UNIVERSITÄT PASSAU

Fakultät für Informatik und Mathematik

Analyse und Korrektur von Teileprogrammen für numerisch gesteuerte Werkzeugmaschinen

Zur Erlangung des akademischen Grades
„Doktor der Naturwissenschaften“
(Dr. rer. nat.)

eingereicht von
Jörg Handeck (M. Sc.)

Betreuer: Professor Dr. Tomas Sauer

Erlangen, 23. Januar 2017

Inhaltsverzeichnis

1 Einleitung **1**
1.1 Motivation . . . 1
1.2 Aufgabenstellung und Umsetzung . . . 1
1.3 Gliederung der Arbeit . . . 3

2 Einführung in die Problemstellung **5**
2.1 Fertigung mit numerisch gesteuerten Werkzeugmaschinen . . . 5
2.2 Problemstellung . . . 8

3 Grundlagen **11**
3.1 Splines: Kurven und Flächen . . . 11
3.1.1 B–Splines . . . 11
3.1.2 Splinekurven . . . 14
3.1.3 Approximation . . . 18
3.1.4 Duale Funktionale . . . 20
3.1.5 Verfeinerung . . . 21
3.1.6 Tensorproduktsplines . . . 28
3.2 Quaternionen . . . 29
3.2.1 Definition und Eigenschaften . . . 30
3.2.2 Spezielle Quaternionen . . . 31
3.2.3 Drehungen . . . 31

4 Analyse der Position **33**
4.1 Approximation G-Code . . . 33
4.1.1 Wahl des Knotenvektors . . . 34
4.1.2 Schoenberg–Approximant . . . 37
4.2 Merkmalserkennung . . . 38
4.2.1 Allgemeiner Multiresolution Ansatz . . . 38
4.2.2 Abklingrate der Waveletkoeffizienten . . . 43
4.2.3 Interpretation des Analysesplines . . . 49
4.3 Anwendung und Korrektur . . . 59
4.3.1 Identifizierung der Problemstellen . . . 59
4.3.2 Berechnung der Korrekturpunkte . . . 61

5 Analyse der Orientierung **69**
5.1 Splines auf der Sphäre $\mathbb{S}^2$ . . . 69
5.1.1 Wahl des Referenzpunktes . . . 71
5.1.2 Sphärischer Approximant . . . 74
5.1.3 Konvexe Hülle . . . 78
5.2 Anwendung . . . 85
5.2.1 Approximation und Merkmalsanalyse . . . 86
5.2.2 Korrekturmöglichkeit . . . 86

6 Schlussbemerkung **89**
6.1 Zusammenfassung 89
6.2 Fazit und Ausblick 90

Symbolverzeichnis **91**

Abbildungsverzeichnis **93**

Literaturverzeichnis **95**

1 Einleitung

1.1 Motivation

In der industriellen Produktion sind Werkzeugmaschinen als Mittel der effizienten und präzisen Fertigung von Werkstücken nicht mehr wegzudenken. Die Qualität der Produkte sowie die Wirtschaftlichkeit der Produktion ist von ihnen in hohem Maße abhängig. Gerade die Wirtschaftlichkeit der Produktion fordert oft eine hohe Fertigungsgeschwindigkeit. Schnellere Produktionszeiten gehen allerdings häufig zu Lasten der Produktionsgenauigkeit und somit der Qualität des Werkstücks. Die größte Herausforderung einer Werkzeugmaschine, genauer gesagt der numerischen Steuerung, ist es, Produktqualität und Wirtschaftlichkeit der Produktion zu vereinen.

Werden entsprechende Toleranzwerte für eine hohe Vorschubgeschwindigkeit maximal ausgenutzt, treten unerwünschte Effekte wie inhomogene Oberflächen an geometrisch markanten Stellen auf, siehe Abbildung 1.1 (links). Diese Effekte gilt es, mit möglichst minimalen Einbußen an Produktivität, zu vermeiden. Hierfür ist es von großem Nutzen, im Voraus zu wissen, an welchen Stellen eine exaktere Abtastung nötig ist, um speziell an diesen Stellen Teileprogramme zu korrigieren, so dass die gewünschte Qualität erreicht wird.

Im Zuge der Digitalisierung werden Produktionsanlagen zunehmend vernetzter. Maschinen und IT-Systeme tauschen mehr Informationen miteinander aus. Somit ist es anzunehmen, dass einer numerischen Steuerung in Zukunft die vollständigen Informationen über die geometrischen Strukturen der zu fertigenden Werkstücke online zur Verfügung stehen. Zur Berechnung von Korrekturen bzw. Ergänzungen sollte somit stets die Möglichkeit in Betracht gezogen werden, auf diese Geometriedaten zurückgreifen zu können.

Die vorliegende Arbeit liefert einen Beitrag dazu, die Ziele Produktqualität und Produktionsgeschwindigkeit zu vereinen.

1.2 Aufgabenstellung und Umsetzung

Ziel dieser Arbeit ist es, die zuvor beschriebenen Problemstellen unter Verwendung der Daten, die einer numerischen Steuerung aktuell zur Verfügung stehen, zu lokalisieren und durch Korrekturen bzw. Ergänzungen des Teileprogramms auf Basis der ursprünglichen Geometrieinformationen zu beheben. Insbesondere soll hierbei ein allgemeines Verfahren zur Detektion dieser Stellen entworfen werden und unter Berücksichtigung der Geometriedaten des Werkstücks und des Werkzeugs eine Möglichkeit der Korrektur aufgezeigt werden. Mit Hilfe des in dieser Arbeit vorgestellten Verfahrens können diese Stellen detektiert und korrigiert werden. Siehe Abbildung 1.1 (rechts).

In den letzten Jahren stieg der Marktanteil der Fünf–Achs–Werkzeugmaschinen stark an. Ein Grund hierfür ist unter anderem, dass Designer mit Hilfe der CAD–Systeme immer leichter

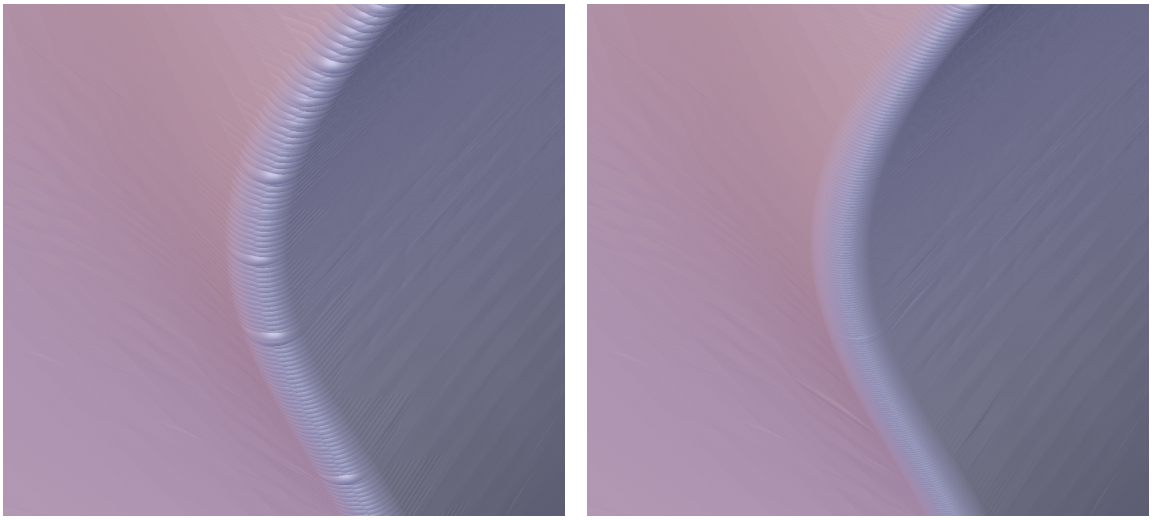


Abb. 1.1: Simulation eines Fräsvorgangs vor (links) und nach (rechts) dem Korrekturverfahren.

komplexe Geometrien erzeugen können. Deren Herstellung erfordert oft die Bearbeitung eines Werkstücks von mehreren Seiten unter verschiedenen Stellungen des Werkzeugs. Dies erfordert zusätzlich zu den drei *Linearachsen*, die die Position des Werkzeugs im Raum festlegen, zwei weitere sogenannte *Rundachsen*. Mit Hilfe dieser Achsen ist es möglich, das Werkzeug relativ zum Werkstück an jedem Punkt und in jedem beliebigen Winkel zur Werkstückoberfläche anzustellen. Als weitere Herausforderung soll somit zusätzlich zur Fräsbahn eines Werkzeugs auch der Verlauf seiner Orientierung auf kritische Stellen analysiert werden. Hierfür gilt es zunächst eine Möglichkeit der Darstellung einer *Orientierungsbahn* zu finden, für die das Analyseverfahren der Fräsbahn ebenfalls anwendbar ist.

Einer numerischen Steuerung stehen aktuell zum Fertigen eines Werkstücks nur die Daten aus einem *Teileprogramm* zur Verfügung. Dabei handelt es sich meist um lineare Sätze, die eine Abtastung der Geometrie innerhalb einer gewissen Toleranz entsprechen. Für die Umsetzung der Lokalisierung kritischer Stellen bzw. für die Analyse dieser Daten wird ein Splineapproximant mit Hilfe eines Multiresolution Ansatzes untersucht. Im Unterschied zu üblichen Verfahren wird dieser Ansatz auf Splinekurven mit adaptiven Knotenfolgen angewandt, da so wesentlich bessere Approximations– bzw. Interpolationsergebnisse erzielt werden können. Hintergrund hierfür ist, dass in einer numerischen Steuerung das Bahnpolynom ebenfalls mit nicht äquidistanten Knotenfolgen generiert wird und auf diesem das vorgestellte Analyseverfahren angewendet werden soll. Mit Hilfe einer Rückfrage an ein CAD–System, genauer gesagt, an die Originalgeometrie, können Korrekturen des Teileprogramms unter Berücksichtigung der verwendeten Bearbeitungsstrategie berechnet werden. Bei den Korrekturen handelt es sich üblicherweise um Ergänzungen des Teileprogramms. Die vorhandenen Punkte im Teileprogramm sind alle innerhalb der jeweils angegebenen Toleranzwerte, also insbesondere nicht falsch. Für die gewünschte Qualität sind sie allerdings nicht ausreichend und repräsentieren zu wenig Information an den entsprechenden Stellen.

Der Verlauf der Orientierung eines Fräsers (Orientierungsbahn) kann durch eine Kurve auf der Oberfläche einer Kugel dargestellt werden. Um die Orientierungsbahn mit dem Verfahren der Positionsbahn analysieren zu können, wurde ein geeigneter Approximant entwickelt. Hierfür wurde ein *homogener Quaternionenspline* verwendet, der die Rotation eines Referenzpunktes entlang der zu approximierenden Daten darstellt. Auf diesen Quaternionenspline kann das in Kapitel 4

hergeleitete Multiresolution Verfahren zur Analyse der Orientierungsbewegung angewandt werden.

1.3 Gliederung der Arbeit

Kapitel 2 dient der Einführung in die Thematik der Fertigung mit numerisch gesteuerten Werkzeugmaschinen. Hierbei wird speziell auf die CAD/CAM/CNC–Prozesskette eingegangen. Des Weiteren werden die Ursachen der oben beschriebenen Problemstellen beschrieben.

Zur Einführung der verwendeten Notation und der wichtigsten bekannten Sätze aus den Bereichen der Splinetheorie sowie der Quaternionen dient Kapitel 3.

In Kapitel 4 wird auf die Analyse der Positionsbahn einer Werkzeugmaschine eingegangen. Es wird zunächst beschrieben, wie die Daten aus einem Teileprogramm approximiert werden können, wobei hier auf die Wahl eines krümmungsadaptierten Knotenvektors eingegangen wird. Für diese Kurven wird ein Spline Multiresolution Ansatz präsentiert, der sich durch die Wahl eines nicht äquidistanten Knotenvektors von einem üblichen Multiresolution Ansatz unterscheidet. Es müssen zur Normalisierung die Abklingfaktoren der Waveletkoeffizienten in den jeweiligen Zerlegungsstufen bestimmt werden. Diese Abklingfaktoren werden hergeleitet. Mit Hilfe der Detailanteile der Waveletzerlegung und der jeweiligen Abklingfaktoren kann ein Analysespline der Positionsbahn angegeben werden. Die Extremwerte dieses Analysesplines repräsentieren Kandidaten für kritische Stellen in der Positionsbahn. Mit Hilfe der Bewertung von Persistenz und Dominanz der Maxima im Analysespline, ist es möglich, diese zu klassifizieren und somit die Problemstellen zu lokalisieren. Als Abschluss des Kapitels wird die Anwendung des vorgestellten Multiresolution Verfahrens auf eine Testgeometrie präsentiert. An der als kritisch erkannten innen liegenden Ecke wird eine Korrektur unter Zuhilfenahme der Originalgeometrie durchgeführt.

In Kapitel 5 wird eine Möglichkeit der Approximation von Daten auf der Sphäre vorgestellt. Es wird zunächst auf die Wahl eines Knotenvektors eingegangen, wobei hier das aus Kapitel 4 bekannte Verfahren für sphärische Daten adaptiert wird. Sodann wird die Konstruktion eines sphärischen Schoenberg–Approximanten angegeben. Für diesen Approximanten wird eine Konvexe–Hülle–Eigenschaft auf der Sphäre bewiesen. Als Abschluss des Kapitels werden die Orientierungsdaten eines Teileprogramms mit dem vorgestellten sphärischen Schoenberg–Approximanten approximiert und anschließend der für diesen Approximanten notwendige Quaternionenspline mit Hilfe des Multiresolution Verfahrens aus Kapitel 4 analysiert.

2 Einführung in die Problemstellung

Dieses Kapitel dient der Einführung in die Thematik der numerisch gesteuerten Werkzeugmaschinen. Es wird speziell die Komponente *numerische Steuerung* bzw. deren Eingabe, das sogenannte *Teileprogramm* behandeln. Des Weiteren wird die Problemstellung dieser Arbeit erörtert.

2.1 Fertigung mit numerisch gesteuerten Werkzeugmaschinen

Werkzeugmaschinen bringen ein Werkstück mit Hilfe eines Werkzeugs in eine bestimmte Form. Die Anwendungsbereiche sind so vielfältig, dass es sehr viele Arten von Maschinentypen gibt, alle mit unterschiedlichen, oft stark spezialisiertn Fähigkeiten zur Fertigung von Werkstücken. Das jeweilige Fertigungsverfahren mit dem eine Werkzeugmaschine ein Werkstück fertigen soll bestimmt hauptsächlich, aber nicht nur, die Bauform der Werkzeugmaschine. Die gängigsten Maschinentypen sind hier die Fräs– und Drehmaschinen. Die im weiteren Verlauf dieser Arbeit betrachteten Maschinentypen sind die Fräsmaschinen. Bei der Fertigung eines Werkstücks auf einer Fräsmaschine fährt das Werkzeug (Fräser) einen vorher festgelegten Weg ab und bearbeitet beim Abfahren dieses Weges, in einem *spanenden Eingriff*, das gewünschte Material. Eine Überblick über verschiedene Fertigungsverfahren, Maschinenarten und Anwendungsbereiche findet man z. B. in [31, 52].

Bevor es jedoch zum eigentlichen Fertigungsprozess kommt, wird in der Produktion üblicherweise die sogenannte *CAD/CAM/CNC–Prozesskette* durchlaufen, siehe Abbildung 2.1. Hierbei stehen die Abkürzungen für:

- CAD (computer aided design) rechnergestütztes Konstruieren,
- CAM (computer aided manufacturing) rechnergestützte Fertigung,
- CNC (computerized numerical control) rechnergestützte numerische Steuerung.

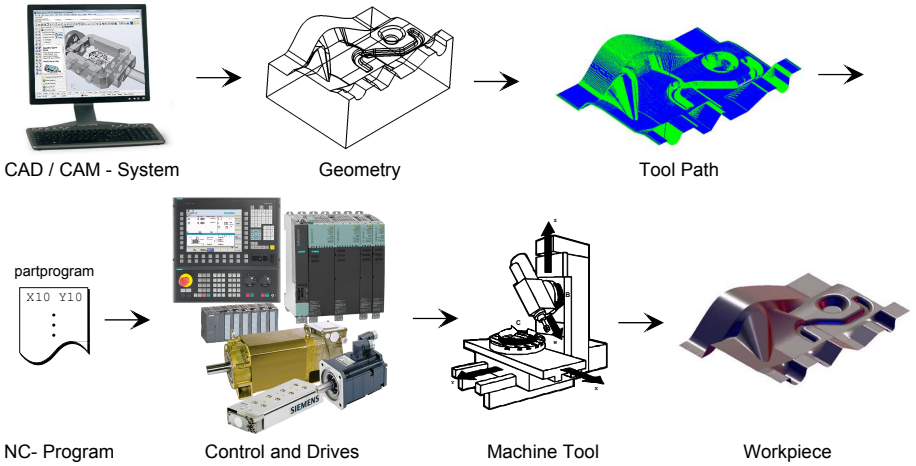


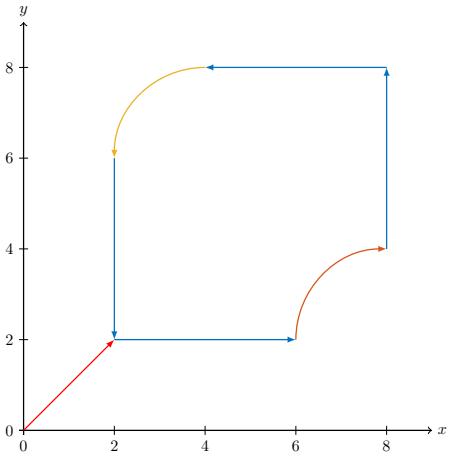
Abb. 2.1: CAD/CAM/CNC–Prozesskette der Fertigung (Quelle: Siemens AG).

Wegbedingung	Funktion
G00	Positionieren im Eilgang
G01	lineare Interpolation
G02	Kreisinterpolation (Uhrzeigersinn)
G03	Kreisinterpolation (gegen Uhrzeigersinn)

Tab. 2.1: G-Code Wegbedingungen

```
      ⋮
N100 X0 Y0
N110 G00 X2 Y2
N120 G01 X6 F1000
N130 G02 X8 Y4 I2 J0
N140 G01 Y8
N150 X4
N160 G03 X2 Y6 I0 J-2
N170 G01 Y2
      ⋮
```

(a) G–Code



(b) Visualisierung des G–Codes.

Abb. 2.2: Teileprogramm und dazugehörige Bewegung.

Die Prozesskette beginnt mit der Konstruktion des Werkstücks. Dies geschieht üblicherweise in einem CAD–System. Die Geometrie liegt zu diesem Zeitpunkt in einer analytischen Form vor. Im nächsten Schritt, dem CAM–System, wird die Geometrie diskretisiert. Dies kann auf viele verschiedene Arten geschehen, wobei diverse technologische Parameter wie zum Beispiel Werkzeugausprägung oder Bearbeitungsstrategie sowie Toleranzwerte berücksichtigt werden müssen. Anhand dieser Diskretisierung der Geometrie werden die Bahnen festgelegt, auf denen sich das Werkzeug bewegt. Das Format, in dem diese Daten üblicherweise vorliegen, ist das sogenannte CLDATA–Format (cutter location data), welches in der DIN 66215 festgelegt ist. Als nächstes wird dieses allgemeingültige Format mit Hilfe des *Postprozessors (PP)* in ein für die jeweilige Steuerung angepasstes Format übersetzt. Hierbei unterstützt der PP die steuerungsspezifischen Ausprägungen. Das entstehende *Teileprogramm (part program)*, auch *NC–Programm* oder *Steuerungsprogramm* genannt, liegt meist im G–Code Format (DIN 66025) vor und beschreibt üblicherweise die Bahn an der Werkzeugspitze (*tool center point (TCP)*). Dieses Format wird von den meisten Steuerungen verstanden, wobei viele Steuerungen herstellerspezifische Spezialbefehle verstehen.

Ein Teileprogramm ist in Sätze (Zeilen) aufgeteilt. Die einzelnen Sätze bestehen aus mehreren Wörtern, wobei ein Wort aus einem Buchstaben und einer Zahl besteht. Mit diesen Wörtern können der Steuerung verschiedene Anweisungen mitgeteilt werden. Das können Fahranweisungen sein, die bestimmen wie (G) und wohin (X,Y,Z) die Maschine das Werkzeug bewegen soll, aber auch technologische Anweisungen, mit denen die Vorschubgeschwindigkeit (F) oder Spindeldrehzahl (S) angegeben wird. Des Weiteren können Schaltbefehle angegeben werden, mit deren Hilfe die Auswahl der Werkzeuge (T) oder das Kühlmittel (M) gesteuert wird. Tabelle 2.1 enthält einige der am häufigsten verwendeten Verfahrbewegungen. In Abbildung 2.2 ist ein

Abb. 2.3: Siemens SINUMERIK 840D sl, CNC–Steuerung mit Bedientafel und Maschinensteuertafel (Quelle: Siemens AG).

kurzer Abschnitt G–Code mit entsprechender Visualisierung der Bahn zu sehen.

Speziell für Fünf–Achs–Werkzeugmaschinen ist es möglich in einem Satz eines Teileprogramms mittels der Bezeichner `A3`, `B3` und `C3` die Komponenten eines Vektors anzugeben.

```
N42 G01 X100 Y50 Z73 A3=0.5774 B3=0.5774 C3=0.5774 F1000
```

Dieser Vektor gibt die Orientierung des Werkzeugs im *Werkstückkoordinatensystem (WKS)* an. Er kann als Richtungsvektor von der Werkzeugspitze zur Werkzeugaufnahme interpretiert werden. Es gibt noch eine Reihe weiterer Möglichkeiten die Orientierung eines Werkzeugs anzugeben, zum Beispiel über die Rundachspositionen oder als Offsetkurve zur Bahn des Werkzeugs.

Für eine ausführlichere Beschreibung der Struktur, Syntax und Semantik der Teileprogramme sei auf die DIN 66025 oder [31] verwiesen.

Der nächste Schritt in der Prozesskette ist die Verarbeitung des Teileprogramms in einer numerischen Steuerung. Abbildung 2.3 zeigt eine numerische Steuerung mit Bedienelementen. Diese generiert aus den Daten des Teileprogramms die nötigen Ansteuerungen der Motoren bzw. der Achsen einer Werkzeugmaschine, um Werkzeug, respektive Werkstück, wie gewünscht zu verfahren. Eine der wichtigsten Komponenten einer numerischen Steuerung ist der sogenannte *numerische Kern*, auch *NCK (numerical control kernel)* genannt. Dieser übernimmt innerhalb der numerischen Steuerung die Aufgabe der Berechnung der Bewegung der Achsen einer Werkzeugmaschine. Die Verarbeitung eines Teileprogrammes im NCK kann im Wesentlichen in vier Schritten zusammen gefasst werden (siehe Abbildung 2.4).

Interpreter: Hier werden die Sätze (Zeilen) des Teileprogramms in die interne Darstellung der Steuerung umgewandelt. Es werden technologischen Daten wie zum Beispiel Vorschub– und Schnittgeschwindigkeit oder welches Werkzeug gerade aktiv ist vervollständigt. Der in Abbildung 2.4 dargestellte Ablaufplan gilt im Wesentlichen nur für sogenannte Verfahrsätze. Das sind Sätze des Teileprogramms, in denen Positions– oder Orientierungsänderungen beschrieben werden. Diese werden vom Interpreter an die nächste Station, den Vorlauf, weitergereicht.

Vorlauf: Im Vorlauf werden die Verfahrsätze, welche im *Werkstückkoordinatensystem* (WKS) angegeben sind, geometrisch aufbereitet. Falls möglich, werden mehrere Sätze zu einem Satz zusammengefasst oder an entsprechenden Stellen Zwischensätze, zum Beispiel zur Verrundung

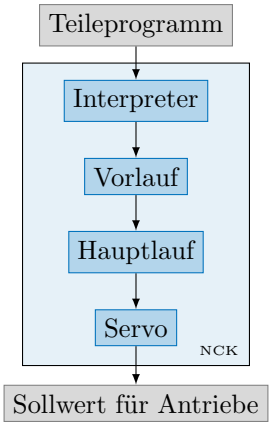


Abb. 2.4: Verarbeitungsschritte des Teileprogramms innerhalb der numerischen Steuerung.

der Geometrie, eingefügt. Des Weiteren werden technologische Aspekte wie zum Beispiel Werkzeugradiuskorrekturen auf die Sollbahn eingerechnet. Die Sollbahn besteht dann meist aus Polynomen bzw. Splinekurven und wird im *Basiskoordinatensystem* (BKS) angegeben. Der Vorlauf arbeitet die aus dem Teileprogramm vorgegebenen Verfahrsätze sequenziell ab und füllt einen Puffer für den Hauptlauf.

Hauptlauf: Der Hauptlauf verarbeitet, im Unterschied zum Vorlauf, nur einen Satz. Für die Verarbeitung des Satzes ist in der Regel nur eine bestimmte Zeitspanne vorgegeben. Es werden Zwischenpunkte der Bahn und die Sollwerte für den Geschwindigkeitsverlauf berechnet. Des Weiteren findet eine weitere Koordinatentransformation statt: vom BKS ins *Maschinenkoordinatensystem* (MKS).

Servo: Die Antriebsregelung berechnet die Sollwerte für die Antriebe unter Berücksichtigung der aktuellen Ist– und Dynamikwerten.

Weitere Informationen über den Aufbau von numerischen Steuerungen, speziell des numerischen Kerns, kann unter anderen [41] entnommen werden.

Das in dieser Arbeit vorgestellte Verfahren zur Korrektur von Teileprogrammen lässt sich in den präparativen Ablauf des NCK (Vorlauf) einbinden. Hier stehen zum einen die Daten aus dem Teileprogramm und zum anderen genug Rechenzeit zur Verfügung. Es ist außerdem vorstellbar, dass an dieser Stelle eine Rückfrage an ein CAD–System bezüglich der Geometrie stattfindet.

2.2 Problemstellung

In der vorliegenden Dissertation wird hauptsächlich ein Problem behandelt, welches in der zuvor beschriebenen Prozesskette bei der Diskretisierung des Werkstücks durch das CAM–System entsteht, in der verschiedene Toleranzwerte bzw. technologische Parameter wie zum Beispiel Sehnenfehler (siehe Abschnitt 4.1.1) oder maximale Satzlänge berücksichtigt werden. Diese Parameter lassen eine gewisse Ungenauigkeit bei der Diskretisierung des Werkstücks zu und hängen somit zum Teil direkt mit der gewünschten Fertigungsgenauigkeit zusammen. Diese Toleranzwerte werden in der Regel voll ausgenutzt, da hierdurch verschiedene Vorteile entstehen.

Ein unerwünschter Effekt hierbei ist allerdings eine unregelmäßige Abtastung des Werkstücks

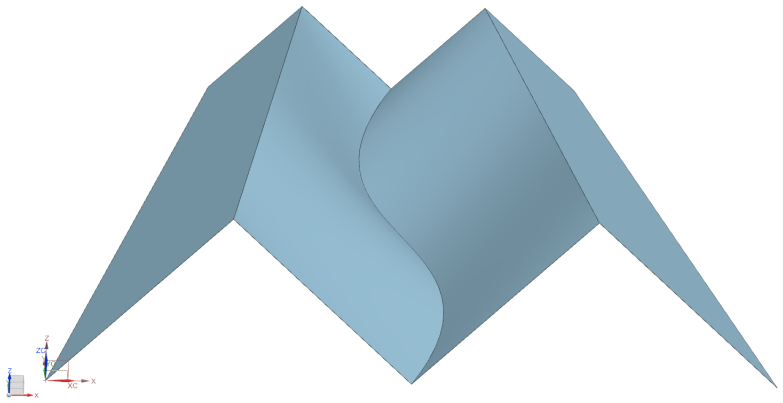


Abb. 2.5: Geometrie eines Werkstücks mit problematischer innen liegenden Ecke.

entlang eines geometrischen Merkmals in benachbarten Bahnen. Abbildung 2.5 zeigt ein solches typisches geometrisches Merkmal, eine innen liegende Ecke. Durch die Toleranzwerte wird dem CAM–System ermöglicht die Ecke abzukürzen, was zu besagter Abtastung führt, siehe Abbildung 2.6. Das Problem einer solchen unregelmäßigen Abtastung an solchen Stellen ist, wie schon in der Einleitung erwähnt, dass ein spürbar inhomogenes Fräsbild entlang dieser Merkmale entsteht, siehe Abb. 1.1 (links).

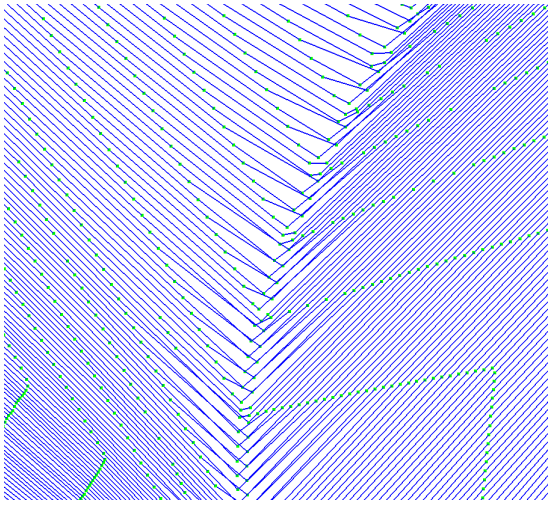


Abb. 2.6: Fräsbahnen über der innen liegenden Ecke der Geometrie aus Abbildung 2.5.

Diese Probleme können naiv durch das Verändern (meist Verkleinern) der entsprechenden Toleranzwerte behoben werden, da hierdurch eine wesentlich feinere Abtastung erreicht wird. In der Regel führt dies allerdings dazu, dass im CAM–System sehr viele, kurze, lineare Sätze erzeugt werden, auch dort, wo es nicht notwendig ist. Die Verarbeitung vieler kurzer Sätze bremst eine numerische Steuerung generell sehr stark aus, da für die Geschwindigkeitsführung der Maschine der Bremsweg stets berücksichtigt wird und dieser höchstens so lang sein darf wie die Steuerung an Bahn überblicken kann. Bei meist kurzen Sätzen ist dieser Weg entsprechend kurz und demzufolge wird die maximale Geschwindigkeit der Maschine gedrosselt. Da die Wirschaftlichkeit der Produktion mit Werkzeugmaschinen von der Fertigungsgeschwindigkeit abhängt, ist diese Lösung nicht praktikabel. Wünschenswert ist ein Verfahren, welches nur an den problematischen Stellen eine Korrektur (Ergänzung) des Teileprogramms vornimmt und demzufolge keine

Auswirkung auf das restliche Teileprogramm hat.

Somit ist ein Analyseverfahren notwendig, mit dessen Hilfe ein Teileprogramm untersucht werden kann, um die problematischen Stellen zu lokalisieren. An den lokalisierten Stellen muss anschließend eine Ergänzung des Teileprogramms berechnet werden.

3 Grundlagen

Dieses Kapitel dient zur Einführung der verwendeten Notation und kann als Referenzkapitel angesehen werden. In der vorliegenden Arbeit werden Matrizen und Vektoren in Fettschrift dargestellt. Vereinzelt kann es bei Vektoren jedoch auch vorkommen, dass diese in normaler Schriftstärke dargestellt werden. Einzelne Elemente einer Matrix werden durch einen doppelt indizierten Kleinbuchstaben in normaler Schriftstärke dargestellt. Für $\boldsymbol{A} \in \mathbb{R}^{n\times m}$, $n, m \in \mathbb{N}$, sei dies zum Beispiel $\boldsymbol{A} = (a_{ij})$ mit $i = 1, \ldots, n$ und $j = 1, \ldots, m$. Aus dem Kontext und durch die Wahl des Buchstabens sollte in der Regel klar sein, welche Matrix repräsentiert wird. Für die Indizes i, j ist die zulässige Definitionsmenge stets durch die Dimension der Matrix beschränkt. Einzelne Symbole und Zeichen können im Symbolverzeichnis auf Seite 91 nachgeschlagen werden. Ferner werden in diesem Abschnitt für die Arbeit wichtige, grundlegende Ergebnisse festgehalten. Hierbei handelt es sich keinesfalls um eine annähernd vollständige Einführung in die jeweiligen Themengebiete. Zu den hier aufgeführten Resultaten wird stets eine Quelle oder die Vorgehensweise eines Beweises angegeben.

3.1 Splines: Kurven und Flächen

Splinekurven gelten als das Mittel der Wahl zur Interpolation und Approximation von Daten. Gründe hierfür sind ihre lokale Einfachheit und gleichzeitige globale Flexibilität. Zudem gibt es sehr effiziente Algorithmen zur Auswertung oder Berechnung der Funktionswerte und Ableitungen sowie des Integrals.

Dieser Abschnitt ist eine reine Aufzählung der für diese Arbeit notwendiger Definitionen und Sätze. Eine ausführliche Ausarbeitung des Thema Splines findet man unter anderem in [17, 18] oder [47]. Die Notation für Splines in dieser Arbeit richtet sich im Wesentlichen nach [45].

3.1.1 B–Splines

Definition 3.1 *Seien $n, m \in \mathbb{N}$ mit $n > m$. Eine endliche Folge*

$$T_{n,m} = (\tau_1, \tau_2, \tau_3, \ldots, \tau_{n+m}, \tau_{n+m+1}) \in \mathbb{R}^{n+m+1} \tag{3.1}$$

heißt Knotenfolge *oder* Knotenvektor *zum Intervall $[\tau_1, \tau_{n+m+1}]$, falls*

$$\begin{aligned} &\tau_j \leq \tau_{j+1}, \quad j = 1, \ldots, n+m \text{ und} \\ &\tau_j < \tau_{j+m+1}, \quad j = 1, \ldots, n \end{aligned} \tag{3.2}$$

ist. Ein Knoten τ_j hat die Vielfachheit η falls

$$\tau_{j-1} < \tau_j = \cdots = \tau_{j+\eta-1} < \tau_{j+\eta}. \tag{3.3}$$

Die Knoten τ_1 bis τ_{m+1} heißen linke Randknoten*, bei τ_{n+1} bis τ_{n+m+1} spricht man von den* rechten Randknoten*. Bei den* inneren Knoten *eines Knotenvektors handelt es sich um $\tau_{m+1}, \ldots, \tau_{n+1}$. Die Knoten eines Knotenvektors haben, wie aus Bedingung* (3.3) *hervorgeht eine* maximale Vielfachheit *von $m + 1$.*

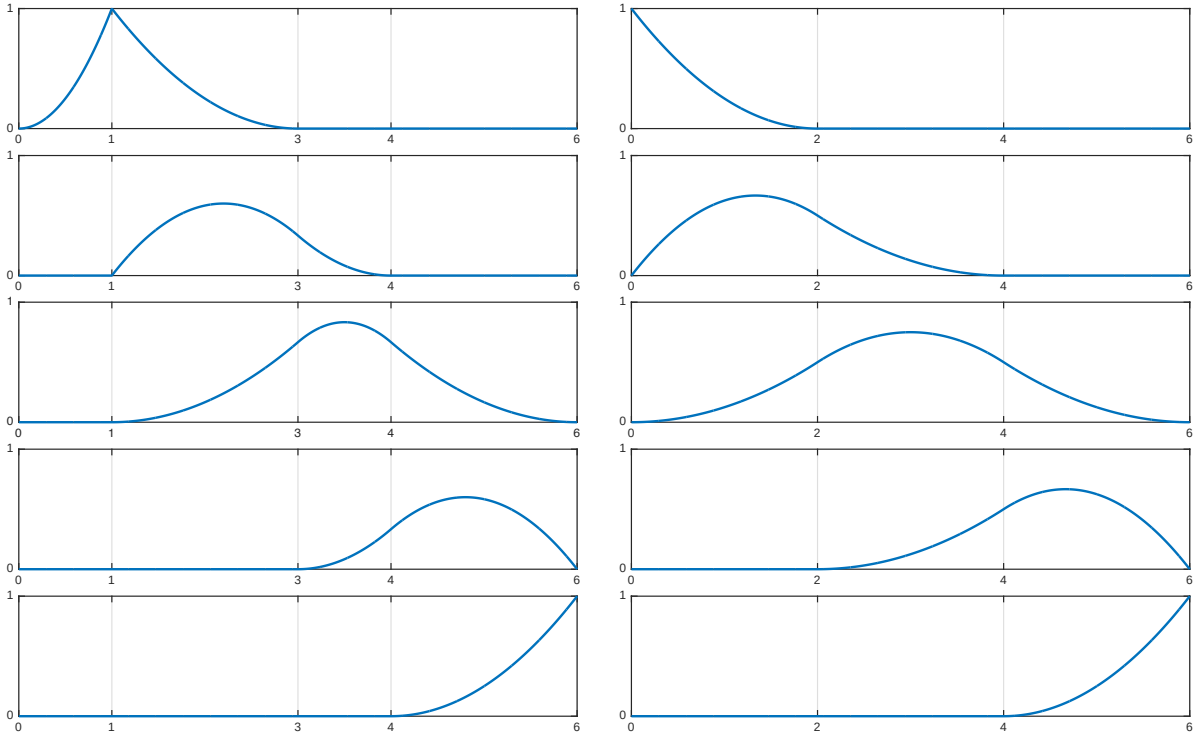


Abb. 3.1: B–Splines vom Grad $m = 2$ zu den Knotenvektoren (links) $T_{5,2} = (0, 1, 1, 3, 4, 6, 6, 6)$ und (rechts) $T_{5,2} = (0, 0, 0, 2, 4, 6, 6, 6)$.

Im Folgenden wird ein Knotenvektor $T_{n,m}$ vereinfacht mit T dargestellt, insofern aus dem Kontext klar wird wie n und m gewählt wurden oder nur ein Knotenvektor vorkommt.

Definition 3.2 *Sei $T_{n,m}$ ein Knotenvektor, dann bezeichnet N_j^k den j-ten* B–Spline *vom Grad k zum Knotenvektor T mit $k = 1, \ldots, m$ und $j = 1, \ldots, n + m - k$ und es gilt:*

$$N_j^0 (t \mid T) := \begin{cases} 1, & t \in [\tau_j, \tau_{j+1}) \\ 0, & \textit{sonst} \end{cases}, \tag{3.4}$$

$$N_j^k (t \mid T) := \frac{t - \tau_j}{\tau_{j+k} - \tau_j} N_j^{k-1} (t \mid T) + \frac{\tau_{j+k+1} - t}{\tau_{j+k+1} - \tau_{j+1}} N_{j+1}^{k-1} (t \mid T). \tag{3.5}$$

Bezüglich einer Möglichen Division durch Null in (3.5) sei an dieser Stelle auf Bemerkung 3.6 verwiesen.

Abbildung 3.1 ist eine Darstellung der B–Splines zu verschiedenen Knotenvektoren. Man beachte hier die Stetigkeit der B–Splines im linken Beispiel an einem m–fachen Knoten.

Die folgenden Eigenschaften für B–Splines lassen sich anhand ihrer Definition leicht durch Induktion über k beweisen. Die Beweise sind zum Beispiel in [18] oder [47] ausgeführt.

Proposition 3.3 *Die B–Splines*

1. *haben einen kompakten Träger: $N_j^k (t \mid T) = 0, t \notin [\tau_j, \tau_{j+k+1})$,*
2. *sind nichtnegativ: $N_j^k \geq 0$,*

3. *entsprechen auf einem Knotenintervall* (τ_j, τ_{j+1}) *einem Polynom vom Grad* $\leq k$*:* $N_j^k \mid_{(\tau_j,\tau_{j+1})} \in \Pi_k$.

Bemerkung 3.4 *Das kleinste Intervall, welches die* inneren Knoten *eines Knotenvektors* $T = T_{n,m}$ *beinhaltet, sei im Folgenden mit*

$$\mathcal{I}_m(T) := [\tau_{m+1}, \tau_{n+1}] \tag{3.6}$$

bezeichnet. Sobald aus dem Kontext klar wird, um welchen Knotenvektor T *es sich handelt, wird die verkürzte Notation* $\mathcal{I}_m$ *verwendet.*

Satz 3.5 *Es sei* $T = T_{n,m}$ *ein Knotenvektor sowie* $N_j^m(t \mid T)$ *die Folge von B–Splines zum Knotenvektor* T*. Für*

$$f(t) := \sum_{j=1}^{n} N_j^m(t \mid T)$$

gilt $f(t)|_{\mathcal{I}_m(T)} = 1$.

Satz 3.5 besagt, dass die B–Splines auf den inneren Knoten eines Knotenvektors eine nichtnegative Teilung der Eins bilden. Hat der entsprechende Knotenvektor allerdings einen $(m+1)$-fachen linken wie rechten Randknoten, bilden die B–Splines zur entsprechenden Knotenfolge auf dem gesamten Intervall $[\tau_1, \tau_{n+m+1}]$ eine Teilung der Eins. Ein Beweis hierzu sowie zum folgenden Lemma 3.7 über die Ableitung eines B–Splines ist zum Beispiel in [18] bzw. [47] nachzulesen.

Bemerkung 3.6 *Nach Definition 3.1 eines Knotenvektors ist es insbesondere möglich einen Knoten mit Vielfachheit* $\eta = m + 1$ *zu haben. In diesem Fall tritt in der Formel (3.5), der Definition der B–Splines, sowie in der Ableitungsformel (3.7) eine Division durch Null auf. Diese ist aber hinfällig, da der Träger des entsprechenden B–Splines leer ist. Gilt also* $\eta_j = m$*, dann ist* $\operatorname{supp} N_j^{m-1}(\,\cdot \mid T) = \emptyset$*. Es gilt die Konvention, dass alle Terme, in denen eine Division durch Null auftritt, weggelassen werden.*

Lemma 3.7 *Sei* T *ein Knotenvektor nach Definition 3.1 sowie* $N_j^m(t \mid T)$ *die entsprechende B–Spline Folge zum Knotenvektor* T*. Eine Ableitung des* j*-ten B–Spline an* $t \in [\tau_1, \tau_{n+m+1}] \setminus T$ *ist unter der Konvention gegeben durch:*

$$\frac{\mathrm{d}}{\mathrm{d}t} N_j^m(t \mid T) = \frac{m}{\tau_{j+m} - \tau_j} N_j^{m-1}(t \mid T) - \frac{m}{\tau_{j+m+1} - \tau_{j+1}} N_{j+1}^{m-1}(t \mid T), \quad j = 0, \ldots, n. \tag{3.7}$$

Eine besondere Rolle spielen die Knotenvektoren $T = \mathbb{Z}$ und $T = (\ldots, 0, 0, 1, 1, \ldots)$. Im Fall $T = \mathbb{Z}$ werden die korrespondierenden B–Splines *kardinale B–Splines* genannt. Der Begriff wurde 1969 von Schoenberg in [46] eingeführt. Die *kardinalen B–Splines* sind ein Musterbeispiel für verfeinerbare Funktionen und finden Anwendung in der Wavelet–Welt, siehe Abschnitt 4.2. Die B–Splines zum Knotenvektor $T = (\ldots, 0, 0, 1, 1, \ldots)$ entsprechen den sogenannten Bernsteinpolynomen, die Bernstein 1912 in [3] einführte. T ist ein Knotenvektor mit nur zwei Einträgen: 0 und 1. Diese haben jeweils die Vielfachheit $\eta_1 = \eta_{m+2} = m + 1$. Die Rekursionsformel (3.5) vereinfacht sich somit zu

$$N_j^k(t \mid T) = t\, N_j^{k-1}(t \mid T) + (1 - t) N_{j+1}^{k-1}(t \mid T).$$

Eine nicht rekursive Darstellung dieser B–Splines wird in Definition 3.8 der Bernsteinbasispolynome angegeben.

Definition 3.8 *Seien $j, m \in \mathbb{N}_0$ und $j \leq m$. Dann heißt die Funktion $B_j^m : [0,1] \to \mathbb{R}$* Bernsteinbasispolynom *und ist durch*

$$B_j^m(t) := \binom{m}{j} t^j (1-t)^{m-j} \tag{3.8}$$

definiert.

Als Spezialfall der B–Splines haben die Bernsteinbasispolynome selbstverständlich ebenfalls alle Eigenschaften aus Proposition 3.3. Der folgende Satz ist mit Beweis in [20] zu finden.

Satz 3.9 *Die Bernsteinbasispolynome B_j^m bilden eine Basis von Π_m.*

3.1.2 Splinekurven

Definition 3.10 *Sei $T = T_{n,m}$ ein Knotenvektor, $H = (\eta_1, \ldots, \eta_{n+m+1})$ der Vektor mit den entsprechenden Knotenvielfachheiten aus T und $I := [\tau_1, \tau_{n+m+1}]$. Dann bezeichnet $\mathbb{S}_m(T)$ den* Splineraum *der Ordnung m zum Knotenvektor T und ist definiert als*

$$\mathbb{S}_m(T) := \left\{ f \in C(I) \mid f|_{(\tau_j, \tau_{j+1})} \in \Pi_m \ \wedge \ f \in C^{m-\eta_j}((\tau_{j-1}, \tau_{j+\eta_j})) \right\}. \tag{3.9}$$

Im Splineraum $\mathbb{S}_m(T)$ liegen demnach alle Funktionen, die eingeschränkt auf ein Knotenintervall $[\tau_j, \tau_{j+1}]$ Polynome mit Höchstgrad m sind und an einem Knoten τ_j der Vielfachheit η_j immer noch $m - \eta_j$ mal stetig differenzierbar sind. Insbesondere kann mit Hilfe von Proposition 3.3 und einer Induktion über k leicht gezeigt werden, dass die Folge von B–Splines $N_j^k(\,\cdot \mid T)$ zum Knotenvektor T Elemente aus $\mathbb{S}_m(T)$ sind, siehe hierfür [45].

Satz 3.11 (Curry & Schoenberg) *Sei $T = T_{n,m}$ ein Knotenvektor mit $\tau_{m+1} < \tau_{m+2}$, dann bildet die zum Knotenvektor T korrespondierende Folge von B–Splines $N_j^m(\,\cdot \mid T)$ eine Basis des Splineraums $\mathbb{S}_m(T)$:*

$$\operatorname{span}\left\{ N_j^m(\,\cdot \mid T) \mid j = 1, \ldots, n \right\} = \mathbb{S}_m(T) \tag{3.10}$$

und es gilt $\dim(\mathbb{S}_m(T)) = n$.

Der Satz von Curry und Schoenberg ist üblicherweise in jeder Standardliteratur zum Thema Splines zu finden, zum Beispiel in [18, Kap. IX, S. 98f].

Definition 3.12 *Sei $T = T_{n,m}$ ein Knotenvektor und $d_j \in \mathbb{R}^k$ mit $j, k \in \mathbb{N}$ eine Koeffizientenfolge in $\mathbb{R}^k$ sowie $\boldsymbol{d} := [d_j \mid j = 1, \ldots, n] \in \mathbb{R}^{k \times n}$ die Komposition der Koeffizienten d_j zu einer Matrix. Dann ist eine* Splinekurve *$s_m\boldsymbol{d} : \mathbb{R} \to \mathbb{R}^k$ definiert durch*

$$s_m\boldsymbol{d}(t) := s_m\boldsymbol{d}(t \mid T) := \sum_{j=1}^{n} d_j \, N_j^m(t \mid T). \tag{3.11}$$

Die Koeffizienten einer Splinekurve werden im Folgenden auch als Kontrollpunkte oder Kontrollpolygon bezeichnet. Diese Begriffe sind äquivalent zu verstehen.

Bemerkung 3.13 *Falls nicht explizit anders angegeben, handelt es sich im Folgenden stets um einen Knotenvektor mit degenerierten, sprich $(m+1)$-fachen Randknoten. Eine wichtige*

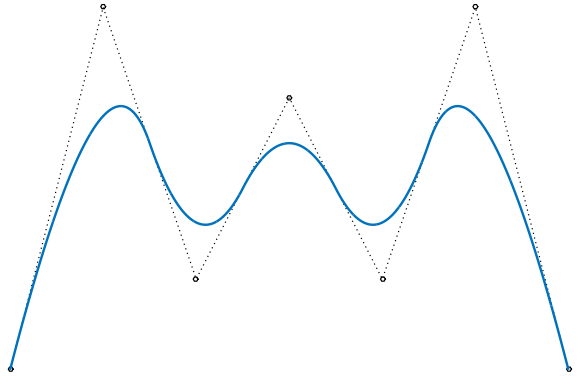


Abb. 3.2: Splinekurve mit $(m+1)$-fachen Randknoten und Kontrollpolygon.

Konsequenz aus dieser Bedingung ist die Endpunktinterpolation*. Das bedeutet, dass der erste und der letzte Kontrollpunkt einer Splinekurve stets interpoliert wird:* $s_m\boldsymbol{d}(\tau_1) = d_1$ *und* $s_m\boldsymbol{d}(\tau_{n+m+1}) = d_n$*.*

Eine Splinekurve mit $(m+1)$–fachen Randknoten sowie ihr Kontrollpolygon sind in Abbildung 3.2 zu sehen.

Definition 3.14 *Sei* $T = T_{n,m}$ *eine Knotenfolge und* $\boldsymbol{t} = (t_1, \ldots, t_k)$ *eine streng monoton steigende Punktfolge mit* $t_i \in [\tau_1, \tau_{n+m+1}]$*,* $i = 1, \ldots, k$*. Dann heißt*

$$\boldsymbol{N}_m := \boldsymbol{N}_{m,T}(\boldsymbol{t}) := \left[N_j^m(t_i \mid T) \mid i = 1, \ldots, n,\ j = 1, \ldots, k\right] \in \mathbb{R}^{n\times k} \tag{3.12}$$

Kollokationsmatrix *zum Knotenvektor* T *und Punktfolge* $\boldsymbol{t}$*.*

Mit Hilfe der Kollokationsmatrix aus Definition 3.14 und der Notation der Koeffizienten d_j einer Splinekurve $s_m\boldsymbol{d}$ als Koeffizientenmatrix $\boldsymbol{d}$, kann die Auswertung einer Splinefunktion durch Matrizenmultiplikation als Produkt von $\boldsymbol{d}$ und $\boldsymbol{N}_m$ ausgedrückt werden. Es folgt demnach für eine Splinekurve wie in Definition 3.12

$$s_m\boldsymbol{d}(\boldsymbol{t}) = \sum_{j=1}^{n} d_j\, N_j^m(\boldsymbol{t} \mid T) = \boldsymbol{d}\boldsymbol{N}_{m,T}(\boldsymbol{t}) = \boldsymbol{d}\boldsymbol{N}_m(\boldsymbol{t}). \tag{3.13}$$

Wertet man die Kollokationsmatrix an mehreren Stellen aus, so erhält man aus (3.13) einen Vektor mit den jeweiligen Funktionswerten.

Satz 3.15 (Schoenberg & Whitney) *Seien die Bedingungen aus Definition 3.14 erfüllt mit* $k = n$ *und die maximale Vielfacheit der Knoten von* T *sei* m*. Dann ist* $\boldsymbol{N}_{m,T}(\boldsymbol{t})$ *genau dann invertierbar, wenn*

$$\tau_j < t_j < \tau_{j+m+1}, \qquad j = 1, \ldots, n.$$

Der Satz von Schoenberg und Whitney sagt, dass man für eine eindeutige Lösung des Interpolationsproblems $s_m\boldsymbol{d}(t_j \mid T) = y_j \in \mathbb{R}^k$, $j = 1, \ldots, n$, die Interpolationsstellen t_j in den jeweiligen Träger der B–Splines mit gleichem Index wählen muss. Einen Beweis des Satzes findet man in [44]. Die Forderung der Schoenberg–Whitney–Bedingung an die Interpolationsstellen führt unmittelbar zu den sogenannten Greville–Abszissen.

Definition 3.16 *Zu einem Knotenvektor $T_{n,m}$ sind die* Greville–Abszissen $\hat{\boldsymbol{t}} = (\hat{t}_1, \ldots, \hat{t}_n)$ *definiert durch*

$$\hat{t}_j = \frac{1}{m} \sum_{k=1}^{m} t_{j+k}, \quad j = 1, \ldots, n. \tag{3.14}$$

Lemma 3.17 *Zu einer gegebenen Splinekurve $s_m \boldsymbol{d}$ mit Knotenvektor $T = T_{n,m}$ und Koeffizienten $\boldsymbol{d}$ kann die Ableitung von $s_m \boldsymbol{d}$ auf $\mathcal{I}_m(T)$ durch*

$$\frac{\mathrm{d}}{\mathrm{d}t}\, s_m \boldsymbol{d}(t \mid T) := m \sum_{j=1}^{n+1} \frac{d_j - d_{j-1}}{\tau_{j+m} - \tau_j} N_j^{m-1}(t \mid T) =: s_{m-1} \boldsymbol{d}^{(1)}(t) = s_m \boldsymbol{d}'(t) \tag{3.15}$$

bestimmt werden. Es gilt die Vereinbarung $d_0 := 0 =: d_{n+1}$.

Diese Ableitungsformel kann leicht mit der Ableitungsformel für B–Splines aus Lemma 3.7 nachgerechnet werden.

Bemerkung 3.18 *Zur Notation einer Ableitung: Im Folgenden bezeichnet ein eingeklammerter Exponent an einer Funktion stets die jeweilige Ableitung einer Funktion bzw. an Koeffizienten die Koeffizienten der Ableitungsfunktion.*

Sei $\boldsymbol{D}_n$ die *Differenzenmatrix* aus $\mathbb{R}^{n \times (n+1)}$ mit

$$\boldsymbol{D}_n := \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{bmatrix} \in \mathbb{R}^{n \times (n+1)} \tag{3.16}$$

und $\Delta_m T \in \mathbb{R}^{(n+1) \times (n+1)}$ eine Knotendifferenzmatrix, die als Einträge auf der Hauptdiagonalen den Reziprokwert der Differenz von je zwei Knoten aus dem Knotenvektor T hat, mit

$$\begin{aligned} \Delta_m T &:= \operatorname{diag}\left[(\tau_{j+m} - \tau_j)^{-1} \,:\, j = 1, \ldots, n+1 \right] \\ &= \begin{bmatrix} \tau_{m+1} - \tau_1 & & \\ & \ddots & \\ & & \tau_{n+m+1} - \tau_{n+1} \end{bmatrix}^{-1} \in \mathbb{R}^{(n+1) \times (n+1)}. \end{aligned} \tag{3.17}$$

Dann lässt sich die Kontrollmatrix $\boldsymbol{d}^{(1)}$ aus (3.15) einfach durch Matrizenmultiplikation berechnen.

$$\boldsymbol{d}^{(1)} = m\, \boldsymbol{d}\, \boldsymbol{D}_n\, \Delta_m T \tag{3.18}$$

Das Ganze lässt sich für höhere Ableitungen iterieren und führt zu

$$\boldsymbol{d}^{(k)} = \boldsymbol{d}\, \frac{m!}{(m-k)!} \prod_{j=0}^{k-1} \boldsymbol{D}_{n+j}\, \Delta_{m-j} T =: \boldsymbol{d}\, \boldsymbol{G}_k. \tag{3.19}$$

Hat der Knotenvektor allerdings $(m+1)$-fache Randknoten existiert die Matrix $\Delta_n T$ nicht, da hier in erster und letzter Zeile/Spalte eine Division durch Null auftritt. Dies ist aber kein Problem, da hier die Träger der entsprechenden B–Splines $N_1^{m-1}(\cdot \mid T)$ bzw. $N_{n+1}^{m-1}(\cdot \mid T)$ verschwinden und somit nichts zum Verhalten der Ableitung auf dem Intervall $\mathcal{I}_m(T)$ beitragen und nach Konvention weggelassen werden können. Bei einer Implementierung von (3.18) bzw. (3.19) ersetzt man die Elemente, bei denen eine Division durch Null auftritt mit 0 oder man implementiert beide Gleichungen nur auf dem relevanten Intervall $\mathcal{I}_m(T)$. Die Matrix $\boldsymbol{G}_k$ aus (3.19) wird *Ableitungsmatrix* einer Splinekurve genannt.

Auswertung

Es kommt oft vor, dass man eine Splinekurve $s_m\boldsymbol{d}$ an einer Stelle $t \in \mathcal{I}_m(T)$ auswerten und gleichzeitig die erste und zweite Ableitung an dieser Stelle erhalten möchte. Hier bietet sich der Algorithmus von de Boor an [18]. Dieser hat den Vorteil, dass man die Basisfunktionen N_j^m an der Stelle t nicht explizit kennen muss. Er nutzt darüber hinaus die Eigenschaft, dass nur $m+1$ Basisfunktionen und somit Koeffizienten bzw. Kontrollpunkte zur Auswertung einer Splinekurve an einer Stelle t beitragen, siehe Abbildung 3.3. Des Weiteren erhält man Ableitungen der Kurve ohne nennenswerten Mehraufwand als Zwischenergebnis des Algorithmus.

Algorithmus 3.19 (de Boor–Algorithmus)
Eingabe:

- *Knotenfolge* $T = T_{n,m}$
- *Koeffizienten bzw. Kontrollpunkte* $\boldsymbol{d} = [d_1, \ldots, d_n] \in \mathbb{R}^{h\times n}$
- $t \in \mathcal{I}_m(T)$

Prozedur:

1. **Bestimme** $\ell \in \{m+1, \ldots, n\}$ *so dass* $t \in [\tau_\ell, \tau_{\ell+1})$
2. **Initialisiere***:*
$$d_j^0(t) := d_j, \quad j = \ell - m, \ldots, \ell$$
3. **FOR** $k = 1, \ldots, m$
   FOR $j = \ell - m + k, \ldots, \ell$
$$\alpha_j(t) := \frac{\tau_{j+m-k+1} - t}{\tau_{j+m-k+1} - \tau_j}$$
$$d_j^k(t) := \alpha_j(t)\, d_{j-1}^{k-1}(t) + (1 - \alpha_j(t))\, d_j^{k-1}(t) \tag{3.20}$$
   END
END

Ergebnis:
Kontrollpunkt $d_\ell^m(t) = s_m\boldsymbol{d}(t)$

In Gleichung (3.20) sieht man, dass im de Boor–Algorithmus nichts anderes geschieht als eine wiederholte Anwendung einer Konvexkombination von Kontrollpunkten. Dies korreliert direkt mit Definition 3.2 der B–Splines. Setzt man die rekursive Definition (3.5) der B–Splines sukzessiv in die Definition einer Splinekurve (3.11) ein und betrachtet nur die Terme, an denen die B–Splines N_j^m zur Auswertungsstelle t nicht verschwinden, erhält man die Formel (3.20). Mit Hilfe der Ableitungsformel für Splinekurven, Lemma 3.17, lassen sich die Formeln zur Erweiterung des de Boor–Algorithmus um die jeweiligen Ableitungen herleiten. Die Erweiterung des Algorithmus 3.19 um die erste und zweite Ableitung einer Splinekurve kann mit Hilfe der Zwischenwerte $d_j^k(t)$ aus Schritt drei geschehen. Hierfür sei

$$c_0 := \frac{m}{\tau_{\ell+1} - \tau_\ell}, \quad c_1 := \frac{m-1}{\tau_{\ell+2} - \tau_\ell}, \quad c_2 := \frac{m-1}{\tau_{\ell+1} - \tau_{\ell-1}}.$$

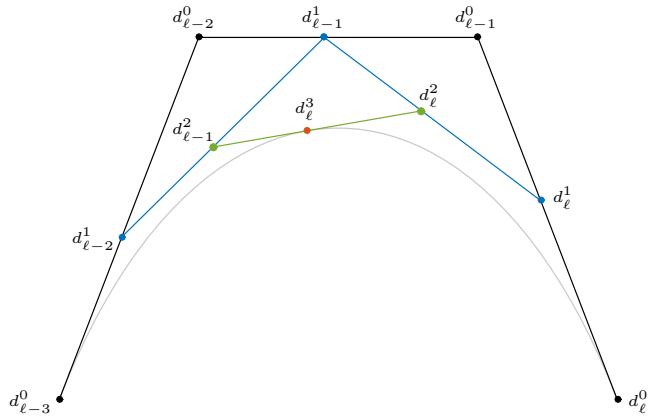


Abb. 3.3: Auswertung einer Splinekurve vom Grad 3 mittels de Boor–Algorithmus.

Dann lassen sich die Ableitungen $s_m \boldsymbol{d}'(t)$ und $s_m \boldsymbol{d}''(t)$ durch

$$s_m \boldsymbol{d}'(t) = d'^{\,m-1}_{\ell}(t) = c_0 \left(d^{m-1}_{\ell}(t) - d^{m-1}_{\ell-1}(t) \right) \tag{3.21}$$

und

$$\begin{aligned} s_m \boldsymbol{d}''(t) = d''^{\,m-2}_{\ell}(t) &= c_0 \left(d'^{\,m-1}_{\ell}(t) - d'^{\,m-1}_{\ell-1}(t) \right) \\ &= c_0 \left(c_1 \left(d^{m-2}_{\ell}(t) - d^{m-2}_{\ell-1}(t) \right) - c_2 \left(d^{m-2}_{\ell-1}(t) - d^{m-2}_{\ell-2}(t) \right) \right) \end{aligned} \tag{3.22}$$

berechnen. Diese beiden Formeln können in einem vierten Schritt des Algorithmus ausgerechnet werden. Hierfür müssen allerdings alle Zwischenergebnisse aus Schritt drei gespeichert werden. Alternativ kann man den Schritt drei so erweitern, dass man die entsprechenden Ableitungen zum jeweiligen Zeitpunkt berechnet und dann die Koeffizienten d^k_j überschreibt. Insgesamt lässt sich der de Boor–Algorithmus so verändern, dass alle nicht trivialen Ableitungen mit berechnet werden können.

3.1.3 Approximation

Mit Hilfe von Satz 3.15 ist klar, welche Bedingungen Interpolationsstellen erfüllen müssen, damit eine Interpolation möglich bzw. das entstehende Gleichungssystem lösbar ist. Eine Bedingung, die in der Praxis zu größeren Einschränkungen führt, ist, dass die Anzahl der Basisfunktionen mit der Anzahl der zu interpolierenden Stellen übereinstimmen muss. Oft ist es der Fall, dass die Anzahl der Datenpunkte so groß wird, dass eine Interpolation allein durch die hohe Anzahl an Basisfunktionen einfach nicht mehr praktikabel ist. Ein weiteres, bekanntes Problem der Interpolation ist das sogenannte Überschwingen des Interpolanten. Wählt man einen natürlichen Splineinterpolanten, wird ein Krümmungsintegral minimiert und das Überschwingen reduziert jedoch nie völlig ausgeschlossen. Diese Probleme können umgangen werden, wenn man die Daten „nur noch“ approximiert.

Ein möglicher Ansatz der Approximation diskreter Daten ist durch den sogenannten Schoenberg–Operator gegeben. Durch ihn können reelle Funktionen sehr leicht durch Splines angenähert werden. Solange die erste bzw. zweite Ableitung dieser Funktionen dann auch noch beschränkt ist, lässt sich der Approximationsfehler abschätzen [4].

Definition 3.20 *Zu einer gegebenen Knotenfolge $T = T_{n,m}$ ist der* Schoenberg–Operator $\mathscr{S}_m : C(\mathcal{I}_m) \to \mathbb{S}_m(T)$ *zu einer Funktion $f \in C(\mathcal{I}_m)$ definiert durch*

$$\mathscr{S}_m f := \sum_{j=1}^{n} f(\hat{t}_j) N_j^m(\cdot \mid T), \tag{3.23}$$

wobei $\hat{t}_j$, $j = 1, \ldots, n$, den Greville–Abszissen zum Knotenvektor T entsprechen.

Satz 3.21 *Der Schoenberg–Operator $\mathscr{S}_m$ besitzt lineare Exaktheit, das heißt, $\mathscr{S}_m f = f$ für $f \in \Pi_1$ und $m \geq 1$.*

Ein Beweis des letzten Satzes findet man zum Beispiel in [45].

Satz 3.22 *Für eine Funktion $f \in C^2(\mathcal{I}_m(T))$ kann der Approximationsfehler des Schoenberg–Operators mit*

$$\|\mathscr{S}_m f - f\|_\infty := \max_{x \in \mathcal{I}_m(T)} |f(x) - \mathscr{S}_m f(x)| \leq m^2 \|f''\|_\infty h^2 \tag{3.24}$$

abgeschätzt werden, wobei

$$h = \max_{j=1,\ldots,n+m} |\tau_{j+1} - \tau_j|$$

der maximale Knotenabstand im Knotenvektor T ist.

Beweis: Der Beweis erfolgt nach [45]. Sei $I_j := [\tau_j, \tau_{j+1})$ ein nicht triviales Knotenintervall und $J_j = [\tau_{j-m+1}, \tau_{j+m}]$ dasjenige Intervall, welches alle Greville–Abszissen enthält, die zur Auswertung von $\mathscr{S}_m f$ auf dem Intervall I_j beitragen. Weiter sei $t^* = \frac{1}{2}(\tau_{j-m+1} + \tau_{j+m})$ der Mittelpunkt des Intervalls J_j. Eine Taylorentwicklung von f um den Punkt t^* ergibt

$$f(x) = f(t^*) + (x - t^*) f'(t^*) + \frac{(x - t^*)^2}{2} f''(\xi) = T_1 f(t^*) + \frac{(x - t^*)^2}{2} f''(\xi), \quad \xi \in (x, t^*).$$

Es folgt für $x \in J_j$

$$\begin{aligned} |f(x) - T_1 f(x)| = \frac{|x - t^*|^2}{2} |f''(\xi)| &\leq \frac{\frac{1}{4}(\tau_{j+m} - \tau_{j-m+1})^2}{2} \max_{x \in J_j} |f''(x)| \\ &\leq \frac{m^2 h^2}{2} \|f''\|_\infty , \end{aligned}$$

da $\tau_{j+m} - \tau_{j-m+1} \leq (2m-1)h$. Insbesondere erhält man für die Greville–Abszissen $\hat{t}_k$, $k = j - m, \ldots, j$,

$$|f - T_1 f| \left(\hat{t}_k\right) \leq \frac{m^2 h^2}{2} \|f''\|_\infty \tag{3.25}$$

und damit

$$\mathscr{S}_m (f - T_1 f)\Big|_{I_j} \leq \frac{m^2 h^2}{2} \|f''\|_\infty \sum_{k=j-m}^{j} N_k^m(\cdot \mid T) \leq \frac{m^2 h^2}{2} \|f''\|_\infty . \tag{3.26}$$

Aus (3.25) und (3.26) erhält man schließlich

$$\begin{aligned} \max_{x \in I_j} |\mathscr{S}_m f - f|(x) &\leq \max_{x \in I_j} |\mathscr{S}_m (f - T_1 f)(x)| + \max_{x \in I_j} |(f - T_1 f)(x)| \\ &\leq m^2 h^2 \|f''\|_\infty . \end{aligned}$$

Da die rechte Seite unabhängig von j ist, gilt die Aussage für alle Intervalle. □

Der Beweis des Satzes 3.22 wurde hier aus verschiedenen Gründen ausgeführt. Zum einen ist die Vorgehensweise in diesem Beweis ähnlich der Vorgehensweise des Beweises zu Satz 4.7 über die Abklingrate der Waveletkoeffizienten, zum anderen kann man an der lokalen Version von Satz 3.22

$$\max_{x\in I_j} |\mathscr{S}_m f - f|(x) \le m^2 h^2 \max_{x\in J_j} |f''(x)|, \qquad j = m+1,\dots,n,$$

erkennen, dass der Schoenberg–Splineapproximant dort gut ist, wo die zweite Ableitung der Funktion klein ist. Anders gesagt, an den Stellen, an denen die zweite Ableitung groß wird, kann man durch eine Verkleinerung von h die Approximationsgüte des Splineapproximanten verbessern. Für die Approximation ist es somit eine gute Wahl die Knoten dort dicht zu legen, an denen eine große zweite Ableitung erwartet wird. Eine ähnliche Aussage zum Thema Knotenwahl findet man in [18].

3.1.4 Duale Funktionale

Aus der Definition 3.10 eines Splineraums $\mathbb{S}_m(T)$ wird klar, dass jedes Polynom $p \in \Pi_m$ auch im Splineraum $\mathbb{S}_m(T)$ liegt. Es gilt also $\Pi_m|_{\mathcal{I}_m(T)} \subset \mathbb{S}_m(T)$. Mit Hilfe des folgenden Satzes von Marsden können die so genannten de Boor–Fix Funktionale $\lambda_{j,m}$ hergeleitet werden [19]. Mit diesen Funktionalen kann ein Polynom $p \in \Pi_m$ zur Basis von $\mathbb{S}_m(T)$ dargestellt werden:

$$p(x) = \sum_{j=1}^{n} \lambda_{j,m}(p)\, N_j^m(x \mid T), \quad x \in \mathcal{I}_m(T).$$

Satz 3.23 (Marsden–Identität) *Sei $T = T_{n,m}$ ein Knotenvektor nach und $N_j^m(\cdot \mid T)$ die korrespondierenden B–Splines nach Definition 3.2. Für $t \in \mathbb{R}$ gilt:*

$$(\cdot - t)^m = \sum_{j=1}^{n} \psi_{j,m}(t)\, N_j^m(\cdot \mid T), \tag{3.27}$$

wobei

$$\psi_{j,m}(t) := (\tau_{j+1} - t)\cdots(\tau_{j+m} - t) = \prod_{k=j+1}^{j+m} (\tau_k - t)$$

ein Polynom vom Grad $m-1$ ist, welches auf den inneren Knoten von $\operatorname{supp} N_j^m(\cdot \mid T)$ verschwindet.

Ein Beweis dieser Identität und die folgende Herleitung der *de Boor–Fix Funktionale* $\lambda_{j,m}$ findet man in [18, Kap. IX, S. 95]. Um die gewünschten Funktionale $\lambda_{j,m}$ herzuleiten, teilt man (3.27) zunächst durch $m!$

$$\frac{(\cdot - t)^m}{m!} = \frac{1}{m!} \sum_{j=1}^{n} \psi_{j,m}(t)\, N_j^m(\cdot \mid T).$$

Als nächstes differenziert man beide Seiten dieses Terms ν-mal nach t, $\nu \ge 0$, und erhält

$$(-1)^\nu \frac{(\cdot - t)^{m-\nu}}{(m-\nu)!} = \frac{1}{m!} \sum_{j=1}^{n} \frac{\mathrm{d}^\nu}{\mathrm{d}t^\nu} \psi_{j,m}(t)\, N_j^m(\cdot \mid T)$$

$$\frac{(\cdot - t)^{m-\nu}}{(m-\nu)!} = \frac{(-1)^\nu}{m!} \sum_{j=1}^{n} \frac{\mathrm{d}^\nu}{\mathrm{d}t^\nu} \psi_{j,m}(t)\, N_j^m(\cdot \mid T). \tag{3.28}$$

Für ein beliebiges Polynom $p \in \Pi_m$ wird zunächst eine Taylorentwicklung um $t \in \mathcal{I}_m(T) \subset \mathbb{R}$ betrachtet

$$p(x) = \sum_{\nu=0}^{m} \frac{(x-t)^{m-\nu}}{(m-\nu)!} \frac{\mathrm{d}^{m-\nu}}{\mathrm{d}t^{m-\nu}} p(t).$$

Anschließend nutzt man die Identität (3.28) um die Taylorentwicklung umzuschreiben. Es folgt

$$\begin{aligned} p(x) &= \sum_{\nu=0}^{m} \frac{(-1)^\nu}{m!} \sum_{j=1}^{n} \frac{\mathrm{d}^\nu}{\mathrm{d}t^\nu} \psi_{j,m}(t)\, N_j^m(x \mid T) \frac{\mathrm{d}^{m-\nu}}{\mathrm{d}t^{m-\nu}} p(t) \\ &= \sum_{j=1}^{n} \sum_{\nu=0}^{m} \frac{(-1)^\nu}{m!} \frac{\mathrm{d}^\nu}{\mathrm{d}t^\nu} \psi_{j,m}(t) \frac{\mathrm{d}^{m-\nu}}{\mathrm{d}t^{m-\nu}} p(t)\, N_j^m(x \mid T). \end{aligned} \tag{3.29}$$

Die Gleichung (3.29) ist eine B–Spline Darstellung des Polynoms p auf dem Intervall $\mathcal{I}_m(T)$. Die Koeffizienten dieser Darstellung entsprechen gerade den im Folgenden definierten de Boor-Fix Funktionalen.

Definition 3.24 *Sei $f \in \Pi_m$, $T = T_{n,m}$ ein Knotenvektor mit dazugehörigen B–Splines $N_j^m(\cdot \mid T)$. Dann sind die* de Boor–Fix Funktionale $\lambda_{j,m}$ *definiert durch*

$$\lambda_{j,m}(f) := \sum_{\nu=0}^{m} \frac{(-1)^\nu}{m!} \frac{\mathrm{d}^\nu}{\mathrm{d}t^\nu} \psi_{j,m}(t_j) \frac{\mathrm{d}^{m-\nu}}{\mathrm{d}t^{m-\nu}} f(t_j), \tag{3.30}$$

wobei $t_j \in [\tau_j, \tau_{j+m+1}]$, $j = 1, \ldots, n$, und

$$\psi_{j,m}(t) := \prod_{k=j+1}^{j+m} (\tau_k - t).$$

Die ν–te Ableitung von $\psi_{j,m}$ kann durch

$$\frac{\mathrm{d}^\nu}{\mathrm{d}t^\nu} \psi_{j,m}(t) = (-1)^\nu\, \nu! \sum_{\substack{K \subseteq \{1,\ldots,m\} \\ \#K = m-\nu}} \prod_{k \in K} (\tau_{j+k} - t), \qquad \nu = 0, \ldots, m, \tag{3.31}$$

berechnet werden. Damit ein Polynom $p \in \Pi_m$ im Splineraum $\mathbb{S}_m(T)$ dargestellt werden kann, ist die Bedingung, dass t_j im Intervall $[\tau_j, \tau_{j+m+1}]$ liegen muss, nicht nötig. Es würde $t \in \mathbb{R}$ reichen, wie man an (3.29) sieht. Damit aber die Funktionale $\lambda_{j,m}$ dual zu den B–Splines sind, ist eben diese Einschränkung nötig. Dual heißt in diesem Fall

$$\lambda_{i,m}(N_j^m(\cdot \mid T)) = \delta_{ij}.$$

Mit Hilfe dieser Bedingung sind aber nicht nur Polynome, sondern auch stückweise Polynome im Splineraum durch die Funktionale $\lambda_{j,m}$ darstellbar.

3.1.5 Verfeinerung

Unter der Verfeinerung eines Knotenvektors T versteht man das Hinzufügen von Knoten zu selbigem. Durch den Knotenvektor wird implizit die Anzahl der Basisfunktionen und somit die Dimension eines Splineraums vorgegeben. Somit ist klar, dass durch das Einfügen eines Knotens in einen Knotenvektor ein neuer Splineraum entsteht. Im folgenden Abschnitt wird gezeigt wie

diese Räume miteinander verbunden sind und wie man mit Hilfe von Knoteneinfügealgorithmen für eine Splinekurve aus einem groben Splineraum die Koeffizienten respektive Kontrollpunkte für den feinen Splineraum berechnet. Zwei der bekanntesten Algorithmen zum Knoteneinfügen sind der Boehm– [5] bzw. Oslo–Algorithmus [10]. Sie unterscheiden sich im Wesentlichen darin, dass beim Boehm–Algorithmus Knoten sukzessive eingefügt werden, beim Oslo–Algorithmus hingegen werden mehrere Knoten simultan eingefügt, das heißt, es wird nur eine Knoteneinfügematrix für alle einzufügenden Knoten erzeugt. Für die Theorie ist jedoch nur der Punkt wichtig, dass in beiden Fällen eine Knoteneinfügematrix aufgestellt werden kann. In der Praxis findet die Verfeinerung eines Knotenvektors vor allem in der Graderhöhung einer Spline– bzw. Bézier–Kurve Anwendung. Aber nicht nur dort. Wie in Kapitel 4 gezeigt wird, ist die Verfeinerung einer Knotenfolge, genauer: das Knotenentfernen, das wichtigste Werkzeug der in dieser Arbeit vorgestellten Spline–Multiresolution. Mehr zum Thema Knoteneinfügen und –entfernen findet man zum Beispiel in [24].

Definition 3.25 *Seien $T = T_{n,m}$ sowie $T^* = T^*_{n^*,m}$ zwei Knotenvektoren mit jeweils $(m+1)$-fachen, identischen Randknoten, also $\tau_1 = \tau_1^*, \ldots, \tau_{m+1} = \tau^*_{m+1}$ und $\tau_{n+1} = \tau^*_{n^*+1}, \ldots, \tau_{n+m+1} = \tau^*_{n^*+m+1}$. Dann heißt T^** Verfeinerung *von T, in Zeichen $T \subseteq T^*$, falls es eine streng monoton steigende Funktion*

$$\nu : \{m+1, \ldots, n-1\} \to \{m+1, \ldots, n^*-1\}$$

gibt, so dass

$$\tau^*_{\nu(j)} = \tau_j, \quad j = m+1, \ldots, n-1.$$

Die Definition 3.25 verlangt insbesondere, dass nur Knoten im Inneren eines Knotenvektors eingefügt werden.

Ein Beweis des folgenden Satzes ist sehr leicht und beruht darauf zu zeigen, dass jedes Polynom bezüglich der Knotenfolge T über einem Knotenintervall $[\tau_j, \tau_{j+1}]$ und somit eigentlich jedes stückweise Polynom auch ein Polynom zur Knotenfolge T^* bleibt.

Satz 3.26 *Ist $T \subseteq T^*$, dann ist $\mathbb{S}_m(T) \subset \mathbb{S}_m(T^*)$.*

Satz 3.26 besagt, dass sich jedes Element aus $\mathbb{S}_m(T)$ eindeutig bezüglich der Basis von $\mathbb{S}_m(T^*)$ darstellen lässt. Es gilt also

$$\mathbb{S}_m(T) \ni s_m\boldsymbol{d}(\cdot \mid T) = \sum_{j=1}^{n} d_j N_j^m(\cdot \mid T) = \sum_{j=1}^{n^*} d_j^* N_j^m(\cdot \mid T^*) = s_m\boldsymbol{d}^*(\cdot \mid T) \in \mathbb{S}_m(T^*). \quad (3.32)$$

Zu jeder Koeffizientenmatrix $\boldsymbol{d}$ zur Basis von $\mathbb{S}_m(T)$ gibt es demnach eine entsprechende Koeffizientenmatrix $\boldsymbol{d}^*$ bezüglich der Basis von $\mathbb{S}_m(T^*)$, so dass beide Koeffizientenmatrizen bezüglich ihrer Basis die gleiche Kurve beschreiben. Interessant ist, wie man nach dem Einfügen eines Knotens die entsprechenden Koeffizienten $\boldsymbol{d}^*$ berechnen kann. Es ist klar, dass der Prozess der Verfeinerung eines Knotenvektors ein lokaler Prozess ist. Fügt man zum Beispiel einen Knoten $\tau^* \in [\tau_j, \tau_{j+1})$ in den Knotenvektor T ein, sind nur diejenigen B–Splines davon betroffen, für die $[\tau_j, \tau_{j+1}) \subseteq \operatorname{supp}(N_k^m(\cdot \mid T))$ gilt. Das sind die B–Splines $N^m_{j-m}, \ldots, N^m_j$, siehe Abbildung 3.4. Es sind also nur die Koeffizienten $d_{j-m}, \ldots, d_j$ für die Berechnung von $\boldsymbol{d}^*$ von besonderem Interesse. Alle anderen Basisfunktionen und ihre Koeffizienten bleiben erhalten.

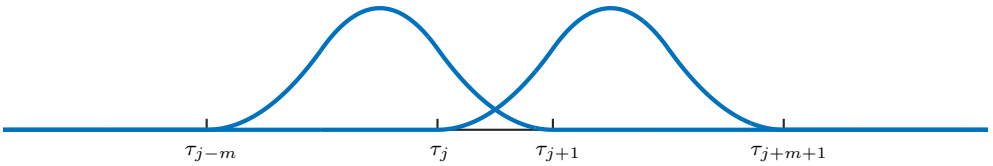


Abb. 3.4: Die B–Splines vom Grad m, die am weitesten links (N^m_{j-m}) bzw. rechts (N^m_j) liegen und über dem Intervall $[\tau_j, \tau_{j+1}]$ nicht Null sind.

Da man für jeden eingefügten Knoten die Anzahl der Basisfunktionen um eins erhöht, gilt für die Dimension der Vektorräume $\mathbb{S}_m(T)$ und $\mathbb{S}_m(T^*)$ nach Einfügen eines Knotens

$$\dim \mathbb{S}_m(T^*) = \dim \mathbb{S}_m(T) + 1. \tag{3.33}$$

Knoteneinfügen

Der folgende Algorithmus, auch Boehm–Algorithmus genannt, berechnet zu einem vorgegebenen Knotenvektor T, einer Koeffizientenmatrix $\boldsymbol{d}$ und einem neuen, einzufügenden Knoten $\tau^* \in [\tau_j, \tau_{j+1})$ die entsprechende Koeffizientenmatrix $\boldsymbol{d}^*$, so dass $s_m\boldsymbol{d}(\cdot \mid T) = s_m\boldsymbol{d}^*(\cdot \mid T^*)$ mit $T^* = T \cup \{\tau^*\}$. Dieser Algorithmus ähnelt dem de Boor–Algorithmus (3.20), da bei diesem an der Auswertungsstelle t im Prinzip ein m-facher Knoten eingefügt wurde, so dass die Kurve den korrespondierenden Kontrollpunkt interpoliert.

Algorithmus 3.27 (Boehm-Algorithmus zum Knoteneinfügen)
Eingabe:

- *Knotenfolge* $T = T_{m,n}$
- *neuer Knoten* τ^* *mit* $\tau_j \le \tau^* < \tau_{j+1}$

Prozedur:

1. **FOR** $k = 1, \ldots, j-m$
$$d^*_k := d_k.$$
END

2. **FOR** $k = j-m+1, \ldots, j$
$$d^*_k := \frac{\tau_{k+m} - \tau^*}{\tau_{k+m} - \tau_k} d_{k-1} + \frac{\tau^* - \tau_k}{\tau_{k+m} - \tau_k} d_{k-1}. \tag{3.34}$$
END

3. **FOR** $k = j+1, \ldots, n+1$
$$d^*_k := d_{k-1}.$$
END

Ergebnis:
Kontrollpunkte $\boldsymbol{d}^*$*, so dass:*

$$s_m\boldsymbol{d}^*(\cdot \mid T^*) = s_m\boldsymbol{d}(\cdot \mid T).$$

Dieser Algorithmus kann mit Hilfe von linearer Algebra wie folgt formalisiert werden.

Definition 3.28 (Knoteneinfügematrix) *Die Knoteneinfügematrix $\boldsymbol{A} = \boldsymbol{A}(\tau^*)$, die einen Knoten τ^* mit $\tau_j \le \tau^* < \tau_{j+1}$ in die Knotenfolge T einfügt, ist definiert als:*

$$\boldsymbol{d}^* = \boldsymbol{d}\boldsymbol{A}\left(\tau^*\right), \quad \boldsymbol{A}\left(\tau^*\right) := \begin{bmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & \alpha_{j-m+1} & & & & & \\ & & & 1-\alpha_{j-m+1} & \ddots & & & & \\ & & & & \ddots & \alpha_j & & & \\ & & & & & 1-\alpha_j & 1 & & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{bmatrix} \in \mathbb{R}^{n\times(n+1)},$$

wobei $\alpha_k = \dfrac{\tau_{k+m} - \tau^*}{\tau_{k+m} - \tau_k}$ *für* $k = j-m+1, \ldots, j$.

Sollen mehrere Knoten $\boldsymbol{\tau}^* = (\tau_1^*, \ldots, \tau_k^*)$ in einen Knotenvektor eingefügt werden, kann dies durch sukzessives Einfügen der Knoten in den Knotenvektor T geschehen. Formal wird dies als

$$\mathbb{R}^{n\times(n+k)} \ni \boldsymbol{A}(\boldsymbol{\tau}) := \boldsymbol{A}(\tau_1^*, \ldots, \tau_k^*) := \boldsymbol{A}(\tau_k^*) \cdots \boldsymbol{A}(\tau_1^*) \tag{3.35}$$

notiert (siehe [45]). Dabei haben die einzelnen Knoteneinfügematrizen $\boldsymbol{A}(\tau_\ell^*)$, $\ell = 1, \ldots, k$, alle unterschiedliche Dimensionen. Entscheidend für die Dimension ist die Position, an der die Matrix $\boldsymbol{A}(\tau_\ell^*)$ in (3.35) steht. Der Begriff der Kommutativität im folgenden Lemma hat nichts mit dem Kommutieren der Knoteneinfügematrizen zu tun. Diese kommutieren schon aus Dimensionsgründen nicht, da sie inkrementell berechnet werden und von der Länge des aktuellen Knotenvektors abhängen. Vielmehr ist hier die Reihenfolge, in der die Knoten eingefügt werden, gemeint.

Lemma 3.29 *Sei $T = T_{n,m}$ eine Knotenfolge und τ_1^* sowie τ_2^* zwei Knoten, die in die Knotenfolge T eingefügt werden sollen. Dann ist der Knoteneinfügeprozess kommutativ und es gilt*

$$\boldsymbol{A}(\tau_1^*, \tau_2^*) = \boldsymbol{A}(\tau_1^*)\boldsymbol{A}(\tau_2^*) = \boldsymbol{A}(\tau_2^*)\boldsymbol{A}(\tau_1^*) = \boldsymbol{A}(\tau_2^*, \tau_1^*).$$

Beweis: Sei $T \subset T^*$, τ_1^* sowie τ_2^* die Knoten aus T^*, die nicht in T vorkommen. $\boldsymbol{A}(\tau_1^*, \tau_2^*)$ ist die Knoteneinfügematrix, die erst τ_1^* und dann τ_2^* in T einfügt. Analog $\boldsymbol{A}(\tau_2^*, \tau_1^*)$. Weiter sei $s_m\boldsymbol{d} \in \mathbb{S}_m(T)$ ein Spline zur Knotenfolge T mit Koeffizienten $\boldsymbol{d}$.

Für die beiden Splinekurven

$$f := f_m\boldsymbol{d}\boldsymbol{A}(\tau_1^*, \tau_2^*)(\cdot \mid T^*) = \sum_{j=1}^{n+2} f_j N_j^m(\cdot \mid T^*) \qquad \text{und}$$

$$g := g_m\boldsymbol{d}\boldsymbol{A}(\tau_2^*, \tau_1^*)(\cdot \mid T^*) = \sum_{j=1}^{n+2} g_j N_j^m(\cdot \mid T^*)$$

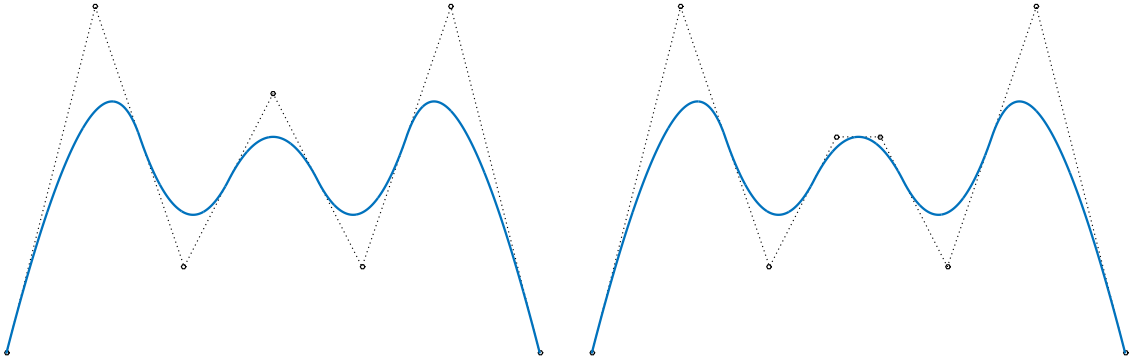


Abb. 3.5: Splinekurve und Kontrollpolygon vor (links) und nach (rechts) dem Einfügen eines Knotens

gilt nach Satz 3.26 $f = s_m \boldsymbol{d} = g$. Wegen der Eindeutigkeit einer Basisdarstellung folgt

$$\begin{aligned}
\sum_{j=1}^{n+2} f_j N_j^m(\cdot \mid T^*) &= \sum_{j=1}^{n+2} g_j N_j^m(\cdot \mid T^*) \\
\sum_{j=1}^{n+2} d_j \boldsymbol{A}(\tau_1^*, \tau_2^*) N_j^m(\cdot \mid T^*) &= \sum_{j=1}^{n+2} d_j \boldsymbol{A}(\tau_2^*, \tau_1^*) N_j^m(\cdot \mid T^*) \\
\boldsymbol{d}\boldsymbol{A}(\tau_1^*, \tau_2^*) &= \boldsymbol{d}\boldsymbol{A}(\tau_2^*, \tau_1^*) \\
\boldsymbol{d}(\boldsymbol{A}(\tau_1^*, \tau_2^*) - \boldsymbol{A}(\tau_2^*, \tau_1^*)) &= 0 \\
\boldsymbol{A}(\tau_1^*, \tau_2^*) &= \boldsymbol{A}(\tau_2^*, \tau_1^*).
\end{aligned}$$

□

Die Knoteneinfügematrizen $\boldsymbol{A}$ sind bandiert und dünn besetzt, somit lässt sich bei mehreren einzufügenden Knoten $\boldsymbol{A}(\boldsymbol{\tau})$ in der Regel schneller berechnen als das wiederholte Ausführen von Algorithmus 3.27. Abbildung 3.5 zeigt eine Splinekurve mit ihrem Kontrolpolygon vor und nach dem Knoteneinfügen.

Im Abschnitt 4.2 wird als abkürzende Schreibweise die Notation $\boldsymbol{A}(T, T^*)$ benutzt. Diese Notation bedeutet, dass $\boldsymbol{A}$ die Knoteneinfügematrix ist, die alle Knoten aus T^*, die nicht in T vorkommen, in den Knotenvektor T einfügt.

Knotenentfernen

Das Knotenentfernen kann als inverser Prozess des Knoteneinfügens verstanden werden. Im Gegensatz zum Knoteneinfügen ist das Knotenentfernen im Allgemeinen kein exakter Prozess, da $\mathbb{S}_m(T) \subset \mathbb{S}_m(T^*)$. Hier ist die Frage ähnlich der beim Knoteneinfügen: Welche Koeffizienten hat eine Splinekurve aus einem feinen Splineraum $\mathbb{S}_m(T^*)$ im gröberen Raum $\mathbb{S}_m(T)$? Dank der Knoteneinfügematrix $\boldsymbol{A}$ aus Definition 3.28 kann man einen direkten Zusammenhang von Koeffizienten aus einem groben zu Koeffizienten eines feinen Splineraum angeben.

Um einen Knoten τ_j^* aus dem Knotenvektor $T^* = T^*_{n^*,m} = (\tau_1^*, \ldots, \tau^*_{n^*+m+1})$ zu entfernen, ist die Vorstellung hilfreich, dass der Knoten zuvor in die Knotenfolge T eingefügt wurde. Man erhält somit eine Beziehung zwischen dem gesuchten Kontrollpolygon $\boldsymbol{d}$ und dem gegebenen Kontrollpolygon $\boldsymbol{d}^*$

$$\boldsymbol{d}^* = \boldsymbol{d}\boldsymbol{A}(\tau_j^*).$$

Im Allgemeinen kann man aber nicht davon ausgehen, dass der Knoten τ_j^* zuvor eingefügt wurde, so dass die Splinekurve aus $\mathbb{S}_m(T)$ die Ausgangskurve nicht mehr reproduziert, sondern nur noch approximiert. Um diese Approximation so gut wie möglich durchzuführen, gibt es verschiedene Ansätze, siehe hierfür zum Beispiel [35, 36] und [24]. Der in dieser Arbeit verfolgte Ansatz ist die Minimierung des Abstands zwischen dem Kontrollpolygon der Originalkurve und dem der approximierenden Kurve aus $\mathbb{S}_m(T)$. Hierfür fügt man den zuvor entfernten Knoten wieder ein und berechnet die Differenz der Koeffizienten. Formal erhält man hierfür eine Lösung für $\boldsymbol{d}$ durch Minimierung der Funktionale

$$\min_{\boldsymbol{d}} \left\| \boldsymbol{e}_\ell^T \left(\boldsymbol{d}\boldsymbol{A}(\tau_j^*) - \boldsymbol{d}^* \right) \right\|_2, \tag{3.36}$$

wobei $\boldsymbol{e}_\ell$ dem ℓ–ten Einheitsvektor des Bildraums $\mathbb{R}^k$ entspricht. Die Minimierung von $\boldsymbol{d}\boldsymbol{A}(\tau_j^*) - \boldsymbol{d}^*$ mittels (3.36) ist komponentenweise (zeilenweise, also über die Dimensionen des Bildraumes) zu verstehen.

Zur Übersicht wird im Folgenden die Knoteneinfügematrix mit $\boldsymbol{A}(\tau_j^*) = \boldsymbol{A}$ abgekürzt. In Anlehnung an die Matlab–Notation zur Auswahl einer Zeile bzw. Spalte einer Matrix wird die Indexnotation $\boldsymbol{d}_{j,:}$ bzw. $\boldsymbol{d}_{:,j}$ für die Auswahl der j–ten Zeile bzw. Spalte aus $\boldsymbol{d}$ verwendet. Die Herleitung der Lösung des Least–Squares–Ansatzes ist wohlbekannt und wird hier der Vollständigkeit halber aufgeführt.

Sei $F_\ell(\boldsymbol{x}) := \left\| \boldsymbol{e}_\ell^T \left(\boldsymbol{x}\boldsymbol{A} - \boldsymbol{d}^* \right) \right\|_2^2 \in \mathbb{R}$ mit $\boldsymbol{x} \in \mathbb{R}^{k \times n}$, $\ell = 1, \ldots, k$, wobei k der Dimension des Bildraumes der Splinekurve $s_m \boldsymbol{d}$ entspricht. Zur Minimierung von (3.36) sucht man die Nullstelle des Gradienten von F_ℓ,

$$\nabla F_\ell(\boldsymbol{x}) = \boldsymbol{0}. \tag{3.37}$$

Diese Nullstelle existiert, da F_ℓ in allen Komponenten quadratisch ist. Für F_ℓ gilt:

$$\begin{aligned} F_\ell(\boldsymbol{x}) &= \left\| \boldsymbol{e}_\ell^T \left(\boldsymbol{x}\boldsymbol{A} - \boldsymbol{d}^* \right) \right\|_2^2 = \left\| \boldsymbol{e}_\ell^T \, \boldsymbol{x} \, \boldsymbol{A} - \boldsymbol{e}_\ell^T \, \boldsymbol{d}^* \right\|_2^2 \\ &= (\boldsymbol{e}_\ell^T \, \boldsymbol{x}\boldsymbol{A})(\boldsymbol{e}_\ell^T \, \boldsymbol{x}\boldsymbol{A})^T - 2\,(\boldsymbol{e}_\ell^T \, \boldsymbol{x}\boldsymbol{A})(\boldsymbol{e}_\ell^T \, \boldsymbol{d}^*)^T + (\boldsymbol{e}_\ell^T \, \boldsymbol{d}^*)(\boldsymbol{e}_\ell^T \, \boldsymbol{d}^*)^T \\ &= \boldsymbol{e}_\ell^T \, \boldsymbol{x}\boldsymbol{A} \, \boldsymbol{A}^T \boldsymbol{x}^T \boldsymbol{e}_\ell - 2\,\boldsymbol{e}_\ell^T \, \boldsymbol{x}\boldsymbol{A} \, \boldsymbol{d}^{*T} \boldsymbol{e}_\ell + \boldsymbol{e}_\ell^T \, \boldsymbol{d}^* \boldsymbol{d}^{*T} \boldsymbol{e}_\ell \\ &= \boldsymbol{x}_{\ell,:} \, \boldsymbol{A} \, \boldsymbol{A}^T \boldsymbol{x}_{\ell,:}^T - 2\,\boldsymbol{x}_{\ell,:} \boldsymbol{A} \, \boldsymbol{d}_{\ell,:}^{*\,T} + \boldsymbol{d}_{\ell,:}^* \boldsymbol{d}_{\ell,:}^{*\,T}. \end{aligned} \tag{3.38}$$

Für die Betrachtung einer partiellen Ableitung $\frac{\partial}{\partial x_{\ell j}} F_\ell(\boldsymbol{x})$ sei zur Vereinfachung der Notation $\widetilde{\boldsymbol{A}} := \boldsymbol{A}\,\boldsymbol{A}^T = (\tilde{a}_{ij})$ und somit

$$\begin{aligned} \frac{\partial}{\partial x_{\ell j}} F_\ell(\boldsymbol{x}) &= \frac{\partial}{\partial x_{\ell j}} \left(\boldsymbol{x}_{\ell,:} \widetilde{\boldsymbol{A}} \, \boldsymbol{x}_{\ell,:}^T - 2\,\boldsymbol{x}_{\ell,:} \boldsymbol{A} \, \boldsymbol{d}_{\ell,:}^{*\,T} + \boldsymbol{d}_{\ell,:}^* \boldsymbol{d}_{\ell,:}^{*\,T} \right) \\ &= \frac{\partial}{\partial x_{\ell j}} \left(\sum_{p=1}^{n} x_{\ell p}^2 \tilde{a}_{pp} + \sum_{p=1}^{n} \sum_{\substack{q=1 \\ p \neq q}}^{n} x_{\ell q} \tilde{a}_{pq} x_{\ell p} - 2 \sum_{p=1}^{n} x_{\ell p} \left(\boldsymbol{A} \boldsymbol{d}_{\ell,:}^{*\,T} \right)_{:,p} + \sum_{p=1}^{n} d_{\ell p}^{*\,2} \right) \\ &= 2\,x_{\ell j} \tilde{a}_{jj} + 2 \sum_{\substack{p=1 \\ p \neq j}}^{n} x_{\ell p} \tilde{a}_{jp} - 2 \left(\boldsymbol{A} \boldsymbol{d}_{\ell,:}^{*\,T} \right)_{j,:} \\ &= 2 \left(\boldsymbol{x}_{\ell,:} \, \widetilde{\boldsymbol{A}} \right)_{:,j} - 2 \left(\boldsymbol{A} \, \boldsymbol{d}_{\ell,:}^{*\,T} \right)_{j,:} = 2 \left(\boldsymbol{x}_{\ell,:} \, \boldsymbol{A} \, \boldsymbol{A}^T \right)_{:,j} - 2 \left(\boldsymbol{A} \, \boldsymbol{d}_{\ell,:}^{*\,T} \right)_{j,:} \\ &= 2 \left(\boldsymbol{x}_{\ell,:} \, \boldsymbol{A} \, \boldsymbol{A}^T \right)_{:,j} - 2 \left(\boldsymbol{d}_{\ell,:}^* \boldsymbol{A}^T \right)_{:,j}. \end{aligned} \tag{3.39}$$

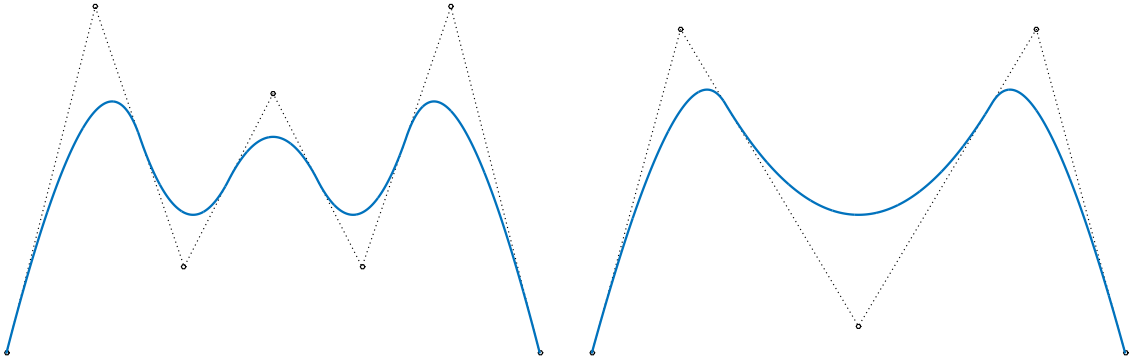


Abb. 3.6: Splinekurve und Kontrollpolygon vor (links) und nach (rechts) dem Entfernen zweier Knoten

Aus (3.39) erkennt man, dass zur Lösung von (3.37) die Normalengleichung

$$\begin{aligned} 2\,\boldsymbol{x}_{\ell,:}\,\boldsymbol{A}\boldsymbol{A}^T - 2\,\boldsymbol{d}^*_{\ell,:}\boldsymbol{A}^T &= \boldsymbol{0} \\ \boldsymbol{x}_{\ell,:}\,\boldsymbol{A}\boldsymbol{A}^T &= \boldsymbol{d}^*_{\ell,:}\boldsymbol{A}^T \end{aligned} \tag{3.40}$$

zu lösen ist. Man erhält somit die ℓ-te Komponente (Zeile) der gesuchten Koeffizientenmatrix $\boldsymbol{d}$. Um die gesamte Koeffizientenmatrix $\boldsymbol{d}$ zu erhalten, löst man demnach k Gleichungssysteme der Form (3.40). Abkürzend hierfür wird die Notation

$$\boldsymbol{d}\boldsymbol{A}\boldsymbol{A}^T = \boldsymbol{d}^*\boldsymbol{A}^T \tag{3.41}$$

eingeführt.

Definition 3.30 *Seien $T = T_{n,m} = (\tau_1, \ldots, \tau_{n+m+1})$ und $T^* = T^*_{n^*,m} = (\tau^*_1, \ldots, \tau^*_{n^*+m+1})$ Knotenfolgen mit $T \subset T^*$. Weiter sei $\boldsymbol{\tau}$ der Vektor mit allen Knoten aus T^*, die nicht in T vorkommen und $\boldsymbol{A}(\boldsymbol{\tau})$ die Knoteneinfügematrix, die $\boldsymbol{\tau}$ in T einfügt. Dann ist die* Knotenentfernmatrix $\boldsymbol{B} = \boldsymbol{B}(\boldsymbol{\tau})$ *definiert als*

$$\boldsymbol{B} = \boldsymbol{B}(\boldsymbol{\tau}) := \boldsymbol{A}(\boldsymbol{\tau})^T \left(\boldsymbol{A}(\boldsymbol{\tau})\boldsymbol{A}(\boldsymbol{\tau})^T\right)^{-1}. \tag{3.42}$$

Mit Hilfe der Knotenentfernmatrix kann nun das gesuchte Kontrollpolygon bzw. die Koeffizientenmatrix $\boldsymbol{d}$ durch $\boldsymbol{d} = \boldsymbol{d}^*\boldsymbol{B}$ berechnet werden. Dies ist aber nur von theoretischem Interesse, da für $\boldsymbol{B}$ eine Inverse von $\boldsymbol{A}\boldsymbol{A}^T$ berechnet werden muss. In der Praxis ist es numerisch wesentlich stabiler das Gleichungssystem (3.41) zu lösen, siehe [1, 25].

Analog zur Notation von $\boldsymbol{A}(T, T^*)$ beschreibt $\boldsymbol{B}(T, T^*)$ die Knotenentfernmatrix, die alle Knoten aus T^* entfernt, welche nicht in T vorkommen.

Entfernt man einen Knoten, den man zuvor hinzugefügt hat, erhält man aus dem Ausgleichsproblem (3.41) die ursprünglichen Koeffizienten der Splinekurve. Es gilt also $\boldsymbol{A}\boldsymbol{B} = \boldsymbol{I}$. Für die Umkehrung, also das Entfernen eines Knotens und wieder Einfügen von selbigem gilt in der Regel $\boldsymbol{B}\boldsymbol{A} \neq \boldsymbol{I}$. In Abbildung 3.6 sind eine Splinekurve mit dazugehörigem Kontrollpolygon vor und nach dem Entfernen eines Knotens abgebildet.

3.1.6 Tensorproduktsplines

Ein Tensorproduktspline ist eine Funktion f, die ein k-dimensionales Parametergebiet $\Omega \subset \mathbb{R}^k$ in einen h-dimensionalen Raum $\mathbb{R}^h$ abbildet. Kurz

$$f : \Omega \to \mathbb{R}^h, \quad \Omega \subset \mathbb{R}^k.$$

Für Tensorproduktsplines ist das Parametergebiet Ω stets ein k-dimensionaler Quader, der durch die entsprechenden Knotenvektoren bestimmt wird.

Zur Vereinfachung der Notation multivariater Splines bzw. Tensorproduktsplines werden zur Indizierung meist Multiindizes eingeführt. Für Indizes $a_1, \ldots, a_k \in \mathbb{N}_0$ wäre dies zum Beispiel $\boldsymbol{a} = (a_1, \ldots, a_k) \in \mathbb{N}_0^k$, ein k-Tupel über den natürlichen Zahlen. Für zwei Multiindizes $\boldsymbol{a} = (a_1, \ldots, a_k)$ und $\boldsymbol{b} = (b_1, \ldots, b_k)$ gelten die für Multiindizes üblichen Konventionen für Relationen und Operationen. Die zwei wichtigsten für diesen Abschnitt sind

$$\begin{aligned} \boldsymbol{a} \leq \boldsymbol{b} \quad &\iff \quad a_1 \leq b_1, \ldots, a_k \leq b_k, \\ \boldsymbol{a} + \boldsymbol{b} \quad &:= \quad (a_1 + b_1, \ldots, a_k + b_k). \end{aligned}$$

Mit Hilfe der Multiindizes kann die gesamte Notation für univariate Splines auf multivariate Splines übertragen werden.

Definition 3.31 *Sei $k \in \mathbb{N}$ und $\boldsymbol{\nu} = (n_1, \ldots, n_k)$ sowie $\boldsymbol{\mu} = (m_1, \ldots, m_k)$ Multiindizes für Grad bzw. Anzahl der Basisfunktionen zu den Knotenvektoren $T_j = T_{n_j, m_j} = (\tau_{j,1}, \ldots, \tau_{j,n_j+m_j+1})$ mit $j = 1, \ldots, k$. Dann ist ein* Tensorprodukt–Knotenvektor *bzw.* Knotenfolge *definiert als das kartesische Produkt aller Knotenvektoren T_j*

$$T := \bigotimes_{j=1}^{k} T_j := \{\tau_{\boldsymbol{\alpha}} = (\tau_{1,\alpha_1}, \ldots, \tau_{k,\alpha_k}) \mid \boldsymbol{\alpha} \leq \boldsymbol{\nu} + \boldsymbol{\mu} + \mathbf{1}\}, \tag{3.43}$$

wobei $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$ ein Multiindex ist.

Die so definierten Tensor–Knotenvektoren spannen für $k = 2$ ein zweidimensionales Gitter auf, siehe Abbildung 3.7.

Definition 3.32 *Für $k \in \mathbb{N}$, einen Tensor–Knotenvektor T, Abtastpunkte $\boldsymbol{t} = (t_1, \ldots, t_k) \in \mathbb{R}^k$ sowie einen Multiindex $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_k)$ ist der korrespondierende* Tensorprodukt–B–Splines *wie folgt definiert*

$$N_{\boldsymbol{\kappa}}^{\boldsymbol{\mu}}(\boldsymbol{t} \mid T) = \prod_{j=1}^{k} N_{\kappa_j}^{m_j}(t_j \mid T_j). \tag{3.44}$$

Definition 3.33 *Sei $k \in \mathbb{N}$, T ein Tensorprodukt–Knotenvektor nach und $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_k)$ ein Multiindex. Zu Kontrollpunkten $\boldsymbol{d} = [\boldsymbol{d}_{\boldsymbol{\kappa}} : \boldsymbol{\kappa} \leq \boldsymbol{\nu}]$ ist ein* Tensorprodukt–Spline *durch*

$$S_{\boldsymbol{\mu}}\boldsymbol{d} := S_{\boldsymbol{\mu}}(\cdot \mid T)\boldsymbol{d} := \sum_{\boldsymbol{\kappa} \leq \boldsymbol{\nu}} \boldsymbol{d}_{\boldsymbol{\kappa}} N_{\boldsymbol{\kappa}}^{\boldsymbol{\mu}}(\cdot \mid T) \tag{3.45}$$

definiert.

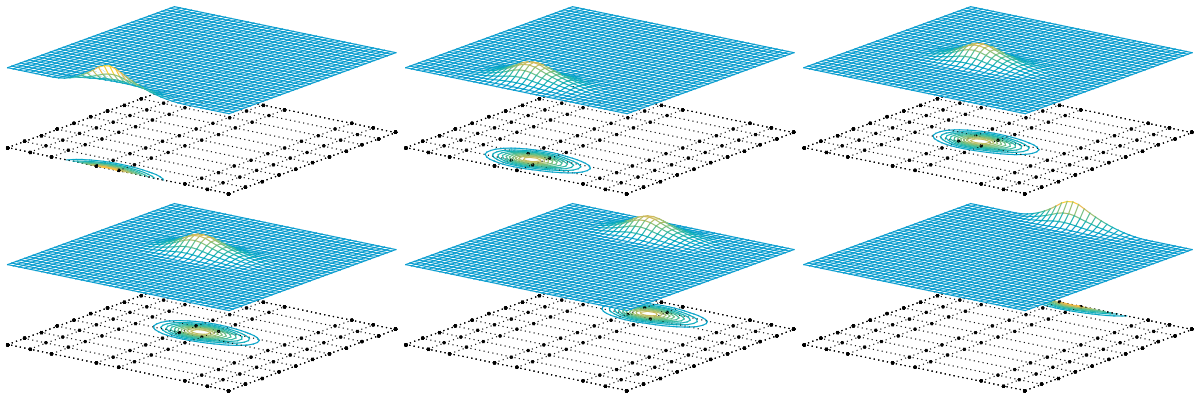


Abb. 3.7: Eine Auswahl von Tensorprodukt–B–Splines über einem Tensorprodukt–Knotengitter.

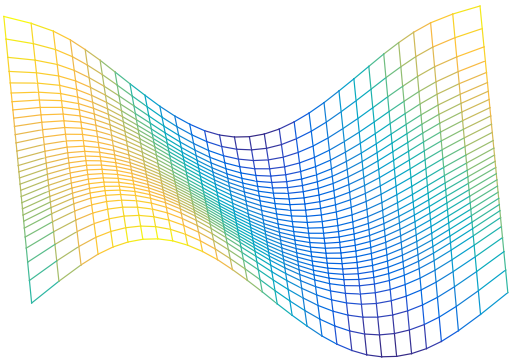


Abb. 3.8: Beispiel einer Tensorproduktsplinefläche zu Basisfunktionen aus Abb. 3.7.

Im weiteren Verlauf dieser Arbeit werden ausschließlich Tensorproduktsplines betrachtet, die einen zweidimensionalen Parameterbereich $\Omega \subset \mathbb{R}^2$ in den $\mathbb{R}^3$ abbilden. Das heißt für die Definitionen 3.31 - 3.33, dass für die Dimension des Parameterbereichs $k = 2$ ist. In Abbildung 3.7 sieht man eine Auswahl an Tensor–B–Splines vom Grad $(2, 2)$ über einem entsprechenden Knotengitter T. Wie im univariaten Fall ist der Träger des Tensor–B–Spline kompakt, siehe Abbildung 3.7. Nicht nur diese, auch viele andere Eigenschaften erben die Tensorproduktsplines von ihren univariaten Verwandten. Sie bilden zum Beispiel eine nichtnegative Teilung der Eins über ihrem Definitionsbereich. Daraus folgt, dass ein Tensorproduktspline stets in der konvexen Hülle des Kontrollpolygons liegt. Diese Eigenschaften werden aber im weiteren Verlauf nicht benötigt, weswegen auf eine Ausführung an dieser Stelle verzichtet wird. Ausführliche Informationen zu diesem Thema findet man unter anderem in [22] oder [29]. Abbildung 3.8 zeigt beispielhaft eine Tensorproduktsplinefläche zu den Basisfunktionen aus Abbildung 3.7.

3.2 Quaternionen

Quaternionen sind ein beliebtes Werkzeug in der Computeranimation, um Kamerabewegungen und Bewegungen von rotierenden Körpern zu simulieren oder in der Luft- und Raumfahrt zur Steuerung von Flugobjekten. Hier werden sie zur Berechnung von Rotationen herangezogen. Ein Vorteil von Quaternionen gegenüber den üblichen Rotationsmatrizen liegt darin, dass zur Berechnung von Drehungen weniger Rechenoperationen benötigt werden als mit einer Rotationsmatrix. Dies wirkt sich besonders dann aus, wenn viele Rotationen hintereinander durchgeführt werden.

Im folgenden Abschnitt werden zunächst die grundlegenden Definitionen und Eigenschaften der Quaternionen angegeben. Anschließend wird auf zwei Arten von Quaternionen eingegangen, die im weiteren Verlauf dieser Arbeit, speziell in Kapitel 5, von besonderem Interesse sind. Dies sind zum einen die *reinen* bzw. *imaginären Quaternion*, die eine Einbettung des $\mathbb{R}^3$ in den Raum der Quaternionen $\mathbb{H}$ repräsentieren, zum anderen die *Einheitsquaternionen*, die die Einheits–3–Sphäre $\mathbb{S}^3$ in $\mathbb{R}^4$ darstellen und die gewünschten Drehungen repräsentieren. Abschließend wird eine entsprechende Abbildung angegeben, mit dessen Hilfe die gewünschten Drehungen durchgeführt werden können.

3.2.1 Definition und Eigenschaften

Der Raum der Quaternionen $\mathbb{H}$ entsteht durch Adjunktion von Einheiten i, j und k zu den reellen Zahlen $\mathbb{R}$. Die Definition der Quaternionen ist denen der komplexen Zahlen $\mathbb{C}$ sehr ähnlich, sie unterscheiden sich aber in der Dimension des Imaginärteils.

Definition 3.34 *Eine* Quaternion $\boldsymbol{q}$ *ist definiert als Ausdruck der Form*

$$\boldsymbol{q} := q_0 + q_1 i + q_2 j + q_3 k, \qquad q_\ell \in \mathbb{R},\ \ell = 0, \ldots, 3, \tag{3.46}$$

wobei $\{1, i, j, k\}$ *eine Basis der* Quaternionen $\mathbb{H}$ *über* $\mathbb{R}$ *bildet.*

Für die Basiselemente $1, i, j$ und k gilt, dass sie die Länge eins haben und orthogonal zueinander sind. Für ihre Multiplikation gelten die so genannten *Hamilton–Regeln*

$$i^2 = j^2 = k^2 = ijk = -1. \tag{3.47}$$

Der *Realteil* einer Quaternion $\boldsymbol{q}$ wird mit

$$\mathrm{Re}(\boldsymbol{q}) := q_0$$

und der *Imaginärteil* mit

$$\mathrm{Im}(\boldsymbol{q}) = \boldsymbol{q}_\mathrm{I} := q_1 i + q_2 j + q_3 k$$

bezeichnet. Häufig wird alternativ zu (3.46) eine Vektorschreibweise für Quaternionen verwendet, wobei hier der Imaginärteil in der Regel zu einem dreikomponentigen Vektor $\boldsymbol{q}_\mathrm{I} := (q_1,\, q_2,\, q_3)^T$ zusammengefasst wird:

$$\boldsymbol{q} = \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \begin{pmatrix} q_0 \\ \boldsymbol{q}_\mathrm{I} \end{pmatrix}, \quad q_0 \in \mathbb{R},\ \boldsymbol{q}_\mathrm{I} \in \mathbb{R}^3.$$

Über $\mathbb{H}$ kann eine Addition $+$ und eine nicht kommutative Multiplikation $\cdot$ definiert werden, so dass $(\mathbb{H}, +, \cdot)$ ein *Schiefkörper* ist. Die Addition wird, wie in Vektorräumen üblich, komponentenweise durchgeführt. Die Multiplikation kann weitestgehend aus der Vektorraumarithmetik (Skalarprodukt) übernommen werden. Hier ist jedoch auf die speziellen Multiplikationsregeln (3.47) zu achten. Für die *Addition* zweier Quaternionen $\boldsymbol{q}$ und $\boldsymbol{p}$ gilt

$$\boldsymbol{q} + \boldsymbol{p} := (q_0 + p_0) + (q_1 + p_1) i + (q_2 + p_2) j + (q_3 + p_3) k$$

sowie für die *Multiplikation*

$$\boldsymbol{q}\boldsymbol{p} := \begin{pmatrix} q_0 p_0 - q_1 p_1 - q_2 p_2 - q_3 p_3 \\ q_0 p_1 + q_1 p_0 + q_2 p_3 - q_3 p_2 \\ q_0 p_2 - q_1 p_3 + q_2 p_0 + q_3 p_1 \\ q_0 p_3 + q_1 p_2 - q_2 p_1 - q_3 p_0 \end{pmatrix} = \begin{pmatrix} q_0 p_0 - \langle \boldsymbol{q}_\mathrm{I}, \boldsymbol{p}_\mathrm{I} \rangle \\ \boldsymbol{q}_\mathrm{I} \times \boldsymbol{p}_\mathrm{I} + q_0 \boldsymbol{p}_\mathrm{I} + p_0 \boldsymbol{q}_\mathrm{I} \end{pmatrix}. \tag{3.48}$$

Die Quaternion

$$\overline{\boldsymbol{q}} := (q_0, -\boldsymbol{q}_{\mathrm{I}})^T$$

wird *Konjugation* der Quaternion $\boldsymbol{q}$ genannt. Die *Norm* einer Quaternion $\boldsymbol{q}$ wird analog zu der Norm einer komplexen Zahl gebildet und mit

$$|\boldsymbol{q}|^2 = \boldsymbol{q}\overline{\boldsymbol{q}} = q_0^2 + q_1^2 + q_2^2 + q_3^2$$

angegeben.

3.2.2 Spezielle Quaternionen

Definition 3.35 *Eine Quaternion $\boldsymbol{q} \in \mathbb{H}$ wird als* reine (imaginäre) Quaternion *bezeichnet, falls ihr Realteil Null ist,* $\mathrm{Re}(\boldsymbol{q}) = 0$. *Die Menge der* reinen Quaternionen *wird mit $\mathbb{H}_{\mathrm{I}}$ bezeichnet.*

$$\mathbb{H}_{\mathrm{I}} := \{\boldsymbol{q} \in \mathbb{H} \mid \mathrm{Re}(\boldsymbol{q}) = 0\} \tag{3.49}$$

Der Vektorraum $\mathbb{R}^3$ wird durch die reinen Quaternionen in den Raum $\mathbb{H}$ eingebettet. Hierfür wird ein $\boldsymbol{x} \in \mathbb{R}^3$ zu

$$\widehat{\boldsymbol{x}} := \begin{pmatrix} 0 \\ \boldsymbol{x} \end{pmatrix} \in \mathbb{H} \tag{3.50}$$

ergänzt. Der Punkt $\boldsymbol{x}$ wird also als Imaginärteil einer reinen Quaternion interpretiert.

Definition 3.36 *Eine Quaternion $\boldsymbol{q} \in \mathbb{H}$ wird als* Einheitsquaternion *bezeichnet, falls ihr Betrag eins ist:*

$$|\boldsymbol{q}| = 1 \;\Leftrightarrow\; q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \;\Leftrightarrow\; \boldsymbol{q}\overline{\boldsymbol{q}} = 1 \;\Leftrightarrow\; \overline{\boldsymbol{q}} = \boldsymbol{q}^{-1}. \tag{3.51}$$

Die Einheitsquaternionen sind abgeschlossen bezüglich der Multiplikation im Sinne von (3.48), und der multiplikativen Inversen, dies geht insbesondere aus der letzten Gleichung von (3.51) hervor. Sie bilden die Einheits–3–Sphäre $\mathbb{S}^3$ im vierdimensionalen Vektorraum $\mathbb{H}$.

3.2.3 Drehungen

Identifiziert man alle Antipodalpunkte $\boldsymbol{q}$ und $-\boldsymbol{q}$ der $\mathbb{S}^3$ miteinander, wird hierdurch eine Äquivalenzrelation induziert. Die entstehenden Äquivalenzklassen können eindeutig mit Elementen der Drehgruppe $SO(3)$ identifiziert werden, siehe [12].

Für eine Einheitsquaternion $\boldsymbol{q} = (q_0, q_1, q_2, q_3)^T \in \mathbb{S}^3$ gilt $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. Sei nun

$$\theta = \cos^{-1}(q_0) \in [0, \pi] \qquad \text{sowie} \tag{3.52}$$

$$\boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \frac{1}{\sqrt{q_1^2 + q_2^2 + q_3^2}} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} \in \mathbb{S}^2, \tag{3.53}$$

dann kann die Einheitsquaternion $\boldsymbol{q}$ mit Hilfe der folgenden Umformungen der beiden Gleichungen (3.52) und (3.53) umgeschrieben werden. Aus (3.52) folgt direkt

$$q_0 = \cos(\theta)$$

und damit

$$\sin(\theta) = \sqrt{1 - \cos^2(\theta)} = \sqrt{1 - q_0^2} = \sqrt{q_1^2 + q_2^2 + q_3^2}$$

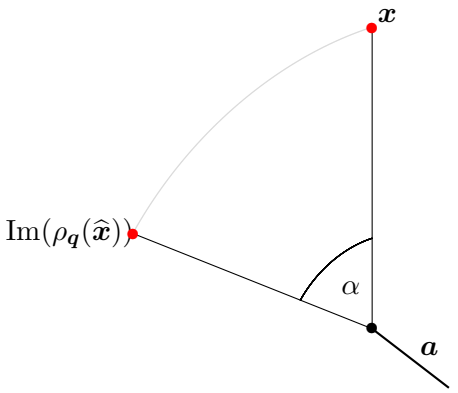


Abb. 3.9: Drehung des Punktes $\boldsymbol{x}$ um die Achse $\boldsymbol{a}$ mit Winkel α

sowie

$$\begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} = \sqrt{q_1^2 + q_2^2 + q_3^2} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \sin(\theta) \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}.$$

Eine Einheitsquaternion $\boldsymbol{q}$ lässt sich somit eindeutig in der Form

$$\boldsymbol{q} = \begin{pmatrix} \cos(\theta) \\ \sin(\theta)\boldsymbol{a} \end{pmatrix}, \qquad \boldsymbol{a} \in \mathbb{R}^3 \tag{3.54}$$

schreiben.

Die Drehung eines Punktes $\boldsymbol{x} \in \mathbb{R}^3$ wird mit Hilfe der Abbildung $\rho_{\boldsymbol{q}}$ durchgeführt. $\rho_{\boldsymbol{q}} : \mathbb{H}_\mathrm{I} \to \mathbb{H}_\mathrm{I}$ ist definiert als

$$\rho_{\boldsymbol{q}}(\widehat{\boldsymbol{x}}) := \boldsymbol{q}\widehat{\boldsymbol{x}}\overline{\boldsymbol{q}} \qquad \widehat{\boldsymbol{x}} = (0,\, \boldsymbol{x})^T. \tag{3.55}$$

Definition 3.37 *Die durch die Einheitsquaternion*

$$\boldsymbol{q} = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right)\boldsymbol{a} \end{pmatrix}$$

induzierte Drehung des Punktes $\boldsymbol{x} \in \mathbb{R}^3$ um die Achse $\boldsymbol{a} \in \mathbb{S}^2$ mit dem Winkel $\theta \in [0, \pi]$ auf den Punkt $\boldsymbol{x}' \in \mathbb{R}^3$ wird durch den Imaginärteil von $\rho_{\boldsymbol{q}}(\widehat{\boldsymbol{x}})$ definiert,

$$\boldsymbol{x}' := \mathrm{Im}(\rho_{\boldsymbol{q}}(\widehat{\boldsymbol{x}})).$$

Da alle Drehungen um eine Ursprungsachse durchgeführt werden, ist $\mathbf{0}$ somit ein Fixpunkt von $\rho_{\boldsymbol{q}}$. Die Hintereinanderausführung von Drehungen entspricht der Multiplikation der Quaternionen

$$\rho_{\boldsymbol{q}_1} \circ \rho_{\boldsymbol{q}_2} = \rho_{\boldsymbol{q}_1\boldsymbol{q}_2}.$$

Die Umkehrung der Drehrichtung entspricht der Drehung mit inverser Quaternion $\boldsymbol{q}^{-1}$

$$\rho_{\boldsymbol{q}}^{-1} = \rho_{\boldsymbol{q}^{-1}}.$$

Eine Visualisierung einer Drehung nach Definition 3.37 ist in Abbildung 3.9 zu sehen.

4 Analyse der Position

In diesem Kapitel wird ein Ansatz beschrieben, mit dessen Hilfe eine Korrektur eines Teileprogramms möglich ist, bei der die Oberflächenstruktur an den in Abschnitt 2.2 beschriebenen Problemstellen homogenisiert wird. Es sollen alle Fräsbahnen, die über einem entsprechenden geometrischen Merkmal liegen, so korrigiert werden, dass die Oberfläche des entstehenden Werkstücks an diesem Merkmal keine Unregelmäßigkeiten aufweist. Hierfür ist es zunächst notwendig die entsprechenden Korrekturstellen zu lokalisieren. In einer numerischen Steuerung stehen aktuell nur die Daten aus einem Teileprogramm, die G–Code Daten (meist Linear–Sätze), zur Verfügung. Diese Daten werden, in einem dem Fräsvorgang vorgelagerten Prozess, zu Polynomen bzw. Splinekurven (Bahnpolynome) weiterverarbeitet. Auf diese Splinekurven kann die in diesem Abschnitt vorgestellte Merkmalserkennung angewendet werden, um die für einen Fertigungsprozess kritischen Stellen zu identifizieren. An diesen Stellen werden Korrekturen der Bahn bzw. des Teileprogramms durchgeführt. Für die Berechnung der Korrekturpunkte werden zusätzliche Geometrieinformationen mittels einer Rückfrage an ein CAD–Programm zur Verfügung gestellt. Eine solche Rückfrage an die Geometrie bzw. die Bereitstellung der Geometriedaten in analytischer Form ist aktuell in einer numerischen Steuerung nicht vorgesehen. Diese Arbeit stellt somit einen neuen Ansatz zur Berechnung von Korrekturen durch Bereitstellung höherwertigerer Geometrieinformationen innerhalb einer numerischen Steuerung dar. Das hier vorgestellte Verfahren ist ein Beispiel dafür, dass eine Vernetzung von Steuerung und CAD–System sinnvoll ist.

In den ersten beiden Abschnitten dieses Kapitels werden die benötigten Verfahren der Merkmalserkennung mathematisch formuliert. Es wird zunächst ein einfaches Verfahren vorgestellt, mit dem die gegebenen Datenpunkte durch eine Splinekurve mit adaptiver Knotenfolge fein approximiert werden können. Anschließend wird ein allgemeiner Multiresolution Ansatz präsentiert, mit dessen Hilfe die Splinekurven sukzessive auf verschiedenen Auflösungsstufen betrachtet werden. Mit dieser Zerlegung kann ein *Analysespline* angegeben werden, mit dessen Hilfe die Merkmalspunkte identifiziert werden können. Sodann wird abschließend mittels einer Persistenz– und Dominanzanalyse eine Quantifizierung der Merkmalspunkte aus dem *Analysespline* angegeben. Im letzten Abschnitt wird das vorgestellte Verfahren an einem Beispiel demonstriert und abschließend exemplarisch gezeigt wie die Korrekturpunkte eines verbesserten Teileprogramms berechnet werden können.

4.1 Approximation G-Code

Bei der Approximation von Datenpunkten $P_j \in \mathbb{R}^d$, $j = 1, \ldots, k$, wird im Gegensatz zur Interpolation kein Wert mehr darauf gelegt, dass die Datenpunkte P_j exakt getroffen werden. Vielmehr sollen die Daten „bestmöglich“ approximiert werden. Unter „bestmöglich“ versteht man in der Regel einen Approximanten $s_m^* \boldsymbol{d}$, der ein gewünschtes Funktional minimiert,

$$s_m^* \boldsymbol{d} = \min_{s_m \boldsymbol{d} \,\in\, \mathbb{S}_m(T)} \sum_{j=1}^{n} D(s_m \boldsymbol{d}(t_j), P_j). \tag{4.1}$$

Die Parameterwerte t_j, die zu den Datenpunkten P_j korrespondieren sollen, sind nicht immer gegeben und müssen zunächst geeignet gewählt werden. Als guter Ansatz für eine Funktion D

haben sich Abstandsbegriffe wie

$$D(x, y) = \|x - y\|_2^2$$

erwiesen. Um die Gestalt einer Splinekurve und somit das Approximationsergebnis zu beeinflussen, kann man ganz offensichtlich die Koeffizienten des Splines variieren. Weniger einfach ist es das Approximationsergebnis durch die Knotenfolge bzw. die Knotenabstände $\Delta\tau_j := |\tau_{j+1} - \tau_j|$ sinnvoll zu verändern. Viele Approximationsverfahren verfolgen die Strategie einen Knotenvektor T vorzugeben und die Koeffizienten des gewünschten Splines durch Minimierung eines Funktionals wie (4.1) zu bestimmen. Beliebte Ansätze zur Wahl eines Knotenvektors sind

- *äquidistante* Knotenabstände: $\Delta\tau_j = \text{const}$,
- *chordale* Knotenabstände: $\Delta\tau_j \sim |P_{j+1} - P_j|$ oder
- *zentripetale* Knotenabstände: $\Delta\tau_j \sim \sqrt{|P_{j+1} - P_j|}$.

Wie bereits in Abschnitt 3.1.3 erwähnt, ist es eine gute Idee, die Knoten an den Stellen dicht zu legen, an denen man eine große zweite Ableitung bzw. Krümmung der zu approximierenden Daten erwartet [18, 43].

Eine weitere theoretische Möglichkeit bei der Approximation mittels Splines besteht darin die Knotenvektoren nicht festzulegen. Hier approximiert man nicht mit einem linearen Funktionenraum $\mathbb{S}_m(T)$, sondern mit allen Splineräumen

$$\mathbb{S}_m^N := \bigcup_{T_{N,m}} \mathbb{S}_m(T_{N,m}),$$

dessen Knotenvektoren zu einer vorgegebenen Anzahl N an Knoten das gewünschte Parameterintervall partitionieren. Dies führt allerdings zu komplexen nichtlinearen Problemen, siehe zum Beispiel [6, 16] oder [48].

In diesem Abschnitt soll hauptsächlich die Analyse der Daten aus einem Teileprogramm mittels einer Splinekurve vorgestellt werden. Diese Daten werden in der Fertigung genauer gesagt in einer numerischen Steuerung innerhalb einer gewissen Toleranz approximiert. Wie solche Approximanten erzeugt werden können, kann man beispielsweise in den Arbeiten von Bock [4] oder Qiao [43] nachlesen. Mit Hilfe des im Folgenden vorgestellten Verfahrens können diese Bahnpolynome bzw. Splines analysiert werden. Es wird zunächst beispielhaft gezeigt wie ein für die Analyse geeigneter Approximant erzeugt werden kann. Hierfür wird eine Überabtastung des Polygonzugs aus dem Teileprogramm mit Hilfe eines Schoenberg–Operators (Abschnitt 4.1.2) zu einer krümmungsadaptierten Knotenfolge (Abschnitt 4.1.1) approximiert.

4.1.1 Wahl des Knotenvektors

Motiviert durch den Beweis von Satz 3.22 sollen die Knoten im Parameterintervall anhand der zweiten Ableitung der zu approximierenden Funktion bzw. Datenpunkte verteilt werden. Ist die zu approximierende Funktion zudem annähernd nach der Bogenlänge parametriert, entspricht die zweite Ableitung der Funktion ungefähr ihrer Krümmung. Unter der Berücksichtigung, dass ein CAM–System ein Werkstück meist anhand eines vorgegebenen *Toleranzwertes* ε diskretisiert und dieser Fehler/*Sehnenfehler* ε für jeden G01–Satz ausgenutzt wird, kann man folgenden Zusammenhang zwischen einer Satzlänge ℓ und dem Sehnenfehler ε angeben:

$$r^2 = \left(\frac{\ell}{2}\right)^2 + (r - \varepsilon)^2 \qquad \Rightarrow \qquad \ell = \sqrt{8r\varepsilon - 4\varepsilon^2}.$$

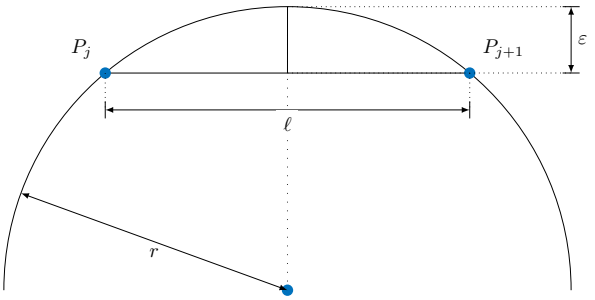


Abb. 4.1: Sehnenfehler zweier aufeinanderfolgende Punkte.

Abbildung 4.1 veranschaulicht diesen Zusammenhang. Der Radius r entspricht dem Reziprokwert der erwarteten Krümmung κ zu diesem Satz. Da im Allgemeinen der Sehnenfehler ε sehr klein gewählt wird und demnach wesentlich kleiner als die Satzlänge ℓ ist, gilt approximativ

$$\ell \approx \sqrt{8r\varepsilon} = \sqrt{8\varepsilon}\frac{1}{\sqrt{\kappa}}.$$

Somit ist die Satzlänge annäherungsweise antiproportional zur Wurzel aus der erwarteten Krümmung an dieser Stelle. Diese Tatsache ist ein weiteres grundlegendes Argument dafür, die Knoten des Splineapproximanten an den Stellen im Parameterbereich dicht zu legen, an denen eine hohe Krümmung erwartet wird. Ein guter Ansatz eine solche Verteilung zu erhalten, besteht darin, das Integral über der Wurzel aus der Krümmung der zu approximierenden Funktion bzw. Daten zwischen je zwei Knoten konstant zu halten. Für einen Knotenvektor $T_{n,m}$ würde mit einer solchen Knotenverteilung

$$\int_{\tau_j}^{\tau_{j+1}} \sqrt{\kappa(x)}\,\mathrm{d}\,x = C, \qquad C \in \mathbb{R}^+ \cup \{0\}, \quad j = 1, \ldots, n+m, \tag{4.2}$$

gelten, wobei κ der Krümmung der zu approximierenden Funktion bzw. Daten entspricht.

Sollen nun die Punkte $P_j \in \mathbb{R}^d$, $j = 1, \ldots, k$, approximiert werden, muss man für eine solche Knotenwahl einen Krümmungswert κ_j zu einem Punkt P_j schätzen.

Bestimmung der Krümmung

Um die diskrete Krümmung einer Folge von Datenpunkten zu schätzen, gibt es verschiedene Ansätze, siehe hierfür [9, 30]. In diesem Abschnitt wird zur Krümmungsschätzung die sehr einfache *Umkreismethode* vorgestellt. Diese Methode reagiert allerdings sehr stark auf verrauschte Daten, speziell, wenn der Abstand der Datenpunkte sehr klein ist. Liegt das erwartete Rauschen deutlich unterhalb des kleinsten Datenpunktabstands liefert diese Methode gute Werte. Als stabile Alternative sei auf eine *flächeninvariante Methode* von Pottmann, Wallner, Huang und Yang verwiesen [42]. Ein Vergleich verschiedener Krümmungsschätzer ist in [43] aufgeführt.

Zu einem Punkt P_j wird die Krümmung κ_j mit Hilfe der Krümmung des Kreises geschätzt, der die drei aufeinander folgenden Punkte P_{j-1}, P_j und P_{j+1} interpoliert. Die Krümmung κ_j zum Punkt P_j entspricht dem Reziprokwert des Radius r des interpolierenden Kreises. Somit kann für den ersten und den letzten Datenpunkt kein Krümmungswert bestimmt werden. Hier wird die Vereinbarung getroffen, dass eine Krümmung von Null vorliegt.

Definition 4.1 *Zu einem gegebenen Satz an Datenpunkten $P_j \in \mathbb{R}^d$, $j = 1, \ldots, k$, ist die zu einem Datenpunkt P_j korrespondierende* Krümmung κ_j *durch*

$$\kappa_j := \begin{cases} 0, & j = 1 \vee j = k, \\ 1/r_j, & j = 2, \ldots, k-1 \end{cases} \tag{4.3}$$

gegeben, wobei

$$r_j = \frac{\|P_{j-1} - P_{j+1}\|_2}{2\sin(\alpha_j)}$$

und

$$\alpha_j = \cos^{-1}\left(\frac{(P_{j+1} - P_j)^T (P_{j-1} - P_j)}{\|P_{j+1} - P_j\|_2 \, \|P_{j-1} - P_j\|_2}\right).$$

Falls $\alpha_j = 0$ oder $\alpha_j = \pi$ gilt die Vereinbarung $\kappa_j = 0$.

Verteilung der Knoten

Die approximierende Splinekurve soll annähernd bogenlängenparametrisiert sein. Das entsprechende Parameterintervall U, über dem die Knoten des Knotenvektors $T_{n,m}$ verteilt werden sollen, wird von Null bis zur Gesamtlänge ℓ des Polygonzugs der zu approximierenden Daten gewählt. Für ℓ gilt

$$\ell = \sum_{j=1}^{k-1} \|P_{j+1} - P_j\|_2 \,. \tag{4.4}$$

Das Parameterintervall entspricht demnach $U = [0, \ell]$. Somit kann jeder Datenpunkt P_s mit seiner entsprechenden Polygonzuglänge mit einem Wert

$$u_s := \sum_{j=1}^{s-1} \|P_{j+1} - P_j\|_2 \,, \qquad s = 1, \ldots, k, \tag{4.5}$$

im Parameterintervall U identifiziert werden.

Wie einführend erwähnt, soll eine Knotenverteilung im Knotenvektor $T_{n,m}$ erreicht werden, so dass (4.2) gilt. Hierfür wird die Funktion f mit

$$f(t) = \int_0^t \sqrt{\kappa(u)}\, du, \qquad t, u \in U,$$

betrachtet. f ist eine monoton wachsende Funktion, da $\kappa(u) \geq 0$. Zu jedem Datenpunkt P_j liegt ein geschätzter Krümmungswert κ_j vor. Somit kann die Funktion f an den Stellen u_j approximativ mittels Quadratur bestimmt werden. Hierfür sei

$$f(u_j) \approx \frac{1}{2} \sum_{h=1}^{j-1} (u_{h+1} - u_h) \left(\sqrt{\kappa(u_h)} + \sqrt{\kappa(u_{h+1})}\right), \qquad \kappa(u_j) := \kappa_j. \tag{4.6}$$

Bei dieser Quadratur, der sogenannten Trapezregel, wird zwischen den bekannten Funktionswerten von $\sqrt{\kappa}$ linear interpoliert. Für die Berechnung der Umkehrfunktion f^{-1} an einer Stelle $t \in [u_j, u_{j+1}]$ folgt somit der Ansatz

$$f^{-1}(t) \approx u_j + \left(\frac{f(t) - f(u_j)}{f(u_{j+1}) - f(u_j)}\right)(u_{j+1} - u_j) \,. \tag{4.7}$$

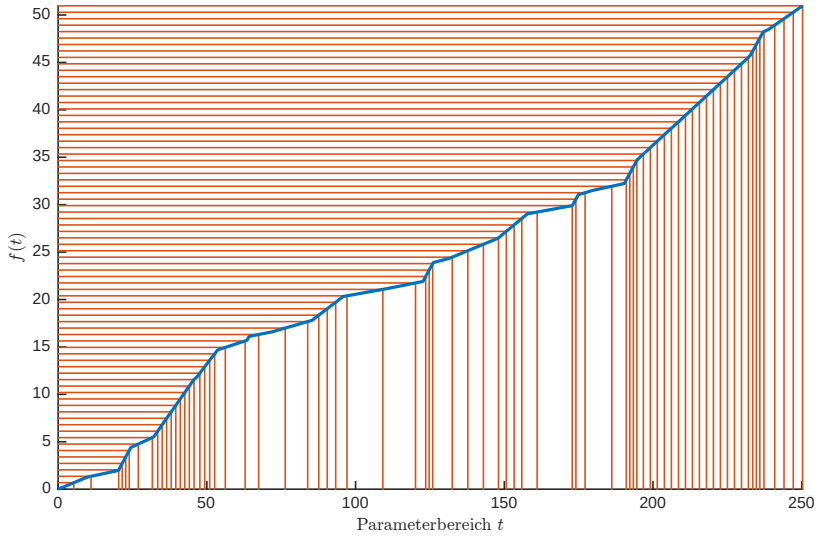


Abb. 4.2: Knotenverteilung zum Testprogramm aus Abbildung 4.3.

Für eine vorgegebene Anzahl $n+m+1$ an Knoten zum Knotenvektor $T_{n,m}$ wird die Konstante C aus (4.2) durch

$$C = \frac{f(\ell)}{n-m}$$

bestimmt. Da mit $(m+1)$–fachen Randknoten gearbeitet werden soll, gilt $\tau_1 = \cdots = \tau_{m+1} = 0$ sowie $\tau_{n+1} = \cdots = \tau_{n+m+1} = \ell$. Für die inneren Knoten $\tau_{m+2}, \ldots, \tau_n$ von $T_{n,m}$ gilt:

$$f(\tau_j) = (j-m-1)C, \qquad j = m+2, \ldots, n.$$

Somit ist (4.2) für je zwei aufeinander folgende Knoten τ_j und τ_{j+1} erfüllt, da

$$f(\tau_{j+1}) - f(\tau_j) = ((j+1)-m-1)C - (j-m-1)C = C.$$

Die inneren Knoten τ_j, $j = m+2, \ldots, n$, können unter Verwendung von (4.7) durch

$$\tau_j = f^{-1}(f(\tau_j))$$

bestimmt werden. Abbildung 4.2 ist eine schematische Darstellung dieses Knotenverteilprozesses. Anschaulich wird die y–Achse äquidistant aufgeteilt und die entsprechenden Werte auf der x–Achse gesucht.

4.1.2 Schoenberg–Approximant

Die durch ein Teileprogramm gegebenen Punkte P_j, $j = 1, \ldots, k$, sollen mittels eines Schoenberg–Operators approximiert werden. Hierfür wird eine krümmungsadaptierte Knotenfolge $T_{n,m}$ nach Abschnitt 4.1.1 gewählt. Die angegebene Knotenfolge ist nach der Bogenlänge der Datenpunkte durch lineare Interpolation bestimmt worden. Demnach können die entsprechenden Greville–Abszissen $\hat{\boldsymbol{t}} = (\hat{t}_1, \ldots, \hat{t}_n)$ zu $T_{n,m}$ mit Positionen auf dem Polygonzug der Datenpunkte P_j identifiziert werden. Eine Greville–Abszisse $\hat{t}_s$ mit

$$\sum_{j=1}^{r-1} \|P_{j+1} - P_j\|_2 \leq t_s \leq \sum_{j=1}^{r} \|P_{j+1} - P_j\|_2 ,$$

die mit ihrem Wert einer Bogenlänge auf dem Polygonzug zwischen den Punkten P_r und P_{r+1} entspricht, wird dem Punkt

$$p(\hat{t}_s) := P_r + \frac{\hat{t}_s - \sum_{j=1}^{r-1} \|P_{j+1} - P_j\|_2}{\|P_{r+1} - P_r\|_2} (P_{r+1} - P_r)$$

zugeordnet. Man erhält eine Abtastung des Polygonzuges in Abhängigkeit von der Anzahl an Knoten im Knotenvektor $T_{n,m}$. Diese Werte bilden nach Definition 3.20 des Schoenberg–Operators die Koeffizienten des Splineapproximanten der Datenpunkte P_j. Für die Koeffizienten $\boldsymbol{d} = (d_1, \ldots, d_n)$ eines Approximanten gilt demnach $d_j = p(\hat{t}_j)$ und somit für den Schoenberg–Spline–Approximanten

$$s_m \boldsymbol{d}(x) = \sum_{j=1}^{n} p(\hat{t}_j) N_j^m (x \mid T).$$

Die Qualität der Approximation des Polygonzuges kann leicht durch eine höhere Anzahl an Knoten und somit an Abtastungen verbessert werden. In der Praxis hat sich $n = \lceil \log_2(k) \rceil$ als guter Wert für die Anzahl an Abtastungen etabliert, wobei k der Anzahl der zu approximierenden Datenpunkte entspricht.

Bemerkung 4.2 *Der Schoenberg–Operator ist ein sogenannter „shape preserving", also Gestalt erhaltender Operator. Zudem besitzt er nach Satz 3.21 lineare Exaktheit und kann die überabgetasteten Polygonzugabschnitte exakt reproduzieren. Er lässt sich äußerst leicht berechnen und wird dadurch zur ersten Wahl für einen solchen Approximanten.*

Der so erzeugte Approximant ist nur ein Vorschlag. Für die folgende Merkmalserkennung kann jede beliebige Art der Approximation von Teileprogrammen zur Analyse herangezogen werden.

4.2 Merkmalserkennung

Ein *Merkmal* oder *Merkmalspunkt* (engl.: *feature*) ist ein Punkt von besonderem Interesse. Es erfüllt eine Reihe von Kriterien, die jeweils davon abhängig sind, was gerade untersucht bzw. gesucht wird. Eine allgemeingültige Definition eines Merkmals ist aufgrund der unterschiedlichen Anwendungsfälle kaum möglich. Merkmalspunkte aus Sicht der Fertigung wären zum Beispiel die schon in Abschnitt 2 erwähnten Ungenauigkeiten im Fräsbild. Diese entstehen meist an Flächenübergängen. Diese problematischen Stellen zeichnen sich meistens durch unstetige Ableitungen am Flächenübergang aus. Ein Merkmal bezeichnet demnach eine Stelle in der Fräsbahn, welche einen unstetigen Flächenübergang reproduzieren soll, wobei hier nicht vorgegeben ist, ab welcher Ableitung die Unstetigkeit auftritt.

4.2.1 Allgemeiner Multiresolution Ansatz

Unter dem Begriff *Multiresolution Analysis* (MRA) versteht man die Zerlegung einer gegebenen Geometrie oder eines Signals in eine gröbere Approximation (globale Form) und einen Detailanteil. Diese Zerlegung wird in der Regel für den approximativen Teil in immer gröber werdenden Auflösungsstufen wiederholt. Der Detailanteil beschreibt dabei stets den Anteil der Geometrie oder des Signals, welcher nicht mehr in der gröberen Auflösung dargestellt werden kann. Für Splinekurven versteht man unter dem Begriff *Auflösung* die Feinheit des Kontrollpolygons und somit indirekt die Dimension des Splineraums. Um die verschiedenen Auflösungsstufen über die

Notation zu kennzeichnen, wird im Folgenden die Auflösungsstufe als Superskript von Koeffizienten bzw. Koeffizientenmatrizen sowie Knoten und Knotenvektoren notiert. Diese Notation ist insbesondere nicht als Potenz zu verstehen. Im Standardfall einer MRA, siehe [8, 15, 37], wird der Funktionenraum V_0 durch die Ganzzahltranslate einer Funktion ϕ aufgespannt,

$$V_0 = \operatorname{span}\{\phi(\cdot - i) : i \in \mathbb{Z}\}.$$

Zusätzlich besteht für diese Funktionen eine *Skalenbeziehung*, was bedeutet, dass jedes ϕ durch eine Linearkombination von einer skalierten Version von sich selbst dargestellt werden kann. ϕ wird aus diesem Grund auch *Skalierungsfunktion* genannt. Es existiert also eine Folge a_i, so dass

$$\phi = \sum_i a_i \phi(2 \cdot - i).$$

Aus dieser Skalenbeziehung folgt $V_0 \subset V_1 = \operatorname{span}\{\phi(2 \cdot - i) : i \in \mathbb{Z}\}$. Für eine Funktion $f \in V_0$ findet man eine Koeffizientenfolge d_j^1, so dass

$$f(\,\cdot\,) = \sum_i d_i^0 \, \phi(\cdot - i) = \sum_i d_i^1 \, \phi(2 \cdot - i).$$

Das Konzept der MRA sieht vor, dass man die Basisfunktionen $\phi(\cdot - i)$ von V_0 mit Basisfunktionen $\psi(\cdot - i)$ so erweitert, dass

$$V_1 = \operatorname{span}\{\phi(\cdot - i) : i \in \mathbb{Z}\} \oplus \operatorname{span}\{\psi(\cdot - i) : i \in \mathbb{Z}\} = V_0 \oplus W_0$$

in eine direkte Summe aus V_0 und W_0 zerlegt werden kann. Die Basisfunktion $\psi \in W_0$ wird *Wavelet* und der Raum W_0 entsprechend *Waveletraum* genannt. Eine Funktion $f \in V_1$ kann somit als Summe von Linearkombinationen zur Basis von V_0 und W_0 dargestellt werden,

$$f(\,\cdot\,) = \sum_i d_i^0 \, \phi(\cdot - i) \; + \; \sum_i w_i^0 \, \psi(\cdot - i).$$

Diese Zerlegung von f in V_0 und W_0 entspricht der zu Beginn beschriebenen Zerlegung. Dabei repräsentiert der Teil in V_0 die grobe Approximation und der Teil in W_0 den Detailanteil der Funktion f.

Um eine solche Zerlegung für Splinefunktionen zu erhalten, benötigt man zunächst eine verschachtelte Folge von Splineräumen. Diese erhält man nach Satz 3.26 durch eine verschachtelte Folge von Knotenvektoren. Ein Musterbeispiel für die oben beschriebenen Skalierungsfunktionen sind die kardinalen B–Splines [46]. Dabei handelt es sich um B–Splines zu einer äquidistanten, unendlichen Knotenfolge $T^j = 2^j \mathbb{Z}$, $j \in \mathbb{N}_0$. Der in diesem Abschnitt vorgestellte Multiresolution Ansatz basiert jedoch auf *beliebigen Knotenfolgen* insbesondere krümmungsoptimierten Knotenfolgen. Die durch diese Knotenfolgen erzeugten Räume erfüllen weder die Bedingungen der Translationsinvarianz noch der Skalenbeziehung. Es wird lediglich die Forderung der verschachtelten Folge von Funktionenräumen erfüllt.

Um nun eine solche Folge von Splineräumen zu erhalten, sei $T^0_{n_0,m} \subseteq T^1_{n_1,m} \subseteq \dots$ mit

$$T^j = T^j_{n_j,m} = (\tau^j_k \;:\; k = 1, \dots, n_j + m + 1)$$

eine Folge von verschachtelten Knotenvektoren und

$$V_j := \mathbb{S}_m(T^j) \qquad \text{mit} \quad V_0 \subseteq \dots \subseteq V_j \subseteq V_{j+1} \subseteq \dots$$

die entsprechende Folge an Splineräumen. Die korrespondierenden B–Splines werden zur Vereinfachung der Notation als Spaltenvektor arrangiert:

$$\boldsymbol{N}^m(\,\cdot \mid T^j) := \left(N_k^m(\,\cdot \mid T^j) \;:\; k = 1, \ldots, n_j\right).$$

Um eine Splinekurve $s_m\boldsymbol{d} \in V_j$ in einem Raum mit feinerer Auflösung (V_k, $k > j$) darstellen zu können, fügt man die entsprechenden Knoten in die Knotenfolge T^j ein. Um sie in einem Raum mit geringerer Auflösung (V_k, $k < j$) darzustellen, müssen entsprechende Knoten entfernt werden. Hierfür werden die in Abschnitt 3.1.5 eingeführten Knoteneinfüge– bzw. Knotenentfernmatrizen $\boldsymbol{A}$ und $\boldsymbol{B}$ verwendet. Mit Hilfe dieser Matrizen können die *Waveleträume* für die gewünschte Zerlegung angegeben werden.

Lemma 4.3 *Für $j \in \mathbb{N}_0$ und einer Folge von Splineräumen $V_0 \subseteq V_1 \subseteq \ldots$ kann $V_{j+1} = \mathbb{S}_m(T^{j+1})$ in eine direkte Summe zerlegt werden. Es gilt*

$$V_{j+1} = V_j \oplus W_j, \qquad W_j = \operatorname{span}\left\{\left(\boldsymbol{I} - \boldsymbol{B}(T^j, T^{j+1})\boldsymbol{A}(T^j, T^{j+1})\right)\boldsymbol{N}^m(\,\cdot \mid T^{j+1})\right\}. \tag{4.8}$$

Beweis: Seien $s_m\boldsymbol{d}^{j+1}(\,\cdot \mid T^{j+1}) = \boldsymbol{d}^{j+1}\boldsymbol{N}^m(\,\cdot \mid T^{j+1}) \in V_{j+1}$ und $\boldsymbol{d}^j := \boldsymbol{d}^{j+1}\boldsymbol{B}(T^j, T^{j+1})$. Als abkürzende Schreibweise wird $\boldsymbol{A} = \boldsymbol{A}(T^j, T^{j+1})$ bzw. $\boldsymbol{B} = \boldsymbol{B}(T^j, T^{j+1})$ verwendet. Dann gilt

$$\begin{aligned}\boldsymbol{d}^{j+1} &= \boldsymbol{d}^j\boldsymbol{A} + \left(\boldsymbol{d}^{j+1} - \boldsymbol{d}^j\boldsymbol{A}\right) = \boldsymbol{d}^j\boldsymbol{A} + \left(\boldsymbol{d}^{j+1} - \boldsymbol{d}^{j+1}\boldsymbol{B}\boldsymbol{A}\right)\\ &= \boldsymbol{d}^j\boldsymbol{A} + \boldsymbol{d}^{j+1}\left(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A}\right).\end{aligned}$$

Mit dieser Zerlegung von $\boldsymbol{d}^{j+1}$ kann der Spline $s_m\boldsymbol{d}^{j+1}$ wie folgt notiert werden

$$\begin{aligned}s_m\boldsymbol{d}^{j+1}(\,\cdot \mid T^{j+1}) &= \left(\boldsymbol{d}^j\boldsymbol{A} + \boldsymbol{d}^{j+1}\left(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A}\right)\right)\boldsymbol{N}^m(\,\cdot \mid T^{j+1})\\ &= \boldsymbol{d}^j\boldsymbol{A}\boldsymbol{N}^m(\,\cdot \mid T^{j+1}) + \boldsymbol{d}^{j+1}\left(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A}\right)\boldsymbol{N}^m(\,\cdot \mid T^{j+1})\\ &= \boldsymbol{d}^j\boldsymbol{N}^m(\,\cdot \mid T^j) + \boldsymbol{d}^{j+1}\left(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A}\right)\boldsymbol{N}^m(\,\cdot \mid T^{j+1}).\end{aligned} \tag{4.9}$$

Somit beschreibt der erste Summand ein Element aus V_j und der zweite Summand ein Element aus W_j. Falls $s_m\boldsymbol{d}^{j+1} \in V_{j+1} \cap V_j$, gilt $\boldsymbol{d}^{j+1} = \boldsymbol{d}^j\boldsymbol{A}$ und somit für den Waveletanteil aus (4.9)

$$\boldsymbol{d}^{j+1}(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A}) = \boldsymbol{d}^j\boldsymbol{A}(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A}) = \boldsymbol{d}^j\boldsymbol{A} - \boldsymbol{d}^j\boldsymbol{A}\boldsymbol{B}\boldsymbol{A} = 0.$$

□

Für die Dimension des Waveletraums W_j gilt:

$$\dim W_j = \dim V_{j+1} - \dim V_j = n_{j+1} - n_j.$$

Somit folgt, dass

$$\boldsymbol{M}^m = \boldsymbol{M}^m(\,\cdot \mid T^{j+1}) := \left(\boldsymbol{I} - \boldsymbol{B}(T^j, T^{j+1})\boldsymbol{A}(T^j, T^{j+1})\right)\boldsymbol{N}^m(\,\cdot \mid T^{j+1})$$

nur ein Erzeugendensystem und keine Basis für W_j ist. Da $\boldsymbol{N}^m(\,\cdot \mid T^{j+1})$ linear unabhängig ist, gilt $\boldsymbol{c}\boldsymbol{M}^m = 0$ genau dann, wenn $\boldsymbol{c} \in \ker(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A})$ für ein $\boldsymbol{c} \neq \boldsymbol{0}$ ist. Somit lassen sich die linearen Abhängigkeiten in $\boldsymbol{M}^m$ leicht durch eine Singulär–Wert–Zerlegung von $(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A})$ eliminieren. Im Kern von $(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A})$ liegen alle Koeffizienten der Splines aus $V_{j+1} \cap V_j = V_j$. Demnach gilt für die Dimension des Kerns

$$\dim\ker(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A}) = \dim V_j.$$

Für die entsprechende Singulär–Wert–Zerlegung gilt somit $\boldsymbol{I} - \boldsymbol{BA} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ mit $\sigma_{11} \geq \cdots \geq \sigma_{n_{j+1}-n_j, n_{j+1}-n_j} > 0 = \sigma_{n_{j+1}-n_j+1, n_{j+1}-n_j+1} = \cdots = \sigma_{n_{j+1}, n_{j+1}}$ als Diagonaleinträge von $\boldsymbol{\Sigma}$. Eine Basis von W_j lässt sich nun durch

$$\boldsymbol{W}^m(\,\cdot \mid T^{j+1}) := \sum_{j=1}^{n_{j+1}-n_j} \sigma_{jj} \boldsymbol{e}_j^T \boldsymbol{V}^T \boldsymbol{N}^m(\,\cdot \mid T^{j+1}) \tag{4.10}$$

angeben, wobei $\boldsymbol{e}_j$ dem j–ten Einheitsvektor entspricht.

Da die „Wavelets“ $\boldsymbol{W}^m(\,\cdot \mid T^{j+1})$ nicht zwingend einen lokalisierten Träger haben, ist es trotz Überrepresentation wesentlich effizienter die Detailanteile durch die B–Spline Basis $\boldsymbol{N}^m(\,\cdot \mid T^{j+1})$ darzustellen.

Korollar 4.4 *Für jede Splinekurve $s_m\boldsymbol{d}^n \in V_N$ zu einer gegebenen Folge V_j, $j = 0, \ldots, N$, von Splineräumen mit $V_0 \subseteq V_1 \subseteq \cdots \subseteq V_N$ existiert eine hierarchische Zerlegung*

$$s_m\boldsymbol{d}^n(\,\cdot \mid T^N) = \boldsymbol{d}^0 \boldsymbol{N}^m(\,\cdot \mid T^0) + \sum_{k=0}^{N-1} \boldsymbol{w}^k \boldsymbol{N}^m(\,\cdot \mid T^{k+1}). \tag{4.11}$$

Eine Darstellung der Zerlegung (4.11) zu einer gegebenen Splinekurve sieht man in Abbildung 4.3. Die entsprechenden Koeffizienten einer Zerlegung von $s_m\boldsymbol{d}^{j+1} \in V_{j+1}$ in

$$s_m\boldsymbol{d}^{j+1}(\,\cdot \mid T^{j+1}) = \boldsymbol{d}^j \boldsymbol{N}^m(\,\cdot \mid T^j) + \boldsymbol{w}^j \boldsymbol{N}(\,\cdot \mid T^{j+1}) \tag{4.12}$$

erhält man durch

$$\boldsymbol{d}^j = \boldsymbol{d}^{j+1} \boldsymbol{B}(T^j, T^{j+1}), \qquad \boldsymbol{w}^j = \boldsymbol{d}^{j+1} \left(\boldsymbol{I} - \boldsymbol{B}(T^j, T^{j+1}) \boldsymbol{A}(T^j, T^{j+1}) \right).$$

Der approximative Teil $\boldsymbol{d}^j \boldsymbol{N}^m(\,\cdot \mid T^j)$ von $s_m\boldsymbol{d}^{j+1}(\,\cdot \mid T^{j+1})$ aus (4.12) wird sukzessive weiter nach diesem Schema zerlegt. Bei einem solchen Zerlegungsschritt spricht man von einer *Zerlegungsstufe* bzw. *Zerlegungslevel*. So erhält man alle Koeffizienten aus (4.11).

In der Praxis werden Zerlegungen wie (4.11) für Kompression oder Entrauschen von Daten verwendet. Hierfür werden Waveletkoeffizienten w_k^j gelöscht bzw. gleich Null gesetzt, falls $\|w_k^j\|_2 < \varepsilon$ für einen Schwellenwert $\varepsilon \in \mathbb{R}$.

Die Merkmalserkennung erfolgt ähnlich. Man sucht alle Waveletkoeffizienten, die betragsmäßig signifikant größer sind als die übrigen Koeffizienten. Diese repräsentieren ein Detail der Kurve, welches mit einer geringeren Auflösung nicht mehr darzustellen ist. Eine schnelle, diskrete Lokalisierung des Merkmals bzw. der Ecke erhält man durch eine einfache Identifizierung des entsprechenden Waveletkoeffizienten mit seiner Greville–Abszisse. Diese Identifizierung ist allerdings sehr ungenau. Die Betrachtung der *Spitzen* (*Peaks*) der skalaren Splinefunktion

$$s_m\boldsymbol{c}^j(\,\cdot \mid T^{j+1}) := \boldsymbol{c}^j \boldsymbol{N}^m(\,\cdot \mid T^{j+1}) = \sum_{k=1}^{n_j} \|w_k^j\|_2 \, N_j^m(\,\cdot \mid T^{j+1}) \tag{4.13}$$

mit $\boldsymbol{c}^j := \left(\|w_k^j\|_2 : k = 1, \ldots, n_j - 1 \right)$ ermöglicht eine wesentlich genauere Lokalisierung der geometrischen Merkmale, siehe Abbildung 4.3. Die Koeffizienten c_k^j entsprechen dem Betrag der

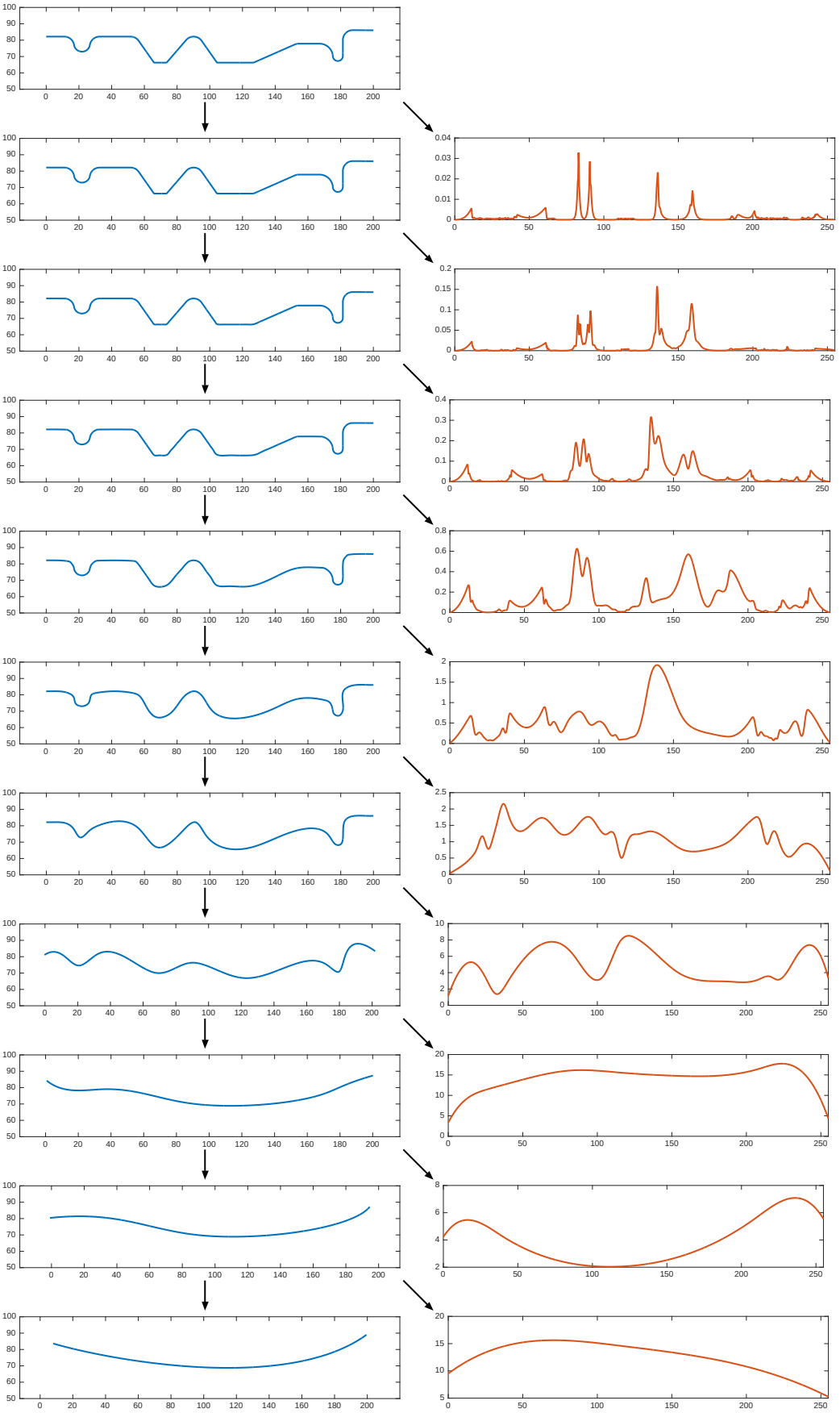


Abb. 4.3: Zerlegung eines Splines (Testprogramm) nach (4.11) in Approximation (blau) und Detailanteil (orange). Die Detailanteile sind als Betrag gegen den Parameterbereich geplottet: $t \mapsto \sum_{k=1}^{n_j} \left\| w_k^j \right\| N_k^m \left(t \mid T^{j+1} \right)$ (ohne Gewichtung $\alpha_{j,k}$).

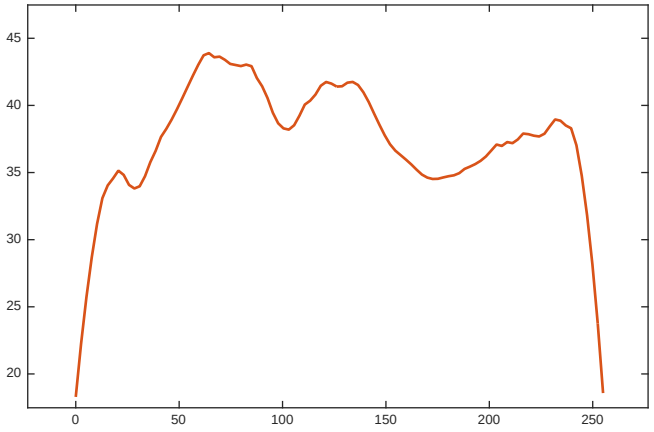


Abb. 4.4: Summation der Detailanteile $s_m \boldsymbol{c}$ ohne Gewichtung nach (4.14).

Waveletkoeffizienten w_k^j von $s_m \boldsymbol{d}^{j+1}$ aus einem Zerlegungsschritt (4.12). Je länger ein geometrisches Merkmal in den verschiedenen Zerlegungsstufen im Detailanteil auftaucht, desto markanter ist es. Somit ist eine Quantifizierung der Merkmalsausprägung möglich. Um einen Vergleich aller geometrischen Merkmale zu erhalten, ist ein erster Ansatz dadurch gegeben, dass die Detailanteile aller Zerlegungsstufen in Summe betrachtet werden. Eine solche Splinefunktion wird im Folgenden *Analysespline* genannt und ist durch

$$s_m \boldsymbol{c}(\,\cdot \mid T^N) := \boldsymbol{c}\boldsymbol{N}^m(\,\cdot \mid T^N) = \sum_{j=0}^{N-1} s_m \boldsymbol{c}^j(\,\cdot \mid T^{j+1}) = \sum_{j=0}^{N-1} \sum_{k=1}^{n_j} \|w_k^j\|_2 \, N_k^m(\,\cdot \mid T^{j+1}) \tag{4.14}$$

definiert. Die Koeffizienten $\boldsymbol{c} = (c_j \mid j = 1, \ldots, N)$ können durch wiederholtes Knoteneinfügen mit Hilfe der Waveletkoeffizienten $\boldsymbol{w}^j$ aus den einzelnen Zerlegungsstufen berechnet werden,

$$c_\ell = \sum_{j=0}^{N-1} \left\| \left(\boldsymbol{w}^j \prod_{k=j}^{N-1} \boldsymbol{A}(T^k, T^{k+1}) \right) \boldsymbol{e}_\ell \right\|_2 .$$

Hierbei entspricht $\boldsymbol{e}_\ell$ dem ℓ–ten Einheitsvektor in $\mathbb{R}^n$. Versucht man anhand der Gestalt von $s_m \boldsymbol{c}$ Merkmalspunkte der Splinekurve $s_m \boldsymbol{d}^n$ zu identifizieren, sieht man, dass viele Details aus den einzelnen Zerlegungsstufen bzw. den einzelnen Detailsplines $s_m \boldsymbol{c}^j$, $j = 1, \ldots, n-1$, verloren gehen. Hierfür können die Detailsplines in Abbildung 4.3 bzw. die Summation nach (4.14) in Abbildung 4.4 verglichen werden. Der Grund hierfür ist, dass die Waveletkoeffizienten einer gewissen natürlichen Abklingrate unterliegen. Diese liegt in der allgemeinen Wavelettheorie [8, 15, 37] bzw. im Fall von gleichverteilten Knotenfolgen (kardinalen Splines) bei einem Faktor von 2^{-j}. Der Exponent j entspricht in der Regel der Zerlegungsstufe bzw. ist von ihr abhängig. Im Fall von nichtäquidistanten Knotenvektoren wird aus einem solchen globalen Abklingfaktor ein lokaler Abklingfaktor für die jeweilige Zerlegungsstufe. Die Formel (4.14) des Analysespline $s_m \boldsymbol{c}$ muss demnach um einen Abklingfaktor $\alpha_{j,k} \geq 0$ ergänzt werden,

$$s_m \boldsymbol{c}(\,\cdot \mid T^n) := \sum_{j=0}^{n-1} \sum_{k=1}^{n_j} \alpha_{j,k} \, \|w_k^j\|_2 \, N_k^m(\,\cdot \mid T^{j+1}). \tag{4.15}$$

4.2.2 Abklingrate der Waveletkoeffizienten

In diesem Abschnitt wird eine Abklingrate der Waveletkoeffizienten und somit der Abklingfaktor α des Analysesplines (4.15) hergeleitet. Die Abklingrate aus [27] konnte nochmals verbessert

werden. Im Folgenden wird diese verbesserte Abschätzung hergeleitet. Um die Waveletkoeffizienten eines Splines abzuschätzen, betrachtet man sich seine Taylorentwicklung. Die Splineräume können den polynomialen Teil der Entwicklung exakt reproduzieren. Somit kann der Anteil eines Splines, der in einem gröberen Splineraum nicht mehr dargestellt werden kann, nur in dem Restglied der Taylorentwicklung enthalten sein. Das Restglied der Entwicklung wird mit Hilfe der de Boor–Fix Funktionale im feineren Splineraum dargestellt und abgeschätzt.

Im folgenden Lemma wird eine Abschätzung des Taylorrestglieds einer Taylorentwicklung vom Grad $m-2$ angegeben.

Lemma 4.5 *Für einen Spline $s_m\boldsymbol{d} = \boldsymbol{d}\boldsymbol{N}^m(\cdot \mid T) \in \mathbb{S}_m(T)$ zu einer Knotenfolge $T_{n,m}$ mit einfachen inneren Knoten bezeichnet $T_{m-2}s_m\boldsymbol{d}$ das Taylorpolynom vom Grad $m-2$. Es gilt:*

$$\|s_m\boldsymbol{d} - T_{m-2}s_m\boldsymbol{d}\|_{\infty,[\tau_{\ell-m},\tau_{\ell+m+1}]} \leq \max_{k=0,\dots,m} \frac{|\tau_{\ell-m+k} - \tau_{\ell+1+k}|^m}{m!} \left\|s_m^{(m-1)}\boldsymbol{d}\right\|_{\infty,[\tau_{\ell-m},\tau_{\ell+m+1}]}. \tag{4.16}$$

Beweis: Sei

$$I_\ell = \bigcup_{j=\ell-m}^{\ell} \operatorname{supp}\left\{N_j^m(\cdot \mid T)\right\} = [\tau_{\ell-m}, \tau_{\ell+m+1}]$$

das Intervall, das gerade der Vereinigung aller Träger derjenigen B–Splines entspricht, die zur Auswertung von $s_m\boldsymbol{d}$ an $t \in [\tau_\ell, \tau_{\ell+1}]$ beitragen.

Eine Taylorentwicklung von $s_m\boldsymbol{d}$ an der Stelle $t^* := \frac{1}{2}(\tau_{\ell+m+1} + \tau_{\ell-m})$ kann unter Verwendung der Restgliedformel von Lagrange mit einem $\xi \in [t^*, t]$ folgendermaßen dargestellt werden

$$s_m\boldsymbol{d}(t \mid T) = \sum_{k=0}^{m-2} \frac{s_m^{(k)}\boldsymbol{d}(t^* \mid T)}{k!}(t-t^*)^k + \max_{\xi\in I_\ell} \frac{|s_m^{(m-1)}\boldsymbol{d}(\xi \mid T)|}{(m-1)!} \max_{t\in I_\ell} |t-t^*|. \tag{4.17}$$

Der Abstand des Splines $s_m\boldsymbol{d}$ zu seinem Taylorpolynom $T_{m-2}s_m\boldsymbol{d}$ kann demnach wie folgt abgeschätzt werden

$$\begin{aligned}\left|s_m\boldsymbol{d}(t \mid T) - T_{m-2}s_m\boldsymbol{d}(t \mid T)\right| &\leq \max_{\xi\in I_\ell} \frac{|s_m^{(m-1)}\boldsymbol{d}(\xi \mid T)|}{(m-1)!} \max_{t\in I_\ell} |t-t^*| \\ &\leq \frac{1}{(m-1)!}\left(\frac{\tau_{\ell+m+1} - \tau_{\ell-m}}{2}\right)^{m-1} \left\|s_m^{(m-1)}\right\|_{\infty,I_\ell}.\end{aligned} \tag{4.18}$$

Für das Intervall I_ℓ gilt entweder $\tau_{\ell+1} - \tau_{\ell-m} \geq \frac{1}{2}(\tau_{\ell+m+1} - \tau_{\ell-m})$ oder $\tau_{\ell+m+1} - \tau_\ell \geq \frac{1}{2}(\tau_{\ell+m+1} - \tau_{\ell-m})$. Somit kann (4.18) weiter mit

$$\max_{k=0,\dots,m} |\tau_{\ell-m+k} - \tau_{\ell+1+k}| \geq \frac{1}{2}(\tau_{\ell+m+1} - \tau_{\ell-m})$$

abgeschätzt werden. Es folgt

$$\left|s_m\boldsymbol{d}(t \mid T) - T_{m-2}s_m\boldsymbol{d}(t \mid T)\right| \leq \frac{1}{(m-1)!} \max_{k=0,\dots,m} |\tau_{\ell-m+k} - \tau_{\ell+1+k}|^{m-1} \left\|s_m^{(m-1)}\right\|_{\infty,I_\ell}.$$

□

Mit Hilfe der de Boor–Fix Funktionale werden im folgenden Lemma die Koeffizienten des Splines abgeschätzt, welcher das Taylorrestglied im feineren Splineraum darstellt.

Lemma 4.6 *Für einen Spline $s_m\boldsymbol{d} \in \mathbb{S}_m(T)$ zu einer Knotenfolge $T_{n,m}$ mit einfachen inneren Knoten und sein Taylorpolynom $T_{m-2}s_m\boldsymbol{d}$ vom Grad $m-2$ gilt für das de Boor–Fix Funktional $\lambda_{\ell,m}$ auf dem Intervall $I_\ell = [\tau_{\ell-m}, \tau_{\ell+m+1}]$*

$$\|\lambda_{\ell,m}\left(s_m\boldsymbol{d} - T_{m-2}s_m\boldsymbol{d}\right)\|_2 \leq \frac{2^{m-1}}{(m-1)!} \max_{r=0,\ldots,m} |\tau_{\ell-m+r} - \tau_{\ell+1+r}|^{m-1} \left\|s_m^{(m-1)}\boldsymbol{d}\right\|_{\infty,I_\ell}. \tag{4.19}$$

Beweis: Sei wieder $I_\ell = [\tau_{\ell-m}, \tau_{\ell+m+1}]$. Aus

$$s_m\boldsymbol{d}(t \mid T) = \sum_{k=0}^{m-2} \frac{s_m^{(k)}\boldsymbol{d}(t^* \mid T)}{k!}(t-t^*)^k + \frac{1}{(m-2)!}\int_{t^*}^{t}(t-\vartheta)^{m-2}\, s_m^{(m-1)}\boldsymbol{d}(\vartheta \mid T)\, d\vartheta. \tag{4.20}$$

folgt für eine Taylorentwicklung von $s_m\boldsymbol{d}$ an $t^* := \frac{1}{2}(\tau_{\ell+m+1} + \tau_{\ell-m})$, dass

$$s_m\boldsymbol{d}(t \mid T) - T_{m-2}s_m\boldsymbol{d}(t \mid T) = \frac{1}{(m-2)!}\int_{t^*}^{t}(t-\vartheta)^{m-2}\, s_m^{(m-1)}\boldsymbol{d}(\vartheta \mid T)\, d\vartheta$$

und somit

$$\frac{\mathrm{d}^\nu}{\mathrm{d}t^\nu}\left(s_m\boldsymbol{d} - T_{m-2}s_m\boldsymbol{d}\right)(t) = \frac{1}{(m-\nu-2)!}\int_{t^*}^{t}(t-\vartheta)^{m-\nu-2}\, s_m^{(m-1)}\boldsymbol{d}(\vartheta \mid T)\, d\vartheta. \tag{4.21}$$

Mit (4.21) und (3.31) folgt für $\nu = 0, \ldots, m$

$$\begin{aligned}\frac{\mathrm{d}^\nu}{\mathrm{d}t^\nu}\psi_{\ell m}(t)\frac{\mathrm{d}^{m-\nu}}{\mathrm{d}t^{m-\nu}}&\left(s_m\boldsymbol{d} - T_{m-2}s_m\boldsymbol{d}\right)(t) = \\ &(-1)^\nu\frac{\nu!}{(\nu-2)!}\sum_{\substack{K\subseteq\{1,\ldots,m\}\\ \#K=m-\nu}}\prod_{k\in K}(\tau_{\ell+k}-t)\int_{t^*}^{t}(t-\vartheta)^{\nu-2}\, s_m^{(m-1)}\boldsymbol{d}(\vartheta \mid T)\, d\vartheta.\end{aligned}$$

Für die de Boor–Fix Funktionale wählt man $t_\ell \in [\tau_\ell, \tau_{\ell+m+1}]$ und mit den gleichen Argumenten wie im Beweis zu Lemma 4.5 folgt

$$\begin{aligned}\left\|\psi_{\ell m}^{(\nu)}(t_\ell)\left(s_m\boldsymbol{d} - T_{m-2}s_m\boldsymbol{d}\right)^{(m-\nu)}(t_\ell)\right\|_2 \leq \\ \frac{\nu!}{(\nu-2)!}\binom{m}{m-\nu}\max_{r=0,\ldots,m}|\tau_{\ell-m+r} - \tau_{\ell+1+r}|^{m-\nu}\int_{t^*}^{t_\ell}(t_\ell-\vartheta)^{\nu-2}\left\|s_m^{(m-1)}\boldsymbol{d}(\vartheta \mid T)\right\|_\infty d\vartheta.\end{aligned}$$

Durch Abschätzen des Splines im Integralausdruck durch sein Maximum auf dem Intervall I_ℓ und anschließendem Ausrechnen des Integrals folgt

$$\begin{aligned}&\left\|\psi_{\ell m}^{(\nu)}(t_\ell)\left(s_m\boldsymbol{d} - T_{m-2}s_m\boldsymbol{d}\right)^{(m-\nu)}(t_\ell)\right\|_2 \\ &\quad\leq \frac{\nu!}{(\nu-2)!}\binom{m}{m-\nu}\max_{r=0,\ldots,m}|\tau_{\ell-m+r} - \tau_{\ell+1+r}|^{m-\nu}\left\|s_m^{(m-1)}\boldsymbol{d}\right\|_{\infty,I_\ell}\int_{t^*}^{t_\ell}(t_\ell-\vartheta)^{\nu-2}\, d\vartheta \\ &\quad\leq \frac{\nu!}{(\nu-2)!}\binom{m}{m-\nu}\max_{r=0,\ldots,m}|\tau_{\ell-m+r} - \tau_{\ell+1+r}|^{m-\nu}\left\|s_m^{(m-1)}\boldsymbol{d}\right\|_{\infty,I_\ell}\frac{1}{\nu-1}|t_\ell - t^*|^{\nu-1} \\ &\quad\leq \nu\binom{m}{m-\nu}\max_{r=0,\ldots,m}|\tau_{\ell-m+r} - \tau_{\ell+1+r}|^{m-1}\left\|s_m^{(m-1)}\boldsymbol{d}\right\|_{\infty,I_\ell}.\end{aligned} \tag{4.22}$$

Mit (3.30) und der Dreiecksungleichung können nun die de Boor–Fix Funktionale mit

$$\|\lambda_{\ell,m}\left(s_m\boldsymbol{d} - T_{m-2}s_m\boldsymbol{d}\right)\|_2 \leq \frac{1}{m!}\sum_{\nu=1}^{m}\left\|\psi_{\ell m}^{(\nu)}(t_\ell)\left(s_m\boldsymbol{d} - T_{m-1}s_m\boldsymbol{d}\right)^{(m-\nu)}(t_\ell)\right\|_2 \tag{4.23}$$

abgeschätzt werden. Einsetzen von (4.22) in (4.23) liefert

$$\begin{aligned}
&\left\|\lambda_{\ell,m}\left(s_m\boldsymbol{d}-T_{m-2}s_m\boldsymbol{d}\right)\right\|_2\\
&\quad\le\ \frac{1}{m!}\sum_{\nu=0}^{m}\left\|\nu\binom{m}{m-\nu}\max_{r=0,\dots,m}\left|\tau_{\ell-m+r}-\tau_{\ell+1+r}\right|^{m-1}\left\|s_m^{(m-1)}\boldsymbol{d}\right\|_{\infty,I_\ell}\right\|_2\\
&\quad\le\ \frac{1}{m!}\max_{r=0,\dots,m}\left|\tau_{\ell-m+r}-\tau_{\ell+1+r}\right|^{m-1}\left\|s_m^{(m-1)}\boldsymbol{d}\right\|_{\infty,I_\ell}\sum_{\nu=0}^{m}\nu\binom{m}{m-\nu}\\
&\quad\le\ \frac{2^{m-1}}{(m-1)!}\max_{r=0,\dots,m}\left|\tau_{\ell-m+r}-\tau_{\ell+1+r}\right|^{m-1}\left\|s_m^{(m-1)}\boldsymbol{d}\right\|_{\infty,I_\ell},
\end{aligned}\tag{4.24}$$

womit die Aussage gezeigt ist. □

Als Nächstes folgt die Hauptaussage dieses Kapitels. Eine Abschätzung der Waveletkoeffizienten $\boldsymbol{w}^j$ einer Zerlegung nach (4.9) bzw. (4.11). Mit dieser Abschätzung kann eine Kompensation $\alpha_{j,k}$ für (4.14) angegeben werden.

Satz 4.7 *Sei $s_m\boldsymbol{d}^{j+1}=\boldsymbol{d}^{j+1}\boldsymbol{N}^m(\cdot\mid T^{j+1}_{m,n_{j+1}})\in\mathbb{S}_m(T^{j+1}_{m,n_{j+1}})$ ein Spline zu einer einfachen Knotenfolge $T^{j+1}_{m,n_{j+1}}$ mit einer Zerlegung*

$$\boldsymbol{d}^{j+1}\boldsymbol{N}^m(\cdot\mid T^{j+1})=\boldsymbol{d}^j\boldsymbol{N}^m(\cdot\mid T^j)+\boldsymbol{w}^j\boldsymbol{N}^m(\cdot\mid T^{j+1})\in V_j\oplus W_j$$

nach (4.11). *Die Waveletkoeffizienten $\boldsymbol{w}^j=\left(w^j_\ell:\ell=1,\dots,n_{j+1}\right)$ erfüllen die Ungleichung*

$$\left\|w^j_\ell\right\|_2\le n_{j+1}\frac{2^{m-1}}{(m-1)!}\max_{r=0,\dots,m}\left|\tau^{j+1}_{k-m+r}-\tau^{j+1}_{k+1+r}\right|^{m-1}\left|(\boldsymbol{I}-\boldsymbol{B}\boldsymbol{A})_{k,\ell}\right|\left\|s_m^{(m-1)}\boldsymbol{d}^{j+1}\right\|_{\infty,[T^{j+1}]},\tag{4.25}$$

wobei

$$k=\operatorname{argmax}\left(\max_{r=0,\dots,m}\left|\tau^{j+1}_{\cdot-m+r}-\tau^{j+1}_{\cdot+1+r}\right|^{m-1}\left|(\boldsymbol{I}-\boldsymbol{B}\boldsymbol{A})_{\cdot,\ell}\right|\right)$$

sowie $\boldsymbol{A}=\boldsymbol{A}(T^j,T^{j+1})$ und $\boldsymbol{B}=\boldsymbol{B}(T^j,T^{j+1})$.

Beweis: Sei $I_\ell=[\tau_{\ell-m},\tau_{\ell+m+1}]$, $\boldsymbol{A}=\boldsymbol{A}(T^j,T^{j+1})$, $\boldsymbol{B}=\boldsymbol{B}(T^j,T^{j+1})$ sowie $s_m\boldsymbol{d}^{j+1}=\boldsymbol{d}^{j+1}\boldsymbol{N}^m(\cdot\mid T^{j+1})\in\mathbb{S}_m(T^{j+1})$ und $T_{m-2}s_m\boldsymbol{d}^{j+1}$ das Taylorpolynom vom Grad $m-2$ an der später festgelegten Stelle t^*. Da $T_{m-2}s_m\boldsymbol{d}^{j+1}\in\Pi_{m-2}\subset V_{j+1}\cap V_j$, existiert eine eindeutige Darstellung von $T_{m-2}s_m\boldsymbol{d}^{j+1}$ bezüglich der B–Spline Basen von V_{j+1} und V_j. Seien $\boldsymbol{g}^j$ sowie $\boldsymbol{g}^{j+1}$ die entsprechenden Koeffizienten, so dass

$$T_{m-2}s_m\boldsymbol{d}^{j+1}=\boldsymbol{g}^j\boldsymbol{N}^m(\cdot\mid T^j)=\boldsymbol{g}^{j+1}\boldsymbol{N}^m(\cdot\mid T^{j+1}).$$

Die Koeffizienten $\boldsymbol{g}^j$ und $\boldsymbol{g}^{j+1}$ können mit Hilfe der de Boor–Fix Funktionale (3.30) berechnet werden. Für sie gilt:

$$\boldsymbol{g}^j\boldsymbol{A}=\boldsymbol{g}^{j+1}\in\ker(\boldsymbol{I}-\boldsymbol{B}\boldsymbol{A}).$$

Seien im Folgenden $\boldsymbol{\lambda}^{j+1}_m:=\left(\lambda^{j+1}_{k,m}:k=1,\dots,n_{j+1}\right)$ die zu einem Zeilenvektor arrangierten de Boor–Fix Funktionale. Für die Waveletkoeffizienten $\boldsymbol{w}^j$ gilt:

$$\boldsymbol{w}^j=\boldsymbol{d}^{j+1}(\boldsymbol{I}-\boldsymbol{B}\boldsymbol{A})=\boldsymbol{\lambda}^{j+1}_m\left(s_m\boldsymbol{d}^{j+1}\right)(\boldsymbol{I}-\boldsymbol{B}\boldsymbol{A})=\boldsymbol{\lambda}^{j+1}_m\left(s_m\boldsymbol{d}^{j+1}-T_{m-2}s_m\boldsymbol{d}^{j+1}\right)(\boldsymbol{I}-\boldsymbol{B}\boldsymbol{A})$$

und damit

$$\left\|w_\ell^j\right\|_2 = \left\|\boldsymbol{w}^j \boldsymbol{e}_\ell\right\|_2 = \left\|\boldsymbol{\lambda}_m^{j+1}\left(s_m \boldsymbol{d}^{j+1} - T_{m-2} s_m \boldsymbol{d}^{j+1}\right)(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A})\,\boldsymbol{e}_\ell\right\|_2 . \tag{4.26}$$

Zur weiteren Abschätzung des ℓ–ten Waveletkoeffizienten sei für diesen der Entwicklungspunkt t_k^* der jeweiligen Taylorentwicklung als Mittelpunkt des Intervalls $I_k := \left[\tau_{k-m}^{j+1}, \tau_{k+m+1}^{j+1}\right]$, also $t_k^* = \frac{1}{2}\left(\tau_{k-m}^{j+1} + \tau_{k+m+1}^{j+1}\right)$ gewählt. Die Auswertungsstelle t_k der Funktionen für die jeweiligen de Boor–Fix Funktionale wird aus dem Intervall $\left[\tau_k^{j+1}, \tau_{k+m+1}^{j+1}\right]$ gewählt. Mit (4.26) und unter Verwendung von Lemma 4.6 kann ein Waveletkoeffizient nun durch

$$\begin{aligned}
&\left\|w_\ell^j\right\|_2 \\
&\leq \left\|\boldsymbol{\lambda}_m^{j+1}\left(s_m \boldsymbol{d}^{j+1} - T_{m-2} s_m \boldsymbol{d}^{j+1}\right)(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A})\boldsymbol{e}_\ell\right\|_2 \\
&\leq \sum_{k=1}^{n_{j+1}} \left\|\lambda_{k,m}^{j+1}\left(s_m \boldsymbol{d}^{j+1} - T_{m-2} s_m \boldsymbol{d}^{j+1}\right)\right\|_2 |(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A})_{k,\ell}| \\
&\leq n_{j+1} \max_{k=1,\ldots,n_{j+1}} \left(\left\|\lambda_{k,m}^{j+1}\left(s_m \boldsymbol{d}^{j+1} - T_{m-2} s_m \boldsymbol{d}^{j+1}\right)\right\|_2 |(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A})_{k,\ell}|\right) \\
&\leq n_{j+1} \max_{k=1,\ldots,n_{j+1}} \left(\frac{2^{m-1}}{(m-1)!} \max_{r=0,\ldots,m} \left|\tau_{k-m+r}^{j+1} - \tau_{k+1+r}^{j+1}\right|^{m-1} \right. \\
&\qquad\qquad \left. \left\|s_m^{(m-1)} \boldsymbol{d}^{j+1}\right\|_{\infty, I_k} |(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A})_{k,\ell}| \right) \\
&\leq n_{j+1} \frac{2^{m-1}}{(m-1)!} \left\|s_m^{(m-1)} \boldsymbol{d}^{j+1}\right\|_{\infty, [T^{j+1}]} \\
&\qquad \max_{k=1,\ldots,n_{j+1}} \left(\max_{r=0,\ldots,m} \left|\tau_{k-m+r}^{j+1} - \tau_{k+1+r}^{j+1}\right|^{m-1} |(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A})_{k,\ell}| \right)
\end{aligned} \tag{4.27}$$

abgeschätzt werden. Fixiert man k, so dass

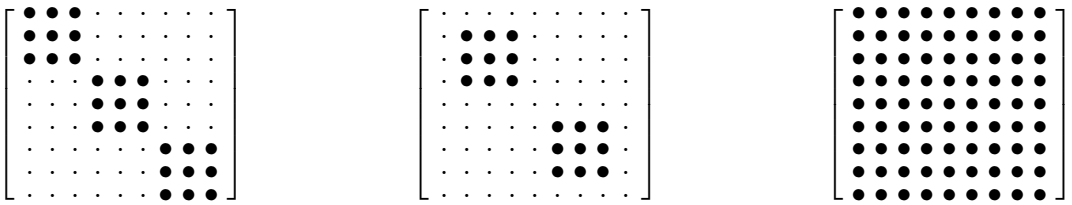
$$k = \operatorname{argmax}\left(\max_{r=0,\ldots,m} \left|\tau_{\cdot-m+r}^{j+1} - \tau_{\cdot+1+r}^{j+1}\right|^{m-1} |(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A})_{\cdot,\ell}| \right),$$

vereinfacht sich der Ausdruck (4.27) zu

$$\left\|w_\ell^j\right\|_2 \leq n_{j+1} \frac{2^{m-1}}{(m-1)!} \max_{r=0,\ldots,m} \left|\tau_{k-m+r}^{j+1} - \tau_{k+1+r}^{j+1}\right|^{m-1} |(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A})_{k,\ell}| \left\|s_m^{(m-1)} \boldsymbol{d}^{j+1}\right\|_{\infty, [T^{j+1}]} . \tag{4.28}$$

□

Der Index k aus (4.25) erscheint zunächst willkürlich aus dem gesamten Summationsindex $1, \ldots, n_{j+1}$ stammen zu können. Tatsächlich wird dieser Index aber *lokal* um den ℓ–ten Eintrag der entsprechenden Spalte aus $(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A})$ gewählt und ist von der jeweiligen Knotenentfernstrategie abhängig. Es gibt drei mögliche Strukturen der Knoteneinfügematrix $\boldsymbol{A}$ bzw. der Waveletmatrix $(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A})$, siehe Abbildung 4.5. Sind die zu entfernenden Knoten exakt $m+2$ Knoten voneinander entfernt, sind die Matrizen $\boldsymbol{A}$ und $(\boldsymbol{I} - \boldsymbol{B}\boldsymbol{A})$ Blockmatrizen (Abbildung 4.5(a)). Sind die Knoten sogar noch weiter voneinander separiert, entstehen in der Waveletmatrix Nullspalten zwischen den Blöcken (Abbildung 4.5(b)). Die entsprechenden Waveletkoeffizienten sind dann ebenfalls Null. Grund hierfür ist, dass mindestens ein B–Spline nicht von der Knotenentfernstrategie betroffen ist, siehe Abbildung 4.6. Ist der Abstand zwischen den entfernten Knoten



(a) Blockgestalt (b) separierte Blöcke (c) vollbesetzte Matrix

Abb. 4.5: Schematische Struktur der Waveletmatrix $(\boldsymbol{I} - \boldsymbol{BA})$.

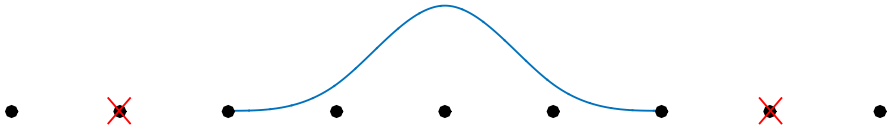


Abb. 4.6: Separierte Knotenentfernstrategie, ein B–Spline ist nicht betroffen.

kleiner als $m+2$, ist die Knoteneinfügematrix $\boldsymbol{A}$ bandiert und die Waveletmatrix $(\boldsymbol{I} - \boldsymbol{BA})$ vollbesetzt (Abbildung 4.5(c)). In allen Fällen klingen die Einträge der ℓ–ten Spalte der Matrix $(\boldsymbol{I} - \boldsymbol{BA})$ zum Rand des Trägers ab, siehe Abbildung 4.7.

Mit Hilfe von (4.25) kann die $(m-1)$–te Ableitung des Splines gegen die jeweiligen Waveletkoeffizienten der verschiedenen Zerlegungsstufen abgeschätzt werden und somit die Abklingrate der Waveletkoeffizienten bestimmt werden. Mit

$$k = \operatorname{argmax}\left(\max_{r=0,\ldots,m}\left|\tau^{j+1}_{\cdot-m+r} - \tau^{j+1}_{\cdot+1+r}\right|^{m-1} \left|(\boldsymbol{I} - \boldsymbol{BA})_{\cdot,\ell}\right|\right)$$

und unter der Voraussetzung, dass $|(\boldsymbol{I} - \boldsymbol{BA})_{k,\ell}| \neq 0$, gilt:

$$\max_{r=0,\ldots,m}\left|\tau^{j+1}_{k-m+r} - \tau^{j+1}_{k+1+r}\right|^{-(m-1)} \left|(\boldsymbol{I} - \boldsymbol{BA})_{k,\ell}\right|^{-1} \left\|w^j_\ell\right\|_2 \leq n_{j+1}\frac{2^{m-1}}{(m-1)!}\left\|s^{(m-1)}_m \boldsymbol{d}^{j+1}\right\|_{\infty,[T^{j+1}]}.$$

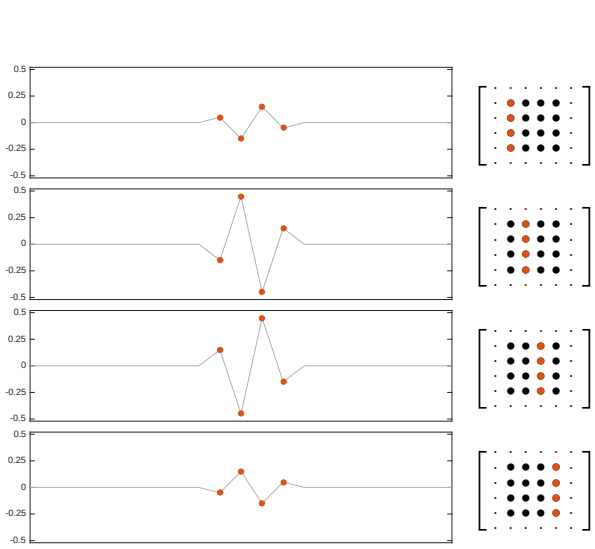
Der Faktor $\frac{2^{m-1}}{(m-1)!}$ ist unabhängig von der Zerlegungsstufe j und somit für alle Zerlegungsstufen konstant. Die Dimension n_{j+1} ist ein Faktor, der indirekt die globale Knotenentfernstrategie repräsentiert. Dieser ist davon abhängig wie viele Knoten im Zerlegungsschritt zuvor entfernt wurden. Die Dimension n_j kann mit n_N als größte Dimension abgeschätzt werden und ist damit ebenfalls unabhängig von der Zerlegungsstufe.

Für die Abklingrate $\alpha_{j,\ell}$ zum Waveletkoeffizienten w^j_ℓ im Analysespline (4.15) sind aus (4.25) im Wesentlichen zwei Faktoren von Interesse. Zum einen der entsprechende Eintrag aus der Waveletmatrix $|(\boldsymbol{I} - \boldsymbol{BA})_{k,\ell}|$, zum anderen die Knotendifferenz

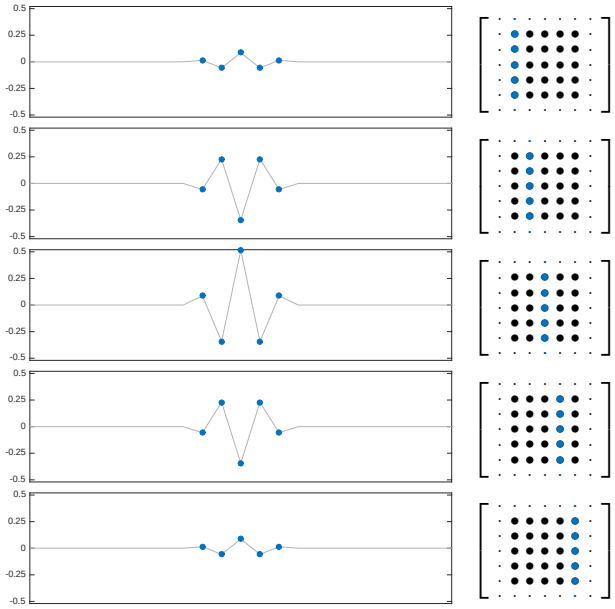
$$\max_{r=0,\ldots,m}\left|\tau^{j+1}_{k-m+r} - \tau^{j+1}_{k+1+r}\right|^{m-1}.$$

Diese Faktoren können geometrisch interpretiert werden:

- Der Faktor $|(\boldsymbol{I} - \boldsymbol{BA})_{k,\ell}|$ ist ein Maß, inwieweit die Knotenentfernstrategie den entsprechenden Waveletkoeffizienten beeinflusst.



(a) gerader Grad ($m = 2$)



(b) ungerader Grad ($m = 3$)

Abb. 4.7: Typische Struktur der Spalten von $(\boldsymbol{I} - \boldsymbol{BA})$.

- Die Knotendifferenz

$$\max_{r=0,\ldots,m} \left| \tau_{k-m+r}^{j+1} - \tau_{k+1+r}^{j+1} \right|^{m-1}$$

  entspricht dem Abklingfaktor. Im Fall einer äquidistanten Knotenfolge und einer Knotenausdünnung von jedem zweiten Knoten pro Zerlegungsstufe entspricht dieser Faktor der erwarteten Abklingrate 2^{-j} von kardinalen Splines [8]. Bei einer nicht gleichverteilten Knotenfolge wird dieser Faktor durch die *lokale Knotendichte* um das Intervall $\left[\tau_k^{j+1}, \tau_{k+1}^{j+1}\right]$ beeinflusst und liefert somit nicht nur einen Zerlegungsstufen, sondern auch einen *lokal abhängigen* Abklingfaktor.

Die Abklingrate $\alpha_{j,\ell}$ zum ℓ–ten Waveletkoeffizienten der j–ten Zerlegungsstufe kann somit wie folgt angegeben werden:

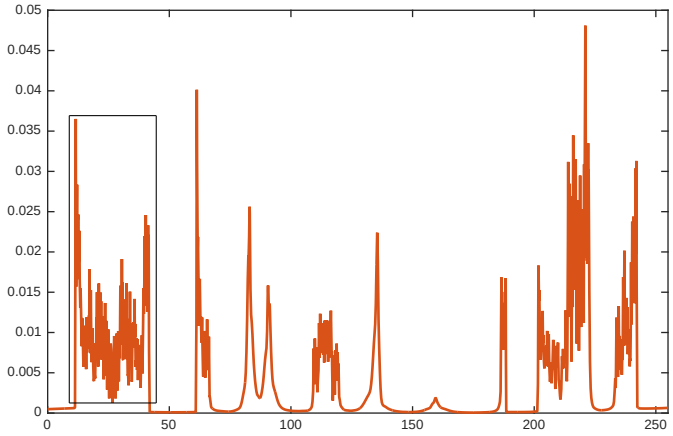
$$\alpha_{j,\ell} := \begin{cases} |(\boldsymbol{I} - \boldsymbol{BA})_{k,\ell}|^{-1} \max\limits_{r=0,\ldots,m} \left| \tau_{k-m+r}^{j+1} - \tau_{k+1+r}^{j+1} \right|^{-m+1}, & |(\boldsymbol{I} - \boldsymbol{BA})_{k,\ell}| \neq 0, \\ 0, & |(\boldsymbol{I} - \boldsymbol{BA})_{k,\ell}| = 0, \end{cases} \tag{4.29}$$

wobei $k = \operatorname{argmax}\left(\max_{r=0,\ldots,m} \left| \tau_{\cdot-m+r}^{j+1} - \tau_{\cdot+1+r}^{j+1} \right|^{m-1} |(\boldsymbol{I} - \boldsymbol{BA})_{\cdot,\ell}| \right)$.

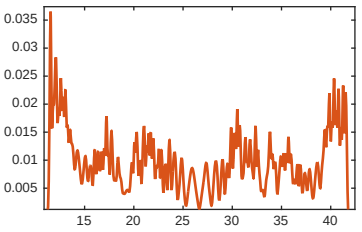
Ein Beispiel für einen Analysespline $s_m \boldsymbol{c}$ nach (4.15) mit einer nach (4.29) gewählten Abklingrate für die Waveletkoeffizienten ist in Abbildung 4.8(a) dargestellt. Im Vergleich mit Abbildung 4.4 erhält man hier durch die Gewichtung klar lokalisierte Spitzen im Analysespline. Diese können unmittelbar über ihren Parameterwert mit einer Ecke bzw. Kante in der Fräsbahn identifiziert werden.

4.2.3 Interpretation des Analysesplines

Die markanten Spitzen des Analysesplines (4.15) repräsentieren jeweils ein geometrisches Merkmal. Um die Spitzen in diesem Spline zu lokalisieren, müssen die Extremwerte bzw. Maxima



(a) Summation mit Gewichtung



(b) Vergrößerung aus linker Abbildung

Abb. 4.8: Summation der Detailanteile aus Abbildung 4.3 zum Testprogramm mit Gewichtung nach (4.15).

berechnet werden. Für einen Analysespline $s_m\boldsymbol{c}$ nach (4.15) gilt

$$s_m\boldsymbol{c}(\cdot \mid T^n) = \boldsymbol{c}\boldsymbol{N}^m(\cdot \mid T^N) = \sum_{j=1}^{N} c_j N_j^m(\cdot \mid T^N), \qquad c_\ell = \sum_{j=0}^{N-1} \alpha_{j,\ell} \left\| \left(\boldsymbol{w}^j \prod_{k=j}^{N-1} \boldsymbol{A}(T^k, T^{k+1}) \right) \boldsymbol{e}_\ell \right\|_2,$$

wobei $\boldsymbol{e}_\ell$ dem ℓ–ten Einheitsvektor aus $\mathbb{R}^N$ sowie $V_N = \mathbb{S}_m(T^N)$ dem Splineraum mit feinster Auflösung entspricht. Für die Dimension von V_N gilt $\dim V_N = n_N$. Um Extremwerte dieser skalaren Splinefunktion zu berechnen, bestimmt man mit den Nullstellen der ersten Ableitung die Kandidaten und prüft diese mittels der zweiten Ableitung auf Maximum bzw. Minimum. Hierfür seien $s_{m-1}\boldsymbol{c}^{(1)}$ und $s_{m-2}\boldsymbol{c}^{(2)}$ die ersten zwei Ableitungen des Analysesplines. Die Koeffizienten $\boldsymbol{c}^{(1)}$ bzw. $\boldsymbol{c}^{(2)}$ können mittels (3.19) berechnet werden.

Extremwertsuche

Um die Nullstellen einer Splinefunktion zu bestimmen, ist das Newton–Verfahren [13] eine der gängigsten Methoden. Bei diesem Verfahren wird iterativ je eine Nullstelle des Splines bestimmt. Für jede Nullstelle muss dabei der Startwert des Verfahrens passend gewählt werden. Wie man in Abbildung 4.8(b) sehen kann, müssen für einen Analysespline in der Regel viele, zum Teil sehr dicht liegende Extremwerte bzw. Nullstellen bestimmt werden. Die Wahl eines passenden Startwertes kann hierbei mitunter sehr schwierig werden. Mørken und Reimers präsentieren in [39] ein Verfahren, welches alle Nullstellen einer skalaren Splinefunktion durch Verfeinerung des Kontrollpolygons simultan ermittelt. Es nutzt die variationsmindernde Eigenschaft eines Splines aus, dass das Kontrollpolygon jede Hyperebene mindestens genau so oft schneidet wie der Spline selbst [38]. Hat also das Kontrollpolygon keine Nullstelle, so hat der Spline auch keine. Das Verfahren konvergiert quadratisch gegen die Nullstelle und ist somit vergleichbar mit dem Newton–Verfahren. Es entfällt außerdem das Problem der Startwertsuche des Newton–Verfahrens.

Motiviert durch den *Schoenberg–Operator* werden die Kontrollpunkte einer skalaren Splinefunktion $s_m\boldsymbol{d} \in \mathbb{S}_m(T_{m,n})$, $\boldsymbol{d} \in \mathbb{R}^{1\times n}$, mit den *Greville–Abszissen* $\hat{\boldsymbol{t}}$ identifiziert. Das Kontrollpolygon ist somit eine stückweise lineare Funktion mit Ecken an $(\hat{t}_j, d_j)$. Der Algorithmus von Mørken und Reimers berechnet zunächst die Nullstellen des Kontrollpolygons. Zum Beispiel ergibt sich

für eine Nullstelle zwischen den Kontrollpunkten d_j und d_{j+1},

$$(1-\lambda)d_j + \lambda\, d_{j+1} = 0, \qquad \lambda \in [0,1],$$
$$\lambda = \frac{-d_j}{d_{j+1} - d_j}. \tag{4.30}$$

Da die Kontrollpunkte mit den Greville–Abszissen identifiziert werden, kann der entsprechende Parameterwert t^* der Nullstelle durch

$$t^* = (1-\lambda)\hat{t}_j + \lambda\, \hat{t}_{j+1} \tag{4.31}$$

mit λ aus (4.30) ermittelt werden. Einsetzen von (4.30) und der Definition der Greville–Abszissen (3.14) in (4.31) liefert

$$t^* = \hat{t}_j - \frac{d_j(\tau_{j+m+1} - \tau_{j+1})}{m(d_j + d_{j+1})}. \tag{4.32}$$

Die Parameterwerte der Nullstellen des Kontrollpolygons werden als neue Knoten in die Knotenfolge eingefügt. Dieser Vorgang wird iteriert bis entweder die gewünschte Genauigkeit erreicht wird oder das Kontrollpolygon keine Nullstellen mehr hat. Zur Vereinfachung wird im folgenden Algorithmus 4.8 nur die erste (im Parameterbereich links) Nullstelle einer Splinefunktion berechnet.

Algorithmus 4.8 (Mørken & Reimers)

Eingabe:

- *Knotenfolge* $T = T_{n,m}$
- *Kontrollpunkte* $\boldsymbol{d} = [d_1, \ldots, d_n] \in \mathbb{R}^{1\times n}$
- *gewünschte Genauigkeit* $\varepsilon > 0$

Prozedur:

1. **Bestimme***:*

$$k := \begin{cases} \operatorname{argmin}_j(d_j d_{j+1} < 0), & \{j : d_j d_{j+1} < 0\} \neq \emptyset \\ 0, & \textit{sonst} \end{cases}$$

2. **Initialisiere***:*

   IF $k \neq 0$

   - $\hat{t}_k := \dfrac{1}{m}\displaystyle\sum_{j=1}^{m} \tau_{k+j}$
   - $x_0 := \hat{t}_k - \dfrac{d_k(\tau_{k+m+1} - \tau_{k+1})}{m(d_k + d_{k+1})}, \qquad x_1 := \infty$

   END

3. **WHILE** $k \neq 0 \quad \wedge \quad |x_0 - x_1| > \varepsilon$
   - $\boldsymbol{d} := \boldsymbol{d}\,\boldsymbol{A}(x_o)$
   - $T := T \cup \{x_0\}$
   - $k := \begin{cases} \operatorname{argmin}_j(d_j d_{j+1} < 0), & \{j : d_j d_{j+1} < 0\} \neq \emptyset \\ 0, & \textit{sonst} \end{cases}$

IF $k \neq 0$

- $\hat{t}_k := \frac{1}{m} \sum_{j=1}^{m} \tau_{k+j}$
- $x_1 := x_0$
- $x_0 := \hat{t}_k - \frac{d_k(\tau_{k+m+1} - \tau_{k+1})}{m(d_k + d_{k+1})}$

ELSE

- $x_0 := NaN$

END

END

Ergebnis:
Nullstelle der Splinefunktion $s_m\boldsymbol{d}$ bei x_0 mit Genauigkeit ε.

Im dritten Schritt des obigen Algorithmus ist es effizienter den Algorithmus 3.27 von Boehm zu verwenden anstatt die Knoteneinfügematrix $\boldsymbol{A}$ aufzustellen, zumal in jedem Schritt nur ein Knoten eingefügt wird. Der Algorithmus 4.8 funktioniert nur für m–mal stetig differenzierbare Splinefunktionen mit $(m+1)$–fachen Randknoten. Mehrere Iterationsschritte dieses Algorithmus sind in Abbildung 4.9 visualisiert. Möchte man, nachdem die erste Nullstelle gefunden wurde, weitere Nullstellen finden, fügt man einen $(m + 1)$–fachen Knoten an der Position der zuletzt gefundenen Nullstelle ein und führt den Algorithmus für die zweite Hälfte des Splines erneut aus. So können alle Nullstellen gefunden werden. Ändert man den Algorithmus 4.8 ab, so dass im dritten Schritt nicht nur die erste, sondern alle Nullstellen des Kontrollpolygons als Knoten eingefügt werden, können alle Nullstellen simultan berechnet werden. Hier ist es oft wieder effizienter das neue Kontrollpolygon mit Hilfe der Knoteneinfügematrix und des Oslo–Algorithmus zu berechnen.

Die ermittelten Nullstellen $\boldsymbol{t}_N$ von $s_{m-1}\boldsymbol{c}^{(1)}$ sind die Kandidaten der Extremwerte. Es werden anschließend die Bedingungen $s_{m-2}\boldsymbol{c}^{(2)}(\boldsymbol{t}_N \mid T^n) < 0$ bzw. $s_{m-2}\boldsymbol{c}^{(2)}(\boldsymbol{t}_N \mid T^n) > 0$ geprüft. Sollte für ein Element aus $\boldsymbol{t}_N$ die zweite Ableitung des Analysesplines verschwinden, wird es aus der Menge der Extremwerte ausgeschlossen. Die Parameterwerte der Maxima werden im Folgenden mit $\boldsymbol{t}^+$ bzw. die Punktmenge der Parameterwerte mit M^+, die der Minima mit $\boldsymbol{t}^-$ respektive M^- bezeichnet.

Persistenz und Dominanz eines Maximums

Aus Abbildung 4.8 wird klar, dass nicht alle Maxima im Analysespline markante Merkmale in der Fräsbahn ausmachen. Somit ist eine Klassifizierung der Extremwerte nötig. Im Folgenden wird für jedes Maximum ein *Persistenz*– sowie ein *Dominanzwert* ermittelt. Hierbei beschreibt die Persistenz eine globale Größe, die charakteristische Höhe des Maximums in der Splinefunktion und die Dominanz den lokalen Abstand eines Maximums zum nächst größeren Funktionswert, siehe Abbildung 4.11 bzw. 4.13.

Der Begriff *Persistenz* stammt aus der algebraischen Topologie, genauer aus dem Bereich der „persistent homology“ [21], einer algebraischen Methode, um topologische Merkmale von Funktionen robust und numerisch stabil zu bestimmen.

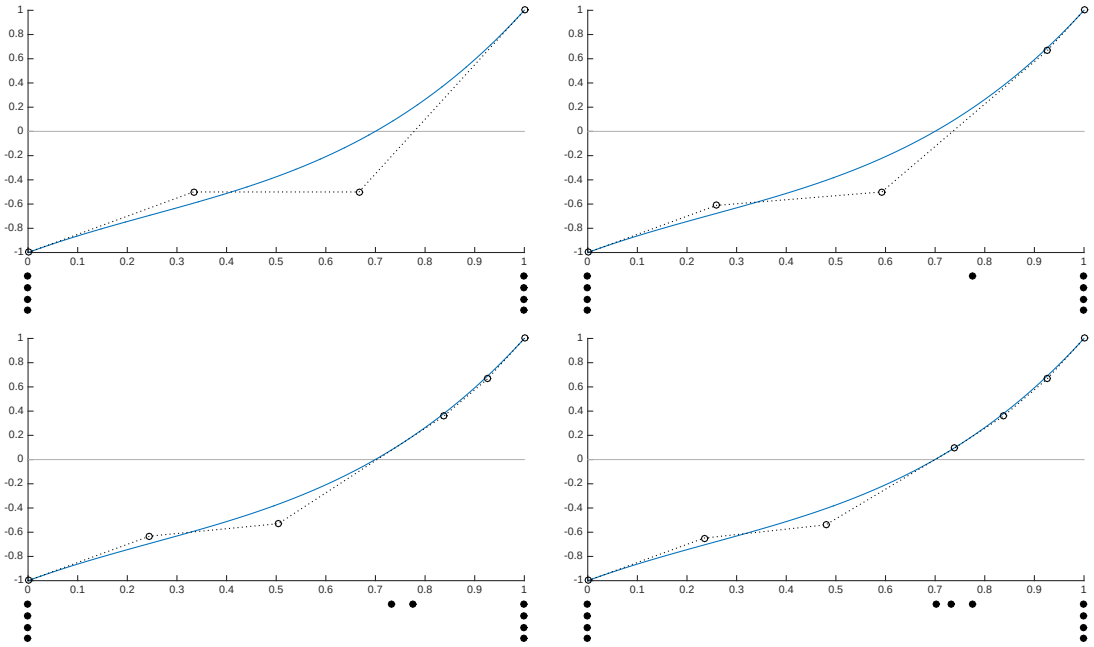


Abb. 4.9: Drei Iterationen des Algorithmus 4.8 zu einer skalaren Splinefunktion mit Knotenvektor $T = (0, 0, 0, 0, 1, 1, 1, 1)$.

Um die Persistenz der Maxima einer Splinefunktion $s_m\boldsymbol{d}(\,\cdot \mid T_{n,m}) \in \mathbb{R}$ zu bestimmen, betrachtet man zu $y \in \mathbb{R}$ die *Subniveaumenge* $\mathbb{R}_y$. Diese enthält alle Parameterwerte, für die die Splinefunktion kleiner als y ist,

$$\mathbb{R}_y := \{t \in \mathcal{I}_m(T_{n,m}) : s_m\boldsymbol{d}(t \mid T_{n,m}) \leq y\}\,.$$

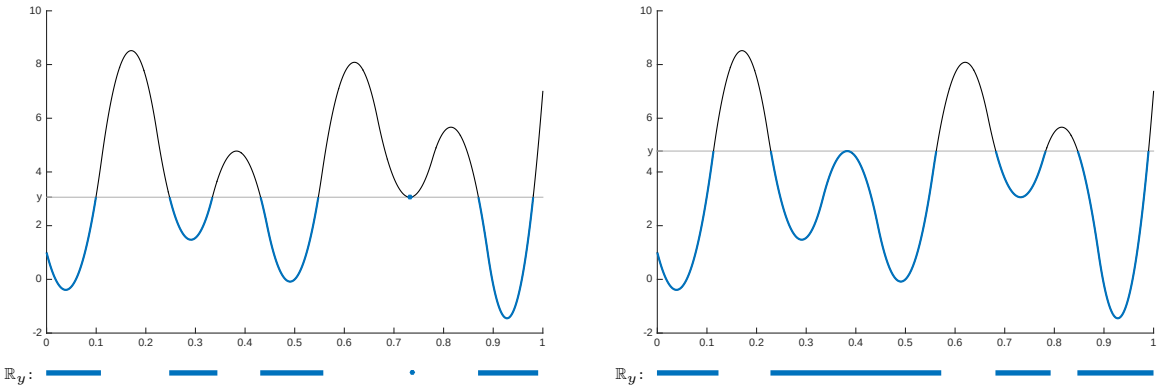
Lässt man y von $-\infty$ bis ∞ laufen, ändert sich, da $s_m\boldsymbol{d}$ stetig ist, die Anzahl der Komponenten der Subniveaumenge $\mathbb{R}_y$ nur dann, wenn y einem Extremwert von $s_m\boldsymbol{d}$ entspricht. Das heißt, an einem lokalen Minimum wird eine neue Komponente in $\mathbb{R}_y$ aufgenommen und an einem lokalen Maximum verschmelzen zwei Komponenten. Abbildung 4.10(a) veranschaulicht die Subniveaumenge $\mathbb{R}_y$ beim Hinzukommen einer neuen Komponente an einem Minimum sowie Abbildung 4.10(b) beim Verschmelzen zweier Komponenten an einem Maximum.

Die Maxima und Minima von $s_m\boldsymbol{d}$ werden paarweise wie folgt zusammengefasst. Immer dann, wenn eine neue Komponente an einem Minimum von $s_m\boldsymbol{d}$ zur Subniveaumenge $\mathbb{R}_y$ hinzugenommen wird, ist dieses Minimum der *Repräsentant* der neuen Komponente. Verschmelzen an einem Maximum zwei Komponenten der Subniveaumenge miteinander, wird dieses Maximum mit dem größeren der beiden Minima, die die verschmelzenden Komponenten repräsentieren, identifiziert. Die Komponenten einer Subniveaumenge werden im Folgenden auch *Topologien* genannt. Für zwei so miteinander identifizierte Parameterwerte t^- und t^+ eines Minimum und Maximum wird die *Persistenz* $\mathcal{P}$ durch

$$\mathcal{P}_{s_m\boldsymbol{d}}(t^-, t^+) := s_m\boldsymbol{d}(t^+ \mid T) - s_m\boldsymbol{d}(t^- \mid T) \tag{4.33}$$

bestimmt. Wenn aus dem Kontext klar ist, zu welcher Splinefunktion die Persistenz berechnet wird, wird zur Vereinfachung der Notation kurz $\mathcal{P}(t^-, t^+)$ geschrieben.

Die Paarungen von Minima und Maxima lassen sich in einem *Persistenz–Diagramm* durch die Abbildung $(t^-, t^+) \mapsto (s_m\boldsymbol{d}(t^-), s_m\boldsymbol{d}(t^+))^T$ visualisieren, siehe Abbildung 4.11. Es ist klar, dass



(a) Subniveaumenge bei Minimum.

(b) Subniveaumenge bei Maximum.

Abb. 4.10: Darstellung der Subniveaumenge $\mathbb{R}_y$.

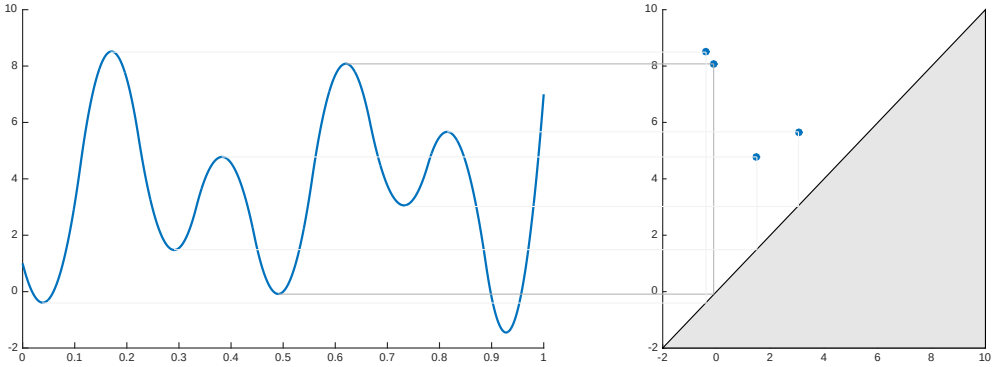


Abb. 4.11: Persistenz–Diagramm eines Splines. Man erkennt die Paarungen der Minima und Maxima.

für eine gültige Paarung immer $s_m\boldsymbol{d}(t^-) < s_m\boldsymbol{d}(t^+)$ gilt, da eine Topologie zunächst entstehen (geboren werden) muss bevor sie mit einer anderen verschmelzen (sterben) kann. Somit sind die Einträge im Persistenz–Diagramm stets oberhalb der Hauptdiagonalen $y_1 = y_2$. Der vertikale Abstand eines Punktes aus diesem Diagramm zur Hauptdiagonalen repräsentiert die Persistenz $\mathcal{P}$ des Maximum $s_m\boldsymbol{d}(t^+)$. Je kleiner dieser Abstand ist, desto wahrscheinlicher ist es, dass dieses Maximum kein markantes Merkmal einer Fräsbahn repräsentiert.

Betrachtet man eine leicht gestörte Version der Splinefunktion aus Abbildung 4.11 so ist das Persistenz–Diagramm dieser Funktion sehr ähnlich zur Originalfunktion, siehe Abbildung 4.12. Dies liegt an der Stabilität der Persistenz–Diagramme, siehe [11]. Kleine Änderungen in der Funktion verursachen nur kleine Änderungen im Persistenz–Diagramm.

Sind die Parameterstellen der Hoch– bzw. Tiefpunkte einer Splinekurve bekannt, kann mit folgendem Algorithmus die Persistenz der Maxima bestimmt werden. Hierfür bezeichnet M^+ die Punktmenge der Parameterwerte aller Maxima sowie M^- die Punktmenge der Parameterwerte aller Minima.

Algorithmus 4.9

Eingabe:

- *Parameterwerte der Minima:* M^-

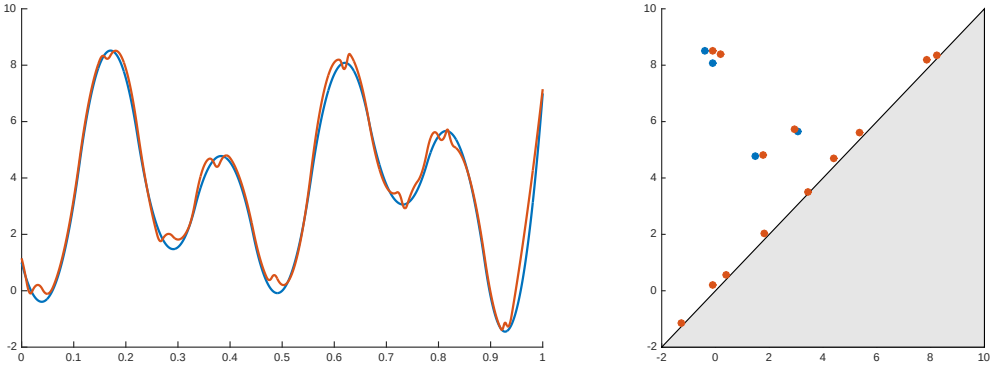


Abb. 4.12: Persistenz–Diagramm zweier Splinefunktionen.

- *Parameterwerte der Maxima:* M^+

Prozedur:

1. **Initialisiere***:*
   - $M := \emptyset$
2. **WHILE** $M^+ \neq \emptyset \ \wedge \ M^- \neq \emptyset$
   - *Parameterwert des kleinsten Maximums*
     $$t^+ := \operatorname{argmin} \left\{ s_m \boldsymbol{d}(t) : t \in M^+ \right\}$$
   - *suche Minimum links bzw. rechts neben aktuellen Maximum*
     $$t_\ell^- := \max \left\{ t : t < t^+ \, , \, t \in M^- \right\}$$
     $$t_r^- := \min \left\{ t : t > t^+ \, , \, t \in M^- \right\}$$
   - *wähle größeres der beiden Minima*
     $$t^- := \operatorname{argmax} \left\{ s_m \boldsymbol{d}(t) : t \in \{ t_\ell^- , t_r^- \} \right\}$$
   - *paare Minimum und Maximum und füge es zur Menge M hinzu*
     $$M := M \cup \left\{ \, (\, s_m \boldsymbol{d}(t^-), s_m \boldsymbol{d}(t^+) \,)^T \, \right\}$$
   - *lösche Parameterwerte des Maximums und Minimums aus der Menge der Maxima bzw. Minima*
     $$\boldsymbol{t}^- := M^- \setminus \{ t^- \}$$
     $$\boldsymbol{t}^+ := M^+ \setminus \{ t^+ \}$$

   END

Ergebnis:
Persistenz Paarungen aller Maxima und Minima der Splinefunktion $s_m \boldsymbol{d}$ in der Menge M.

Nach Ausführung des Algorithmus 4.9 erhält man alle Persistenz–Paarungen von Minima und Maxima. Der entsprechende Persistenzwert kann mittels (4.33) zu jedem Element aus M berechnet werden. Es ist möglich, dass die Anzahl der Maxima um eins größer ist als die Anzahl der Minima. In einem solchen Fall wird dem Maximum, welchem mit Hilfe des Algorithmus 4.9 keinem Minimum zugeordnet werden konnte, ein Persistenzwert von ∞ zugeordnet. Somit kann zu jedem Maximum einer Splinefunktion ein Persistenzwert ermittelt werden.

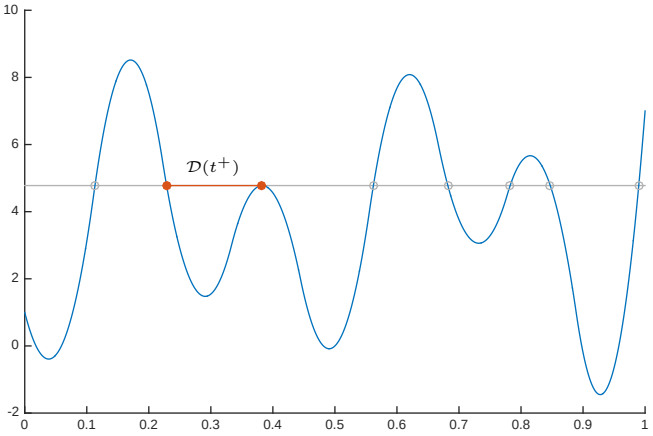
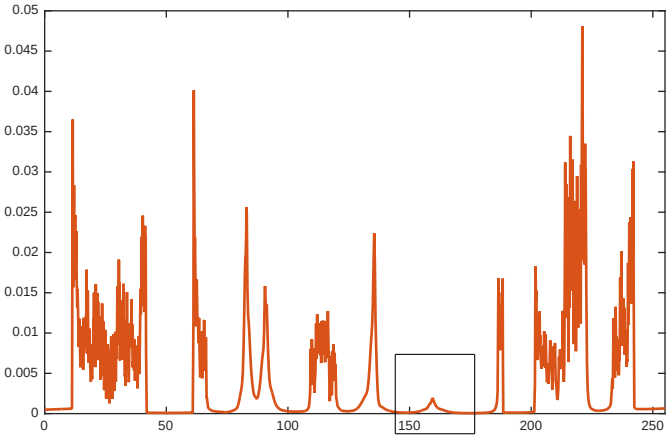
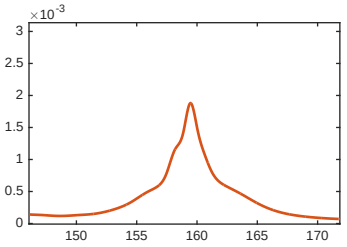


Abb. 4.13: Dominanz $\mathcal{D}$ eines Maximums.



(a) Summation mit Gewichtung



(b) Kleine Spitze mit großer Dominanz

Abb. 4.14: Summation der Detailanteile aus Abbildung 4.3 zum Testprogramm mit Gewichtung nach (4.15) sowie Vergrößerung einer „dominanten“ Spitze.

Die *Dominanz* $\mathcal{D}$ eines Maximums beschreibt lokal den horizontalen Abstand (im Parameterbereich) zum nächst größeren Funktionswert. Formal kann dieser Wert für ein $t^+ \in M^+$ durch

$$\mathcal{D}(t^+) = \begin{cases} \min\left\{ \left| t - t^+ \right| : s_m \boldsymbol{d}(t) = s_m \boldsymbol{d}(t^+) \right\}, & \left\{ t : s_m \boldsymbol{d}(t) \geq s_m \boldsymbol{d}(t^+) \right\} \neq \emptyset \\ \infty, & \text{sonst} \end{cases} \tag{4.34}$$

angegeben werden. Abbildung 4.13 veranschaulicht die Dominanz eines Maximums. Die entsprechenden Schnittpunkte können mit Hilfe des Nullstellen–Algorithmus 4.8 berechnet werden. Eine alleinige Betrachtung der Persistenz reicht zur gewünschten Merkmalserkennung nicht aus. In Abbildung 4.14 bzw. in der entsprechenden Vergrößerung 4.14(b) kann man erkennen, dass es Maxima mit sehr kleiner Persistenz gibt, die lokal allerdings sehr markant sind und deswegen mit in die Merkmalserkennung aufgenommen werden sollten. Dies wird durch Hinzufügen der Dominanz gewährleistet.

Klassifizierung der Maxima

Durch die Bestimmung der Persistenz– und Dominanzwerte der Maxima ist eine Klassifizierung möglich. Ein großer Persistenzwert bezeichnet ein topologisches Merkmal im Analysespline und

somit auch ein Merkmal in der Fräsbahn. Ein großer Dominanzwert ist ein Alleinstellungsmerkmal über einen gewissen Parameterbereich und kennzeichnet demnach einen markanten Punkt über einen gewissen Bereich im Analysespline sowie in der Fräsbahn. Es sind demzufolge diejenigen Maxima von Interesse, welche einen hohen Persistenz– oder Dominanzwert haben. Im Folgenden werden heuristisch Schwellenwerte für Persistenz– und Dominanzwert mittels einer Histogramm–Analyse ermittelt.

Es seien M^+, P und D Mengen mit $\#M^+ = \#P = \#D = k$. Die Menge M^+ beinhaltet die Parameterwerte aller Maxima eines Analysesplines $s_m\boldsymbol{d}(\,\cdot \mid T_{n,m})$, P die jeweiligen Persistenzwerte und D die Dominanzwerte dieser Maxima. In den Mengen P und D können insbesondere Elemente mehrfach vorkommen. Für die Konstruktion der Histogramme müssen zunächst die Wertemengen $W_{\mathcal{P}}$ der Persistenz– bzw. $W_{\mathcal{D}}$ der Dominanzwerte in Klassen aufgeteilt werden, wobei die Wertemengen mit

$$W_{\mathcal{P}} = [\min P, \max P] \quad \text{bzw.} \quad W_{\mathcal{D}} = [\min D, \max D]$$

angegeben werden können, falls $\infty \notin P$ bzw. $\infty \notin D$. Sollte ∞ in den Mengen P, D vorkommen, wird dieses Element zuvor aus den Mengen ausgeschlossen, $P = P \setminus \{\infty\}$, $D = D \setminus \{\infty\}$. Die Anzahl c der Histogrammklassen kann für eine k–elementige Menge heuristisch nach einer Formel von Sturges [50] gewählt werden:

$$c = \lceil 1 + log_2(k) \rceil \,.$$

Hieraus ergibt sich eine Klassenbreite $h_{\mathcal{P}}$ bzw. $h_{\mathcal{D}}$ der Histogrammklassen zu P und D von

$$h_{\mathcal{P}} := \frac{1}{c} \left|\max P - \min P\right| \quad \text{bzw.} \quad h_{\mathcal{D}} := \frac{1}{c} \left|\max D - \min D\right| .$$

In einer Histogrammklasse $H^j_{\mathcal{P}}$ bzw. $H^j_{\mathcal{D}}$, $j = 1, \ldots, c$, befinden sich nun alle Werte aus P bzw. D, die größer als die Klassenuntergrenze aber kleiner oder gleich der Klassenobergrenze sind:

$$\begin{aligned} H^j_{\mathcal{P}} &= \{x \in P \mid (j-1)\, h_{\mathcal{P}} + \min P < x \leq j\, h_{\mathcal{P}} + \min P\} \quad \text{bzw.} \\ H^j_{\mathcal{D}} &= \{x \in D \mid (j-1)\, h_{\mathcal{D}} + \min D < x \leq j\, h_{\mathcal{D}} + \min D\} \,. \end{aligned}$$

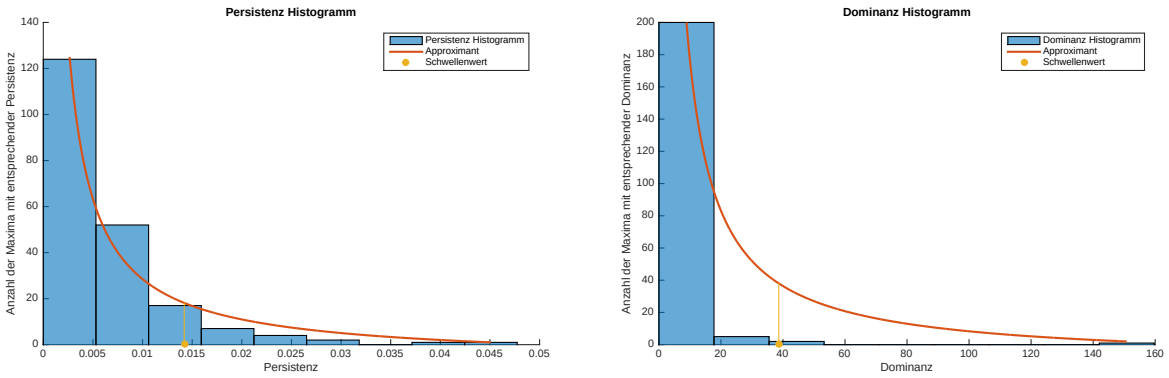
Die Elemente $\min P$ bzw. $\min D$ werden den Histogrammklassen $H^1_{\mathcal{P}}$ bzw. $H^1_{\mathcal{D}}$ zugeordnet. Da für die Histogramme eine konstante Klassenbreite gewählt wird, ist die Kardinalität der Histogrammklassen proportional zur Höhe des entsprechenden Balkens im Histogramm. Somit kann die Höhe eines Balkens direkt mit der Kardinalität der Histogrammklasse identifiziert werden. Im Allgemeinen haben die Histogramme zu Persistenz– und Dominanzwerten eines Analysesplines eine L–förmige Gestalt, siehe Abbildung 4.15, da es sehr viele Maxima mit kleinen Persistenz– bzw. Dominanzwerten gibt und nur wenige mit großen Werten. Zur Bestimmung eines Schwellenwertes werden diese Histogramme mit einer Funktion vom Typ

$$f(x) := a_1 x + a_2 + a_3 x^{-1}$$

approximiert. Hierfür werden die Mittelpunkte der Wertebereiche einer Histogrammklasse mit der Klassenkardinalität identifiziert. Ein solcher *Histogrammpunkt* kann wie folgt angegeben werden:

$$\boldsymbol{p}^j_{\mathcal{P}} := \begin{pmatrix} \min P + (j - \frac{1}{2})\, h_{\mathcal{P}} \\ \#H^j_{\mathcal{P}} \end{pmatrix} \quad \text{bzw.} \quad \boldsymbol{p}^j_{\mathcal{D}} := \begin{pmatrix} \min D + (j - \frac{1}{2})\, h_{\mathcal{D}} \\ \#H^j_{\mathcal{D}} \end{pmatrix}, \quad j = 1, \ldots, c.$$

Zur Vereinfachung der Notation werden die Histogrammpunkte für Persistenz und Dominanz im Folgenden mit $\boldsymbol{p}_j = (x_j, y_j)^T$, $j = 1, \ldots, c$, abgekürzt. Möchte man diese Punkte mit Hilfe der



(a) Persistenz–Histogramm (b) Dominanz–Histogramm

Abb. 4.15: Persistenz– und Dominanz–Histogramme mit Approximant und entsprechende Schwellenwerten.

Funktion f interpolieren, erhält man ein unter– bzw. überbestimmtes Gleichungssystem, falls die Anzahl der Histogrammklassen ungleich drei ist. Da man in der Regel mehr Histogrammklassen hat, wird für den Approximanten eine Least–Squares Lösung dieses Gleichungssystems unter der Nebenbedingung der Endpunktinterpolation angesetzt:

$$\min_{\boldsymbol{a}} \|\boldsymbol{H}\boldsymbol{a} - \boldsymbol{y}\|_2^2, \qquad f(x_1) = y_1 \ \wedge \ f(x_c) = y_c, \tag{4.35}$$

wobei

$$\boldsymbol{H} = \begin{pmatrix} x_1 & 1 & x_1^{-1} \\ \vdots & \vdots & \vdots \\ x_c & 1 & x_c^{-1} \end{pmatrix}, \qquad \boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_c \end{pmatrix} \quad \text{und} \quad \boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}.$$

Die Minimierung von (4.35) läuft auf ein System von Normalengleichungen hinaus und lässt sich mit Hilfe von Lagrange–Multiplikatoren $\boldsymbol{\lambda} \in \mathbb{R}^2$ wie folgt formulieren:

$$\begin{pmatrix} & & x_1 & x_c \\ \boldsymbol{H}^T\boldsymbol{H} & & 1 & 1 \\ & & x_1^{-1} & x_c^{-1} \\ x_1 & 1 & x_1^{-1} & \\ x_c & 1 & x_c^{-1} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{H}^T\boldsymbol{y} \\ y_1 \\ y_c \end{pmatrix}. \tag{4.36}$$

Mit einem durch (4.36) bestimmten Approximanten f kann nun ein Schwellenwert für die Persistenz– bzw. Dominanzwerte ermittelt werden. Hierfür wird die Position von f gesucht, an der die erste Ableitung f' einen bestimmten Wert z annimmt, also

$$f'(x) = z \qquad \Rightarrow \qquad x_{1/2} = \pm\sqrt{-\frac{a_3}{z - a_1}}.$$

Hier ist natürlich nur die positive Lösung von Interesse, da sowohl Persistenz– als auch Dominanzwerte nicht negativ sind. Für die Wahl von z hat sich

$$z = -\frac{\#H^1_{\mathcal{P}}}{\max P - \min P} \quad \text{bzw.} \quad z = -\frac{\#H^1_{\mathcal{D}}}{\max D - \min D}$$

als geeigneter Wert erwiesen. In Abbildung 4.15 sind die Histogramme zum Analysespline $s_m\boldsymbol{c}$ des Testprogramms aus Abbildung 4.8(a) mit entsprechenden Schwellenwerten abgebildet.

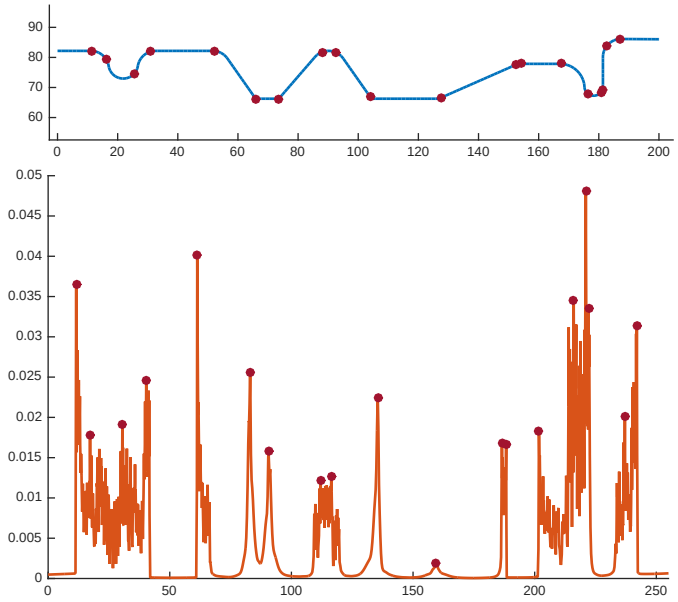


Abb. 4.16: Bahn und Analysespline zum Testprogramm mit markierten Merkmalspunkten.

Mit Hilfe der so bestimmten Schwellenwerte kann entschieden werden, welche Maxima des Analysesplines Merkmalspunkte in der Fräsbahn beschreiben. Abbildung 4.16 zeigt das Testprogramm und den dazugehörigen Analysespline mit Markierungen an den Stellen, die als Merkmalspunkte identifiziert wurden. Es ist gut zu erkennen, dass alle Unstetigkeit in der Fräsbahn erkannt wurden. Zudem werden die Stellen der Fräsbahn mit Krümmungswechsel bzw. Sprüngen in der zweiten Ableitung identifiziert.

4.3 Anwendung und Korrektur

Als Abschluß dieses Kapitels wird im Folgenden beispielhaft das in Abschnitt 4.2 beschriebene Verfahren zur Merkmalserkennung an einem Teileprogramm zu der aus Abschnitt 2.2 bekannten Geometrie durchgeführt. Anschließend wird das Teileprogramm an den identifizierten Problemstellen mit Hilfe von zusätzlichen Geometrieinformationen aus einem CAD–System korrigiert. Es wird beispielhaft gezeigt wie die Berechnung solcher Korrekturpunkte für einen speziellen Fräser bzw. Fräsergeometrie durchgeführt werden kann.

4.3.1 Identifizierung der Problemstellen

Im vorangegangenen Abschnitt wurde beschrieben wie mittels einer Spline–Multiresolution eine Merkmalserkennung für Fräsbahnen durchgeführt werden kann. Diese wird auf das Teileprogramm (Abbildung 4.17) zur bekannten Geometrie aus Kapitel 2 angewandt. In Abbildung 4.18 sieht man eine einzelne Bahn des Teileprogramms mit dazugehörigem Analysespline. Zusätzlich sind die Problemstellen auf dem Teileprogramm markiert. Alle erkannten Problemstellen sind über der Geometrie in Abbildung 4.19 zu sehen. In einem CAD–System sind in der Regel die Kanten des Werkstücks als Flächengrenzen oder Schnittkurven bekannt. Somit kann jeder Merkmalspunkt, der nahe genug an einer entsprechenden Kante liegt mit dieser identifiziert werden. Liegen an einer Kante genug Merkmalspunkte, werden alle Trajektorien der Fräsbahn, die über dieser Kante liegen, korrigiert.

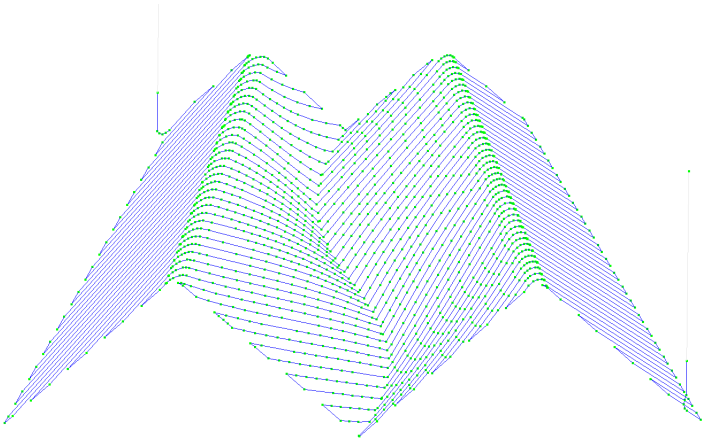


Abb. 4.17: Teileprogramm zur Geometrie aus Abbildung 4.19.

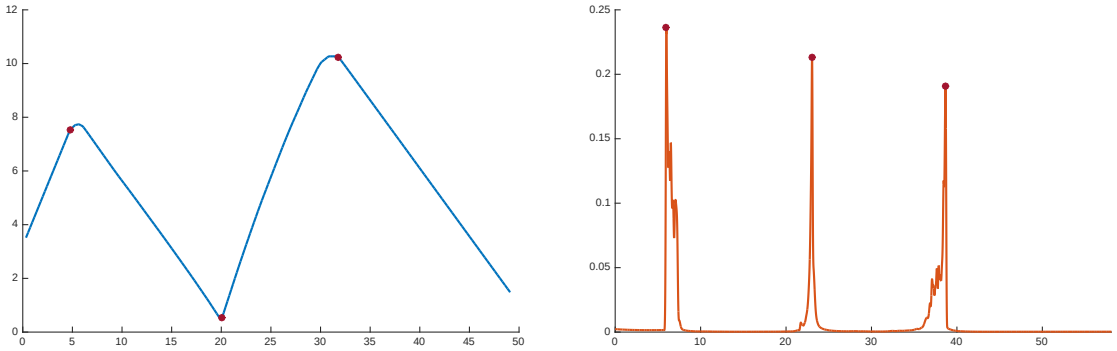


Abb. 4.18: Einzelne Bahn und dazugehöriger Analysespline zur Geometrie aus Abbildung 4.19.

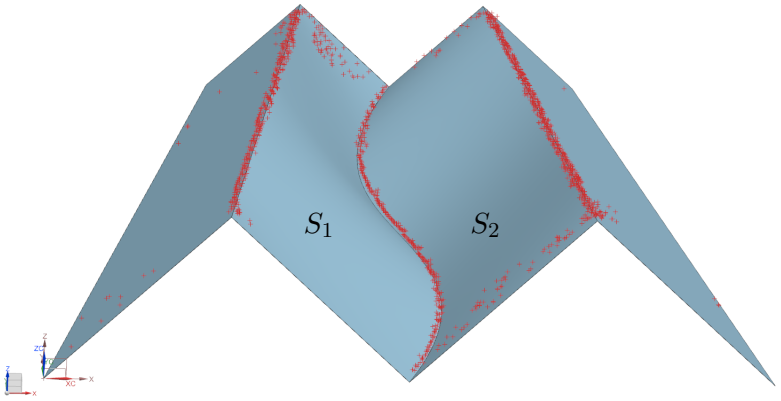


Abb. 4.19: Geometrie mit erkannten Merkmalspunkten zum Teileprogramm aus Abbildung 4.17.

4.3.2 Berechnung der Korrekturpunkte

Um einen Korrekturpunkt zu einem Teileprogramm zu berechnen, sind unterschiedliche Faktoren wichtig. Zunächst muss die Geometrie expliziet bekannt sein. Dies ist in einem CAD–Programm in der Regel der Fall. Freiformflächen liegen zudem oft als Tensorproduktsplineflächen (siehe Definition 3.33) vor. Außerdem ist es wichtig, welcher Punkt des Werkzeugs als Bezugspunkt zur Angabe der Fräsbahn heran gezogen wird. Die gängigsten Angaben hierfür sind der *Kontaktpunkt* und der sogenannte *Tool–Center–Point* (TCP, Werkzeugmittelpunkt). Die Angabe der Fräsbahn über den Kontaktpunkt beschreibt den Punkt, an dem das Werkzeug Kontakt mit der Sollfläche der zu fertigenden Geometrie hat. Diese Angabe ist nicht immer eindeutig, da es zu mehrfachen Kontaktpunkten kommen kann. Der TCP wiederum ist ein fester Bezugspunkt auf der Werkzeugoberfläche. In der Regel ist hier der gedachte Durchstoßpunkt der Werkzeugachse mit der Werkzeugoberfläche gemeint. Des Weiteren spielt die Fräsergeometrie für die Berechnung der Korrekturpunkte eine entscheidende Rolle. Für jeden Fräsertypen muss eine extra Berechnungsstrategie aufgestellt werden. Die in der Praxis wichtigsten Fräsertypen sind:

- Der *Zylinder–* oder auch *Torusfräser*, dieser wird in der Regel bei den groben Schruppvorgängen (Vorbearbeitung) oder auch beim Umfangfräsen eingesetzt.
- Der *Kugelfräser*, dieser findet in der Regel bei den feinen Schlichtprozessen Anwendung.

In Abbildung 4.20 sind die Querschnitte eines Zyllinder– und Kugelfräsers zu sehen. Der TCP ist jeweils rot markiert. Die Oberfläche dieser rotationssymmetrischen Fräser kann durch

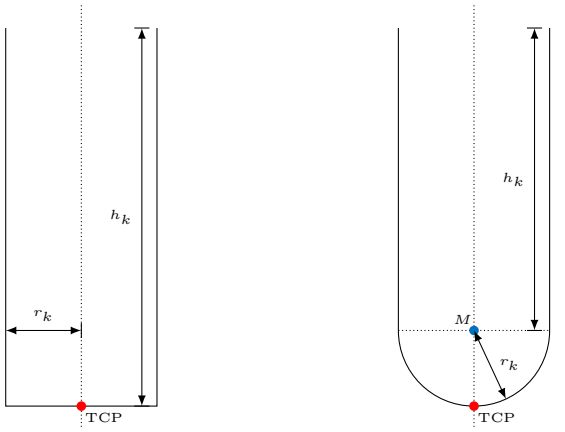
$$O_Z(q,\varphi,s) = \begin{pmatrix} q\, r_z \cos(\varphi) \\ q\, r_z \sin(\varphi) \\ s\, h_z \end{pmatrix} \qquad \text{bzw.} \qquad O_K(\varphi,\theta,s) = \begin{pmatrix} r_k \cos(\varphi)\sin(\theta) \\ r_k \sin(\varphi)\cos(\theta) \\ s\, h_k - r_k \cos(\theta) \end{pmatrix} \tag{4.37}$$

beschrieben werden, wobei O_Z die Oberfläche des Zylinderfräsers und O_K die Oberfläche des Kugelfräsers beschreibt. Für O_Z gilt entweder, dass $s = 0$ fixiert ist und $q \in [0,1]$ oder $q = 0$ fixiert ist und $s \in [0,1]$. Der Winkel φ durchläuft in beiden Fällen stets den gesamten Parameterbereich $[0, 2\pi]$. Hierdurch wird die Stirnseite bzw. die Mantelfläche des Fräsers beschrieben. Für O_K dürfen niemals die beiden Variablen s und θ ihre Parameterbereiche gleichzeitig durchlaufen. Entweder ist $s = 0$ fixiert und $\theta \in [0,\pi]$ sowie $\varphi \in [0, 2\pi]$ oder $\theta = \frac{\pi}{2}$ ist fixiert und $s \in [0,1]$ sowie $\varphi \in [0, 2\pi]$. Hierdurch wird entweder die Halbkugel oder der Schaft des Kugelfräsers beschrieben. Beide Oberflächenbeschreibungen sind Beschreibungen zu einer orthonormierten Basis mit Ursprung im TCP. Eine Herleitung der Formeln (4.37) findet man zum Beispiel in [29].

Damit die Positionierung eines Fräsers bestimmt werden kann, müssen einige Bedingungen eingehalten werden.

- Die Sollfläche darf nicht verletzt werden. Das heißt, der Fräser darf die Oberfläche des Werkstücks nur berühren und nicht schneiden.
- Die Verfahrbewegung bzw. das Fräsmuster, welches vom CAM–System vorgegeben wird, soll eingehalten werden.

Im folgenden Beispiel wird die gewünschte Geometrie mit einem Kugelfräser in einem mäandrischen Fräsmuster (Zickzack) abgefahren. Unter einer solchen Frässtrategie versteht man das Abfahren der Geometrie in parallel verlaufenden Bahnen. Am Ende einer Bahn versetzt das Werkzeug so, dass die benachbarte Bahn, die noch nicht abgefahren wurde, in entgegengesetzter Richtung abgefahren werden kann. Das heißt, ein Korrekturpunkt liegt in der gleichen (Hyper-)



(a) Zylinderfräser (b) Kugelfräser

Abb. 4.20: Querschnitte verschiedener Fräsertypen.

Ebene wie die Punkte aus der aktuell abgefahrenen Bahn. Im vorliegenden Beispiel sollen die Fräsbahnen über einer innen liegenden Ecke korrigiert werden. Der gesuchte Korrekturpunkt soll den Fräser so nahe wie möglich an die Ecke bringen ohne dabei die gewünschte Kontur zu verletzen oder von der aktuellen Bahn abzuweichen. Aus diesen Bedingungen lässt sich ein nichtlineares Gleichungssystem aufstellen. Hierfür ist es zunächst notwendig die Ebene zu bestimmen, in der sich die aktuelle Fräsbahn befindet. Anschließend kann mit Hilfe dieser Ebene und den Bedingungen, dass der Kugelfräser die beiden angrenzenden Flächen nur berühren darf, das gewünschte Gleichungssystem aufgestellt werden. Hier ist insbesondere darauf zu achten, dass man das zunächst als zweidimensional wirkende Problem der Berechnung eines Korrekturpunktes in einer Ebene nicht als solches behandelt. Würde man das Problem der Berechnung des neuen Punktes nur in der Bahnebene betrachten und darauf achten, dass der Fräser in dieser Ebene keine Kollision mit der Sollfläche fährt, kann es passieren, dass links oder rechts neben der Bahn durch die räumliche Ausdehnung des Fräsers eine Kollision mit der Sollfläche gefahren wird.

Berechnung der Ausgleichsebene

Da der Korrekturpunkt in der gleichen Ebene, der *Bahnebene*, liegen soll, ebenso wie die restlichen Punkte der aktuellen Bahn, muss zunächst diese Ebene bestimmt werden. Die Punkte der Bahn unterliegen numerischen Ungenauigkeiten und befinden sich somit nicht exakt in einer Ebene. Es wird eine Ausgleichsebene zu diesen Punkten $p_j \in \mathbb{R}^3$, $j = 1, \ldots, n$, bestimmt. Für die Bahnebene bestimmt man die Koeffizienten der Ebenengleichung E (Normalenform)

$$E(\boldsymbol{x}) = a_1 x_1 + a_2 x_2 + a_3 x_3 + a_4 = 0.$$

Es ist bekannt, dass sich das Bestimmen einer Ausgleichsebene auf ein Eigenwertproblem der Matrix $\widetilde{\boldsymbol{P}} = \boldsymbol{P}^T \boldsymbol{P}$ mit

$$\boldsymbol{P} = \begin{pmatrix} (p_1 - \bar{p})^T \\ \vdots \\ (p_n - \bar{p})^T \end{pmatrix}, \qquad \text{mit} \qquad \bar{p} := \frac{1}{n} \sum_{j=1}^{n} p_j,$$

zurückführen lässt. Die Koeffizienten $\tilde{\boldsymbol{a}} = (a_1, a_2, a_3)^T$ der Ausgleichsebene E entsprechen dem Eigenvektor zum kleinsten Eigenwert von $\widetilde{\boldsymbol{P}}$. Somit gilt

$$\tilde{\boldsymbol{P}} \tilde{\boldsymbol{a}} = \lambda \tilde{\boldsymbol{a}}, \qquad \lambda \in \mathbb{R}.$$

Der letzte Koeffizient a_4 von E kann durch

$$a_4 = \tilde{\boldsymbol{a}}^T \bar{p}$$

bestimmt werden. Falls die Matrix $\boldsymbol{P}$ schlecht konditioniert ist, ist die Konditionszahl für $\widetilde{\boldsymbol{P}}$ quadratisch schlechter, da sie durch Quadrierung der Matrix $\boldsymbol{P}$ entsteht, siehe [28]. Es ist demnach numerisch stabiler die Eigenwerte von $\widetilde{\boldsymbol{P}}$ durch eine Singulärwertzerlegung von $\boldsymbol{P}$ in $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ zu bestimmen, da die Einträge auf der Hauptdiagonalen von $\boldsymbol{\Sigma}$ den Wurzeln der Eigenwerte von $\boldsymbol{P}^T\boldsymbol{P} = \widetilde{\boldsymbol{P}}$ sowie die Spalten der orthogonalen Matrix $\boldsymbol{V}$ den Eigenvektoren von $\widetilde{\boldsymbol{P}}$ entsprechen, siehe [26]. Dies kann leicht durch die beiden folgenden Gleichungen nachvollzogen werden. Sei hierfür $\boldsymbol{v}_j$ der j–te Spaltenvektor aus $\boldsymbol{V}$, $\boldsymbol{e}_j$ der j–te Einheitsvektor und σ_j das j–te Diagonalelement der Matrix $\boldsymbol{\Sigma}$. Dann gilt

$$\begin{aligned}\widetilde{\boldsymbol{P}} = \boldsymbol{P}^T\boldsymbol{P} &= \left(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\right)^T \left(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\right) \\ &= \boldsymbol{V}\boldsymbol{\Sigma}^T\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \\ &= \boldsymbol{V}\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}\boldsymbol{V}^T = \boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^T\end{aligned}$$

und somit

$$\widetilde{\boldsymbol{P}}\boldsymbol{v}_j = \boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^T\boldsymbol{v}_j = \boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{e}_j = \boldsymbol{V}\sigma_j^2\boldsymbol{e}_j = \boldsymbol{v}_j\sigma_j^2 = \sigma_j^2\boldsymbol{v}_j.$$

Für die ersten drei Koeffizienten der Ausgleichsebene $\tilde{\boldsymbol{a}} = (a_1, a_2, a_3)^T$ ermittelt man nunmehr den entsprechenden Spaltenvektor aus $\boldsymbol{V}$ zum kleinsten Singulärwert σ_j und normiert diesen. Die implizite Darstellung der Ebene ist für weiter Berechnungen nicht geeignet und wird deswegen in ihre Parameterdarstellung

$$E(s,t) = \boldsymbol{o} + s\,\boldsymbol{r}_1 + t\,\boldsymbol{r}_2, \qquad \boldsymbol{o}, \boldsymbol{r}_1, \boldsymbol{r}_2 \in \mathbb{R}^3,$$

umgeformt.

Berechnung des Korrekturpunktes

Im Teileprogramm zur Beispielbahn wird als Bezugspunkt der TCP angegeben. Die folgenden Berechnungen vereinfachen sich jedoch, wenn man sich auf den Kugelmittelpunkt M des Kugelfräsers bezieht, siehe Abbildung 4.20(b), und anschließend den Versatz zum TCP zurückrechnet. Es wird nun angenommen, dass dieser Mittelpunkt ebenfalls in der Bahnebene liegt. Solange der Fräser mit seiner Orientierung (Werkzeugachse) in der Bahnebene liegt, ist diese Annahme korrekt. Sollte dies nicht der Fall sein, muss zunächst anhand der Anstellung des Fräsers eine zur Bahnebene parallele Ebene bestimmt werden, in der sich der Kugelmittelpunkt befinden kann.

Durch die Rückfrage an ein CAD–System können die relevanten Flächen S_1 und S_2 (siehe Abbildung 4.19) als Tensorprodukt–Splines $S_{1\boldsymbol{\mu}_1}\boldsymbol{d}((u_1, v_1) \mid T)$ und $S_{2\boldsymbol{\mu}_2}\boldsymbol{d}((u_2, v_2) \mid T)$ angegeben werden. Ein Korrekturpunkt, der den Fräser so nahe wie möglich an die Ecke zwischen S_1 und S_2 heranbringt, hat die Eigenschaft, dass der Fräser sowohl Kontakt zur Fläche S_1 wie auch zur Fläche S_2 hat. Das bedeutet, dass der Kugelmittelpunkt M des Fräsers sowohl zur Fläche S_1 wie auch zur Fläche S_2 den Abstand r_k hat. Unter der Voraussetzung, dass $M \in E(s,t)$ als Element der Bahnebene $E(s,t)$ aufgefasst werden kann, können die ersten Gleichungen des gesuchten Gleichungssystem aufgestellt werden:

$$\begin{aligned}\left\| S_{1\boldsymbol{\mu}_1}\boldsymbol{d}(u_1, v_1) - E(s,t) \right\|_2^2 - r_k = 0, \\ \left\| S_{2\boldsymbol{\mu}_2}\boldsymbol{d}(u_2, v_2) - E(s,t) \right\|_2^2 - r_k = 0.\end{aligned} \tag{4.38}$$

Aus der Einschränkung, dass der Fräser die Flächen S_1 und S_2 nicht schneiden darf, also nur tangential berührt, folgen die nächsten Gleichungen. Denn der Vektor vom Berührpunkt des Fräsers auf S_1 bzw. S_2 zum Kugelmittelpunkt M steht senkrecht auf den partiellen Ableitungen der Flächen an dieser Stelle:

$$\begin{aligned} \Big(S_{1\boldsymbol{\mu}_1}\boldsymbol{d}(u_1,v_1) - E(s,t)\Big)^T \ \frac{\partial}{\partial u_1} S_{1\boldsymbol{\mu}_1}\boldsymbol{d}(u_1,v_1) = 0, \\ \Big(S_{1\boldsymbol{\mu}_1}\boldsymbol{d}(u_1,v_1) - E(s,t)\Big)^T \ \frac{\partial}{\partial v_1} S_{1\boldsymbol{\mu}_1}\boldsymbol{d}(u_1,v_1) = 0 \end{aligned} \tag{4.39}$$

und

$$\begin{aligned} \Big(S_{2\boldsymbol{\mu}_2}\boldsymbol{d}(u_2,v_2) - E(s,t)\Big)^T \ \frac{\partial}{\partial u_2} S_{2\boldsymbol{\mu}_2}\boldsymbol{d}(u_2,v_2) = 0, \\ \Big(S_{2\boldsymbol{\mu}_2}\boldsymbol{d}(u_2,v_2) - E(s,t)\Big)^T \ \frac{\partial}{\partial v_2} S_{2\boldsymbol{\mu}_2}\boldsymbol{d}(u_2,v_2) = 0. \end{aligned} \tag{4.40}$$

Die sechs Gleichungen aus (4.38), (4.39) und (4.40) gilt es nun für die Parameterwerte s, t, u_1, v_1, u_2 und v_2 simultan zu lösen. Hierfür setzt man die Funktion $F : \mathbb{R}^6 \to \mathbb{R}^6$ mit

$$F(s,t,u_1,v_1,u_2,v_2) := \begin{pmatrix} \Big\|S_{1\boldsymbol{\mu}_1}\boldsymbol{d}(u_1,v_1) - E(s,t)\Big\|_2^2 - r_k \\ \Big\|S_{2\boldsymbol{\mu}_2}\boldsymbol{d}(u_2,v_2) - E(s,t)\Big\|_2^2 - r_k \\ \Big(S_{1\boldsymbol{\mu}_1}\boldsymbol{d}(u_1,v_1) - E(s,t)\Big)^T \ \frac{\partial}{\partial u_1} S_{1\boldsymbol{\mu}_1}\boldsymbol{d}(u_1,v_1) \\ \Big(S_{1\boldsymbol{\mu}_1}\boldsymbol{d}(u_1,v_1) - E(s,t)\Big)^T \ \frac{\partial}{\partial v_1} S_{1\boldsymbol{\mu}_1}\boldsymbol{d}(u_1,v_1) \\ \Big(S_{2\boldsymbol{\mu}_2}\boldsymbol{d}(u_2,v_2) - E(s,t)\Big)^T \ \frac{\partial}{\partial u_2} S_{2\boldsymbol{\mu}_2}\boldsymbol{d}(u_2,v_2) \\ \Big(S_{2\boldsymbol{\mu}_2}\boldsymbol{d}(u_2,v_2) - E(s,t)\Big)^T \ \frac{\partial}{\partial v_2} S_{2\boldsymbol{\mu}_2}\boldsymbol{d}(u_2,v_2) \end{pmatrix} \tag{4.41}$$

an und sucht deren Nullstellen. Um diese zu finden, kann das Newton–Verfahren verwendet werden. Der Iterationsschritt des Newton–Verfahrens im mehrdimensionalen Fall ist dem univariaten Fall sehr ähnlich. Da die Ableitung im Mehrdimensionalen die Jacobi–Matrix ist, wird im Iterationsschritt nicht durch die Ableitung geteilt, sondern mit der Inversen multipliziert. Ein Iterationsschritt für $\boldsymbol{x} = (s,t,u_1,v_1,u_2,v_2)^T$ sieht wie folgt aus:

$$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - J_F(\boldsymbol{x}_n)^{-1} F(\boldsymbol{x}_n),$$

wobei J_F die Jacobi–Matrix der Funktion F ist. Für J_F gilt:

$$J_F(\boldsymbol{x}) = \begin{pmatrix} \nabla F_1(\boldsymbol{x})^T \\ \vdots \\ \nabla F_6(\boldsymbol{x})^T \end{pmatrix}, \qquad \nabla F_j(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial}{\partial s} F_j(\boldsymbol{x}) \\ \frac{\partial}{\partial t} F_j(\boldsymbol{x}) \\ \frac{\partial}{\partial u_1} F_j(\boldsymbol{x}) \\ \frac{\partial}{\partial v_1} F_j(\boldsymbol{x}) \\ \frac{\partial}{\partial u_2} F_j(\boldsymbol{x}) \\ \frac{\partial}{\partial v_2} F_j(\boldsymbol{x}) \end{pmatrix}, \qquad j = 1,\ldots,6,$$

wobei mit dem Index j die j–te Komponente der Funktion F beschrieben wird. Die partiellen

Ableitungen von F sehen wie folgt aus:

$$\begin{aligned}\frac{\partial}{\partial s}F_1(\boldsymbol{x}) &= \frac{\partial}{\partial s}\sum_{j=1}^{3}\Big(S_{1,j_{\boldsymbol{\mu}}}\boldsymbol{d}(u_1,v_1)-E_j(s,t)\Big)^2-r_k^2\\ &= -2\left(\frac{\partial}{\partial s}E(s,t)\right)^T\Big(S_{1_{\boldsymbol{\mu}_1}}\boldsymbol{d}(u_1,v_1)-E(s,t)\Big),\end{aligned}$$

$$\frac{\partial}{\partial u_1}F_1(\boldsymbol{x}) = 2\left(\frac{\partial}{\partial u_1}S_{1_{\boldsymbol{\mu}_1}}\boldsymbol{d}(u_1,v_1)\right)^T\Big(S_{1_{\boldsymbol{\mu}_1}}\boldsymbol{d}(u_1,v_1)-E(s,t)\Big),$$

$$\frac{\partial}{\partial u_2}F_1(\boldsymbol{x}) = 0 = \frac{\partial}{\partial v_2}F_1(\boldsymbol{x}),$$

sowie

$$\begin{aligned}\frac{\partial}{\partial s}F_3(\boldsymbol{x}) &= \frac{\partial}{\partial s}\left(\Big(S_{1_{\boldsymbol{\mu}_1}}\boldsymbol{d}(u_1,v_1)-E(s,t)\Big)^T\frac{\partial}{\partial u_1}S_{1_{\boldsymbol{\mu}_1}}\boldsymbol{d}(u_1,v_1)\right)\\ &= -\frac{\partial}{\partial s}E(s,t)^T\frac{\partial}{\partial u_1}S_{1_{\boldsymbol{\mu}_1}}\boldsymbol{d}(u_1,v_1)\end{aligned}$$

$$\begin{aligned}\frac{\partial}{\partial u_1}F_3(\boldsymbol{x}) &= \frac{\partial}{\partial u_1}\left(\Big(S_{1_{\boldsymbol{\mu}_1}}\boldsymbol{d}(u_1,v_1)-E(s,t)\Big)^T\frac{\partial}{\partial u_1}S_{1_{\boldsymbol{\mu}_1}}\boldsymbol{d}(u_1,v_1)\right)\\ &= \frac{\partial}{\partial u_1}S_{1_{\boldsymbol{\mu}_1}}\boldsymbol{d}(u_1,v_1)^T\frac{\partial}{\partial u_1}S_{1_{\boldsymbol{\mu}_1}}\boldsymbol{d}(u_1,v_1)+\Big(S_{1_{\boldsymbol{\mu}_1}}\boldsymbol{d}(u_1,v_1)-E(s,t)\Big)^T\frac{\partial^2}{\partial u_1^2}S_{1_{\boldsymbol{\mu}_1}}\boldsymbol{d}(u_1,v_1)\end{aligned}$$

$$\frac{\partial}{\partial u_2}F_3(\boldsymbol{x}) = 0 = \frac{\partial}{\partial v_2}F_3(\boldsymbol{x}).$$

Alle weiteren nicht aufgeführten partiellen Ableitungen von F berechnet man analog zu den hier gezeigten. Als Startwert des Newton–Verfahrens wählt man den Punkt p_j aus der aktuellen Bahn, der der Ecke am nächsten liegt. Die Parameterwerte u_1, v_1, u_2 sowie v_2 der Tensorsplineflächen und die Parameterwerte s und t der Bahnebene können mittels eines „nähester–Punkt“–Algorithmus (Newton–Bisektions–Hybrid, siehe zum Beispiel [51]) bzw. eines linearen Gleichungssystems ermittelt werden.

Der folgende Algorithmus fasst die Schritte zur Berechnung eines Korrekturpunktes an einer innen liegenden Ecke für einen Kugelfräser zusammen.

Algorithmus 4.10 (Korrekturpunkt)
Eingabe:

- *Bahnpunkte p_j, $j = 1,\ldots,n$*
- *Tensorprodukt–Splineflächen $S_{1_{\boldsymbol{\mu}_1}}\boldsymbol{d}$ und $S_{2_{\boldsymbol{\mu}_2}}\boldsymbol{d}$*
- *Kugelfräsergeometrie r_k*
- *Genauigkeit ε*

Prozedur:

1. **Initialisiere***:*
   - *berechne Bahnebene* $E(s,t)$ *zu* p_j, $j = 1, \ldots, n$
   - *bestimme Startwert des Newton–Verfahrens* $\boldsymbol{x}_0$, *nächstgelegener Punkt aus* $\{p_j : j = 1, \ldots, n\}$ *zur Ecke mit entsprechenden Parameterwerten* $(s, t, u_1, v_1, u_2, v_2)$
   - $\boldsymbol{F} := F(\boldsymbol{x}_0)$
2. **WHILE** $\|\boldsymbol{F}\|_2 > \varepsilon$
   - $\boldsymbol{J} := J_F(\boldsymbol{x}_0)$
   - $\boldsymbol{x}_1 := \boldsymbol{x}_0 - \boldsymbol{J}^{-1}\boldsymbol{F}$
   - $\boldsymbol{x}_0 := \boldsymbol{x}_1$
   - $\boldsymbol{F} := F(\boldsymbol{x}_0)$
3. ***Ergebnis:***
   Parameterwerte s_0, t_0 *des Kugelmittelpunkts* $M = E(s_0, t_0)$ *in der Ebene* $E(s,t)$, *wobei* s_0, t_0 *die ersten beiden Komponenten von* $\boldsymbol{x}_0 = (s_0, t_0, u_{10}, v_{10}, u_{20}, v_{20})^T$ *sind.*

Ausgehend von dem nach Algorithmus 4.10 berechneten Kugelmittelpunkt M rechnet man in Richtung der Werkzeugachse mit Hilfe des Radius r_k zurück auf den TCP. Dieser Punkt kann dann im entsprechenden Satz im Teileprogramm eingefügt werden. Im Algorithmus 4.10 wird zur Vereinfachung der Notation mit der Inversen der Jacobi–Matrix gearbeitet. Es ist numerisch effizienter und stabiler an dieser Stelle das entsprechende Gleichungssystem ($\boldsymbol{J_F} x = \boldsymbol{F}$) zu lösen.

Abbildung 4.21 zeigt einen Vergleich der Fräsbahnen mit und ohne Korrekturpunkte sowie die Frässimulation dieser beiden Teileprogramme. Die Simulation wurde mit VisuTool, einem Siemens internen Software–Tool erzeugt.

Abschließend sei nochmals erwähnt, dass für jede Frässtrategie und Fräsergeometrie die Berechnung der Korrekturpunkte neu anzugeben ist. Im vorgestellten Beispiel konnte die Verwendung der expliziten Fräsergeometrie (4.37) umgangen werden. Dies ist jedoch nicht immer möglich.

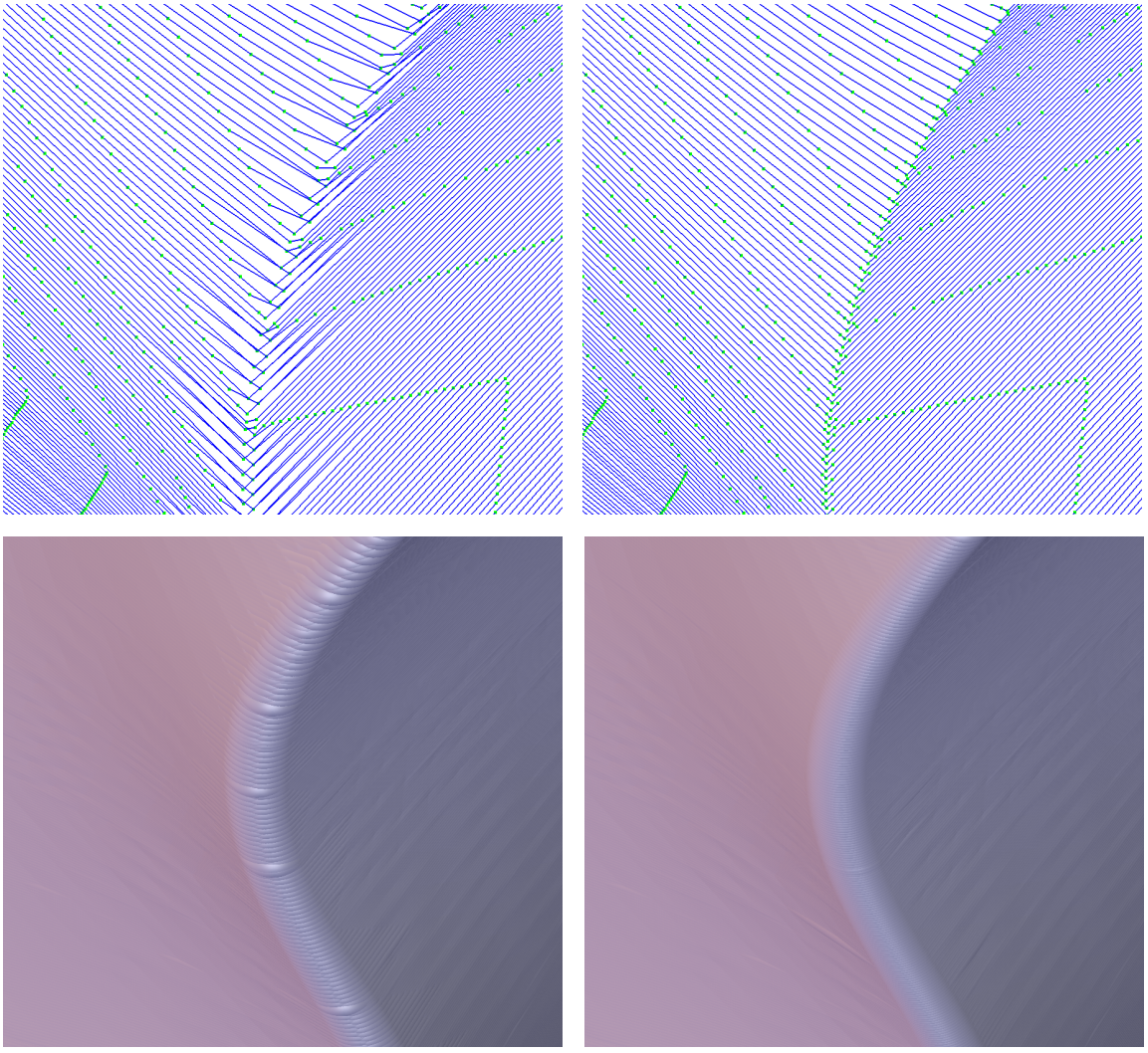


Abb. 4.21: Fräsbahnen und Simulation vor (links) und nach (rechts) der Korrektur.

5 Analyse der Orientierung

In Kapitel 4 wurde ausschließlich auf die Analyse der *Fräsbahn* eingegangen. Mit der Möglichkeit in heutigen CAD–Systemen immer leichter Freiformflächen zu entwerfen, steigt die Anforderung im Bereich der Fertigung enorm an, denn solche Freiformflächen lassen sich oft nur noch mit Hilfe einer Fünf–Achs–Werkzeugmaschine fertigen. Mit Fünf–Achs–Maschinen ist es möglich, die Orientierung eines Fräsers während der Bearbeitung bzw. Abarbeitung einer Fräsbahn zu ändern. Dies geschieht mit den sogenannten *Rundachsen*. In einem Teileprogramm wird hierfür zusätzlich zur Fräserposition eine Orientierung des Fräsers an dieser Stelle angegeben, siehe Abschnitt 2.1. Eine numerische Steuerung muss somit in der Lage sein, nicht nur die Positionen sondern auch die Orientierung eines Fräsers zu interpolieren bzw. zu approximieren. In diesem Kapitel wird eine Möglichkeit präsentiert, die Orientierungsdaten eines Teileprogramms mittels eines Quaternionen–Splines zu approximieren. Ziel hierbei war es, das in Kapitel 4 vorgestellte Verfahren zur Detektion von Problemstellen mittels eines MRA Ansatzes auch auf diese Kurven anwenden zu können. Des Weiteren konnte eine Konvexe–Hülle–Eigenschaft als wichtige geometrische Eigenschaft für diese Kurven bewiesen werden.

Die Orientierung eines Fräsers wird als Vektor vom *Tool–Center–Point* (TCP) zur Werkzeugaufnahme angegeben. Für die Orientierung ist nur die *Richtung* des Vektors und nicht seine *Länge* von Interesse. Daher wird der Vektor stets zu eins normiert. Wird die Orientierung in einem Teileprogramm mittels eines Vektors angegeben, so ist dieser in der Regel normiert. Fasst man den TCP eines Fräsers als Ursprung eines Koordinatensystems auf, beschreiben die (normierten) Orientierungsvektoren Punkte auf der Sphäre $\mathbb{S}^2$. Um einen kontinuierlichen Orientierungsverlauf angeben zu können, ist ein Approximant notwendig, der auf der Oberfläche der Einheitskugel lebt.

5.1 Splines auf der Sphäre $\mathbb{S}^2$

Als ersten Ansatz Orientierungsvektoren auf der Sphäre zu interpolieren, könnte man einen Interpolanten $f(t)$ im euklidischen Raum $\mathbb{R}^3$ berechnen. Es gibt jedoch keine Garantie, dass der Interpolant zwischen den Interpolationspunkten auf der Sphäre bleibt. Dies könnte jedoch durch Normierung der Funktion

$$f(t) = \frac{f(t)}{\|f(t)\|_2}$$

erzwungen werden. Durch die Normierung können allerdings unerwünschte Effekte wie Schleifen oder Ecken auftreten [40]. Um Splines auf der Sphäre zu erzeugen, ist es hilfreich sich zu überlegen, wie man Splines üblicherweise im euklidischen Raum definiert. Dies kann auf zwei Arten geschehen: zum einen algebraisch mit Hilfe ihrer Basisfunktionen (Definition 3.12), zum anderen geometrisch durch rekursive lineare Interpolation (Algorithmus 3.19, [2]). Ersetzt man im geometrischen Ansatz die lineare Interpolation durch die *sphärische linearen Interpolation* (*„slerp“, spherical linear interpolation*) dem Äquivalent auf der Sphäre, erhält man eine einfache Möglichkeit, zu gegebenen Kontrollpunkten glatte Kurven auf der Sphäre zu erzeugen. Hierbei wird in Formel (3.20) aus dem de Boor–Algorithmus 3.19 die lineare Interpolation der Kontrollpunkte

durch eine Großkreisinterpolation ersetzt. Diese sieht für $d_1, d_2 \in \mathbb{S}^2$ wie folgt aus:

$$\frac{\sin((1-t)\varphi)}{\sin(\varphi)} d_1 + \frac{\sin((t)\varphi)}{\sin(\varphi)} d_2, \qquad t \in [0,1], \qquad \varphi = \cos^{-1}(\, d_1^T d_2\,) \tag{5.1}$$

und entspricht der kürzeren der beiden Großkreisverbindungen auf der Sphäre $\mathbb{S}^2$. Im Folgenden ist stets die kürzere Großkreisverbindung zweier Punkte auf der Kugel gemeint, wenn von sphärischer linearer Interpolation oder Großkreisverbindung die Rede ist. Sind die beiden Punkte antipodal, so ist die Großkreisverbindung nicht mehr eindeutig. Die Formel (5.1) der sphärischen linearen Interpolation von Glenn Davis wurde unter anderem von Shoemake [49] zur Konstruktion von Quaternionenkurven verwendet. Ein großer Nachteil dieser Konstruktion von Splines ist, dass Interpolationsprobleme in einem nichtlinearen Gleichungssystem enden, dessen Lösung nur noch approximativ bestimmt werden kann [23, Kap. 29]. Zudem können Ableitungen nicht mehr als Nebenprodukt des Auswertungs–Algorithmus berechnet werden wie es beim de Boor–Algorithmus der Fall ist. Das Knoteneinfügen durch analoges Anpassen des Boehm–Algorithmus 3.27 ist zudem nicht mehr ohne Veränderung der Kurve möglich. Dies ist allerdings eine Grundvoraussetzung des, in Kapitel 4 eingeführten, MRA Verfahrens. Somit scheidet eine solche Konstruktion aus.

Andere Ansätze, wie in [32, 33], zur Generierung von Rotationskurven mit Hilfe von Einheitsquaternionen, für die es leicht zu bestimmende Ableitungen gibt, basieren auf einer kumulierten Basisdarstellung der B–Splines

$$\hat{N}_k^m(\,\cdot \mid T_{n,m}) = \sum_{j=k}^{n} N_j^m(\,\cdot \mid T_{n,m}), \qquad k = 1, \ldots, n,$$

und nutzen eine Formel von Shoemake [49] für die sphärische lineare Interpolation auf $\mathbb{S}^3$ aus. Für $\boldsymbol{q}_0, \ldots, \boldsymbol{q}_n \in \mathbb{S}^3$ gilt:

$$f(t) = \boldsymbol{q}_0^{\hat{N}_1^m(t|T)} \prod_{j=1}^{n} \exp\left(\hat{N}_j^m(t \mid T) \log(\boldsymbol{q}_{j-1}^{-1}\boldsymbol{q}_j)\right),$$

wobei die Funktionen exp und log den Funktionen aus den folgenden Definitionen 5.2 bzw. 5.3 entsprechen. Diese Art von Kurven ist für den vorgestellten MRA Ansatz ebenfalls nicht geeignet, da sie nicht durch eine Linearkombination der Basisfunktionen dargestellt werden. Eine Übersicht über verschiedene Möglichkeiten Quaternionenkurven zu erzeugen, findet man unter anderem in [14].

Die in diesem Abschnitt vorgestellte Strategie eine Kurve auf der Sphäre zu erzeugen, lässt sich anschaulich wie folgt erklären. Man wählt einen beliebigen Punkt (*Referenzpunkt*) auf der Sphäre $\mathbb{S}^2$ aus und bewegt diesen Punkt durch Rotation in die gewünschten Positionen. Die dabei hinterlassene Spur entspricht der gewünschten Kurve auf der Sphäre. Die jeweiligen Rotationen werden hierbei durch einen *homogenen Quaternionenspline* berechnet. Hierfür bestimmt man die Rotationen, die benötigt werden, um den Referenzpunkt $\boldsymbol{r} \in \mathbb{S}^2$ in die jeweils zu interpolierenden Punkte $\boldsymbol{p}_j \in \mathbb{S}^2$, $j = 1, \ldots, n$, zu drehen. Diese Rotationen können durch Einheitsquaternionen (Definition 3.36) $\boldsymbol{q}_j \in \mathbb{S}^3$, $j = 1, \ldots, n$, mit

$$\boldsymbol{q}_j = \begin{pmatrix} \cos(\theta/2) \\ \sin(\theta/2)\ \boldsymbol{a} \end{pmatrix}, \qquad \boldsymbol{a} = \frac{\boldsymbol{r} \times \boldsymbol{p}_j}{\|\boldsymbol{r} \times \boldsymbol{p}_j\|_2}, \qquad \theta = \cos^{-1}(\boldsymbol{r}^T \boldsymbol{p}_j),$$

für $\pm\boldsymbol{r} \neq \boldsymbol{p}_j$ dargestellt werden. Die Quaternionen $\boldsymbol{q}_j$ sind Repräsentanten der zu interpolierenden bzw. approximierenden Punkte $\boldsymbol{p}_j$, $j = 1, \ldots, n$, im Raum der Quaternionen $\mathbb{H}$. Es wird vereinbart, dass der Referenzpunkt $\boldsymbol{r}$ der Quaternion $(1, 0, 0, 0)^T \in \mathbb{H}$ entspricht. Die Punkte $\boldsymbol{q}_j$, $j = 1, \ldots, n$, im vierdimensionalen Vektorraum $\mathbb{H}$ können nun beliebig mittels eines Splines interpoliert bzw. approximiert werden. Die Koordinaten des entstehenden Splines werden als *homogene Koordinaten* aufgefasst. Das heißt, alle Elemente eines eindimensionalen Unterraums in $\mathbb{H}$ werden miteinander identifiziert. Somit können linear abhängige Quaternionen mit einem bzw. zwei antipodalen Punkten auf $\mathbb{S}^3$ identifiziert werden und beschreiben die gleiche Rotation. Vor der jeweiligen Ausführung einer Drehung nach Definition 3.37 ist eine Normierung der entsprechenden Quaternion notwendig. Für weitere Informationen zu homogenen Koordinaten im Zusammenhang mit Quaternionen sei zum Beispiel auf [23, 49] verwiesen.

5.1.1 Wahl des Referenzpunktes

Im Folgenden seien die n Punkte $\boldsymbol{p}_j$, $j = 1, \ldots, n$, stets die zu interpolierenden bzw. approximierenden Punkte auf der Sphäre $\mathbb{S}^2$, so gewählt, dass sie alle in einer Hemisphäre liegen und mindestens ein Punkt $\boldsymbol{p}_j$ im Inneren dieser Hemisphäre liegt.

Der Referenzpunkt $\boldsymbol{r} \in \mathbb{S}^2$ ist frei wählbar und könnte zum Beispiel auf den ersten zu interpolierenden Punkt festgelegt werden: $\boldsymbol{r} = \boldsymbol{p}_1$. Dies wird nicht durchgeführt, da so dem ersten Punkt unbeabsichtigt eine besondere Rolle verliehen würde.

Eine Gleichberechtigung aller Punkte $\boldsymbol{p}_j$ scheint sinnvoller, was im euklidischen Raum mit dem Ansatz den Mittelwert

$$\bar{\boldsymbol{p}} = \frac{1}{n} \sum_{j=1}^{n} \boldsymbol{p}_j$$

der Punkte $\boldsymbol{p}_j$ als Referenzpunkt zu wählen, realisiert wird. Auf der Sphäre ist es allerdings nicht möglich einen solchen Mittelwert durch eine Linearkombination der Punkte zu bestimmen. Im euklidischen Raum kann der Mittelwert äquivalent durch Minimierung (Least–Squares) der Funktion

$$f(\boldsymbol{t}) = \frac{1}{2n} \sum_{j=1}^{n} \left\| \boldsymbol{t} - \boldsymbol{p}_j \right\|_2^2 \tag{5.2}$$

bestimmt werden. In [7] wird dieser Least–Squares–Ansatz zur Mittelwertberechnung auf die Sphäre übertragen. Hierfür ist zunächst ein Abstandsbegriff nötig. Die kürzeste Verbindung zweier Punkte auf der Sphäre, die *Geodäte*, ist bekanntermaßen die Großkreisverbindung zwischen diesen Punkten. Der minimale Abstand zweier Punkte auf einer Einheitskugel kann demnach als Winkel zwischen den beiden Punkten angegeben werden.

Definition 5.1 *Für $\boldsymbol{q}, \boldsymbol{p} \in \mathbb{S}^2$ ist der* minimale Abstand $\operatorname{dist}_{\mathbb{S}^2}(\boldsymbol{q}, \boldsymbol{p})$ *der beiden Punkte $\boldsymbol{q}$ und $\boldsymbol{p}$ auf der Sphäre $\mathbb{S}^2$ durch*

$$\operatorname{dist}_{\mathbb{S}^2}(\boldsymbol{q}, \boldsymbol{p}) := \cos^{-1}(\boldsymbol{q}^T \boldsymbol{p}) \tag{5.3}$$

definiert.

Hiermit lässt sich ein zu (5.2) analoger Ausdruck für die Mittelwertberechnung auf der Sphäre angeben:

$$f(\boldsymbol{t}) = \frac{1}{2n} \sum_{j=1}^{n} \operatorname{dist}_{\mathbb{S}^2}(\boldsymbol{t}, \boldsymbol{p}_j)^2. \tag{5.4}$$

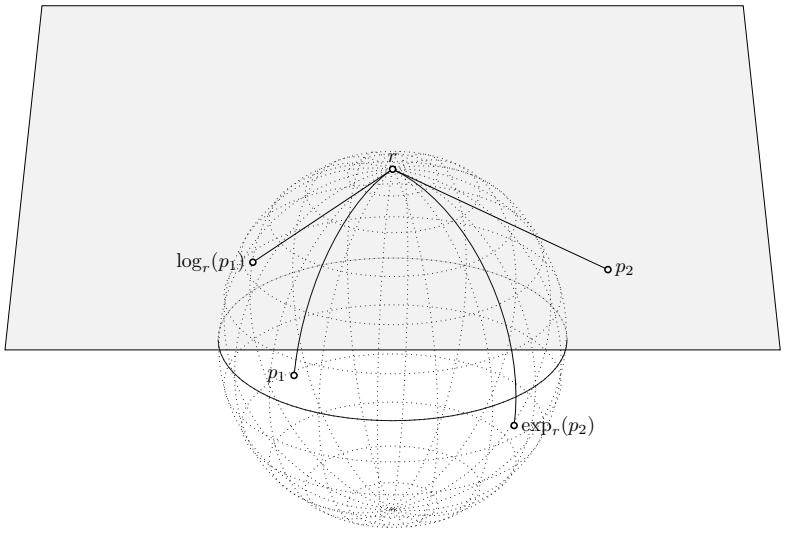


Abb. 5.1: $\log_r$– und $\exp_r$–Abbildungen mit $r = (0, 0, 1)^T$.

Buss und Fillmore geben in [7] einen Algorithmus zur Lösung des Minimierungsproblems $\min_{\boldsymbol{t}} f(\boldsymbol{t})$ an. Hierfür ist zunächst die Definition der Abbildungen $\exp_{\boldsymbol{r}}$ und $\log_{\boldsymbol{r}}$ notwendig. Mit diesen Funktionen kann ein Punkt auf der Sphäre in den Tangentialraum an $\boldsymbol{r} \in \mathbb{S}^2$ abgebildet werden und umgekehrt vom Tangentialraum wieder auf die Sphäre. Zur Vereinfachung der Abbildungen kann ohne Einschränkung angenommen werden, dass der Punkt $\boldsymbol{r} \in \mathbb{S}^d$, an dem der Tangentialraum anliegt, vor der Anwendung der Abbildung immer in den Nordpol der Sphäre gedreht wird und das Ergebnis anschließend wieder in die Ausgangsposition zurück gedreht wird. Abbildung 5.1 veranschaulicht die Funktionen $\exp_{\boldsymbol{r}}$ und $\log_{\boldsymbol{r}}$.

Definition 5.2 *Sei $T_{\boldsymbol{r},\mathbb{S}^d}$ der Tangentialraum von $\mathbb{S}^d$ an $\boldsymbol{r} \in \mathbb{S}^d$ sowie $\boldsymbol{y} = (y_1, \ldots, y_d)^T \in T_{\boldsymbol{r},\mathbb{S}^d}$. Die* Exponential–Abbildung $\exp_{\boldsymbol{r}} : T_{\boldsymbol{r},\mathbb{S}^d} \to \mathbb{S}^d$ *ist durch*

$$\exp_{\boldsymbol{r}}(\boldsymbol{y}) = \begin{cases} \left(y_1 \frac{\sin(\|\boldsymbol{y}\|_2)}{\|\boldsymbol{y}\|_2},\ \ldots,\ y_d \frac{\sin(\|\boldsymbol{y}\|_2)}{\|\boldsymbol{y}\|_2},\ \cos(\|\boldsymbol{y}\|_2)\right)^T, & \boldsymbol{y} \neq (0, \ldots, 0)^T \\ (0, \ldots, 0, 1)^T, & \boldsymbol{y} = (0, \ldots, 0) \end{cases} \tag{5.5}$$

definiert.

Definition 5.3 *Sei $\boldsymbol{x} = (x_1, \ldots, x_{d+1})^T \in \mathbb{S}^d$ und $T_{\boldsymbol{r},\mathbb{S}^d} \subseteq \mathbb{R}^d$ der Tangentialraum von $\mathbb{S}^d$ an $\boldsymbol{r} \in \mathbb{S}^d$, dann ist die* Logarithmus–Abbildung $\log_{\boldsymbol{r}} : \mathbb{S}^d \to T_{\boldsymbol{r},\mathbb{S}^d}$ *durch*

$$\log_{\boldsymbol{r}}(\boldsymbol{x}) = \frac{\theta}{\sin(\theta)} \begin{pmatrix} x_1 \\ \vdots \\ x_d \end{pmatrix}, \qquad \theta = \cos^{-1}(x_{d+1}) \tag{5.6}$$

für $\theta \neq 0$ definiert.

Anschaulich wird bei diesen Abbildungen der Punkt $\boldsymbol{r} \in \mathbb{S}^d$ als Ursprung des Tangentialraums $T_{\boldsymbol{r},\mathbb{S}^d}$ aufgefasst. Somit beschreibt $\|\boldsymbol{y}\|_2$ den Abstand von $\boldsymbol{y} \in T_{\boldsymbol{r},\mathbb{S}^d}$ zu $\boldsymbol{r}$. Sei nun ohne Einschränkung $\boldsymbol{r} = (0, \ldots, 0, 1)^T \in \mathbb{S}^d$. Da die Sphäre einen Radius von 1 hat, kann leicht nachvollzogen werden, dass der Winkel bzw. der sphärische Abstand zwischen $\boldsymbol{r}$ und $\exp_{\boldsymbol{r}}(\boldsymbol{y})$ ebenfalls $\|\boldsymbol{y}\|_2$

entspricht, da

$$\cos^{-1}(\boldsymbol{r}^T \exp_{\boldsymbol{r}}(\boldsymbol{y})) = \cos^{-1}(0 + \cdots + 0 + \cos(\|\boldsymbol{y}\|_2)) = \|\boldsymbol{y}\|_2$$

für $\|\boldsymbol{y}\|_2 < \pi$. Die Abbildung $\exp_{\boldsymbol{r}}$ ist somit eine längenerhaltende Abbildung. Weitere Informationen zu den $\exp_{\boldsymbol{r}}$– und $\log_{\boldsymbol{r}}$–Abbildungen im Zusammenhang mit Quaternionen und Computergrafik findet man zum Beispiel in [14, 33]. Hier kann es durch unterschiedliche Definitionen zu Verwirrungen kommen. Es ist insbesondere darauf zu achten wie die Komponenten der Abbildungen angeordnet sind.

Algorithmus 5.4 (sphärischer Mittelwert (Buss und Fillmore))
Eingabe:

- *Punkte* $\boldsymbol{p}_j \in \mathbb{S}^d$, $d \in \mathbb{N}$, $j = 1, \ldots, n$
- *gewünschte Genauigkeit* ε

Prozedur:

1. **Initialisiere***:*
   - $\bar{\boldsymbol{p}} := \dfrac{\frac{1}{n}\sum_{j=1}^{n} \boldsymbol{p}_j}{\left\|\frac{1}{n}\sum_{j=1}^{n} \boldsymbol{p}_j\right\|_2}$
   - $\boldsymbol{u} := (\infty, \ldots, \infty)^T$
2. **WHILE** $\|\boldsymbol{u}\| > \varepsilon$
   - **FOR** $j = 1, \ldots, n$
     $\boldsymbol{p}_j^* := \log_{\bar{\boldsymbol{p}}}(\boldsymbol{p}_j)$
     END
   - $\boldsymbol{u} := \dfrac{1}{n}\displaystyle\sum_{j=1}^{n}\left(\boldsymbol{p}_j^* - \bar{\boldsymbol{p}}\right)$
   - $\bar{\boldsymbol{p}} := \exp_{\bar{\boldsymbol{p}}}(\bar{\boldsymbol{p}} + \boldsymbol{u})$

   END

Ergebnis:
Sphärischer Mittelwert $\bar{\boldsymbol{p}}$*.*

Zur Initialisierung des Algorithmus von Buss und Fillmore wird als erste Annäherung der euklidische Mittelwert der Punkte $\boldsymbol{p}_j$, $j = 1, \ldots, n$, bestimmt und dieser durch Normierung auf die Sphäre projiziert. Anschließend werden mit Hilfe der $\log_{\bar{\boldsymbol{p}}}$–Funktion die Punkte $\boldsymbol{p}_j$ in den Tangentialraum an $\bar{\boldsymbol{p}}$ abgebildet und dort ein neuer euklidischer Mittelwert im Tangentialraum berechnet. Der so bestimmte neue Mittelwert wird mit der $\exp_{\bar{\boldsymbol{p}}}$–Funktion wieder auf die Sphäre abgebildet. Ist die Differenz vom neuen zum alten Mittelwert klein genug, wird abgebrochen. An dieser Stelle sei nochmal darauf hingewiesen, dass vor der Anwendung der $\log_{\bar{\boldsymbol{p}}}$–Funktion alle Punkte so gedreht werden müssen, dass $\bar{\boldsymbol{p}}$ im Nordpol der Sphäre liegt. Nach der Abbildung des neuen Mittelwertes mittels $\exp_{\bar{\boldsymbol{p}}}$ auf die Sphäre ist diese Rotation wieder rückgänig zu machen.

Der Algorithmus 5.4 ist ein Spezialfall des in [7] präsentierten Algorithmus zur Berechnung eines sphärischen *gewichteten* Mittelwertes,

$$\min_{\boldsymbol{t}\in\mathbb{S}^d} \frac{1}{2}\sum_{j=1}^{n} \omega_j \operatorname{dist}_{\mathbb{S}^2}(\boldsymbol{t},\boldsymbol{p}_j)^2, \qquad \sum_{j=1}^{n}\omega_j = 1, \quad \omega_j \geq 0. \tag{5.7}$$

Die Gewichtung wurde hier gleichverteilt mit dem Faktor $1/n$ gewählt. Zudem wird in [7] die Eindeutigkeit des sphärischen Mittelwertes sowie die lineare Konvergenzrate des Algorithmus 5.4 bewiesen.

In ersten Praxistests hat sich die Wahl des sphärischen Mittelwerts für den Referenzpunkt als geeignet erwiesen. Zudem erfüllt dieser die Forderung innerhalb der Großkreishülle bzw. der konvexen Hülle der Punkte $\boldsymbol{p}_j$, $j = 1,\ldots,n$, zu liegen, siehe Abschnitt 5.1.3.

5.1.2 Sphärischer Approximant

Im Folgenden sei der Referenzpunkt $\boldsymbol{r} \in \mathbb{S}^2$ festgelegt und die zu approximierenden Punkte $P = \left(\boldsymbol{p}_j \in \mathbb{S}^2 : j = 1,\ldots,n\right)$ liegen in einer Hemisphäre und mindestens ein $\boldsymbol{p}_j$ liegt im Inneren der Hemisphäre.

Mit Hilfe der folgenden Abbildung $\mathcal{Q}_{\boldsymbol{r}} : \mathbb{S}^2 \to \mathbb{S}^3$ können Repräsentanten der zu approximierenden Punkte P im Raum der Quaternionen bestimmt werden.

$$\mathcal{Q}_{\boldsymbol{r}}(\boldsymbol{p}) := \begin{pmatrix} \cos(\theta/2) \\ \sin(\theta/2)\,\boldsymbol{a} \end{pmatrix}, \qquad \boldsymbol{a} = \frac{\boldsymbol{r}\times\boldsymbol{p}}{\|\boldsymbol{r}\times\boldsymbol{p}\|_2}, \qquad \theta = \cos^{-1}(\boldsymbol{r}^T\boldsymbol{p}). \tag{5.8}$$

Die Quaternion $\mathcal{Q}_{\boldsymbol{r}}(\boldsymbol{p})$ entspricht anschaulich der Drehung, die den Referenzpunkt $\boldsymbol{r}$ auf den Punkt $\boldsymbol{p}$ dreht. Die Anwendung der Abbildung $\mathcal{Q}$ auf die Punkte P wird im Folgenden durch

$$\boldsymbol{q}_j = \mathcal{Q}_{\boldsymbol{r}}(\boldsymbol{p}_j)$$

abgekürzt. Die Punkte $\boldsymbol{q}_j \in \mathbb{H}$ können nun mit einem Spline $s_m\boldsymbol{d}$ in $\mathbb{H}$ approximiert oder interpoliert werden. Auf diesen Spline im Raum der Quaternionen, im Folgenden auch *Quaternionenspline* genannt, lässt sich das in Kapitel 4 beschriebene Verfahren zur Merkmalserkennung anwenden.

Bemerkung 5.5 *Das Knoteneinfügen in einen Quaternionenspline verläuft analog zum bekannten Verfahren aus dem euklidischen Raum $\mathbb{R}^4$, da der Raum der Quaternionen $\mathbb{H}$ in den $\mathbb{R}^4$ eingebettet ist. Der Quaternionenspline*

$$s_m\boldsymbol{d}(\,t \mid T) : \mathbb{R} \to \mathbb{H} \hookrightarrow \mathbb{R}^4$$

kann demnach als Spline im euklidischen Raum $\mathbb{R}^4$ interpretiert werden. Zum Knoteneinfügen wird der bekannte Knoteneinfüge–Algorithmus 3.27 verwendet. Für einen neuen Knoten τ^*, der in die Knotenfolge T mit $\tau_j \leq \tau^* < \tau_{j+1}$ eingefügt werden soll, lassen sich die interessanten Koeffizienten d_k, $k = j-m+1,\ldots,j$, analog zu (3.34) mit

$$d_k^* := \frac{\tau_{k+m}-\tau^*}{\tau_{k+m}-\tau_k}d_{k-1} + \frac{\tau^*-\tau_k}{\tau_{k+m}-\tau_k}d_{k-1}$$

berechnen.

Wie zu Beginn dieses Abschnitts erwähnt, werden die Koordinaten des Splines als homogene Koordinaten aufgefasst. Um sie mit der entsprechenden Einheitsquaternion zu identifizieren, wird der Approximant oder Interpolant normiert

$$s(t) := \frac{s_m \boldsymbol{d}(\, t \mid T)}{\|s_m \boldsymbol{d}(\, t \mid T)\|_2}.$$

Die gewünschte Kurve auf der Sphäre ergibt sich nun aus den durch $s(t)$ beschriebenen Rotationen des Punktes $\boldsymbol{r}$. Hierfür sei $\rho_{s(t)}(\hat{\boldsymbol{r}})$, die durch (3.55) festgelegte Drehung von $\boldsymbol{r}$. Betrachtet man nur den Imaginärteil dieser Abbildung, erhält man die gewünschte Kurve auf der Sphäre.

Die durch $s(t) \in \mathbb{S}^3$ festgelegte Drehung kann in eine Drehmatrix umgeschrieben werden. Hierfür sei

$$\mathcal{S}(\boldsymbol{q}) : \mathbb{S}^3 \to \mathbb{R}^{3\times 3}$$

durch

$$\mathcal{S}(\boldsymbol{q}) := \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \tag{5.9}$$

mit $\boldsymbol{q} = (q_0,\, q_1,\, q_2,\, q_3)^T$ definiert. Mit Hilfe der Abbildung $\mathcal{S}$ kann somit die zu einer Einheitsquaternion korrespondierende Drehmatrix berechnet werden. Eine Übersicht der Konvertierung einer Drehung in ihre verschiedenen Darstellungsmöglichkeiten als Einheitsquaternion, Drehmatrix oder Euler–Winkel kann zum Beispiel in [49] nachgelesen werden.

Für $\boldsymbol{x} \in \mathbb{R}^3$ und $\boldsymbol{q} \in \mathbb{S}^3$ beschreiben $\mathrm{Im}(\rho_{\boldsymbol{q}}(\widehat{\boldsymbol{x}}))$ und die Multiplikation von $\mathcal{S}(\boldsymbol{q})$ und $\boldsymbol{x}$ die gleiche Drehung des Punktes $\boldsymbol{x}$. Es gilt demnach

$$\mathcal{S}(\boldsymbol{q})\boldsymbol{x} = \mathrm{Im}(\rho_{\boldsymbol{q}}(\widehat{\boldsymbol{x}})).$$

Somit kann die gewünschte Kurve auf der Sphäre durch

$$\mathrm{Im}(\rho_{s(t)}(\widehat{\boldsymbol{r}})) \qquad \text{oder} \qquad \mathcal{S}(s(t))\boldsymbol{r}$$

dargestellt werden. Eine effizientere Berechnung erhält man allerdings durch die Quaternionendarstellung.

Der Approximant $s_m\boldsymbol{d}$ der Punkte $\boldsymbol{q}_j$, $j = 1, \ldots, n$, ist bei einfachen inneren Knoten von T eine $(m-1)$–mal stetig differenzierbare Funktion vom Grad m. Durch die Abbildungen $\mathrm{Im}(\rho_{s(t)}(\hat{\boldsymbol{r}}))$ oder $\mathcal{S}(s(t))\boldsymbol{r}$ auf die Sphäre erhöht sich der Grad auf $2m$. Dies kann leicht mit Hilfe der Komponenten von $\mathcal{S}(s(t))$ nachvollzogen werden. Durch die Normierung von $s_m\boldsymbol{d}$ wird die sphärische Kurve zudem rational. Eine Einführung in sphärische rationale Kurven ist zum Beispiel in [23, Kap. 29] von Jüttler und Wagner gegeben.

Wie schon erwähnt, lässt sich auf $s_m\boldsymbol{d}$ zur Analyse der Orientierungsbewegung die in Abschnitt 4.2 vorgestellte Merkmalserkennung anwenden. Mit Hilfe dieser Merkmalserkennung können starke Orientierungswechsel erkannt werden.

Im nächsten Abschnitt wird eine Möglichkeit vorgestellt die Knotenfolge des Approximanten $s_m\boldsymbol{d}$ zu bestimmen.

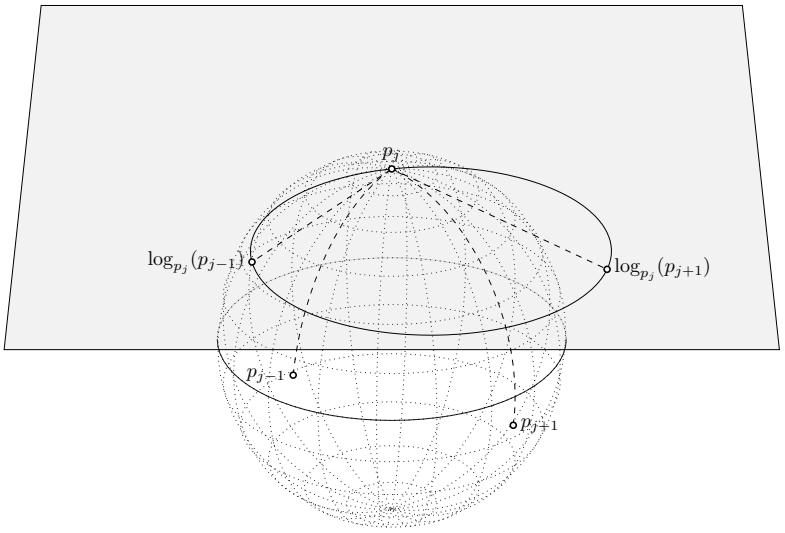


Abb. 5.2: Schätzung der Krümmung am Datenpunkt $\boldsymbol{p}_j$.

Wahl des Knotenvektors

Im euklidischen Fall ist es eine gute Idee die Knoten anhand der Krümmung der Datenpunkte zu verteilen. Hierfür wurde in Abschnitt 4.1.1 zu jedem Datenpunkt ein Krümmungswert mit Hilfe eines interpolierenden Kreises geschätzt. Wendet man diesen Krümmungsschätzer nun auf die Punkte $\boldsymbol{p}_j \in \mathbb{S}^2$, $j = 1, \ldots, n$, an, erhält man für drei aufeinander folgende Punkte $\boldsymbol{p}_{j-1}$, $\boldsymbol{p}_j$ und $\boldsymbol{p}_{j+1}$, die auf einem Großkreis liegen, diesen Großkreis als interpolierenden Kreis und somit eine Krümmung von 1 für den entsprechenden Datenpunkt. Da der Großkreis auf der Sphäre das Pendant zur Geraden im euklidischen Raum ist, erwartet man, dass ein Krümmungsschätzer in einer solchen Situation einen Krümmungswert von 0 liefert. Das Verfahren der Krümmungsschätzung aus Abschnitt 4.1.1 muss somit an die Sphäre angepasst werden. Eine Möglichkeit einen solchen Krümmungsschätzer zu erhalten, kann zum Beispiel mit Hilfe der exp– und log–Abbildungen erreicht werden, da diese Abbildungen längen– und winkeltreu sind. Die folgende Definition stellt eine Möglichkeit der Krümmungsschätzung zu Datenpunkten auf der Sphäre vor.

Definition 5.6 *Zu einem gegebenen Satz an Datenpunkten* $P = \left(\boldsymbol{p}_j \in \mathbb{S}^2 : j = 1, \ldots, n\right)$*, ist die zu einem Datenpunkt* $\boldsymbol{p}_j$ *korrespondierende* Krümmung κ_j *durch*

$$\kappa_j := \begin{cases} 0, & j = 1 \vee j = n \\ 1/r_j, & j = 2, \ldots, n-1 \end{cases} \tag{5.10}$$

gegeben, wobei

$$r_j = \frac{\left\|\boldsymbol{p}^*_{j-1} - \boldsymbol{p}^*_{j+1}\right\|_2}{2\sin(\alpha_j)}, \quad \alpha_j = \cos^{-1}\left(\frac{(\boldsymbol{p}^*_{j+1} - \boldsymbol{p}^*_j)^T(\boldsymbol{p}^*_{j-1} - \boldsymbol{p}^*_j)}{\left\|\boldsymbol{p}^*_{j+1} - \boldsymbol{p}^*_j\right\|_2 \left\|\boldsymbol{p}^*_{j-1} - \boldsymbol{p}^*_j\right\|_2}\right), \quad \boldsymbol{p}^*_k = \log_{\boldsymbol{p}_j}(\boldsymbol{p}_k)$$

für $k = j-1$, j, $j+1$*. Falls* $\alpha_j = 0$ *oder* $\alpha_j = \pi$ *gilt die Vereinbarung, dass* $\kappa_j = 0$*.*

Abbildung 5.2 veranschaulicht diese Methode der Krümmungsschätzung auf der Sphäre. Hierbei werden die drei Punkte $\boldsymbol{p}_{j-1}$, $\boldsymbol{p}_j$ und $\boldsymbol{p}_{j+1}$ in den Tangentialraum der Sphäre an $\boldsymbol{p}_j$ abgebildet und in diesem Tangentialraum der interpolierende Kreis bzw. dessen Radius/Krümmung bestimmt.

Falls die drei Punkte auf einem Großkreis liegen, werden diese Punkte im Tangentialraum auf eine Gerade abgebildet. Nach Vereinbarung entspricht dies einer Krümmung von Null.

Mit Hilfe der nach Definition 5.6 bestimmten Krümmung κ_j zu jedem Datenpunkt $\boldsymbol{p}_j$ kann analog zum euklidischen Fall, Abschnitt 4.1.1, eine Verteilung der Knoten angegeben werden. Hierfür wird der euklidische Abstand jeweils durch die Länge der Großkreisverbindung ersetzt. Der Polygonzug der Datenpunkte P entspricht somit den Großkreisverbindungen zwischen den jeweiligen Punkten $\boldsymbol{p}_j$ auf der Sphäre. Für die Gesamtlänge ℓ des „Polygonzuges“ ergibt sich somit

$$\ell = \sum_{j=1}^{n-1} \operatorname{dist}_{\mathbb{S}^2}(\boldsymbol{p}_j, \boldsymbol{p}_{j+1}).$$

Die Datenpunkte $\boldsymbol{p}_k$, $k = 2, \ldots, n$, werden ebenfalls mit ihren jeweiligen „Polygonzuglängen“ identifiziert

$$u_k := \sum_{j=1}^{k-1} \operatorname{dist}_{\mathbb{S}^2}(\boldsymbol{p}_j, \boldsymbol{p}_{j+1}).$$

Analog zum euklidischen Fall kann nun mit Hilfe von (4.6) bzw. (4.7) die Position der Knoten τ_j im Intervall $[0, \ell]$ durch

$$\tau_j = f^{-1}(f(\tau_j))$$

bestimmt werden.

Sphärischer Schoenberg–Approximant

Zu den gegebenen Punkten $P = \left(\boldsymbol{p}_j \in \mathbb{S}^2 : j = 1, \ldots, \ell\right)$, dem Referenzpunkt $\boldsymbol{r}$ und einer Knotenfolge $T_{n,m}$, die, wie im Abschnitt zuvor beschrieben, bestimmt wurde, kann ein *sphärischer Schoenberg–Approximant* wie folgt angegeben werden. Das Parameterintervall bzw. die Knotenverteilung in T ist annähernd nach dem Großkreisabstand der Punkte $\boldsymbol{p}_j$ bestimmt worden. Somit können die Greville–Abszissen $\hat{\boldsymbol{t}} = (\hat{t}_1, \ldots, \hat{t}_n)$ zum Knotenvektor $T_{n,m}$ mit Positionen auf den entsprechenden Großkreisen zwischen den Punkten $\boldsymbol{p}_j$ identifiziert werden. Eine Greville–Abszisse $\hat{t}_k$, die mit ihrem Wert einer „Bogenlänge“ zwischen den Punkten $\boldsymbol{p}_r$ und $\boldsymbol{p}_{r+1}$ entspricht,

$$\sum_{j=1}^{r-1} \operatorname{dist}_{\mathbb{S}^2}(\boldsymbol{p}_j, \boldsymbol{p}_{j+1}) \le \hat{t}_k \le \sum_{j=1}^{r} \operatorname{dist}_{\mathbb{S}^2}(\boldsymbol{p}_j, \boldsymbol{p}_{j+1}),$$

kann mit dem Punkt

$$p(\hat{t}_k) := \frac{\sin(1-\alpha_k)\varphi}{\sin(\varphi)}\boldsymbol{p}_r + \frac{\sin(\alpha_k)\varphi}{\sin(\varphi)}\boldsymbol{p}_{r+1} \tag{5.11}$$

identifiziert werden, wobei

$$\alpha_k = \frac{\hat{t}_k - \sum_{j=1}^{r-1} \operatorname{dist}_{\mathbb{S}^2}(\boldsymbol{p}_j, \boldsymbol{p}_{j+1})}{\operatorname{dist}_{\mathbb{S}^2}(\boldsymbol{p}_r, \boldsymbol{p}_{r+1})}, \qquad \varphi = \cos^{-1}(\boldsymbol{p}_r^T \boldsymbol{p}_{r+1}).$$

Mit den Greville–Abszissen $\hat{t}_k$ und den dazugehörigen Punkten $p(\hat{t}_k)$, $k = 1, \ldots, n$, erhält man eine Abtastung des „Polygonzuges“ auf der Sphäre. Für diese Abtastungen werden die entsprechenden Drehungen bzw. Quaternionen ermittelt, die benötigt werden, um den Punkt $\boldsymbol{r}$ in die Punkte $p(\hat{t}_k)$, $k = 1, \ldots, n$ zu drehen:

$$\boldsymbol{q}_k := \mathcal{Q}_{\boldsymbol{r}}(p(\hat{t}_k)), \qquad k = 1, \ldots, n.$$

Im Raum der Quaternionen kann hiermit ein Schoenberg–Approximant durch

$$s_m \boldsymbol{d}(\, t \mid T) = \sum_{j=1}^{n} \boldsymbol{q}_j N_j^m \,(t \mid T)$$

angegeben werden. Der entsprechende *sphärische Schoenberg–Approximant* zu den Daten P ist durch

$$\mathrm{Im}(\rho_{s(t)}(\hat{\boldsymbol{r}})) \qquad \text{bzw.} \qquad \mathcal{S}(s(t))\boldsymbol{r}$$

gegeben, wobei

$$s(t) = \frac{s_m \boldsymbol{d}(\, t \mid T)}{\|s_m \boldsymbol{d}(\, t \mid T)\|_2}.$$

Definition 5.7 *Sei* $P = \left(\boldsymbol{p}_j \in \mathbb{S}^2 : j = 1, \ldots, n\right)$ *eine Folge von Datenpunkten,* $\boldsymbol{r} \in \mathbb{S}^2$ *ein Referenzpunkt auf der Sphäre sowie* $T_{n,m}$ *ein Knotenvektor mit* $(m+1)$*–fachen Randknoten und einfachen inneren Knoten. Der* sphärische Schoenberg–Approximant *ist durch*

$$\mathscr{S}_{m,\mathbb{S}} P_{\boldsymbol{r}}(\, t \mid T) := \mathcal{S}\left(s(t)\right) \boldsymbol{r}, \qquad s(t) = \frac{\sum_{j=1}^{n} \mathcal{Q}_{\boldsymbol{r}}(\boldsymbol{p}_j) N_j^m \,(t \mid T)}{\left\| \sum_{j=1}^{n} \mathcal{Q}_{\boldsymbol{r}}(\boldsymbol{p}_j) N_j^m \,(t \mid T) \right\|_2} \tag{5.12}$$

definiert. Für einen entsprechenden sphärischen Schoenberg–Operator *gilt*

$$\boldsymbol{p}_j = f(\hat{t}_j)$$

zu einer passenden Knotenfolge T*.*

Bemerkung 5.8 *Für eine entsprechende Approximationsgüte wird in Definition 5.7 vorausgesetzt, dass die Datenpunkte mit den jeweiligen Greville–Abszissen der Knotenfolge identifiziert werden können. Sollte dies nicht der Fall sein, interpretiert man die Datenpunkte als „sphärisch stückweise lineare" Funktion und berechnet eine Abtastung dieses „sphärischen Polygonzuges" die zu den Greville–Abszissen der gewünschten Knotenfolge passt, wie in diesem Abschnitt beschrieben wurde.*

5.1.3 Konvexe Hülle

Die Wahl des Referenzpunktes nimmt auf den Verlauf des Quaternionensplines Einfluss. In Abbildung 5.3 sind die nach Abschnitt 5.1.2 bestimmten Quaternionensplines zu identischen Datenpunkten $\boldsymbol{p}_j \in \mathbb{S}^d$, aber unterschiedlichen Referenzpunkten abgebildet. Wie man gut erkennen kann, haben die Kurven unterschiedliche Verläufe. In diesem Abschnitt wird gezeigt, dass die Wahl eines Referenzpunktes im Inneren der konvexen Hülle der Datenpunkte $\boldsymbol{p}_j$ zur Folge hat, dass auch die entsprechende Splinekurve im Inneren der konvexen Hülle liegt. Hierfür ist zunächst die Definition einer sphärischen konvexen Hülle notwendig.

Definition 5.9 *Eine Teilmenge* S *der Sphäre* $\mathbb{S}^d$ *heißt* $\mathbb{S}^d$–konvex*, falls für* $\boldsymbol{s}_1, \boldsymbol{s}_2 \in S$ *die kürzere Großkreisverbindung zwischen* $\boldsymbol{s}_1$ *und* $\boldsymbol{s}_2$ *ebenfalls in* S *liegt,*

$$\frac{\sin((1-t)\varphi)}{\sin(\varphi)} \boldsymbol{s}_1 + \frac{\sin(t\,\varphi)}{\sin(\varphi)} \boldsymbol{s}_2 \in S, \qquad t \in [0,1], \qquad \varphi = \cos^{-1}(\boldsymbol{s}_1^T \boldsymbol{s}_2). \tag{5.13}$$

Im Folgenden ist der Begriff konvex mit $\mathbb{S}^d$–konvex synonym, da nur Mengen auf der Sphäre betrachtet werden.

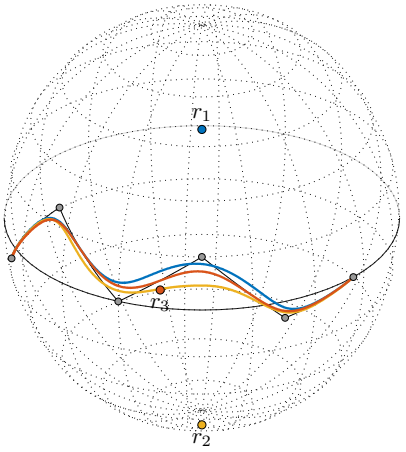


Abb. 5.3: Sphärischer Schoenberg–Approximant zu unterschiedlichen Referenzpunkten r. Der Referenzpunkt r_3 entspricht dem nach Algorithmus 5.4 bestimmten sphärischen Mittelwert der Datenpunkte p_j (grau).

Definition 5.10 *Die Menge $C \subseteq \mathbb{S}^d$ heißt* konvexe Hülle *von $P \subseteq \mathbb{S}^d$, falls C der Durchschnitt aller konvexer Mengen $K \subseteq \mathbb{S}^d$ ist, die P enthalten,*

$$C := \bigcap_{\substack{P \subseteq K \\ K \, konvex}} K. \tag{5.14}$$

Die Menge C ist somit die kleinste konvexe Menge, die P enthält. Abkürzend wird als Symbol der konvexen Hülle einer Menge P die Notation $[P]$ verwendet.

Diese Definition einer konvexen Menge erlaubt es, dass zwei antipodale Punkte in einer konvexen Menge liegen dürfen solange eine Geodäte (kürzeste Großkreisverbindung) existiert, die diese Punkte verbindet und innerhalb dieser Menge liegt. Besteht eine Menge aus genau zwei antipodalen Punkten existiert keine konvexe Hülle, da es unendlich viele Geodäten zwischen diesen beiden Punkten gibt. Im euklidischen Raum ist es zudem möglich die konvexe Hülle einer Menge X als Menge aller endlichen Konvexkombinationen anzugeben, siehe zum Beispiel [34]. Für eine Menge $X \in \mathbb{R}^d$ gilt demnach:

$$[X] = \left\{ \sum_{j=1}^{n} \lambda_j x_j \;\middle|\; x_j \in X,\, \lambda_j \geq 0,\, \sum_{j=1}^{n} \lambda_j = 1,\, n \in \mathbb{N} \right\}.$$

Auf der Sphäre gibt es eine ähnliche Möglichkeit die konvexe Hülle zu einer *endlichen* Menge $S = \{s_1, \ldots, s_k\} \subset \mathbb{S}^d$ anzugeben. Der folgende Satz findet sich mit Beweis in [7].

Satz 5.11 *Sei $S = \{s_1, \ldots, s_k\} \subset \mathbb{S}^d$ mit $s_1 \neq s_2$ für $k = 2$. Liegen alle Punkte s_j in einer Hemisphäre $\mathcal{H}$ der Sphäre $\mathbb{S}^d$ und mindestens ein Punkt im Inneren von $\mathcal{H}$, so gilt*

$$[S] = \left\{ \operatorname*{argmin}_{t} \sum_{j=1}^{k} \omega_j \operatorname{dist}_{\mathbb{S}^d}(t, s_j) \;\middle|\; \omega_j \geq 0,\, \sum_{j=1}^{k} \omega_j = 1 \right\} \tag{5.15}$$

und $[S] \subset \mathcal{H}$. Sollten die Punkte s_j nicht in einer Hemisphäre liegen, so gilt:

$$[S] = \mathbb{S}^d.$$

Eine Folgerung aus Satz 5.11 ist, dass ein nach Algorithmus 5.4 bestimmter sphärischer Mittelwert der Punkte $P = \left(\boldsymbol{p}_j \in \mathbb{S}^d : j = 1, \ldots, n\right)$ stets in der konvexen Hülle $[P]$ dieser Punkte liegt.

Als nächstes wird gezeigt, dass ein sphärischer Schoenberg–Approximant der Punkte $P = \left(\boldsymbol{p}_j \in \mathbb{S}^d : j = 1, \ldots, n\right)$ innerhalb der konvexen Hülle $[P]$ der zu approximierenden Punkte liegt, falls der Referenzpunkt $\boldsymbol{r}$ ebenfalls innerhalb der konvexen Hülle liegt.

In den Beweisen zu Lemma 5.12 und 5.13 werden für einige Umformungen die Doppelwinkelformeln

$$\sin(2x) = 2\,\sin(x)\cos(x) \tag{5.16}$$
$$\cos(2x) = \cos^2(x) - \sin^2(x) \tag{5.17}$$

des Sinus und Kosinus ausgenutzt, wobei (5.17) in umgeformter Version Anwendung findet:

$$\cos^2(x) = \frac{1}{2}(\cos(2x) + 1). \tag{5.18}$$

Lemma 5.12 *Für $\alpha, \beta \in [0, 2\pi]$ gilt:*

$$4\cos^2(\alpha/2)\cos^2(\beta/2) - 2\cos^2(\alpha/2)\cos(\beta) - 2\cos^2(\beta/2)\cos(\alpha) = 1 - \cos(\alpha)\cos(\beta). \tag{5.19}$$

Beweis: Seien $\alpha, \beta \in [0, 2\pi]$.

$$\begin{aligned}
&4\cos^2(\alpha/2)\cos^2(\beta/2) - 2\cos^2(\alpha/2)\cos(\beta) - 2\cos^2(\beta/2)\cos(\alpha) \\
&\quad= 4\left(\frac{1}{2}\left(\cos(\alpha)+1\right)\right)\left(\frac{1}{2}\left(\cos(\beta)+1\right)\right) - 2\left(\frac{1}{2}\left(\cos(\alpha)+1\right)\right)\cos(\beta) \\
&\qquad - 2\left(\frac{1}{2}\left(\cos(\beta)+1\right)\right)\cos(\alpha) \\
&\quad= \left(\cos(\alpha)+1\right)\left(\cos(\beta)+1\right) - \left(\cos(\alpha)+1\right)\cos(\beta) - \left(\cos(\beta)+1\right)\cos(\alpha) \\
&\quad= \cos(\alpha)\cos(\beta) + \cos(\alpha) + \cos(\beta) + 1 - \cos(\alpha)\cos(\beta) - \cos(\beta) - \cos(\alpha)\cos(\beta) - \cos(\alpha) \\
&\quad= 1 - \cos(\alpha)\cos(\beta)
\end{aligned}$$

□

Lemma 5.13 *Seien $\boldsymbol{r} \neq \boldsymbol{a} \neq \boldsymbol{b} \in \mathbb{S}^2$ und $\boldsymbol{r}$ liegt nicht auf dem Großkreis, der durch $\boldsymbol{a}$ und $\boldsymbol{b}$ verläuft. Weiter seien $\boldsymbol{q}_1, \boldsymbol{q}_2 \in \mathbb{H}$ mit $\boldsymbol{q}_1 = \mathcal{Q}_{\boldsymbol{r}}(\boldsymbol{a})$ und $\boldsymbol{q}_2 = \mathcal{Q}_{\boldsymbol{r}}(\boldsymbol{b})$ sowie $\mathcal{H}$ die Hemisphäre, die durch Teilung von $\mathbb{S}^2$ mittels des Großkreises durch $\boldsymbol{a}$ und $\boldsymbol{b}$ entsteht und $\boldsymbol{r}$ enthält. Dann gilt:*

$$f_{\boldsymbol{r}}(t) := \frac{1}{\sum_{j=0}^{3} q_j(t)^2}\,\mathcal{S}(q(t))\boldsymbol{r} \;\in\; \mathcal{H}, \tag{5.20}$$

wobei $q(t) = (1-t)\boldsymbol{q}_1 + t\,\boldsymbol{q}_2$, $t \in [0, 1]$.

Beweis: Es kann ohne Einschränkung angenommen werden, dass die Punkte $\boldsymbol{a}$ und $\boldsymbol{b}$ stets auf dem Äquator liegen und der Referenzpunkt $\boldsymbol{r}$ in der nördlichen Hemisphäre liegt. Ist das nicht

so, wählt man ein entsprechendes Koordinatensystem. Bei einer solchen Punktkonfiguration von $\boldsymbol{r}$, $\boldsymbol{a}$ und $\boldsymbol{b}$ ist zu zeigen, dass $f_{\boldsymbol{r}}$ stets in der nördlichen Hemisphäre von $\mathbb{S}^2$ liegt. Hierfür reicht es zu zeigen, dass die dritte Komponente von $f_{\boldsymbol{r}}$ stets größer oder gleich Null ist.

Da $\boldsymbol{a} = (a_1, a_2, a_3)^T$ sowie $\boldsymbol{b} = (b_1, b_2, b_3)^T$ auf dem Äquator liegen, gilt

$$a_3 = 0 = b_3. \tag{5.21}$$

Für die Quaternionen $\boldsymbol{q}_1$ und $\boldsymbol{q}_2$ gilt mit (5.21) und $\|\boldsymbol{r}\|_2 \|\boldsymbol{a}\|_2 = 1$

$$\begin{aligned}
\boldsymbol{q}_1 = \mathcal{Q}_{\boldsymbol{r}}(a) &= \begin{pmatrix} \cos(\varphi_1/2) \\ \sin(\varphi_1/2)\,(r_2\,a_3 - r_3\,a_2)\,/\,\|\boldsymbol{r}\times\boldsymbol{a}\|_2 \\ \sin(\varphi_1/2)\,(r_3\,a_1 - r_1\,a_3)\,/\,\|\boldsymbol{r}\times\boldsymbol{a}\|_2 \\ \sin(\varphi_1/2)\,(r_1\,a_2 - r_2\,a_1)\,/\,\|\boldsymbol{r}\times\boldsymbol{a}\|_2 \end{pmatrix} \\
&= \begin{pmatrix} \cos(\varphi_1/2) \\ \sin(\varphi_1/2)\,(r_2\,a_3 - r_3\,a_2)\,/\,(\|\boldsymbol{r}\|_2\,\|\boldsymbol{a}\|_2 \sin(\varphi_1)) \\ \sin(\varphi_1/2)\,(r_3\,a_1 - r_1\,a_3)\,/\,(\|\boldsymbol{r}\|_2\,\|\boldsymbol{a}\|_2 \sin(\varphi_1)) \\ \sin(\varphi_1/2)\,(r_1\,a_2 - r_2\,a_1)\,/(\|\boldsymbol{r}\|_2\,\|\boldsymbol{a}\|_2 \sin(\varphi_1)) \end{pmatrix} \\
&= \begin{pmatrix} \cos(\varphi_1/2) \\ \left(\sin(\varphi_1/2)/\sin(\varphi_1)\right)(r_2\,a_3 - r_3\,a_2) \\ \left(\sin(\varphi_1/2)/\sin(\varphi_1)\right)(r_3\,a_1 - r_1\,a_3) \\ \left(\sin(\varphi_1/2)/\sin(\varphi_1)\right)(r_1\,a_2 - r_2\,a_1) \end{pmatrix} \\
&= \begin{pmatrix} \cos(\varphi_1/2) \\ \left(\sin(\varphi_1/2)/\sin(\varphi_1)\right)(-r_3\,a_2) \\ \left(\sin(\varphi_1/2)/\sin(\varphi_1)\right)(r_3\,a_1) \\ \left(\sin(\varphi_1/2)/\sin(\varphi_1)\right)(r_1\,a_2 - r_2\,a_1) \end{pmatrix}
\end{aligned} \tag{5.22}$$

sowie

$$\boldsymbol{q}_2 = \mathcal{Q}_{\boldsymbol{r}}(\boldsymbol{b}) = \begin{pmatrix} \cos(\varphi_2/2) \\ \left(\sin(\varphi_2/2)/\sin(\varphi_2)\right)(-r_3\,b_2) \\ \left(\sin(\varphi_2/2)/\sin(\varphi_2)\right)(r_3\,b_1) \\ \left(\sin(\varphi_2/2)/\sin(\varphi_2)\right)(r_1\,b_2 - r_2\,b_1) \end{pmatrix}, \tag{5.23}$$

wobei $\varphi_1 = \cos^{-1}(\boldsymbol{r}^T\boldsymbol{a})$ und $\varphi_2 = \cos^{-1}(\boldsymbol{r}^T\boldsymbol{b})$. Für die beiden Winkel gilt $\varphi_k \in (0, \pi)$, $k = 1, 2$, da $\boldsymbol{r}$ in der nördlichen Hemisphäre, aber nicht auf dem Äquator liegt. Einsetzen von (5.22) und (5.23) in $q(t)$ liefert

$$\begin{aligned}
q_0(t) &= (1-t)\cos(\varphi_1/2) + t\cos(\varphi_2/2)\,, \\
q_1(t) &= (1-t)\frac{\sin(\varphi_1/2)}{\sin(\varphi_1)}(-r_3\,a_2) + t\,\frac{\sin(\varphi_2/2)}{\sin(\varphi_2)}(-r_3\,b_2)\,, \\
q_2(t) &= (1-t)\frac{\sin(\varphi_1/2)}{\sin(\varphi_1)}(r_3\,a_1) + t\,\frac{\sin(\varphi_2/2)}{\sin(\varphi_2)}(r_3\,b_1)\,, \\
q_3(t) &= (1-t)\frac{\sin(\varphi_1/2)}{\sin(\varphi_1)}(r_1\,a_2 - r_2\,a_1) + t\,\frac{\sin(\varphi_2/2)}{\sin(\varphi_2)}(r_1\,b_2 - r_2\,b_1)\,.
\end{aligned}$$

Für $f_{\boldsymbol{r}}$ gilt:

$$f_{\boldsymbol{r}}(t) = \frac{1}{\sum_{j=0}^{3} q_j^2}\begin{pmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{pmatrix}\boldsymbol{r}.$$

Zur Vereinfachung der Notation wurde das Argument t von $q_j(t)$, $j = 0, \ldots, 3$, weggelassen. Für die dritte Komponente von $f_{\boldsymbol{r}}$ gilt somit

$$f_{\boldsymbol{r},3}(t) = \frac{1}{\sum_{j=0}^{3} q_j^2} \Big(2(q_1 q_3 - q_0 q_2) r_1 + 2(q_2 q_3 + q_0 q_1) r_2 + (q_0^2 - q_1^2 - q_2^2 + q_3^2) r_3 \Big). \tag{5.24}$$

Da $q(t)$ die Quaternionen $\boldsymbol{q}_1$ und $\boldsymbol{q}_2$ an $t = 0$ bzw. $t = 1$ interpoliert, gilt $f_{\boldsymbol{r}}(0) = \boldsymbol{a}$ sowie $f_{\boldsymbol{r}}(1) = \boldsymbol{b}$ und damit $f_{\boldsymbol{r},3}(0) = 0 = f_{\boldsymbol{r},3}(1)$. Der Nenner in $f_{\boldsymbol{r},3}$ ist stets positiv und kann somit zum Nachweis, dass $f_{\boldsymbol{r},3}$ stets größer oder gleich Null ist vernachlässigt werden. Sodann bleibt zu zeigen, dass $g(t) \geq 0$ gilt, wobei

$$g(t) := 2(q_1 q_3 - q_0 q_2) r_1 + 2(q_2 q_3 + q_0 q_1) r_2 + (q_0^2 - q_1^2 - q_2^2 + q_3^2) r_3.$$

Für $g(t)$ gilt ebenfalls

$$g(0) = 0 = g(1). \tag{5.25}$$

Die Funktion $g(t)$ ist ein quadratisches Polynom in t und kann somit als Linearkombination von Bernsteinpolynomen B_j^2 (siehe Definition 3.8) dargestellt werden,

$$g(t) = \sum_{j=0}^{2} d_j B_j^2(t).$$

Die Koeffizienten d_j, $j = 0, 1, 2$, können wie folgt berechnet werden

$$\begin{aligned} d_0 = {} & \left(\frac{\sin(\varphi_1/2)}{\sin(\varphi_1)} \right)^2 \left(-r_3^3 a_2^2 - r_3^3 a_1^2 - r_3 \left(r_1 a_2 - r_2 a_1 \right)^2 \right) \\ & + r_3 \cos^2(\varphi_1/2) - r_1 r_3 a_1 - r_2 r_3 a_2, \end{aligned} \tag{5.26}$$

$$\begin{aligned} d_1 = {} & \frac{\sin(\varphi_1/2)\sin(\varphi_2/2)}{\sin(\varphi_1)\sin(\varphi_2)} \Big(r_1 r_2 r_3 (a_1 b_2 + a_2 b_1) - r_2^2 r_3 a_1 b_1 - r_1^2 r_3 a_2 b_2 - r_3^3 (a_1 b_1 + a_2 b_2) \Big) \\ & - \frac{\cos(\varphi_1/2)\sin(\varphi_2/2)}{\sin(\varphi_2)} \left(r_1 r_3 b_1 + r_2 r_3 b_2 \right) - \frac{\cos(\varphi_2/2)\sin(\varphi_1/2)}{\sin(\varphi_1)} \left(r_1 r_3 a_1 + r_2 r_3 a_2 \right) \\ & + \cos(\varphi_1/2)\cos(\varphi_2/2) r_3, \end{aligned} \tag{5.27}$$

$$\begin{aligned} d_2 = {} & \left(\frac{\sin(\varphi_2/2)}{\sin(\varphi_2)} \right)^2 \left(-r_3^3 b_2^2 - r_3^3 b_1^2 - r_3 \left(r_1 b_2 - r_2 b_1 \right)^2 \right) \\ & + r_3 \cos^2(\varphi_2/2) - r_1 r_3 b_1 - r_2 r_3 b_2. \end{aligned} \tag{5.28}$$

Aus (5.25) folgt $d_0 = 0 = d_2$. Somit ist $g(t) \geq 0$, $t \in [0, 1]$ äquivalent zu $d_1 \geq 0$. Der Koeffizient d_1 lässt sich wie folgt umformen:

$$\begin{aligned} d_1 = \frac{r_3}{\sin(\varphi_1)\sin(\varphi_2)} \Big(& \sin(\varphi_1/2)\sin(\varphi_2/2) \Big(r_1 r_2 (a_1 b_2 + a_2 b_1) - r_2^2 a_1 b_1 - r_1^2 a_2 b_2 - r_3^2 (a_1 b_1 + a_2 b_2) \Big) \\ & - \cos(\varphi_1/2)\sin(\varphi_2/2)\sin(\varphi_1) \left(r_1 b_1 + r_2 b_2 \right) \\ & - \cos(\varphi_2/2)\sin(\varphi_1/2)\sin(\varphi_2) \left(r_1 a_1 + r_2 a_2 \right) \\ & + \cos(\varphi_1/2)\cos(\varphi_2/2)\sin(\varphi_1)\sin(\varphi_2) \Big). \end{aligned}$$

Anwenden der Doppelwinkelformel (5.16) und weiteres Faktorisieren liefert

$$\begin{aligned} d_1 = \frac{\sin(\varphi_1/2)\sin(\varphi_2/2)}{\sin(\varphi_1)\sin(\varphi_2)}\, r_3\Big(&\Big(r_1r_2(a_1b_2+a_2b_1) - r_2^2a_1b_1 - r_1^2a_2b_2 - r_3^2(a_1b_1+a_2b_2)\Big) \\ &- 2\cos^2(\varphi_1/2)\,(r_1b_1+r_2b_2) - 2\cos^2(\varphi_2/2)\,(r_1a_1+r_2a_2) \\ &+ 4\cos^2(\varphi_1/2)\cos^2(\varphi_2/2)\Big). \end{aligned}$$

Im Folgenden seien

$$A := \frac{\sin(\varphi_1/2)\sin(\varphi_2/2)}{\sin(\varphi_1)\sin(\varphi_2)}\, r_3$$

und

$$\begin{aligned} B := {} & r_1r_2(a_1b_2+a_2b_1) - r_2^2a_1b_1 - r_1^2a_2b_2 - r_3^2(a_1b_1+a_2b_2) \\ & - 2\cos^2(\varphi_1/2)\,(r_1b_1+r_2b_2) - 2\cos^2(\varphi_2/2)\,(r_1a_1+r_2a_2) \\ & + 4\cos^2(\varphi_1/2)\cos^2(\varphi_2/2), \end{aligned}$$

so dass $d_1 = AB$. Da $\varphi_1, \varphi_2 \in (0,\pi)$ und $\boldsymbol{r}$ in der nördlichen Hemisphäre liegt, gilt $A > 0$. Wegen $a_3 = 0 = b_3$ kann B zu einem Skalarprodukt $\boldsymbol{r}^T\boldsymbol{a}$ bzw. $\boldsymbol{r}^T\boldsymbol{b}$ ergänzt werden. Diese inneren Produkte können durch $\cos(\varphi_1)$ bzw. $\cos(\varphi_2)$ ersetzt werden, da $\boldsymbol{r}, \boldsymbol{a}, \boldsymbol{b} \in \mathbb{S}^2$ und somit Länge eins haben. Für B gilt demnach

$$\begin{aligned} B = {} & r_1r_2(a_1b_2+a_2b_1) - r_2^2a_1b_1 - r_1^2a_2b_2 - r_3^2(a_1b_1+a_2b_2) \\ & - 2\cos^2(\varphi_1/2)\,\boldsymbol{r}^T\boldsymbol{b} - 2\cos^2(\varphi_2/2)\,\boldsymbol{r}^T\boldsymbol{a} \\ & + 4\cos^2(\varphi_1/2)\cos^2(\varphi_2/2) \\ \\ = {} & r_1r_2(a_1b_2+a_2b_1) - r_2^2a_1b_1 - r_1^2a_2b_2 - r_3^2(a_1b_1+a_2b_2) \\ & - 2\cos^2(\varphi_1/2)\cos(\varphi_2) - 2\cos^2(\varphi_2/2)\cos(\varphi_1) \\ & + 4\cos^2(\varphi_1/2)\cos^2(\varphi_2/2). \end{aligned}$$

Mit Lemma 5.12 folgt

$$\begin{aligned} B = {} & r_1r_2(a_1b_2+a_2b_1) - r_2^2a_1b_1 - r_1^2a_2b_2 - r_3^2(a_1b_1+a_2b_2) \\ & + 1 - \cos(\varphi_1)\cos(\varphi_2). \end{aligned}$$

Für die Cosinus–Ausdrücke in B gilt $\cos(\varphi_1) = \boldsymbol{r}^T\boldsymbol{a}$ sowie $\cos(\varphi_2) = \boldsymbol{r}^T\boldsymbol{b}$. Somit kann der Ausdruck B durch Ersetzen der beiden Cosinusfunktionen mit dem entsprechenden inneren Produkt wie folgt umgeformt werden:

$$\begin{aligned} B = {} & r_1r_2(a_1b_2+a_2b_1) - r_2^2a_1b_1 - r_1^2a_2b_2 - r_3^2(a_1b_1+a_2b_2) \\ & + 1 - (r_1a_1+r_2a_2+r_3a_3)(r_1b_1+r_2b_2+r_3b_3) \\ \\ = {} & 1 - r_1^2(a_1b_1+a_2b_2) - r_2^2(a_1b_1+a_2b_2) - b_3^2(a_1b_1+a_2b_2+a_3b_3) \\ & - r_1r_3(a_1b_3+a_3b_1) - r_2r_3(a_2b_3+a_3b_2). \end{aligned}$$

Alle Summanden mit a_3 bzw. b_3 können wieder weggelassen werden, da $a_3 = 0 = b_3$.

$$\begin{aligned} B &= 1 - r_1^2(a_1b_1+a_2b_2) - r_2^2(a_1b_1+a_2b_2) - b_3^2(a_1b_1+a_2b_2) \\ &= 1 - (r_1^2+r_2^2+r_3^3)(a_1b_1+a_2b_2) \\ &= 1 - (a_1b_1+a_2b_2+a_3b_3) \\ &= 1 - \boldsymbol{a}^T\boldsymbol{b} \end{aligned}$$

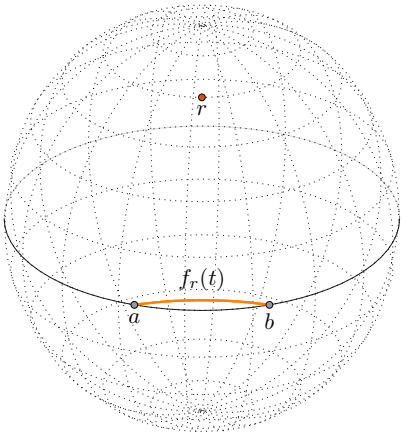


Abb. 5.4: Konvexkombination der Quaternionen $\mathcal{Q}_r(\boldsymbol{a})$ und $\mathcal{Q}_r(\boldsymbol{b})$ auf der Sphäre.

Da $\boldsymbol{a}^T\boldsymbol{b} = \cos(\alpha)$, wobei $\alpha = \cos^{-1}(\boldsymbol{a}^T\boldsymbol{b})$ gilt $B \geq 0$. Aus $A > 0$ und $B \geq 0$, folgt $d_1 \geq 0$. Somit liegt $f_{\boldsymbol{r}}(t)$ in der nördlichen Hemisphäre. □

Die Abbildung 5.4 veranschaulicht die Aussage des Lemma 5.13, mit dessen Hilfe der folgende Satz über die konvexe Hülle eines sphärischen Schoenberg–Approximanten bewiesen werden kann.

Satz 5.14 *Für $n > 2$ seien $P = \left(\boldsymbol{p}_j \in \mathbb{S}^2 : j = 1, \ldots, n\right)$ die zu approximierenden Datenpunkte. Weiter sei $\boldsymbol{r} \in \mathbb{S}^2$ ein Referenzpunkt auf der Sphäre. Damit gilt:*

$$\boldsymbol{r} \in [P] \quad \Rightarrow \quad \mathscr{S}_{m,\mathbb{S}} P_{\boldsymbol{r}}(t) \in [P]. \tag{5.29}$$

Beweis: Für den Beweis des Satzes gilt es zwei Fälle zu unterscheiden.

1. Fall:
Die Punkte sind auf der Sphäre $\mathbb{S}^2$ so verteilt, dass man keine Hemisphäre finden kann, die alle Punkte enthält. In diesem Fall gilt nach Satz 5.11 $[P] = \mathbb{S}^2$ und die Aussage folgt trivialerweise.

2. Fall:
Die Punkte aus P liegen in einer Hemisphäre $\mathcal{H}$ von $\mathbb{S}^2$ sowie $\boldsymbol{r} \in [P]$. Um zu zeigen, dass $\mathscr{S}_{m,\mathbb{S}} P_{\boldsymbol{r}} \in [P]$ ist, reicht es zu zeigen, dass die normierten Konvexkombinationen von $\mathcal{Q}_{\boldsymbol{r}}(\boldsymbol{p}_j)$, $j = 1, \ldots, n$, unter der Abbildung

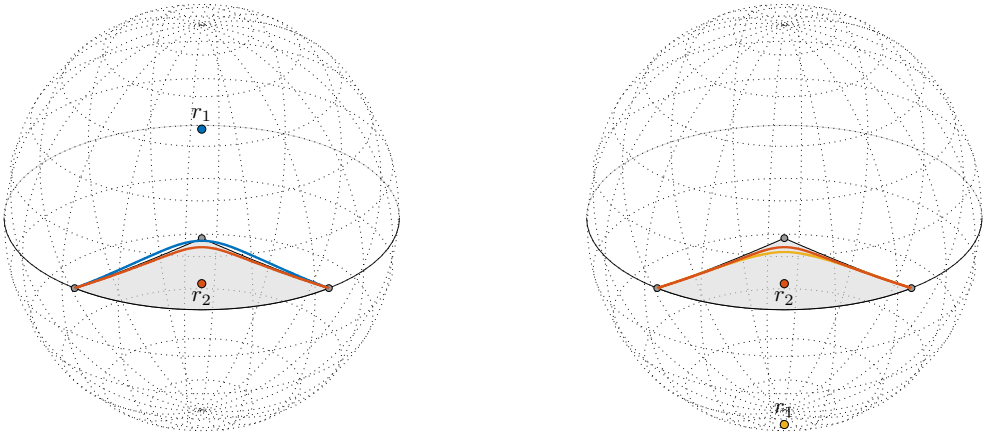
$$f(\,\cdot\,) := \mathcal{S}(\,\cdot\,)\boldsymbol{r}$$

in der konvexen Hülle von P liegen, denn dies sind nach Definition von $\mathscr{S}_{m,\mathbb{S}} P_{\boldsymbol{r}}$ gerade die Kontrollpunkte des Quaternionensplines, der die Drehungen des Referenzpunktes beschreibt. Da $\mathcal{S}(\mathcal{Q}_{\boldsymbol{r}}(\boldsymbol{p}_j))\boldsymbol{r} = \boldsymbol{p}_j$, $j = 1, \ldots, n$, und nach Lemma 5.13 die normierte Konvexkombination

$$q_{j,k}(t) := \frac{(1-t)\mathcal{Q}_{\boldsymbol{r}}(\boldsymbol{p}_j) + t\mathcal{Q}_{\boldsymbol{r}}(\boldsymbol{p}_k)}{\left\|(1-t)\mathcal{Q}_{\boldsymbol{r}}(\boldsymbol{p}_j) + t\mathcal{Q}_{\boldsymbol{r}}(\boldsymbol{p}_k)\right\|_2}$$

unter der Abbildung f auf der Seite des Großkreises durch $\boldsymbol{p}_j$ und $\boldsymbol{p}_k$ liegt auf der auch der Referenzpunkt $\boldsymbol{r}$ liegt, gilt:

$$f(q_{j,k}(t)) \in [P].$$



(a) Sphärische Splines innerhalb und außerhalb der konvexen Hülle.

(b) Beide sphärischen Splines innerhalb der konvexen Hülle.

Abb. 5.5: Sphärische Splines innerhalb und außerhalb der konvexen Hülle. Der Referenzpunkt der jeweiligen Kurve ist mit $\boldsymbol{r}_k$, $k = 1, 2$, in der gleichen Farbe gekennzeichnet.

Wiederholtes Anwenden einer Konvexkombination von verschiedenen $q_{j,k}$ liegt mit den gleichen Argumenten wieder in $[P]$. Da der Quaternionenspline nach Definition 3.2 der B–Splines durch iteratives Anwenden von Konvexkombinationen der Kontrollpunkte entsteht, liegt somit auch $\mathscr{S}_{m,\mathbb{S}} P_{\boldsymbol{r}}$ in $[P]$. □

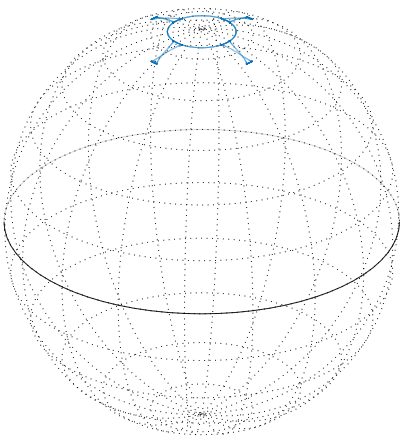
Korollar 5.15 *Ein sphärischer Schoenberg–Approximant mit Referenzpunkt in der konvexen Hülle der Datenpunkte ist variationsmindernd. Das bedeutet, er schneidet einen Großkreis nie öfter als sein sphärisches Kontrollpolygon.*

Eine Visualisierung des letzten Satzes liefert Abbildung 5.5. Hier sind in Abbildung 5.5(a) zwei sphärische Schoenberg–Approximanten angegeben mit Referenzpunkten innerhalb bzw. außerhalb der konvexen Hülle der Datenpunkte. In Abbildung 5.5(b) ist zu erkennen, dass die Umkehrung des Satzes 5.14 nicht gilt.

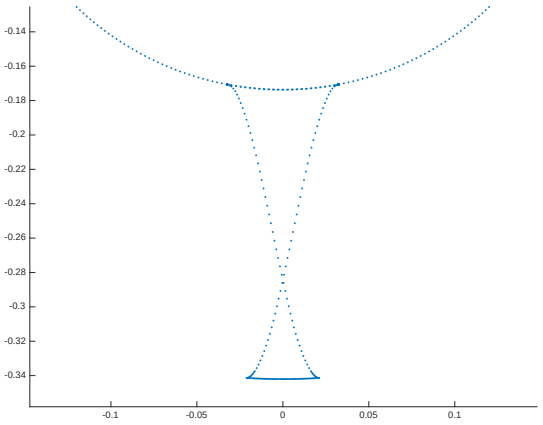
Der Satz 5.14 ist von hohem praktischem Nutzen, da hierdurch eine gewisse Kontrolle des Approximanten gegeben ist. Man kann im Vorfeld schon durch die konvexe Hülle bestimmen, wo der Approximant verlaufen wird oder was unter Umständen wichtiger sein kann, wo er nicht verlaufen wird. Als Anwendungsbeispiel sei hier das Umfangsfräsen genannt. Soll zum Beispiel die Wandfläche einer Tasche (Vertiefung) gefertigt werden, kann durch diese *Konvexe–Hülle–Eigenschaft* sichergestellt werden, dass der Fräser durch seine Orientierung niemals die Wandfläche (als Sollfläche) verletzt, vorausgesetzt, diese lässt sich durch eine konvexe Menge auf der Sphäre angeben.

5.2 Anwendung

Als Abschluss dieses Kapitels wird die Anwendung des vorgestellten Approximationsverfahrens für Orientierungsdaten an einem Teileprogramm vorgeführt. Zusätzlich wird das in Kapitel 4 vorgestellte MRA Verfahren auf den für die Approximation verwendeten Quaternionenspline angewandt.

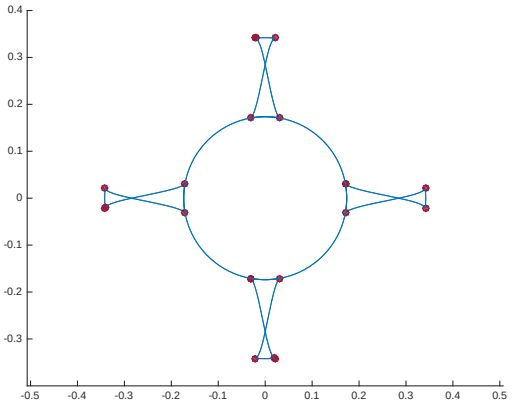


(a) Orientierungsdaten eines Teileprogramms auf der Sphäre.

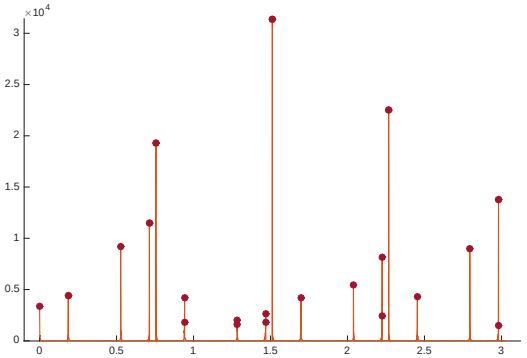


(b) Zweidimensionale Vergrößerung der Orientierungsdaten.

Abb. 5.6: Orientierungsdaten aus einem Teileprogramm mit einer entsprechenden Vergrößerung.



(a) Projektion der Orientierungsbahn.



(b) Analysespline zum Quaternionensplines.

Abb. 5.7: Orientierungsbahn in zweidimensionaler Projektion (links) und Analysespline des Quaternionensplines (rechts) mit jeweils markierten Merkmalspunkten.

5.2.1 Approximation und Merkmalsanalyse

In Abbildung 5.6 sind die Orientierungsvektoren aus einem Teileprogramm auf der Oberfläche einer Kugel sowie eine zweidimensionale Vergrößerung in xy–Ebene dargestellt. Ein sphärischer Schoenberg–Approximant dieser Daten mit markierten kritischen Stellen, die durch eine MRA Analyse des Quaternionensplines detektiert wurden, ist in Abbildung 5.7(a) zu sehen. Der entsprechende Analysespline mit ebenfalls markierten Merkmalspunkten ist in Abbildung 5.7(b) dargestellt.

5.2.2 Korrekturmöglichkeit

Im Gegensatz zur kartesischen Bahn ist es bei der Orientierungsbahn nicht ganz so offensichtlich wie die Orientierungsdaten korrigiert werden können, so dass ein besseres Verhalten der Werkzeugmaschine erreicht wird. Was ein „besseres“ Verhalten ist, muss zunächst festgelegt werden. Grundsätzlich ist eine glatte Bahnführung wünschenswert, da hierdurch Sprünge in Geschwindigkeit, Beschleunigung und Ruck der jeweiligen Achsen vermieden werden. Dies gilt sowohl für

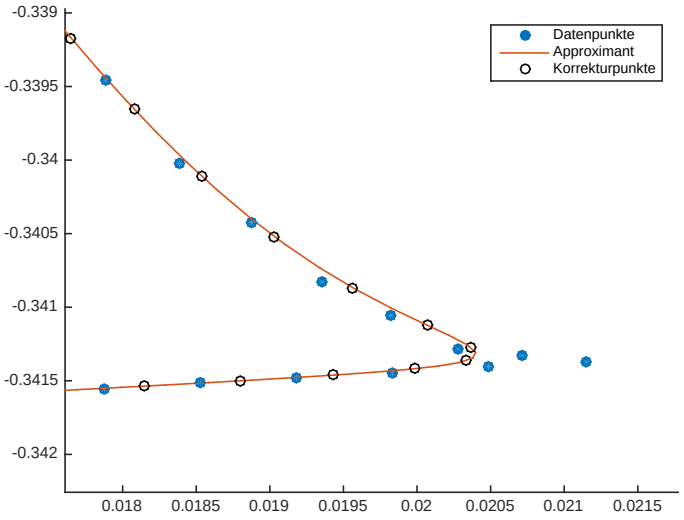


Abb. 5.8: Vergrößerung der Orientierungsbahn aus Abbildung 5.6. Auf dem Approximanten sind mögliche Korrekturpunkte zum ersetzten der aktuellen Daten im Teileprogramm markiert. Hierdurch soll eine glattere Orientierungsbahn entstehen.

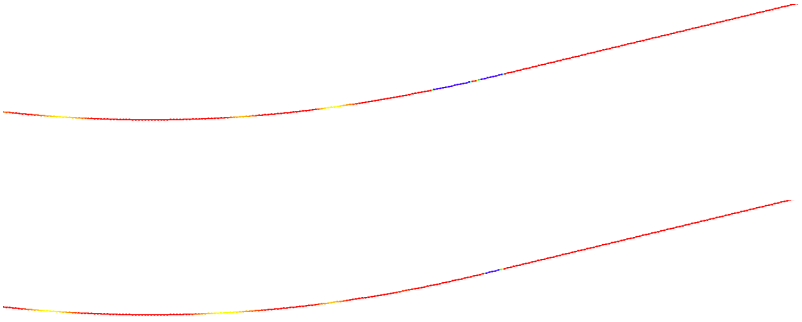


Abb. 5.9: Geschwindigkeitsprofil farbig auf der Bahn dargestellt. Vor (oben) und nach (unten) der Korrektur der Orientierung.

die kartesische Bahn als auch für die Orientierung des Werkzeugs. An den als kritisch erkannten Stellen in der Orientierungsbahn, siehe Abbildung 5.7(a), könnte man für einen glatteren Orientierungsverlauf eine entsprechende Abtastung des Approximanten wählen, siehe Abbildung 5.8. Bei einem solchen Vorgehen ist insbesondere darauf zu achten, dass durch eine veränderte Orientierung des Fräsers keine Kollision mit der zu fertigenden Geometrie erzeugt wird. Außerdem kann die Orientierung auf mehreren Sätzen im Teileprogramm konstant bleiben. Das bedeutet, während die Position des Fräsers verändert wird, ändert sich dessen Orientierung nicht. Somit sind insbesondere Stillstände in der Orientierung möglich. Dies ist bei der Approximation bzw. bei der Abtastung eines Approximanten zu beachten. Selbstverständlich kann auch der umgekehrte Fall eintreten. Während der Fräser an einer Position stehen bleibt, wird die Orientierung geändert. Solche Stillstände haben oft zur Folge, dass auf der Oberfläche des Werkstücks Marken oder andere, sich vom üblichen Fräsbild absetzende Effekte, auftreten. Diese sollten vermieden werden. Abbildung 5.9 zeigt den passenden Ausschnitt der Bahn zur Orientierung aus Abbildung 5.8. Das Geschwindigkeitsprofil ist farbig auf der Bahn dargestellt. Hierbei steht rot für schnell und blau für langsam. Wie man erkennen kann, ist der Geschwindigkeitsverlauf durch die neue Orientierung gleichmäßiger geworden.

6 Schlussbemerkung

6.1 Zusammenfassung

Ziel der vorliegenden Dissertation war es, ein Analyseverfahren für diskrete Daten (Teileprogramme) zu entwickeln, welches in der Lage ist, Merkmale der zugrundeliegenden Geometrie zu lokalisieren. Der Anwendungsbereich liegt hierbei im Umfeld von numerisch gesteuerten Werkzeugmaschinen. Das Verfahren sollte somit auf einer numerischen Steuerung realisierbar sein. Des Weiteren soll eine Korrektur dieser Daten durchgeführt werden, so dass die Fehler der Diskretisierung an den geometrischen Merkmale der Werkstückoberfläche durch die Ergänzungen geringer werden. Für diese Korrektur wird die analytische Form der Werkstückoberflächen aus dem CAD–System genutzt.

Da in einer numerischen Steuerung die diskreten Daten zur Bahnplanung üblicherweise durch Polynome bzw. Splines approximiert werden, wurde in Kapitel 4 ein Verfahren vorgestellt, welches die Merkmalspunkte der zunächst diskreten Daten durch Analyse von approximierenden Splines detektiert. Zur Analyse wird eine waveletähnliche Multiresolution–Zerlegung dieser Approximanten durch Knotenentfernen bzw. –einfügen erzeugt. Die Besonderheit dabei lag darin, dass die jeweiligen Approximanten mit nichtäquidistanten Knotenfolgen erzeugt wurden. Es mussten die Abklingraten der Waveletkoeffizienten bestimmt werden, um vergleichbare Skalierungen in den verschiedenen Auflösungsstufen der Koeffizienten zu erhalten. Des Weiteren wurde ein stabiles und effizientes Verfahren zur Auswertung des durch das Multiresolution Verfahren erhaltenen Analysesplines vorgestellt. Für dieses Verfahren war es notwendig die Spitzen des Analysesplines zu klassifizieren. Es wurde auf ein Verfahren aus dem Bereich der algebraischen Topologie zurückgegriffen, der „persistent homology“. Hierbei werden die topologischen Eigenschaften des Analysesplines ausgezeichnet. Zusätzlich zu dieser globalen Größe war es notwendig, die lokale Dominanz einer Spitze des Analysesplines zu berücksichtigen. Durch die Möglichkeit auf die Originalgeometrie aus dem CAD–System zurückgreifen zu können, kann eine Korrekturpunktberechnung unter Berücksichtigung diverser technologischer Parameter wie z. B. Werkzeugparameter, Bearbeitungsstrategie usw. durchgeführt werden.

Eine weitere Herausforderung war es, dieses Verfahren, welches zunächst nur für die Position eines Werkzeugs vorgesehen war, für die Orientierung des Werkzeugs zu adaptieren. Hierfür war es notwendig, die Orientierung des Werkzeugs mit Hilfe einer Splinekurve darzustellen. In Kapitel 5 wurde dies mit Hilfe spezieller sphärischer Splinekurven erarbeitet. Diese Kurven können mit Hilfe von Quaternionen als Splinekurven in einem vierdimensionalen Vektorraum dargestellt werden. Es wurde eine Verfahren entwickelt, eine krümmungsoptimierte Knotenfolge für solche Kurven zu generieren. Somit ist das erarbeitete Analyseverfahren auf diese Kurven anwendbar. Zusätzlich wurde eine Konvexe–Hülle–Eigenschaft für diese Kurven auf der Sphäre bewiesen. Diese Eigenschaft hat einen hohen praktischen Nutzen. Hierdurch kann sichergestellt werden, dass sich ein Fräser, zum Beispiel beim Umfangsfräsen, immer innerhalb einer vorher definierten Hülle befindet.

6.2 Fazit und Ausblick

Das vorgestellte Verfahren zur Analyse und Korrektur von Teileprogrammen kann im präparativen Teil einer numerischen Steuerung realisiert werden. Es ist denkbar, dass das Verfahren auch als nachgelagerter Prozess eines CAM–Systems direkt bei der Erzeugung eines Teileprogramms umgesetzt werden kann. Da allerdings davon auszugehen ist, dass auf einer numerischen Steuerung Teileprogramme aus vielen verschiedenen CAM–Systemen laufen, ist eine Realisierung auf der numerischen Steuerung, das heißt unabhängig vom CAD–System, zu empfehlen. Für die Korrektur eines Teileprogramms, die analytische Form der Werkstückoberfläche aus dem CAD–System zu Hilfe zu nehmen, ist aus heutigen Gesichtspunkten sinnvoll, da durch die fortschreitende Digitalisierung von Produktionsanlagen die Produktionsmaschinen und insbesondere auch numerische Steuerungen immer vernetzter werden. Dementsprechend stehen einer numerischen Steuerung zunehmend mehr Daten zur Verfügung.

Das präsentierte Multiresolution Verfahren basiert im Wesentlichen auf dem Entfernen von Knoten aus der Knotenfolge des Splines. Die in dieser Arbeit verwendete Strategie jeden n–ten Knoten aus der Knotenfolge zu entfernen hat sich als geeignet erwiesen. Dennoch besteht an dieser Stelle potential eine geignetere Strategie des Knotenentfernens zu verfolgen. Hier könnten weitere mögliche Arbeiten anschließen.

Wie die Korrektur der Positionsdaten aussieht bzw. wie das Fräsergebnis nach einer Korrektur auszusehen hat, ist intuitiv klar. Bei der Korrektur von Orientierungsdaten ist dies nicht so. In dieser Arbeit wurde als Korrekturmöglichkeit für einen glatteren Orientierungswechsel eine Abtastung des Orientierungsspline vorgeschlagen. An dieser Stelle gibt es noch Möglichkeiten weitere Korrekturen der Orientierungsdaten anzugeben. Zum Beispiel unter Berücksichtigung von Ausgleichsbewegungen anderer Achsen.

Das Ziel dieser Arbeit wurde erreicht und die Bearbeitungsqualität an den entsprechenden Stellen wurde erheblich verbessert.

Symbolverzeichnis

Allgemein

$\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$	Mengen der *natürlichen, ganzen, rationalen* und *reellen* Zahlen, wobei $\mathbb{N} \cup \{0\} =: \mathbb{N}_0$.
$\mathbb{H}$	Menge der Quaternionen (Definition 3.34, S. 30)
$\wedge$, $\vee$	logisches *und*, logisches *oder*
!	Fakultät: $n! = n \cdot (n-1) \cdot \ldots \cdot 2 \cdot 1$, $n \in \mathbb{N}$
$\binom{n}{k}$	Binomialkoeffizient $\binom{n}{k} := \frac{n!}{k!(n-k)!}$ für $n, k \in \mathbb{N}$ mit $n > k$
Π_n	Menge aller Polynome vom Höchstgrad n
$C(I)$	Menge aller stetigen Funktionen über I, $\{f : I \to \mathrm{R} \mid f \text{ stetig}\}$
$C^k(I)$	Menge aller k-fach stetig differenzierbaren Funktionen über dem Intervall I
$\boldsymbol{I}$, $\boldsymbol{I}_n$	Einheitsmatrix aus $\mathbb{R}^{n \times n}$
$\boldsymbol{e}_j$	j-ter Einheitsvektor (meist Spaltenvektor)
$\boldsymbol{D}_n$	Differenzenmatrix (siehe (3.16), S. 16)
δ_{ij}	Kronecker–Delta $\delta_{ij} := \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$
$\frac{\mathrm{d}^k}{\mathrm{d}x^k}$, $\frac{\mathrm{d}^k}{\mathrm{d}x^k} f(x)$	Differentialoperator
$\frac{\partial}{\partial x_j}$, $\frac{\partial}{\partial x_j} f(\boldsymbol{x})$	Partielle Ableitung zur j-ten Komponente
$SO(3)$	Drehgruppe

Splines

T, $T_{n,m}$	Knotenfolge (Definition 3.1, S. 11)
η, η_j	Vektor mit Vielfachheit der Knoten einer Knotenfolge, Vielfachheit des j-ten Knoten (siehe (3.3), S. 11)
$\mathcal{I}_m(T_{n,m})$	Intervall über die inneren Knoten von T (siehe (3.6), S. 13)
$N_j^m(\cdot \mid T)$	j-ter B–Spline vom Grad m zum Knotenvektor T (Definition 3.2, S. 12)
$\boldsymbol{N}_m$, $\boldsymbol{N}_{m,T}(t)$	B–Spline Kollokationsmatrix (Definition 3.14, S. 15)

$\mathbb{S}_m(T)$	Splineraum (Definition 3.10, S. 14)
$\boldsymbol{d}$	Koeffizienten einer Splinekurve spaltenweise arrangiert zu einer Matrix (Definition 3.12, S. 14)
d_j	j-ter Koeffizient einer Splinekurve (Definition 3.12, S. 14)
$s_m\boldsymbol{d}$, $s_m\boldsymbol{d}(\,\cdot \mid T)$	Splinekurve vom Grad m zum Knotenvektor T mit Koeffizienten $\boldsymbol{d}$ (Definition 3.12, S. 14)
$\hat{\boldsymbol{t}}$	Greville–Abszissen (Definition 3.16, S. 16)
$\Delta_m T$	Knotendifferenzmatrix (siehe (3.17), S. 16)
$\boldsymbol{G}_k$	Ableitungsmatrix einer Splinekurve (siehe (3.19), S. 16)
$\boldsymbol{A}$, $\boldsymbol{A}(\tau)$	Knoteneinfügematrix (Definition 3.28, S. 24)
$\boldsymbol{B}$, $\boldsymbol{B}(\tau)$	Knotenentfernmatrix (Definition 3.30, S. 27)
$\mathscr{S}_m$, $\mathscr{S}_m f$	Schoenberg–Operator (Definition 3.20, S. 18)
λ_{jm}, $\lambda_{jm} f$	de Boor–Fix Funktional (Definition 3.24, S. 21)
$B_j^m(\,\cdot\,)$	j–tes Bernsteinbasispolynom vom Grad m (Definition 3.8, S. 14)
$\langle\,\cdot,\cdot\,\rangle$	Skalarprodukt

Abbildungsverzeichnis

1.1 Simulation einer Fräsoberfläche 2

2.1 CAD/CAM/CNC–Prozesskette 5
2.2 Teileprogramm 6
2.3 Siemens Steuerung 7
2.4 Ablauf NCK 8
2.5 Geometrie eines Werkstücks mit problematischer innen liegenden Ecke. 9
2.6 Fräsbahnen über der innen liegenden Ecke der Geometrie aus Abbildung 2.5. . . 9

3.1 B–Splines 12
3.2 Splinekurve 15
3.3 de Boor–Algorithmus 18
3.4 B–Splines 23
3.5 Knoteneinfügen 25
3.6 Knotenentfernen 27
3.7 Tensor–B–Splines 29
3.8 Tensorproduktsplinefläche 29
3.9 Drehung mit Quaternion 32

4.1 Sehnenfehler zweier aufeinanderfolgende Punkte. 35
4.2 Knotenverteilung 37
4.3 Zerlegung eines Splines 42
4.4 Summe der Detailsplines. 43
4.5 Schematische Struktur der Waveletmatrix $(\boldsymbol{I} - \boldsymbol{BA})$. 48
4.6 Separierte Knotenentfernstrategie, ein B–Spline ist nicht betroffen. 48
4.7 Typische Struktur der Spalten von $(\boldsymbol{I} - \boldsymbol{BA})$. 49
4.8 Summe der Detailsplines 50
4.9 Drei Iterationen des Algorithmus 4.8. 53
4.10 Darstellung der Subniveaumenge $\mathbb{R}_y$. 54
4.11 Persistenz–Diagramm einer Splinefunktion 54
4.12 Persistenz–Diagramm zweier Splinefunktionen. 55
4.13 Dominanz $\mathcal{D}$ eines Maximums. 56
4.14 Summe der Detailsplines, Spitze mit kleiner Dominanz 56
4.15 Persistenz– und Dominanz–Histogramm 58
4.16 Bahn und Analysespline mit Merkmalspunkten 59
4.17 Teileprogramm 60
4.18 Bahn und Analysespline 60
4.19 Geometrie mit Merkmalspunkten 60
4.20 Fräsertypen 62
4.21 Fräsbahnen und Simulation 67

5.1 $\log_r$– und $\exp_r$–Abbildungen mit $r = (0, 0, 1)^T$. 72
5.2 Schätzung der Krümmung am Datenpunkt $\boldsymbol{p}_j$. 76

5.3 Sphärische Schoenberg Approximanten mit unterschiedlichen Referenzpunkten. . 79
5.4 Konvexkombination der Quaternionen $\mathcal{Q}_{\boldsymbol{r}}(\boldsymbol{a})$ und $\mathcal{Q}_{\boldsymbol{r}}(\boldsymbol{b})$ auf der Sphäre. 84
5.5 Sphärische Splines und konvexe Hülle 85
5.6 Orientierungsdaten auf der Sphäre 86
5.7 Orientierungsbahn und Analysespline 86
5.8 Korrektur Orientierungsbahn 87
5.9 Geschwindigkeits 87

Literaturverzeichnis

[1] Barlow, Jesse L.: 9 Numerical aspects of solving linear least squares problems. In: *Computational Statistics* Bd. 9. Elsevier, 1993. – ISSN 0169–7161, S. 303 – 376

[2] Barry, Phillip J. ; Goldman, Ronald N.: A Recursive Evaluation Algorithm for a Class of Catmull-Rom Splines. In: *SIGGRAPH Comput. Graph.* 22 (1988), Juni, Nr. 4, S. 199–204. – ISSN 0097–8930

[3] Bernstein, Sergei N.: Démonstration du théorème de Weierstrass fondée sur le calcul des probabilités. In: *Comm. Kharkov Math. Soc.* 13 (1912), S. 1–2

[4] Bock, Marco: *Steuerung von Werkzeugmaschinen mit redundanten Achsen.* Gießen, Justus-Liebig-Universität Gießen, Dissertation, August 2010

[5] Boehm, Wolfgang: Inserting new knots into B-spline curves. In: *Computer-Aided Design* 12 (1980), Nr. 4, S. 199 – 201. – ISSN 0010–4485

[6] Braess, Dietrich: Chebyshev approximation by spline functions with free knots. In: *Numerische Mathematik* 17 (1971), Nr. 5, S. 357–366. – ISSN 0029–599X

[7] Buss, Samuel R. ; Fillmore, Jay P.: Spherical Averages and Applications to Spherical Splines and Interpolation. In: *ACM Trans. Graph.* 20 (2001), April, Nr. 2, S. 95–126. – ISSN 0730–0301

[8] Chui, Charles ; Chui, Charles (Hrsg.): *An Introduction to Wavelets.* Elsevier, 1992 (Academic Press). – 266 S. – ISBN 978–0–12–174584–4

[9] Coeurjolly, David ; Miguet, Serge ; Tougne, Laure: Discrete Curvature Based on Osculating Circle Estimation. In: *Proceedings of the 4th International Workshop on Visual Form.* London, UK, UK : Springer-Verlag, 2001 (IWVF-4). – ISBN 3–540–42120–3, S. 303–312

[10] Cohen, Elaine ; Lyche, Tom ; Riesenfeld, Richard: Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics. In: *Computer Graphics and Image Processing* 14 (1980), Nr. 2, S. 87 – 111. – ISSN 0146–664X

[11] Cohen-Steiner, David ; Edelsbrunner, Herbert ; Harer, John: Stability of Persistence Diagrams. In: *Discrete & Computational Geometry* 37 (2006), Nr. 1, S. 103–120. – ISSN 1432–0444

[12] Curtis, Morton L.: *Matrix Groups.* 2nd. ed. Springer-Verlag New York, 1984. – ISBN 978–0–387–96074–6

[13] Dahmen, Wolfgang ; Reusken, Arnold: *Numerik für Ingenieure und Naturwissenschaftler.* Springer-Verlag Berlin Heidelberg, 2008. – xviii + 633 S. – ISBN 978–3–540–76492–2

[14] Dam, Erik B. ; Koch, Martin ; Lillholm, Martin: Quaternions, interpolation and animation / DIKU. 1998 (5). – Forschungsbericht. –

[15] Daubechies, I.: *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992. – xix + 350 S. – ISBN 978–0–89871–274–2

[16] de Boor, Carl: Good Approximation by Splines with Variable Knots. In: Meir, A. (Hrsg.) ; Sharma, A. (Hrsg.): *Spline Functions and Approximation Theory* Bd. 21. Birkhäuser, Basel, 1973. – ISBN 978–3–0348–5980–6, S. 57–72

[17] de Boor, Carl: *Splinefunktionen*. Basel [u.a.] : Birkhäuser, Basel, 1990 (Lectures in Mathematics, ETH Zürich). – ISBN 0–8176–2514–3

[18] de Boor, Carl: *A practical guide to splines*. Rev. ed. New York, NY: Springer, 2001. – xviii + 346 S. – ISBN 0–387–95366–3

[19] de Boor, Carl ; Fix, G. J.: Spline approximation by quasiinterpolants. In: *Journal of Approximation Theory* 8 (1973), Nr. 1, S. 19 – 45. – ISSN 0021–9045

[20] Deuflhard, Peter ; Hohmann, Andreas: *Numerische Mathematik 1 Eine algorithmische Einführung*. De Gruyter, 2008. – ISBN 978–3–11–020355–4

[21] Edelsbrunner, Herbert ; Letscher, David ; Zomorodian, Afra: Topological Persistence and Simplification. In: *Discrete & Computational Geometry* 28 (2002), Nr. 4, S. 511–533. – ISSN 1432–0444

[22] Farin, Gerald ; Farin, Gerald (Hrsg.): *Curves and Surfaces for CAGD (Fifth Edition)*. Fifth Edition. San Francisco : Morgan Kaufmann, 2002 (The Morgan Kaufmann Series in Computer Graphics). – ISBN 978–1–55860–737–8

[23] Farin, Gerald ; Hoschek, Josef ; Kim, Myung-Soo: *Handbook of Computer Aided Geometric Design*. Amsterdam : North-Holland, 2002. – ISBN 978–0–444–51104–1

[24] Goldman, Ronald N. ; Lyche, Tom ; Goldman, Ronald N. (Hrsg.) ; Lyche, Tom (Hrsg.): *Knot Insertion and Deletion Algorithms for B-Spline Curves and Surfaces*. Society for Industrial and Applied Mathematics, 1993. – ISBN 0–89871–306–4

[25] Golub, Gene H.: Numerical methods for solving linear least squares problems. In: *Numerische Mathematik* 7 (1965), Nr. 3, S. 206–216. – ISSN 0029–599X

[26] Golub, Gene H. ; Ortega, James M.: *Scientific Computing: An Introduction with Parallel Computing*. San Diego, CA, USA : Academic Press Professional, Inc., 1993. – ISBN 0–12–289253–4

[27] Hamm, Carsten ; Handeck, Jörg ; Sauer, Tomas: Spline multiresolution and wavelet-like decompositions. In: *Computer Aided Geometric Design* 31 (2014), Nr. 7–8, S. 521 – 530. – ISSN 0167–8396

[28] Higham, Nicholas J.: *Accuracy and Stability of Numerical Algorithms*. Second. Society for Industrial and Applied Mathematics, 2002

[29] Hoschek, Josef ; Lasser, Dieter: *Grundlagen der geometrischen Datenverarbeitung (2. Aufl.)*. Teubner, 1992. – ISBN 978–3–519–12962–2

[30] Kerautret, B. ; Lachaud, J.-O. ; Naegel, B.: Comparison of Discrete Curvature Estimators and Application to Corner Detection. In: Bebis, George (Hrsg.) ; Boyle, Richard (Hrsg.) ; Parvin, Bahram (Hrsg.) ; Koracin, Darko (Hrsg.) ; Remagnino, Paolo (Hrsg.)

; Porikli, Fatih (Hrsg.) ; Peters, Jörg (Hrsg.) ; Klosowski, James (Hrsg.) ; Arns, Laura (Hrsg.) ; Chun, YuKa (Hrsg.) ; Rhyne, Theresa-Marie (Hrsg.) ; Monroe, Laura (Hrsg.): *Advances in Visual Computing* Bd. 5358. Springer Berlin Heidelberg, 2008. – ISBN 978–3–540–89638–8, S. 710–719

[31] Kief, Hans B. ; Roschiwal, Helmut A. ; Schwarz, Karsten: *CNC-Handbuch 2015/2016*. Hanser Fachbuchverlag, 2015. – ISBN 978–3–446–44090–6

[32] Kim, Myoung-Jun ; Kim, Myung-Soo ; Shin, Sung Y.: A General Construction Scheme for Unit Quaternion Curves with Simple High Order Derivatives. In: *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*. New York, NY, USA : ACM, 1995 (SIGGRAPH '95). – ISBN 0–89791–701–4, S. 369–376

[33] Kim, Myoung-Jun ; Kim, Myung-Soo ; Shin, Sung Y.: A Compact Differential Formula for the First Derivative of a Unit Quaternion Curve. In: *The Journal of Visualization and Computer Animation* 7 (1996), Nr. 1, S. 43–57. – ISSN 1099–1778

[34] Lay, Steven R.: *Convex Sets and Their Applications*. Dover Publications, 2007. – ISBN 978–0–486–45803–8

[35] Lyche, Tom ; Mørken, Knut: Knot Removal for Parametric B-spline Curves and Surfaces. In: *Comput. Aided Geom. Des.* 4 (1987), Nr. 3, S. 217–230. – ISSN 0167–8396

[36] Lyche, Tom ; Mørken, Knut: A Data-Reduction Strategy for Splines with Applications to the Approximation of Functions and Data. In: *IMA Journal of Numerical Analysis* 8 (1988), Nr. 2, S. 185–208

[37] Mallat, Stephane: *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. 3. Boston : Academic Press, 2008. – ISBN 0–1237–4370–2

[38] Marsden, Martin ; Schoenberg, Isaac J.: On Variation Diminishing Spline Approximation Methods. In: de Boor, Carl (Hrsg.): *I. J. Schoenberg Selected Papers*. Boston, MA : Birkhäuser Boston, 1988. – ISBN 978–1–4899–0433–1, Kapitel 12, S. 247–268

[39] Mørken, Knut ; Reimers, Martin: An unconditionally convergent method for computing zeros of splines and polynomials. In: *Mathematics of Computation* 76 (2007), S. 845–865

[40] Nielson, Gregory M.: Smooth Interpolation of Orientations. In: Thalmann, Nadia M. (Hrsg.) ; Thalmann, Daniel (Hrsg.): *Models and Techniques in Computer Animation*. Tokyo : Springer Japan, 1993. – ISBN 978–4–431–66911–1, Kapitel 8, S. 75–93

[41] Papiernik, Wolfgang: Architecture and Design of Modern CNC/Drive Systems. In: *Proceedings on PCIM-Conference* Bd. 29, 1996, S. 271..280

[42] Pottmann, Helmut ; Wallner, Johannes ; Huang, Qi-Xing ; Yang, Yong-Liang: Integral Invariants for Robust Geometry Processing. In: *Comput. Aided Geom. Des.* 26 (2009), Nr. 1, S. 37–60. – ISSN 0167–8396

[43] Qiao, Min: *Approximation of CNC Part Program within a Given Tolerance Band*. Gießen, Justus-Liebig-Universität Gießen, Dissertation, September 2010

[44] Sauer, Tomas: *Numerische Mathematik 1*. Justus-Liebig-Universität Gießen : Vorlesungsskript, Wintersemester 1999/2000

[45] Sauer, Tomas: *Splinekurven und -flächen in Theorie und Anwendung.* Justus-Liebig-Universität Gießen : Vorlesungsskript, Wintersemester 2007/08

[46] Schoenberg, Isaac J.: Cardinal interpolation and spline functions. In: *Journal of Approximation Theory* 2 (1969), Nr. 2, S. 167 – 206. – ISSN 0021–9045

[47] Schumaker, Larry: *Spline Functions: Basic Theory.* 3rd. Cambridge Mathematical Library, 2007. – ISBN 978–0–521–70512–7

[48] Schwetlick, Hubert ; Schütze, Torsten: Least squares approximation by splines with free knots. In: *BIT Numerical Mathematics* 35 (1995), Nr. 3, S. 361–384. – ISSN 0006–3835

[49] Shoemake, Ken: Animating Rotation with Quaternion Curves. In: *SIGGRAPH Comput. Graph.* 19 (1985), Juli, Nr. 3, S. 245–254. – ISSN 0097–8930

[50] Sturges, Herbert A.: The Choice of a Class Interval. In: *Journal of the American Statistical Association* 21 (1926), Nr. 153, S. 65–66

[51] Wagenführ, Dominik: *Visualisierung von Fräsfehlern auf Basis von CAD-Daten.* Erlangen, Justus-Liebig-Universität Gießen, Dissertation, April 2008

[52] Weck, Manfred: *Werkzeugmaschinen 1: Maschinenarten und Anwendungsbereiche.* 6. Springer Berlin Heidelberg New York, 2005 (VDI-Buch). – ISBN 3–540–22504–8