem CAD-System NX, übernimmt wieder eine DLL, die sich mit dem laufenden NX-Prozess sowie einer MATLAB-Instanz verbindet. Dem Benutzer wird über ein graphisches Interface die Möglichkeit gegeben, diejenige Kante zu bestimmen, entlang derer die ausgewählte Fläche erweitert wer-

den soll. Durch eine entsprechende Reparametrisierung kann die Problemstellung stets auf die gerade beschriebene Situation, also eine Erweiterung in  $v$ -Richtung, zurückgeführt werden. Ist die Fläche aus NX extrahiert und entsprechend reparametrisiert, besteht der nächste Schritt in MATLAB darin, die partiellen Ableitungen

$$\frac{\partial}{\partial v} F(u_{i,F}, \max(F.T_v)), \quad \frac{\partial^2}{\partial v^2} F(u_{i,F}, \max(F.T_v)), \quad \frac{\partial^3}{\partial v^3} F(u_{i,F}, \max(F.T_v))$$

für jeden Punkt der gewählten Abtastung zu berechnen (vgl. (2.9)). Mit diesen Informationen können alle vier Kontrollpunkttrien für den neuen Patch bestimmt werden. Mit einem Newton-Verfahren können die Bogenlängen in  $v$ -Richtung des neuen Patches approximiert und anschließend daraus ein Mittelwert gebildet werden. Mit dieser Approximation kann man dem Benutzer die Eingabe einer Länge für die Flächenerweiterung ermöglichen. Der letzte Schritt des Algorithmus in MATLAB besteht darin, durch dreifaches Knotenentfernen aus zwei Teilflächen eine Gesamtfläche zu erzeugen. Der Grund für diesen Merge-Vorgang liegt in der parameterabhängigen Werkzeugbahngenerierung und wird im letzten Abschnitt dieses Kapitels anhand der Ergebnisse eines Maschinentests näher erläutert. Abschließend wird die neue Flächengeometrie in das CAD-System importiert.

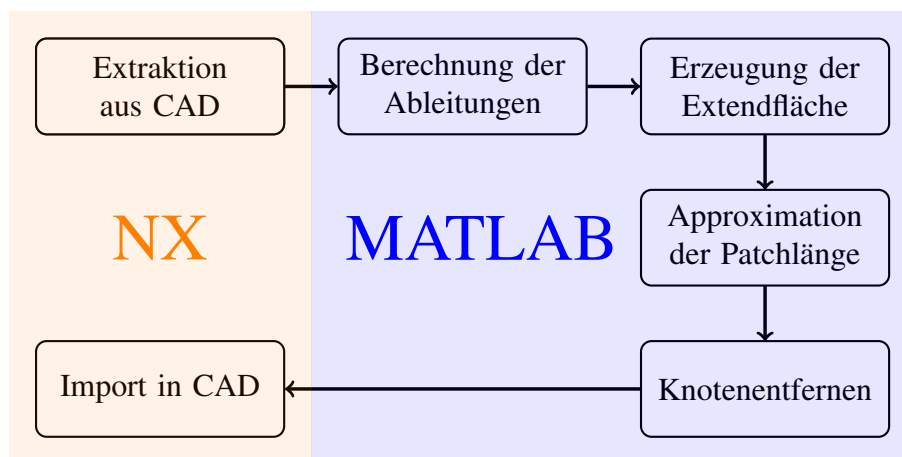


Abb. 5.8: Ablaufschema des Algorithmus zur Erzeugung einer  $C^2$ -stetigen Erweiterungsfläche. In NX muss im ersten Schritt die zu erweiternde Fläche extrahiert werden. In MATLAB werden zunächst alle notwendigen Ableitungen berechnet, bevor damit eine Extend-Fläche erzeugt wird. Durch ein Newton-Verfahren kann die Patchlänge der neuen Fläche approximiert werden und abschließend durch Knotenentfernen zu einer Flächengeometrie vereinigt werden. Im letzten Schritt wird die neue Fläche in das CAD-System importiert.

## 5.4.2 Erzeugung der Extendfläche

Das Vorgehen zur Erzeugung der Extendfläche ist dem des Algorithmus zur Erzeugung eines Patches mit  $G^2$ -Stetigkeitsbedingungen an beiden Rändern sehr ähnlich. Zunächst werden der



Grad sowie die beiden Knotenvektoren in  $u$ - und  $v$ -Richtung festgelegt

$$\begin{aligned} H.m_u &= 3, \\ H.m_v &= 3, \\ H.T_u &= F.T_u, \\ H.T_v &= [0, 0, 0, 0, a, a, a, a], \quad a \in \mathbb{R}^+ \setminus \{0\}. \end{aligned}$$

Damit ergeben sich insgesamt vier zu bestimmende Reihen des Kontrollpolygons, die iterativ, beginnend mit der ersten Reihe, approximiert werden.

### Erste Reihe des Kontrollpolygons

Die erste Reihe des Kontrollpolygons ist bereits implizit durch das Kontrollpolygon der Splinnekurve  $s_F$  gegeben, da der Knotenvektor in  $u$ -Richtung dem der Fläche  $F$  entspricht. Damit gilt

$$H.d_{(1,;,:)} = s_F.d. \quad (5.1)$$

### Zweite Reihe des Kontrollpolygons

Damit der Flächenübergang von  $F$  und  $H$  an einer Stelle

$$F(u_{i,F}, \max(F.T_v)) = H(u_{i,F}, \min(H.T_v))$$

$G^1$ -stetig ist, muss eine Reparametrisierung  $\varphi(u_{i,F}, \min(H.T_v))$  existieren, so dass

$$D^1 F(u_{i,F}, \max(F.T_v)) = D^1 (H \circ \varphi)(u_{i,F}, \min(H.T_v))$$

gilt. Mit anderen Worten muss für jeden Punkt der gemeinsamen Kante

$$\|\mathbf{J}_F(u_{i,F}, \max(F.T_v)) - \mathbf{J}_H(u_{i,F}, \min(F.T_v))\mathbf{X}(u_{i,F}, \min(F.T_v))\|_F \leq \tau$$

gelten, wobei  $\varphi'(\cdot) = \mathbf{X}(\cdot)$  ist. Dieser Zusammenhang gilt insbesondere für die Gleichheit der Jakobi-Matrizen von  $F$  und  $H$ . Aufgrund der Konstruktion des Knotenvektors und der ersten Kontrollpunktreihe von  $H$  ist die  $u$ -Ableitung beider Flächen bereits identisch. Wählt man nun die zweite Kontrollpunktreihe von  $H$  derart, dass für ihre  $v$ -Ableitungen an den jeweiligen Parameterpaaren

$$F_v(u_{i,F}, \max(F.T_v)) = H_v(u_{i,F}, \min(F.T_v)) \quad (5.2)$$

übereinstimmen, so ist der Übergang  $G^1$ -stetig und nach Bemerkung 3.21 sogar  $C^1$ -stetig. Da der Knotenvektor der Erweiterungsfläche in  $v$ -Richtung durch

$$H.T_v = [0, 0, 0, 0, a, a, a, a]$$

gegeben ist, folgt für die Ableitung der Vandermonde-Matrix

$$\mathbf{V}'_v(\min(H.T_v)) = \begin{pmatrix} -\beta \\ \beta \\ 0 \\ 0 \end{pmatrix}.$$

Damit Gleichung (5.2) erfüllt ist, muss für die zweite Reihe des Kontrollpolygons von  $H$

$$\mathbf{V}_u(U_{s_F})^T H.d_{(2,;,)}^T = (p_1 \ p_2 \ \dots)^T, \quad (5.3)$$

mit  $p_i = \frac{F_v(u_{i,F}, \max(F.T_v))}{\beta} + H(u_{i,F}, \min(H.T_v))$  gelten.

### Dritte Reihe des Kontrollpolygons

Bei der Bestimmung der dritten Reihe des Kontrollpolygons kann man wieder die Aussage des Linkage Curve Theorems [45] verwenden. Damit ein  $G^1$ -stetiger Übergang  $G^2$ -stetig ist, reicht es also aus, in einer Tangentialrichtung ungleich in Richtung der gemeinsamen Kante selbst, dieselbe Normalenkrümmungen zu besitzen. Wählt man die dritte Kontrollpunktreihe von  $H$  derart, dass die zweiten partiellen Ableitungen in  $v$ -Richtung

$$F_{vv}(u_{i,F}, \max(F.T_v)) = H_{vv}(u_{i,F}, \min(F.T_v))$$

übereinstimmen, ist der Übergang aufgrund der identischen Ableitungsvektoren (sowohl in Richtung als auch Länge) sogar  $C^2$ -stetig. Die zweite Ableitung Vandermonde-Matrix von  $H$  hat aufgrund des Knotenvektors  $H.T_v = [0, 0, 0, 0, a, a, a, a]$  die Form

$$\mathbf{V}''_v(\min(H.T_v)) = \begin{pmatrix} \delta_1 \\ -\delta_2 \\ \delta_1 \\ 0 \end{pmatrix}.$$

Für  $q_i, i = 1, \dots, n$  der Form

$$q_i = \frac{F_{vv}(u_{i,F}, \max(F.T_v)) + \delta_2 p_2 - \delta_1 p_1}{\delta_1} + H(u_{i,F}, \min(H.T_v)).$$

ergibt sich für die dritte Kontrollpunktreihe

$$\mathbf{V}_u(U_{s_F})^T H.d_{(3,;,)}^T = (q_1 \ q_2 \ \dots)^T. \quad (5.4)$$

Dem hier beschriebenen Vorgehen liegt die Anschlusskonstruktion von Stärk [51] zugrunde.

**Satz 5.1.** *Eine Flächenerweiterung  $H$  einer Fläche  $F$ , deren Kontrollpunktfolgen nach den Vorschriften (5.1), (5.3) und (5.4) erzeugt wurden, grenzt an die ursprüngliche Geometrie  $C^2$ -stetig an.*

**Beweis:** Nach der Konstruktion aus (5.1) besitzen sowohl  $F$  als auch  $H$  dieselbe Kontrollpunktreihe bei identischem Knotenvektor in  $u$ -Richtung, d.h. der Übergang ist  $C^0$ -stetig. Insbesondere stimmen auch alle höheren Ableitungen in  $u$ -Richtung entlang der gesamten Kante überein. Aufgrund der Gleichheitsbedingung der  $v$ -Ableitungen  $F_v$  und  $H_v$  in (5.3) sind die Jakobimatrizen entlang der gemeinsamen Kante identisch und  $\varphi'$  ist die  $(2 \times 2)$ -Einheitsmatrix. Damit gilt für die Reparametrisierung der Hesse-Matrix  $\mathbf{H}_H^\dagger$

$$(\mathbf{I}_3 \otimes \mathbf{X}(\cdot)^T) \mathbf{H}_H^\dagger(\cdot) \mathbf{X}(\cdot) = \mathbf{H}_H^\dagger(\cdot).$$

Nach der Konstruktionsvorschrift (5.4) sind alle zweiten partiellen Ableitungen in  $v$ -Richtung gleich und es gilt

$$F_{vv}(\cdot) = H_{vv}(\cdot)$$

entlang der gemeinsamen Kante. Damit stimmen sowohl die zweiten Ableitungen in  $u$ -Richtung als auch in  $v$ -Richtung beider Flächen überein. Es bleibt zu zeigen, dass dies auch für die gemischten Ableitungen  $F_{uv}(\cdot) = H_{uv}(\cdot)$  gilt. In dieser Betrachtung tauchen Vandermonde-Matrizen der Basisfunktionen von  $F$  und  $H$  in den jeweiligen Parameterrichtungen auf. Aus diesem Grund wird an dieser Stelle ein zusätzlicher Index eingeführt, um die Vandermonde-Matrix der entsprechenden Flächen zugeordnet werden kann. Für die Auswertung der Basisfunktionen der Fläche  $F$  in  $u$ -Parameterrichtung an Parameterstellen  $U_F$  wird folgende Notation

$$\mathbf{V}_{u,F}(U_F) = \begin{pmatrix} N_1^m(u_1|F.T_u) & N_1^m(u_2|F.T_u) & \dots & N_1^m(u_q|F.T_u) \\ N_2^m(u_1|F.T_u) & N_2^m(u_2|F.T_u) & \dots & N_2^m(u_q|F.T_u) \\ \vdots & \vdots & \ddots & \vdots \\ N_n^m(u_1|F.T_u) & N_n^m(u_2|F.T_u) & \dots & N_n^m(u_q|F.T_u) \end{pmatrix}, \quad u_i \in U_F,$$

verwendet. Die komponentenweise Betrachtung der gemischten partiellen Ableitung  $F_{uv,j}$  für  $j = 1, \dots, 3$  an einer Parameterstelle  $(u_{i,F}, \max(F.T_v))$  der gemeinsamen Kante liefert

$$F_{uv,j}(u_{i,F}, \max(F.T_v)) = \mathbf{V}'_{u,F}(u_{i,F})^T F \cdot d_{(:, :, j)}^T \mathbf{V}'_{v,F}(\max(F.T_v)).$$

Die Auswertung von  $\mathbf{V}'_{v,F}$  am  $(m+1)$ -fachen rechten Randknoten ergibt einen Spaltenvektor der Form

$$\mathbf{V}'_{v,F}(\max(F.T_v)) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -\alpha \\ \alpha \end{pmatrix},$$

d.h. zur Berechnung der gemischten partiellen Ableitung  $F_{uv}$  an der gemeinsamen Kante muss die Differenz aus vorletzter und letzter Kontrollpunktreihe (ausgewertet an der entsprechenden Parameterstelle  $u_{i,F}$ ) berechnet werden. Betrachtet man nun die gemischte partielle Ableitung

von  $H$  an derselben Stelle der gemeinsamen Kante

$$H_{uv,j}(u_{i,F}, \min(H.T_v)) = \mathbf{V}'_{u,H}(u_{i,F})^T H \cdot d_{(:,j)}^T \mathbf{V}'_{v,H}(\min(H.T_v)),$$

muss entsprechend die Differenz aus erster und zweiter Kontrollpunktreihe gebildet werden, da

$$\mathbf{V}'_{v,H}(\min(H.T_v)) = \begin{pmatrix} -\beta \\ \beta \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

gilt. Durch die Konstruktion des  $C^1$ -stetigen Übergangs stimmen entlang der gesamten Kante alle ersten Ableitungen in  $u$ - und  $v$ -Richtungen überein. Damit stimmen genau die Differenzen der letzten und vorletzten Kontrollpunktreihe von  $F$  an einer ausgewerteten Stelle  $u$  mit der Differenz der ersten und zweiten Kontrollpunktreihe von  $H$  überein, da insbesondere  $F.T_u = H.T_u$  gilt.

□

## Vierte Reihe des Kontrollpolygons

Um die letzte Reihe des Kontrollpolygons zu bestimmen, wird heuristisch vorgegangen und die dritte partielle  $v$ -Ableitung von  $F$  verwendet, da sich diese Variante in praktischen Tests bewährt hat. Die dritte Ableitung der Vandermonde-Matrix  $\mathbf{V}_v(\min(H.T_v))$  liefert

$$\mathbf{V}_v'''(\min(H.T_v)) = \begin{pmatrix} -\gamma_1 \\ \gamma_2 \\ -\gamma_2 \\ \gamma_1 \end{pmatrix}.$$

Aufgelöst nach dem gesuchten Kontrollpunkt ergibt sich für die zu interpolierenden Punkte

$$r_i = \frac{F_{vvv}(u_{i,F}, \max(F.T_v)) + \gamma_1 p_1 - \gamma_2 p_2 + \gamma_2 p_3}{\gamma_1}.$$

Die vierte Kontrollpunktreihe erhält man wieder durch Lösen eines Interpolationsproblems der Form

$$\mathbf{V}_u(U_{s_F})^T H \cdot d_{(4,,:)}^T = (r_1 \quad r_2 \quad \dots)^T$$

oder im Falle zweier Flächen in Bézier-Form mithilfe des Algorithmus von de Casteljau, wie in [51] gezeigt wurde.

### 5.4.3 Approximation der Patchlänge

Da die Größe der zu bearbeitenden Flächen variiert, ist es in der Praxis hilfreich, dem Benutzer die Möglichkeit zu geben, die Länge der Flächenerweiterung beeinflussen zu können. Dafür bietet sich z.B. die bereits vorgestellte Bogenlängenapproximation einer Splinekurve (vgl. Algorithmus 4.2) an. Dazu berechnet man (näherungsweise) die Länge der beiden Splinekurven, die man bei der Betrachtung der ersten und letzten Spalte des Kontrollpolygons von  $H$  erhält (vgl. rote Randkurven in Abbildung 5.9) und bildet daraus das arithmetische Mittel.

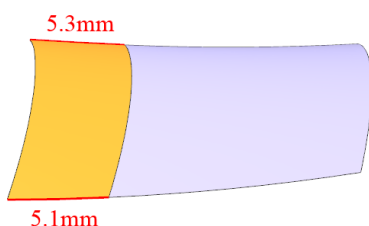
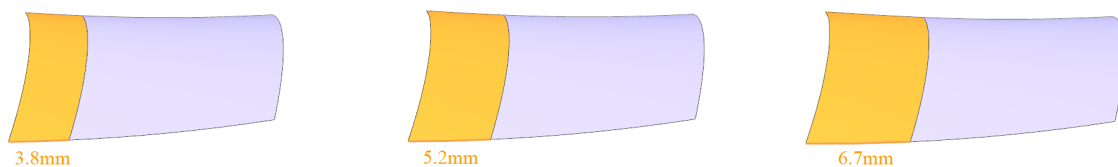


Abb. 5.9: In rot sind die beiden Randkurven der Erweiterungsfläche dargestellt. Für beide Randsplines werden die Bogenlängen approximiert und das arithmetische Mittel gebildet.

Anschließend erzeugt man den Patch  $H$  erneut und führt alle bisherigen Schritte wieder aus. Das nicht-degenerierte Knotenintervall von  $H.T_v$  wird in jedem derartigen Iterationsschritt vergrößert bzw. verkleinert, sodass man für die verschiedenen Knotenvektoren verschiedene Mittelwerte der approximierten Bogenlängen erhält. Die drei Ableitungen von  $F$  müssen hierfür allerdings nur einmal berechnet werden. In Abbildung 5.10 sind die Erweiterungsflächen nach dem hier vorgestellten Schema mit einigen fix gewählten Knotenvektoren dargestellt. Die Bogenlängen der Splinekurven am Rand jeder Erweiterungsfläche wurden mithilfe des Algorithmus 4.2 approximiert.



$$H.T = [0, 0, 0, 0, 0.4, 0.4, 0.4, 0.4] \quad H.T = [0, 0, 0, 0, 0.8, 0.8, 0.8, 0.8] \quad H.T = [0, 0, 0, 0, 1.2, 1.2, 1.2, 1.2]$$

Abb. 5.10: In dieser Abbildung sind die verschiedenen mit MATLAB berechneten Lösungen für unterschiedliche Knotenvektoren abgebildet. Für jeden Knotenvektor wurden zunächst die vier Kontrollpunktzeilen berechnet und anschließend die Bogenlängen der jeweiligen Splinekurven am Rand des neuen Patches approximiert.

Die Paare, bestehend aus der Größe des nicht-trivialen Knotenintervalls und der jeweiligen approximierten Bogenlänge, können nun in einem abschließenden Schritt approximiert werden, um dem Benutzer eine Möglichkeit zu geben, eine Länge für die Erweiterungsfläche vorzugeben.

Während Fertigungstoleranzen in diesem Kontext normalerweise im  $\mu m$ -Bereich vorgegeben werden, muss diese Genauigkeit bei An- und Abfahrbewegungen nicht zwingend eingehalten werden. Der Vorteil dieser Variante ist allerdings, dass der Benutzer ein probates Mittel hat, im CAD-System die Ausmaße der Erweiterungsfläche für seine Anforderungen ungefähr anzupassen.

### 5.4.4 Knotenentfernen - Merge beider Flächen

Aus theoretischer Sicht ist die  $C^2$ -stetige Erweiterungsfläche an dieser Stelle erzeugt und der Algorithmus somit beendet. Bei genauerer Untersuchung der erzeugten NC-Programme stellte sich allerdings heraus, dass die Ausgangsdaten des CAM-Systems am Flächenübergang kritische Stellen aufweisen, die in Abschnitt 5.5 beschrieben werden. Da der Übergang nach Konstruktion  $C^2$ -stetig ist und sowohl die ursprüngliche Geometrie als auch die Erweiterungsfläche kubisch sind, ist es möglich, die beiden Flächen durch Zusammenfügen der Knotenvektoren und anschließendes (dreifaches) Knotenentfernen ohne Informationsverlust als eine einzige Tensorprodukt-Fläche darzustellen. Im ersten Schritt müssen die beiden Kontrollpolygone von  $F$  und  $H$  zusammengefügt werden

$$\tilde{F}.d_{(:, :, j)} = \begin{pmatrix} F.d_{(:, :, j)} \\ H.d_{(2, :, j)} \\ H.d_{(3, :, j)} \\ H.d_{(4, :, j)} \end{pmatrix}.$$

Abbildung 5.11 zeigt den Grund, weswegen nur die Reihen 2-4 des in MATLAB erzeugten Patches  $H$  eingefügt werden. Die erste Reihe  $H.d_{(1, :, j)}$  ist die gemeinsame Kante selbst, die in der neu zusammengesetzten Fläche nicht doppelt auftauchen soll.

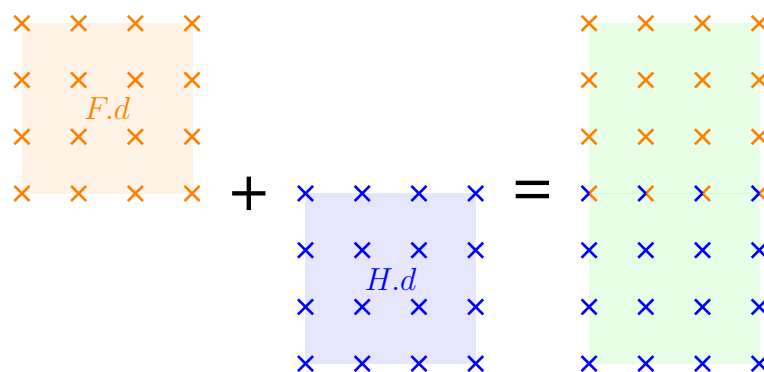


Abb. 5.11: Zusammenfügen der zwei Kontrollpolygone von  $F$  und  $H$ . Hierbei gilt es zu beachten, dass die gemeinsame Kante nicht doppelt eingefügt wird.

Der Knotenvektor  $H.T_v$  wird zunächst an das Ende von  $F.T_v$  transliert, bevor die Vektoren zu-

sammengesetzt werden. Seien die beiden Vektoren der Form

$$\begin{aligned} F.T_v &= [0, 0, 0, 0, \dots, 1, 1, 1, 1] \\ H.T_v &= [0, 0, 0, 0, a, a, a, a] \end{aligned}$$

gegeben. Der neue Vektor der zusammengesetzten Fläche besitzt vor dem Knotenentfernen die Form

$$\tilde{F}.T_v = [0, 0, 0, 0, \dots, 1, 1, 1, a + 1, a + 1, a + 1, a + 1].$$

Anschließend berechnet man die Knoteneinfügematrix  $\mathbf{A}(t)$  mit  $t = 1$  (siehe Definition 2.43) und entfernt den ersten Knoten  $t = 1$  durch Lösen der Gleichung

$$d\mathbf{A}(t)\mathbf{A}(t)^T = \tilde{F}.d\mathbf{A}(t)^T.$$

Das Kontrollpolygon und der Knotenvektor der zusammengesetzten Fläche  $\tilde{F}$  werden aktualisiert

$$\begin{aligned} \tilde{F}.d &= d, \\ \tilde{F}.T_v &= [0, 0, 0, 0, \dots, 1, 1, a + 1, a + 1, a + 1, a + 1], \end{aligned}$$

bevor obiger Schritt wiederholt und erneut eine Knoteneinfügematrix für  $t = 1$  (diesmal allerdings zum Knotenvektor  $\tilde{F}.T_v$ , der nur noch einen doppelten Knoten an der Stelle  $t = 1$  besitzt) berechnet wird. Nach diesem Prozess besitzt die neue Splinefläche einen Knotenvektor der Form

$$\tilde{F}.T_v = [0, 0, 0, 0, \dots, 1, a + 1, a + 1, a + 1, a + 1],$$

mit einem einzelnen Knoten an der Stelle  $t = 1$ . Ihre Stetigkeit bleibt an dieser Stelle unverändert

$$m - \mu = 3 - 1 = 2$$

und damit  $C^2$ -stetig. Abbildung 5.12 zeigt die zwei Flächen  $F$  und  $H$  vor dem Zusammenfügen. Die orange Fläche  $H$  wurde in MATLAB berechnet und schließt  $C^2$ -stetig an  $F$  an.

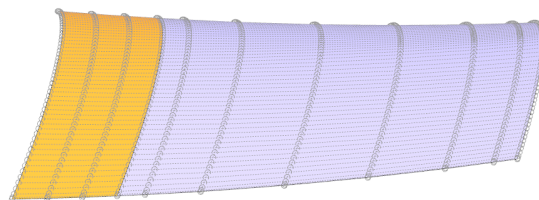


Abb. 5.12: Vor dem Zusammenfügen sind die ursprüngliche Geometrie (lila) und die in MATLAB erzeugte Fläche (orange) an ihrer gemeinsamen Kante  $C^2$ -stetig und als separate Flächen im CAD-System NX gespeichert.

Durch Konkatenation der Kontrollpolygone und Zusammenfügen der Knotenvektoren mit anschließendem Knotenentfernen kann die Fläche als eine einzige Tensorprodukt-Fläche im CAD-

System dargestellt werden.

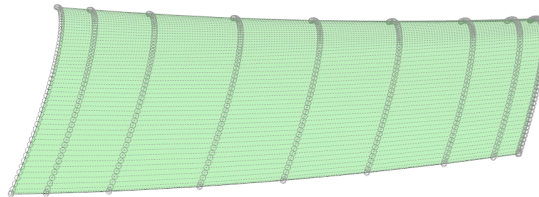


Abb. 5.13: Durch das Zusammenfügen beider Kontrollpolygone und Knotenvektoren mit anschließendem dreifachen Knotenentfernen kann die ursprünglich aus zwei Flächen bestehende Geometrie zu einer einzelnen Tensorprodukt-Fläche vereinfacht werden.

Obwohl beide Varianten im CAD-Modell unterschiedliche Knotenvektoren besitzen, ist die zugrundeliegende Geometrie stets identisch. Falls es, wie in diesem Fall, möglich ist, durch Knotenentfernen einen Flächenübergang zu vermeiden, ist diese Variante immer vorzuziehen, da das nachgelagerte CAM-System manchmal sehr anfällig auf Übergangsstellen reagiert. Im nachfolgenden Abschnitt wird genau dieser Randeffekt anhand eines Maschinentests näher untersucht.

## 5.5 Maschinentest an einer 5-Achs-Fräsmaschine

Bei der Entwicklung und Implementierung des im Zuge dieser Arbeit entstandenen Algorithmus zur  $C^2$ -stetigen Erweiterung einer Freiformfläche im CAD-System NX über die Schnittstelle NXOpen lag ein Hauptaugenmerk auf der praxistauglichen Umsetzung. Durch die Generierung verschiedener CAM-Bahnen mit NX und entsprechender Simulationen konnte die gesamte Prozesskette vom CAD-Modell bis zur Fertigung an einer realen 5-Achs-Fräsmaschine realisiert werden. Die konkrete Fragestellung, die in diesem Zusammenhang geklärt werden sollte, ergab sich bei der Auswertung und der Analyse des NC-Programms, das nach der Erzeugung der Erweiterungsfläche entstanden. Es fiel auf, dass am Übergang beider Flächen, trotz der hohen Glattheit, inhomogene Abtastungen der Bahnpunkte auftreten, die an einer echten Werkzeugmaschine mit hoher Wahrscheinlichkeit zu einem Einbruch der Vorschubgeschwindigkeit führen würden. Dieser Möglichkeit gegenübergestellt war die Vermutung, dass dieser Effekt am Übergang durch das Zusammenfügen beider Flächen zu einer einzigen Flächen (ohne Verlust von Geometrieinformationen) nicht auftritt. Ein erster Vergleich der Punkte und Orientierungen in MATLAB bekräftigte diese These. Abbildung 5.14 zeigt die euklidischen Abstände der Punkte  $(x_i, y_i, z_i)$  einer einzelnen Bahn. Während links das Zwei-Flächen Modell im CAD als Basis für die Bahnerzeugung fungiert, wurden rechts vor dem CAM-Prozess die beiden Flächen zu einer vereint.

Zwei Dinge sind bereits bei dieser Betrachtung deutlich zu sehen. Zum einen besitzt links jede Fläche eine separate Abtastung und zum anderen taucht ein doppelter Punkt am Übergang auf. Rechts hingegen ist die Fläche homogen und gleichmäßig abgetastet. In dieser Analyse fehlt allerdings eine wichtige (zu diesem Zeitpunkt noch) maschinenunabhängige Größe zum qualitativen Vergleich beider Varianten. Setzt man den Winkel der Anstellungsvektoren  $(a_i, b_i, c_i)^T$  zwei-



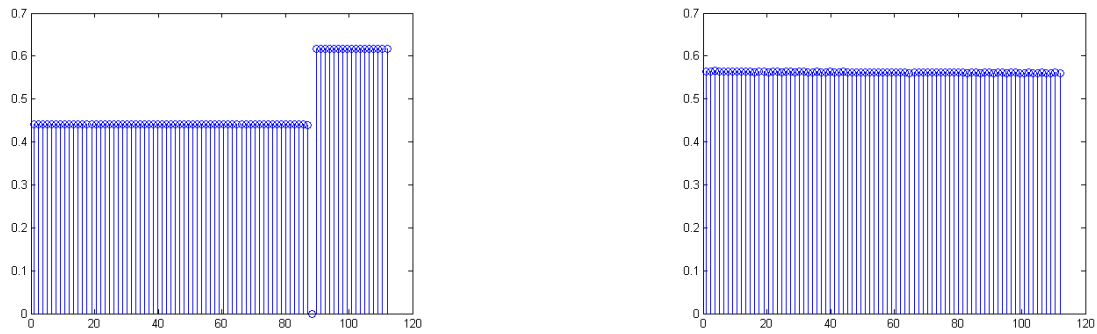


Abb. 5.14: Euklidische Abstände benachbarter Punkte auf einer einzelnen Bahn. Links ist die Analyse aus dem 2-Flächen Modell des CAD-Systems dargestellt, während rechts eine einzelne Fläche als Basis für das CAM-System dient.

er benachbarter NC-Sätze in Verhältnis zu dem euklidischen Abstand ihrer Punkte  $(x_i, y_i, z_i)^T$ , erhält man die Winkeländerungsrate pro Bahnweg.

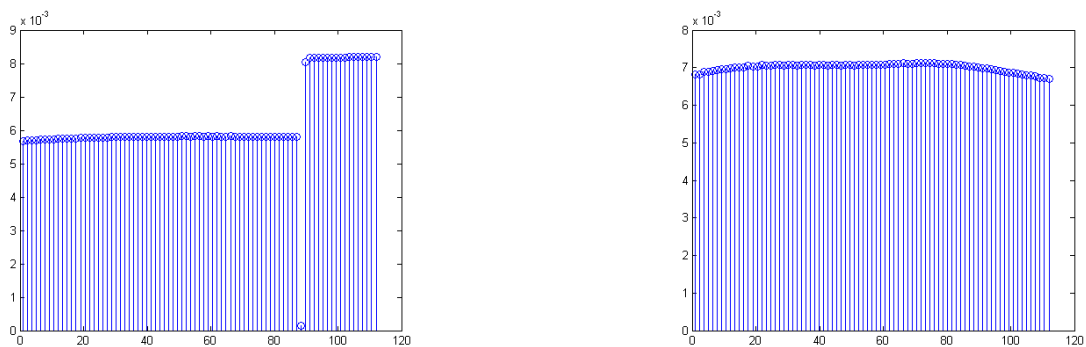


Abb. 5.15: Quotient aus Winkeländerung und euklidischen Abständen. Links wurde wieder das 2-Flächenmodell verwendet, während rechts auf eine einzelne Fläche im CAD-Modell zurückgegriffen worden ist. Der Quotient weist im ersten Fall eine Unstetigkeit an der Übergangsstelle auf. Im zweiten Fall ist der Verlauf homogen und in dieser Betrachtung unkritisch. Bemerkenswert ist an dieser Stelle, dass trotz unterschiedlicher Abtastungen der einzelnen Flächen der Quotient bis auf die Übergangsstelle konstant ist.

Trotz dieser Vorverarbeitungsschritte und Betrachtungen lässt sich an dieser Stelle noch keine konkrete Aussage über das Ausmaß und die Wirkung der kritischen (bzw. unkritischen) Stellen in den Werkzeugbahnen treffen, da die Verarbeitung und Bahnplanung des erzeugten NC-Programms durch die numerische Steuerung noch aussteht. Die folgenden Fragestellungen sollen durch einen konkreten Maschinentest geklärt werden:

- Wie stark ist der Einfluss der unterschiedlichen NC-Programme auf die Bewegung der verschiedenen Maschinenachsen?
- Ist die Flächenerweiterung im CAD-Modell am Werkstück zu sehen?
- Gibt es einen Unterschied am Werkstück bei Verwendung der zwei CAD-Modelle?

➤ Kann man mit entsprechenden Einstellungen die Ergebnisse verbessern?

Für den ersten Test wurde in MATLAB eine  $C^2$ -stetige Erweiterungsfläche, die anschließend in das CAD-System NX importiert wurde, generiert. Das in NX integrierte CAM-System sowie ein Postprozessor erzeugten auf Basis dieser Zwei-Flächen Geometrie ein NC-Programm (in Abbildung 5.17 NC-Programm 1). Als Verfahrgometrie wurde die Methode *Surface Area* verwendet. Das NC-Programm wird anschließend in die numerische Steuerung (CNC) eingespeist und an der Maschine mit üblichen Fertigungstoleranzen abgefahren. Die Kante ist auf der zu fertigenden Oberfläche deutlich zu sehen. Im zweiten Test wurden die Toleranzen maximal erhöht, um bei dem selben NC-Programm diesen Randeffect möglichst zu unterdrücken. In der Tat gelang es, insbesondere die Bahnen am Rand des Werkstücks damit zu glätten. Die Kante ist nur in Teilen des Flächenübergangs zu sehen. Allerdings sei hier erwähnt, dass derartige Toleranzen zwar visuell ein besseres Ergebnis liefern, allerdings aufgrund der maximalen Flächenabweichungen in der Praxis keine Anwendung finden. Im letzten Test wurde die nach Anwendung des hier vorgestellten Merge-Verfahrens erhaltene Flächengeometrie als Ausgangspunkt für das CAM-System verwendet. Hierbei lieferte die Steuerung bei gängigen Fertigungstoleranzen optimale Ergebnisse. In Abbildung 5.16 ist das Ergebnis des Maschinentests nach Anwendung des hier vorgestellten Algorithmus mit Knotenentfernen dargestellt. Der ursprüngliche Flächenübergang im CAD-System ist dort nicht mehr sichtbar. In Abbildung 5.17 ist der gesamte Maschinentest mit verschiedenen Varianten zu sehen. Die Marken auf den ersten beiden Varianten entstehen durch einen Einbruch der Vorschubgeschwindigkeit an der Übergangsstelle.



Abb. 5.16: Detailansicht des gefertigten Werkstücks nach Anwendung des Algorithmus zur  $C^2$ -stetigen Erweiterung einer Fläche mit anschließendem Merge-Verfahren. Eine genaue Betrachtung und Auswertung der hochauflösenden Bildern zeigt keinerlei Marke im Bereich des Flächenübergangs.

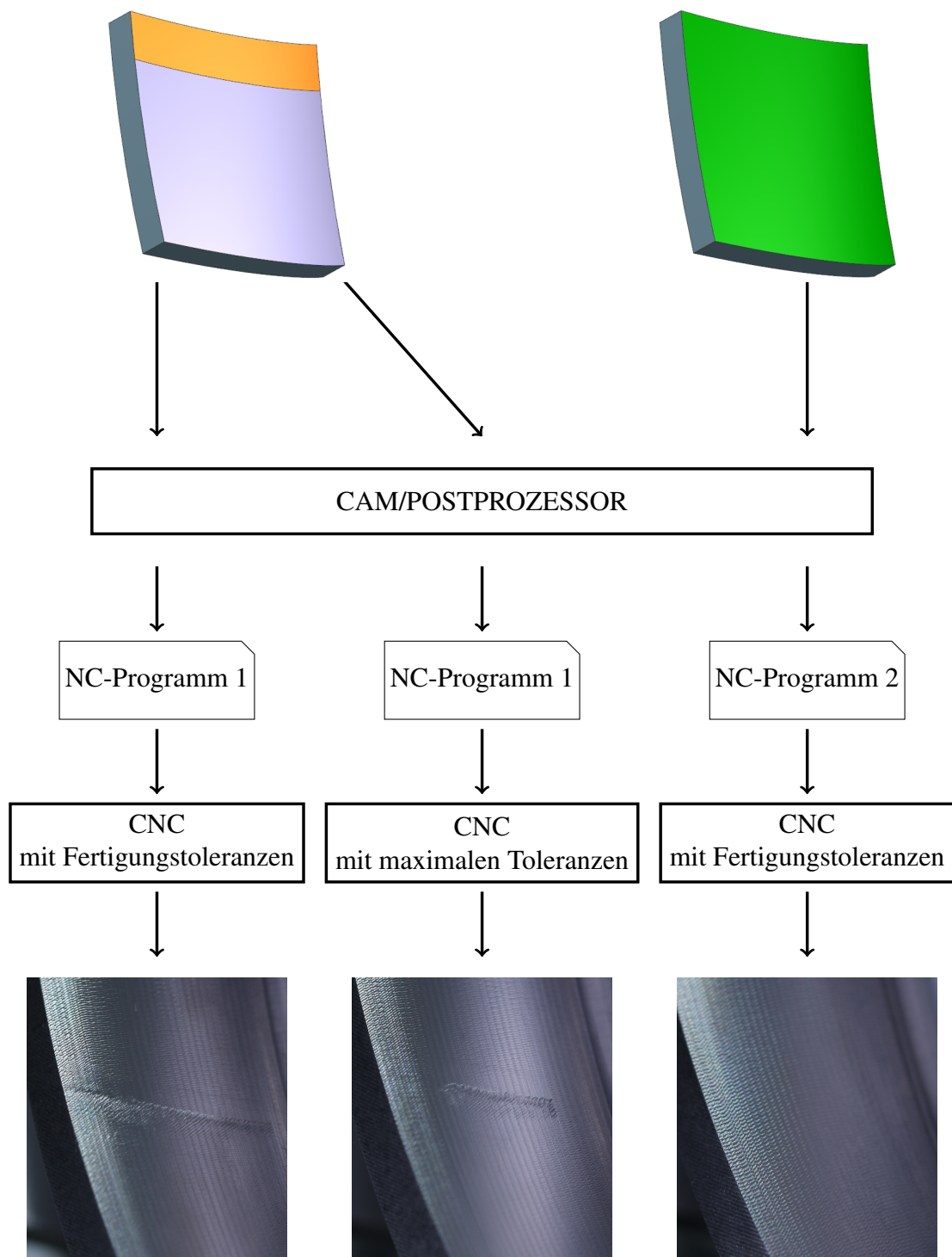


Abb. 5.17: Gesamte Prozesskette des Maschinentests. Den ersten beiden Varianten liegt eine Zwei-Flächen Geometrie im CAD-Modell zugrunde, während in der letzten Variante (rechts) nach Anwendung des hier vorgestellten Merge-Verfahrens für einen  $C^2$ -stetigen Übergang zweier Tensorprodukt-Flächen lediglich eine Fläche zur Beschreibung derselben Geometrie verwendet wurde. Ganz links ist das Ergebnis mit üblichen Fertigungsparametern zu sehen. In der Mitte wurde versucht, durch Anpassung der Toleranzen ein visuell bestmögliches Resultat zu erzielen. Verwendet man als Basis zur Beschreibung der exakt selben Geometrie nur eine Fläche, ist auf den Fotos der Werkstücke keine Marke zu sehen.

Das Ziel dieser Dissertation war es, Verfahren und Algorithmen zu entwickeln, die es vom Ausgangspunkt der numerischen Steuerung betrachtet ermöglichen, an kritischen Stellen durch gezielte Rückfrage an das CAD-System und anschließende Optimierung des Modells über eine Schnittstelle Werkzeugbahnen zu verbessern. Da es sich hierbei sehr häufig um Flächenübergänge im CAD-Modell handelt, wurden zunächst Methoden zur numerisch stabilen Prüfung auf  $G^1$ - und  $G^2$ -Stetigkeit zweier Tensorprodukt-Flächen mit einer gemeinsamen Kante vorgestellt. Die dabei eingeführte Notation ermöglichte es, diese Fragestellungen als lineare Ausgleichsprobleme zu formulieren.

Der zweite Schritt dieser Verarbeitungskette bestand darin, Werkzeugbahnen, die an den Übergangsstellen zweier Flächen Unregelmäßigkeiten aufweisen, durch Adaption des CAD-Modells zu glätten bzw. zu reparieren. Dazu wurde zunächst ein Bereich um die Kante aus dem Modell entfernt und anschließend ein Algorithmus vorgestellt, der es ermöglicht, den entstandenen getrimmten Bereich wieder zu füllen. Die dabei neu erzeugte Fläche besitzt die Eigenschaft, an Start- und Endkante  $G^2$ -stetig an die ursprünglichen Geometrien anzuschließen. Durch eine erneute Erzeugung des NC-Programms (CAM-System und Postprozessor), die ihrerseits wieder über die Schnittstelle NXOpen gesteuert wurde, können so kritische Bereiche durch neu erzeugten Code unter Verwendung des bestehenden CAM-Systems ersetzt werden. Neben der Verwendung der umfangreichen Schnittstelle NXOpen war insbesondere die Effizienz der hier entwickelten Verfahren eine Herausforderung. Abgesehen von Bogenlängenapproximationen, die durch ein Newton-Verfahren implementiert wurden, konnten alle Problemstellungen letztlich auf das Lösen linearer Gleichungssysteme zurückgeführt werden, was schnell und numerisch stabil durchgeführt werden kann.

Mit dem vorgestellten Verfahren kann ein weiteres in der Praxis auftretendes Problem gelöst werden. Durch Anpassungen und Weiterentwicklungen der Algorithmen konnten  $C^2$ -stetige Erweiterungsflächen im CAD-System erzeugt werden. Auf Basis dieser Flächen ist der Benutzer in der Lage stetige An- und Abfahrbewegungen für die Werkzeugmaschine zu generieren. Nachdem die veränderte Aufgabenstellung im CAD-Modell bearbeitet werden konnte, wurde die gesamte Prozesskette der Fertigung mit Werkzeugmaschinen durchgeführt. Dazu gehörte zunächst die Erzeugung von NC-Programmen mit CAM-System und Postprozessor und die anschließende detaillierte Analyse des Teileprogramms. Durch die Anwendung der Algorithmen in praxisgerechten Bearbeitungssituationen sowie durch im Vorfeld berechnete Simulationen, konnte ein Maschinentest im realen Fertigungsumfeld erfolgreich durchgeführt werden. Dabei konnten signifikante Verbesserungen im Bereich der Oberflächenbearbeitung erzielt werden (vgl. Abbildung 5.17).

Diese Arbeit liefert neben den vorgestellten Verfahren zur Optimierung von Werkzeugbahnen durch entsprechende Analysen und Anpassungen im CAD-Modell einen Beitrag zur gesamtgesellschaftlichen Betrachtung der Prozesskette und zeigt das enorme Potential der Verknüpfung aller einzelnen Teilkomponenten (CAD, CAM, CNC) im Zuge der Digitalisierung. Die Ergebnisse dieser Arbeit zeigen darüber hinaus, dass die Komplexität heutiger CAD-Modelle und

Computersysteme keine generische und rein computergesteuerte Lösung zulassen. Die Optimierung der Fertigungsqualität auf allerhöchstem Niveau muss für spezielle Anwendungsfälle separat erfolgen. In den nächsten Jahren ist zu erwarten, dass mehr Informationen im Bereich der Technologie und Maschinendaten in CAx-Systemen vorhanden sein werden und im Gegenzug numerische Steuerungen in Zukunft über mehr Geometrieinformationen des zugrundeliegenden CAD-Modells verfügen werden. Damit stellt die korrekte Verarbeitung und der daraus gezogene Mehrwert solcher Informationen in jedem Bereich dieser Prozesskette auch zukünftig ein wichtiges Forschungsgebiet im Bereich der industriellen Fertigung dar.

# LITERATURVERZEICHNIS

- [1] ANGER, H. E. G. IN: *Das große Freiformflächen-Buch (Free Form Features) NX 9*. OrtmannTeam GmbH, Ainring, 2014.
- [2] BARNER, J., C. GIANNELLI, D. GROSSMANN, B. JÜTTLER, G. KISS und U. ZORE: *Adaptive CAD Model (Re-)Construction with THB-splines*. Graph. Models, 76(5):273–288, 2014.
- [3] BARSKY, B. A. und D. T. D.: *Geometric Continuity, Shape Parameters, and Geometric Constructions for Catmull-Rom Splines*. ACM Trans. Graph., 7(1):1–41, Jan. 1988.
- [4] BARSKY, B. A. und A. D. DEROSE: *Geometric Continuity of Parametric Curves*. Techn. Ber., University of California at Berkeley, Berkeley, CA, USA, 1984.
- [5] BENKER, H.: *Ingenieurmathematik kompakt - Problemlösungen mit MATLAB*. Springer-Verlag Berlin Heidelberg, 1 Aufl., 2010.
- [6] BOEHM, W.: *Inserting new knots into B-spline curves*. Computer-Aided Design, 12(4):199–201, 1980.
- [7] BOEHM, W.: *Curvature Continuous Curves and Surfaces*. Computer Aided Geometric Design, 2(4):313–323, 1985.
- [8] BOOR, C. DE: *On calculating with B-splines*. Journal of Approximation Theory, 6:50–62, 1972.
- [9] BOOR, C. DE: *Efficient Computer Manipulation of Tensor Products*. ACM Transactions on Mathematical Software, 5(2):173–182, 1979.
- [10] BOOR, C. DE: *A Practical Guide to Splines*. Applied mathematical sciences. Springer, 2001.
- [11] BOOR, C. DE, K. HOELLIG und M. SABIN: *High accuracy geometric Hermite interpolation*. Computer Aided Geometric Design, 4(4):269 – 278, 1987.
- [12] BRECHER, C., M. DANIELS, A. EPPLE, S. NEUS und F. WELLMANN: *Realisierung effizienter Zerspanprozesse: Ergebnisbericht des BMBF Verbundprojekts ReffiZ*. Shaker Verlag GmbH, Christian Brecher (Hrsg.), 2015.
- [13] BRECHER, C. und M. WECK: *Werkzeugmaschinen 1: Maschinenarten und Anwendungsbereiche*. Springer Berlin Heidelberg, 6 Aufl., 2005.
- [14] BRECHER, C. und M. WECK: *Werkzeugmaschinen 2: Konstruktion und Berechnung*. Springer Berlin Heidelberg, 8 Aufl., 2006.
- [15] CARDON, D., G. FINNIGAN, T. LYCHE, N. NORTH, T. SEDERBERG und J. ZHENG: *T-splines Simplification and Local Refinement*. ACM Transactions on Graphics, 23, 2004.
- [16] COHEN, E., T. LYCHE und R. RIESENFELD: *Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics*. Computer Graphics and Image Proces-

- sing, 14:87–111, 1979.
- [17] DEROSE, T. D. und B. A. BARSKY: *An Intuitive Approach to Geometric Continuity for Parametric Curves and Surfaces*. In: *Proceedings of Graphics Interface '85 on Computer-generated Images: The State of the Art*, S. 159–175, New York, NY, USA, 1985. Springer-Verlag New York, Inc.
  - [18] DEUFLHARD, P. und A. HOHMANN: *Numerische Mathematik: Eine algorithmisch orientierte Einführung*. De Gruyter, 1 Aufl., 2008.
  - [19] ECK, M. und J. HADENFELD: *Knot Removal for B-spline Curves*. *Comput. Aided Geom. Des.*, 12:259–282, 1995.
  - [20] ENDL, K. und W. LUH: *Analysis II Eine integrierte Darstellung*. Akademische Verlagsgesellschaft Wiesbaden, 5 Aufl., 1981.
  - [21] FARIN, G.: *A Construction for Visual C1 Continuity of Polynomial Surface Patches*. *Computer Graphics and Image Processing*, 20:272–282, 1982.
  - [22] FARIN, G.: *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Academic Press Professional, Inc., 3 Aufl., 1993.
  - [23] FARIN, G., J. HOSCHEK und M.-S. KIM: *Handbook of Computer Aided Geometric Design*. Elsevier, 2002.
  - [24] FORSTER, O.: *Analysis 2*. Vieweg+Teubner Verlag, 6 Aufl., 2005.
  - [25] GAUTSCHI, W.: *Numerical Analysis*. Birkhaeuser Boston, 2 Aufl., 2012.
  - [26] GIBBS, J. und E. WILSON: *Vector analysis: A text-book for the use of students of mathematics*. Charles Scribners Sons, 1901.
  - [27] GOLDMAN, R. und T. LYCHE: *Knot Insertion and Deletion Algorithms for B-Spline Curves and Surfaces*. SIAM, 1993.
  - [28] GOLUB, G. und C. V. LOAN: *Matrix Computations*. Johns Hopkins University Press, 1996.
  - [29] HANDECK, J.: *Analyse und Korrektur von Teileprogrammen für numerisch gesteuerte Werkzeugmaschinen*. Doktorarbeit, University of Passau, Germany, 2017.
  - [30] HERON, G.: *Techniques for Visual Continuity*. *Geometric Modeling Algorithms and new trends*, S. 163–174, 1987.
  - [31] KIEF, H., H. ROSCHIWAL und K. SCHWARZ: *CNC-Handbuch 2015/2016*. Carl Hanser Verlag GmbH, 30 Aufl., 2015.
  - [32] KOLDA, T. G. und B. W. BADER: *Tensor Decompositions and Applications*. *SIAM Rev.*, 51(3):455–500, 2009.
  - [33] KREYSZIG, E.: *Differential Geometry*. Dover Publications Inc, 1991.
  - [34] KRUPPA, E.: *Analytische und konstruktive Differentialgeometrie*. Springer Verlag, 1 Aufl., 1957.



- [35] LAMPING, F.: *Kronecker-Produkte und multivariate Tensorprodukt Splines auf gestreuten Daten*. Doktorarbeit, University of Gießen, Germany, 2015.
- [36] LUTTERKORT, D., J. PETERS und U. REIF: *Polynomial degree reduction in the  $L_2$ -norm equals best Euclidean approximation of Bezier coefficients*. *Computer Aided Geometric Design*, 16(7):607 – 612, 1999.
- [37] LYCHE, T. und K. MORKEN: *Knot Removal for Parametric B-spline Curves and Surfaces*. *Comput. Aided Geom. Des.*, 4(3):217–230, 1987.
- [38] MARCUS, M. und H. MINC: *A Survey of Matrix Theory and Matrix Inequalities*. Dover Publications, 2010.
- [39] MATHWORKS: *External Interfaces R2014a*. The MathWorks, Inc., 2014.
- [40] MORETON, H.: *Minimum Curvature Variation Curves, Networks, and Surfaces for Fair Free-Form Shape Design*. Doktorarbeit, EECS Department, University of California, Berkeley, Mar 1993.
- [41] NOCEDAL, J. und S. WRIGHT: *Numerical Optimization*. Springer Verlag, 2 Aufl., 2006.
- [42] PAPIERNIK, W.: *Architecture and Design of Modern CNC/Drive Systems*. *Proceedings on PCIM-Conference*, 29:271–280, 1996.
- [43] PATRIKALAKIS, N. und T. MAEKAWA: *Shape Interrogation for Computer Aided Design and Manufacturing*. Springer Verlag Berlin-Heidelberg, 1 Aufl., 2002.
- [44] PAUL, J.-C., K.-L. SHI, J.-G. SUN, J.-H. YONG, H. ZHANG und S. ZHANG:  *$G^2$  B-Spline Interpolation to a Closed Mesh*. *Computer-Aided Design*, 43:145–160, 2011.
- [45] PEGNA, J. und F.-E. WOLTER: *Geometrical Criteria to Guarantee Curvature Continuity to Blend Surfaces*. *Journal of Mechanical Design*, 114:201–210, 1992.
- [46] PETERS, J.: *Geometric Continuity*, 2001.
- [47] PRAUTZSCH, H., W. BOEHM und M. PALUSZNY: *Bezier and B-Spline Techniques*. Springer Verlag Berlin-Heidelberg, 1 Aufl., 2002.
- [48] ROGERS, D.: *An Introduction to NURBS*. Elsevier Inc., 2001.
- [49] SAUER, T.: *Splinekurven und -flächen in CAGD und Anwendung*. Vorlesungsskript Universität Passau, 2012.
- [50] SCHOENBERG, I. und A. WHITNEY: *On Polya Frequency Functions. III. The positivity of translation determinants with an application to the interpolation problem by spline curves*. *Trans. Amer. Math. Soc.*, 74, 1953.
- [51] STÄRK, E.: *Mehrfach differenzierbare Bezier-Kurven und -Flächen*. Dissertation TU Braunschweig, 1976.
- [52] THORPE, J. A.: *Elementary Topics in Differential Geometry*. Springer Verlag New York, 1 Aufl.,

1979.

- [53] VERON, M., G. RIS und J.-P. MUSSE: *Continuity of biparametric surface patches*. Computer Aided Design, 8:267–273, 1976.
- [54] VINACUA, A. und P. BRUNET: *A Construction for VCI Continuity of Rational Bézier Patches*. Mathematical Methods in Computer Aided Geometric Design, 1987.
- [55] WOLF, J.: *C++ Das umfassende Handbuch*. Galileo Computing, 3 Aufl., 2014.

# SYMBOLVERZEICHNIS

$\langle \cdot \rangle$	Skalarprodukt
$\text{vec}(\mathbf{A})$	Vektorisierte Form einer Matrix $\mathbf{A}$
$\text{trace}(\mathbf{A})$	Spur einer Matrix $\mathbf{A}$
$\mathbf{T}^\dagger$	Matrifizierung eines Tensors $\mathbf{B}$
$\otimes$	Kronecker-Produkt
$\mathbf{I}_n$	$(n \times n)$ -Einheitsmatrix
$f, g$	Splinekurven
$F, G$	Tensorprodukt-Flächen
$F_u, F_v$	Partielle Ableitung von $F$ in Richtung $u$ bzw. $v$
$f.T$	Knotenvektor der Kurve $f$
$F.T_u, F.T_v$	Knotenvektor der Tensorprodukt-Fläche in $u$ - bzw. $v$ -Richtung
$f.d, F.d$	Kontrollpolygon einer Splinekurve bzw. Tensorprodukt-Fläche
$n_F$	Flächennormale von $F$
$\hat{n}_F$	Normierte Flächennormale von $F$
$D^n F$	$n$ -tes Differential von $F$
$\frac{\partial F}{\partial u}(u, v), \frac{\partial F}{\partial v}(u, v)$	Partielle Ableitung in $u$ - bzw. $v$ -Richtung
$\mathbf{V}$	Vandermonde-Matrix einer Splinekurve
$\mathbf{V}_u, \mathbf{V}_v$	Vandermonde-Matrix einer Tensorprodukt-Fläche in $u$ - bzw. $v$ -Richtung
$\kappa$	Krümmung einer Kurve
$\kappa_G$	Gauß-Krümmung einer Fläche
$\kappa_n$	Normalenkrümmung einer Fläche



# LISTE ALLER ALGORITHMEN

2.1	Algorithmus von de Boor . . . . .	27
2.2	Boehm-Algorithmus . . . . .	31
2.3	Interpolation einer Punktwolke . . . . .	40
4.1	Homogenisierung zweier Knotenvektoren . . . . .	89
4.2	Bogenlängenapproximation einer Splinekurve . . . . .	92
4.3	Äquidistante Abtastung einer Splinekurve . . . . .	93
4.4	Projektion der cbd-Punkte . . . . .	96
4.5	Berechnung der $cbd_{s_F}^2$ . . . . .	99



# INDEX

- $C^n$ -Stetigkeit, 47
- $G^n$ -Stetigkeit, 48
- $n$ -te Ableitung, 17
- (Flächen-)Normalenvektor, 19
- (Flächen-)Reparametrisierung, 18
  
- B-Splines, 25
- Boehm-Algorithmus, 30
- Bogenlänge, 11
  
- Differenzenmatrix, 29
  
- Einheitstangentenvektor, 11
- Erste Fundamentalform, 20
- Extrudieren, 24
  
- Gauß-Krümmung, 24
- Geodätische Krümmung, 22
- Greville-Abszissen, 33
  
- Hauptkrümmungen, 23
- Hauptkrümmungsrichtungen, 23
- Hesse-Matrix, 17
  
- Jakobi-Matrix, 17
  
- Knoteneinfügematrix, 32
- Knotenfolge, 25
- Kontrollpolygon, 26
  
- Krümmungsradius, 13
- Kurvenkrümmung, 13
- Kurvennormale, 13
  
- Mittlere Krümmung, 24
- Moore-Penrose Pseudo-Inverse, 66
  
- Normalenebene, 12
- Normalenkrümmung, 22
  
- Oslo-Algorithmus, 30
  
- Parametrisierung, 9
- Parametrisierung einer Fläche, 16
  
- Randquerableitung, 91
- Reguläre Reparametrisierung, 10
  
- Schmiegeebene, 12
- Schmiegekreis, 15
- Smoothing Spline, 34
- Spline-Raum, 25
- Splinekurve, 26
  
- Tangentialebene, 19
- Tensorprodukt-Fläche, 37
  
- Vandermonde-Matrix, 27
  
- Zweite Fundamentalform, 21





# ERKLÄRUNG DER SELBSTÄNDIGKEIT

Mai 2017

Ich versichere, dass ich diese Dissertation selbständig, ohne fremde Hilfe und ohne Benutzung anderer als der genannte Quellen und Hilfsmittel verfasst habe und dass sämtliche Ausführungen, die wörtlich oder sinngemäß übernommen wurden, als solche gekennzeichnet sind.

---

Lorenz, Florian