

# Optimierungsansätze bei der Interpolation verteilter Daten

Der Technischen Fakultät der  
Universität Erlangen-Nürnberg

zur Erlangung des Grades

DOKTOR-INGENIEUR

vorgelegt von

**Andreas Kolb**

Erlangen – 1995

Als Dissertation genehmigt von  
der Technischen Fakultät der  
Universität Erlangen-Nürnberg

Tag der Einreichung:	24. August 1995
Tag der Promotion:	22. November 1995
Dekan:	Prof. Dr. H. Stoyan
Berichterstatter:	Prof. Dr. H.-P. Seidel
	Prof. Dr. H. Pottmann

# Inhaltsverzeichnis

<b>Einleitung</b>	<b>1</b>
<b>1 Grundlagen</b>	<b>5</b>
1.1 Notation . . . . .	5
1.2 Kurven und Flächen . . . . .	5
1.3 Variationsprinzipien . . . . .	13
<b>2 Übersicht bekannter Interpolationsverfahren</b>	<b>17</b>
2.1 Methoden zur Lösung des funktionalen Interpolationsproblems . . . . .	17
2.1.1 Klassische Methoden . . . . .	18
2.1.2 Minimum Norm Netzwerke . . . . .	21
2.1.3 Das Verfahren von Arge, Dæhlen und Tveito . . . . .	23
2.2 Methoden zur Lösung des parametrischen Interpolationsproblems . . . . .	25
2.2.1 Nicht optimierende Verfahren . . . . .	26
2.2.2 Optimierende Verfahren . . . . .	31
2.2.3 Splineflächen mit beliebiger Topologie . . . . .	33
<b>3 Ein Ritz-Galerkin-Ansatz für das funktionale Interpolationsproblem</b>	<b>37</b>
3.1 Die Wahl der Knotenvektoren . . . . .	38
3.2 Der Variationsansatz . . . . .	40
3.3 Vergleich der funktionalen Interpolationsmethoden . . . . .	43
<b>4 Parametrische Interpolation basierend auf <math>G^1</math>-Kurvennetzen</b>	<b>53</b>
4.1 Das kubische $G^1$ -Kurvennetz . . . . .	53
4.1.1 Repräsentation des $G^1$ -Kurvennetzes . . . . .	54
4.1.2 Bestimmung der lokalen Parameter . . . . .	55
4.1.3 Optimierung des $G^1$ -Kurvennetzes . . . . .	56
4.1.4 Ergebnisse . . . . .	58

4.2	Ausfüllen des $G^1$ -Kurvennetzes . . . . .	62
4.3	Simultane Optimierung . . . . .	65
4.4	Ergebnisse . . . . .	67
<b>5</b>	<b>Parametrische Interpolation basierend auf <math>G^2</math>-Kurvennetzen</b>	<b>71</b>
5.1	Das quintische $G^2$ -Kurvennetz . . . . .	71
5.1.1	Repräsentation des $G^2$ -Kurvennetzes . . . . .	72
5.1.2	Bestimmung der lokalen Parameter . . . . .	74
5.1.3	Optimierung des $G^2$ -Kurvennetzes . . . . .	74
5.1.4	Ergebnisse . . . . .	76
5.2	Ausfüllen des $G^2$ -Kurvennetzes . . . . .	78
5.3	Simultane Optimierung . . . . .	80
5.4	Vergleich der parametrischen Interpolationsmethoden . . . . .	82
<b>6</b>	<b>Erweiterungen und Anwendungen</b>	<b>101</b>
6.1	Polyeder mit konvexen $n$ -eckigen Seitenflächen . . . . .	101
6.2	Glätten von Polyedern . . . . .	102
6.3	Interpolation gegebener Bézierflächen . . . . .	104
6.4	Ergebnisse . . . . .	105
<b>7</b>	<b>Implementation</b>	<b>111</b>
7.1	Objektorientiertes Design . . . . .	111
7.2	Die Spline-Tool Bibliothek . . . . .	112
7.3	Spezielle Klassen . . . . .	114
	<b>Zusammenfassung und Ausblick</b>	<b>117</b>
<b>A</b>	<b>Mathematischer Anhang</b>	<b>119</b>
A.1	Globale Lösbarkeit für den Ritz-Galerkin-Ansatz . . . . .	119
A.2	Lösbarkeit der Kurvennetz-Optimierungsprobleme . . . . .	119
A.2.1	Lösbarkeit für das $G^1$ -Kurvennetz . . . . .	120
A.2.2	Lösbarkeit für das $G^2$ Kurvennetz . . . . .	122
A.3	Transformation der inneren Produktmatrix . . . . .	125
<b>B</b>	<b>Tabellen</b>	<b>127</b>

# Einleitung

Methoden zur Interpolation verteilter Daten haben eine große Bedeutung in den CAD/CAM-basierten Technologien, wie in der Automobilindustrie, in der Luft- und Raumfahrtindustrie und in der Medizintechnik (vgl. [LMD92, Nie93]). Wir beschränken uns hier auf den Fall, daß die Menge der Interpolationspunkte bereits mit Nachbarschaftsinformationen versehen ist, wobei wir konkret davon ausgehen, daß die gegebenen Daten in *Polyederform* vorliegen. Die Kanten des Polyeders beschreiben hierbei die Nachbarschaft zwischen einzelnen Interpolationspunkten. Verschiedene Verfahren zur Generierung von Nachbarschaftsbeziehungen ausgehend von einer 3D Punktemenge finden sich in [HDD<sup>+</sup>92, HDD<sup>+</sup>93, EM94, Vel94].

Bei der Betrachtung des Interpolationsproblems wird zwischen dem *funktionalen* und dem *parametrischen* Fall unterschieden. Der funktionale Fall zeichnet sich dadurch aus, daß sich das Polyeder mit Hilfe einer 2D-Triangulierung und einer Liste von reellen Zahlen, die den Abstand der Interpolationspunkte von der Parameterebene beschreiben, darstellen läßt. Der parametrische Fall umfaßt alle anderen Situationen.

Der funktionale Fall weist, im Gegensatz zur parametrischen Situation, mehr Struktur auf: Die Interpolationsfläche kann durch eine *globale* Parametrisierung über der 2D-Triangulierungsebene dargestellt werden. Im parametrischen Fall hingegen muß mit *lokalen* Flächenrepräsentationen gearbeitet werden, die in ihrer Gesamtheit die Interpolationsfläche beschreiben.

Zur Lösung des funktionalen Interpolationsproblems gibt es eine Vielzahl von Verfahren, die sich in ihrer Funktionsweise stark unterscheiden. Im parametrischen Fall wird hingegen meist folgende Zweischrithmethode zur Generierung der Interpolationsfläche verwendet:

**Kurvennetz:** In einem ersten Schritt wird ein interpolierendes Kurvennetz definiert, wobei jede Kante des Polyeders durch eine Raumkurve ersetzt wird. In den Interpolationspunkten, in denen mehrere Kurven zusammenkommen, müssen Übergangsbedingungen (z.B. gemeinsame Tangentialebene) eingehalten werden.

**Füllen des Kurvennetzes:** Im zweiten Schritt werden die durch das Kurvennetz berandeten Öffnungen durch Flächenstücke so ausgefüllt, daß die Randkurven interpoliert werden und benachbarte Flächenstücke glatt aneinander anschließen.

In beiden Situationen, im funktionalen wie im parametrischen Fall, unterscheiden wir zwischen *optimierenden* und *nicht optimierenden* Verfahren. Optimierende Methoden zeichnen

sich durch Verwendung eines Optimierungsansatzes mittels eines Funktionals zur Bewertung von Kurven- und Flächeneigenschaften (wie z.B. der Krümmung) aus. Ziel dieser Verfahren ist es, Parameter die nicht durch die Interpolations- und Glattheitsforderungen festgelegt sind, so zu bestimmen, daß die Kurven bzw. Flächen ein möglichst günstiges Verhalten hinsichtlich der durch das Funktional beschriebenen Eigenschaft besitzen. Nicht optimierende Methoden verwenden im Gegensatz dazu *heuristische Regeln* zur Festlegung dieser freien Parameter.

Die Verwendung *quadratischer* Funktionale spielt für die optimierenden Methoden eine große Rolle, da die resultierenden Optimierungsprobleme auf ein lineares Gleichungssystem führen, das schnell gelöst werden kann. Der Einsatz nicht quadratischer Funktionale hingegen erfordert numerische Lösungsmethoden, die meist wesentlich längere Rechenzeiten mit sich bringen.

In der vorliegenden Arbeit werden zunächst einige der wichtigsten bekannten Verfahren zur Bestimmung von Interpolationsflächen beschrieben. Anschließend werden insgesamt drei neue Verfahren zur Interpolation verteilter Daten vorgestellt (Kapitel 3 - 5). Der Schwerpunkt liegt dabei in der Bereitstellung von Algorithmen, die, ausgehend von einem beliebigen 3D-Polyeder *ohne* Voraussetzung weiterer Informationen, wie beispielsweise Flächennormalen, *glatte* Interpolationsflächen, d.h. Flächen ohne unerwünschte Wellen und Knicke, generieren. Dabei sollen die Algorithmen mit einem *vertretbaren Zeitaufwand* arbeiten. Ein weiteres Ziel ist die Darstellung der Interpolationsfläche in einem CAD-kompatiblen Format, d.h. die resultierenden Flächen sollen im B-Spline bzw. im Bézier-Format vorliegen.

Im einzelnen gliedert sich die Arbeit wie folgt: In Kapitel 1 werden kurz die notwendigen Bezeichnungen und Grundlagen behandelt, die zum weiteren Verständnis wichtig sind. Kapitel 2 beschäftigt sich mit bekannten Verfahren zur Lösung des funktionalen und des parametrischen Interpolationsproblems.

In Kapitel 3 wird ein neues Verfahren zur Lösung des funktionalen Interpolationsproblems vorgestellt. Der hierbei verfolgte Ansatz beruht auf der Verwendung von TP-B-Splineflächen in Kombination mit einem Ritz-Galerkin Verfahren zum Glätten der Interpolationsfläche.

Eine neue Methode zur Lösung des parametrischen Interpolationsproblems wird in Kapitel 4 vorgestellt. Dieses Verfahren verwendet ein optimiertes  $G^1$ -Kurvennetz, das mit Hilfe von optimierten Splitflächen zu einer glatten Interpolationsfläche fortgesetzt wird.

Kapitel 5 befaßt sich mit einer weiteren neuen Methode zur Behandlung des parametrischen Interpolationsproblems. Hierbei wird ein optimiertes  $G^2$ -Kurvennetz verwendet, dessen dreieckige Öffnungen mit jeweils einer optimierten Bézier-Dreiecksfläche ausgefüllt werden.

Kapitel 6 zeigt eine Reihe von Anwendungen der beiden Verfahren aus Kapitel 4 und 5. Hierzu zählen insbesondere die Erweiterung dieser Methoden auf Polyeder mit konvexen,  $n$ -eckigen Seitenflächen, ein Approximationsansatz, sowie die Interpolation von vorgegebenen Bézierflächen.

Kapitel 7 gibt einen kurzen Einblick in die zugrundeliegende Implementation. Abschließende Bemerkungen sowie einen Ausblick auf mögliche weitere Untersuchungen im Bereich

der Interpolation verteilter Daten finden sich im Anschluß.

An dieser Stelle sei darauf hingewiesen, daß große Teile der Abschnitte 4.1 und 5.1 zur Repräsentation der Kurvennetze und deren Optimierung, sowie Teile der Abschnitte 4.2 und 5.2 über die Bestimmung der lokalen Interpolationsflächen bereits in [KPS95a] veröffentlicht wurden. Des weiteren wird in [KPS95b] Bezug auf den Inhalt des Abschnittes 4.3 genommen.

Abschließend möchte ich Herrn Professor Seidel für die gute Zusammenarbeit und Unterstützung bei dieser Arbeit danken. Ein weiterer Dank ergeht an Herrn Professor Pottmann und an Herrn Dr. Greiner für die vielen hilfreichen Anregungen und Gespräche. Ebenso danken möchte ich den Studenten und den Mitarbeitern am Lehrstuhl für Graphische Datenverarbeitung für die vielseitige Unterstützung und die Diskussionsbereitschaft.



# Kapitel 1

## Grundlagen

In diesem Kapitel werden die Notation und die grundlegenden Begriffe, welche in den nachfolgenden Kapiteln benötigt werden, eingeführt. Hierzu werden in Abschnitt 1.1 kurz die wichtigsten Notationen aufgelistet. In Abschnitt 1.2 wird dann auf Kurven und Flächen eingegangen. Abschnitt 1.3 gibt einen Einblick in die Variationsprinzipien, die den verschiedenen Optimierungsmethoden zugrundeliegen.

### 1.1 Notation

Die wesentlichsten Bezeichnungen, die sich durch die gesamte Arbeit durchziehen, sind hier überblicksartig aufgeführt:

$a, \alpha$	ein reeller, skalarer Wert
$\mathbf{A}$	ein Punkt im $\mathbb{R}^3$
$\mathbf{a}$	ein Punkt/Parameterwert im $\mathbb{R}^2$
$\vec{\mathbf{a}}$	ein Vektor im $\mathbb{R}^2$ bzw. im $\mathbb{R}^3$
$\hat{\mathbf{a}}$	ein normierter Vektor im $\mathbb{R}^2$ bzw. im $\mathbb{R}^3$
$\langle \cdot   \cdot \rangle$	das standard Skalarprodukt im $\mathbb{R}^2$ bzw. im $\mathbb{R}^3$
$\vec{\mathbf{a}} \times \vec{\mathbf{b}}$	das äußere oder Vektorprodukt im $\mathbb{R}^3$ .

Alle weiteren Begriffe werden an der Stelle eingeführt, an der sie benötigt werden.

### 1.2 Kurven und Flächen

Alle Verfahren, die in dieser Arbeit beschrieben werden, bauen auf bestimmten Arten von Kurven und Flächen auf. In diesem Abschnitt werden kurz die wesentlichen geometrischen Eigenschaften von Kurven und Flächen im allgemeinen eingeführt. Einige spezielle Klassen von Kurven werden anschließend kurz beschrieben. Ein weiterer Schwerpunkt liegt in der

Betrachtung von Übergangsbedingungen zwischen zwei oder mehreren Flächenstücken zur Erzeugung global „glatter“ Flächen.

## Geometrische Eigenschaften

Raumkurven können stets als *parametrische Kurven* über dem reellen Parameterintervall  $[a, b]$ ,  $a < b$  aufgefaßt werden:

$$\mathbf{C}(u) = \begin{pmatrix} C_1(u) \\ C_2(u) \\ C_3(u) \end{pmatrix}, \quad u \in [a, b].$$

Die wesentlichen geometrischen Größen einer Kurve  $\mathbf{C}$  sind (vgl. [dC92]):

$$\begin{aligned} \text{Tangente: } \hat{\mathbf{t}}_{\mathbf{C}} &= \frac{\mathbf{C}'}{\|\mathbf{C}'\|}, \\ \text{Krümmungsvektor: } \vec{\mathbf{k}}_{\mathbf{C}} &= \frac{\mathbf{C}' \times \mathbf{C}'' \times \mathbf{C}'}{\|\mathbf{C}'\|^4}, \\ \text{Krümmung: } \kappa_{\mathbf{C}} &= \|\vec{\mathbf{k}}\| = \frac{\|\mathbf{C}' \times \mathbf{C}''\|}{\|\mathbf{C}'\|^3} \text{ und} \\ \text{Torsion: } \tau_{\mathbf{C}} &= \frac{\det(\mathbf{C}', \mathbf{C}'', \mathbf{C}''')}{|\mathbf{C}' \times \mathbf{C}''|}. \end{aligned}$$

Flächen können, im Gegensatz zu Kurven, i.a. nicht global durch eine Parametrisierung beschrieben werden. Läßt sich eine Fläche global durch eine Parametrisierung darstellen,

$$\mathbf{S}(u, v) = \begin{pmatrix} S_1(u, v) \\ S_2(u, v) \\ S_3(u, v) \end{pmatrix}, \quad (u, v) \in \Omega \subseteq \mathbb{R}^2,$$

so wird sie als *parametrische Fläche* über dem Parametergebiet  $\Omega$  bezeichnet. Im allgemeinen kann man beliebige Flächen *lokal* als parametrische Flächen darstellen (vgl. [dC92]).

Die wichtigsten geometrischen Größen von Flächen sind (vgl. [dC92]):

$$\begin{aligned} \text{Normalenvektor}^1: \hat{\mathbf{n}}_{\mathbf{S}} &= \frac{\mathbf{S}_u \times \mathbf{S}_v}{\|\mathbf{S}_u \times \mathbf{S}_v\|}, \\ \text{Erste Fundamentalform: } \mathbf{I}_{\mathbf{S}} &= \begin{pmatrix} E & F \\ F & G \end{pmatrix} = \begin{pmatrix} \langle \mathbf{S}_u | \mathbf{S}_u \rangle & \langle \mathbf{S}_u | \mathbf{S}_v \rangle \\ \langle \mathbf{S}_u | \mathbf{S}_v \rangle & \langle \mathbf{S}_v | \mathbf{S}_v \rangle \end{pmatrix}, \\ \text{Zweite Fundamentalform: } \mathbf{II}_{\mathbf{S}} &= \begin{pmatrix} e & f \\ f & g \end{pmatrix} = \begin{pmatrix} \langle \mathbf{S}_{uu} | \hat{\mathbf{n}}_{\mathbf{S}} \rangle & \langle \mathbf{S}_{uv} | \hat{\mathbf{n}}_{\mathbf{S}} \rangle \\ \langle \mathbf{S}_{uv} | \hat{\mathbf{n}}_{\mathbf{S}} \rangle & \langle \mathbf{S}_{vv} | \hat{\mathbf{n}}_{\mathbf{S}} \rangle \end{pmatrix}. \end{aligned}$$

<sup>1</sup> $\mathbf{S}_u$  bezeichnet die partielle Ableitung in  $u$ -Richtung.

Die erste Fundamentalform  $l_{\mathbf{S}}(u, v)$  im Punkt  $(u, v)$  induziert eine Metrik auf der Tangentialebene  $\mathbf{T}_{\mathbf{S}(u,v)}(\mathbf{S})$  im Punkt  $\mathbf{S}(u, v)$ . Ist  $\vec{v} = (v_1, v_2) \neq 0$  ein Vektor im  $\mathbb{R}^2$ , so gilt für den Tangentialvektor  $\frac{\partial \mathbf{S}}{\partial \vec{v}}(u, v)$

$$\left\| \frac{\partial \mathbf{S}}{\partial \vec{v}}(u, v) \right\|^2 = (v_1, v_2) l_{\mathbf{S}}(u, v) \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}.$$

Die zweite Fundamentalform  $ll_{\mathbf{S}}$  dient vor allem der Bestimmung der *Normalenkrümmung*. In Richtung  $\hat{\mathbf{t}} = \frac{\partial \mathbf{S}}{\partial \vec{t}}(u, v)$  mit  $\vec{t} = (t_1, t_2)$  wird die Normalenkrümmung bestimmt durch:

$$\kappa_{\vec{t}} = \frac{(t_1, t_2) ll_{\mathbf{S}} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}}{(t_1, t_2) l_{\mathbf{S}} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}}.$$

Die durch die Matrix  $\mathcal{K}_{\mathbf{S}} = ll_{\mathbf{S}} \cdot l_{\mathbf{S}}^{-1}$  induzierte Abbildung wird auch *Weingartenabbildung* genannt. Die *Hauptkrümmungen*  $\kappa_1, \kappa_2$  ergeben sich als minimaler bzw. maximaler Wert von  $\kappa_{\hat{\mathbf{t}}}$ , wobei  $\hat{\mathbf{t}}$  bzgl. der Metrik in der Tangentialebene normiert ist:  $(t_1, t_2) l_{\mathbf{S}}(u, v) \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = 1$  für  $\hat{\mathbf{t}} = t_1 \mathbf{S}_u + t_2 \mathbf{S}_v$ . Die zugehörigen Richtungen  $\hat{\mathbf{e}}_1$  und  $\hat{\mathbf{e}}_2$  sind die *Hauptkrümmungsrichtungen*. Man erhält die Hauptkrümmungen und die Hauptkrümmungsrichtungen auch als Eigenwerte bzw. Eigenvektoren der Weingartenabbildung.

Neben den Hauptkrümmungen spielen die *Gaußkrümmung*  $K$  und die *mittlere Krümmung*  $H$  einer Fläche eine wichtige Rolle bei der Analyse von Flächen:

$$\mathcal{K}_{\mathbf{S}} = \kappa_1 \kappa_2 = \det(\mathcal{K}_{\mathbf{S}}) = \frac{eg - f^2}{EG - F^2}, \quad H_{\mathbf{S}} = \frac{\kappa_1 + \kappa_2}{2} = \frac{1}{2} \text{spur}(\mathcal{K}_{\mathbf{S}}) = \frac{eG - 2fF + gE}{2(EG - F^2)}.$$

Für viele Verfahren spielt die *Geometrische Kompatibilität* zwischen Kurven und Flächen in einem betrachteten Punkt eine große Rolle.

**Bezeichnung 1.2.1** (a) Die Kurve  $\mathbf{C}$  ist mit der Fläche  $\mathbf{S}$  im Punkt  $\mathbf{P} = \mathbf{C}(t_0) = \mathbf{S}(u_0, v_0)$   $G^k$ -kompatibel, falls es eine Kurve  $\bar{\mathbf{C}}$  auf  $\mathbf{S}$  gibt, die mit  $\mathbf{C}$  in  $\mathbf{P}$   $G^k$ -steig zusammentrifft. D.h. es gibt eine Kurve  $\mathbf{U} : [-\delta, \delta] \rightarrow \Omega$  mit  $\mathbf{U}(0) = (u_0, v_0)$ , so daß  $\mathbf{C}$  bei  $t = t_0$   $G^k$ -stetig an  $\mathbf{S} \circ \mathbf{U}$  bei  $t = 0$  anschließt.

(b) Die Kurven  $\mathbf{C}_1, \dots, \mathbf{C}_l$  sind im Punkte  $\mathbf{C}_i(t_{i_0})$   $G^k$ -kompatibel, falls es eine Fläche  $\mathbf{S}$  und einen Parameterwert  $(u_0, v_0)$  gibt, so daß  $\mathbf{C}_i, i = 1, \dots, l$  alle im Punkt  $\mathbf{C}_i(t_{i_0}) = \mathbf{S}(u_0, v_0)$  mit  $\mathbf{S}$   $G^k$ -kompatibel sind.

Folgende Charakterisierungen der  $G^k$ -Kompatibilität sind von praktischer Bedeutung. Die Kurve  $\mathbf{C}$  im Punkt  $\mathbf{P}_{\mathbf{C}} = \mathbf{C}(t_0)$  und die Fläche  $\mathbf{S}$  im Punkte  $\mathbf{P}_{\mathbf{S}} = \mathbf{S}(u_0, v_0), (u_0, v_0)$  sind

$G^0$ -kompatibel, falls  $\mathbf{P}_{\mathbf{C}} = \mathbf{P}_{\mathbf{S}}$ ,

$G^1$ -kompatibel, falls sie  $G^0$  kompatibel sind und  $\langle \hat{\mathbf{t}}_{\mathbf{C}} \mid \hat{\mathbf{n}}_{\mathbf{S}} \rangle = 0$ ,

$G^2$ -kompatibel, falls sie  $G^1$  kompatibel sind und  $\langle \vec{\mathbf{k}}_{\mathbf{C}} \mid \hat{\mathbf{n}}_{\mathbf{S}} \rangle = (t_1, t_2) ll_{\mathbf{S}} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} = \kappa_{\hat{\mathbf{t}}_{\mathbf{C}}}$ ,

wobei  $\hat{\mathbf{t}}_{\mathbf{C}} = t_1 \mathbf{S}_u + t_2 \mathbf{S}_v$ .<sup>2</sup>

## Repräsentation von Kurven und Flächen

In diesem Abschnitt werden kurz die Klassen von Kurven und Flächen angesprochen, die für die im weiteren beschriebenen Verfahren relevant sind. Dabei geht es hauptsächlich um die Bezeichnungsfindung. Details können in der einschlägigen Fachliteratur nachgelesen werden (vgl. [BBB87, HL92, Far93]).

Im Bereich der Geometrischen Modellierung haben sich vor allem zwei Kurvenschemata durchgesetzt: *Bézier*- und *B-Spline*-Kurven. Bézier-Kurven sind polynomielle Kurven in der *Bernstein-Bézier* Basisdarstellung mit den Basisfunktionen  $B_i^n(u) = \sum_{i=0}^n \binom{n}{i} u^i (1-u)^{n-i}$ ,  $i = 0, \dots, n$  vom Grad  $n$ . B-Spline-Kurven werden mittels einer stückweise polynomiellen Basis, der *B-Spline-Basis*, über dem Knotenvektor  $T = (t_0, \dots, t_{m+n+1})$ ,  $t_i \leq t_{i+1}$ ,  $i = 0, \dots, m+n$ ,  $t_0 < t_{n+1}$ ,  $t_m < t_{m+n+1}$ , definiert.  $N_i^n$ ,  $i = 0, \dots, m$  bezeichnet die *normierte* B-Spline-Basisfunktion vom Grad  $n$  mit Träger  $[t_i, t_{i+n+1}]$ . Die Funktionen  $N_0^n, \dots, N_m^n$  bilden über dem Intervall  $[t_n, t_{m+1}]$  eine Basis für die stückweisen Polynome von Grad  $n$  über  $T$ . Die Stetigkeit, mit der die einzelnen Polynome aneinander anschließen, hängt von der Vielfachheit der Knoten ab.

Bei den Flächen sind die auf den Bézier- und B-Spline-Kurven aufbauenden Tensorproduktflächen, kurz TP-Flächen, weit verbreitet. TP-Bézier- und TP-B-Splinefläche sind gegeben durch

$$\mathbf{S}(u, v) = \sum_{(i,j)=(0,0)}^{(n_u, n_v)} \mathbf{b}_{i,j} B_{i,j}^{n_u, n_v}(u, v) \quad \text{bzw.} \quad \mathbf{S}(u, v) = \sum_{(i,j)=(0,0)}^{(m_u, m_v)} \mathbf{b}_{i,j} N_{i,j}^{m_u, m_v}(u, v).$$

Hier bezeichnen  $B_{i,j}^{n_u, n_v}(u, v) = B_i^{n_u}(u) \cdot B_j^{n_v}(v)$  und  $N_{i,j}^{m_u, m_v}(u, v) = N_i^{m_u}(u) \cdot N_j^{m_v}(v)$  die entsprechenden TP-Basisfunktionen der Flächen.

Neben den TP-Flächen finden die *Dreiecks-Bézierflächen* häufig Anwendung. Diese Flächen basieren auf den *baryzentrischen Koordinaten*  $b_r, b_s, b_t$  eines Punktes  $\mathbf{u} \in \mathbb{R}^2$  bzgl. eines nicht entarteten Dreiecks  $\Delta(\mathbf{r}, \mathbf{s}, \mathbf{t}) \subseteq \mathbb{R}^2$ , definiert durch:  $\mathbf{u} = b_r \mathbf{r} + b_s \mathbf{s} + b_t \mathbf{t}$ ,  $b_r + b_s + b_t = 1$ . Die Dreiecks-Bézierfläche ergibt sich dann zu<sup>3</sup>:

$$\mathbf{S}(\mathbf{u}) = \sum_{|I|=n} \mathbf{b}_I B_I^\Delta(\mathbf{u}),$$

mit den Kontrollpunkten  $\mathbf{b}_I$ ,  $|I| = n$ , und den *Bernstein-Bézier* Basisfunktionen über dem Dreieck  $\Delta$

$$B_I^\Delta(\mathbf{u}) = \binom{n}{ijk} b_r^i b_s^j b_t^k.$$

Als weiterer Flächenklasse werden noch die sogenannten *transfiniten Interpolanten* diskutiert (vgl. [Nie79]). Zur Definition des transfiniten  $C^1$  Interpolanten geht man davon aus,

<sup>2</sup>Diese Bedingung ergibt sich aus dem *Satz von Meusnier* (vgl. [dC92]).

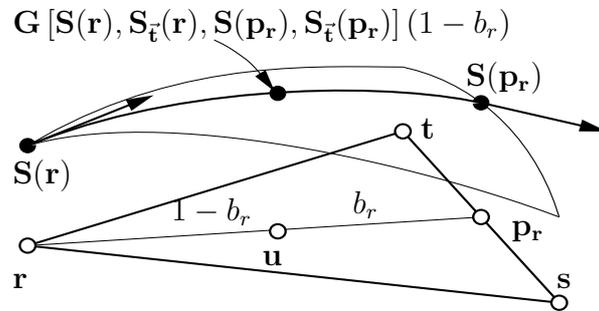
<sup>3</sup>Unter Verwendung der Multiindex-Schreibweise  $I = (i, j, k)$ ; hier bedeutet  $|I| = i + j + k$ .

daß über dem Rand des Dreiecks  $\Delta(\mathbf{r}, \mathbf{s}, \mathbf{t})$  die  $C^1$ -Daten einer zu bestimmenden Fläche  $\mathbf{S}$  gegeben sind. Sei  $\mathbf{G} \left[ \mathbf{P}_0, \vec{\mathbf{t}}_0, \mathbf{P}_1, \vec{\mathbf{t}}_1 \right]$  eine über  $[0, 1]$  parametrisierte Kurve, die für  $t \in \{0, 1\}$  die beliebig vorgegebene Position  $\mathbf{P}_0$  bzw.  $\mathbf{P}_1$  und Ableitung  $\vec{\mathbf{t}}_0$  bzw.  $\vec{\mathbf{t}}_1$  interpoliert (z.B. ein kubisches Hermitepolynom).  $\mathbf{G}$  wird im weiteren als *erzeugende Kurve* für die transfiniten Interpolanten bezeichnet. Für die Ecke  $\mathbf{r}$  des Dreiecks  $\Delta$  wird folgender Operator definiert:

$$D_r [\mathbf{S}] (\mathbf{u}) = \mathbf{G} \left[ \mathbf{S}(\mathbf{r}), \frac{\partial \mathbf{S}}{\partial \vec{\mathbf{t}}}(\mathbf{r}), \mathbf{S}(\mathbf{p}_r), \frac{\partial \mathbf{S}}{\partial \vec{\mathbf{t}}}(\mathbf{p}_r) \right] (1 - b_r),$$

mit  $\mathbf{u} = b_r \mathbf{r} + b_s \mathbf{s} + b_t \mathbf{t}, \mathbf{p}_r = \frac{b_s \mathbf{s} + b_t \mathbf{t}}{1 - b_r}$  und  $\vec{\mathbf{t}} = \mathbf{p}_r - \mathbf{r}$ .

Der Operator  $D_r [\mathbf{S}]$  interpoliert die gegebenen  $C^1$ -Daten auf der Kante  $\overline{\mathbf{st}}$ . In der Ecke  $\mathbf{r}$  werden Position und Richtungsableitungen von  $\mathbf{S}$  interpoliert. Mit Hilfe einer Konvexkombination unter Verwendung der Gewichtsfunktionen  $\omega_r = (b_s^2 b_t^2) / (b_r^2 b_s^2 + b_s^2 b_t^2 + b_t^2 b_r^2)$  (analog für  $\omega_s, \omega_t$ ) wird die Interpolationsfläche definiert:



$D [\mathbf{S}] = \omega_r D_r [\mathbf{S}] + \omega_s D_s [\mathbf{S}] + \omega_t D_t [\mathbf{S}]$ . Abbildung 1.1: Der Operator  $D_r [\mathbf{S}]$  ausgewertet an der Stelle  $\mathbf{u}$ .

$D [\mathbf{S}]$  interpoliert auf dem Rand des Dreiecks  $\partial \Delta$  Position und erste partielle Ableitungen der Fläche  $\mathbf{S}$ . In den Ecken besitzt  $D [\mathbf{S}]$  hebbare Singularitäten.

## Übergangsbedingungen für Bézierflächen

Dieser Abschnitt befaßt sich mit dem Problem der Konstruktion von „glatten“ Übergängen zwischen zwei Bézierflächen entlang einer vorgegebenen Kante. Dazu wird der Begriff „Glätte“ näher spezifiziert und verschiedene Methoden zur Erzeugung glatter Übergänge diskutiert. Das sogenannte *Eckenumschlußproblem* wird abschließend angesprochen. Dieses tritt auf, wenn man um einen Punkt mehrere Bézierflächen konstruieren möchte, so daß alle Übergänge glatt sind.

Zum Begriff der Glätte vgl. [Pip87, LH89, Pet91]. Sei  $\mathbf{S}_1$  und  $\mathbf{S}_2$  (TP- oder Dreiecks-) Bézierflächen.  $\mathbf{S}_1$  sei durch die Parameter  $u$  und  $v$ ,  $\mathbf{S}_2$  durch die Parameter  $w$  und  $u$  parametrisiert (vgl. Abbildung 1.2). Hierbei sei  $\mathbf{S}_1(u, 0) = \mathbf{S}_2(0, u)$ .

**Bezeichnung 1.2.2** (i) Die Flächen  $\mathbf{S}_1$  und  $\mathbf{S}_2$  sind an der gemeinsamen Kante *tangentialstetig*, falls es skalarwertige Gewichtsfunktionen  $\alpha, \beta$  und  $\gamma$  gibt, mit

$$\alpha(\mathbf{S}_1)_u + \beta(\mathbf{S}_1)_v + \gamma(\mathbf{S}_2)_w \equiv 0. \tag{1.1}$$

- (ii) Die Flächen  $\mathbf{S}_1$  und  $\mathbf{S}_2$  sind entlang der gemeinsamen Kante  $G^1$ -stetig, falls sie dort tangentialstetig sind und zudem gilt:

$$(\mathbf{S}_1)_u \times (\mathbf{S}_1)_v \neq 0 \text{ (Regularität)}, \quad \beta\gamma > 0 \text{ (Orientierung)}. \quad (1.2)$$

Peters [Pet89] zeigt, daß die Gewichtsfunktionen bis auf einen gemeinsamen Faktor Polynome mit folgendem Grad sind:

$$\begin{aligned} \deg(\alpha) &\leq \deg((\mathbf{S}_1)_v) + \deg((\mathbf{S}_2)_w), \\ \deg(\beta) &\leq \deg((\mathbf{S}_1)_u) + \deg((\mathbf{S}_2)_w), \\ \deg(\gamma) &\leq \deg((\mathbf{S}_1)_u) + \deg((\mathbf{S}_1)_v). \end{aligned}$$

Im folgenden werden einige Konstruktionsmöglichkeiten für tangentialstetige und für  $G^1$ -stetige Übergänge vorgestellt. Dazu sei die Randkurve  $\mathbf{S}_1(u, 0)$  durch die Kontrollpunkte  $\mathbf{H}_i$ , die Kontrollpunkte in der ersten Reihe für  $\mathbf{S}_1$  mit  $\mathbf{F}_i$ , für  $\mathbf{S}_2$  mit  $\mathbf{G}_i$  bezeichnet (vgl. Abbildung 1.2). Der Grad der von  $(\mathbf{S}_1)_u$ ,  $(\mathbf{S}_1)_v$  und  $(\mathbf{S}_2)_w$  sei  $n_c, n_1$  bzw.  $n_2$ . Je nach Darstellung kann die Randkurve von niedrigerem Grad sein als die Fläche  $\mathbf{S}_1$  oder  $\mathbf{S}_2$ .

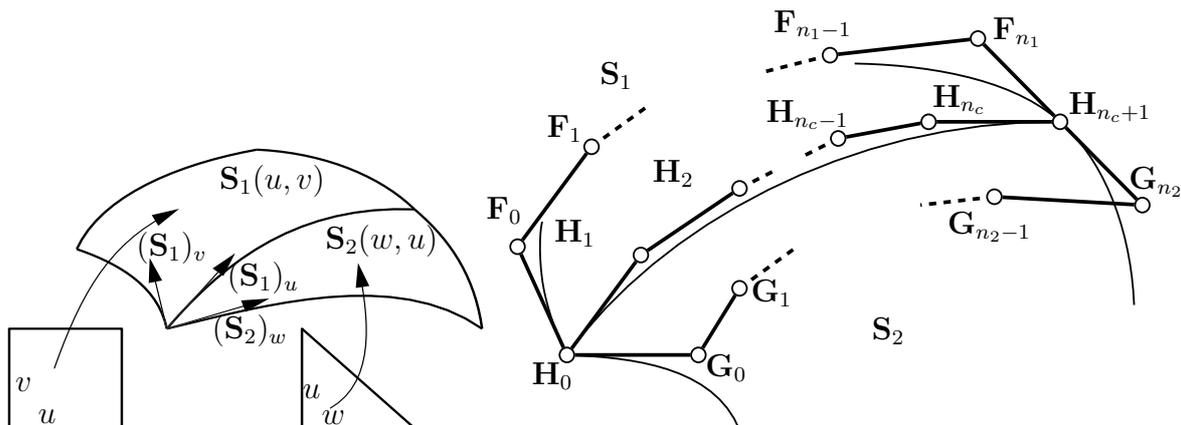


Abbildung 1.2: Tangentialstetigkeit bzw.  $G^1$ -Stetigkeit und die Kontrollpunkte für den Übergang zwischen den Flächen  $\mathbf{S}_1$  und  $\mathbf{S}_2$ .

### Farin [Far82]

Die  $G^1$ -Bedingungen nach Farin beziehen sich auf den Fall, daß  $\mathbf{S}_1(u, 0)$  eine Kurve vom Grad  $n$  und die Fläche  $\mathbf{S}_i, i = 1, 2$  entweder eine TP-Bézierfläche vom Grad  $(n, n)$  oder eine Dreiecks-Bézierfläche vom Grad  $n + 1$  ist. Damit ist  $n_c = n - 1$  und  $n_1 = n_2 = n$  gewährleistet.<sup>4</sup>

<sup>4</sup>Ist die Dreiecks-Bézierfläche  $\mathbf{S}(u, v)$  vom Grad  $n$  mit Kontrollpunkten  $\mathbf{b}_i^n$  und die Randkurve von  $\mathbf{S}$  entlang der Kante  $\overline{\mathbf{st}}$  vom Grad  $n - 1$  mit Kontrollpunkten  $\mathbf{b}_{0,i,n-i-1}^{n-1}$ , so ist die radiale Ableitung  $\mathbf{S}_{\vec{t}}$ ,  $\vec{t} = ((1-u)\mathbf{s} + u\mathbf{t}) - \mathbf{r}$  auf  $\overline{\mathbf{st}}$  durch  $n \sum_{i=0}^{n-1} (\mathbf{b}_{1,i,n-i-1}^n - \mathbf{b}_{0,i,n-i-1}^{n-1}) B_i^{n-1}(u)$ , gegeben.

Die Gewichtsfunktionen werden folgendermaßen angesetzt:  $\alpha(u) = (1-u)\alpha_0 + u\alpha_1$ , linear,  $\beta = \beta_0, \gamma = \gamma_0$  konstant. Da

$$\begin{aligned} (\mathbf{S}_1)_v &= k \sum_{i=0}^n (\mathbf{F}_i - \mathbf{H}_i) B_i^n, \quad k = n \text{ bzw. } n+1, \\ (\mathbf{S}_2)_w &= k \sum_{i=0}^n (\mathbf{G}_i - \mathbf{H}_i) B_i^n, \quad k = n \text{ bzw. } n+1, \end{aligned}$$

kann  $\beta = 1$  gesetzt werden, d.h.  $((1-u)\alpha_0 + u\alpha_1)(\mathbf{S}_1)_u + (\mathbf{S}_1)_v + \gamma_0(\mathbf{S}_2)_w \equiv 0$ . Durch einfachen Koeffizientenvergleich ergeben sich die *Farin-Bedingungen*:

$$\begin{aligned} \mathbf{F}_i &= \mathbf{H}_i - \gamma_0 (\mathbf{G}_i - \mathbf{H}_i) - \left( \frac{n-i}{n} \alpha_0 (\mathbf{H}_{i+1} - \mathbf{H}_i) + \frac{i}{n} \alpha_0 (\mathbf{H}_i - \mathbf{H}_{i-1}) \right) \\ &= \frac{n-i}{n} (-\alpha_0 \mathbf{H}_{i+1} + (1 - \gamma_0 + \alpha_0) \mathbf{H}_i + \gamma_0 \mathbf{G}_i) \\ &\quad + \frac{i}{n} (\alpha_1 \mathbf{H}_{i-1} + (1 + \gamma_0 - \alpha_1) \mathbf{H}_i - \gamma_0 \mathbf{G}_i), \quad i = 0, \dots, n. \end{aligned} \quad (1.3)$$

Die Farin-Bedingungen werden häufig in der zu (1.3) äquivalenten Form

$$\mathbf{F}_i = \frac{n-i}{n} (\bar{\alpha}_0 \mathbf{H}_{i+1} + \bar{\alpha}_1 \mathbf{H}_i + \bar{\gamma}_0 \mathbf{G}_i) + \frac{i}{n} (\bar{\alpha}_2 \mathbf{H}_i + \bar{\alpha}_3 \mathbf{H}_{i-1} - \bar{\gamma}_0 \mathbf{G}_i), \quad i = 0, \dots, n.$$

mit  $\bar{\alpha}_0 + \bar{\alpha}_1 + \bar{\gamma}_0 = 1$  und  $\bar{\alpha}_2 + \bar{\alpha}_3 + \bar{\gamma}_0 = 1$  geschrieben.

Ist die Fläche  $\mathbf{S}_2$  bereits bekannt und entlang der betrachteten Kante regulär, d.h.  $(\mathbf{S}_2)_w \times (\mathbf{S}_2)_u \neq 0$ , so können mittels der Farin-Bedingungen (1.3) direkt die Kontrollpunkte  $\mathbf{F}_i$  so bestimmt werden, daß  $\mathbf{S}_1$  und  $\mathbf{S}_2$  entlang des gemeinsamen Randes  $G^1$ -stetig aneinander anschließen.

### Chiyokura und Kimura [CK83]

Chiyokura und Kimura gehen davon aus, daß die gemeinsame kubische Randkurve  $\mathbf{S}_1(u, 0) = \mathbf{S}_2(0, u)$  und die Kontrollpunkte  $\mathbf{F}_0, \mathbf{F}_3, \mathbf{G}_0, \mathbf{G}_3$ , gegeben sind.<sup>5</sup> Die Flächen  $\mathbf{S}_i, i = 1, 2$  werden als bikubische TP-Bézierflächen angesetzt. Entlang der gemeinsamen Randkurve wird ein lineares Querableitungsvektorfeld definiert:  $\vec{\mathbf{q}}(u) = (1-u)\hat{\mathbf{q}}_0 + u\hat{\mathbf{q}}_1$ . Die Vektoren  $\hat{\mathbf{q}}_0$  und  $\hat{\mathbf{q}}_1$  werden orthogonal zu  $\mathbf{H}_1 - \mathbf{H}_0$  bzw.  $\mathbf{H}_3 - \mathbf{H}_2$  gewählt, derart, daß  $\langle \hat{\mathbf{q}}_0 | \mathbf{F}_0 - \mathbf{H}_0 \rangle > 0$  und  $\langle \hat{\mathbf{q}}_1 | \mathbf{F}_3 - \mathbf{H}_3 \rangle > 0$ . Die Ableitungen  $(\mathbf{S}_1)_v$  und  $(\mathbf{S}_2)_w$  ergeben sich dann zu:

$$(\mathbf{S}_1)_v = \alpha_1 (\mathbf{S}_1)_u + \beta_1 \vec{\mathbf{q}}, \quad (\mathbf{S}_2)_w = \alpha_2 (\mathbf{S}_1)_u + \beta_2 \vec{\mathbf{q}},$$

wobei  $\alpha_1, \beta_1, \alpha_2, \beta_2$  lineare Gewichtspolynome sind. Die Gewichtspolynome sind durch die in den Ecken bekannten Werte von  $(\mathbf{S}_1)_v$  und  $(\mathbf{S}_2)_w$  eindeutig bestimmt.

Diese Methode kann ohne Schwierigkeit auch auf quartische Dreiecks-Bézierflächen erweitert werden (vgl. die Methode nach Farin).

<sup>5</sup>Hierbei wird natürlich noch vorausgesetzt, daß die Kontrollpunkte  $\mathbf{F}_0, \mathbf{F}_3, \mathbf{G}_0, \mathbf{G}_3$ , so gesetzt sind, daß z.B. die Vektoren  $\mathbf{H}_1 - \mathbf{H}_0$  und  $\mathbf{F}_0 - \mathbf{H}_0$  einen (orientierten) Winkel  $\phi$  in der Tangentialebene mit  $\phi \in ]0, \pi[$  einschließen.

Das Verfahren von Chiyokura und Kimura generiert einen tangentialstetigen Übergang zwischen den beiden Flächen  $\mathbf{S}_1$  und  $\mathbf{S}_2$ . Gilt  $(\mathbf{S}_1)_u \times \vec{\mathbf{q}} \neq 0$  entlang der Kante, so ist stets  $\beta_1(u) > 0$  und  $\beta_2(u) < 0$ , womit der Übergang in dieser Situation  $G^1$ -stetig ist.

### DeRose [DeR90]

DeRose gibt eine eindeutige Charakterisierung für die Tangentialstetigkeit zwischen den Flächen  $\mathbf{S}_1$  und  $\mathbf{S}_2$  beliebigen Grades. Diese Charakterisierung beruht auf der zur Gleichung (1.1) äquivalenten Definition der Tangentialstetigkeit:

$$\det((\mathbf{S}_1)_u, (\mathbf{S}_1)_v, (\mathbf{S}_2)_w)(u) \equiv 0.$$

Werden die Ableitungen durch die Bernstein-Bézier Darstellungen mit den entsprechenden Kontrollvektoren  $\vec{\mathbf{u}}_i, \vec{\mathbf{v}}_j, \vec{\mathbf{w}}_k$  ersetzt, so ergibt sich durch einfache Umformulierung unter Ausnutzung der Multilinearität der Determinante:

$$\sum_{m=0}^{n_c+n_1+n_2} \underbrace{\left( \sum_{i+j+k=m} \frac{\binom{n_c}{i} \binom{n_1}{j} \binom{n_2}{k}}{\binom{n_c+n_1+n_2}{m}} |\vec{\mathbf{u}}_i, \vec{\mathbf{v}}_j, \vec{\mathbf{w}}_k| \right)}_{K_m} B_m^{n_c+n_1+n_2}(u) \equiv 0. \quad (1.4)$$

DeRose zeigt, daß das Verschwinden von  $K_m, m = 0, \dots, n_c + n_1 + n_2$  notwendig und hinreichend für Tangentialstetigkeit ist.

### Eckenumschlußproblem

Das Eckenumschlußproblem tritt dann auf, wenn man mehrere Bézierflächen um einen Punkt gruppieren möchte, so daß alle Übergänge glatt, d.h. tangential- oder  $G^1$ -stetig werden. Hierbei wird vorausgesetzt, daß die Flächen schon  $G^0$ -stetig aneinander anschließen, d.h., daß benachbarte Flächen eine gemeinsame Randkurve besitzen (vgl. Abbildung 1.3). Die Randkurven müssen im Punkte  $\mathbf{V}_0$   $G^1$  kompatibel sein.

Betrachtet man zunächst das Problem der Konstruktion von  $C^1$  Übergängen, so ergibt sich für jede Kante eine Bedingung der Form

$$\mathbf{b}_{i+1,11} = \alpha_i \mathbf{b}_{i,1} + \beta_1 \mathbf{b}_{i,2} + \gamma_i \mathbf{b}_{i-1,11}. \quad (1.5)$$

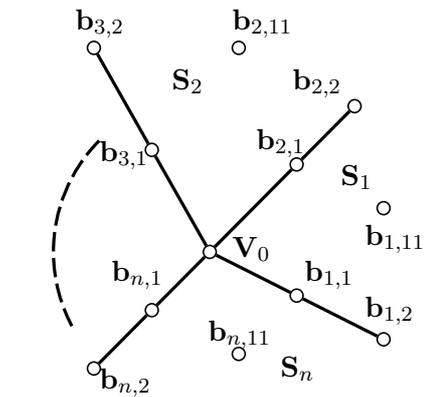


Abbildung 1.3: Das Eckenumschlußproblem

Daraus ergibt sich folgendes Gleichungssystem für die bislang unbestimmten Kontrollpunkte  $\mathbf{b}_{i,11}$ :

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & \gamma_1 \\ \gamma_2 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \gamma_n & 1 \end{pmatrix} \begin{pmatrix} \mathbf{b}_{1,11} \\ \vdots \\ \mathbf{b}_{n,11} \end{pmatrix} = \text{RHS.}$$

Die Determinante der Matrix ergibt sich unmittelbar zu  $1 - (-1)^n \prod_{i=1}^n \gamma_i$ . Aufgrund der Eigenschaft

$$\mathbf{b}_{i+1,1} = \alpha_i \mathbf{V}_0 + \beta_1 \mathbf{b}_{i,1} + \gamma_i \mathbf{b}_{i-1,1}, \quad i = 1, \dots, n,$$

gilt:  $\prod_{i=1}^n \gamma_i = 1$ . Daher ist die Matrix für ungerades  $n$  regulär, jedoch für gerades  $n$  singulär und damit das zugehörige Gleichungssystem in diesem Fall i.a. nicht lösbar.

Die Analyse des allgemeinen Falles für  $G^1$ -Übergänge ergibt, daß keine weiteren Freiheitsgrade hinzukommen, die eine Lösung für gerades  $n$  garantieren (vgl. [Pet91]).

## 1.3 Variationsprinzipien

Eine der wichtigsten Anforderungen des Kurven- und Flächendesigns ist die Berücksichtigung der Zielvorstellungen des Benutzers bzw. des Designers. Der Designer hat meist klare Vorstellungen von dem *was* er modellieren will. Das vollständige Übertragen dieser Vorstellungen auf ein CAD-System alleine mit den üblichen Kurven- und Flächen-Kontrollstrukturen (wie z.B. Kontrollnetze, Trimmkurven etc.) ist allerdings kaum möglich. Vielmehr muß das System eine Möglichkeit bieten, wie unter Angabe *einiger* wesentlicher Kenngrößen (wie z.B. die Lage weniger Kontrollpunkte) eine globale Lösung berechnet werden kann. Hierzu ist es nötig die übrigen, bislang noch nicht festgelegten Größen mit Hilfe vorgegebener Kriterien automatisch zu bestimmen. Der *Variationsansatz* hat sich in diesem Zusammenhang als sehr erfolgreich erwiesen. Dabei wird die *Glätte* einer Kurve oder Fläche mittels eines Funktionals beschrieben, welches bestimmte geometrische Eigenschaften bewertet. Die unbekanntenen Größen werden nun so bestimmt, daß unter allen zulässigen Kurven bzw. Flächen diejenige ausgewählt wird, die bzgl. des angesetzt Funktional als am besten bewertet wird. Damit ist diejenige Kurve oder Fläche gefunden, welche die vorgegebenen Bedingungen erfüllt und welche der durch das Funktional gemessenen geometrische Eigenschaft an besten entspricht.

Im folgende werden die verschiedenen Variationsansätze kurz skizziert. Da es hier mehr um eine prinzipielle Darstellung geht, werden die grundsätzlichen Arbeitsweisen anhand folgenden Interpolationsproblems erläutert:

**Gegeben:** Ein Knotenvektor  $T = \{u_1, \dots, u_N\}$ ,  $u_i < u_{i+1}$ ,  $i = 1, \dots, n-1$ ,  $u_0 = 0$ ,  $u_N = 1$ , und reelle Datenwerte  $f_1, \dots, f_N$ .

**Gesucht:** Eine „möglichst glatte“ Kurve  $C : [0, 1] \rightarrow \mathbb{R}$ , welche die Datenwerte  $f_i$  interpoliert, d.h.  $C(u_i) = f_i$ ,  $i = 1, \dots, N$ .

Sind das Funktional  $\sigma$  und der lineare Funktionenraum  $\mathcal{V}$ , aus dem die Kurve  $C$  gewählt werden kann, bekannt, so stellt sich das dem Interpolationsproblem zugeordnete *Variationsproblem*<sup>6</sup>:

<sup>6</sup>Im folgenden wird auch der Begriff *Optimierung* bzw. *Optimierungsproblem* verwendet. Hier sollen beide Begriff als synonym angesehen werden.

**Problem:** Bestimme unter allen  $C \in \mathcal{V}_N = \{C \in \mathcal{V} : C(u_i) = f_i\}$ , diejenige Kurve  $C_0$  mit minimalem Wert  $\sigma(C_0)$ .

Je nach Art des Funktionals  $\sigma$  gibt es verschiedene Ansätze, um das obige Variationsproblem zu lösen.

Ist  $\sigma$  *quadratisch*, d.h. es existiert eine symmetrische Bilinearform  $\langle \cdot | \cdot \rangle_\sigma$  mit  $\sigma(C) = \langle C | C \rangle_\sigma$ , für alle  $C \in \mathcal{V}$ , dann können Methoden, wie sie aus der *Finiten Elemente* Rechnung bekannt sind, eingesetzt werden (vgl. [Bra92]). Der Raum  $\mathcal{V}$  wird dabei durch Hinzunahme von Randbedingungen eingeschränkt, etwa durch:  $C'(u_1) = f'_1$  und  $C'(u_N) = f'_N$ ,  $\forall C \in \mathcal{V}$ . Eine wichtige Voraussetzung für die folgende Vorgehensweise ist, daß alle Nebenbedingungen, wie z.B. Interpolationsbedingungen, linear sind. Damit ist der Raum  $\mathcal{V}_N$  ein affiner Teilraum von  $\mathcal{V}$ .

Für das oben angeführten Beispiel wäre  $\sigma(C) = \int_0^1 (C''(u))^2 du$ , mit der zugeordneten Bilinearform  $\langle C_1 | C_2 \rangle_\sigma = \int_0^1 C_1''(u)C_2''(u) du$ , ein solches Funktional. Ein geeigneter Funktionenraum ist  $\mathcal{V} = \{C : C''' \in \mathcal{L}_2[0, 1]\}$ .<sup>7</sup> Kennzeichnet  $\mathcal{V}_0$  den linearen Teilraum von  $\mathcal{V}$ , welcher die homogenen Bedingungen  $\mathcal{V}_0 = \{C \in \mathcal{V} : C(u_i) = 0\}$  erfüllt und ist  $\langle \cdot | \cdot \rangle_\sigma$  auf  $\mathcal{V}_0$  positiv definit, so läßt sich  $C_0$  eindeutig durch die Beziehung<sup>8</sup>

$$\langle C_0 | C \rangle = 0, \quad \text{für alle } C \in \mathcal{V}_0, \quad (1.6)$$

charakterisieren.

Kann aus Gleichung (1.6) mittels partieller Integration ein äquivalenter Ausdruck in Form einer Differentialgleichung, der sogenannten *Eulergleichung*, gewonnen werden, so kommen als Lösungen  $C_0$  nur Lösungen dieser Differentialgleichung in Frage. Für das Beispiel erhält man die Eulergleichung  $C^{(4)} \Big|_{[u_i, u_{i+1}]} = 0, i = 1, \dots, N-1$ , womit  $C_0$  ein stückweises Polynom vom Grad 3 sein muß.<sup>9</sup>

Läßt sich aus der Gleichung (1.6) keine Eulergleichung ableiten oder besitzt die Eulergleichung keine geschlossene Lösung, so wird häufig folgende Einschränkung gemacht: Ist  $\mathcal{V}$  nicht endlich dimensional, so wird ein geeigneter endlich dimensionaler Teilraum  $\mathcal{U} = \text{span}\{B_0, \dots, B_n\} \subseteq \mathcal{V}$  ausgewählt und nur in diesem nach einer Lösung gesucht. Der affine Lösungsraum ergibt sich zu  $\mathcal{U}_N = \mathcal{V}_N \cap \mathcal{U}$ . Mit Hilfe einer Fundamentallösung  $\bar{C} \in \mathcal{U}_N$  kann man  $\mathcal{U}_N$  darstellen durch:  $\mathcal{U}_N = \bar{C} + \mathcal{U}_0$ .  $\mathcal{U}_0 = \mathcal{V}_0 \cap \mathcal{U}$  ist der Lösungsraum des homogenen Problems.  $C_0$  ergibt sich dann als eindeutige Lösung von

$$\langle C_0 | C \rangle = 0, \quad \text{für alle } C \in \mathcal{U}_0. \quad (1.7)$$

Aus der Gleichung (1.7) läßt sich ein Gleichungssystem in den Koeffizienten von  $C_0 = \bar{C} + \sum_{i=0}^n c_i B_i$  ableiten. Diese Methode der Einschränkung wird *Ritz-* oder *Galerkin-Verfahren* genannt.

<sup>7</sup> $\mathcal{L}_2[a, b]$  bezeichnet den Raum der quadratisch Lebesque-integrierbaren Funktionen über dem Intervall  $[a, b]$ .

<sup>8</sup>Streng genommen muß hier noch vorausgesetzt werden, daß  $\mathcal{V}_0 \cap \{C \in \mathcal{V} : \sigma(C) = 0\}$  nur die triviale Nullfunktion enthält. In dem angegebenen Beispiel ist dies der Fall.

<sup>9</sup>Die B-Splines über dem Knotenvektor  $T$  können durch diese Optimaleigenschaft charakterisiert werden.

In dem eingeführten Beispiel würde das Funktional  $\sigma(C) = \int_0^1 (C'''(u))^2 + (C''(u))^2 + (C'(u))^2 du$ , mit einem Ritz-Galerkin-Ansatz behandelt werden, da die Auflösung der zu  $\sigma$  gehörige Eulergleichung  $C^{(6)} + C^{(4)} + C'' = 0$  eine Fallunterscheidung nötig macht (vgl. [Pot90]).

Ist, im Gegensatz zu der bisherigen Annahme,  $\sigma$  kein quadratisches Funktional, so ist die Lösung des Optimierungsproblems i.a. nicht mehr eindeutig. Falls der Funktionenraum  $\mathcal{V}$  unendlich dimensional ist, ist das Auffinden einer Lösung  $C_0$  kaum mehr möglich. Deshalb wird hier von vorneherein ein Ritz-ähnlicher Ansatz gemacht, d.h. es wird auf einem endlich dimensionalen Teilraum  $\mathcal{U} \subseteq \mathcal{V}$  optimiert. Numerische Verfahren, wie z.B. das Gradientenverfahren oder das Newton-Verfahren können angesetzt werden um ein (lokales) Minimum von  $\sigma$  zu bestimmen (vgl. [Sto93]). Bei dem hier verfolgten Beispiel kann das nicht quadratische Funktional  $\sigma = \int_0^1 \kappa_C^2 ds = \int_0^1 (C''(u))^2 / (1 + C'(u))^{\frac{5}{2}} du$ , auf diese Weise behandelt werden.  $\sigma$  beschreibt die Biegeenergie eines dünnen elastischen Stabes, der an den Stellen  $(u_i, f_i)$  eingespannt ist.

Eine Variante bei den quadratischen Funktionalen stellt der Ansatz von Greiner [Gre94a] dar. Greiner verwendet *datenabhängige* quadratische Funktionale, die von der Geometrie des betrachteten Kurven- oder Flächenvariationsproblems abhängen. Dieser Ansatz beruht für Kurven auf der Erweiterung der Begriffe *Gradient* und *Divergenz* vom funktionalen Problem auf Funktionen auf Kurven. Der Gradient einer Funktion  $f : [0, 1] \rightarrow \mathbb{R}$  bzw. die Divergenz eines Tangentialfeldes<sup>10</sup>  $\vec{t} : [0, 1] \rightarrow \mathbf{TC}$  auf der Referenzkurve  $C^0 : [0, 1] \rightarrow \mathbb{R}$  wie wird folgt definiert:

$$\text{grad}_{C^0}(f) = \frac{f'}{1 + (C^{0'})^2}, \quad \text{div}_{C^0}(\vec{t}) = \frac{(\vec{t} \sqrt{1 + (C^{0'})^2})'}{\sqrt{1 + (C^{0'})^2}}. \quad (1.8)$$

Der *Laplace-Beltrami-Operator* (vgl. [Kli78]) für die Kurve  $C$  bzgl. der Referenzkurve  $C^0$  ergibt sich zu

$$\Delta_{C^0}(C) = \text{div}_{C^0}(\text{grad}_{C^0}(C)) = \frac{C''}{1 + (C^{0'})^2} - \frac{(C^0)'(C^0)''C'}{(1 + (C^{0'})^2)^2},$$

woraus sich für die Kurve  $C$  das quadratische Funktional  $\sigma(C) = \int_0^1 \Delta_{C^0}(C)^2 ds$  ableitet. Hierbei wird nach der Bogenlänge von  $C^0$  integriert. Der Vorteil dieses geometrieabhängigen Funktionals liegt darin, daß es für  $C \approx C^0$  eine gute Approximation an das nicht quadratische Funktional  $\int_0^1 (C''(u))^2 / (1 + C'(u))^{\frac{5}{2}} du = \int_0^1 \kappa_C^2 ds$  darstellt.

Für die neuen Algorithmen, die in dieser Arbeit vorgestellt werden, werden ausschließlich quadratische Funktionale zur Optimierung verwendet.

<sup>10</sup> $\mathbf{TC}$  bezeichnet den Tangentialraum der Kurve  $\mathbf{C}$ .



# Kapitel 2

## Übersicht bekannter Interpolationsverfahren

In diesem Abschnitt werden eine Reihe von Verfahren zur Lösung sowohl des funktionalen wie des parametrischen Interpolationsproblems beschrieben (vgl. [LMD92]). Traditionell unterteilt man die verschiedenen Verfahren in *lokale* und *globale* Methoden. Dabei wird eine Methode als lokal bezeichnet, falls sich die Änderung eines Interpolationspunktes auf die Interpolationsfläche nur lokal auswirkt. Ist dies nicht der Fall, so spricht man von einem globalen Verfahren. Globale Verfahren können meist durch geeignete Modifikation „lokalisiert“ werden.

Wichtiger ist hier jedoch die Unterteilung in die Kategorien der *nicht optimierenden* und der *optimierenden* Verfahren. Charakteristisch für ein nicht optimierendes Verfahren ist die Verwendung heuristischer Regeln zur Festlegung aller freien Parameter, die nicht durch die Interpolations- oder Glattheitsforderungen bestimmt sind. Diese Verfahren sind i.a. einfach zu implementieren, die entsprechenden Algorithmen sind zudem relativ schnell. Mann [Man92] weist jedoch für den parametrischen Fall darauf hin, daß diese Verfahren in vielen Situationen Flächen minderer Qualität erzeugen.

Im Gegensatz dazu verwenden die optimierenden Verfahren Variationsansätze um alle oder einen Teil dieser freien Parameter festzulegen. Dadurch sind die Algorithmen i.a. langsamer, ebenso nimmt der Implementationsaufwand für solche Verfahren zu. Die erzielte Qualität der Interpolationsfläche rechtfertigt jedoch in vielen Fällen den Mehraufwand.

In den folgenden Abschnitten werden verschiedene Methoden zur Lösung des funktionalen und des parametrischen Problems diskutiert. Hierbei wird zwischen den klassischen, nicht optimierenden und den optimierenden Verfahren unterschieden.

### 2.1 Methoden zur Lösung des funktionalen Interpolationsproblems

Das funktionale Interpolationsproblem wird seit vielen Jahren intensiv erforscht:

**Gegeben:** Datenpunkte  $\mathbf{u}_1, \dots, \mathbf{u}_N$  in der Ebene und zugehörige reelle Datenwerte  $f_1, \dots, f_N$ .

**Gesucht:** Eine bivariate Funktion bzw. funktionale Fläche  $S : \Omega \rightarrow \mathbb{R}$  mit:  $\mathbf{u}_i \in \Omega \subseteq \mathbb{R}^2$  und  $S(\mathbf{u}_i) = f_i$ ,  $i = 1, \dots, N$ .

In [Alf89, FN91, HL92] werden die meisten der bekannten Verfahren im Überblick vorgestellt.

Zu den in diesem Abschnitt beschriebenen Verfahren gehören eine Reihe klassischer, nicht optimierender Methoden, sowie die *Minimum Norm Netzwerk (MNN)*-Methode und das Differenzenverfahren von Arge, Dæhlen und Tveito. Die letzteren Verfahren zählen zu den optimierenden Methoden.

### 2.1.1 Klassische Methoden

Die wohl wichtigsten Vertreter der klassischen Verfahren zur Lösung des funktionalen Interpolationsproblems sind die *Shepard-Methoden* und die *radialen Basisfunktions-Methoden*. Neben diesen Methoden werden zwei weitere Ansätze diskutiert, welche beide Dreiecks-Bézierflächen zur Konstruktion der Interpolationsfläche verwenden: das *Clough-Tocher*- und das *Powell-Sabin* Splitverfahren.

#### Shepard:

Die zu den nicht optimierenden Verfahren gehörende Methode nach Shepard [She68] basiert auf dem folgenden Ansatz für die Interpolationsfläche:

$$S(\mathbf{u}) = \sum_{i=1}^N f_i \omega_i(\mathbf{u}) \quad \text{mit} \quad \omega_i(\mathbf{u}_j) = \begin{cases} 1 & \text{für } i = j, \\ 0 & \text{sonst.} \end{cases}$$

Der Wahl der Gewichtsfunktionen  $\omega_i$  kommt dabei eine große Bedeutung zu. Sie beeinflussen das lokale Verhalten und das globale Aussehen der resultierenden Interpolationsfunktion. In [Alf89] und [HL92] werden verschiedene Möglichkeiten angegeben, wie die  $\omega_i$  gewählt werden können. Die Erzeugung von Interpolationsflächen mit „weichem“ Verlauf, d.h. mit z.B. möglichst gleichmäßiger Krümmungsverteilung, erweist sich jedoch als sehr schwierig.

#### Radiale Basisfunktionen:

Die Verfahren zur Lösung des funktionalen Interpolationsproblems mittels radialer Basisfunktionen  $B : [0, \infty) \rightarrow \mathbb{R}$ , beruht auf Funktionsansatz:

$$S(\mathbf{u}) = \sum_{i=1}^N b_i B(d(\mathbf{u}_i, \mathbf{u})) + p_m(\mathbf{u}), \quad \text{mit } p_m \in \Pi_m.$$

$m$  wird als *polynomielle Präzision* bezeichnet,  $d(\mathbf{u}_i, \mathbf{u})$  ist der euklidische Abstand vom Parameterwert  $\mathbf{u}$  zum Datenpunkt  $\mathbf{u}_i$ .

Man erhält ein lineares Gleichungssystem, welches das Interpolationsproblem und die zusätzlich geforderten *Gleichgewichtsbedingungen*  $\sum_{i=1}^N b_i q(\mathbf{u}_i) = 0, \forall q \in \Pi_m$  beschreibt. Die Lösbarkeit dieses Gleichungssystems ist i.a. nicht garantiert und muß für jede spezielle Basisfunktion  $B$  gesondert betrachtet werden.

Interessant in unserem Zusammenhang sind die zu den optimierenden Verfahren zu zählenden *Thin Plate Splines*. Diese erhält man unter Verwendung der Basisfunktion  $B(u) = u^2 \log u$  für  $m = 1$ . Die Thin Plate Splines haben die Eigenschaft, daß sie unter allen bivariaten Funktionen  $S \in \mathcal{L}_2(\mathbb{R}^2)$ , welche die Interpolationsbedingungen  $S(\mathbf{u}_i) = f_i, i = 1, \dots, N$ , erfüllen, das Thin Plate-Funktional

$$\sigma^{TP}(S) = \int_{\mathbb{R}^2} (S_{uu})^2 + 2(S_{uv})^2 + (S_{vv})^2 du dv, \quad (2.1)$$

minimieren (vgl. [Duc77]). Diese Minimaleigenschaft der Thin Plate Splines ist analog zu der Minimaleigenschaft der univariaten B-Splines zu sehen (vgl. Abschnitt 1.3). Daher werden diese Flächen gelegentlich auch als *Surface Splines* bezeichnet.

### Methoden basierend auf Dreiecks-Bézierflächen:

Farin [Far86] gibt einen Überblick über die verschiedenen Verfahren zur Lösung des funktionalen Interpolationsproblems unter Verwendung von Dreiecks-Bézierflächen. Dazu wird das Parametergebiet zunächst trianguliert, wobei meist die *Delaunay*-Triangulierung gewählt wird (vgl. [Sch87, Sch93]). Die Datenpunkte  $\mathbf{u}_i$  dienen dabei als Eckpunkte für die Triangulierung.

Alle Verfahren, die wenigstens eine  $C^1$ -stetige Interpolationsfläche erzeugen, benötigen außer den gegebenen Datenwerten  $f_i$  weitere Informationen wie z.B. Ableitungsdaten in den Datenpunkten  $\mathbf{u}_i$ .

Natürlicherweise möchte man möglichst wenig zusätzliche Information zur Lösung des Interpolationsproblems benötigen. Hier eignen sich die sogenannten *Split*-Ansätze.

Hierbei werden mehrere Dreiecks-Bézierfläche über einem Dreieck der Triangulierung definiert, die dann eine globale  $C^1$ -stetige Fläche ergeben. Die wichtigsten Vertreter dieser Verfahren sind der aus der Finiten Element Literatur bekannte Clough-Tocher Ansatz (vgl. [SF73]) und die Methode nach Powell und Sabin [PS77].

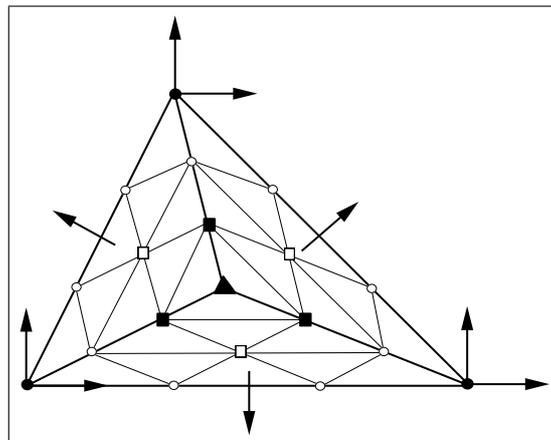


Abbildung 2.1: Die Clough-Tocher Splitfläche. Die Kontrollparameter sind durch gefüllte Kreise • (Kontrollpunkte) und durch Pfeile (partielle Ableitungen) gekennzeichnet. Die Punkte o und □ ergeben sich direkt aus den Kontrollparametern, alle übrigen Bézier-Kontrollpunkte ergeben sich aufgrund der  $C^1$  Übergangsbedingungen entlang der inneren Kanten.

Das Clough-Tocher Verfahren unterteilt jedes Dreieck der Triangulierung, auch *Makrodreieck* genannt, an einem Punkte im Inneren des Dreiecks in drei Mikrodreiecke (vgl. Abbildung 2.1). Üblicherweise wird die Unterteilung am Schwerpunkt durchgeführt. Über jedem Mikrodreieck wird eine kubische Dreiecks-Bézierfläche definiert. Zur eindeutigen Festlegung der Clough-Tocher Splitfläche werden neben den Datenwerten  $f_i$  die partiellen Ableitungen in den Ecken und eine Querableitung in der Mitte jeder Kante des Makrodreiecks benötigt. Diese müssen geeignet abgeschätzt werden. Dies geschieht beispielsweise durch Mittelung der Normalen der Dreiecke, die sich bei stückweise linearer  $C^0$ -Interpolation ergeben oder durch Verwendung eines lokalen Interpolanten, der nur eine kleine Zahl naheliegender Datenpunkte interpoliert.

Powell und Sabin [PS77] stellten eine ganze Reihe von verschiedenen Methoden vor, mit denen stückweise Polynome über Dreiecken definiert werden können. Für das funktionale Interpolationsproblem eignen sich die Methoden zur Unterteilung eines Makrodreiecks in 6 oder 12 Mikrodreiecke. In beiden Fällen wird über jedem Mikrodreieck eine quadratische Dreiecks-Bézierfläche definiert. Als zusätzliche Informationen werden alleine die partiellen Ableitungen in den Datenpunkten  $\mathbf{u}_i$  benötigt.

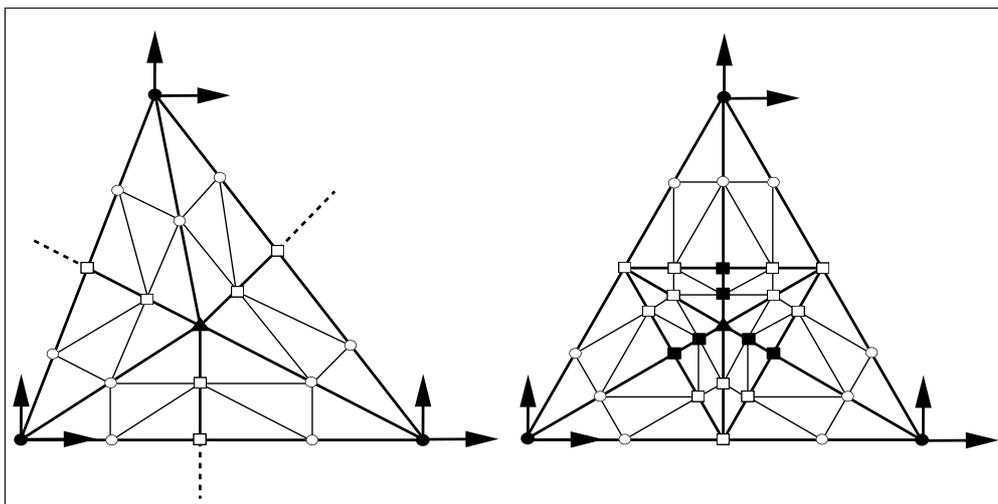


Abbildung 2.2: Die Powell-Sabin Splitfläche, links in 6, rechts in 12 Mikrodreiecke unterteilt. Die Kontrollparameter sind durch gefüllte Kreise • (Kontrollpunkte) und durch Pfeile (partielle Ableitungen) gekennzeichnet. Die Kontrollpunkte ◦ ergeben sich aus diesen Daten in den Ecken; die Kontrollpunkte ◻ werden durch die  $C^1$ -Bedingungen und im Falle der 12-Splitfläche durch Vorgabe der linearen Querableitung entlang der äußeren Kante fixiert. Alle übrigen Kontrollpunkte ergeben sich aufgrund der  $C^1$  Übergangsbedingungen entlang der inneren Kanten.

Zur Definition der 6-Split-Flächen wird jedes Makrodreieck der Triangulierung an einem Punkt im Inneren und an einem Punkte auf jeder Kante des Dreiecks unterteilt (siehe Abbildung 2.2).

Die 12-Splitfläche ergibt sich durch Unterteilung an einem Punkt im Inneren (i.a. der Schwerpunkt) des Dreiecks und an einem Punkt (i.a. der Mittelpunkt) auf jeder Kante. Hier wird zusätzlich gefordert, daß die Querableitung senkrecht zum Rand linear verläuft.

Sollen zwei benachbarte 6-Splitflächen  $C^1$ -stetig aneinander anschließen, so muß der Unterteilungspunkt auf der gemeinsamen Kante auf der Verbindungslinie zwischen den jeweiligen inneren Unterteilungspunkten der Makrodreiecke liegen. Diese geometrische Bedingung ist i.a. nicht leicht zu erfüllen. Powell und Sabin schlagen daher eine Mischung von 6- und 12-Splitflächen vor, wobei die 6-Splitflächen stets am Umkreismittelpunkt und auf den Kantenmitten unterteilt werden. Dadurch verlaufen die entsprechenden inneren Kanten immer senkrecht zu den äußeren Kanten. Wird der größte Innenwinkel größer als 75 Grad, so wird eine 12-Splitfläche anstelle der 6-Splitfläche verwendet.<sup>1</sup>

Die entstehende Interpolationsfläche ist global  $C^1$ -stetig. Die fehlenden Ableitungsdaten werden wie beim Clough-Tocher Split abgeschätzt.

### 2.1.2 Minimum Norm Netzwerke

Der *Minimum Norm Netzwerk (MNN)*-Ansatz geht auf Nielson [Nie83] zurück. Innerhalb der letzten Jahre wurden verschiedene Erweiterungen des ursprünglichen Ansatzes erarbeitet (vgl. [NF84, Pot91, KS95]). Die grundsätzliche Idee des Verfahrens ist die Aufteilung des funktionalen Interpolationsproblems in die folgenden drei Schritte (vgl. Abbildung 2.3):

1. **Schritt:** Das Parametergebiet  $\Omega$  wird trianguliert, wobei die Datenpunkte als Eckpunkte für die Triangulierung dienen.
2. **Schritt:** Auf den Kanten der unter Schritt 1 bestimmten Triangulierung wird mittels eines Variationsansatzes ein Kurvennetz (Netzwerk) generiert, welches die Datenwerte  $f_i$  interpoliert.
3. **Schritt:** Unter Verwendung transfiniter Interpolanten (vgl. Abschnitt 1.2 auf Seite 8) wird das Kurvennetz zu einer glatten Fläche fortgesetzt.

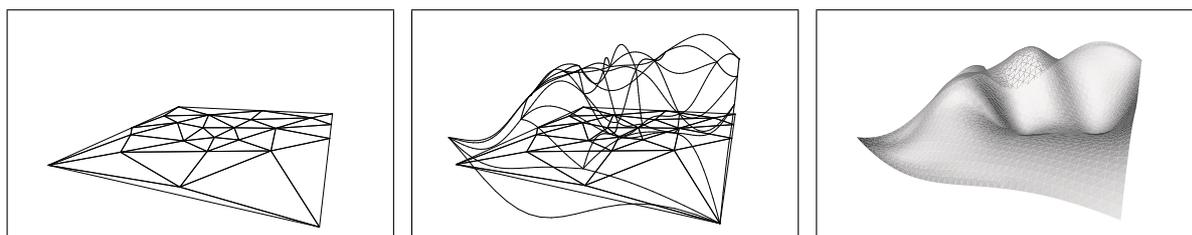


Abbildung 2.3: Die drei Schritte des MNN-Verfahrens: Links eine triangulierte Datenpunktmenge; Mitte das mit dem Funktional von Kolb und Seidel optimierte Kurvennetz (vgl. Tabelle 2.1); Rechts der resultierende Interpolant.

In der klassischen Form der MNN-Methode in [Nie83] beruht die Variation im 2. Schritt i.w. auf die Charakterisierung der univariaten B-Splines, d.h. es wird das Funktional

<sup>1</sup>Die 75 Grad Grenze ist rein heuristisch; wichtig ist allerdings daß der Umkreismittelpunkt innerhalb des Dreiecks liegt.

$$\sigma(N) = \sum_{ij \in E} \int_{\mathbf{e}_{ij}} \left( \frac{\partial^2 N}{\partial \vec{\mathbf{e}}_{ij}^2} \right)^2 d\mathbf{e}_{ij} \quad (2.2)$$

angesetzt. Hierbei bezeichnet  $\mathbf{e}_{ij}$  und  $\vec{\mathbf{e}}_{ij}$  die Kante bzw. den Kantenvektor zwischen den Datenpunkten  $\mathbf{u}_i$  und  $\mathbf{u}_j$ .  $E$  entspricht die Menge der Doppelindizes, welche die Kanten der Triangulierung repräsentieren. Zur Menge aller zugelassenen Kurvennetze für die Variation gehören diejenigen über den Kanten  $\mathbf{e}_{ij}$ ,  $ij \in E$  definierten Funktionen  $N$  (Kurvennetze) mit

- $(N'')|_{\mathbf{e}_{ij}} \in \mathcal{L}_2(\mathbf{e}_{ij})$ ,  $ij \in E$  und
- $N$  ist in allen Datenpunkten  $\mathbf{u}_i$   $C^1$ -kompatibel (vgl. Abschnitt 1.2.1 auf Seite 7).

Die eindeutige Lösung zu obigem Variationsproblem ergibt sich als stückweise kubisches Polynom-Kurvennetz. Die partiellen Ableitungsdaten  $S_u(\mathbf{u}_i)$ ,  $S_v(\mathbf{u}_i)$ , welche die Tangentialebene  $\mathbf{T}_{(\mathbf{u}_i, f_i)}(S)$  festlegen, ergeben sich als Lösung eines regulären linearen  $2N \times 2N$ -Gleichungssystems.

Zur Interpolation des Kurvennetzes mittels transfiniten Interpolanten werden für die einzelnen Blendoperatoren kubische Polynome als erzeugende Kurven angesetzt.

Verfahren	Funktional für Kurve $C$ auf Kante $\mathbf{e}$	Kurventyp	Stetigk.
Nielson/Franke [NF84]	$\int_{\mathbf{e}} \left\{ \left( \frac{\partial^2 C}{\partial \vec{\mathbf{e}}^2} \right)^2 + \alpha^2 \left( \frac{\partial C}{\partial \vec{\mathbf{e}}} \right)^2 \right\} d\mathbf{e}$	Spline in Tension	$C^1$
Pottmann [Pot91]	$\int_{\mathbf{e}} \left\{ \sum_{k=0}^r \alpha_k \left( \frac{\partial^{r+1} C}{\partial \vec{\mathbf{n}}^k \partial \vec{\mathbf{e}}^{r-k+1}} \right)^2 \right\} d\mathbf{e}$	Polynom vom Grad $2r + 1$	$C^r$
Kolb/Seidel [KS95]	$\int_{\mathbf{e}} \left\{ \left( \frac{\partial^3 C}{\partial \vec{\mathbf{e}}^3} \right)^2 + \alpha^2 \left( \frac{\partial^2 C}{\partial \vec{\mathbf{e}}^2} \right)^2 \right\} d\mathbf{e}$	Verall. Spline in Tension	$C^2$

Tabelle 2.1: Die verschiedenen Erweiterungen des ursprünglichen MNN-Ansatzes von Nielson. Das Kurvennetz-Funktional ergibt aus der Aufsummierung aller Kurvenfunktionale (vgl. zweite Spalte). „Kurventyp“ bezeichnet die Klasse der Kurve aus dem das optimale Kurvennetz besteht.  $\vec{\mathbf{n}}$  bezeichnet die Richtung in der Parameterebene senkrecht zu  $\vec{\mathbf{e}}$ .

Die verschiedenen Erweiterungen der MNN-Methode zielen in zwei Richtungen: Einfügen von Shape-Parametern wie z.B. Tension-Parametern und die Erzeugung höherstetiger Kurvennetze und Flächen. Eine Übersicht über die unterschiedlichen Methoden gibt Tabelle

2.1. Die in der Spalte „Kurventyp“ aufgeführten Splines in Tension und der Verallgemeinerten Splines in Tension sind detailliert in [Sch68] bzw. [Pot90] beschrieben.

Gegenüber den klassischen Methoden, wie z.B. den radialen Basisfunktions-Methoden, bietet der MNN-Ansatz den Vorteile, daß er sehr gut auf nicht konvexen Parametergebieten eingesetzt werden kann. Zudem ist man in der Wahl des Optimierungsfunktionalen viel freier als bei den Thin Plate Splines.

Die Nachteile des MNN-Verfahrens liegen hauptsächlich in der numerisch schwierigen Behandlung von Flächen über dünnen und langen Dreiecken der Triangulierung. Des weiteren ist die Verwendung der transfiniten Interpolationsflächen dann problematisch, wenn Kompatibilität mit CAD-Systemen angestrebt wird.

### 2.1.3 Das Verfahren von Arge, Dæhlen und Tveito

Ein weiteres Verfahren, welches explizit einen Variationsansatz zur Lösung der funktionalen Interpolationsproblems verwendet, wurde von Arge, Dæhlen und Tveito [ADT94] vorgestellt. Hierbei muß allerdings auf folgende Besonderheit aufmerksam gemacht werden. Da das Verfahren von Arge, Dæhlen und Tveito auf einem Differenzenansatz unter Verwendung von *Gitterfunktionen* beruht, können nur Datenwerte exakt interpoliert werden, deren zugehöriger Datenpunkt auf einen Gitterpunkt fällt (vgl. Abbildung 2.4). Alle anderen Datenwerte werden nur approximiert.

Die Methode läßt sich in die folgenden zwei Schritte unterteilen:

- 1. Schritt:** Es werden diejenigen Gitterpunkte des regulären Gitters und deren Ordinatenwerte bestimmt, die nahe bei den in allgemeiner Lage befindlichen Datenpunkten  $\mathbf{u}_i$  liegen.
- 2. Schritt:** Mit Hilfe eines Glättungsansatzes<sup>2</sup> auf der diskreten Menge der Gitterpunkte werden die restlichen Ordinatenwerte, die nicht im ersten Schritt bestimmt wurden, festgelegt.

Das reguläre Gitter, welches über das Parametergebiet gelegt wird, muß geeignet gewählt werden. Dabei spielen i.w. zwei Anforderungen eine Rolle. Einerseits sollen möglichst wenig Gitterlinien verwendet werden, andererseits darf der Abstand zwischen einem Datenpunkt  $\mathbf{u}_i$  und dem nächstliegenden Gitterpunkt nicht zu groß sein.

Die Bestimmung der nahe an dem Datenpunkt  $\mathbf{u}_i$  liegenden Gitterpunkten erfolgt normalerweise mittels der  $\|\cdot\|_\infty$ -Norm. Für diese Gitterpunkte wird der Ordinatenwert unter Ansatz einer lokalen Interpolationsfunktion, z.B. eines Thin-Plate-Splines (vgl. Abschnitt 2.1.1 auf Seite 18) auf die nächstliegenden Datenpunkte bestimmt (vgl. Abbildung 2.4). Alternativ werden die Datenwerte einfach übernommen (*shift*), d.h. der lokale Interpolant wird als konstant angesetzt.

---

<sup>2</sup>Der hier verwendete Glättungsansatz unterscheidet sich von einem Variationsansatz insofern, als daß die geometrische Eigenschaft durch eine Differentialgleichung und nicht durch ein Funktional dargestellt wird.

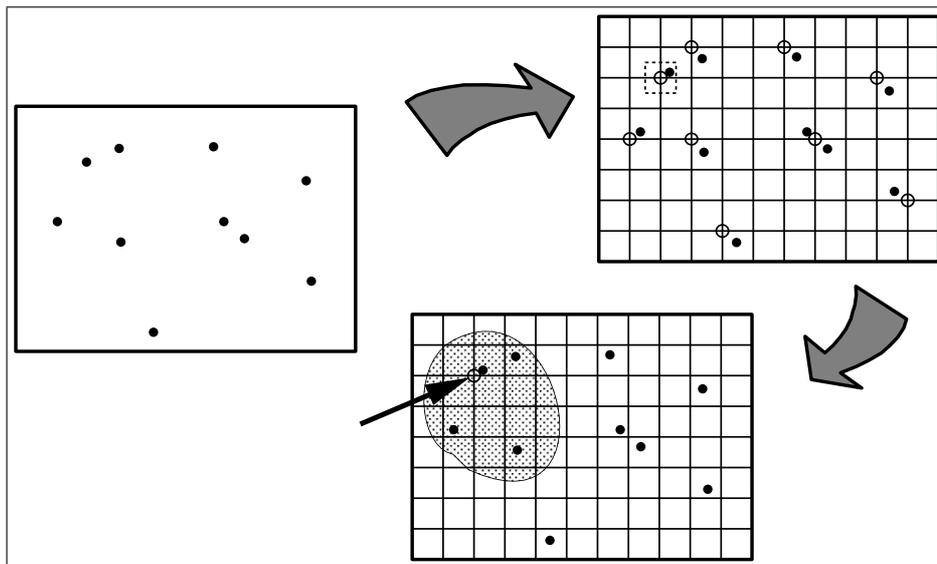


Abbildung 2.4: Der erste Schritt des Verfahrens von Arge, Dæhlen und Tveito. Das gestrichelte Kästchen in der zweiten Figur veranschaulicht eine mögliche Wahl der Umgebung für einen Gitterpunkt. In der dritten Figur ist zu sehen, welche Datenpunkte zur Definition des lokalen Interpolanten für den markierten Gitterpunkt gewählt werden.

Im zweiten Schritt werden alle übrigen Ordinatenwerte berechnet. Hierzu werden Diskretisierungen von Differentialgleichungen der Form<sup>3</sup>:

$$S_{uuuu} + S_{vvvv} = 0, \quad \Delta(S) = 0 \quad \text{bzw.} \quad \Delta^2(S) = 0, \quad (2.3)$$

angesetzt. Unter Einhaltung der Nebenbedingung (Interpolation der bereits bestimmten Ordinatenwerte) ergibt sich daraus ein relativ großes lineares Gleichungssystem in den unbekanntem Ordinatenwerten, welches mit dem *konjugierten Gradientenverfahren* aufgelöst wird (vgl. [SS76]). Die Lösbarkeit des Gleichungssystems ist für die oben angegebenen Differentialgleichungen bzw. geeigneten Diskretisierungen davon, gewährleistet (für eine allgemeine Betrachtung vgl. [ADT94]).

Arge, Dæhlen und Tveito verwenden bilineare Interpolation, um aus den diskreten Daten eine kontinuierliche Interpolationsfläche zu erzeugen.

Ebenso wie das MNN-Verfahren kann diese Methode auf beliebigen Parametergebieten angewendet werden. Hierbei ist zu erwähnen, daß die Diskretisierungen in Rand- oder Eckregionen derart modifiziert werden müssen, daß ausschließlich Gitterpunkte, die innerhalb des Parametergebietes liegen, in den Ansatz eingehen.

<sup>3</sup> $\Delta$  bezeichnet den gewöhnlichen Laplace Operator für bivariate Funktionen:  $\Delta(S) = S_{uu} + S_{vv}$ ;  $\Delta^2(S)$  bedeutet zweifache Anwendung des Laplace-Operators.

## 2.2 Methoden zur Lösung des parametrischen Interpolationsproblems

Das parametrische Interpolationsproblem besitzt im Gegensatz zu dem funktionalen Problem weniger Struktur. So kann das Interpolationsproblem i.a. nicht durch Ansatz *einer* parametrischen Fläche  $\mathbf{F} : \Omega \rightarrow \mathbb{R}^3$  gelöst werden. Betrachtet wird das parametrische Interpolationsproblem in folgender Form:

**Gegeben:** Ein Polyeder  $\mathcal{P}$  in  $\mathbb{R}^3$  mit Ecken  $\mathbf{V}_1, \dots, \mathbf{V}_N$  und dreieckigen Seitenflächen, das geschlossen oder berandet sein kann.<sup>4</sup>

**Gesucht:** Eine globale tangential- oder  $G^1$ -stetige Fläche, welche die Ecken des Polyeders interpoliert und dessen Topologie erhält.<sup>5</sup>

Alle vorgestellten Verfahren zur Lösung des parametrischen Interpolationsproblems generieren ein Kurvennetz, wobei jede Polyederkante  $\overline{\mathbf{V}_i \mathbf{V}_j}$  durch eine Raumkurve  $\mathbf{C}_{ij}$  ersetzt wird. Das Kurvennetz wird so konstruiert, daß die Kurven mit gemeinsamem Endpunkten  $G^1$ - oder  $G^2$ -kompatibel sind und die Ecken des Polyeders interpolieren. Solche Kurvennetze werden als (interpolierende)  $G^1$ - bzw.  $G^2$ -Kurvennetze bezeichnet.

Um ausgehend von einem Kurvennetz eine reguläre tangentialstetige Fläche zu bestimmen, muß das Kurvennetz gewisse Eigenschaften erfüllen.

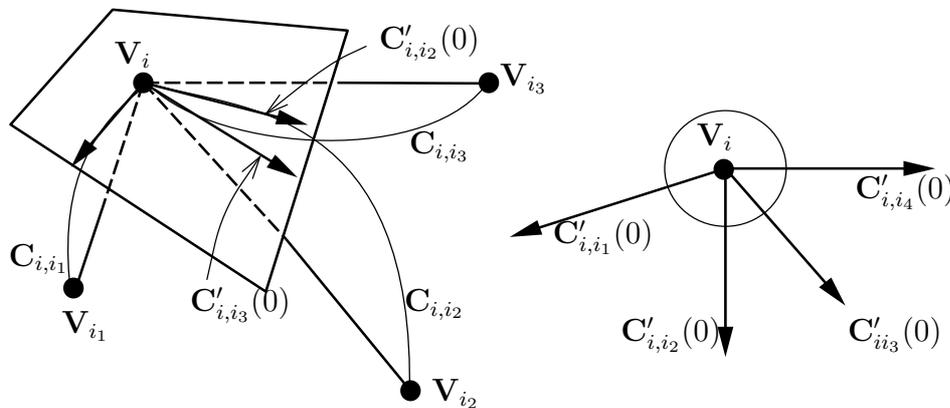


Abbildung 2.5: Eigenschaften eines Kurvennetzes: Links ein nicht ordnungserhaltendes Kurvennetz; die Ordnung der Kanten  $\overline{\mathbf{V}_i \mathbf{V}_{i_1}}, \overline{\mathbf{V}_i \mathbf{V}_{i_2}}, \overline{\mathbf{V}_i \mathbf{V}_{i_3}}$  entspricht nicht der Ordnung der Tangenten der Kurven in der Tangentialebene bei  $\mathbf{V}_i$ . Rechts ein nicht winkel-reguläre Konfiguration von Tangentialvektoren; der Winkel zwischen  $\mathbf{C}'_{i,i_4}(0)$  und  $\mathbf{C}'_{i,i_1}(0)$  ist größer als  $\pi$ .

<sup>4</sup>Genauer: Es handelt sich um eine, möglicherweise geschlossene, orientierte, triangulierte, zweidimensionale Mannigfaltigkeit.

<sup>5</sup>D.h., es gibt einen Homöomorphismus zwischen dem Polyeder und der Fläche, der die Eckpunkte des Polyeders festhält.

- Bezeichnung 2.2.1** (i) Betrachten wir die Ecke  $\mathbf{V}_i$  des Polyeders  $\mathcal{P}$  und die dort ausgehenden, geordneten Kanten  $\overline{\mathbf{V}_i \mathbf{V}_{i_k}}$ ,  $k = 0, \dots, l$ . Ein  $G^1$ -Kurvennetz  $(\mathbf{C}_{ij})_{ij \in E}$  heißt *ordnungserhaltend* bzgl. des Polyeders  $\mathcal{P}$ , falls die Tangenten der Kurven in jeder Ecke  $\mathbf{V}_i$  die durch die Polyederkanten vorgegebene Ordnung erhalten (vgl. Abbildung 2.5 links).
- (ii) Ein Kurvennetz heißt *winkel-regulär*, falls alle auftretenden Winkel zwischen den Tangenten der Kurven in den Ecken  $> 0$  und  $< \pi$  sind. Ausgenommen sind hier Winkel zwischen Tangenten, deren Kurven zum Rand des Kurvennetzes gehören (vgl. Abbildung 2.5 rechts).

Wie für den funktionalen Fall werden im weiteren zunächst die nicht optimierenden und die optimierenden Verfahren diskutiert. Abschließend wird eine Modifikation des Ansatzes von Loop zur Generierung von Splineflächen mit beliebiger Topologie aufgezeigt, mit der diese Methode zur Lösung des parametrischen Interpolationsproblems eingesetzt werden kann.

## 2.2.1 Nicht optimierende Verfahren

Aus der Vielzahl der Verfahren, die zu den nicht optimierenden Interpolationsmethoden im parametrischen Fall gehören, sollen hier nur drei Repräsentanten vorgestellt werden: die Methode nach Shirman und Séquin, das Verfahren von Peters und der Ansatz von DeRose und Mann.

### Shirman und Séquin:

Das Verfahren von Shirman und Séquin wurde in [SS87] vorgestellt. In der ursprünglichen Form war es fehlerhaft, so daß in [SS91] eine Errata erschien.

Das Verfahren besteht aus den folgenden zwei Schritten:

**Schritt 1:** Ein  $G^1$ -Kurvennetz bestehend aus kubischen Bézier-Kurven wird generiert.

**Schritt 2:** Die dreieckigen Öffnungen des Netzwerkes werden mit jeweils drei Dreiecks-Bézierflächen vom Grad 4, d.h. mit einer  $3 \times 4$  Splitfläche, gefüllt, so daß eine sich eine global  $G^1$ -stetig Fläche ergibt.

Zur Bestimmung des kubischen  $G^1$ -Kurvennetzes wird zunächst an jeder Ecke des Polyeders die Flächennormale  $\hat{\mathbf{n}}_i$  abgeschätzt. Dies geschieht mit Hilfe der Normalen der angrenzenden Dreiecksseiten des Polyeders, welche gewichtet aufsummiert werden. Gewichtet werden kann z.B. gleichmäßig, mittels des Flächeninhalts der entsprechenden Dreiecksseiten des Polyeders, mit dem reziproken Flächeninhalt der Dreiecksseiten oder mit dem einschließenden Winkel der Dreiecksseite bei der entsprechenden Ecke.

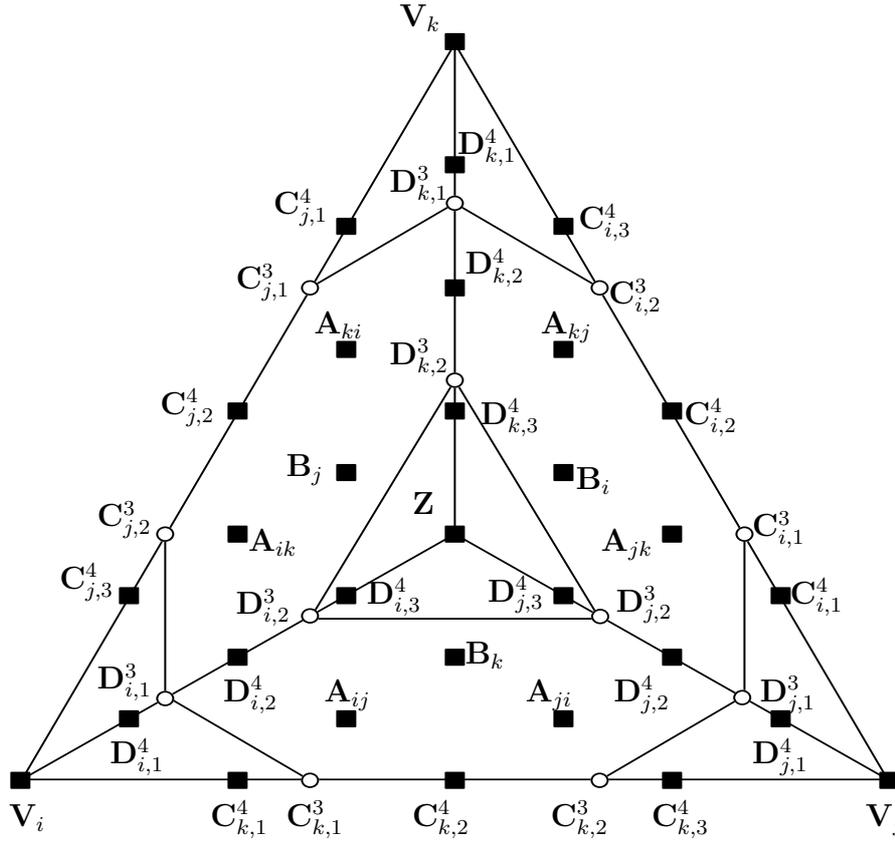


Abbildung 2.6: Die schematische Darstellung des  $3 \times 4$ -Splits nach Shirman und Séquin. Für die Randkurven werden sowohl die kubischen als auch die quartischen Kontrollpunkte, gekennzeichnet durch den Superskript <sup>3</sup> bzw. <sup>4</sup> benötigt.

Für die zu der Kante  $\overline{V_i V_j}$  gehörende kubische Kurve  $C_{ij}$  des  $G^1$ -Kurvennetzes müssen noch die Tangenten an den Endpunkten festgelegt werden. Dazu wird die Kante  $\overline{V_i V_j}$  durch Projektion entlang der geschätzten Flächennormalen  $\hat{\mathbf{n}}_i$  auf die Tangentialebene bei  $V_i$  projiziert. Zudem wird die Länge des Tangentenvektors der Kurve durch  $\|C'_{ij}(0)\| = \|V_i - V_j\|$  festgelegt. Durch analoges Vorgehen bei  $V_j$  ist die kubische Kurve  $C_{ij}$  vollständig bestimmt.

Das Einfügen der Kontrollpunkte der Dreiecks-Bézierflächen wird für  $(r, s, t) \in \{(i, j, k), (j, k, i), (k, i, j)\}$  wie folgt durchgeführt:

- (i)  $D_{r,1}^3$  ist das Zentrum des Dreiecks  $\Delta(V_r, C_{t,1}^3, C_{s,2}^3)$ , d.h. mit  $\beta_r = \frac{1}{3}$ :

$$D_{r,1}^3 = (1 - 2\beta_r) V_r + \beta_r (C_{t,1}^3 + C_{s,2}^3) \quad (2.4)$$

- (ii)  $A_{rs}$  und  $A_{sr}$  werden mit Hilfe der Methode von Chiyokura und Kimura (vgl. Abschnitt 1.2 auf Seite 11) berechnet. Dazu werden in den Ecken Querableitungsvektoren  $\vec{\mathbf{q}}_{t,0}$  und  $\vec{\mathbf{q}}_{t,2}$  bestimmt, die senkrecht zu den Tangentialvektoren  $C_{t,1}^3 - V_r$  bzw.  $V_s - C_{t,2}^3$  in den entsprechenden Tangentialebenen liegen. Für lineare Gewichtsfunktionen  $a_t, b_t$  und

$$D_{r,1}^4 - V_r = a_{t,0} (C_{t,1}^3 - V_r) + b_{t,0} \vec{\mathbf{q}}_{t,0} \quad \text{und} \quad D_{s,1}^4 - V_s = a_{t,1} (V_s - C_{t,2}^3) + b_{t,1} \vec{\mathbf{q}}_{t,2},$$

ergibt sich

$$\mathbf{A}_{rs} = \mathbf{C}_{t,1}^3 + \frac{1}{3} (2a_{t,0} (\mathbf{C}_{t,2}^3 - \mathbf{C}_{t,1}^3) + a_{t,1} (\mathbf{C}_{t,1}^3 - \mathbf{V}_r) + b_{t,0} \vec{\mathbf{q}}_{t,1} + b_{t,1} \vec{\mathbf{q}}_{t,0}). \quad (2.5)$$

Einen analogen Ausdruck erhält man für  $\mathbf{A}_{sr}$ . Die Querableitungsvektoren  $\vec{\mathbf{q}}_{t,1}$ ,  $\vec{\mathbf{q}}_{r,1}$  und  $\vec{\mathbf{q}}_{s,1}$  werden durch Mittelung der Querableitungsvektoren an den Endpunkten bestimmt. Damit entsteht ein lineares Querableitungsvektorfeld.

- (iii) Der innere Kontrollpunkt  $\mathbf{D}_{r,2}^3$  ergibt sich aufgrund der  $G^1$ -Bedingungen nach Farin [Far82] (vgl. Abschnitt 1.2 auf Seite 10) mit:

$$\mathbf{D}_{r,2}^3 = 2\beta_r (\mathbf{A}_{rs} + \mathbf{A}_{rt}) + \frac{\beta_r}{2} \mathbf{V}_r + \left(1 - \frac{9}{2}\beta_r\right) \mathbf{D}_{r,1}^3. \quad (2.6)$$

- (iv) Der Kontrollpunkt  $\mathbf{B}_r$  ergibt sich zu:

$$\begin{aligned} \mathbf{B}_r = & \frac{1}{4} (\mathbf{D}_{r,1}^3 - \mathbf{D}_{s,1}^3 - \mathbf{D}_{t,1}^3) + \left(\frac{1}{12} \left(\frac{1}{\beta_r} + \frac{1}{2\beta_s} + \frac{1}{2\beta_t}\right) - \frac{5}{4}\right) \mathbf{D}_{r,2}^3 \\ & \left(\frac{1}{12} \left(\frac{1}{2\beta_t} - \frac{1}{\beta_s} - \frac{1}{2\beta_r}\right) + \frac{5}{4}\right) \mathbf{D}_{s,2}^3 + \left(\frac{1}{12} \left(\frac{1}{2\beta_s} - \frac{1}{\beta_t} - \frac{1}{2\beta_r}\right) + \frac{5}{4}\right) \mathbf{D}_{t,2}^3 \end{aligned} \quad (2.7)$$

- (v) Schließlich wird das Zentrum  $\mathbf{Z}$  in den Schwerpunkt des Dreiecks  $\Delta (\mathbf{D}_{r,2}^3, \mathbf{D}_{s,2}^3, \mathbf{D}_{t,2}^3)$  gelegt, d.h.:

$$\mathbf{Z} = \frac{1}{3} (\mathbf{D}_{r,2}^3 + \mathbf{D}_{s,2}^3 + \mathbf{D}_{t,2}^3). \quad (2.8)$$

Wir haben hier eine erweiterte Darstellung gewählt. In der ursprünglichen Version von Shirman und Séquin wird stets  $\beta_r = \beta_s = \beta_t = \frac{1}{3}$  gesetzt, was die Formeln wesentlich vereinfacht. In Abschnitt 4.2 werden wir aber eine Modifikation des ursprünglichen Verfahrens angeben, die auf dieser allgemeineren Darstellung aufbaut.

Das Verfahren von Shirman und Séquin löst das oben gestellt Interpolationsproblem, falls das generierte Kurvennetz winkel-regulär und ordnungserhaltend bzgl. des Polyeders  $\mathcal{P}$  ist. In diesem Fall erhält man sogar eine globale  $G^1$  Interpolationsfläche.

### Peters:

Peters [Pet91] untersucht das allgemeine Problem, zu einem gegebenen ordnungserhaltenden und winkelregulären kubischen  $G^1$ -Kurvennetz eine globale  $G^1$  Fläche zu konstruieren. Hierbei betrachtet er Kurvennetze mit drei- und viereckigen Öffnungen und versucht nur eine einzige Bézierfläche pro Öffnung zu verwenden. Aufgrund des Eckenumschlußproblems (abgekürzt: EUP; vgl. Abschnitt 1.2 auf Seite 12) müssen Einschränkungen an das gegebene Kurvennetz gemacht werden, d.h. es wird die Lösbarkeit des EUP an jeder Polyederecke vorausgesetzt.

Die ursprüngliche Methode von Peters sieht eine starke Einschränkung bei der Konfiguration von aneinanderstoßenden Dreiecks-Bézierflächen vor. Dies geschieht um die Polynomgrade der Bézierflächen auf 4 bzw.  $3 \times 3$  zu beschränken. Läßt man den Grad 5 zu, so entfällt diese Einschränkung.

Eine genaue Untersuchung der Methode von Peters zeigt, daß für die speziellen Ansätze der  $G^1$  Übergänge zwischen zwei Teilflächen  $\mathbf{S}_1$  und  $\mathbf{S}_2$ , d.h. für

$$\alpha(u)(\mathbf{S}_1)_u(u) + \beta(u)(\mathbf{S}_1)_v(u) + \gamma(u)(\mathbf{S}_2)_w(u) \equiv 0, \quad \beta(u)\gamma(u) > 0$$

der Grad des Gewichtspolynoms  $\alpha$  zu niedrig angesetzt ist. Die Folge ist, daß die Gleichungssysteme, die sich aus dem EUP an jeder Ecke ergeben, i.a. nicht entkoppelt sind. Zur Entkoppelung muß gefordert werden:  $\deg(\alpha) \geq 3$ . Zur Lösung dieses Konflikts gibt es i.w. drei Möglichkeiten:

**Höherer Flächengrad:** Da unter bestimmten geometrischen Voraussetzungen  $\deg(\beta) = 0$  oder  $\deg(\gamma) = 0$  sein kann, muß der Polynomgrad 6 angesetzt werden.

**Gradreduktion der Gewichte:** Die Gradreduktion der Gewichtspolynome *nach* Lösen der entkoppelt betrachteten EUP führt i.a. auf nicht exakte  $G^1$  Übergänge.

**Approximatives Lösen des EUP:** Unter Fixierung der Gewichtsfunktionen  $\alpha, \beta$  und  $\gamma$  werden die Twistpunkte so bestimmt, daß sie bestmöglich, d.h. mit kleinstem Fehlerquadrat, das EUP lösen.

Da die Lösbarkeit des EUP bei kubischen Kurven für allgemeine Konfigurationen nicht garantieren werden kann (auch [Pet91] gibt keine Methode an, *wie* ein EUP-kompatibles kubische  $G^1$ -Kurvennetz konstruiert werden kann), haben wir uns für das approximative Lösen des EUP entschieden.

### DeRose und Mann:

DeRose und Mann befassen sich in [DM92] mit der *Approximation* von gegebenen Flächen durch kubische Dreiecks-Bézierflächen (vgl. auch [Man92]). Hierzu werden an diskreten Punkte auf der zu approximierenden Fläche die Normalenvektoren und die Krümmungen ermittelt. Diese Punkte werden dann geeignet zu einem Polyeder mit Dreiecksseiten zusammengefügt.

Bei der Konstruktion des kubischen  $G^1$ -Kurvennetzes wird versucht, die bereits vorhandene Normal- und Krümmungsinformation an den Ecken des Polyeders zu interpolieren. DeRose und Mann verwenden dazu die *Geometrischen Hermite-Polynome* von de Boor, Höllig und Sabin [dBHS87].

Dabei handelt es sich um ebene kubische Kurven. Durch Anpassung der Tangentenlängen  $\delta_0, \delta_1$  an den Endpunkten wird versucht die vorgeschriebenen Krümmungen  $\kappa_1$  und  $\kappa_2$  zu interpolieren (vgl. Abbildung 2.7). Dieser Ansatz führt auf eine Gleichung vierten Grades in den unbekanntenen Tangentenlängen  $\delta_0$  und  $\delta_1$ , die nicht immer lösbar ist.

Durch Definition einer Ebene  $\mathbf{E}$ , die die Ecken  $\mathbf{V}_i$  und  $\mathbf{V}_j$  sowie den gemittelten Normalenvektor  $\hat{\mathbf{n}}_i + \hat{\mathbf{n}}_j$  enthält, wird die Raumkurve  $\mathbf{C}_{ij}$  des kubischen Kurvennetzes mit dem Verfahren von de Boor, Höllig und Sabin definiert.

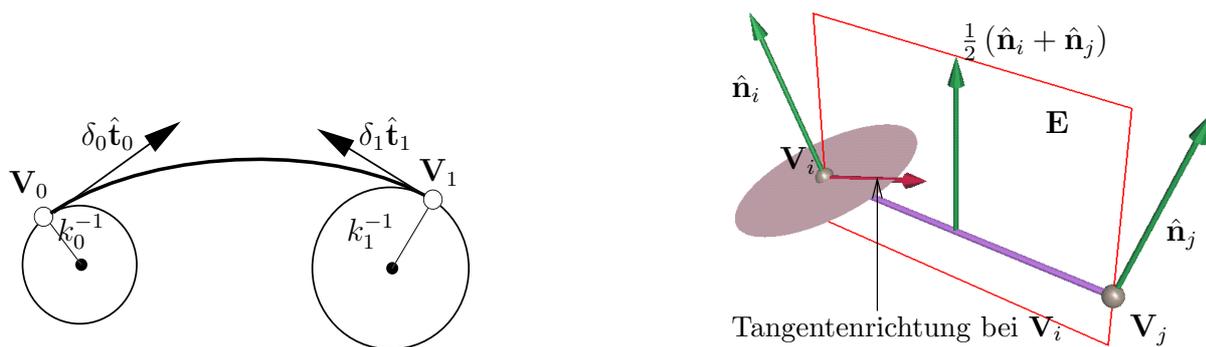


Abbildung 2.7: Die kubische geometrische Hermite-Interpolationskurve zu gegebener Position, Tangente und Krümmungsradius und das Einpassen der Ebene  $E$  zu der Kante  $\overline{V_i V_j}$ .

Zur vollständigen Definition der kubischen Dreiecks-Bézierflächen muß noch der innerste Kontrollpunkt  $P_{111}$  bestimmt werden. DeRose und Mann versuchen  $P_{111}$  so zu legen, daß die entstehende Fläche in den drei Eckpunkten die gegebene Krümmungsinformation interpoliert. Damit diese Bedingung an der Ecke  $V_i$  erfüllt ist, muß die gemischte Ableitung der betrachteten Bézierfläche  $S$  der folgenden Kompatibilitätsbedingung mit der zweiten Fundamentalform  $\|_{V_i}$  genügen (hier ist  $V_i = P_{300}$ ):

$$\|_{V_i}(\vec{u}, \vec{v}) = \langle \hat{n}_i | S_{\vec{u}\vec{v}} \rangle = 6 \langle \hat{n}_i | P_{111} - P_{300} \rangle,$$

wobei  $\vec{u} = 3(P_{210} - P_{300})$  und  $\vec{v} = 3(P_{201} - P_{300})$  die entsprechenden Tangentialvektoren der Randkurven in  $V_i = P_{300}$  sind.

Diese Bedingung ist genau dann eindeutig an allen drei Ecken  $V_i, V_j, V_k$  der Dreiecksfläche erfüllbar, wenn die drei Normalenvektoren  $\hat{n}_i, \hat{n}_j, \hat{n}_k$  linear unabhängig sind.

Die resultierende Interpolationsfläche ist nicht global tangentialstetig. DeRose und Mann geben eine Verfeinerungsmethode an, bei der durch Hinzunahme weiterer Punkte der ursprünglichen Fläche (incl. Normalen und Krümmungsinformation) der Fehler in der Tangentialstetigkeit verringert werden kann ( $\varepsilon - G^1$ -Stetigkeit).

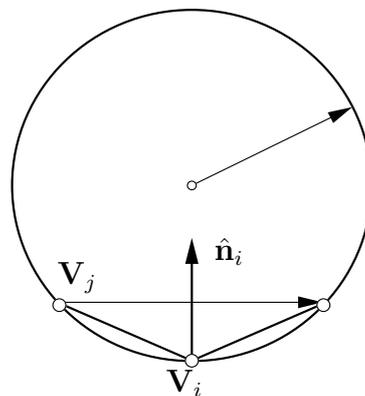


Abbildung 2.8: Abschätzung der Normalenkrümmung nach Moreton und Séquin.

Setzt man die Methode von DeRose und Mann auf das Interpolationsproblem im parametrischen Fall an, so müssen die Normalen- und die Krümmungsinformationen in den Polyederecken  $V_i$  abgeschätzt werden. Die Normalen lassen sich wie beim Verfahren von Shirman und Séquin (vgl. Seite 26) ermitteln. Zur Abschätzung der zweiten Fundamentalform wird das von Moreton und Séquin [MS92] angegebene Verfahren verwendet. Dieses beruht auf der Abschätzung der Normalenkrümmungen für die Tangentialrichtungen  $\hat{t}_{ij}$ ,

die durch die Projektion der Kanten  $\overline{\mathbf{V}_i \mathbf{V}_j}$  auf der Tangentialebene bei  $\mathbf{V}_i$  gegeben sind. Der Krümmungskreis bei  $\mathbf{V}_i$  in Richtung  $\mathbf{t}_{ij}$  wird durch die Punkte  $\mathbf{V}_i, \mathbf{V}_j$  und durch den an der Normalen  $\hat{\mathbf{n}}_i$  gespiegelten Punkt  $\mathbf{V}_j$  bestimmt (vgl. Abbildung 2.8). Die approximative Lösung des entstehenden, i.a. überbestimmten, Gleichungssystems liefert Schätzwerte für die Einträge der zweiten Fundamentalform.

### 2.2.2 Optimierende Verfahren

Aus der Klasse der optimierenden Verfahren werden der Vorschlag Nielsons zur Erweiterung des MNN-Ansatzes auf den parametrischen Fall und das Verfahren von Moreton und Séquin beschrieben.

#### Nielsons Minimum-Norm-Netzwerke

In [Nie88] schlägt Nielson eine Erweiterung seiner in [Nie83] eingeführten MNN-Methode auf den parametrischen Fall vor. Im einzelnen besteht diese Methode aus folgenden Schritten:

**Lokale Parametrisierung:** Durch einfache Mittelwertbildung werden an den Eckpunkten des Polyeders aus den Normalen der umliegenden Dreiecksseiten Vektoren  $\hat{\mathbf{w}}_i$  abgeschätzt. Die Vektoren  $\hat{\mathbf{w}}_i$  representieren die lokale  $z$ -Achse bei  $\mathbf{V}_i$ .  $\hat{\mathbf{w}}_i$  wird zu einem orthonormalen Koordinatensystem  $\{\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i, \hat{\mathbf{w}}_i\}$  ergänzt.

Durch orthogonale Projektion der entsprechenden Polyederkanten  $\overline{\mathbf{V}_i \mathbf{V}_j}$  auf die  $(\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i)$ -Ebene und anschließender Normierung erhält man Richtungsvektoren  $\hat{\mathbf{e}}_{ij}$ .

**Optimierung der Normalen und Generierung des Kurvennetzes:** Die Variablen für die Optimierung im funktionalen Fall sind die unbekanntenen partiellen Ableitungen  $S_u(\mathbf{u}_i), S_v(\mathbf{u}_i)$  in den Datenpunkten. Zusätzliche geometrische Größen, die im funktionalen Fall eine Rolle spielen, sind die Koeffizienten  $\alpha_{ij}, \beta_{ij}$  der Kantenvektoren  $\mathbf{u}_j - \mathbf{u}_i, ij \in E$  in der kanonischen  $(u, v)$ -Basis der Parameterebene (vgl. Abschnitt 2.1.2 auf Seite 21).

Für den parametrischen Fall modifiziert Nielson das aus dem funktionalen Fall explizit bekannte Gleichungssystem zur Optimierung des Kurvennetzes. Dazu werden die Koeffizienten  $\alpha_{ij}, \beta_{ij}$  durch  $\hat{\mathbf{e}}_{ij} = \alpha_{ij} \hat{\mathbf{u}}_i + \beta_{ij} \hat{\mathbf{v}}_i$  definiert. Die Variablen  $S_u(\mathbf{u}_i), S_v(\mathbf{u}_i)$  werden durch die Komponenten  $\hat{\mathbf{u}}_{i,k}, \hat{\mathbf{v}}_{i,k}, k = 1, 2, 3$  der zu optimierenden lokalen Basisvektoren  $\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i$  ersetzt. Es entsteht ein  $6N \times 6N$  Gleichungssystem.

Die optimierte Normale ergibt sich durch die neu bestimmten Vektoren  $\vec{\mathbf{u}}_i$  und  $\vec{\mathbf{v}}_i$  :  $\hat{\mathbf{n}}_i = (\vec{\mathbf{u}}_i \times \vec{\mathbf{v}}_i) / \|\vec{\mathbf{u}}_i \times \vec{\mathbf{v}}_i\|$ .

Die Richtung der Tangente der kubischen Kurve  $\mathbf{C}_{ij}$  des  $G^1$ -Kurvennetzes wird durch Projektion der Kante  $\overline{\mathbf{V}_i \mathbf{V}_j}$  auf die durch  $\hat{\mathbf{n}}_i$  definierte Tangentialebene bestimmt. Die Länge der Tangente wird als Shape-Parameter verwendet: Der Benutzer spezifiziert den Parameter  $\gamma_i$  bzw.  $\gamma_{ij}$  für jede Ecke bzw. für jede Kante des Polyeders. Die Längen der Tangente der Kurve  $\mathbf{C}_{ij}$  am Anfangspunkt wird durch  $\max\{\gamma_i, \gamma_{ij}\}$  festgelegt.

**Generierung der Flächen:** Durch Verwendung der in [Nie87] für den parametrischen Fall vorgestellten Variante des transfiniten Interpolanten über Dreiecken (vgl. Abschnitt 1.2 und [Nie79]) wird das Kurvennetz zu einer globalen  $G^1$  Fläche fortgesetzt.

In dieser ursprünglichen Version wird das Kurvennetz nur einmal optimiert. Die parametrischen Kurvennetzmethoden, wie sie in den Abschnitten 4 und 5 vorgestellt werden, arbeiten iterativ, d.h. die durch die Optimierung des Kurvennetzes erzeugten Normalen werden im nächsten Schritt als lokale  $z$ -Achse verwendet. Die Methode von Nielson wurde in der vorliegenden Implementation ebenfalls um einem derartigen Iterationsansatz erweitert.

Problematisch scheint bei der von Nielson vorgeschlagenen Methode die direkte Abwandlung der Optimierungsmatrix des funktionalen Problems. Zudem ist die Handhabung der Shape-Parameter in diesem Fall schwieriger als im funktionalen Fall: Wählt man, was häufig gemacht wird, einen globalen Parameter  $\gamma = \gamma_i = \gamma_{ij}$ , so sind, unbeachtet der Geometrie des Problems, *alle* Endtangente der Kurven des Kurvennetz *gleich lang*. Ähnlich verhält es sich bei alleiniger Verwendung der Tensions-Parameter  $\gamma_i$  in den Ecken. Wie unsere Tests gezeigt haben, muß diese „Steifigkeit“ in der Festlegung der Tangentenlängen aufgehoben werden. Dies kann z.B. dadurch erreicht werden, daß die Tangentenlängen zunächst wie beim Verfahren von Shirman und Séquin entsprechend der Länge der Polyederkante bestimmt werden (vgl. Abschnitt 2.2.1 auf Seite 26). Anschließend werden die Tangenten mit dem Shape-Parameter  $\gamma$  skaliert.

### Verfahren von Moreton und Séquin

Moreton und Séquin stellen in [MS92] ein Verfahren zur Lösung des parametrischen Interpolationsproblems vor, das auf einer funktionalen Optimierung des Kurvennetzes und der eingefügten Flächenstücke beruht (vgl. auch [MS91, Mor92]). Hierbei werden die exakten Funktionale zur Bestimmung der Krümmungsvariation des Kurvennetzes und der Flächensegmente angesetzt. Das optimierte Kurvennetz wird als *MVC-Kurvennetz* (*minimum variation curves*), die optimierte Flächen als *MVS-Flächen* (*minimum variation surfaces*) bezeichnet.

Das Kurvennetz wird aus quintischen Bézier-Kurven zusammengesetzt, wobei die Kurven an den gemeinsamen Eckpunkten  $G^2$  kompatibel sind. Die von Moreton und Séquin verwendete Repräsentation des quintischen  $G^2$ -Kurvennetzes können neben den Flächenparametern in den Eckpunkten (Flächennormalen, zweiten Fundamentalform) auch alle Kurvenparameter variiert werden, die nicht durch die  $G^2$ -Kompatibilität bestimmt sind.

Die Optimierung des Kurvennetzes beruht auf dem Funktional zur Bewertung der Krümmungsvarianz einer Kurve  $\mathbf{C}$ :

$$\sigma(\mathbf{C}) = \int_0^{l(s)} \left( \frac{d\mathbf{k}}{ds} \right)^2 ds, \quad (2.9)$$

wobei bzgl. der Bogenlänge von  $\mathbf{C}$  integriert wird. Das Funktional für das gesamte Kurvennetz ergibt sich als Aufsummierung aller Kurvenfunktionale. Die Optimierung kann aufgrund des hochgradig nichtlinearen Funktionals nur numerisch gehandhabt werden. Moreton und Séquin verwenden hier das konjugierte Gradientenverfahren (vgl. [SS76]). Da das

Kurvennetz nicht linear von den variablen Größen des Kurvennetzes abhängt, kommt eine Linearisierung unter Ansatz eines quadratischen Funktionals nicht in Betracht (vgl. Abschnitt 1.3).

Das Kurvennetz wird mit quintischen Dreiecks-Bézierflächen aufgefüllt. Optional lassen Moreton und Séquin auch viereckige Polyederseiten zu und setzen bei solchen Seiten dann biquintische TP-Bézierflächen an.

In der ersten Variante des Verfahrens wird das optimierte Kurvennetz von den einzupassenden Bézierflächen interpoliert. Alternativ können gleichzeitig die Randkontrollpunkte und die inneren Kontrollpunkte der Bézierflächen variiert werden. Das angesetzte MVS-Flächenfunktional hat folgende Gestalt

$$\sigma(\mathbf{S}) = \int \left( \frac{\partial \kappa_1}{\partial \hat{\mathbf{e}}_1} \right)^2 + \left( \frac{\partial \kappa_2}{\partial \hat{\mathbf{e}}_2} \right)^2 d\omega.$$

Anders als für das Kurvennetz, integrieren Moreton und Séquin beim Einpassen der Flächen nicht die Bedingungen für die Tangentialstetigkeit in die Darstellung des Flächennetzes. Sie setzen eine Abwandlung der DeRose-Bedingungen für Tangentialstetigkeit als *Fehlerfunktional* an (vgl. Abschnitt 1.2 auf Seite 12 und [DeR90]) und kombinieren dieses mit dem Optimierungsfunktional  $\sigma$ . Für einen konkreten Übergang zwischen den Flächen  $\mathbf{S}_1$  und  $\mathbf{S}_2$  verwenden Moreton und Séquin den folgenden Ansatz:

$$\rho(\mathbf{S}) = \sum_{m=0}^{n_c+n_1+n_2} \left( \sum_{i+j+k=m} \frac{\binom{n_c}{i} \binom{n_1}{j} \binom{n_2}{k}}{\binom{m}{n_c+n_1+n_2}} |\vec{\mathbf{u}}_i, \vec{\mathbf{v}}_j, \vec{\mathbf{w}}_k| \right)^2.$$

Für  $\rho(\mathbf{S}) = 0$  ist nach DeRose [DeR90] Tangentialstetigkeit von  $\mathbf{S}_1$  und  $\mathbf{S}_2$  entlang der betrachteten Kante garantiert.

Moreton und Séquin setzen eine Kombination der beiden Funktionale  $\sigma$  und  $\rho$  an, wobei im Verlauf der Optimierung das MVS-Funktional  $\sigma$  „ausgeblendet“ wird. Dadurch wird erreicht, daß der Einfluß des Fehlerfunktionals gegen Ende der Optimierung zunimmt, und damit der Fehler in der Tangentialstetigkeit der Flächen minimiert wird.

Das Verfahren von Moreton und Séquin liefert i.a. Interpolationsflächen von sehr hoher Qualität. Der Preis hierfür ist ein sehr großer Rechenaufwand. Zudem ist die Wahl der geeigneten Gewichtung des Fehler- und des Krümmungsfunktionalen bei der Flächenoptimierung eine diffizile Angelegenheit (vgl. [Mor92, Wei94]).

### 2.2.3 Splineflächen mit beliebiger Topologie

In den letzten Jahren wurde eine Reihe von Methoden zur Generierung von polynomiellen Splineflächen mit Kontrollpolyedern beliebiger Topologie vorgestellt (vgl. [LD90, Pet93, Loo94a, Loo94b, Pet94]). Die grundsätzliche Idee dieser Verfahren ist es, möglichst einfache Algorithmen zur Erzeugung von stückweise polynomiellen Flächen für allgemeinere Kontrollstrukturen als für reguläre TP-Kontrollnetze zu erhalten.

Diese meisten dieser Verfahren arbeiten nach folgendem Prinzip:

**Verfeinerung des Polyeders:** In einem ersten Schritt wird das Ausgangs- oder Kontrollpolyeder durch Anwendung von Unterteilungsschritten verfeinert. Normalerweise werden ein oder zwei solcher Unterteilungsschritte durchgeführt.

**Generierung von Bézierflächen:** Unter Ausnutzung lokaler Regularitätseigenschaften des verfeinerten Polyeders lassen sich Bézierflächen (Dreiecks- oder TP-Bézierflächen) zu diesem verfeinerten Polyeder definieren, die relativ einfachen  $G^1$  Übergangsbedingungen genügen.

Der Vorteil dieser Verfahren liegt in der einfachen Formulierung der  $G^1$  Übergänge, die eine sehr schnelle Generierung der einzelnen Bézierflächen ermöglicht. Zudem haben die Bézierflächen relativ niedrigen Polynomgrad. Die Ecken des Ausgangspolyeders werden von diesen Verfahren nicht interpoliert. Diese Algorithmen generieren meist sehr viele Bézierflächen.

Am Beispiel der Methode von Loop [Loo94a] wird im folgenden das Potential dieser Splinesflächen zur Lösung des parametrischen Interpolationsproblems untersucht. Hierzu wird zunächst der Algorithmus nach Loop beschrieben. Anschließend wird eine Möglichkeit angegeben, wie dieser Algorithmus auf das parametrische Interpolationsproblem angewendet werden kann.

### Das Verfahren von Loop

Die in [Loo94a] beschriebene Methode zur Definition von Splineflächen verwendet quartische Dreiecks-Bézierflächen zur lokalen Repräsentation der Fläche. Im Gegensatz zu den Algorithmen von Peters [Pet93, Pet94] wird bei Loop nur ein Unterteilungsschritt zur Verfeinerung des Polyeders durchgeführt. Hierzu wird für jede Seitenfläche des Polyeders eine Reihe neuer Punkte konstruiert, welche dann die Eckpunkte des neuen, verfeinerten Polyeders ergeben.

Sei  $\mathcal{F}$  eine  $k$ -eckige Seitenfläche des Ausgangspolyeders mit den Ecken  $\mathbf{V}_{i_1}, \dots, \mathbf{V}_{i_k}$ .  $\mathcal{O}$  sei das einfache Mittel der Eckpunkte, d.h.  $\mathcal{O} = \frac{1}{k}\{\mathbf{V}_{i_1} + \dots + \mathbf{V}_{i_k}\}$ . Für die Ecke  $\mathbf{V}_{i_j}$  wird die neue Ecke  $\mathbf{Q}_{i_j}$  definiert durch:

$$\mathbf{Q}_{i_j} = \frac{1}{4}\mathcal{O} + \frac{1}{8}\mathbf{V}_{i_{j-1}} + \frac{1}{2}\mathbf{V}_{i_j} + \frac{1}{8}\mathbf{V}_{i_{j+1}}. \quad (2.10)$$

Das verfeinerte Polyeder ergibt sich durch geeignete Verbindung der neuen Punkte: Jeder neue Punkt  $\mathbf{Q}_{i_j}$  wird mit seinen Nachbarn  $\mathbf{Q}_{i_{j-1}}$  und  $\mathbf{Q}_{i_{j+1}}$  derselben Polyederseite sowie mit den jeweiligen neuen Punkten der an  $\mathcal{F}$  in  $\mathbf{V}_{i_j}$  angrenzenden Seiten verbunden. Die Ecken des verfeinerten Polyeders sind damit immer mit genau 4 Punkten verbunden, sofern sie nicht auf dem Rand des Polyeders liegen (vgl. Abbildung 2.9).

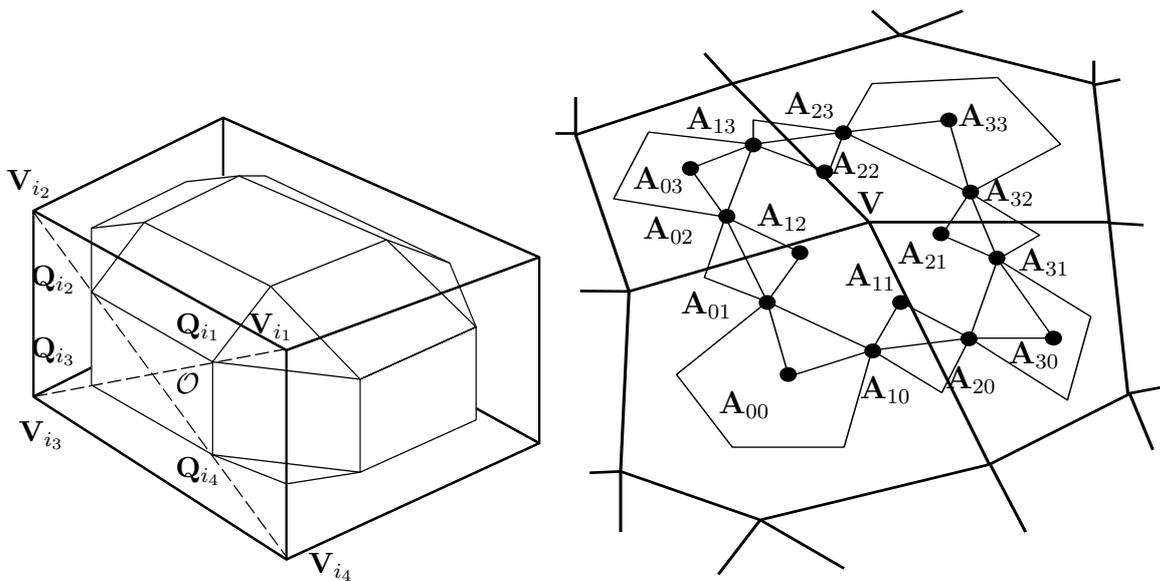


Abbildung 2.9: Links: Die Durchführung des Verfeinerungsschrittes am Beispiel eines Quaders; Rechts: Zu der Ecke  $V$  des verfeinerten Polyeders wird ein Quad-Net generiert.

Nach der Verfeinerung des Polyeders wird für jede Ecke  $V$  des verfeinerten Polyeders ein sogenanntes *Quad-Net* definiert. Quad-Nets beschreiben die Splinefläche lokal und werden als Zwischenstruktur für die Generierung der eigentlichen Flächenelemente, den Dreiecks-Bézierflächen, verwendet.

Das Quad-Net wird durch die vier kubischen Bézierkurven mit den Kontrollpunkten  $[A_{00}, A_{10}, A_{20}, A_{30}]$ ,  $[A_{30}, A_{31}, A_{32}, A_{33}]$ ,  $[A_{33}, A_{23}, A_{13}, A_{03}]$  und  $[A_{03}, A_{02}, A_{01}, A_{00}]$  begrenzt (vgl. Abbildung 2.9). Diese Randpunkte werden von benachbarten Quad-Nets geteilt. Die Ecken  $A_{00}$ ,  $A_{30}$ ,  $A_{03}$  und  $A_{33}$  des betrachteten Quad-Nets fallen auf die Mittelpunkte der umliegenden Seitenflächen des verfeinerten Polyeders. Die zusätzlichen Kontrollpunkte  $A_{11}$ ,  $A_{21}$ ,  $A_{22}$  und  $A_{12}$  werden zur Definition der Querableitungen entlang der entsprechenden Randkurve benötigt.

Man beachte, daß die Bezeichnung der Quad-Net Kontrollpunkte so gewählt ist, daß  $A_{30}$  und  $A_{03}$  Mittelpunkte von Vierecken sind. Dies ist immer möglich, da bei der Verfeinerung des Ausgangspolyeders für jede Kante eine viereckige Seitenfläche entsteht.

Die Quad-Net Kontrollpunkte um einen Eckpunkt  $A_{00}$  werden so gewählt, daß sie ein gleichseitiges  $k$ -Eck mit Zentrum  $A_{00}$  bilden, wobei  $k$  die Zahl der in der betrachteten Ecke zusammenkommenden Quad-Nets ist. Analog werden die Kontrollpunkte  $A_{30}$ ,  $A_{33}$  und  $A_{03}$  ermittelt. Das Einpassen des gleichseitigen  $k$ -Ecks erfolgt so, daß sich für TP-artige Kontrollpolyeder gerade die Bézier-Darstellung der B-Splinekurven entlang der Knotenlinien ergibt. Auf ähnliche Weise werden die zusätzlichen Kontrollpunkte  $A_{11}$ ,  $A_{21}$ ,  $A_{22}$  und  $A_{12}$  festgelegt.

Als letzter Schritt werden für jedes Quad-Net vier Dreiecks-Bézierflächen bestimmt, die die Randkurven und die Querableitungen des Quad-Nets interpolieren. Da in den Ecken  $A_{00}$  die durch die Querableitungen gegebenen gemischten partiellen Ableitungen für die beiden Kanten i.a. nicht übereinstimmen, kann nicht mit einer einzigen TP-Bézierfläche

gearbeitet werden (Details zur Bestimmung der Bézier-Kontrollpunkte gibt [Loo94b]).

Loop behandelt noch einige Spezialfälle in denen bei gewissen regulären Eigenschaften des Kontrollpolyeders mit Flächen niedrigerer Ordnung gearbeitet werden kann. An der grundsätzlichen Vorgehensweise ändert sich dabei nichts.

### Interpolation von Polyederecken

In diesem Abschnitt wird eine Methode angegeben, mit der die Ecken des ursprünglichen Polyeders interpoliert werden können. Die grundsätzliche Idee ist es den ursprünglichen Polyeder „aufzublasen“, so daß die Splinefläche die durch das „aufgeblasene“ Polyeder definiert ist, die Ecken des ursprünglichen Polyeders interpoliert. Ein ähnlicher Ansatz wird auch von Halstead, Kass und DeRose verwendet um für eine spezielle Klasse von Subdivision-Flächen die Interpolationseigenschaft zu erhalten (vgl. [HKD93]).

Bei der Verfeinerung des Polyeders entsteht für jede Ecke, für jede Kante und für jede Seite des Ausgangspolyeders eine neue Seitenfläche (vgl. Abbildung 2.9). Die Mittelpunkte der Seitenflächen des verfeinerten Polyeders werden durch die Splinefläche exakt interpoliert. Dies legt es nahe den Mittelpunkt der aus der Ecke  $\mathbf{V}$  entstehenden Seitenfläche als Bildpunkt  $\bar{\mathbf{V}}$  von  $\mathbf{V}$  zu betrachten. Die Lage von  $\bar{\mathbf{V}}$  hängt von allen Ecken der bei  $\mathbf{V}$  anliegenden Seitenflächen des Ausgangspolyeders ab.

Werden die Eckpunkte des ursprünglichen und des verfeinerten Polyeders um die Ecke  $\mathbf{V}$  wie in Abbildung 2.10 bezeichnet, so ergibt sich der Bildpunkt  $\bar{\mathbf{V}}$  im Spezialfall einer Würfelcke zu  $\bar{\mathbf{V}} = \frac{9}{16}\mathbf{V} + \sum_{i=1}^3 \left(\frac{1}{8}\mathbf{V}_i + \frac{1}{48}\mathbf{V}_{i1}\right)$ . Es folgt daraus, daß die Transformationsmatrix  $T$  mit

$$T \begin{pmatrix} \mathbf{V}_1 \\ \vdots \\ \mathbf{V}_N \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{V}}_1 \\ \vdots \\ \bar{\mathbf{V}}_N \end{pmatrix} \quad (2.11)$$

diagonaldominant und damit invertierbar ist. Die Diagonaldominanz gilt auch für den allgemeinen Fall, wenn  $k$  Kanten bei  $\mathbf{V}$  anliegen. Die Kontrollpunkte des „aufgeblasenen“ Polyeders  $\tilde{\mathbf{V}}_i$  werden durch Auflösen von (2.11) mit entsprechender rechter Seite gewonnen. Damit ist sichergestellt, daß die Splinefläche zu dem durch die Kontrollpunkte  $\tilde{\mathbf{V}}_i$  definierten Polyeder genau die Ecken  $\mathbf{V}_i, i = 1, \dots, N$  des ursprünglichen Polyeders interpoliert.

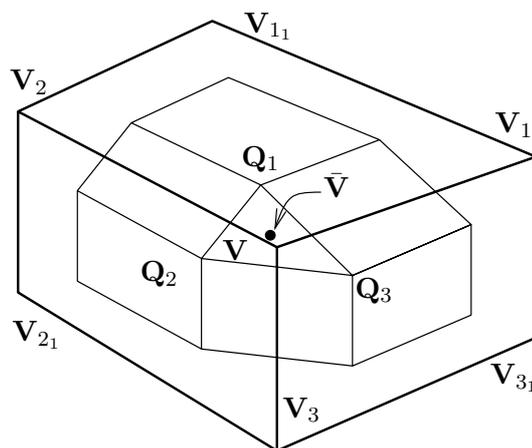


Abbildung 2.10: Bestimmung des Bildpunktes  $\bar{\mathbf{V}}$  der Ecke  $\mathbf{V}$  auf der Splinefläche.

# Kapitel 3

## Ein Ritz-Galerkin-Ansatz für das funktionale Interpolationsproblem

Im vorangegangenen Kapitel wurden verschiedene Verfahren zur Lösung des funktionalen Interpolationsproblems beschrieben (vgl. Abschnitt 2.1). Bis auf das Verfahren von Arge, Dæhlen und Tveito wurde dabei keine explizite Optimierung der Interpolationsfläche durchgeführt.

In diesem Kapitel wird ein neues Verfahren basierend auf einem Ritz-Galerkin-Ansatz vorgestellt (vgl. [Rie95]). Als Interpolationsflächen werden hier TP-B-Splineflächen angesetzt. Dieses Verfahren steht mit der in Abschnitt 2.1.3 beschriebenen Methode von Arge, Dæhlen und Tveito in etwa im gleichen Verhältnis wie der Ritz-Galerkin-Ansatz zu dem Differenzenansatz bei der numerischen Behandlung partieller Differentialgleichungen.

Als Interpolationsfläche wird eine bikubische TP-B-Splinefläche angesetzt. Im Gegensatz zum univariaten Fall, für den eine eindeutige Charakterisierung für die Lösbarkeit des Interpolationsproblems mit B-Splines existiert (der Satz von *Schoenberg-Whitney*, vgl. [dB78]), kann im bivariaten Fall keine eindeutige Aussage gemacht werden.

Die grundsätzliche Idee des im folgenden beschriebenen Verfahrens besteht darin, ein relativ feines Knotengitter für die B-Splinefläche anzusetzen und alle nicht durch die Interpolationsbedingungen festgelegten Freiheitsgrade durch einen Variationsansatz zu bestimmen. Das Verfahren besteht aus den folgenden zwei Schritten:

**Knotenwahl:** In  $u$ - und in  $v$ -Richtung werden *geeignete* äquidistante Knotenvektoren  $U = \{u_{-2}, \dots, u_{n+2}\}$  und  $V = \{v_{-2}, \dots, v_{m+2}\}$  mit konstanten Knotenabständen  $h_u, h_v$  in  $u$  bzw.  $v$ -Richtung gewählt.<sup>1</sup> Die Interpolationsflächen wird als bikubische TP-B-Spline über dem durch  $U$  und  $V$  gegebenen Knotengitter angesetzt. Wir betrachten die Basisfunktionen  $N_{i_1, i_2}^{3,3} = N_I$  immer symmetrisch bzgl. der Knoten  $\mathbf{x}_I$ , d.h. für  $I = (i_1, i_2)$  hat  $N_I$  den Träger  $[u_{i_1-2}, u_{i_1+2}] \times [v_{i_2-2}, v_{i_2+2}]$ .

**Variationsproblem:** Unter allen interpolierenden TP-B-Splineflächen  $S$  über dem Knotengitter  $U \times V$  mit  $S(\mathbf{u}_i) = f_i$ ,  $i = 1, \dots, N$  wird diejenige Fläche  $S_0$  bestimmt, die ein gegebenes Funktional  $\sigma$  minimiert:  $\sigma(S_0) \leq \sigma(S)$ .

---

<sup>1</sup>Die Knoten des Knotengitters werden mit  $\mathbf{x}_I$  für den Doppelindex  $I = (i_1, i_2)$  bezeichnet.

Die Lösung des Variationproblems beruht auf der Betrachtung von *lokalen Interpolationsproblemen*. Anhand der Lösbarkeit von lokalen Problemen kann einerseits auf die Existenz einer globalen Interpolationsfläche geschlossen werden. Andererseits wird die gesuchte Fläche  $S_0$  durch das iterative Lösen der lokalen Interpolationsprobleme bestimmt.

Der hier verfolgte Ansatz hat gegenüber dem Verfahren von Arge, Dæhlen und Tveito vor allem folgende Vorteile:

- Die vorgegebenen Datenwerte  $f_i$ ,  $i = 1, \dots, N$  werden exakt interpoliert.
- Die entstehenden Gleichungssysteme für die Optimierung sind i.a. kleiner.
- Die resultierende Interpolationsfläche ist  $C^2$ -stetig.
- Die Interpolationsfläche liegt in einem CAD-kompatiblen Format vor.

In den folgenden Abschnitten werden die beiden Konstruktionsschritte im Detail beschrieben. Hierbei wird zunächst die Wahl der Knotenvektoren erläutert (Abschnitt 3.1). Der Variationsansatz wird dann anschließend in Abschnitt 3.2 behandelt. Abschließend wird die neue Methode in Abschnitt 3.3 mit einigen der bekannten Methoden aus Abschnitt 2.1 verglichen.

### 3.1 Die Wahl der Knotenvektoren

Zunächst diskutieren wir wie ein *geeignetes*, äquidistantes Knotengitter über dem Parametergebiet definiert werden kann. Hierbei ist vor allem die Lösbarkeit der folgenden lokalen Interpolationsprobleme von Bedeutung:

Betrachtet wird ein  $4 \times 4$  Teilmenge von benachbarten Basisfunktionen  $N_I$  mit Indizes  $\mathcal{G}_I = \{J = I + (k, l) : k, l = 0, \dots, 3\}$ . Sei  $F_I$  die Menge der Indizes der Datenpunkte  $\mathbf{u}_i$  innerhalb des Trägers der  $4 \times 4$  Basisfunktionen

$$\mathcal{R}_I = \bigcup_{J \in \mathcal{G}_I} \text{supp}(N_J). \quad (3.1)$$

Das lokale Interpolationsproblem  $(\mathcal{G}_I, F_I)$  besteht nun darin, eine Fläche  $S_I(u, v) = \sum_{J \in \mathcal{G}_I} b_J N_J(u, v)$  zu finden mit  $S_I(\mathbf{u}_i) = f_i$  für alle  $i \in F_I$ .

Die Lösbarkeit der lokalen Interpolationsprobleme  $(\mathcal{G}_I, F_I)$ ,  $I = (4k, 4l)$  garantiert die Lösbarkeit des globalen Interpolationsproblems, d.h. es gibt eine, i.a. nicht eindeutig bestimmte, TP-B-Splinefläche  $S$  über dem Knotengitter mit  $S(\mathbf{u}_i) = f_i$ ,  $i = 1, \dots, N$  (vgl. Anhang A.1 auf Seite 119).

Daraus ergeben sich folgende Anforderungen an das Knotengitter:

- (i) Das Knotengitter ist so zu wählen, daß alle Datenpunkte  $\mathbf{u}_i$  im Inneren  $\Omega_0$ , d.h. in dem rechteckigen Bereich  $[u_1, u_{n-1}] \times [v_1, v_{m-1}]$  liegen.<sup>2</sup>
- (ii) Hinsichtlich des oben beschriebenen lokalen Interpolationsproblems muß gelten (vgl. Abbildung: 3.1):
- Die Zahl der B-Spline Basisfunktionen in  $u$  bzw.  $v$  Richtung, also  $n + 1$  und  $m + 1$ , muß durch 4 teilbar sein.
  - Die Anzahl der Datenpunkte  $\mathbf{u}_i$  in  $\mathcal{R}_I$  soll ein vorgegebenes Maximum nicht überschreiten. Dies ist notwendig um lokal eine ausreichende Anzahl von Freiheitsgraden für die Optimierung der Interpolationsfläche zu haben. In der Praxis hat sich hier die Einschränkung auf maximal sieben Datenpunkte als geeignet erwiesen.
  - Die lokalen Interpolationsprobleme  $(\mathcal{G}_I, F_I)$ ,  $I = (4k, 4l)$ ,  $k = 0, \dots, \frac{n+1}{4} - 1$ ,  $l = 0, \dots, \frac{m+1}{4} - 1$  müssen lösbar sein.

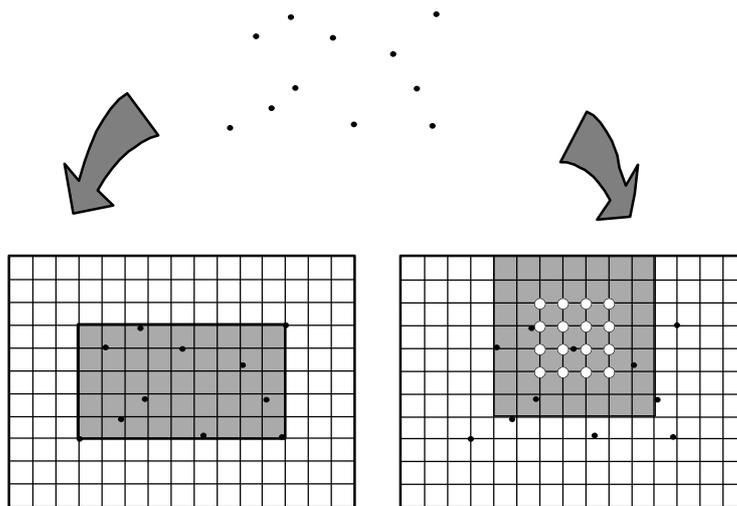


Abbildung 3.1: Anforderungen an das Knotengitter: Alle Datenpunkte  $\mathbf{u}_i$  liegen im Inneren  $\Omega_0$ , die lokalen Interpolationsprobleme  $(\mathcal{G}_I, F_I)$  sind lösbar und es liegen nicht mehr als sieben Datenpunkte in den Regionen  $\mathcal{R}_I$ .

Ein geeignetes Knotengitter ist nun dadurch ausgezeichnet, daß es diese Anforderungen erfüllt.

Zur Überprüfung der Lösbarkeit der lokalen Interpolationsprobleme  $(\mathcal{G}_I, F_I)$ ,  $I = (4k, 4l)$  wird einfach das aus den Interpolationsbedingungen  $\sum_{J \in \mathcal{G}_I} b_J N_J(\mathbf{u}_i) = f_i$ ,  $i \in F_I$  hervorgehende Gleichungssystem

<sup>2</sup>Über diesem Bereich bilden die Funktionen  $N_I$  eine Basis des Raumes der stückweise bikubischen,  $C^2$ -stetigen Polynome.

$$P_I \begin{pmatrix} b_I \\ \vdots \\ b_{I+(3,3)} \end{pmatrix} = \begin{pmatrix} N_I(\mathbf{u}_{i_1}) & \cdots & N_{I+(3,3)}(\mathbf{u}_{i_1}) \\ \vdots & \ddots & \vdots \\ N_I(\mathbf{u}_{i_{d_I}}) & \cdots & N_{I+(3,3)}(\mathbf{u}_{i_{d_I}}) \end{pmatrix} \begin{pmatrix} b_I \\ \vdots \\ b_{I+(3,3)} \end{pmatrix} = \begin{pmatrix} f_{i_1} \\ \vdots \\ f_{i_{d_I}} \end{pmatrix} \quad (3.2)$$

mit  $d_I = \|F_I\| \leq 7$  untersucht. Hat  $P_I$  vollen Rang, so ist das lokale Interpolationsproblem  $(\mathcal{G}_I, F_I)$  lösbar.

Das Knotengitter wird, beginnend mit einer Ausgangsunterteilung die Bedingung (iia) genügt, solange verfeinert, bis auch die Bedingungen (iib) und (iic) erfüllt sind. In der Praxis hat es sich gezeigt, daß Knotengitter mit maximal sieben Datenpunkte  $\mathbf{u}_i$  in den Regionen  $\mathcal{R}_I$ ,  $I = (4k, 4l)$  bereits der Bedingung (iic) hinsichtlich der Lösbarkeit des lokalen Interpolationsproblems genügen.

## 3.2 Der Variationsansatz

In diesem Abschnitt beschreiben wir den Variationsansatz, mit dem die nicht durch die Interpolationsbedingungen bestimmten Freiheitsgrade der TP-B-Splinefläche bestimmt werden.

Als einfaches Funktional wird das Thin-Plate-Funktional  $\sigma^{TP}$  verwendet (vgl. Gleichung (2.1) auf Seite 19). Alternativ wird das im folgenden beschriebene datenabhängige Thin-Plate-Funktional eingesetzt (vgl. [Gre94b, Rie95]).

Betrachtet man das zweidimensionale Vektorfeld  $\vec{\mathbf{t}}(u, v) = (t_1(u, v), t_2(u, v))$ , so ergibt sich das Differential von  $\vec{\mathbf{t}}$  zu

$$D\vec{\mathbf{t}} = \begin{pmatrix} (t_1)_u & (t_1)_v \\ (t_2)_u & (t_2)_v \end{pmatrix}.$$

Für eine skalarwertigen Funktion  $h$  kann das Thin-Plate Funktional mit Hilfe dieses Differentialtails und des Gradienten  $\mathbf{grad}(h)$  folgendermaßen geschrieben werden:<sup>3</sup>

$$\int_{\Omega} \mathbf{tr} (D(\mathbf{grad}(h)))^2 \, dudv = \int_{\Omega} h_{uu}^2 + 2h_{uv}^2 + h_{vv}^2 \, dudv.$$

Im übrigen ist  $\mathbf{Hess}(h) = D(\mathbf{grad}(h))$  gerade die *Hesse-Matrix* der Funktion  $h$ , die sich in diesem Fall als die zu der quadratischen Form

$$\langle D(\mathbf{grad}(h)) \vec{\mathbf{a}} \mid \vec{\mathbf{b}} \rangle$$

bzgl. des gewöhnlichen inneren Produkts gehörige Matrix ergibt.

Zur Definition des Thin-Plate Funktionals für die Funktion  $h$  auf der Fläche  $\mathbf{S}^0$  wird anstelle des gewöhnlichen Gradienten  $\mathbf{grad}$  der Gradient  $\mathbf{grad}_{\mathbf{S}^0}$  bzgl. der Fläche  $\mathbf{S}^0$  verwendet. Für eine konkrete Parametrisierung der Fläche  $\mathbf{S}^0 : \Omega \rightarrow \mathbb{R}^3$ ,  $\Omega \subseteq \mathbb{R}^2$  und eine skalarwertige Funktion  $f : \Omega \rightarrow \mathbb{R}$  wird der Gradient von  $f$  bzgl.  $\mathbf{S}^0$  mit Hilfe der ersten Fundamentalform von  $\mathbf{S}^0$  gebildet (vgl. [Kli78]):  $\mathbf{grad}_{\mathbf{S}^0}(f) = (f_u, f_v)|_{\mathbf{S}^0}^{-1}$ .

<sup>3</sup> $\mathbf{tr}(M)$  ist die Spur von  $M$ .

Das Differential bzw. die Ableitung des Vektorfeldes wird durch den Begriff der *kovarianten Ableitung*  $\nabla_{\mathbf{S}^0}(\vec{\mathbf{t}})$ , wie er für das tangential Vektorfeld  $\vec{\mathbf{t}}$  auf der Fläche  $\mathbf{S}^0$  definiert ist, ersetzt. Die kovariante Ableitung  $\nabla_{\mathbf{S}^0}(\vec{\mathbf{t}})$  eines tangentialen Vektorfeldes  $\vec{\mathbf{t}}$  ist eine lineare Abbildung des Tangentialraumes in sich selbst. Für eine Parametrisierung des tangentialen Vektorfeldes  $\vec{\mathbf{t}}(u, v) = (t_1(u, v), t_2(u, v))$  wird die kovariante Ableitung durch die folgende  $2 \times 2$  Matrix beschrieben (vgl. [Kli78]):

$$\nabla_{\mathbf{S}^0}(\vec{\mathbf{t}}) = \begin{pmatrix} (t_1)_u + t_1\Gamma_{11}^1 + t_2\Gamma_{21}^1 & (t_1)_v + t_1\Gamma_{12}^1 + t_2\Gamma_{22}^1 \\ (t_2)_u + t_1\Gamma_{11}^2 + t_2\Gamma_{21}^2 & (t_2)_v + t_1\Gamma_{12}^2 + t_2\Gamma_{22}^2 \end{pmatrix}$$

wobei  $\Gamma_{ij}^k$  die sogenannten *Christoffel-Symbole zweiter Art* sind. Bezeichnen wir die erste Fundamentalform und ihre Inverse mit  $\mathbf{l} = (g_{ij})_{i,j}$  bzw. mit  $\mathbf{l}^{-1} = (g^{ij})_{i,j}$ , so ergeben sich die Christoffel-Symbole erster und zweiter Art zu

$$\Gamma_{ij1} = \begin{pmatrix} \langle \mathbf{S}_u^0 | \mathbf{S}_{uu}^0 \rangle & \langle \mathbf{S}_u^0 | \mathbf{S}_{uv}^0 \rangle \\ \langle \mathbf{S}_v^0 | \mathbf{S}_{uv}^0 \rangle & \langle \mathbf{S}_v^0 | \mathbf{S}_{vv}^0 \rangle \end{pmatrix} \quad \Gamma_{ij2} = \begin{pmatrix} \langle \mathbf{S}_v^0 | \mathbf{S}_{uu}^0 \rangle & \langle \mathbf{S}_v^0 | \mathbf{S}_{uv}^0 \rangle \\ \langle \mathbf{S}_v^0 | \mathbf{S}_{uv}^0 \rangle & \langle \mathbf{S}_v^0 | \mathbf{S}_{vv}^0 \rangle \end{pmatrix}$$

bzw.

$$\Gamma_{ij}^k = g^{k1}\Gamma_{ij1} + g^{k2}\Gamma_{ij2}.$$

Die entsprechende Form der Hesse-Matrix  $\text{Hess}_{\mathbf{S}^0}$  für die Funktion  $h$  auf  $\mathbf{S}^0$  wird über

$$\langle \nabla_{\mathbf{S}^0}(\text{grad}_{\mathbf{S}^0}(h)) \vec{\mathbf{a}} | \vec{\mathbf{b}} \rangle_{\mathbf{T}_{\mathbf{S}^0}(u,v)(\mathbf{S}^0)}$$

bzgl. der Metrik im Tangentialraum  $\mathbf{T}_{\mathbf{S}^0}(u,v)(\mathbf{S}^0)$  bestimmt. Konkret ergibt sich:

$$\text{Hess}_{\mathbf{S}^0}(h) = \mathbf{l}_{\mathbf{S}^0}^{-1} \left[ \begin{pmatrix} h_{uu} & h_{uv} \\ h_{uv} & h_{vv} \end{pmatrix} - h_u \begin{pmatrix} \Gamma_{11}^1 & \Gamma_{12}^1 \\ \Gamma_{21}^1 & \Gamma_{22}^1 \end{pmatrix} - h_v \begin{pmatrix} \Gamma_{11}^2 & \Gamma_{12}^2 \\ \Gamma_{21}^2 & \Gamma_{22}^2 \end{pmatrix} \right].$$

Als datenabhängiges Thin-Plate Funktional wird für die parametrische Fläche  $\mathbf{S}(u, v) = (S_1(u, v), S_2(u, v), S_3(u, v))^t$  angesetzt:

$$\sigma_{\mathbf{S}^0}^{TP}(\mathbf{S}) = \int_{\Omega} \sum_{i=1}^3 \text{tr}(\text{Hess}_{\mathbf{S}^0}(S_i)^2) d\omega_{\mathbf{S}^0}.$$

Hierbei ist  $d\omega_{\mathbf{S}^0}$  das Flächenelement der Referenzfläche  $\mathbf{S}^0$ .

Ist die Fläche  $\mathbf{S}$  identisch mit der Referenzfläche  $\mathbf{S}^0$ , so gilt für dieses Funktional (vgl. [Rie95]):

$$\sigma_{\mathbf{S}^0}^{TP}(\mathbf{S}) = \int_{\Omega} \kappa_1^2 + \kappa_2^2 d\omega_{\mathbf{S}}.$$

In unserem Fall ist  $\mathbf{S}^0$  selbst funktional, d.h.  $\mathbf{S}^0(u, v) = (u, v, S^0(u, v))^t$ . Integriert wird stets über dem Inneren  $\Omega_0$  (vgl. Bedingung (i) auf Seite 39).

Ziel ist es nun, unter allen TP-B-Splineflächen  $S$  mit  $S(\mathbf{u}_i) = f_i$ ,  $i = 1, \dots, N$  diejenige Fläche  $S_0$  mit minimalem Funktionalwert  $\sigma(S_0) \leq \sigma(S)$ ,  $\sigma \in \{\sigma^{TP}, \sigma_{\mathbf{S}^0}^{TP}\}$  zu finden.

Der Lösungsansatz beruht auf der Lagrange Methode. Da die Knotenvektoren und damit die Gesamtzahl der angesetzten B-Spline Basisfunktionen relativ groß sind, kommt ein direktes Auflösen der entstehenden Matrix nicht in Betracht.

Formal erhält man durch den Lagrange Ansatz folgendes Gleichungssystem:

$$\begin{pmatrix} A & P^t \\ P & 0 \end{pmatrix} \begin{pmatrix} \mathbf{b} \\ \mathbf{d} \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{f} \end{pmatrix}, \quad (3.3)$$

wobei  $\mathbf{b}$  und  $\mathbf{d}$  in Vektorschreibweise alle Kontrollpunkte  $b_I$  und alle Lagrangeschen Multiplikatoren beinhalten.  $\mathbf{f}$  ist der entsprechende Vektor der alle Datenwerte  $f_i$  enthält. Die Matrix  $A$  ist die innere Produktmatrix  $A = (\langle N_I | N_J \rangle_\sigma)$  bzgl. des dem Funktional zugeordneten inneren Produkts  $\langle \cdot | \cdot \rangle_\sigma$ . Die Interpolationsbedingungen  $S(\mathbf{u}_i) = f_i$ ,  $i = 1, \dots, N$  werden durch die Matrix  $P$  repräsentiert, d.h. durch  $(P\mathbf{b})_i = f_i$ .

Das Gleichungssystem (3.3) läßt sich nicht mit dem einfachen Gauß-Seidel oder SOR Verfahren lösen, da die Matrix verschwindende Diagonalelemente besitzt. Wir verwenden daher eine Blockversion dieser iterativen Verfahren. Geeigneter Blöcke ergeben sich durch eindeutige Zuordnung der Datenpunkte  $\mathbf{u}_i$  zu den  $4 \times 4$  benachbarten Basisfunktionen mit Indizes  $\mathcal{G}_I$ ,  $I = (k, l)$ . Ordnen wir alle Datenpunkte aus dem halboffenen Rechteck  $[u_k - \frac{h_u}{2}, u_k + \frac{7h_u}{2}] \times [v_l - \frac{h_v}{2}, v_l + \frac{7h_v}{2}]$  den entsprechenden  $4 \times 4$  Basisfunktionen zu, so ergibt sich ein neues lokales Interpolationsproblem:  $(\mathcal{G}_I, \tilde{F}_I)$  mit  $\tilde{F}_I \subseteq F_I$  (vgl. Abbildung 3.2).

Durch diese Anordnung ergibt sich das zu (3.3) äquivalente System:

$$\begin{pmatrix} C_{(0,0)} & * & \cdots & * \\ * & C_{(4,0)} & \cdots & * \\ \vdots & \ddots & & \vdots \\ * & \cdots & & C_{(\frac{n+1}{4}-1, \frac{m+1}{4}-1)} \end{pmatrix} \mathbf{x} = \mathbf{r}, \quad (3.4)$$

wobei  $C_I$  die Matrix des lokalen Optimierungsproblems  $(\mathcal{G}_I, \tilde{F}_I)$ , ist. In Lagrange-Form stellt sich das lokale Interpolationsproblem  $(\mathcal{G}_I, \tilde{F}_I)$  folgendermaßen dar:

$$\underbrace{\begin{pmatrix} A_I & \tilde{P}_I^t \\ \tilde{P}_I & 0 \end{pmatrix}}_{=C_I} \begin{pmatrix} \mathbf{b}_I \\ \lambda \end{pmatrix} = \begin{pmatrix} * \\ \mathbf{f}_I \end{pmatrix}. \quad (3.5)$$

Die Ausdrücke  $A_I$ ,  $\tilde{P}_I$ ,  $\mathbf{b}_I$  und  $\mathbf{f}_I$  sind dieselben Größen wie in Gleichung (3.3) angewendet auf das lokale Problem  $(\mathcal{G}_I, \tilde{F}_I)$ . Die hier vorgenommene Zuordnung der Datenpunkte garantiert die Lösbarkeit der lokalen Interpolationsprobleme.<sup>4</sup>

Damit ist das Lösen des lokalen Optimierungsproblems  $(\mathcal{G}_I, \tilde{F}_I)$  in Gleichung (3.5) äquivalent mit der Durchführung eines Block-Gauß-Seidel oder Block-SOR Teilschrittes bzgl. des ursprünglichen Problems (3.3).

<sup>4</sup>Die Lösbarkeit des Interpolationsproblems ist äquivalent damit, daß  $\tilde{P}_I$  vollen Rang hat. Da aber schon  $P_I$  vollen Rang hat und  $\tilde{P}_I$  aus  $P_I$  durch Streichen gewisser Zeilen hervorgeht, hat  $\tilde{P}_I$  immer vollen Rang.

Alternativ können auch die lokalen Interpolationsprobleme  $(\mathcal{G}_I, F_I)$  wie in Bedingung (ii) angesetzt werden. Auf das Konvergenzverhalten des Algorithmus hat dies in der Praxis keine Auswirkung. In allen Fällen wird über dem Träger  $\mathcal{R}_I$  integriert (vgl. Gleichung (3.1)).

Der Algorithmus zur Lösung des Variationsproblems betrachtet nacheinander alle lokalen Interpolationsprobleme  $(\mathcal{G}_I, \tilde{F}_I)$ ,  $I = (4k, 4l)$  bzw.  $(\mathcal{G}_I, F_I)$ ,  $I = (4k, 4l)$ .

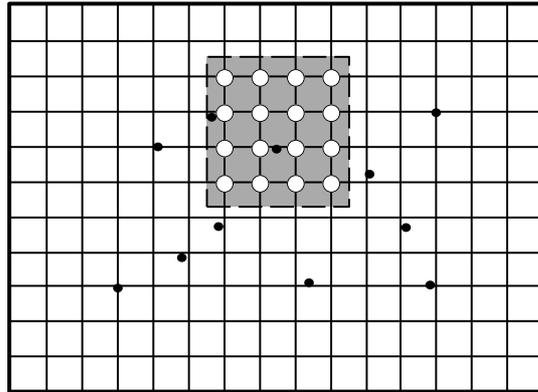


Abbildung 3.2: Die Zuordnung der Datenpunkte durch Umordnung des globalen Gleichungssystems.

Wird das datenabhängige Thin-Plate Funktional zur Optimierung verwendet, so muß eine geeignete Referenzfläche gewählt werden. Dies geschieht über eine least-squares Approximation an die Datenwerte  $f_i$ . Die Approximationsfläche wird über einem größeren Knotengitter mit Gitterabstand  $\tilde{h}_u, \tilde{h}_v$  definiert.  $\tilde{h}_u$  bzw.  $\tilde{h}_v$  wird als Vielfaches von  $h_u$  bzw.  $h_v$  gewählt, wobei die entsprechende Anzahl der Kontrollpunkte wesentlich kleiner als die Zahl der Datenpunkte sein muß.

### 3.3 Vergleich der funktionalen Interpolationsmethoden

In Abschnitt 2.1 wurden verschiedene bekannte Verfahren zur Lösung des funktionalen Interpolationsproblems beschrieben. Des weiteren wurde in den diesem Kapitel eine neue Methode basierend auf einem Ritz-Galerkin Ansatz vorgestellt. Dieses neue Verfahren wird im folgenden mit einigen bekannten Verfahren verglichen.

Die Vergleiche beruhen auf der bei der Interpolation verteilter Daten im funktionalen Fall weit verbreiteten Vorgehensweise: Von einer bekannten Funktion  $S$ , der sogenannten *Testfunktion*, werden für zufällig verteilten Datenpunkten  $\mathbf{u}_i$  die Funktionswerte  $S(\mathbf{u}_i)$  ermittelt. Die verschiedenen Rekonstruktionsverfahren werden anschließend auf den so generierten Satz von Interpolationsdaten angewendet.

Als Gütekriterium für die einzelnen Interpolationsverfahren werden neben dem visuellen Vergleich zwischen Testfunktion und Interpolationsfläche der erzeugten Interpolationsfläche vor allem *Krümmungsplots* verwendet. Hierbei wird meist die *Gaußkrümmung* oder die *mittlere Krümmung* durch Abbildung der Krümmungswerte auf eine festgelegt Farbskala von Rot nach Blau visualisiert. In der Praxis zeigt sich, daß ungewünschte, lokale Deformationen in einer Fläche, die mit bloßem Auge nicht mehr sichtbar sind, mit Hilfe der Krümmungsvisualisierung erkannt werden können. Zur Konvertierung der Krümmungswerte auf die Farbskala wird eine lineare Abbildung eines vorgegebenen Intervalls von Krümmungswerten verwendet. Die Farbskala wird zur Orientierung in den nachfolgenden

Abbildungen mit dargestellt.

Als Testfunktionen werden die Funktionen von Lyche und Morken [LM87]:

$$S_1(x, y) = (1 - \frac{x}{2})^6 (1 - \frac{y}{2})^6 + 1000(1 - x)^3 x^3 (1 - y)^3 y^3,$$

von Franke [Fra82] :

$$S_2(x, y) = \frac{1}{2} \exp\left(-\frac{(9x-2)^2 + (9y-2)^2}{4}\right) + \frac{3}{4} \exp\left(-\frac{(9x+1)^2}{49} + \frac{9y+1}{10}\right) \\ + \frac{1}{2} \exp\left(-\frac{(9x-7)^2 + (9y-3)^2}{4}\right) + \frac{1}{5} \exp\left(-\frac{(9x-4)^2}{5} - \frac{(9y-7)^2}{5}\right)$$

und von Ritchie [Rit78]:

$$S_3(x, y) = \begin{cases} 1 & \text{falls } y - 2.1x + 0.1 \geq \frac{1}{2} \\ 2(y - 2.1x + 0.1) & \text{falls } 0 \leq y - 2.1x + 0.1 < \frac{1}{2} \\ \frac{\cos(4\pi r(x,y)) + 1}{2} & \text{falls } r(x, y) \leq \frac{1}{4} \\ 0 & \text{sonst,} \end{cases}$$

mit  $r(x, y) = \sqrt{(2.1x - 1.6)^2 + (y - 0.5)^2}$  eingesetzt. Abbildung 3.3 zeigt die Testfunktionen mit ihren zugehörigen Krümmungsplots, Gaußkrümmung für  $S_1$  und  $S_2$ , mittlere Krümmung für  $S_3$ , über dem Parametergebiet  $[0, 1]^2$ . Für jede Funktion wird jeweils eine feste Datenpunktmenge zur Interpolation benutzt. Die Datenpunktmenge zu  $S_1$  hat 75, die Datenpunktmenge zu  $S_2$  100 und diejenige zu  $S_3$  200 Punkte.

Da beim Verfahren nach Arge, Dæhlen und Tveito die Interpolationsfläche durch bilineare Interpolation der Gitterwerte definiert wird, besitzt die Interpolationsfläche *keine* elliptischen Punkte. Daher werden für diese Methode keine Krümmungsvisualisierung angegeben.

Von den klassischen Verfahren zur Lösung des funktionalen Interpolationsproblems liefern die radialen Basismethoden i.a. Interpolationsflächen von sehr guter Qualität. Verglichen werden die Ergebnisse deshalb mit den Thin-Plate Splines nach Duchon. Zusätzlich beziehen wir das Verfahren nach Arge, Dæhlen und Tveito in den Vergleich mit ein. Der neue Ansatz wird sowohl mit dem einfachen, wie mit dem datenabhängigen Thin-Plate Funktional eingesetzt.

Das Interpolationsfläche nach Arge, Dæhlen und Tveito hat große Probleme in den Randregionen. Diese Probleme beruhen wahrscheinlich auf den fehlenden Datenpunkten in einigen Randbereichen (vgl. Abbildungen 3.4 – 3.6). Die Thin-Plate Splines erzeugen in allen Fällen Interpolationsflächen von guter Qualität. Einzig bei der Ritchie-Fläche zeigen sich einige Probleme nahe den  $C^0$ -Übergängen der Originalfläche.

Der Ritz-Galerkin Ansatz mit dem einfachen Thin-Plate Funktional liefert für die Lyche/Morken Fläche sehr gute Resultate (vgl. Abbildung 3.4). Für die Franke- und für die Ritchie-Fläche zeigen sich jedoch noch einige Probleme, die vor allem im Randbereich auftreten. Wird das datenabhängige Funktional angesetzt, so ergibt sich eine deutlich verbesserte Interpolationsfläche (vgl. Abbildung 3.5 – 3.6). Dies gilt insbesondere für die sehr schwer zu handhabende Ritchie-Fläche. Allerdings schlägt hierbei der erhöhte Rechenaufwand negativ zu Buche (vgl. Tabelle B.2 und B.3).

Das genaue Laufzeitverhalten sowie die verwendeten Parameter sind im Anhang B in den Tabellen B.1 – B.3 aufgelistet.

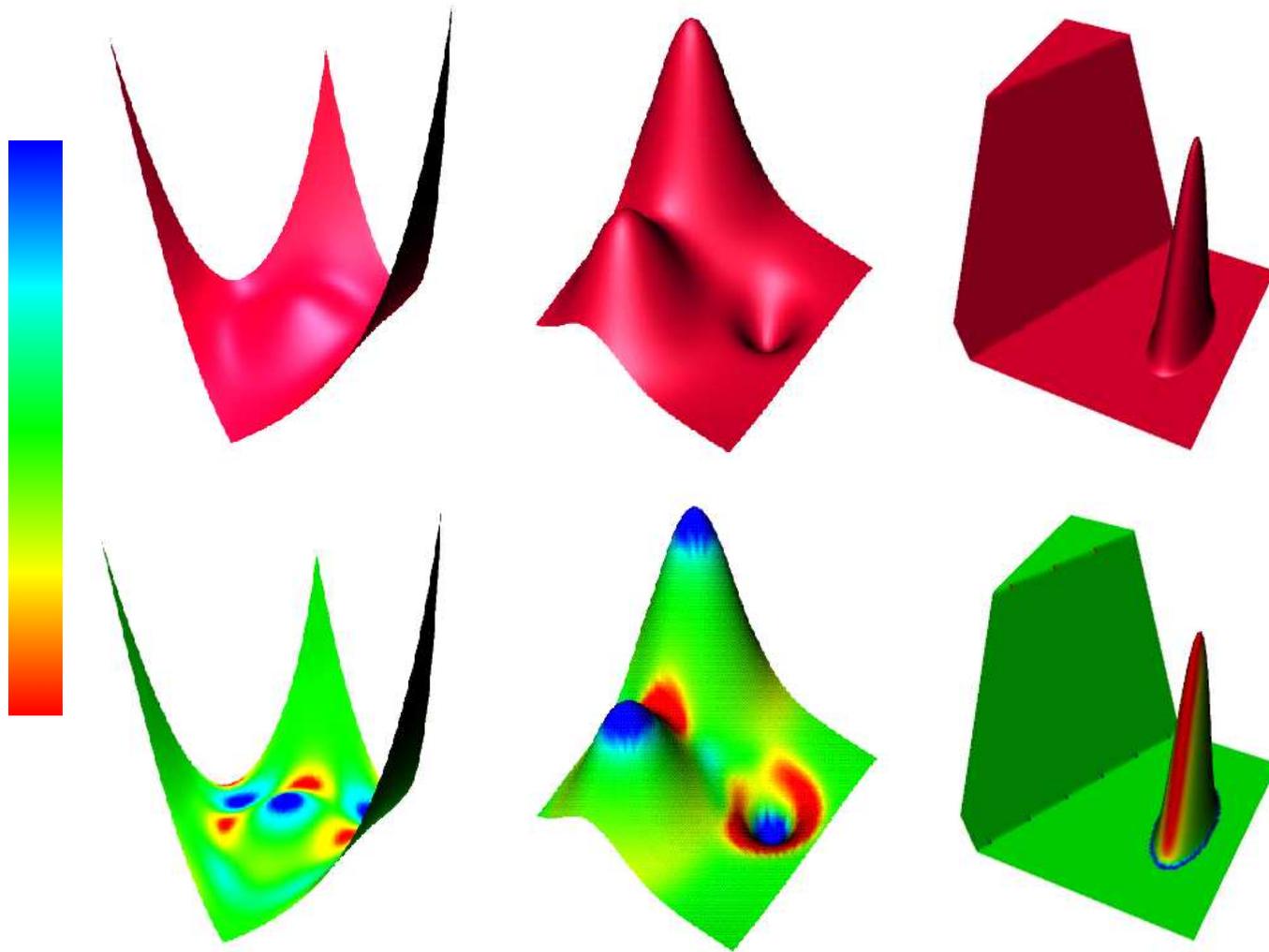


Abbildung 3.3: Die Testfunktionen (oben) nach Lyche/Morken (links), Franke (Mitte) und Ritchie (rechts) mit den zugehörigen Krümmungsplots (unten): Gaußkrümmung für die Lyche- und die Franke-Fläche, mittlere Krümmung für die Ritchie-Fläche.



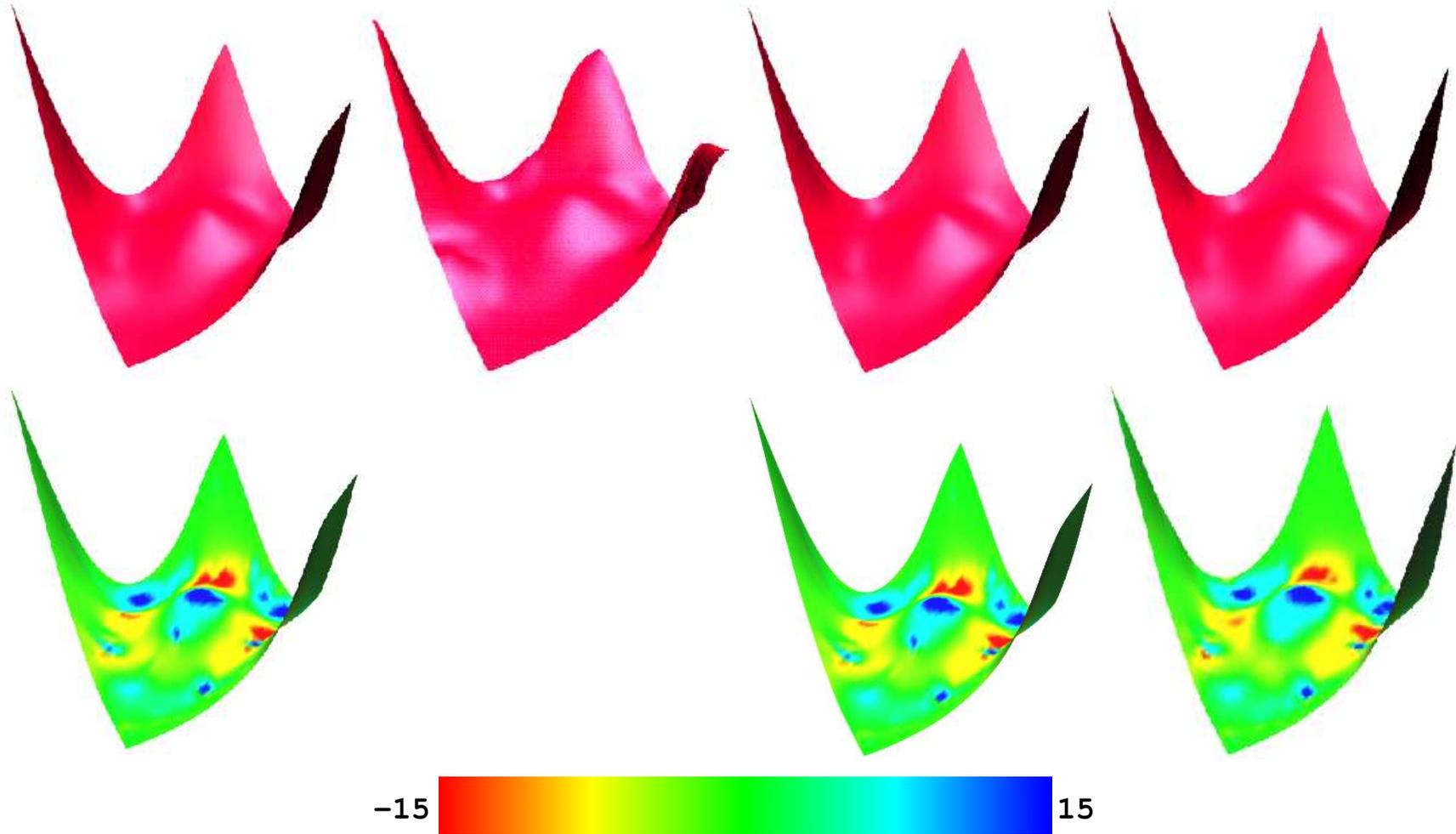


Abbildung 3.4: Die Lyche-Fläche mit Graußkrümmung: Der Thin-Plate Spline Interpolant, die Fläche nach Arge, Dæhlen und Tveito, die Interpolationsfläche mit dem Ritz-Galerkin-Ansatz mit einfachem und datenabhängigem Thin-Plate Funktional (von links nach rechts).



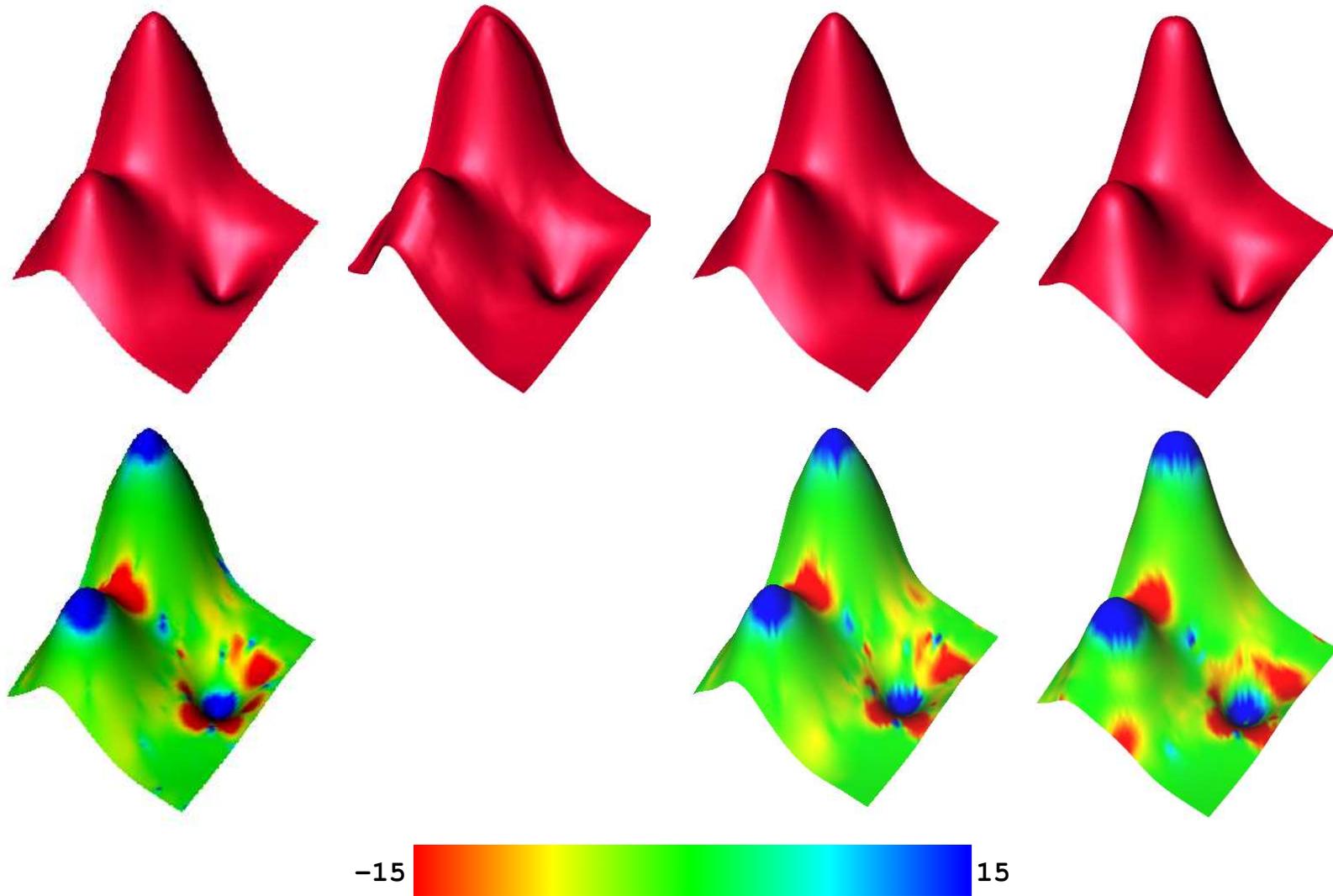


Abbildung 3.5: Die Franke-Fläche mit Gaußkrümmung: Der Thin-Plate Spline Interpolant, die Fläche nach Arge, Dæhlen und Tveito, die Interpolationsfläche mit dem Ritz-Galerkin-Ansatz mit einfachem und datenabhängigem Thin-Plate Funktional (von links nach rechts).



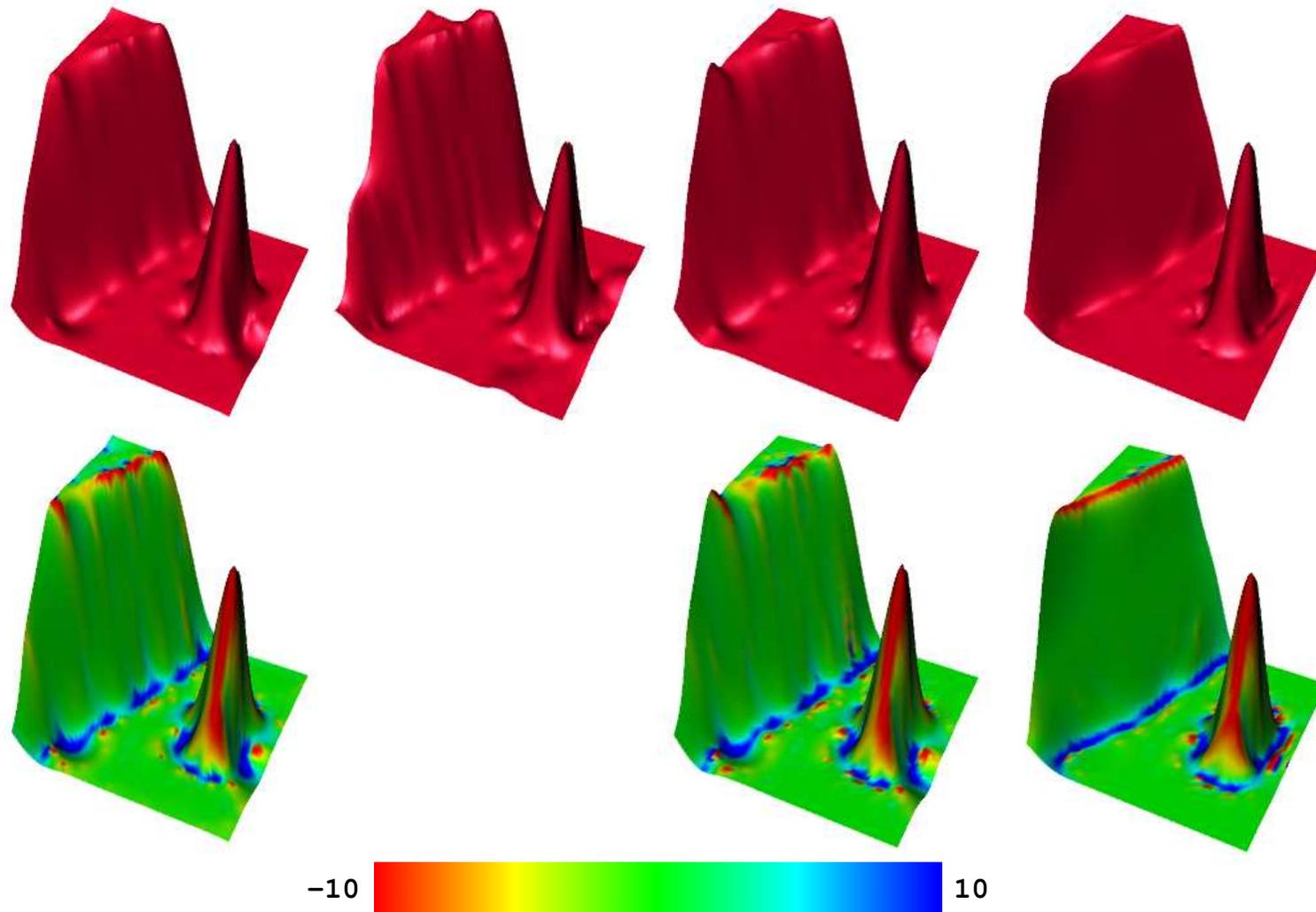


Abbildung 3.6: Die Ritchie-Fläche mit Krümmungsplot für die mittlererKrümmung: Der Thin-Plate Spline Interpolant, die Fläche nach Arge, Dæhlen und Tveito, die Interpolationsfläche mit dem Ritz-Galerkin-Ansatz mit einfachem und datenabhängigem Thin-Plate Funktional (von links nach rechts).



# Kapitel 4

## Parametrische Interpolation basierend auf $G^1$ -Kurvennetzen

In diesem Kapitel wird ein neues Verfahren zum Lösen des parametrischen Interpolationsproblems vorgestellt. Dieses Verfahren basiert auf Methoden zur Generierung eines kubischen  $G^1$ -Kurvennetzes und von Splitflächen bestehend aus drei Bézier-Dreiecksflächen vom Grad 4. Im weiteren wird dieses Verfahren  $G^1$ -*Kurvennetzmethode* genannt.

Zunächst werden in Abschnitt 4.1 die Repräsentation des Kurvennetzes, sowie die Bestimmung der sogenannten *lokalen Parameter* und die Optimierung des Kurvennetzes beschrieben. In Abschnitt 4.2 wird dann eine Erweiterung des Verfahrens von Shirman und Séquin (vgl. Abschnitt 2.2.1 auf Seite 26) zum Einfügen von  $3 \times 4$  Splitflächen in das Kurvennetz angegeben. Dieses Verfahren verwendet einen Variationsansatz zur Bestimmung gewisser Flächenparameter. In Abschnitt 4.3 wird eine Alternative zu dem Zweischrittverfahren (getrennte Optimierung des Kurvennetzes und der  $3 \times 4$  Splitflächen) vorgestellt, eine Methode zur simultanen Optimierung aller für die Variation eingesetzten Kurven- und Flächenparameter. Zudem werden in diesem Abschnitt die verschiedenen Funktionale und Optimierungsmethoden miteinander verglichen.

### 4.1 Das kubische $G^1$ -Kurvennetz

Die Repräsentation des kubischen Kurvennetzes beruht auf der Wahl von *lokalen Koordinaten* in den Eckpunkten des durch die Interpolationsaufgabe gegebenen Polyeders (Abschnitt 4.1.1). Die für diese Repräsentation zu bestimmenden *lokalen Parameter* des Kurvennetzes werden mit den in Abschnitt 4.1.2 angegebenen Methoden berechnet. In Abschnitt 4.1.3 werden die verschiedenen Variationsansätze erläutert mit denen das  $G^1$ -Kurvennetz optimiert wird. Abschließend werden in Abschnitt 4.1.4 die vorgestellten Methoden zur Bestimmung der lokalen Parameter und die Funktionale zur Optimierung des Kurvennetzes miteinander verglichen und bewertet.

### 4.1.1 Repräsentation des $G^1$ -Kurvennetzes

Um die Wahl der lokalen Koordinaten und die Repräsentation des Kurvennetzes zu motivieren, betrachten wir zunächst eine funktionale Fläche als Spezialfall einer parametrischen Fläche. Für eine gegebene funktionale Fläche  $S : \mathbb{R}^2 \rightarrow \mathbb{R}$  erhalten wir ihre parametrische Darstellung durch:

$$\mathbf{S}(u, v) = \begin{pmatrix} u \\ v \\ S(u, v) \end{pmatrix}. \quad (4.1)$$

Die folgenden Eigenschaften der Fläche  $\mathbf{S}$  lassen sich leicht nachprüfen:

$$\mathbf{S}_u = (1, 0, S_u)^t = \hat{\mathbf{u}} + S_u \hat{\mathbf{w}}, \quad (4.2)$$

$$\mathbf{S}_v = (0, 1, S_v)^t = \hat{\mathbf{v}} + S_v \hat{\mathbf{w}}, \quad (4.3)$$

$$\hat{\mathbf{n}}_s = \frac{(-S_u, -S_v, 1)^t}{\|(-S_u, -S_v, 1)^t\|} = \frac{\hat{\mathbf{w}} - (S_u \hat{\mathbf{u}} + S_v \hat{\mathbf{v}})}{\sqrt{1 + S_u^2 + S_v^2}}, \quad (4.4)$$

wobei  $\hat{\mathbf{u}} = (1, 0, 0)$ ,  $\hat{\mathbf{v}} = (0, 1, 0)$  und  $\hat{\mathbf{w}} = (0, 0, 1)$  zunächst die kanonischen Vektoren des  $\mathbb{R}^3$  darstellen. Für die Ableitung von  $\mathbf{S}$  in eine beliebige, nicht notwendigerweise normierte Parameterrichtung  $\vec{\mathbf{t}} = (a, b) = a\hat{\mathbf{u}} + b\hat{\mathbf{v}}$  ergibt sich damit (vgl. Abbildung 4.1):

$$\mathbf{S}_{\vec{\mathbf{t}}} = \vec{\mathbf{t}} + (aS_u + bS_v)\hat{\mathbf{w}}. \quad (4.5)$$

Wir stellen das Kurvennetz in den Ecken lokal als funktionales Kurvennetz dar. Hierzu werden zunächst in jeder Ecke  $\mathbf{V}_i$  des Polyeders Vektoren  $\hat{\mathbf{w}}_i$  bestimmt. Die  $\hat{\mathbf{w}}_i$ 's werden mit Hilfe der in Abschnitt 2.2.1 auf Seite 26 beschriebenen Verfahren zur Abschätzung von Flächennormalen ermittelt. In der Ebene normal zu  $\hat{\mathbf{w}}_i$  werden zwei orthonormale Vektoren  $\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i$  derart bestimmt, daß  $\{\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i, \hat{\mathbf{w}}_i\}$  eine positiv orientierte Orthonormalbasis des  $\mathbb{R}^3$  bildet. Die genaue Wahl der Vektoren  $\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i$  hat keinen Einfluß auf die im folgenden beschriebenen Algorithmen. Das  $G^1$ -Kurvennetz wird lokal mit Hilfe der oben eingeführten lokalen Koordinaten in den Ecken  $\mathbf{V}_i$  beschrieben.

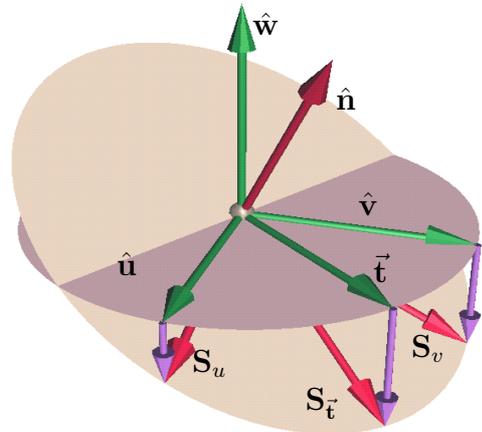


Abbildung 4.1: Lokale Koordinaten.

Die Formeln (4.2), (4.3) und (4.5) zeigen, daß für eine bekannte Parameterrichtung  $\vec{\mathbf{t}}$  der resultierende Tangentialvektor  $\mathbf{S}_{\vec{\mathbf{t}}}$  einer Interpolationsfläche (und damit auch einer Kurve des Kurvennetzes, die diesen Tangentialvektor interpoliert) linear von den partiellen Ableitungen  $S_{u_i}$  und  $S_{v_i}$  der unbekannt funktionalen Fläche  $S_i$  abhängt. Diese Eigenschaft ist aufgrund unserer direkten Übertragung der funktionalen Darstellung völlig analog zu der entsprechenden Eigenschaft der funktionalen Kurvennetze (vgl. Abschnitt 2.1.2).

Zur Definition des  $G^1$ -Kurvennetzes wird für jede Kante  $\overline{\mathbf{V}_i \mathbf{V}_j}$  des Polyeders eine kubische Kurve  $\mathbf{C}_{ij}$  in Form eines Hermite-Interpolanten angesetzt:

$$\begin{aligned}
\mathbf{C}_{ij}(u) &= \mathbf{V}_i H_0^3(u) + \mathbf{S}_{\vec{\mathbf{t}}_{ij}} H_1^3(u) + \mathbf{S}_{\vec{\mathbf{t}}_{ji}} H_1^3(1-u) + \mathbf{V}_j H_0^3(1-u) \\
&= \mathbf{V}_i H_0^3(u) + \left( \vec{\mathbf{t}}_{ij} + (a_{ij} S_{ui} + b_{ij} S_{vi}) \hat{\mathbf{w}}_i \right) H_1^3(u) + \\
&\quad \left( \vec{\mathbf{t}}_{ji} + (a_{ji} S_{uj} + b_{ji} S_{vj}) \hat{\mathbf{w}}_j \right) H_1^3(1-u) + \mathbf{V}_j H_0^3(1-u).
\end{aligned} \tag{4.6}$$

Hierbei sind  $H_0^3(u) = 2u^3 - 3u^2 + 1$  und  $H_1^3(u) = u^3 - 2u^2 + u$  die entsprechenden kubische Hermite-Polynome. Die Vektoren  $\vec{\mathbf{t}}_{ij}$  werden als *lokale Parameterrichtungen*, die skalaren Größen  $a_{ij}, b_{ij}$  als *lokale Parameter* für die Kurve  $\mathbf{C}_{ij}$  bzgl. des lokalen Koordinatensystems  $\{\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i, \hat{\mathbf{w}}_i\}$  bezeichnet.

Wir verwenden hier eine symmetrische Schreibweise bzgl. der Endtangente von  $\mathbf{C}_{ij}$ , d.h.  $\mathbf{C}'_{ij}(1) = -\mathbf{S}_{\vec{\mathbf{t}}_{ji}}$ . Diese Schreibweise bietet den Vorteil, daß die eigentliche Orientierung der Kurve  $\mathbf{C}_{ij}$  bei Bestimmung der Größen  $\vec{\mathbf{t}}_{ij}, \vec{\mathbf{t}}_{ji}$  keine Rolle spielt.

Für gegebene lokale Parameter  $a_{ij}, b_{ij}$  und partielle Ableitungen  $S_{ui}, S_{vi}$  in den Ecken  $\mathbf{V}_i$  ist das  $G^1$ -Kurvennetz bestehend aus den Kurven  $\mathbf{C}_{ij}$  eindeutig bestimmt. Im weiteren bezeichnen wir dieses Kurvennetz mit  $\mathbf{N}$ .

### 4.1.2 Bestimmung der lokalen Parameter

Die verschiedenen Methoden zur Bestimmung der lokalen Parameter des  $G^1$ -Kurvennetzes werden im folgenden angegeben. Dabei gehen wir in zwei Schritten vor: Zunächst wird die Richtung  $\hat{\mathbf{t}}_{ij}$  des Vektors  $\vec{\mathbf{t}}_{ij}$  und anschließend dessen Länge festgelegt.

Zur Bestimmung der Richtung des Parametervektors  $\vec{\mathbf{t}}_{ij}$  stehen die folgenden zwei Möglichkeiten zur Verfügung:

**Projektion der Kanten (PK):** Die Polyederkante  $\overline{\mathbf{V}_i \mathbf{V}_j}$  wird auf die lokale Parameterebene bei  $\mathbf{V}_i$  projiziert. Diese Methode wird von Shirman und Séquin zur Definition der Tangentenrichtungen des Kurvennetzes verwendet (vgl. Abschnitt 2.2.1 und [SS87]).

**Ebene Kurve (EK):** DeRose und Mann verwenden diese Methode zur Definition einer ebenen Kurve ihres kubischen  $G^1$ -Kurvennetzes (vgl. Abschnitt 2.2.1). Dabei wird eine Ebene  $\mathbf{E}$ , die die Ecken  $\mathbf{V}_i$  und  $\mathbf{V}_j$  sowie den gemittelten Vektor  $\hat{\mathbf{w}}_i + \hat{\mathbf{w}}_j$  enthält, definiert. Die gesuchte Richtung ergibt sich aus dem Schnitt von  $\mathbf{E}$  mit der lokalen  $(\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i)$ -Parameterebene.

Die Länge des Parametervektors  $\vec{\mathbf{t}}_{ij}$  kann mit einer der folgenden Methoden ermittelt werden:

**Kantenlänge (KL):** Es wird einfach  $\|\vec{\mathbf{t}}_{ij}\| = \|\overline{\mathbf{V}_i \mathbf{V}_j}\|$  gesetzt (vgl. das Verfahren von Shirman und Séquin in Abschnitt 2.2.1).

**Kreissegment (KS):** Die Positionen  $\mathbf{V}_i$  und  $\mathbf{V}_j$  definieren zusammen mit der tangentialen Richtung  $\hat{\mathbf{t}}_{ij}$  ein eindeutig bestimmtes Kreissegment  $s$  (vgl. Abbildung 4.2).  $\|\vec{\mathbf{t}}_{ij}\|$  wird der Bogenlänge von  $s$  gleichgesetzt.

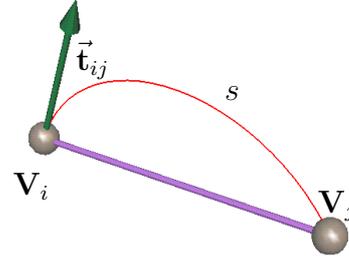


Abbildung 4.2: Die Kreissegment Methode zur Bestimmung der Tangentenlänge.

**Vertauschtes Kreissegment (vKS):** Hier wird wie bei der KS-Methode mittels zweier Positionen und einer Richtung die Länge eines Kreissegmente berechnet. Im Unterschied werden jedoch  $\mathbf{V}_i$ ,  $\mathbf{V}_j$  und die Richtung  $\hat{\mathbf{t}}_{ji}$  zur Fixierung der Länge von  $\vec{\mathbf{t}}_{ij}$  herangezogen.

**De Boor-Höllig-Sabin Kurve (dBHS):** Die von DeRose und Mann verwendeten geometrischen Hermite-Kurven nach de Boor, Höllig und Sabin bestimmen die Länge der Tangenten in den Endpunkten durch Interpolation von Krümmungsinformation (vgl. Abschnitt 2.2.1). In unserer Situation muß die fehlende Krümmungsinformation abgeschätzt werden. Dazu wird die Methode von Moreton und Séquin (vgl. Abbildung 2.8 und [MS92]) benutzt. Wir übernehmen die so bestimmten Tangentenlängen als Länge der lokalen Parameterrichtung  $\vec{\mathbf{t}}_{ij}$ .

### 4.1.3 Optimierung des $G^1$ -Kurvennetzes

In diesem Abschnitt werden die Methoden zur Optimierung des oben beschriebenen kubischen  $G^1$ -Kurvennetzes diskutiert. Hierbei wird zwischen zwei verwendeten Funktionaltypen unterschieden: den *einfachen Funktionalen* und den *datenabhängigen Funktionalen* (vgl. Abschnitt 1.3).

Sei  $E$  die Menge der Indexpaare  $ij$ , die die Kanten  $\overline{\mathbf{V}_i \mathbf{V}_j}$  des Polyeders beschreiben. Zur Definition des einfachen Funktionals wird jede Kurve  $\mathbf{C}_{ij}$ ,  $ij \in E$  über dem Intervall  $[0, \|\overline{\mathbf{V}_i \mathbf{V}_j}\|]$  parametrisiert:  $\tilde{\mathbf{C}}_{ij}(s) = \mathbf{C}_{ij}\left(\frac{s}{\|\overline{\mathbf{V}_i \mathbf{V}_j}\|}\right)$ .

Das einfache Funktional ergibt sich dann zu:

$$\begin{aligned} \sigma^S(\mathbf{N}) &= \sum_{ij \in E} \int_{[0, \|\overline{\mathbf{V}_i \mathbf{V}_j}\|]} \left\| \tilde{\mathbf{C}}_{ij}'''(s) \right\|^2 + \alpha \left\| \tilde{\mathbf{C}}_{ij}''(s) \right\|^2 + \beta \left\| \tilde{\mathbf{C}}_{ij}'(s) \right\|^2 ds, \quad \alpha, \beta \geq 0 \quad (4.7) \\ &= \sum_{ij \in E} \int_0^1 \frac{\left\| \mathbf{C}_{ij}'''(u) \right\|^2}{\|\overline{\mathbf{V}_i \mathbf{V}_j}\|^5} + \alpha \frac{\left\| \mathbf{C}_{ij}''(u) \right\|^2}{\|\overline{\mathbf{V}_i \mathbf{V}_j}\|^3} + \beta \frac{\left\| \mathbf{C}_{ij}'(u) \right\|^2}{\|\overline{\mathbf{V}_i \mathbf{V}_j}\|} du, \quad \alpha, \beta \geq 0. \end{aligned}$$

Funktionale dieser Form werden häufig zur Optimierung von Kurven verwendet. Die Parametrisierung über dem Intervall  $[0, \|\bar{\mathbf{V}}_i \bar{\mathbf{V}}_j\|]$  dient der Gewichtung der Beiträge der einzelnen Kurven  $\mathbf{C}_{ij}$  entsprechend ihrer Länge.

Da für die verwendeten Hermite-Polynome das Integral in Gleichung (4.7) symbolisch ausgewertet werden kann, kann auch die aus der Variation des  $G^1$ -Kurvennetzes bzgl. der partiellen Ableitungen  $S_{ui}, S_{vi}$ ,  $i = 1, \dots, N$  entstehende Matrix explizit aufgestellt werden.

Alternativ werden datenabhängige Funktionale basierend auf dem Laplace-Beltrami-Operator oder Operatoren höherer Ordnung angesetzt (vgl. Abschnitt 1.3 und [Gre94b, Gre94a, Gre94c]). Diese datenabhängigen Funktional bieten den Vorteil, daß sie eine gute Approximation an die exakten Funktionale zur Bewertung der Krümmung bzw. der Krümmungsvariation einer Kurve darstellen. Die Güte der Approximation hängt dabei allerdings davon ab, wie „weit“ die Kurve  $\mathbf{C}$ , die mit dem Funktional bewertet werden soll, von der sogenannten *Referenzkurve*  $\mathbf{C}^0$  entfernt liegt (vgl. [Gre94b]).

Im Falle des Laplace-Beltrami-Operators für die Kurve  $\mathbf{C}$  bzgl. der Referenzkurve  $\mathbf{C}^0$  (vgl. Abschnitt 1.3)

$$\Delta_{\mathbf{C}^0}(\mathbf{C}) = \operatorname{div}_{\mathbf{C}^0}(\operatorname{grad}_{\mathbf{C}^0}(\mathbf{C})) = \frac{\mathbf{C}''}{\langle\langle \mathbf{C}^0 \rangle' | \langle \mathbf{C}^0 \rangle' \rangle} - \frac{\langle\langle \mathbf{C}^0 \rangle' | \langle \mathbf{C}^0 \rangle'' \rangle \mathbf{C}'}{\langle\langle \mathbf{C}^0 \rangle' | \langle \mathbf{C}^0 \rangle' \rangle^2},$$

ergibt sich das datenabhängige Funktional

$$\sigma^{LB}(\mathbf{N}) = \sum_{ij \in E} \int \left\| \Delta_{\mathbf{C}_{ij}^0}(\mathbf{C}_{ij}(u)) \right\|^2 ds, \quad (4.8)$$

wobei hier nach der Bogenlänge von  $\mathbf{C}_{ij}^0$  integriert wird. Als Referenzkurven  $\mathbf{C}_{ij}^0$  dienen die kubischen Kurven des Kurvennetzes mit denselben lokalen Parametern und mit den partiellen Ableitungen  $S_{ui} = S_{vi} = 0$ ,  $i = 1, \dots, N$ .

Für den Fall  $\mathbf{C}_{ij} \equiv \mathbf{C}_{ij}^0$  kann gezeigt werden, daß  $\Delta_{\mathbf{C}_{ij}^0}(\mathbf{C}_{ij})$  gerade das Quadrat der Krümmung der Kurve beschreibt. Damit gilt für  $\mathbf{C}_{ij} \equiv \mathbf{C}_{ij}^0$ ,  $ij \in E$ :

$$\sigma^{LB}(\mathbf{N}) = \sum_{ij \in E} \int (\kappa_{\mathbf{C}_{ij}})^2 ds.$$

Für das Funktional  $\sigma^{LB}$ , das quadratisch in den unbekanntenen partiellen Ableitungen ist, kann das Integral nicht mehr symbolisch ausgewertet werden. Wir verwenden hier eine einfache numerische Integration mittels der Gauß-Quadratur (vgl. [Sto93]).

Unter Verwendung des Begriffes des Gradienten und der Divergenz für Funktionen auf Kurven kann ein Funktional dritter Ordnung definiert werden. Die entsprechende Form für das Kurvennetz lautet:

$$\sigma^{TO}(\mathbf{N}) = \sum_{ij \in E} \int \langle \operatorname{grad}_{\mathbf{C}_{ij}^0} \left( \Delta_{\mathbf{C}_{ij}^0}(\mathbf{C}_{ij}(u)) \right) | \operatorname{grad}_{\mathbf{C}_{ij}^0} \left( \Delta_{\mathbf{C}_{ij}^0}(\mathbf{C}_{ij}(u)) \right) \rangle_{\mathbf{T}\mathbf{C}_{ij}^0} ds, \quad (4.9)$$

wobei wiederum über die Bogenlänge von  $\mathbf{C}_{ij}^0$  integriert wird.  $\langle \cdot | \cdot \rangle_{\mathbf{TC}_{ij}^0}$  bezeichnet das innere Produkt bzgl. des Tangentialraums  $\mathbf{TC}_{ij}^0$  der Referenzkurve  $\mathbf{C}_{ij}^0$  für den entsprechenden Parameterwert  $u$ :  $\langle \vec{\mathbf{v}}_1 | \vec{\mathbf{v}}_2 \rangle_{\mathbf{TC}_{ij}^0} = \langle \vec{\mathbf{v}}_1 | \vec{\mathbf{v}}_2 \rangle / \langle (\mathbf{C}_{ij}^0)'(u) | (\mathbf{C}_{ij}^0)'(u) \rangle$ . Für den Operator dritter Ordnung gilt im Fall  $\mathbf{C}_{ij} \equiv \mathbf{C}_{ij}^0$ :

$$\text{grad}_{\mathbf{C}_{ij}^0} \left( \Delta_{\mathbf{C}_{ij}^0}(\mathbf{C}_{ij}(u)) \right) = \frac{d \vec{\mathbf{k}}_{\mathbf{C}_{ij}}}{ds},$$

womit  $\sigma^{TO}(\mathbf{C}_{ij})$  eine gute Approximation an die Variation des Krümmungsvektors von  $\mathbf{C}_{ij}$  darstellt.

Satz A.2.2 im Anhang A.2 gibt Auskunft über die Lösbarkeit des aus dem Variationsproblems abgeleiteten Gleichungssystems bei Verwendung der verschiedenen Funktionale.

Der Prozeß der Optimierung wird auf folgende Weise iteriert. Die im ersten Optimierungsschritt bestimmten Normalen an den Ecken  $\mathbf{V}_i$  des Polyeders werden als neue lokale  $z$ -Achsen, d.h. als neue Vektoren  $\hat{\mathbf{w}}_i$  herangezogen. Auf dieser Basis werden dann neue lokale Parameter  $a_{ij}, b_{ij}$  berechnet und die Optimierung erneut durchgeführt. Die vielen Tests mit diesem Iterationsansatz zeigen, daß in den meisten Fällen der Prozeß nach weniger als 5 Iterationsschritten konvergiert. In Extremfällen kann durch die Optimierung allerdings ein nicht ordnungserhaltendes oder ein nicht winkel-reguläres Kurvennetz entstehen (vgl. Abschnitt 2.2 auf Seite 25).

#### 4.1.4 Ergebnisse

In diesem Abschnitt werden die verschiedenen Verfahren zur Bestimmung der lokalen Parameter, sowie die unterschiedlichen Funktionale verglichen.

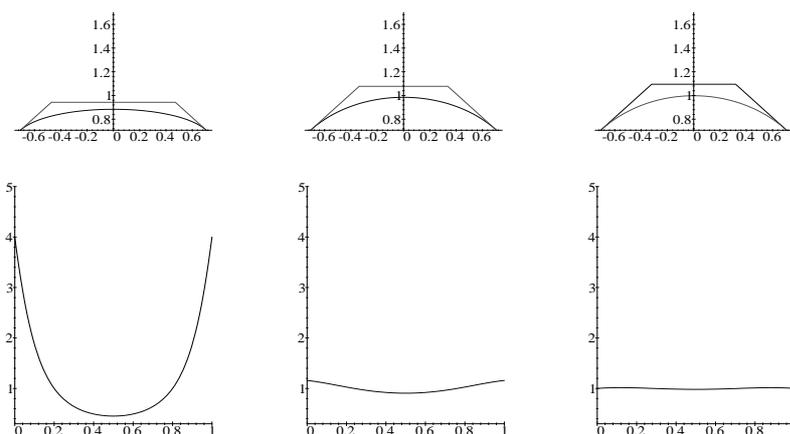


Abbildung 4.3: Die kubischen Interpolationskurven (oben) mit den zugehörigen Krümmungsplots (unten) für die verschiedenen Methoden zur Bestimmung der Tangentenlänge bei Viertelkreisdaten. Links die KL-Methode, in der Mitte die KS- bzw. vKS-Methode und ganz rechts die dBHS-Methode.

Abbildung 4.3 zeigt die Ergebnisse der verschiedenen Methoden zur Bestimmung der Länge von  $\vec{t}_{ij}$  bei Ansatz auf Viertelkreisdaten, d.h. Position und Tangentrichtungen entsprechen den Randdaten eines Viertelkreises. Dazu korrespondierend sind die jeweiligen Krümmungsplots der Kurven aufgetragen. Der Idealfall einer Kurve, die diese Daten interpoliert, wäre der Viertelkreis selbst mit konstanter Krümmung von 1.

Die KL-Methode zeigt einen sehr ungünstigen Verlauf (idealerweise müßte der Punkt  $(0, 1)$  interpoliert werden). Der entsprechende Krümmungsplot weist eine Konzentration der Krümmung nahe den Rändern auf. Der Kurvenverlauf bei der KS-Methode ist wesentlich näher an dem angestrebten Viertelkreis (hier erzeugen die KS- und die vKS-Methode dieselben Ergebnisse). Ebenso zeigt der Krümmungsverlauf eine günstigere Verteilung der Krümmung über die Gesamtkurve. Für die dBHS-Methode wurden die exakten Krümmungsdaten des Viertelkreises vorgegeben. Die resultierende Kurve approximieren den Viertelkreis sehr gut, wie der Krümmungsplot veranschaulicht.

Das zweite Beispiel ist im Gegensatz zum Vorhergehenden unsymmetrisch. Abbildung 4.4 zeigt die Positions- und Richtungdaten und ein zu diesen Daten eingepaßtes Ellipsensegment. Vergleicht man die Ergebnisse der verschiedenen Methoden, so erkennt man, daß keine Methode sehr nahe an das als Vergleichskurve verwendete Ellipsensegment herankommt. Jedoch ergeben sich erhebliche Unterschiede im Krümmungsverhalten.

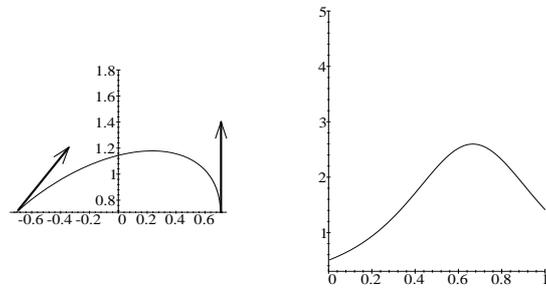


Abbildung 4.4: Das eingepaßte Ellipsensegment und der zugehörige Krümmungsplot.

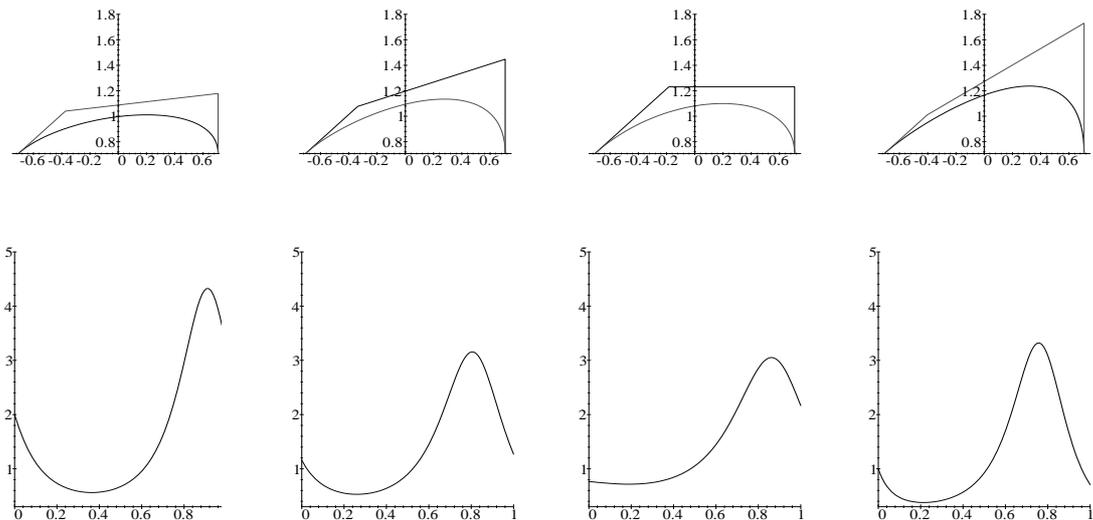


Abbildung 4.5: Die kubischen Interpolationskurven (oben) mit den zugehörigen Krümmungsplots (unten) für die verschiedenen Methoden zur Bestimmung der Tangentenlänge bei unsymmetrischen Daten. Links die KL-Methode, in der Mitte-links die KS-Methode, Mitte-rechts die vKS-Methode und rechts die dBHS-Methode.

Abbildung 4.5 zeigt die Ergebnisse zu diesem Beispiel. Die vKS-Methode schneidet hier am besten ab. Bei der dBHS-Methode kommt das Problem der Krümmungsabschätzung zum tragen. Die hier verwendeten Krümmungsdaten wurden mittels eines de Boor-Höllig-Sabin Kurvennetzes nach DeRose und Mann bestimmt (vgl. Abschnitt 2.2.1). Das zugrundeliegende Polyeder wurde dabei so gewählt, daß das Kurvennetz eine Kurve mit genau den Endpositionen und Endtangenterichtungen, wie in diesem Beispiel angesetzt, enthält.

Aufgrund der vielen Test, die im Laufe der Zeit mit den verschiedenen Methoden durchgeführt wurden, läßt sich folgendes feststellen:

**Richtungsbestimmung:** Bei der Bestimmung der Richtung von  $\vec{t}_{ij}$  liefert die EK-Methode meist die besseren Ergebnisse als die PK-Methode. Dies gilt allerdings nicht, wenn die Geometrie des gegebenen Polyeders dazu führt, daß das Kurvennetz nahezu nicht winkel-regulär oder nicht ordnungserhaltend ist (vgl. Bezeichnung 2.2.1 auf Seite 25). In solchen Situationen verhält sich die PK-Methode wesentlich stabiler als die EK-Methode.

**Längenbestimmung:** Die KL-Methode generiert immer dort zu flache Kurvennetze, wo bereits aufgrund der abgeschätzten  $\hat{\mathbf{w}}$ -Vektoren stärker gekrümmte Kurven zu erwarten sind. Die KS- bzw. die vKS-Methode liefern in nahezu allen Situationen gute Ergebnisse. Diese beiden Methoden halten sich hinsichtlich der Qualität der resultierenden Kurvennetze in etwa die Waage, wenngleich immer wieder Situationen beobachtet werden, in denen eine der Methoden erkennbar bessere Ergebnisse liefert. Die dBHS-Methode ist vom Ansatz her sehr gut. Das große Problem liegt allerdings in der Abschätzung der Krümmungsdaten. Die Methode von Moreton und Séquin zur Krümmungsschätzung liefert häufig nicht gut genug Ergebnisse, so daß die resultierenden Längen der Parameterrichtungen  $\vec{t}_{ij}$  zu Kurvennetzen mit schlechter Krümmungsverteilung führen.

Zum Abschluß geben wir ein einfaches Beispiel an, um die Wirkungsweise der Optimierung mit den verschiedenen Funktionalen zu demonstrieren. Gegeben sind hierbei vier Punkte auf dem Einheitskreis, die von einer geschlossenen Kurve interpoliert werden (vgl. Abbildungen 4.6 und 4.7). Die Länge der Parameterrichtungen  $\vec{t}_{ij}$  wird mit dem KS-Verfahren bestimmt. Die Initialisierung zeigt eine schlechte Verteilung der Krümmung. Die Ergebnisse einer einmaligen Optimierung mit dem einfachen, dem Laplace und dem dritte Ordnung Funktional zeigen ein deutlich verbessertes Verhalten sowohl hinsichtlich des visuellen Eindrucks wie der Krümmungsverteilung.

Wird der Prozeß iteriert, so erkennt man, daß die Ergebnisse basierend auf dem Laplace-Beltrami wie auf dem einfachen Funktional nahezu identisch sind obwohl nach nur einem Optimierungsschritt noch sehr große Unterschiede festzustellen sind. Bemerkenswert ist auch die in anderen Beispielen festgestellte Neigung des Laplace-Beltrami Ansatzes zur Generierung von krümmungsstetigen Kurven.

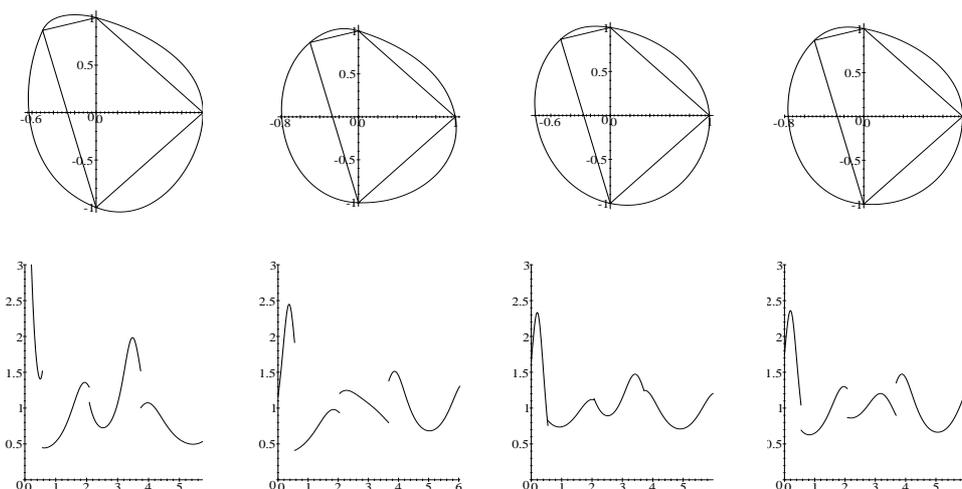


Abbildung 4.6: Die  $G^1$ -Kurvennetze (oben) mit den zugehörigen Krümmungsplots (unten) für die verschiedenen Funktionale nach einmaliger Optimierung bei den Kreisdaten. Links die Initialisierung, Mitte-links das einfache Funktional, Mitte-rechts das Laplace-Beltrami Funktional und rechts das Funktional dritter Ordnung.

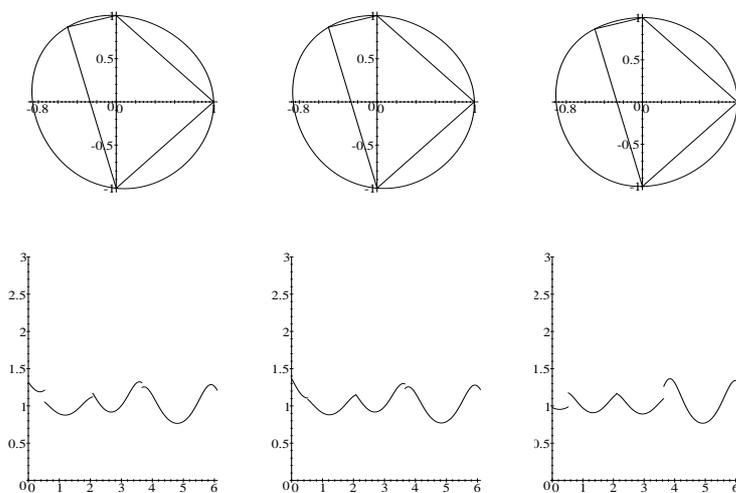


Abbildung 4.7: Die  $G^1$ -Kurvennetze (oben) mit den zugehörigen Krümmungsplots (unten) für die verschiedenen Funktionale nach bis zur Konvergenz iterierter Optimierung bei den Kreisdaten. Links das einfache Funktional, Mitte das Laplace-Beltrami Funktional und rechts das Funktional dritter Ordnung.

Über die Verwendung der verschiedenen Funktionale zur Optimierung des  $G^1$ -Kurvennetzes läßt sich zusammenfassend feststellen:

**Einfaches Funktional:** Dieses wird i.a. mit den Einstellungen  $\alpha = 1, \beta = 0$  eingesetzt. Es liefert in den meisten Situationen Kurvennetze mit guter Krümmungsverteilung.

Allerdings neigt es dazu „überschwingende“ Kurvennetzen zu generieren.<sup>1</sup> Von den drei hier vorgestellten Funktionalen hat es das beste Laufzeitverhalten, da explizite Formeln für die Stammfunktionen verwendet werden.

**Laplace-Beltrami Funktional:** Die Verwendung des Laplace-Beltrami Funktional ist zwar etwas rechenaufwendiger, vermeidet jedoch größtenteils das Überspringen, wie es bei Einsatz des einfachen Funktional vorkommen kann. Zudem zeigt es ein sehr stabiles Konvergenzverhalten.

**Funktional dritter Ordnung:** Dieses Funktional erfordert mehr Rechenaufwand als das Laplace-Beltrami Funktional, da höhere Ableitungen bestimmt werden müssen und der Funktionalausdruck deutlich komplexer ist. Es kann nur eingeschränkt verwendet werden (vgl. Anhang A.2). Die Qualität der Kurvennetze unterscheidet sich kaum von der Qualität der resultierenden Kurvennetze beim Laplace-Beltrami Funktional.

## 4.2 Ausfüllen des $G^1$ -Kurvennetzes

In diesem Abschnitt wird eine neue Variante des Verfahrens von Shirman-Séquin (vgl. Abschnitt 2.2.1 auf Seite 26) zum Ausfüllen des optimierten  $G^1$ -Kurvennetzes vorgestellt. Hierbei werden eine Reihe von Parametern, die nicht durch die Interpolations- bzw. die Glattheitsbedingungen bestimmt sind, variiert. Dies geschieht einerseits unter Verwendung einer geometrieabhängigen, heuristischen Regel. Andererseits können verschiedene Parameter mittels eines Variationsansatzes optimiert werden. Am Ende dieses Abschnittes werden die verschiedenen Funktionale und Optimierungsmethoden kurz miteinander verglichen.

Die Formeln in Abschnitt 2.2.1 zur Berechnung der Kontrollpunkte der Teilflächen des Shirman-Séquin Verfahrens sind bereits in allgemeinerer Form als in der ursprünglichen Version angegeben (vgl. [SS87]). Dadurch wird erkennbar, daß eine  $3 \times 4$  Splitfläche im ganzen fünf skalare und drei vektorwertige Freiheitsgrade beinhaltet, welche nicht durch die Übergangs- oder die Interpolationsbedingungen bestimmt sind. Zwei der skalaren Bedingungen stecken in der Bestimmung des Zentrums  $\mathbf{Z}$  (vgl. Gleichung (2.8) auf Seite 28). Die anderen drei skalaren Freiheitsgrade sind durch die  $\beta_i, \beta_j, \beta_k$  gegeben, welche die Lage des ersten Kontrollpunktes der inneren kubischen Randkurve festlegen (vgl. Gleichung (2.4) und Abbildung 2.6). Die drei vektorwertigen Freiheitsgrade  $\vec{\mathbf{q}}_{i,1}, \vec{\mathbf{q}}_{j,1}, \vec{\mathbf{q}}_{k,1}$  beschreiben das Querableitungsvektorfeld entlang einer Kurve des Kurvennetzes (vgl. Gleichung (2.5)).

Wie Gleichung (2.7) zeigt, hängen einige Kontrollpunkte der resultierenden Splitfläche nicht linear von den Skalaren  $\beta_i, \beta_j, \beta_k$  ab. Damit können diese Werte nicht zur Optimierung mit einem quadratischen Funktional herangezogen werden. Jedoch ist die Wahl von  $\beta_i = \beta_j = \beta_k = \frac{1}{3}$ , wie sie von Shirman und Séquin getroffen wird, zu unflexibel. Es zeigt sich, daß diese Wahl gute Ergebnisse liefert, falls die Kurven des Kurvennetz für eine Öffnung eine nahezu flache Splitfläche erwarten lassen. Sind die Kurven jedoch stark gekrümmt, so generiert das ursprüngliche Verfahren von Shirman und Séquin Flächen mit starker Krümmung am Rand und sehr flachem Verlauf in der Mitte.

<sup>1</sup>Diese Eigenschaft, auch „overshooting“ genannt, wird ebenfalls bei univariaten B-Splines und funktionalen Kurvennetzen beobachtet (vgl. [NF84]).

Wir setzen, im Unterschied zu Shirman und Séquin, den Skalarwert  $\beta_i$  in Abhängigkeit von der Geometrie der gegebenen Randkurven. Hierzu verwenden wir die KS-Methode, wie sie zur Bestimmung der Länge der Parameterichtung eingesetzt wird (vgl. Abschnitt 4.1.2 auf Seite 55).  $s$  beschreibt den durch die Ecke  $\mathbf{V}_i$ , den Kurvenmittelpunkt  $\mathbf{C}_{jk}(0.5)$  und die gegebene Richtung  $\mathbf{D}_{i,1}^3 - \mathbf{V}_i$  definierten Kreisbogen (vgl. Abbildung 4.8). Wir skalieren  $\beta_i$  mit dem Quotient  $\|s\| / \|l\|$ , wobei  $l$  die Entfernung von  $\mathbf{V}_i$  zum gegenüberliegenden Kantenmittelpunkt  $\frac{1}{2}(\mathbf{V}_j + \mathbf{V}_k)$  ist:  $\beta_i = \frac{\|s\|}{3\|l\|}$ .

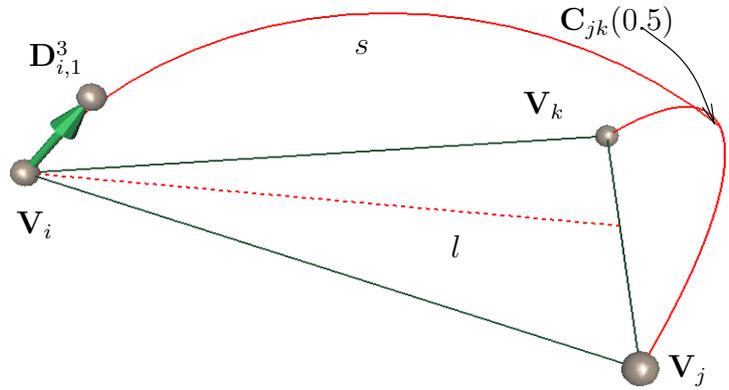


Abbildung 4.8: Geometrieabhängige Definition des Flächenparameters  $\beta_r$ .

Im Gegensatz zu den Skalaren  $\beta_i, \beta_j, \beta_k$  können die Querableitungsvektoren  $\vec{\mathbf{q}}_{i,1}, \vec{\mathbf{q}}_{j,1}, \vec{\mathbf{q}}_{k,1}$  durch einen Variationsansatz mit einem quadratischen Funktional bestimmt werden. Gleichungen (2.5)-(2.8) zeigen, daß alle Kontrollpunkte der Splitfläche linear von diesen Querableitungsvektoren abhängen.

Ähnlich wie im Kurvenfall werden zur Optimierung der Splitfläche verschiedene Funktionale eingesetzt. Hierbei kommen neben den einfachen Funktionalen (Laplace- und Thin-Plate Funktional) auch datenabhängige Funktionale zum Einsatz: Das Laplace-Beltrami- und das datenabhängige Thin-Plate-Funktional, sowie das Funktional dritter Ordnung (vgl. [Gre94b, Gre94a, Gre94c]). Der Einsatz datenabhängiger Funktionale ist wiederum durch deren gute Approximationseigenschaft an Funktionale motiviert, die exakte Krümmungseigenschaften von Flächen, wie die mittlere Krümmung, bewerten.

Das Laplace- und das Thin-Plate-Funktional sind gegeben durch:

$$\sigma^L(\mathbf{S}) = \int_{\Omega} \langle \Delta \mathbf{S} \mid \Delta \mathbf{S} \rangle du dv, \quad \text{bzw.} \quad \sigma^{TP}(\mathbf{S}) = \int_{\Omega} (\mathbf{S}_{uu})^2 + 2(\mathbf{S}_{uv})^2 + (\mathbf{S}_{vv})^2 du dv. \quad (4.10)$$

Wie im Kurvenfall werden zudem die von Greiner eingeführten geometrieabhängigen Funktionale verwendet

Zur Beschreibung des Laplace-Beltrami-Operator benötigen wir, ähnlich wie im Kurvenfall, den Gradienten von Funktionen und die Divergenz von tangentialen Vektorfeldern auf einer Fläche  $\mathbf{S}^0 : \Omega \rightarrow \mathbb{R}^3$ ,  $\Omega \subseteq \mathbb{R}^2$ . Sei  $f : \Omega \rightarrow \mathbb{R}$  eine skalarwertige Funktion und  $\vec{\mathbf{f}} = (f_1, f_2)$  ein tangentiales Vektorfeld auf  $\mathbf{S}^0$ .<sup>2</sup> Der Gradient  $\text{grad}_{\mathbf{S}^0}(f)$  von  $f$  und die Divergenz

<sup>2</sup> $f(u, v)$  wird hierbei mit dem Wert von  $f$  bei dem Flächenpunkt  $\mathbf{S}^0(u, v)$  identifiziert; analoges gilt für  $\vec{\mathbf{f}}$ .

$\operatorname{div}_{\mathbf{S}^0}(\vec{\mathbf{f}})$  von  $\vec{\mathbf{f}}$  bzgl. der Fläche  $\mathbf{S}^0$  sind unter Verwendung der ersten Fundamentalform  $\mathbf{l}_{\mathbf{S}^0}$  von  $\mathbf{S}^0$  definiert als

$$\operatorname{grad}_{\mathbf{S}^0}(f) = (f_u, f_v)\mathbf{l}_{\mathbf{S}^0}^{-1} \quad \text{bzw.} \quad \operatorname{div}_{\mathbf{S}^0}(\vec{\mathbf{f}}) = \frac{1}{\sqrt{g}} \left( (\sqrt{g}f_1)_u + (\sqrt{g}f_2)_v \right), \quad (4.11)$$

wobei  $g = \det(\mathbf{l}_{\mathbf{S}^0})$ .

Der Laplace-Beltrami Operator für die Fläche  $\mathbf{S}(u, v) : \Omega \rightarrow \mathbb{R}^3$ ,  $\mathbf{S}(u, v) = (S_1(u, v), S_2(u, v), S_3(u, v))^t$  ergibt sich zu

$$\Delta_{\mathbf{S}^0}(\mathbf{S}) = \operatorname{div}_{\mathbf{S}^0}(\operatorname{grad}_{\mathbf{S}^0})(\mathbf{S}) = \begin{pmatrix} \operatorname{div}_{\mathbf{S}^0}(\operatorname{grad}_{\mathbf{S}^0})(S_1) \\ \operatorname{div}_{\mathbf{S}^0}(\operatorname{grad}_{\mathbf{S}^0})(S_2) \\ \operatorname{div}_{\mathbf{S}^0}(\operatorname{grad}_{\mathbf{S}^0})(S_3) \end{pmatrix}.$$

Mit Hilfe dieses Operators wird das entsprechende Funktional definiert:

$$\sigma_{\mathbf{S}^0}^{LB}(\mathbf{S}) = \int_{\Omega} \langle \Delta_{\mathbf{S}^0}(\mathbf{S}) \mid \Delta_{\mathbf{S}^0}(\mathbf{S}) \rangle d\omega_{\mathbf{S}^0}.$$

Hierbei ist  $d\omega_{\mathbf{S}^0}$  das Flächenelement der Referenzfläche  $\mathbf{S}^0$ .

Ist die Fläche  $\mathbf{S}$  identisch mit der Referenzfläche  $\mathbf{S}^0$ , so gilt (vgl. [Gre94b]):

$$\sigma_{\mathbf{S}^0}^{LB}(\mathbf{S}) = \int_{\Omega} (\kappa_1 + \kappa_2)^2 d\omega_{\mathbf{S}} = 4 \int_{\Omega} \mathbf{H}_{\mathbf{S}}^2 d\omega_{\mathbf{S}}.$$

Wie im Kurvenfall kann auch für Flächen ein Operator dritter Ordnung unter Verwendung des Gradienten und der Divergenz angesetzt werden (vgl. Abschnitt 1.3 auf Seite 13 und Gleichung (4.11)):  $\operatorname{grad}_{\mathbf{S}^0}(\operatorname{div}_{\mathbf{S}^0}(\operatorname{grad}_{\mathbf{S}^0}))(\mathbf{S})$ . Das hierauf aufbauende Funktional hat folgende Gestalt:

$$\sigma_{\mathbf{S}^0}^{TO}(\mathbf{S}) = \int_{\Omega} \langle \operatorname{grad}_{\mathbf{S}^0}(\Delta_{\mathbf{S}^0})(\mathbf{S}) \mid \operatorname{grad}_{\mathbf{S}^0}(\Delta_{\mathbf{S}^0})(\mathbf{S}) \rangle_{\mathbf{T}\mathbf{S}^0} d\omega_{\mathbf{S}^0}.$$

Für das Funktional dritter Ordnung wird das innere Produkt im Tangentialraum der Referenzfläche  $\mathbf{S}^0$  verwendet. Seien  $\vec{\mathbf{v}}_1$  und  $\vec{\mathbf{v}}_2$  zwei tangentielle Vektorfelder für die Fläche  $\mathbf{S}^0$ , so ist  $\langle \vec{\mathbf{v}}_1 \mid \vec{\mathbf{v}}_2 \rangle_{\mathbf{T}\mathbf{S}^0} = \vec{\mathbf{v}}_1^t \mathbf{l}_{\mathbf{S}^0} \vec{\mathbf{v}}_2$ .

Fallen für das Funktional dritter Ordnung die Flächen  $\mathbf{S}$  und  $\mathbf{S}^0$  zusammen, so gilt nach [Gre94b]:

$$\sigma_{\mathbf{S}^0}^{TO}(\mathbf{S}) = 4 \int_{\Omega} \langle \operatorname{grad}_{\mathbf{S}^0}(\mathbf{H}_{\mathbf{S}}) \mid \operatorname{grad}_{\mathbf{S}^0}(\mathbf{H}_{\mathbf{S}}) \rangle_{\mathbf{T}\mathbf{S}^0} + 4 (\kappa_1^2 + \kappa_2^2) \mathbf{H}_{\mathbf{S}}^2 d\omega_{\mathbf{S}}.$$

Als weiteres Funktional wird das in Abschnitt 3.2 auf Seite 40 beschriebenen datenabhängige Thin-Plate Funktional eingesetzt.

Die Kontrollpunkte der jeweiligen Referenzfläche  $\mathbf{S}^0$  ergeben sich durch die Abschätzung  $\vec{\mathbf{q}}_{k,1} = \frac{1}{2} (\vec{\mathbf{q}}_{k,0} + \vec{\mathbf{q}}_{k,2})$  aus den Formeln (2.4)–(2.8). Wird das Verfahren iteriert, so werden

die im vorherigen Schritt bestimmten Querableitungsvektoren  $\vec{q}_{k,1}$  zur Bestimmung der Referenzfläche im nächsten Schritt verwendet.

Das Funktional für die Gesamtfläche ergibt sich durch Aufsummierung der Teilbeiträge aller einzelnen  $3 \times 4$ -Splitflächen. Wird das Funktional  $\sigma$  mit dem zugehörigen inneren Produkt  $\langle \cdot | \cdot \rangle_\sigma$ ,  $\sigma(\mathbf{S}) = \langle \mathbf{S} | \mathbf{S} \rangle_\sigma$  verwendet, so müssen zur Lösung des Optimierungsproblems unter Variation der Querableitungsvektoren  $\vec{q}_{i,1}$  die Gleichungen

$$\frac{\partial}{\partial \vec{q}_{i,1}} \sigma(\mathbf{S}) = 0, \quad i = 1, \dots, N_e, \quad (4.12)$$

gelöst werden.  $\vec{q}_{i,1}$  bezeichnet hier die zu optimierenden Querableitungsvektoren,  $N_e$  ist die Anzahl der Kanten des Polyeders. Für feste Parameter  $\beta_i$  hängt die Shirman-Séquin Fläche linear von den Kontrollpunkten der Kurven des Kurvennetzes und von den Querableitungen  $\vec{q}_{i,1}$  ab. Um Gleichung (4.12) auflösen zu können, benötigt man eine Basisdarstellung der Splitfläche  $\mathbf{S}$  in den Kontrollparametern  $\mathbf{V}_r, \dots, \mathbf{C}_{s,2}^3, \vec{q}_{t,1}, \vec{q}_{r,1}, \vec{q}_{s,1}$  mit den Basisfunktionen  $B_{\mathbf{V}_r}, \dots, B_{\vec{q}_{s,1}}$ . Diese Basisdarstellung ist schwierig zu handhaben. Man kann jedoch zeigen (vgl. A.3), daß sich die gesuchten Werte der inneren Produkte für verschiedene Basisdarstellungen einfach ineinander überführen lassen. Wir bestimmen daher die inneren Produkte für die Bernstein-Bézier-Basis der einzelnen Teilflächen und ermitteln daraus die benötigten inneren Produkte bzgl. der Basis  $\{B_{\mathbf{V}_r}, \dots, B_{\vec{q}_{s,1}}\}$ .

### 4.3 Simultane Optimierung

In den vorherigen Abschnitten wurde gezeigt, wie das parametrische Interpolationsproblem basierend auf dem optimierten  $G^1$ -Kurvennetz gelöst wird. Dabei werden das Kurvennetz und die  $3 \times 4$ -Splitflächen in zwei getrennten Schritten betrachtet und separat optimiert. Wie die Analyse dieses Abschnitts zeigt, können diese Optimierungen auch in einem einzigen Schritt durchgeführt werden. Damit werden alle freien Parameter in der Kurvennetz- und in der Flächendarstellung durch ein *Flächenfunktional* bewertet.

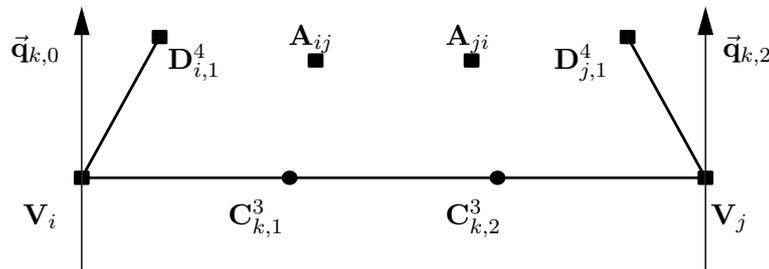


Abbildung 4.9: Die Konstruktion der Kontrollpunkte  $\mathbf{A}_{ij}$  und  $\mathbf{A}_{ji}$  mittels der Methode von Chiyokura-Kimura: Die Randkurve ist durch die Kontrollpunkte  $\{\mathbf{V}_i, \mathbf{C}_{k,1}^3, \mathbf{C}_{k,2}^3, \mathbf{V}_j\}$ , das Querableitungsvektorfeld in den Eckpunkten durch  $\vec{q}_{k,0}$  bzw.  $\vec{q}_{k,2}$  bestimmt. Die ersten Kontrollpunkte der inneren Randkurven der Splitfläche,  $\mathbf{D}_{i,1}^4$  und  $\mathbf{D}_{j,1}^4$  sind in quartischer Darstellung gegeben.

Die wesentliche Voraussetzung zur Durchführung einer simultanen Optimierung ist die lineare Abhängigkeit der Kontrollpunkte der  $3 \times 4$  Splitfläche von den Kurvennetzparameter  $S_{ui}, S_{vi}, i = 1, \dots, N$ .

Im folgenden zeigen wir, daß diese lineare Abhängigkeit der Kontrollpunkte von der Splitfläche gegeben ist. Abbildung 4.9 zeigt die entsprechenden Kontrollpunkte und -vektoren bei der Kante  $\overline{\mathbf{V}_i \mathbf{V}_j}$ , die für die weitere Betrachtung von Bedeutung sind.

**Der Kurvenkontrollpunkt  $\mathbf{C}_{i,1}^3$ :** Der Kontrollpunkt  $\mathbf{C}_{k,1}^3$  ergibt sich für die entsprechende Parameterrichtung  $\vec{\mathbf{t}}_{ij} = a_{ij}\hat{\mathbf{u}}_i + b_{ij}\hat{\mathbf{v}}_i$  in den lokalen Koordinaten  $\{\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i, \hat{\mathbf{w}}_i\}$  zu (vgl. Gleichung (4.5) auf Seite 54)<sup>3</sup>:

$$\mathbf{C}_{i,1}^3 = \mathbf{V}_i + \frac{1}{3}\mathbf{C}'_{ij}(0) = \frac{1}{3} \begin{pmatrix} a_{ij} \\ b_{ij} \\ a_{ij}S_{ui} + b_{ij}S_{vi} \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 1 & 0 & * \\ 0 & 1 & * \\ S_{ui} & S_{vi} & * \end{pmatrix} \begin{pmatrix} a_{ij} \\ b_{ij} \\ 0 \end{pmatrix}. \quad (4.13)$$

Im weiteren bezeichnen wir die Matrix auf der rechten Seite von 4.13 mit  $A_i$ .

$\mathbf{C}_{i,1}^3$  und analog  $\mathbf{C}_{i,2}^3$  hängt linear über die durch  $A_i$  bzw.  $A_j$  induzierte affine Abbildung von  $S_{ui}, S_{vi}$  bzw.  $S_{uj}, S_{vj}$  ab.

**Der innere Kontrollpunkt  $\mathbf{D}_{i,1}^3$ :** Für festes  $\beta_i$  ergibt sich der Punkt  $\mathbf{D}_{i,1}^3$  als affines Bild eines entsprechenden Punktes in der  $(\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i)$ -Ebene (vgl. Gleichung (2.4) auf Seite 27):

$$\mathbf{D}_{i,1}^3 = \begin{pmatrix} \beta_i(a_{ij} + a_{ik}) \\ \beta_i(b_{ij} + b_{ik}) \\ \beta_i(a_{ij} + a_{ik})S_{ui} + \beta_i(b_{ij} + b_{ik})S_{vi} \end{pmatrix} = \frac{1}{3}A_i \begin{pmatrix} \beta_i(a_{ij} + a_{ik}) \\ \beta_i(b_{ij} + b_{ik}) \\ 0 \end{pmatrix}.$$

**Der Querableitungsvektor  $\vec{\mathbf{q}}_{k,0}$ :** Wählen wir den Querableitungsvektor  $\vec{\mathbf{q}}_{k,0}$  in den lokalen Koordinaten  $\{\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i, \hat{\mathbf{w}}_i\}$  zu

$$\vec{\mathbf{q}}_{k,0} = \begin{pmatrix} e_{ij} \\ f_{ij} \\ e_{ij}S_{ui} + f_{ij}S_{vi} \end{pmatrix} = A_i \begin{pmatrix} e_{ij} \\ f_{ij} \\ 0 \end{pmatrix}, \quad (4.14)$$

so ist  $\vec{\mathbf{q}}_{k,0}$  das affine Bild von  $(e_{ij}, f_{ij}, 0)^t$  unter derselben, von  $A_i$  induzierten, affinen Abbildung.

**Unabhängigkeit der Gewichtspolynome:** Für die lineare Abhängigkeit der Kontrollpunkte  $\mathbf{A}_{ij}$  und  $\mathbf{A}_{ji}$  der Splitfläche von den Parametern  $S_{ui}, S_{vi}$  ist die Unabhängigkeit der Gewichtspolynome  $a_k, b_k$  von diesen Parametern entscheidend. Es gilt:

$$\begin{aligned} (\mathbf{D}_{i,1}^4 - \mathbf{V}_i) &= a_{k,0}(\mathbf{C}_{k,1}^3 - \mathbf{V}_i) + b_{k,0}\vec{\mathbf{q}}_{k,0} \\ \iff \frac{3\beta_i}{4} \begin{pmatrix} \beta_i(a_{ij} + a_{ik}) \\ \beta_i(b_{ij} + b_{ik}) \\ (a_{ij} + a_{ik})S_{ui} + (b_{ij} + b_{ik})S_{vi} \end{pmatrix} &= a_{k,0} \begin{pmatrix} a_{ij} \\ b_{ij} \\ a_{ij}S_{ui} + b_{ij}S_{vi} \end{pmatrix} + b_{k,0} \begin{pmatrix} e_{ij} \\ f_{ij} \\ e_{ij}S_{ui} + f_{ij}S_{vi} \end{pmatrix} \\ \iff A_i \left( \frac{3\beta_i}{4} \begin{pmatrix} a_{ij} + a_{ik} \\ b_{ij} + b_{ik} \\ 0 \end{pmatrix} - a_{k,0} \begin{pmatrix} a_{ij} \\ b_{ij} \\ 0 \end{pmatrix} - b_{k,0} \begin{pmatrix} e_{ij} \\ f_{ij} \\ 0 \end{pmatrix} \right) &= 0, \end{aligned}$$

<sup>3</sup>Die Matrixeinträge \* sind beliebig und im weiteren nicht von Bedeutung.

womit die gewünschte Unabhängigkeit von  $S_{ui}$  und  $S_{vi}$  gezeigt ist.

Die Formeln zur Bestimmung aller weiteren Kontrollpunkte der Teilflächen der  $3 \times 4$  Splitfläche hängen linear von den bereits bestimmten Kontrollpunkten ab (vgl. Gleichungen (2.4) – (2.8) auf Seite 28). Werden die Querableitungsvektoren  $\vec{\mathbf{q}}_{r,0}, \vec{\mathbf{q}}_{r,2}$ ,  $r \in \{i, j, k\}$  wie in (4.14) angesetzt, so sind alle Kontrollpunkte der Splitfläche linear in den Parametern  $S_{ur}, S_{vr}, \vec{\mathbf{q}}_{r,1}$ ,  $r \in \{i, j, k\}$ .

Durch Ansatz eines quadratischen Funktionals zur Bewertung der gesamten Splitfläche können die freien Parameter  $S_{ur}, S_{vr}, \vec{\mathbf{q}}_{r,1}$ ,  $r \in \{i, j, k\}$  durch Ansatz eines Flächenfunktionals simultan optimiert werden.

Die simultane Optimierung kann ebenfalls iterativ angesetzt werden. Hierbei wird, wie bei der Optimierung des  $G^1$ -Kurvennetzes, die durch die Optimierung entstehenden Normalen in den Ecken des Polyeders als neue lokale  $z$ -Achsen, d.h. als Vektoren  $\hat{\mathbf{w}}_i$ , verwendet. In jedem Schritt sind die lokalen Parameter neu zu bestimmen.

## 4.4 Ergebnisse

In diesem Abschnitt werden die Einflüsse der in Abschnitt 4.2 beschriebenen Methode zur Bestimmung der Parameter  $\beta_i, \beta_j, \beta_k$ , sowie die Wahl des Funktionals und der Optimierungsart (getrennte bzw. simultane Optimierungs) auf die resultierende Interpolationsfläche diskutiert.

**Wahl der Parameter  $\beta_i, \beta_j, \beta_k$ :** In Abbildung 4.10, unten recht, ist die Gaußkrümmung für die Interpolationsflächen zu einem Beispieldatensatz angegeben. Bei Verwendung der Standardheuristik nach Shirman-Séquin zeigen sich starke Krümmungen nahe den Übergängen zwischen den Teilflächen. Die oben beschriebene, geometrieabhängige Methode liefert hingegen eine Fläche mit wesentlich besserer Krümmungsverteilung.

**Einfluß des Funktionals:** Im ganzen stehen fünf verschiedene Funktionale zur Optimierung der Flächenparameter zur Verfügung: Das einfache Laplace- und das einfache Thin-Plate Funktional, sowie drei datenabhängige Funktionale (Laplace-Beltrami, datenabhängiges Thin-Plate Funktional und Funktional dritter Ordnung). Abbildung 4.10, oben, zeigt die verschiedenen Ergebnisse im Vergleich zur nicht optimierten Fläche bei Anwendung auf der Sphärendaten (vgl. Abbildung 5.6). Hierbei wurde in allen Fällen, ausgehend von einem nicht optimierten Kurvennetz, simultan optimiert. Die drei datenabhängigen Funktional liefern in diesem Beispiel bessere Ergebnisse als die entsprechenden einfachen Funktionale. In allen Fällen ist eine deutliche Verbesserung der Flächenqualität erkennbar.

**Zweischritt- und simultane Optimierung:** Wie Abbildung 4.10, unten links, belegt, führt die simultane Optimierung zu wesentlich besseren Krümmungsverteilungen als die getrennte Optimierung von Kurvennetz- und Flächenparametern.

In diesem Beispiel liefern die beiden datenunabhängigen Funktionale, das Laplace und das Thin-Plate Funktional in etwa dieselben Ergebnisse. In anderen Fällen zeigt sich meist ein leicht besseres Ergebnis bei Verwendung des Thin-Plate Funktionals. Die datenabhängigen Funktionale liefern in dem angeführten Beispiel bessere Ergebnisse als einfache Funktionale. In anderen Fällen tendiert das Laplace-Beltrami Funktional jedoch dazu, Teilflächen flach

zu machen. Dies zeigt sich gerade bei der simultanen Optimierung. Die Ergebnisflächen bei der Optimierung mit dem datenabhängigen Thin-Plate Funktional und dem Funktional dritter Ordnung weisen in fast allen Tests die günstigsten Krümmungsverteilungen auf.

Sinnvoll ist stets eine Aufwands- und Nutzenabschätzung, da die datenabhängigen Funktionale einen größeren Rechenaufwand erfordern als die einfachen Funktionale. In vielen Situationen sind die bringt die Verwendung einfachen Funktionale bereits gut Ergebnisse.

Des weiteren weisen die mit der simultanen Optimierung generierten Interpolationsflächen in fast allen Fällen eine gleichmäßigere Krümmungsverteilung auf als die mit der Zweischrittmethod bestimmten Flächen. Allerdings steigt gegenüber der Zweischrittoptimierung bei simultanen Optimierung der Aufwand zur Lösung des Optimierungssystems. Dies ist vor allem darauf zurückzuführen, daß Zahl der sich gegenseitig beeinflussenden Optimierungsparameter bei der simultanen Optimierung wesentlich größer ist als bei der getrennten Optimierung.

Es zeigt sich außerdem, daß bei simultaner Optimierung unter Ansatz des Laplace- und des Laplace-Beltrami-Funktional oder des Funktional dritter Ordnung in einigen Situationen das zugehörige Gleichungssystem nicht lösbar ist. Ein theoretische Ergründung dieses Phänomens ist bislang nicht gelungen. Es deutet vieles darauf hin, daß ein Hauptproblem in der *Semi*-Positivität des Laplace-Operators auf Dreiecks-Bézierflächen, die nur in den Ecken fixiert sind, liegt.<sup>4</sup>

Bei Anwendung der Thin-Plate bzw. datenabhängigen Thin-Plate Funktionalen werden diese Probleme nicht beobachtet.

Die Stabilitätsprobleme bei der simultanen Optimierung unter Verwendung des Laplace- und des Laplace-Beltrami-Funktionalen, sowie des Funktional dritter Ordnung treten ausschließlich bei Polyedern mit Rand auf. Zudem ergeben sich bei Polyedern mit größeren flachen Regionen häufiger Probleme als bei stärker gekrümmten Polyedern.

Abschließend sei bemerkt, daß es sich in der Praxis gezeigt hat, daß bei der Flächenoptimierung wenige, meist sogar nur ein Optimierungsschritt ausreicht, um eine qualitativ gute Interpolationsfläche zu erhalten. Die Qualitätsverbesserung durch weitere Iterationen steht i.a. in keinem Vergleich zum Aufwand.

---

<sup>4</sup>Beispielsweise verschwindet das quadratische Polynom  $S(u, v) = uv$  für  $(u, v) \in \{(0, 0), (1, 0), (0, 1)\}$  und es gilt:  $\Delta(S) = 0$ .

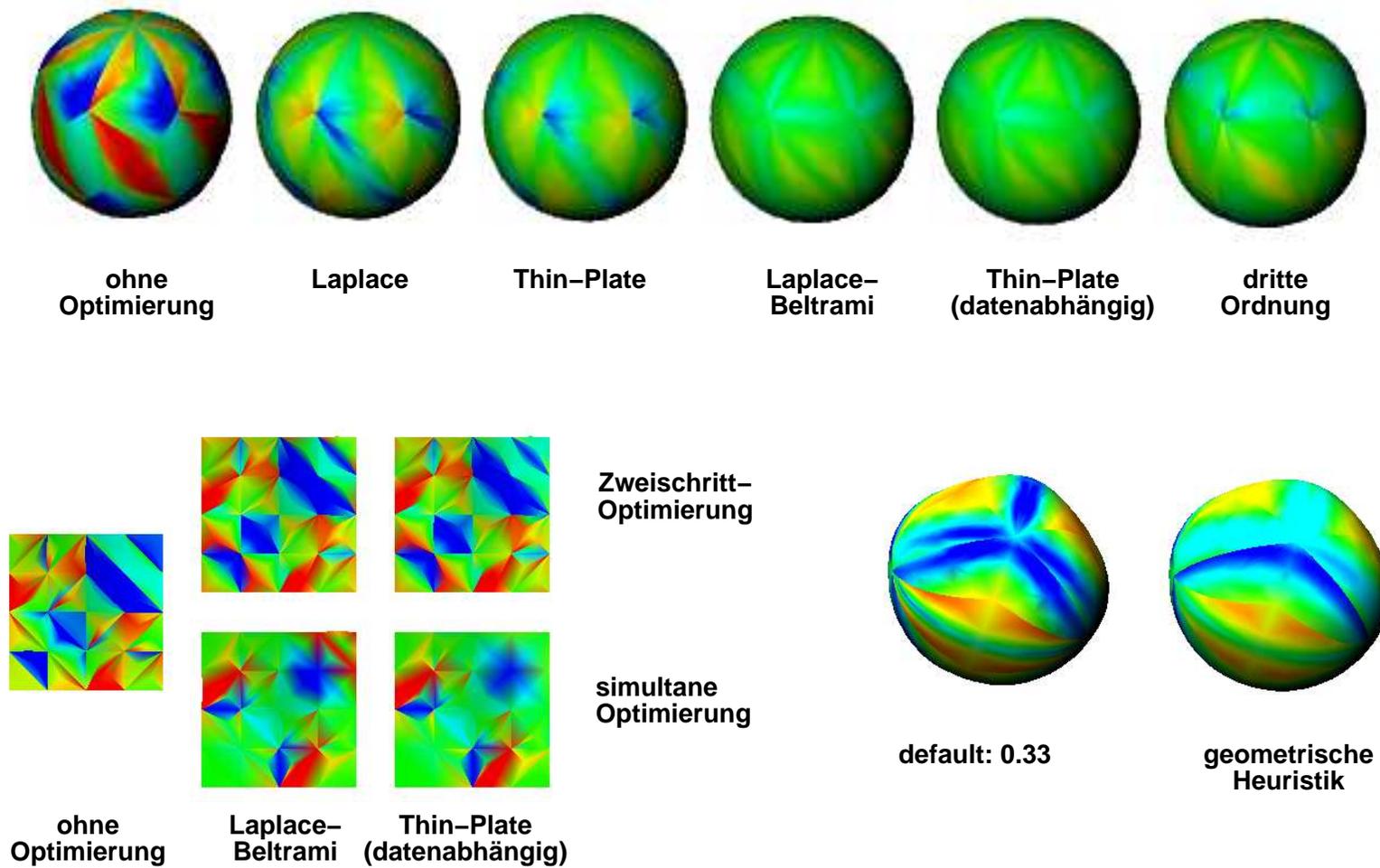


Abbildung 4.10: Gauß-Krümmungsplots zum Vergleich der verschiedenen Methoden und Optimierungsfunktionale: Oben der Vergleich der Funktionale; unten links der Vergleich der getrennten und der simultanen Optimierung; unten rechts die Ergebnisse bei Verwendung der Standardheuristik nach Shirman-Séquin und der geometrieabhängigen Methode.



# Kapitel 5

## Parametrische Interpolation basierend auf $G^2$ -Kurvennetzen

In diesem Kapitel wird eine neue Methode zur Lösung des parametrischen Interpolationsproblems vorgestellt, die auf der Konstruktion und Optimierung eines  $G^2$ -stetigen Kurvennetzes beruht. Diese Methode wird im weiteren  *$G^2$ -Kurvennetzmethode* genannt. In Abschnitt 5.1 wird zunächst die Repräsentation des verwendeten quintischen  $G^2$ -Kurvennetzes beschrieben. Die Erweiterung des quintischen Kurvennetzes zu einer global tangentialstetigen Fläche unter Verwendung von einer Dreiecks-Bézierfläche vom Grad 7 pro Öffnung des Kurvennetzes wird in Abschnitt 5.2 behandelt. Wie für die  $G^1$ -Kurvennetzmethode kann auch für die  $G^2$ -Kurvennetzmethode eine simultane Optimierung durchgeführt werden (Abschnitt 5.3). Abschließend werden in Abschnitt 5.4 einige bekannte Interpolationsverfahren mit der  $G^1$ - und der  $G^2$ -Kurvennetzmethode verglichen.

### 5.1 Das quintische $G^2$ -Kurvennetz

Die Erweiterung der Repräsentation des  $G^1$ -Kurvennetzes auf  $G^2$  stetige Kurvennetze beruht auf ähnlichen Ansätzen wie im funktionalen Fall. Dort gibt es, wie bereits in Abschnitt 2.1.2 auf Seite 21 erläutert, Methoden zur Generierung von  $C^2$ -stetigen Kurvennetzes. Für höher stetige Kurvennetze wurden zudem Querableitungsvektorfelder in die Beschreibung der Kurvennetze integriert (vgl. Tabelle 2.1 und [Pot91]). Im parametrischen Fall wird das Kurvennetz ebenfalls durch Raumkurven und Querableitungsvektorfelder definiert. Dadurch entsteht eine Kurvennetzbeschreibung, die „flächenähnlich“ ist.

In Abschnitt 5.1.1 wird die Repräsentation des Kurvennetzes unter Einbeziehung von Querableitungsvektorfeldern beschrieben. Abschnitt 5.1.2 zeigt auf, wie die zusätzlichen lokalen Parameter für das  $G^2$ -Kurvennetz gewählt werden. Die Optimierung des Kurvennetzes wird in Abschnitt 5.1.3 dargelegt. Abschließend werden in Abschnitt 5.1.4 die Methoden zu Bestimmung der lokalen Parameter, sowie die  $G^1$ - und die  $G^2$ -Kurvennetze miteinander verglichen.

### 5.1.1 Repräsentation des $G^2$ -Kurvennetzes

In Abschnitt 4.1 wurde gezeigt, wie ein kubisches  $G^1$ -Kurvennetz lokal in den Ecken  $\mathbf{V}_i$  des gegebenen Polyeders dargestellt wird. Um ein  $G^2$ -Kurvennetz zu erhalten, muß, wie nach Bezeichnung 1.2.1 auf Seite 7 dargestellt, neben der  $G^1$ -Kompatibilität der Kurven  $\mathbf{C}_{ij}$  in den Ecken  $\mathbf{V}_i$  noch eine weitere Bedingung erfüllt werden. Der Krümmungsvektor  $\vec{\mathbf{k}}_{\mathbf{C}_{ij}}$  von  $\mathbf{C}_{ij}$  muß für  $u = 0$  bzgl. der Flächennormalen  $\hat{\mathbf{n}}_{\mathbf{S}}$  und der zweiten Fundamentalform  $\mathbb{I}_{\mathbf{V}_i}$  folgender Beziehung genügen:

$$\langle \vec{\mathbf{k}}_{\mathbf{C}_{ij}} \mid \hat{\mathbf{n}}_{\mathbf{S}} \rangle = (t_1, t_2) \mathbb{I}_{\mathbf{V}_i} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}. \quad (5.1)$$

Hierbei sind  $t_1, t_2$  die Komponenten der normierten Tangente  $\hat{\mathbf{t}}_{\mathbf{C}_{ij}} = t_1 \mathbf{S}_u + t_2 \mathbf{S}_v$  von  $\mathbf{C}_{ij}$  in der durch  $\mathbf{S}_u$  und  $\mathbf{S}_v$  gegebenen Basis der Tangentialebene in der Ecke  $\mathbf{V}_i$ .

Wie zur Definition der lokalen Repräsentation einer Kurve des  $G^1$ -Kurvennetzes verwenden wir auch hier die Darstellung einer funktionalen Fläche  $S : \mathbb{R}^2 \rightarrow \mathbb{R}$  in der üblichen parametrischen Darstellung (vgl. Gleichung (4.1)):

$$\mathbf{S}(u, v) = \begin{pmatrix} u \\ v \\ S(u, v) \end{pmatrix}.$$

Analog zu den partiellen Ableitungen und dem Normalenvektor der Fläche  $\mathbf{S}$  (vgl. Gleichungen (4.2) – (4.4)) ergibt sich die Zweite Fundamentalform zu:

$$\mathbb{I}_{\mathbf{S}} = \begin{pmatrix} \langle \mathbf{S}_{uu} \mid \hat{\mathbf{n}}_{\mathbf{S}} \rangle & \langle \mathbf{S}_{uv} \mid \hat{\mathbf{n}}_{\mathbf{S}} \rangle \\ \langle \mathbf{S}_{uv} \mid \hat{\mathbf{n}}_{\mathbf{S}} \rangle & \langle \mathbf{S}_{vv} \mid \hat{\mathbf{n}}_{\mathbf{S}} \rangle \end{pmatrix} = \frac{1}{\sqrt{1 + S_u^2 + S_v^2}} \begin{pmatrix} S_{uu} & S_{uv} \\ S_{uv} & S_{vv} \end{pmatrix}. \quad (5.2)$$

Der Krümmungsvektor  $\vec{\mathbf{k}}_{\mathbf{C}_{ij}}$  der Kurve  $\mathbf{C}_{ij}$  in  $\mathbf{V}_i$  ergibt sich nach Abschnitt 1.2 auf Seite 6 zu  $\vec{\mathbf{k}}_{\mathbf{C}_{ij}} = (\mathbf{C}'_{ij} \times \mathbf{C}''_{ij} \times \mathbf{C}'_{ij}) / \|\mathbf{C}'_{ij}\|^4$ . Aufgrund von  $(\vec{\mathbf{a}} \times \vec{\mathbf{b}}) \times \vec{\mathbf{c}} = \langle \vec{\mathbf{a}} \mid \vec{\mathbf{c}} \rangle \vec{\mathbf{b}} - \langle \vec{\mathbf{b}} \mid \vec{\mathbf{c}} \rangle \vec{\mathbf{a}}$  und der Koordinatendarstellung  $(a_{ij}, b_{ij})$  von  $\mathbf{C}'_{ij}(0)$  bzgl. der Basis  $\{(1, 0, S_u)^t, (0, 1, S_v)^t\}$  der Tangentialebene kann Gleichung (5.1) auf die folgende äquivalente Form gebracht werden:

$$\begin{aligned} \frac{\langle \mathbf{C}''_{ij}(0) \mid \hat{\mathbf{n}}_{\mathbf{S}}(\mathbf{V}_i) \rangle}{\|\mathbf{C}'_{ij}\|^2} &= \frac{1}{\sqrt{1 + S_{ui}^2 + S_{vi}^2}} (t_1, t_2) \begin{pmatrix} S_{uui} & S_{uvi} \\ S_{uvi} & S_{vvi} \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} &\iff \\ \langle \mathbf{C}''_{ij}(0) \mid \begin{pmatrix} -S_{ui} \\ -S_{vi} \\ 1 \end{pmatrix} \rangle &= (a_{ij}, b_{ij}) \begin{pmatrix} S_{uui} & S_{uvi} \\ S_{uvi} & S_{vvi} \end{pmatrix} \begin{pmatrix} a_{ij} \\ b_{ij} \end{pmatrix}. \end{aligned} \quad (5.3)$$

Wir setzen die zweite Ableitung der Kurve  $\mathbf{C}_{ij}$  in der Ecke  $\mathbf{V}_i$  wie folgt an:

$$\mathbf{C}''_{ij}(0) = \vec{\mathbf{s}}_{ij} + \begin{pmatrix} c_{ij} S_{ui} + d_{ij} S_{vi} + (a_{ij}, b_{ij}) \begin{pmatrix} S_{uui} & S_{uvi} \\ S_{uvi} & S_{vvi} \end{pmatrix} \begin{pmatrix} a_{ij} \\ b_{ij} \end{pmatrix} \end{pmatrix} \hat{\mathbf{w}}_i, \quad (5.4)$$

mit der zusätzlichen *lokalen Parameterrichtung*  $\vec{\mathbf{s}}_{ij} = c_{ij} \hat{\mathbf{u}}_i + d_{ij} \hat{\mathbf{v}}_i$ .

Die Definition (5.4) der zweiten Ableitung der Kurve  $\mathbf{C}_{ij}$  erfüllt tatsächlich die  $G^2$ -Kompatibilitätsbedingungen, denn die Betrachtung von (5.3) bzgl. des  $\{\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i, \hat{\mathbf{w}}_i\}$ -Koordinatensystems ergibt:

$$\begin{aligned} \langle \mathbf{C}_{ij}''(0) \mid \begin{pmatrix} -S_{ui} \\ -S_{vi} \\ 1 \end{pmatrix} \rangle &= \left\langle \begin{pmatrix} c_{ij} \\ d_{ij} \\ c_{ij}S_{ui} + d_{ij}S_{vi} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ (a_{ij}, b_{ij}) \begin{pmatrix} S_{uui} & S_{uvi} \\ S_{uvi} & S_{vvi} \end{pmatrix} \begin{pmatrix} a_{ij} \\ b_{ij} \end{pmatrix} \right\rangle \left| \begin{pmatrix} -S_{ui} \\ -S_{vi} \\ 1 \end{pmatrix} \right\rangle \\ &= (a_{ij}, b_{ij}) \begin{pmatrix} S_{uui} & S_{uvi} \\ S_{uvi} & S_{vvi} \end{pmatrix} \begin{pmatrix} a_{ij} \\ b_{ij} \end{pmatrix}. \end{aligned}$$

Wie Gleichung (5.4) zeigt, hängt  $\mathbf{C}_{ij}''(0)$  linear von den zusätzlichen partiellen Ableitungen  $S_{uui}, S_{uvi}, S_{vvi}$  in der Ecke  $\mathbf{V}_i$  ab. Die Kurve  $\mathbf{C}_{ij}$  des  $G^2$ -Kurvennetzes wird als quintischer Interpolant angesetzt, wobei die schon für das  $G^1$ -Kurvennetz eingesetzte, symmetrische Notation verwendet wird:

$$\begin{aligned} \mathbf{C}_{ij}(u) &= \mathbf{V}_i H_0^5(u) + \mathbf{S}_{\vec{\mathbf{t}}_{ij}} H_1^5(u) + \mathbf{C}_{ij}''(0) H_2^5(u) + \\ &\quad \mathbf{C}_{ji}''(0) H_2^5(1-u) + \mathbf{S}_{\vec{\mathbf{t}}_{ji}} H_1^5(1-u) + \mathbf{V}_j H_0^5(1-u). \end{aligned} \quad (5.5)$$

Die Richtungsableitungen  $\mathbf{S}_{\vec{\mathbf{t}}_{ij}}$  in der Fläche  $\mathbf{S}$  in den lokalen Koordinaten  $\{\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i, \hat{\mathbf{w}}_i\}$  entspricht der für das  $G^1$ -Kurvennetz eingeführten Darstellung (vgl. Gleichung (4.5) auf Seite 54). Die quintischen Hermite-Polynome  $H_i^5, i = 0, 1, 2$ , haben folgende Gestalt:

$$\begin{aligned} H_0^5(u) &= -6u^5 + 15u^4 - 10u^3 + 1, & H_1^5(u) &= -3u^5 + 8u^4 - 6u^3 + u, \\ H_2^5(u) &= \frac{-1}{2}u^5 + \frac{3}{2}u^4 - \frac{3}{2}u^3 + \frac{1}{2}u^2. \end{aligned}$$

Wie Eingangs erwähnt, werden neben den Kurven  $\mathbf{C}_{ij}$  noch Querableitungsvektorfelder  $\vec{\mathbf{q}}_{ij}$  für jede Kante in die Beschreibung des  $G^2$ -Kurvennetzes eingebracht.  $\vec{\mathbf{q}}_{ij}$  muß natürlich in den Ecken  $\mathbf{V}_i, \mathbf{V}_j$  mit den Tangentialebenen und der zweiten Fundamentalform kompatibel sein. Um dies zu erreichen, definieren wir

$$\vec{\mathbf{q}}_{ij}(0) = \vec{\mathbf{r}}_{ij} + (e_{ij}S_{ui} + f_{ij}S_{vi}) \hat{\mathbf{w}}_i, \quad (5.6)$$

$$\frac{\partial \vec{\mathbf{q}}_{ij}}{\partial \vec{\mathbf{t}}_{ij}}(0) = \left( (a_{ij}, b_{ij}) \begin{pmatrix} S_{uui} & S_{uvi} \\ S_{uvi} & S_{vvi} \end{pmatrix} \begin{pmatrix} e_{ij} \\ f_{ij} \end{pmatrix} \right) \hat{\mathbf{w}}_i. \quad (5.7)$$

mit  $\vec{\mathbf{r}}_{ij} = e_{ij}\hat{\mathbf{u}}_i + f_{ij}\hat{\mathbf{v}}_i$  als weiterer lokaler Parameterrichtung.

$\vec{\mathbf{t}}_{ij} = a_{ij}\hat{\mathbf{u}}_i + b_{ij}\hat{\mathbf{v}}_i$  entspricht der lokalen Parameterrichtung der Kurve  $\mathbf{C}_{ij}$  in der Ecke  $\mathbf{V}_i$  (vgl. Gleichung (4.5)). Gleichung (5.6) garantiert, daß das Querableitungsvektorfeld  $\vec{\mathbf{q}}_{ij}$  bei  $\mathbf{V}_i$  in der entsprechenden Tangentialebene liegt und mit (5.7) ist die Kompatibilität mit der zweiten Fundamentalform in  $\mathbf{V}_i$  gewährleistet. Das Querableitungsvektorfeld  $\vec{\mathbf{q}}_{ij}$  wird als kubischer Hermite-Interpolant angesetzt.

Zusätzlich zu den lokalen Parametern  $a_{ij}, b_{ij}$ , die die lokale Parameterrichtung  $\vec{\mathbf{t}}_{ij}$  festlegen, kommen für das  $G^2$ -Kurvennetz noch die lokale Parameter  $c_{ij}, d_{ij}, e_{ij}$  und  $f_{ij}$  hinzu. Wie diese bestimmt werden, wird im nächsten Abschnitt beschrieben.

### 5.1.2 Bestimmung der lokalen Parameter

Wie im Falle des  $G^1$ -Kurvennetzes müssen alle lokalen Parameter durch heuristische Methoden bestimmt werden, bevor eine Optimierung der partiellen Ableitungen  $S_{ui}, \dots, S_{vvi}$  durchgeführt werden kann. Die Parameter  $a_{ij}$  und  $b_{ij}$  werden mit den bereits in Abschnitt 4.1.2 beschriebenen Verfahren bestimmt. Im folgenden werden Methoden angegeben, mit denen die zusätzlichen lokalen Parameter ermittelt werden.

Betrachten wir zunächst die lokalen Parameter  $c_{ij}$  und  $d_{ij}$ , die dem  $\hat{\mathbf{u}}_i$ - bzw. dem  $\hat{\mathbf{v}}_i$ -Anteil von  $\mathbf{C}_{ij}''(0)$  in den lokalen Koordinaten  $\{\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i, \hat{\mathbf{w}}_i\}$  entsprechen.

**Nullsetzen (N):** Eine einfache Möglichkeit ist die Wahl:  $c_{ij} = d_{ij} = 0$ . Durch Nullsetzen von  $c_{ij}, d_{ij}$  wird zudem erreicht, daß für  $S_{ui} = S_{vi} = 0$ , also z.B. im Konvergenzfall des iterativen Optimierungsprozesses, die Kurven- und die Flächennormale in der Ecke  $\mathbf{V}_i$  übereinstimmen. Damit ist  $\mathbf{C}_{ij}$  lokal geodätisch in  $\mathbf{V}_i$  bzgl. der resultierenden Interpolationsfläche.<sup>1</sup>

**Kubische Kurve (KK):** Nach Definition der zweiten Ableitung (5.4) entspricht  $c_{ij}\hat{\mathbf{u}}_i + d_{ij}\hat{\mathbf{v}}_i$  dem  $(\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i)$ -Anteil von  $\mathbf{C}_{ij}''(0)$ . Wir bestimmen die durch die Ecken  $\mathbf{V}_i$  und  $\mathbf{V}_j$  sowie den Parameterrichtungen  $\vec{\mathbf{t}}_{ij}$  und  $\vec{\mathbf{t}}_{ji}$  eindeutige kubische Kurve  $\tilde{\mathbf{C}}_{ij}$ .  $\vec{\mathbf{s}}_{ij}$  wird durch Projizieren von  $\tilde{\mathbf{C}}_{ij}''(0)$  auf die lokale Parameterebene ermittelt (vgl. Abbildung 5.1).

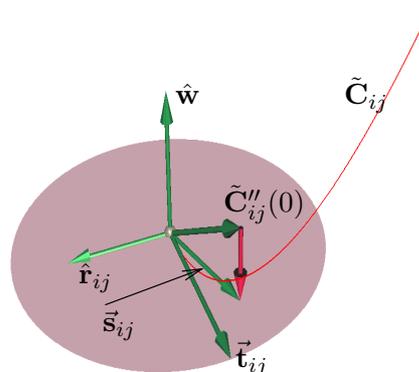


Abbildung 5.1: Bestimmung der lokalen Parameterrichtung  $\vec{\mathbf{s}}_{ij}$  (KK-Methode) und der lokalen Parameterrichtung  $\vec{\mathbf{r}}_{ij}$ .

Die Parameter  $e_{ij}$  und  $f_{ij}$ , die in die Definition des Querableitungsvektorfeldes  $\vec{\mathbf{q}}_{ij}$  eingehen, werden folgendermaßen berechnet: Der Vektor  $\hat{\mathbf{r}}_{ij} = e_{ij}\hat{\mathbf{u}}_i + f_{ij}\hat{\mathbf{v}}_i$  wird orthogonal zu der entsprechenden Parameterrichtung  $\vec{\mathbf{t}}_{ij}$  gewählt und anschließend normiert.

Es sei darauf hingewiesen, daß natürlich auf eine geeignete Orientierung zu achten ist, d.h.  $\vec{\mathbf{r}}_{ij}$  und  $\vec{\mathbf{r}}_{ji}$  müssen „auf dieselbe Seite“ der Kante  $\overline{\mathbf{V}_i\mathbf{V}_j}$  zeigen.

### 5.1.3 Optimierung des $G^2$ -Kurvennetzes

Die Optimierung des  $G^2$ -Kurvennetzes beruht auf ähnlichen Verfahren wie die des  $G^1$ -Kurvennetz. Die eigentliche Besonderheit liegt in der Berücksichtigung der Querableitungsvektorfelder  $\vec{\mathbf{q}}_{ij}$ .

Für die Optimierung des  $G^2$ -Kurvennetzes setzen wir ein Funktional an, daß sowohl die

<sup>1</sup>Eine Kurve  $\mathbf{C}$  auf der Fläche  $\mathbf{S}$  heißt im Punkt  $\mathbf{C}(t_0) = \mathbf{S}(u_0, v_0)$  geodätisch, falls hier Kurven- und Flächennormale übereinstimmen; damit entspricht die Kurvenkrümmung genau der Normalenkrümmung der Fläche in Richtung der Tangente  $\mathbf{C}'(t_0)$ .

quintischen Kurven  $\mathbf{C}_{ij}$  als auch die Querableitungsvektorfelder  $\vec{\mathbf{q}}_{ij}$  bewertet. Zur Bewertung der Kurven  $\mathbf{C}_{ij}$  wird sowohl das einfache Funktional wie in (4.7) auf Seite 56, als auch die datenabhängigen Laplace-Beltrami und das Funktional dritter Ordnung verwendet (vgl. Gleichungen (4.8) und (4.9)). Bei Verwendung datenabhängiger Funktionale wird die entsprechende Referenzkurve  $\mathbf{C}_{ij}^0$  aus der Repräsentation des Kurvennetzes für  $S_{ui} = \dots = S_{vvi} = 0$  gewonnen.

Zur Bewertung der Querableitungsvektorfelders betrachten wir zunächst das einfache Funktional. Wie die Kurve  $\mathbf{C}_{ij}$  wird auch das Vektorfeld  $\vec{\mathbf{q}}_{ij}$  über dem Intervall  $[0, \|\mathbf{V}_i \mathbf{V}_j\|]$  parametrisiert:  $\vec{q}_{ij}(s) = \vec{\mathbf{q}}_{ij} \left( \frac{s}{\|\mathbf{V}_i \mathbf{V}_j\|} \right)$ . Die Bewertung des Netzwerks der Querableitungsvektorfelder  $\vec{\mathbf{Q}}$  erfolgt nun durch

$$\begin{aligned} \sigma(\vec{\mathbf{Q}}) &= \sum_{ij \in E} \int_{[0, \|\mathbf{V}_i \mathbf{V}_j\|]} \left\| \vec{q}_{ij}''(s) \right\|^2 + \alpha \left\| \vec{q}_{ij}'(s) \right\|^2 + \beta \left\| \vec{q}_{ij}(s) \right\|^2 ds, \quad \alpha, \beta \geq 0 \quad (5.8) \\ &= \sum_{ij \in E} \int_0^1 \frac{\left\| \vec{\mathbf{q}}_{ij}''(u) \right\|^2}{\|\mathbf{V}_i \mathbf{V}_j\|^3} + \alpha \frac{\left\| \vec{\mathbf{q}}_{ij}'(u) \right\|^2}{\|\mathbf{V}_i \mathbf{V}_j\|} + \beta \left\| \vec{\mathbf{q}}_{ij}(u) \right\|^2 \|\mathbf{V}_i \mathbf{V}_j\| du, \quad \alpha, \beta \geq 0. \end{aligned}$$

Für das Laplace-Beltrami Funktional und das Funktional dritter Ordnung werden die Querableitungsvektorfelder  $\vec{\mathbf{q}}_{ij}$  geometrieabhängig bewertet. Betrachten wir hierzu folgende Darstellung des Laplace-Beltrami Operators der Kurve  $\mathbf{C}$  bzgl. der Referenzkurve  $\mathbf{C}^0$ :

$$\Delta_{\mathbf{C}^0}(\mathbf{C})(u) = \frac{d}{ds} \left( \frac{\mathbf{C}'(u)}{\|(\mathbf{C}^0)'(u)\|} \right),$$

wobei  $s$  die Bogenlänge der Referenzkurve  $\mathbf{C}^0$  ist. Dies motiviert die folgende Definition eines Operators für das Querableitungsvektorfeld  $\vec{\mathbf{q}}$ :

$$L_{\vec{\mathbf{q}}^0}(\vec{\mathbf{q}})(u) = \frac{d}{ds} \left( \frac{\vec{\mathbf{q}}(u)}{\|\vec{\mathbf{q}}^0(u)\|} \right). \quad (5.9)$$

Umformuliert ergibt sich der Operator  $L_{\vec{\mathbf{q}}^0}$  zu:

$$L_{\vec{\mathbf{q}}^0}(\vec{\mathbf{q}}) = \frac{d}{du} \left( \frac{\vec{\mathbf{q}}(u)}{\|\vec{\mathbf{q}}^0(u)\|} \right) \cdot \frac{du}{ds} = \left( \frac{\vec{\mathbf{q}}'}{\|\vec{\mathbf{q}}^0\|} - \frac{\langle \vec{\mathbf{q}}^0 | (\vec{\mathbf{q}}^0)' \rangle \vec{\mathbf{q}}}{\|\vec{\mathbf{q}}^0\|^3} \right) \cdot \|(\mathbf{C}^0)'\|^{-1}. \quad (5.10)$$

Zur Bewertung von  $\vec{\mathbf{Q}}$  wird für jedes Querableitungsvektorfeld  $\vec{\mathbf{q}}_{ij}$  ein Funktional basierend auf dem Operator  $L_{\vec{\mathbf{q}}_{ij}^0}(\vec{\mathbf{q}}_{ij})$  angesetzt. Damit ergibt sich das folgende Funktional:

$$\begin{aligned} \sigma(\vec{\mathbf{Q}}) &= \sum_{ij \in E} \int \left( \frac{d}{ds} \left( \frac{\vec{\mathbf{q}}_{ij}(u)}{\|\vec{\mathbf{q}}_{ij}(u)\|} \right) \right)^2 ds \\ &= \sum_{ij \in E} \int \frac{1}{\|(\mathbf{C}_{ij}^0)'(u)\|^2} \left( \frac{\vec{\mathbf{q}}_{ij}'(u)}{\|\vec{\mathbf{q}}_{ij}^0(u)\|} - \frac{\langle \vec{\mathbf{q}}_{ij}^0(u) | (\vec{\mathbf{q}}_{ij}^0(u))' \rangle}{\|\vec{\mathbf{q}}_{ij}^0(u)\|^3} \right)^2 ds, \end{aligned}$$

wobei hier nach der Bogenlänge der Referenzkurve  $\mathbf{C}_{ij}^0$  integriert wird.

Bezeichnet  $\sigma_N(\mathbf{N})$  das Funktional zur Bewertung der Kurven  $\mathbf{C}_{ij}$ ,  $ij \in E$  und  $\sigma_Q(\vec{\mathbf{Q}})$  das Funktional zur Bewertung der Querableitungsvektorfelder, so ergibt sich das Gesamtfunktional zu

$$\sigma = \sigma_N + \gamma\sigma_Q \quad \text{mit} \quad \gamma > 0.$$

Die Bedingung  $\gamma > 0$  ist notwendig um die Lösbarkeit des dem Optimierungsproblem zugeordneten Gleichungssystems zu garantieren (vgl. Anhang A.2.2 auf Seite 122).

Wie für das  $G^1$ -Kurvennetz wird auch die Optimierung des  $G^2$ -Kurvennetzes iteriert. Die zweiten partiellen Ableitungen  $S_{uui}$ ,  $S_{uvi}$  und  $S_{vvi}$  werden bei jedem einzelnen Optimierungsschritt auf 0 initialisiert.

### 5.1.4 Ergebnisse

Abschließend werden die beiden Methoden zur Festlegung der lokalen Parameterrichtung  $\vec{s}_{ij}$  kurz miteinander verglichen. Außerdem werden, z.T. anhand des Beispiels aus Abschnitt 4.1.4, die Unterschiede zwischen dem  $G^1$ - und dem  $G^2$ -Kurvennetz erläutert.

Überraschenderweise ergeben sich bei Anwendung der KK-Methode zur Bestimmung der lokalen Parameterrichtung  $\vec{s}_{ij}$  fast immer schlechtere Ergebnisse als bei der N-Methode. Abbildung 5.2 zeigt die resultierenden Kurven bei Anwendung auf das unsymmetrische Beispiel aus Abschnitt 4.1.4.<sup>2</sup>Zwar ist das Kontrollpolygon bei Anwendung der N-Methode unregelmäßiger als für bei der KK-Methode, dennoch weist der Krümmungsplot ein günstigeres Verhalten für die N-Methode auf.

Um die  $G^2$ -Kurvennetzmethode mit der  $G^1$ -Kurvennetzmethode zu vergleichen, verwenden wir das Beispiel der vier Punkte auf dem Einheitskreis aus Abschnitt 4.1.4. Hierbei betrachten wir zunächst die Initialisierung ohne weitere Optimierung und die Ergebnisse des jeweils einmalig optimierten Kurvennetzes mit dem einfachen Funktional, dem Laplace-Beltrami Funktional und dem Funktional dritter Ordnung.

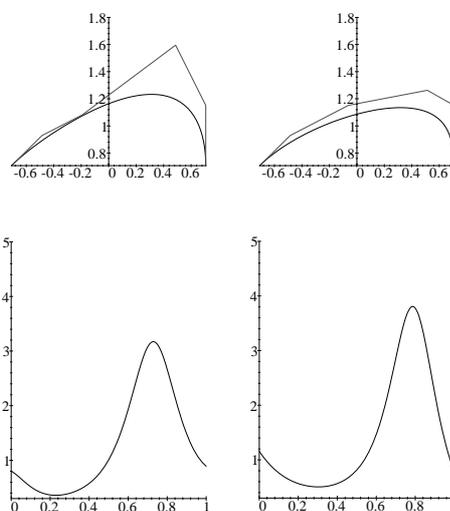


Abbildung 5.2: Die Anwendung der N-Methode (links) und der KK-Methode (rechts) auf das unsymmetrische Beispiel aus Abschnitt 4.1.4, jeweils die Kurve (oben) mit zugehörigem Krümmungsplot (unten).

<sup>2</sup>Hierbei wurde die Kurve einmal mit dem einfachen Funktional optimiert, damit die Unterschiede zu Geltung kommen. Ohne Optimierung ist  $S_{uui} = S_{uvi} = S_{vvi} = 0$ , wodurch die Parameterrichtung  $\vec{s}_{ij}$  ohne Einfluß bleibt.

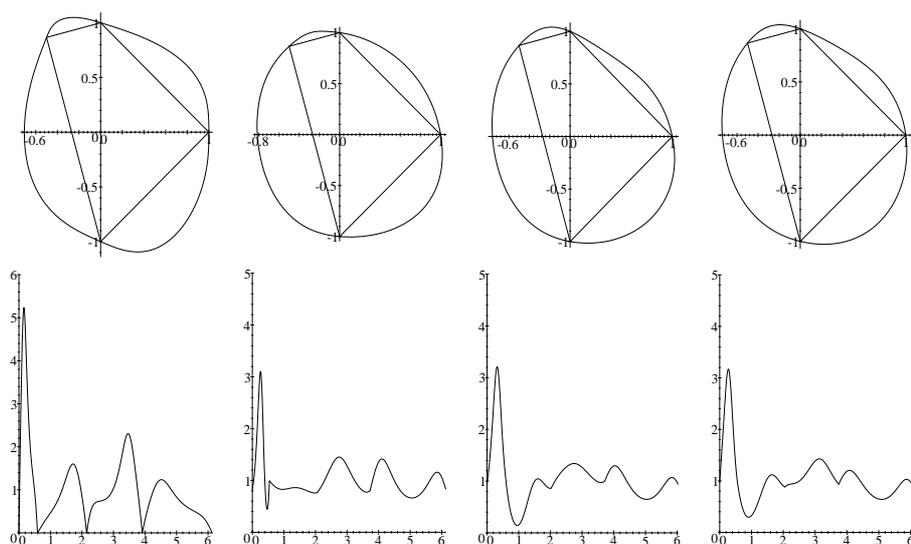


Abbildung 5.3: Die  $G^2$ -Kurvennetze (oben) mit den zugehörigen Krümmungsplots (unten) für die verschiedenen Funktionale nach einmaliger Optimierung bei den Kreisdaten. Links die Initialisierung, Mitte-links das einfache Funktional, Mitte-rechts das Laplace-Beltrami Funktional und rechts das Funktional dritter Ordnung.

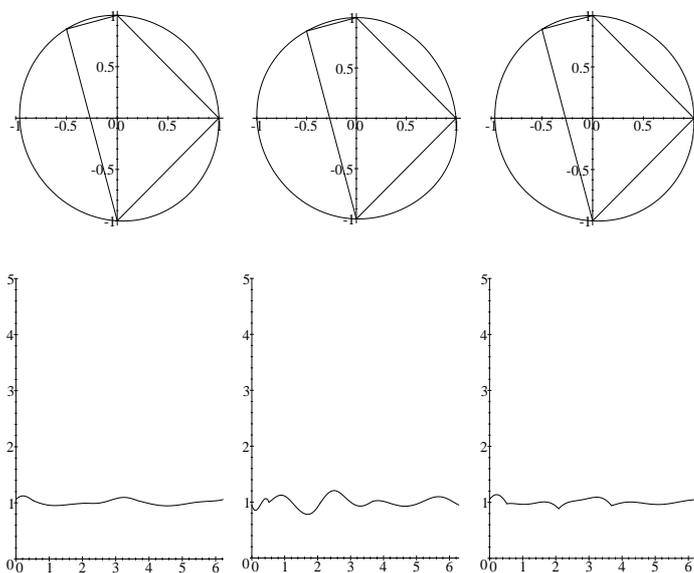


Abbildung 5.4: Die  $G^2$ -Kurvennetze (oben) mit den zugehörigen Krümmungsplots (unten) für die verschiedenen Funktionale nach bis zur Konvergenz iterierter Optimierung bei den Kreisdaten. Links das einfache Funktional, in der Mitte das Laplace-Beltrami Funktional und rechts das Funktional dritter Ordnung.

In allen Fällen werden die Tangentenlängen mit der KS-Methode bestimmt. Die Parameterrichtungen  $\vec{s}_{ij}$  werden mit der N-Methode ermittelt.

Die Initialisierung liefert zunächst eine Kurve mit schlechtem Krümmungsverhalten, da hier wegen  $S_{uui} = S_{uvi} = S_{vvi} = 0$  für jede Kurve  $\mathbf{C}_{ij}$  die Krümmung in den Endpunkten verschwindet (vgl. Abbildung 5.3).

In Abbildung 5.4 sind die Ergebnisse der bis zur Konvergenz durchgeführten Optimierung angegeben. Im Vergleich zu den optimierten  $G^1$ -Kurvennetzen (vgl. Abbildung 4.7 auf Seite 61) erkennt man ein verbessertes Krümmungsverhalten; die resultierenden Kurven approximieren den Einheitskreis sehr gut.

Die Unterschiede bei Verwendung der verschiedenen Funktionale zur Optimierung des  $G^2$ -Kurvennetzes entsprechen denen beim  $G^1$ -Kurvennetz. Deshalb sei hier auf die Ausführungen in Abschnitt 4.1.4 verwiesen.

Im Vergleich mit dem  $G^1$ -Kurvennetz ergeben sich bei Anwendung der  $G^2$ -Kurvennetze, von wenigen Ausnahmen abgesehen, Kurvennetze mit günstigerem Krümmungsverhalten. Insbesondere zeigt sich, daß sich das  $G^2$ -Kurvennetz, aufgrund der Einbindung der Querableitungen, wesentlich stabiler verhält als das  $G^1$ -Kurvennetz. Dies wird vor allem dann deutlich, wenn das Kurvennetz durch die Geometrie des zugrundeliegenden Polyeders fast nicht mehr winkel-regulär oder ordnungserhaltend ist. (vgl. Bezeichnung 2.2.1 und die Beispiele in Abschnitt 5.4).

## 5.2 Ausfüllen des $G^2$ -Kurvennetzes

Im Folgenden wird beschrieben, wie das optimierte  $G^2$ -Kurvennetz unter Verwendung von einer Dreiecks-Bézierfläche vom Grad 7 für jede Öffnung des Kurvennetzes zu einer glatten Fläche fortgesetzt werden kann. Hierbei wird neben den Randkurven  $\mathbf{C}_{ij}$  des Kurvennetzes auch das Querableitungsvektorfeld  $\vec{\mathbf{q}}_{ij}$  interpoliert.

Sei  $\mathbf{S}$  die Fläche, die für die Dreiecksseite  $\Delta(\mathbf{V}_i, \mathbf{V}_j, \mathbf{V}_k)$  des Polyeders angesetzt wird und über den Kanten  $\overline{\mathbf{V}_i\mathbf{V}_j}$ ,  $\overline{\mathbf{V}_j\mathbf{V}_k}$  und  $\overline{\mathbf{V}_k\mathbf{V}_i}$  die durch das Kurvennetz und die Querableitungsvektorfelder gegebenen Informationen interpolieren soll. Die Kontrollpunkte von  $\mathbf{S}$  bezeichnen wir mit  $\mathbf{b}_I^7$ ,  $I = (i_1, i_2, i_3)$ ,  $i_1 + i_2 + i_3 = 7$ .

Sei die Fläche  $\mathbf{S}$  über dem Dreieck  $\Delta(\mathbf{r}, \mathbf{s}, \mathbf{t})$  parametrisiert. Die durch die Kanten des Parameterdreiecks gegebenen Richtungen seien bezeichnet durch:  $\vec{\mathbf{u}}_r = \mathbf{s} - \mathbf{r}$ ,  $\vec{\mathbf{v}}_r = \mathbf{t} - \mathbf{r}$ . Damit ergibt sich die Querableitung der Fläche  $\mathbf{S}$  entlang der Kante  $\overline{\mathbf{rs}}$  zu:

$$\mathbf{S}_{\vec{\mathbf{v}}_r}(u) = \sum_{i=0}^6 (\mathbf{b}_{6-k,k,1}^7 - \mathbf{b}_{7-k,k,0}^7) B_i^6(u).$$

Um die Interpolation der durch das Tangentialfeld der Kurve  $\mathbf{C}_{ij}$  und durch das Querableitungsvektorfeld  $\vec{\mathbf{q}}_{ij}$  gegebenen Tangentialebene zu garantieren, wird ein Ansatz ähnlich zu dem von Chiyokura und Kimura gemacht (vgl. Abschnitt 1.2 auf Seite 11):

$$\mathbf{S}_{\vec{\mathbf{v}}_r}(u) = \alpha_{ij} \mathbf{C}'_{ij}(u) + \beta_{ij} \vec{\mathbf{q}}_{ij}(u). \quad (5.11)$$

Analoge Ansätze für  $\mathbf{S}_{\vec{\mathbf{v}}_s}$  und  $\mathbf{S}_{\vec{\mathbf{v}}_t}$  entlang der Kante  $\overline{\mathbf{st}}$  bzw.  $\overline{\mathbf{tr}}$  stellen sicher, daß die Querableitung von  $\mathbf{S}$  in der jeweiligen Tangentialebene liegt. Die Gewichtspolynome  $\alpha_{ij}, \beta_{ij}$

werden quadratisch bzw. kubisch gewählt. Es sei angemerkt, daß die Regularität des (Normalen-) Vektorfeldes  $\mathbf{C}'_{ij} \times \vec{\mathbf{q}}_{ij}$  entlang  $\overline{\mathbf{rs}}$  nicht per Konstruktion garantiert werden kann. In allen bisherigen Tests wurde allerdings kein einziger Fall beobachtet, in dem  $\mathbf{C}'_{ij} \times \vec{\mathbf{q}}_{ij}$  singular geworden ist.

Um alle drei Bedingungen der Form (5.11) für  $\mathbf{S}$  erfüllen zu können, müssen die Gewichtspolynome so gewählt werden, daß die Tangentialbedingungen und die sogenannten *Twistbedingungen* in den Ecken erfüllt werden. Die Tangentialbedingung für die Kante  $\overline{\mathbf{rs}}$  ergibt sich zu:

$$(\mathbf{b}_{601}^7 - \mathbf{b}_{700}^7) = \alpha_{ij,0} \mathbf{C}'_{ij,0} + \beta_{ij,0} \vec{\mathbf{q}}_{ij,0}, \quad (5.12)$$

wobei die Ableitung von  $\mathbf{C}_{ij}$  durch

$$\mathbf{C}'_{ij}(u) = \sum_{k=0}^4 \mathbf{C}'_{ij,k} B_k^4(u)$$

gegeben ist. Die Twistbedingung der entsprechenden Kante wird wie folgt formuliert (vgl. Abbildung 5.5):

$$\mathbf{b}_{610}^7 + (\mathbf{b}_{511}^7 - \mathbf{b}_{610}^7) = \mathbf{b}_{502}^7 + (\mathbf{b}_{511}^7 - \mathbf{b}_{502}^7). \quad (5.13)$$

Da wir ein  $G^2$ -Kurvennetz vorliegen haben, sind die Twistbedingungen stets erfüllbar.

Die Normalenkomponente von  $\mathbf{b}_{511}^7$  ist dadurch bestimmt, daß die gemischte Ableitung der Fläche  $\mathbf{S}$  in  $\mathbf{r}$  mit der zweiten Fundamentalform kompatibel gewählt wird. Zur Festlegung der tangentialen Komponente von  $\mathbf{b}_{511}^7$  werden die Unbekannten  $\beta_{ij,1}$  und  $\beta_{ki,2}$  so bestimmt, daß Gleichung (5.13) erfüllt wird.

Konkret bedeutet dies, daß die sich aus Gleichung (5.11) ergebende Darstellung von  $\mathbf{b}_{511}^7 - \mathbf{b}_{610}^7$ :

$$6(\mathbf{b}_{511}^7 - \mathbf{b}_{610}^7) = 3(\beta_{ij,0} \vec{\mathbf{q}}_{ij,1} + \beta_{ij,1} \vec{\mathbf{q}}_{ij,0}) + 4\alpha_{ij,0} \mathbf{C}'_{ij,1} + 2\alpha_{ij,1} \mathbf{C}'_{ij,0} \quad (5.14)$$

und der analog Ausdruck für  $\mathbf{b}_{511}^7 - \mathbf{b}_{502}^7$ :

$$6(\mathbf{b}_{511}^7 - \mathbf{b}_{502}^7) = 3(\beta_{ki,3} \vec{\mathbf{q}}_{ki,2} + \beta_{ki,2} \vec{\mathbf{q}}_{ki,3}) + 4\alpha_{ij,2} \mathbf{C}'_{ij,3} + 2\alpha_{ij,1} \mathbf{C}'_{ij,4} \quad (5.15)$$

in die Gleichung (5.13) eingefügt und nach den Unbekannten  $\beta_{ij,1}$  und  $\beta_{ki,2}$  aufgelöst wird. Für winkel-reguläre Kurvennetze (vgl. Bezeichnung 2.2.1 auf Seite 25) sind die Vektoren  $\vec{\mathbf{q}}_{ij,0}$  und  $\vec{\mathbf{q}}_{ki,3}$  immer linear unabhängig (vgl. Abschnitt 5.1.2). Somit können  $\beta_{ij,1}$  und  $\beta_{ki,2}$  immer so gewählt werden, daß Gleichung (5.13) erfüllt ist. Der bislang unbestimmte Kontrollpunkt  $\alpha_{ij,1}$  wird durch einfache Mittelung aus  $\alpha_{ij,0}$  und  $\alpha_{ij,2}$  berechnet.

Die Regularität von  $\mathbf{C}'_{ij} \times \vec{\mathbf{q}}_{ij}$  vorausgesetzt, muß bei geeigneter Orientierung von  $\vec{\mathbf{q}}_{ij}$   $\beta_{ij}(u) > 0, u \in [0, 1]$  gelten um eine reguläre Fläche und damit einen  $G^1$  Übergang zu

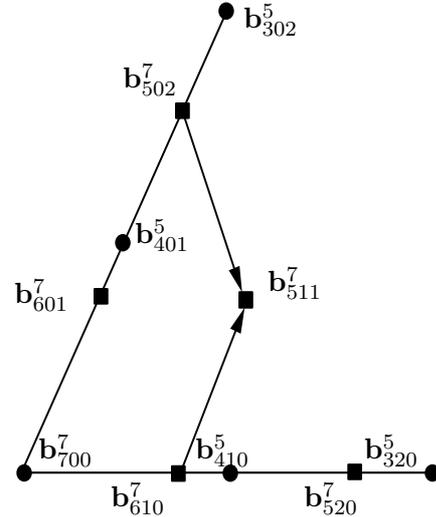


Abbildung 5.5: Die Twistbedingung: Der Twistpunkt  $\mathbf{b}_{511}^7$  geht in die Interpolationsbedingungen über zwei Kanten ein. Kontrollpunkte der Fläche  $\mathbf{S}$  vom Grad 7 sind durch ■, die Kontrollpunkte des Kurvennetzes mit ● gekennzeichnet.

erhalten. Außer in wenigen Ausnahmesituationen ist dies immer der Fall, obwohl diese Bedingung in der Bestimmung der Kontrollpunkte  $\beta_{ij,1}$  und  $\beta_{ki,2}$  nicht explizit berücksichtigt werden kann. Dennoch, i.a. kann hier nur Tangentialstetigkeit garantiert werden.

Sind die Gewichtsfunktionen  $\alpha_{ij}, \alpha_{jk}, \alpha_{ki}$  und  $\beta_{ij}, \beta_{jk}, \beta_{ki}$  ermittelt, so sind die Kontrollpunkte  $\mathbf{B}_i^7$  auf der ersten Reihe durch Gleichung (5.11) bestimmt.

Die Kontrollpunkte  $\mathbf{b}_{322}, \mathbf{b}_{232}$  und  $\mathbf{b}_{223}$  werden zunächst durch die Forderung

$$\mathbf{S}_{\bar{\mathbf{u}}_r, \bar{\mathbf{u}}_r, \bar{\mathbf{v}}_r, \bar{\mathbf{v}}_r}(\mathbf{r}) = 0,$$

fixiert.

In vielen Situationen ergibt sich bereits durch diese Festlegung der innersten Kontrollpunkte eine qualitativ gute Interpolationsfläche.

Wie schon bei der Konstruktion der  $3 \times 4$ -Flächen für das  $G^1$ -Kurvennetz können diese Parameter durch eine Optimierung eines quadratischen Funktionals bestimmt werden. Hierbei kann allerdings jede einzelne Bézierfläche für sich optimiert werden. Zur Optimierung werden alle in Abschnitt 4.2 beschriebenen Funktionale eingesetzt.

### 5.3 Simultane Optimierung

In Abschnitt 4.3 haben wir gesehen, daß für die  $G^1$ -Kurvennetzmethode die beiden Optimierungsschritte für das Kurvennetz und für die  $3 \times 4$  Splitflächen zu einem simultanen Optimierungsschritt zusammengefaßt werden können. Auf ähnliche Weise können auch im Fall des  $G^2$ -Kurvennetzes alle freien Kurven- und Flächenparameter unter Ansatz eines quadratischen Funktionals zur Bewertung der Fläche optimiert werden.

Bei der simultanen Optimierung des  $G^1$ -Kurvennetzes ist die Unabhängigkeit der angesetzten Gewichtsfunktionen für den Chiyokura-Kimura Ansatz von den zu optimierenden Parameter  $S_{ui}, \dots, S_{vvi}$  des Kurvennetzes entscheidend (vgl. die Ausführungen in Abschnitt 4.3).

Für das  $G^2$ -Kurvennetz werden die Kontrollpunkte  $\alpha_{ij,0}$  und  $\beta_{ij,0}$  der Gewichtspolynome durch die Tangentialbedingung (5.12) festgelegt. Aufgrund der Darstellung von  $\mathbf{C}'_{ik}(0)$ ,  $\mathbf{C}'_{ij}(0)$  und  $\bar{\mathbf{q}}_{ij}(0)$  in den lokalen Koordinaten ergibt sich die Unabhängigkeit von  $\alpha_{ij,0}$  und  $\beta_{ij,0}$  von den partiellen Ableitungen  $S_{ui}, S_{vvi}$  völlig analog wie beim  $G^1$ -Kurvennetz (vgl. Gleichung (4.13) auf Seite 66).

Um zu zeigen, daß die Bestimmung der inneren Kontrollpunkte  $\alpha_{ij,1}$  und  $\beta_{ij,1}$  der Gewichtspolynome ebenso unabhängig von den partiellen Ableitung bei  $\mathbf{V}_i$  ist, betrachten wir Gleichung (5.13). Demnach ergibt sich der Twistpunkt  $\mathbf{B}_{511}^7$  unter Verwendung von Gleichung (5.14) zu:

$$\mathbf{b}_{511}^7 = \mathbf{V}_i + \frac{1}{7}\mathbf{C}'_{ij,0} + \frac{1}{6} \left( 2\alpha_{ij,1}\mathbf{C}'_{ij,0} + 4\alpha_{ij,0}\mathbf{C}'_{ij,1} + 3(\beta_{ij,1}\bar{\mathbf{q}}_{ij,0} + \beta_{ij,0}\bar{\mathbf{q}}_{ij,1}) \right). \quad (5.16)$$

Unter Verwendung von

$$\begin{aligned}\mathbf{C}'_{ij}(0) &= 4(\mathbf{C}'_{ij,1} - \mathbf{C}'_{ij,0}) = s_{ij} + \left( c_{ij}S_{ui} + d_{ij}S_{vi} + (a_{ij}, b_{ij}) \begin{pmatrix} S_{uui} & S_{uvi} \\ S_{uvi} & S_{vvi} \end{pmatrix} \begin{pmatrix} a_{ij} \\ b_{ij} \end{pmatrix} \right) \hat{\mathbf{w}}_i, \\ \vec{\mathbf{q}}'_{ij}(0) &= 3(\vec{\mathbf{q}}_{ij,1} - \vec{\mathbf{q}}_{ij,0}) = \left( (a_{ij}, b_{ij}) \begin{pmatrix} S_{uui} & S_{uvi} \\ S_{uvi} & S_{vvi} \end{pmatrix} \begin{pmatrix} e_{ij} \\ f_{ij} \end{pmatrix} \right) \hat{\mathbf{w}}_i,\end{aligned}$$

kann (5.16) auf folgende äquivalente Form gebracht werden:

$$\mathbf{b}_{511}^7 = \mathbf{V}_i + \left( \frac{1}{7} + \frac{1}{3}\alpha_{ij,1} + \frac{2}{3}\alpha_{ij,0} \right) \mathbf{C}'_{ij,0} + \frac{1}{2}(\beta_{ij,0} + \beta_{ij,1}) \vec{\mathbf{q}}_{ij,0} + \frac{1}{6}\alpha_{ij,0} \mathbf{C}''_{ij}(0) + \frac{1}{6}\beta_{ij,0} \vec{\mathbf{q}}'_{ij}(0).$$

Fassen wir alle  $(\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i)$ -Komponenten zu  $k_i\hat{\mathbf{u}}_i + l_i\hat{\mathbf{v}}_i$  zusammen, so ergibt sich die folgende Darstellung in dem lokalen  $\{\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i, \hat{\mathbf{w}}_i\}$  Koordinatensystem:

$$\begin{aligned}\mathbf{b}_{511}^7 &= \begin{pmatrix} k_i \\ l_i \\ k_i S_{ui} + l_i S_{vi} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ (a_{ij}, b_{ij}) \begin{pmatrix} S_{uui} & S_{uvi} \\ S_{uvi} & S_{vvi} \end{pmatrix} \left( \frac{1}{6}\alpha_{ij,0} \begin{pmatrix} a_{ij} \\ b_{ij} \end{pmatrix} + \frac{1}{6}\beta_{ij,0} \begin{pmatrix} e_{ij} \\ f_{ij} \end{pmatrix} \right) \end{pmatrix} \\ &= \begin{pmatrix} k_i \\ l_i \\ k_i S_{ui} + l_i S_{vi} \end{pmatrix} + \frac{1}{42} \begin{pmatrix} 0 \\ 0 \\ (a_{ij}, b_{ij}) \begin{pmatrix} S_{uui} & S_{uvi} \\ S_{uvi} & S_{vvi} \end{pmatrix} \begin{pmatrix} a_{ik} \\ b_{ik} \end{pmatrix} \end{pmatrix}.\end{aligned}\quad (5.17)$$

Die letzte Gleichheit folgt aus Gleichung (5.12) und der Tatsache, daß die lokale Parameterrichtungen von  $\mathbf{C}'_{ij,0}$  bzw. von  $\mathbf{b}_{601} - \mathbf{b}_{700}$  durch  $(a_{ij}, b_{ij})^t$  bzw.  $\frac{1}{7}(a_{ik}, b_{ik})^t$  gegeben sind.

Durch analoge Vorgehensweise erhalten wir aus Gleichung (5.13) unter Verwendung von Gleichung (5.15) folgende Darstellung von  $\mathbf{b}_{511}^7$ :

$$\mathbf{b}_{511}^7 = \begin{pmatrix} \bar{k}_i \\ \bar{l}_i \\ \bar{k}_i S_{ui} + \bar{l}_i S_{vi} \end{pmatrix} + \frac{1}{42} \begin{pmatrix} 0 \\ 0 \\ (a_{ij}, b_{ij}) \begin{pmatrix} S_{uui} & S_{uvi} \\ S_{uvi} & S_{vvi} \end{pmatrix} \begin{pmatrix} a_{ik} \\ b_{ik} \end{pmatrix} \end{pmatrix}.$$

Da beide Darstellungen von  $\mathbf{b}_{511}^7$  für eine bestimmte Wahl von partiellen Ableitungen  $S_{ui}, \dots, S_{vvi}$  übereinstimmen, folgt:  $k_i = \bar{k}_i, l_i = \bar{l}_i$ . Somit gilt Gleichung (5.13) für jede beliebige Wahl von  $S_{ui}, \dots, S_{vvi}$ , die Kontrollpunkte  $\beta_{ij,1}, \beta_{ki,2}$  sind also unabhängig von diesen partiellen Ableitungen.

Die Kontrollpunkte  $\mathbf{b}_{601} - \mathbf{b}_{061}$  werden über Gleichung (5.11) bestimmt und hängen aufgrund der festen Gewichtspolynome linear von den Kurven  $\mathbf{C}_{ij}$  bzw. von den Querableitungsvektorfeldern  $\vec{\mathbf{q}}_{ij}$  des Kurvennetzes und damit auch von den partiellen Ableitungen  $S_{ui}, \dots, S_{vvi}$  ab.

Die in Abschnitt 4.4 gemachten Aussagen über die Unterschiede bei Verwendung der verschiedenen Funktionale, sowie bei Einsatz der Zweischnitt- bzw. der simultanen Optimierung, treffen auch für die  $G^2$ -Kurvennetzmethod zu.

## 5.4 Vergleich der parametrischen Interpolationsmethoden

In Abschnitt 2.2 wurden verschiedene bekannte Verfahren zur Lösung des parametrischen Interpolationsproblems beschrieben. In diesem Abschnitt werden einige dieser Verfahren mit den in den Kapiteln 4 und 5 eingeführten  $G^1$ - und  $G^2$ -Kurvennetzmethoden verglichen. Des Weiteren werden die  $G^1$ - und die  $G^2$ -Kurvennetzmethode auf einige Beispieldatensätze mit mehreren hundert Punkten angewendet.

Zum Vergleich der  $G^1$ - und der  $G^2$ -Kurvennetzmethode mit den bekannten Verfahren zur Lösung des parametrischen Interpolationsproblems werden folgende zwei Datensätze verwendet (vgl. Abbildung 5.6): Sphärendaten (14 Punkte; 24 Seitenflächen) und Torusdaten (100 Punkte; 200 Seitenflächen).

Abbildungen 5.8 und 5.10 zeigen die Interpolationsflächen, sowie die zugehörigen Gauß-Krümmungsplots, nach dem Verfahren von Moreton und Séquin, Shirman und Séquin und Nielson<sup>3</sup>, jeweils von links nach rechts. Wie zu erwarten liefert das Verfahren nach Moreton und Séquin eine sehr gute Approximation an die exakte Sphäre bzw. an den exakten Torus. Bei den Torusdaten wurde allerdings die Flächenoptimierung nach 30 Iterationen und einer Laufzeit von über 50 Stunden (!) abgebrochen<sup>4</sup>. Für die Shirman-Séquin Fläche sind in beiden Fällen sehr ungleichmäßige Krümmungsverteilung erkennbar. Die Nielson Fläche hingegen hat im Fall der Sphärendaten wesentlich größere Krümmungsschwankungen als im Fall der Torusdaten.

Abbildung 5.9 und 5.11 zeigt die Interpolationsflächen mit dem interpolatorischen Splineflächen nach Loop, sowie die Ergebnisse der  $G^1$ - und der  $G^2$ -Kurvennetzmethode (von links nach rechts).

Die Splineflächen nach Loop haben große Probleme mit der Unsymmetrie in den Daten, d.h. mit den eingefügten Diagonalkanten. Dies wird vor allem im Vergleich mit den „entschärften“ Polyedern deutlich. Abbildung 5.12 zeigt die entsprechenden Interpolationsflächen für die Sphären- und die Torusdaten ohne die Diagonalkanten. Bei allgemeiner Lage der Interpolationpunkte ist eine derartige „Entschärfung“ allerdings nicht möglich. Die  $G^1$ -Kurvennetzmethode, die als wesentlichen Bestandteil eine Variante des Shirman-Séquin Verfahrens beinhaltet, zeigt ein sehr viel besseres Krümmungsverhalten als die ursprüngliche Shirman-Séquin Methode. Die Güte der mit der  $G^2$ -Kurvennetzmethode generierten Interpolationsflächen kommen sehr nahe an die Ergebnisse von Moreton und Séquin heran.

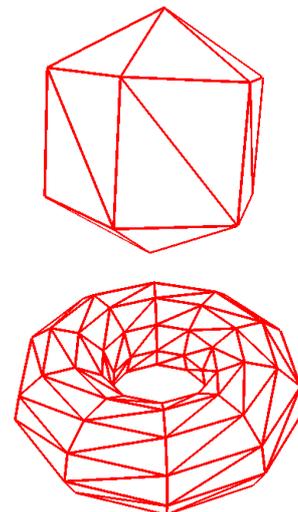


Abbildung 5.6: Die Testdatensätze: Sphäre und Torus.

<sup>3</sup>Hierbei wurde die in Abschnitt 2.2.2 auf Seite 31 beschriebene Modifikation bei der Bestimmung der Tangententlängen angewendet.

<sup>4</sup>Moreton und Séquin erreichen eine Beschleunigung ihrer Methode durch Verwendung von Symmetrieinformation; hier wurde diese zusätzliche Information nicht benutzt.

Die verwendeten Parameter, Optimierungsmethoden, sowie das Laufzeitverhalten der einzelnen Verfahren ist in den Tabellen B.4 – B.7 im Anhang B festgehalten.

Im folgenden werden die  $G^1$ - und die  $G^2$ -Kurvennetzmethode anhand einiger größerer Beispiele getestet. Abbildung 5.7 zeigt zwei Polyeder, die von CAD-Flächen zu Simulationszwecken zunächst regulär abgetastet wurden. Anschließend wurden die regulären Polyeder einer Datenreduktion unterzogen.<sup>5</sup> Zum Vergleich werden die Polyeder mit dem Shirman-Séquin Verfahren interpoliert.

Abbildung 5.13 zeigt die Interpolationsflächen zum ersten Polyeder. Die Shirman-Séquin Fläche weist starke Krümmungsschwankungen in den Rundungsbereichen auf. Die  $G^1$ -Kurvennetzmethode hingegen besitzt ein sehr gleichmäßiges Krümmungsverhalten. Die leichten Krümmungsschwankungen, die noch erkennbar sind, sind vor allem auf die in diesem Regionen auftretenden dünnen Dreiecke zurückzuführen. Etwas mehr Probleme ergeben sich mit der  $G^2$ -Kurvennetzmethode. Die Anwendung der „stabilen“ Funktionale (Thin-Plate bzw. datenabhängiges Thin-Plate Funktional) führt hier nicht auf das gewünschte Ergebnis. Die Anwendung des Funktionals dritter Ordnung in Zusammenhang mit der simultanen Optimierung erweist sich bei diesem Beispiel als extrem instabil. Die Qualität der resultierenden Interpolationsfläche ist daher auch nicht ganz so gut, wie bei der  $G^1$ -Kurvennetzmethode. Für die Optimierung des  $G^1$ - und des  $G^2$ -Kurvennetz kann aufgrund der großen, flachen Bereiche im zugrundeliegenden Polyeder nur das einfache Funktional mit Parametern  $\alpha > 0$  oder  $\beta > 0$  oder das Laplace-Beltrami Funktional eingesetzt werden.

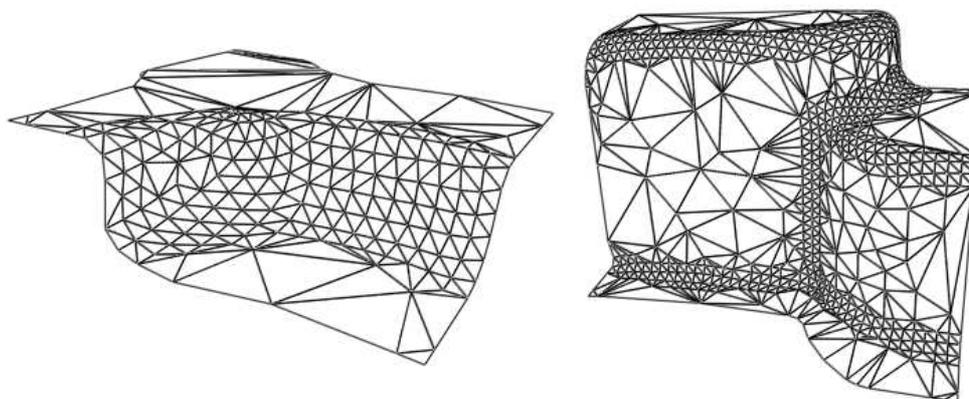


Abbildung 5.7: Die beiden CAD-Flächen: Links die erste CAD-Fläche mit 191 Ecken und 341 Dreiecksseiten; Rechts die zweite CAD-Fläche mit 551 Ecken und 1049 Dreiecksseiten.

Die Interpolationsflächen zur zweiten CAD-Fläche sind in Abbildung 5.14 dargestellt. Bei diesem Beispiel wird die mittlere Krümmung visualisiert, da die zugrundeliegende Fläche in einigen Bereichen nahezu zylindrisches Verhalten hat.<sup>6</sup> Die Shirman-Séquin Methode liefert

<sup>5</sup>Die Datensätze wurden dankenswerterweise von Herrn Dr. Boerger von der Firma composite\_consultants, Erlangen, zur Verfügung gestellt. Die Datenreduktion wurde von Herrn Karbacher vom Physikalischen Institut der Universität Erlangen, Lehrstuhl für Optik, durchgeführt.

<sup>6</sup>Bekanntlich verschwindet die Gaußkrümmung für zylindrischen Flächen, da eine Hauptkrümmung null ist.

auch hier eine Interpolationsfläche mit sehr ungünstigem Krümmungsverhalten. Deutlich zeigt sich die Verbesserung bei Anwendung der  $G^1$ - bzw. der  $G^2$ -Kurvennetzmethode. Für diesen Datensatz liefert das Thin-Plate Funktional zur Bewertung der Interpolationsfläche für die  $G^1$ - und die  $G^2$ -Kurvennetzmethode gute Ergebnisse.

Der Rechenaufwand bei der  $G^2$ -Kurvennetzmethode steigt aufgrund der großen Anzahl der Optimierungsparameter und der relativ schlechten Konditionierung der Optimierungsmatrix.

Tabelle B.8 und B.9 in Anhang B zeigt die verwendeten Parameter und das Laufzeitverhalten für die beiden CAD-Flächen.

Abschließend testen wir die beiden Kurvennetzmethoden an einem Polyeder, das ein Gesicht darstellt. Der Datensatz wurde aus einer Messung mit einem Cyberware<sup>TM</sup>-Scanner mit anschließender Datenreduktion gewonnen.<sup>7</sup> Das Polyeder besteht aus 1496 Punkten und 2927 Dreiecken.

Abbildung 5.15 zeigt in der Mitte das Ausgangspolyeder. Links befindet sich die mit der  $G^1$ -Kurvennetzmethode konstruierte Fläche mit einer Nahaufnahme des Nasenbereichs. Das  $G^1$ -Kurvennetz konnte in diesem Fall nicht bis zur Konvergenz optimiert werden, da im Nasenbereich mit zunehmender Zahl der Iterationen ein nicht winkel-reguläres Kurvennetz entsteht. Die Interpolationsfläche wurde dann mit dem einfachen Thin-Plate Funktional unter Verwendung der simultanen Optimierung berechnet. Die Fläche weist leichte Schwächen im Mund- und im Augenbereich auf (die oberhalb des Auge erkennbar Hautpustel ist im Datensatz schon vorhanden). Die  $G^2$  Kurvennetzmethode generiert eine Interpolationsfläche mit global glatterem Aussehen. Die Probleme im Nasenbereich treten hier nur noch sehr schwach auf. Die Schwierigkeiten im Mund- und Augenbereich sind im Vergleich zur  $G^1$ -Kurvennetzmethode nur wenig verringert. Hier wurde das datenabhängige Thin-Plate Funktional zusammen mit der simultanen Optimierung angewendet.

Die  $G^1$ - und die  $G^2$ -Kurvennetzmethode führen, gegenüber den nicht optimierenden Methoden, in allen bislang getesteten Situationen auf Interpolationsflächen mit deutlich verbesserter Qualität. Auf der anderen Seite ergibt sich auch ein wesentlich günstigeres Laufzeitverhalten als bei der Methode von Moreton und Séquin. Die dabei erzielte Flächenqualität ist durchaus mit der von Moreton und Séquin vergleichbar. Aufgrund der langen Rechenzeiten des Moreton-Séquin-Verfahrens konnten hierbei allerdings nur für relativ kleine Beispiele Vergleiche angestellt werden. Die aufgeführten Beispiele belegen, daß die  $G^1$ - und die  $G^2$ -Kurvennetzmethode auf Polyeder mit bis zu 1500 Punkten eingesetzt werden können.

Im Vergleich der beiden Methoden untereinander erweist sich die  $G^2$ -Kurvennetzmethode meist als das bessere Verfahren: Die resultierenden Interpolationsflächen haben i.a. besseres Krümmungsverhalten, der Algorithmus verhält sich bei Polyedern mit topologisch schwierigen Teilregionen stabiler. Allerdings ist die  $G^1$ -Kurvennetzmethode, bei Einsatz derselben Funktionale und derselben Optimierungsmethode immer schneller.

---

<sup>7</sup>Dieser Datensatz wurde dankenswerterweise von Herrn Keeve vom Institut für Elektrotechnik der Universität Erlangen, Lehrstuhl für Nachrichtentechnik, zur Verfügung gestellt.

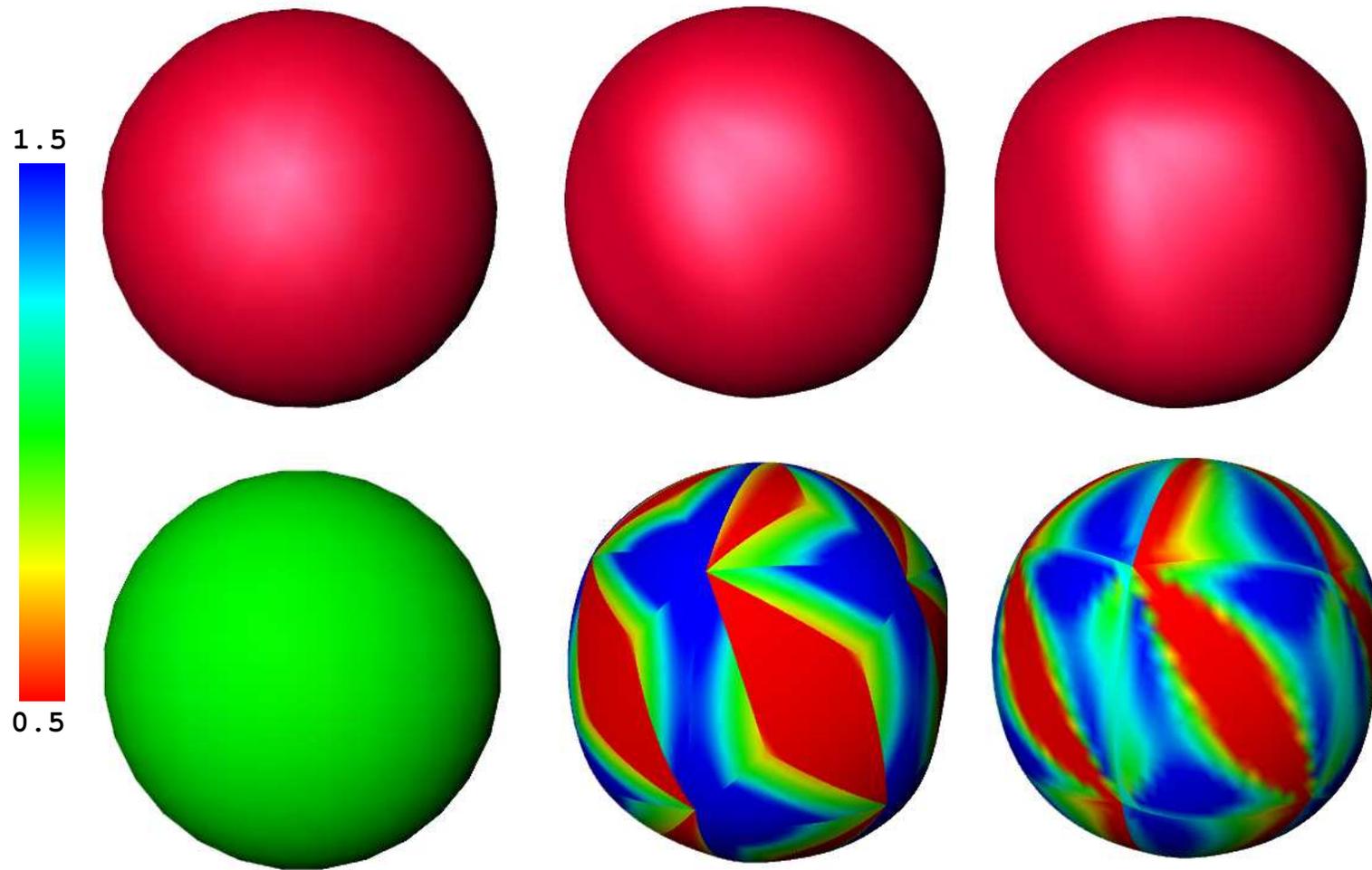


Abbildung 5.8: Interpolationsflächen (oben) und Gauß-Krümmungsplots (unten) für die Sphärendaten: Das Verfahren nach Moreton und Séquin (links), nach Shirman und Séquin (Mitte) und nach Nielson (rechts).



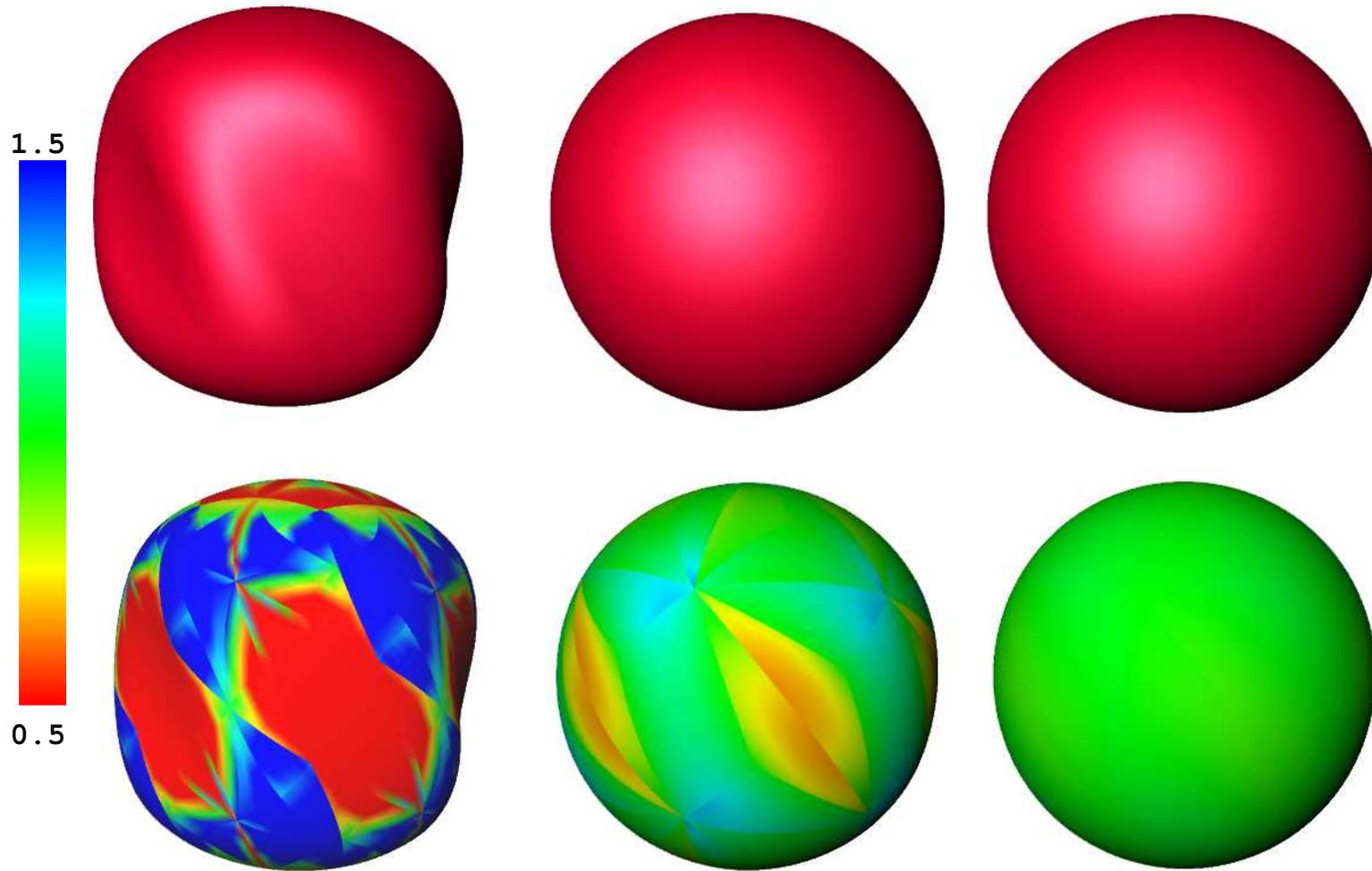


Abbildung 5.9: Interpolationsflächen (oben) und Gauß-Krümmungsplots (unten) für die Sphärendaten: Die Splineflächen nach Loop (links), die  $G^1$ - bzw.  $G^2$ -Kurvennetzmethode (Mitte und rechts).



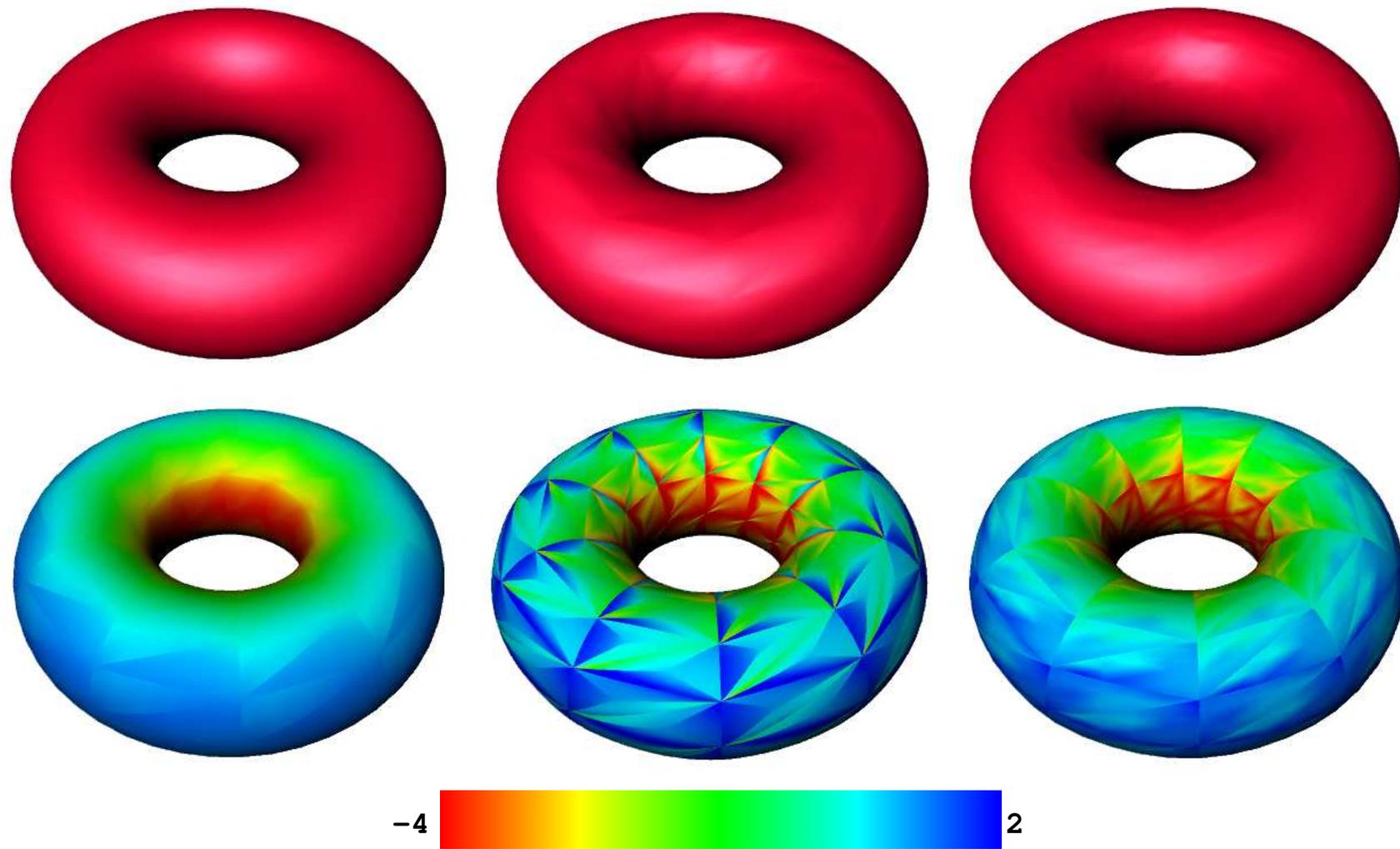


Abbildung 5.10: Interpolationsflächen (oben) und Gauß-Krümmungsplots (unten) für die Torusdaten: Das Verfahren nach Moreton und Séquin (links), nach Shirman und Séquin (Mitte) und nach Nielson (rechts).



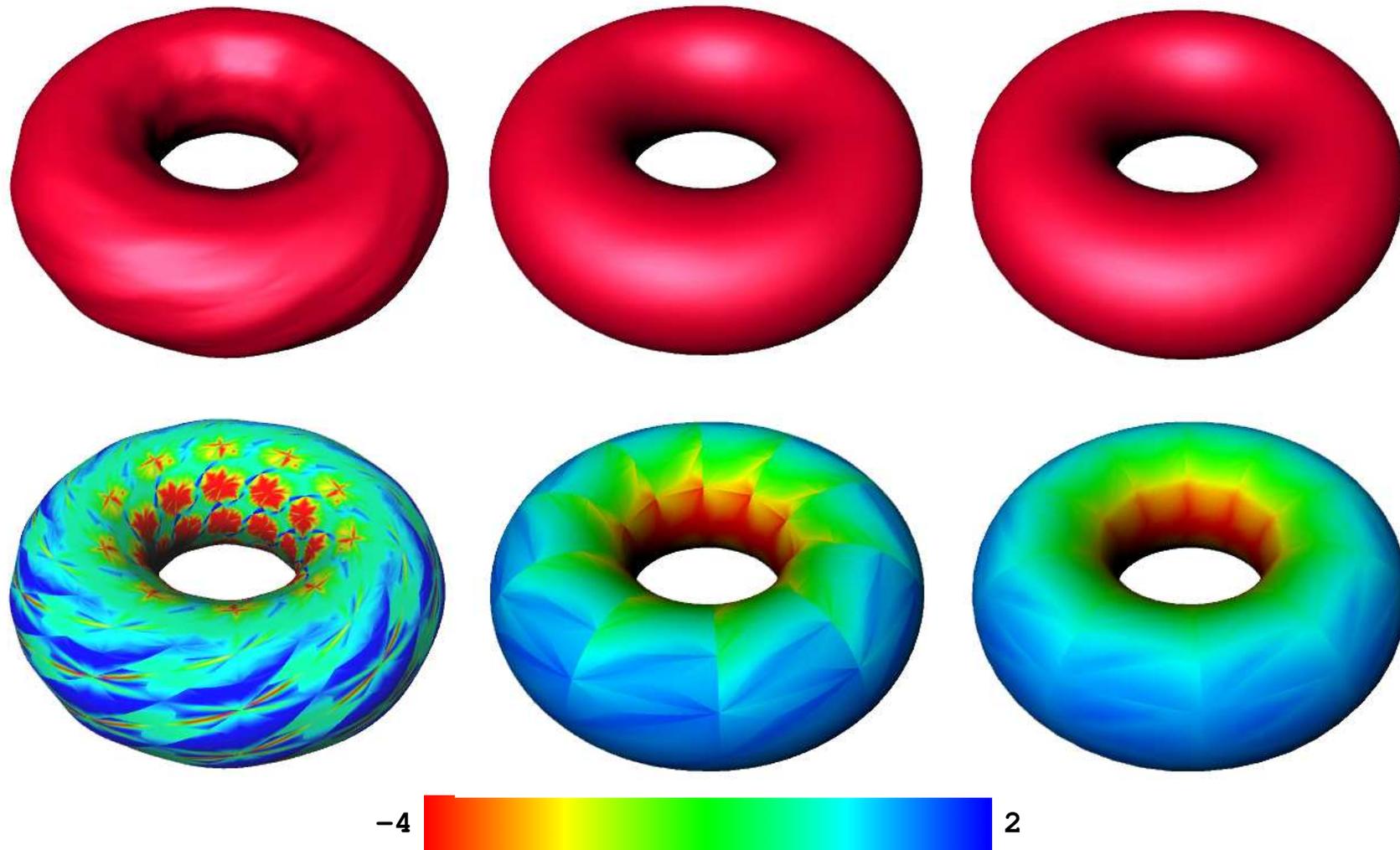


Abbildung 5.11: Interpolationsflächen (oben) und Gauß-Krümmungsplots (unten) für die Torusdaten: Die Splineflächen nach Loop (links) und die  $G^1$ - bzw. der  $G^2$ -Kurvennetzmethode (Mitte und rechts).



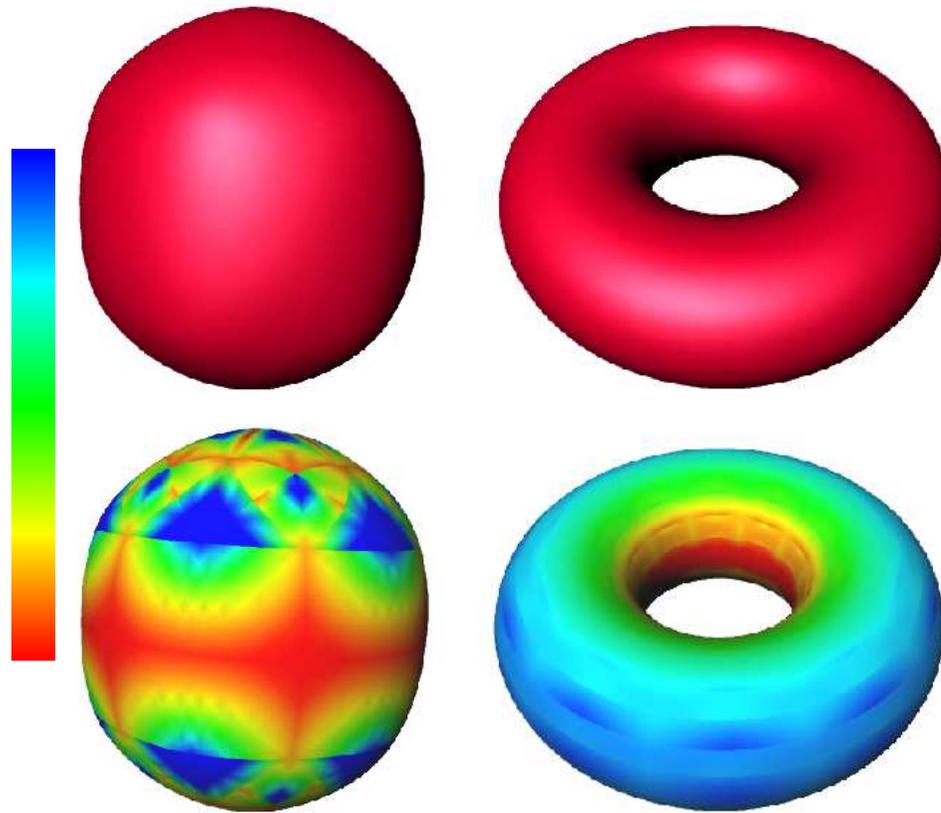


Abbildung 5.12: Interpolationsflächen (oben) und Gauß-Krümmungsplots (unten) für die „entschärften“ Sphären- und Torusdaten mit dem Schema von Loop.



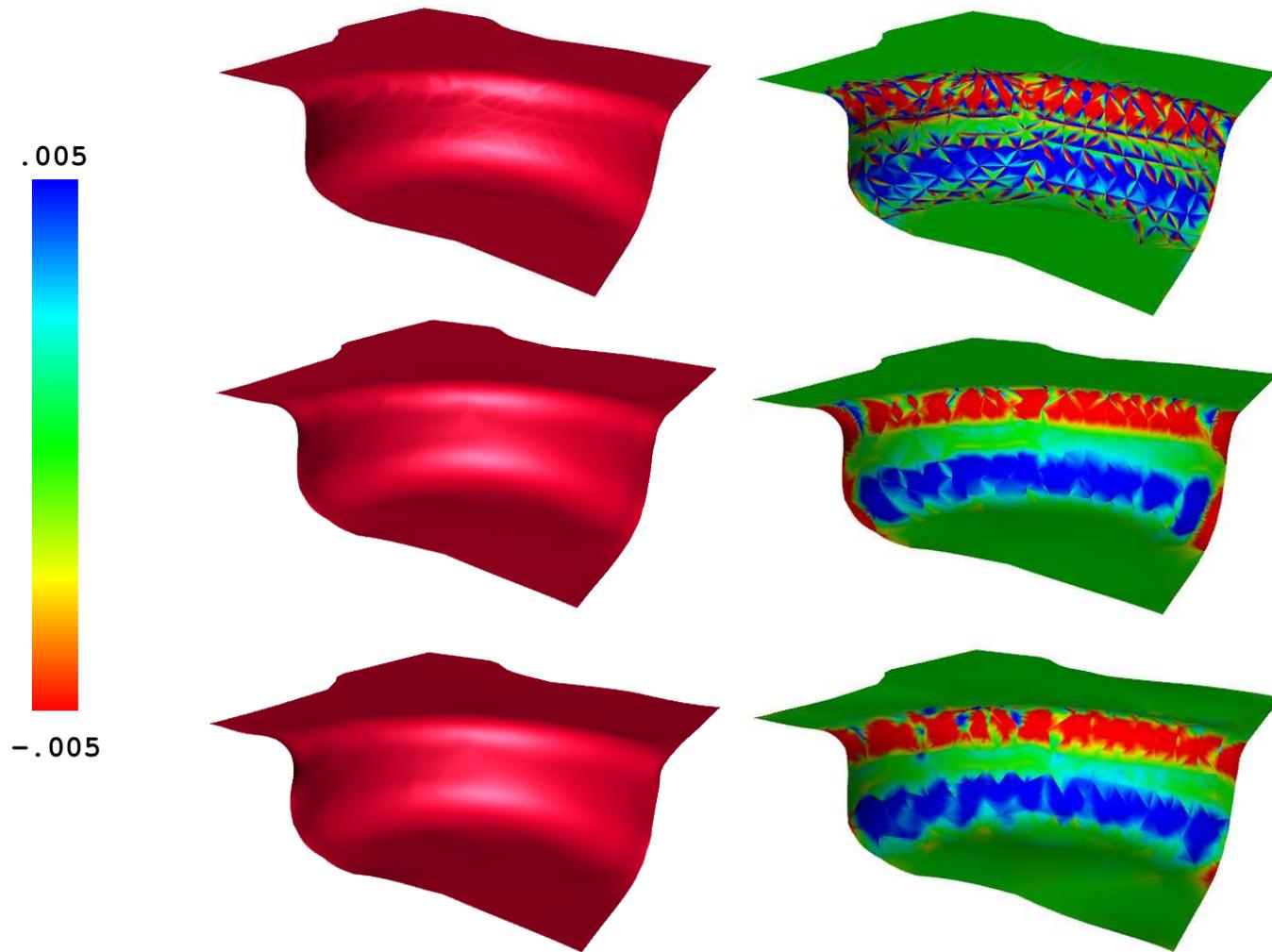


Abbildung 5.13: Interpolationsflächen (links) und Gauß-Krümmungsplots (rechts) für die erste CAD-Fläche: Die Methode nach Shirman und Séquin (oben) und die  $G^1$ - und die  $G^2$ -Kurvennetzmethod (Mitte und unten).



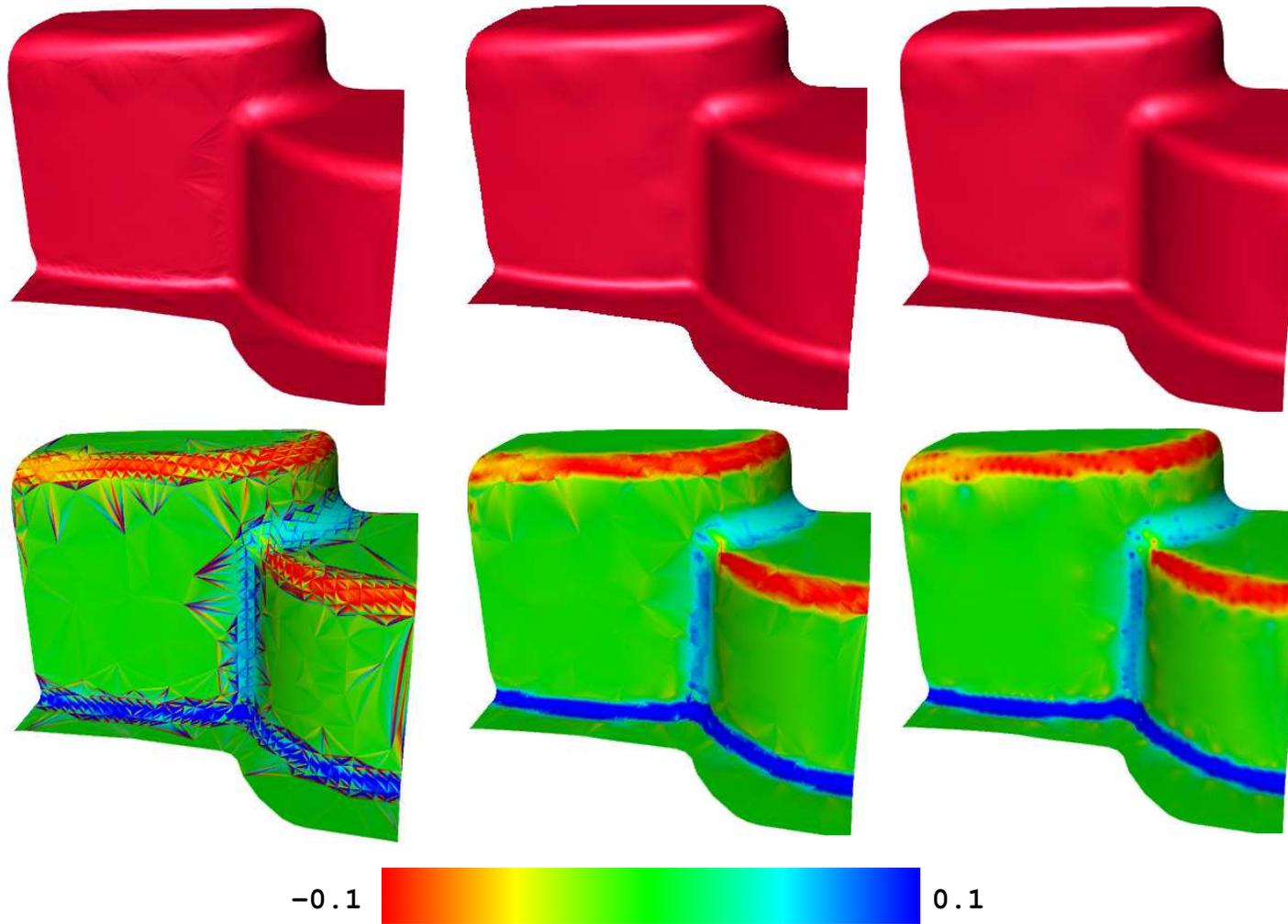


Abbildung 5.14: Interpolationsflächen (oben) und Krümmungsplots für mittlere Krümmung (unten) für die zweite CAD-Fläche: Die Methode nach Shirman und Séquin (links) und die  $G^1$ - und die  $G^2$ -Kurvennetzmethode (Mitte und rechts).





Abbildung 5.15: Das Gesichtsdaten-Polyeder (Mitte) rekonstruiert mit der  $G^1$ - (linke Spalte) und der  $G^2$ -Kurvennetzmethode (rechte Spalte): Die Gesamtansicht (jeweils oben), sowie Nahansichten des Nasenbereichs (jeweils unten).



# Kapitel 6

## Erweiterungen und Anwendungen

In den Kapiteln 4 und 5 wurden Verfahren zur Optimierung von  $G^1$ - und  $G^2$ -Kurvennetzen, sowie Methoden zum Generierung glatter Flächen ausgehend von diesen Kurvennetzen beschrieben. Dabei haben wir uns stets auf das Problem konzentriert, die Ecken eines *triangulierten* Polyeders zu *interpolieren*. In diesem Kapitel werden einige Erweiterungen und Anwendungen der eingeführten Kurvennetzverfahren diskutiert, bei denen die Einschränkungen der dreieckigen Polyederseiten und der Interpolation der Polyederecken z.T. aufgehoben werden. In Abschnitt 6.1 wird zunächst aufgezeigt, wie die Kurvennetzmethoden auf beliebige Polyeder erweitert werden können. Die dabei verwendete Idee der *freien Knoten* ermöglicht einen Approximationsansatz (Abschnitt 6.2). In Abschnitt 6.3 befassen wir uns mit dem Problem der Interpolation gegebener Bézierflächen über ausgewählten Seitenflächen eines Polyeders. Ein Spezialfall dieser Problematik ist das Füllen von  $n$ -eckigen Löchern, die von Bézierflächen berandet sind. Eine weitere Einsatzmöglichkeit besteht in der Konstruktion von Blendflächen beliebiger Topologie. Abschließend werden einige Beispiele zu den verschiedenen Anwendungen gegeben.

### 6.1 Polyeder mit konvexen $n$ -eckigen Seitenflächen

In diesem Abschnitt befassen wir uns mit der Frage, wie die Ecken eines beliebigen Polyeders interpoliert werden können. Es gibt grundsätzlich zwei Möglichkeiten dieses Problem anzugehen. Einerseits kann man versuchen für jede  $n$ -eckige Seitenfläche des Polyeders *eine*  $n$ -eckige Fläche einzupassen (vgl. [LD89, LD90]). Alternativ können *mehrere* Flächen einfacher Form (z.B TP-Flächen oder Dreiecks-Bézierflächen) über einer einziger  $n$ -seitige Seitenfläche angesetzt werden (vgl. [SS87, Hah89, GZ94]).

Betrachtet man die Repräsentationen des  $G^1$ - und des  $G^2$ -Kurvennetzes (vgl. Gleichungen (4.6) auf Seite 55 und (5.5) auf Seite 73), so erkennt man, daß die Position des Kurvennetz-knoten bei  $\mathbf{V}_i$  stets linear in die Darstellung einfließen. Daher kann der Netzwerkknoten bei  $\mathbf{V}_i$  als Parameter für die Optimierung angesetzt werden. Wird die Knoten des Kurvennetzes für die Ecke  $\mathbf{V}_i$  als freier Parameter verwendet, so sprechen wir von einem *freien Knoten* und bezeichnen ihn mit  $\mathbf{S}_i$ .

Das Konzept der freien Knoten machen es nun möglich, daß parametrische Interpolations-

problem auf Polyeder mit beliebigen, *konvexe* Seitenflächen auszuweiten (vgl. Abbildung 6.1):

- (i) Das ursprüngliche Polyeder wird durch Einfügen zusätzlicher Ecken zu einem Polyeder mit ausschließlich dreieckigen Seitenflächen verfeinert. Hierzu wird für jede Seitenfläche mit mehr als drei Ecken eine Ecke eingefügt und alle ursprünglichen Ecken der Seitenfläche mit dieser durch eine Kante verbunden. Die Knoten des Kurvennetzes werden an den zusätzlichen Ecken als frei markiert.
- (ii) Die Repräsentation des Kurvennetzes, die Bestimmung der lokalen Parameter und die Optimierung des Kurvennetzes arbeiten analog zu den in Kapitel 4 und Kapitel 5 beschriebenen Verfahren. Der einzige Unterschied besteht darin, daß neben den partiellen Ableitungen  $S_{ui}, S_{vi}$  bzw.  $S_{ui}, \dots, S_{vvi}$  auch der Kurvennetzknotten  $\mathbf{S}_i$  bei jeder zusätzlichen Ecke  $\mathbf{V}_i$  variiert werden.
- (iii) Zum Ausfüllen des Kurvennetzes werden ebenfalls die bereits beschriebenen Methoden angewendet. Auch die simultane Optimierung von Kurvennetz und Flächen (vgl. Abschnitte 4.3 und 5.3) kann eingesetzt werden, da die freien Knoten auch linear in die Flächendarstellung eingehen.

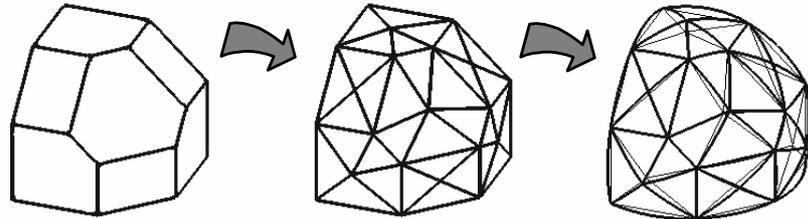


Abbildung 6.1: Behandlung von Polyedern mit  $n$ -eckigen Seitenflächen: Im ersten Schritt werden zusätzliche Ecken eingeführt; bei der Optimierung verändern die entsprechenden freien Knoten ihre Position (rechts ist das verfeinerte Polyeder dünn, das optimierte Kurvennetz dick eingezeichnet).

## 6.2 Glätten von Polyedern

Das Glätten von Daten unter Verwendung von Interpolationsmethoden wird häufig angewendet, wenn die exakte Interpolation der Daten nicht explizit gefordert wird bzw. aufgrund von Ungenauigkeiten, z.B. bei der Datenerfassung, gar nicht erwünscht ist. Das Ziel besteht dann darin, eine Fläche zu finden, die möglichst glatt ist (also z.B. gleichmäßigen Krümmungsverlauf aufweist) und die gegebenen Daten ausreichend gut approximiert.

Für den Kurvenfall stellt sich die prinzipielle Vorgehensweise, die auch unter dem Begriff „smoothing“ bekannt geworden ist, wie folgt dar (vgl. Abbildung 6.2 und [HL92]):

Zu gegebenen Parameterwerten  $u_i$ ,  $i = 1, \dots, N$  in dem reellen Parameterintervall  $[a, b]$  und zu gegebenen *Aufmaßpunkten*  $\mathbf{P}_i$ ,  $i = 1, \dots, N$  ist eine Kurve  $\mathbf{C} : [a, b] \rightarrow \mathbb{R}^3$  gesucht, die das Funktional

$$\sigma(\mathbf{C}) = \sigma_C(\mathbf{C}) + \sum_{i=1}^N c_i \|\mathbf{C}(u_i) - \mathbf{P}_i\|^2, \quad c_i > 0$$

minimiert.  $\sigma_C$  bezeichnet dabei ein quadratisches Funktional zur Bewertung der Glätte. Die reellen Werte  $c_i$  werden als *Steifigkeitskoeffizienten* oder *Federkonstanten* bezeichnet.

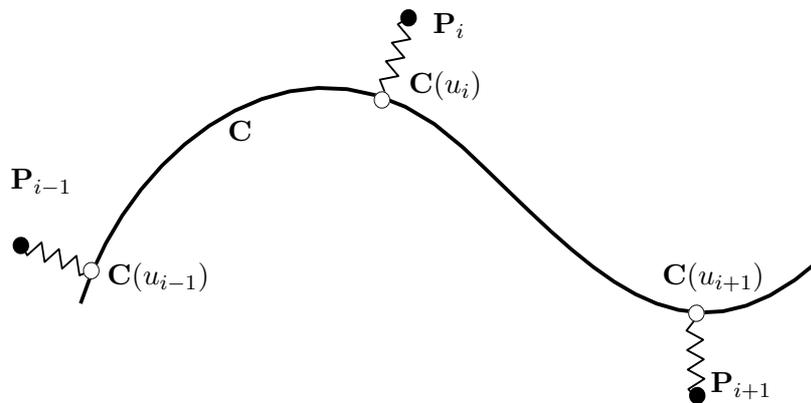


Abbildung 6.2: Die Idee des Glättungsansatzes: Die Aufmaßpunkte werden nicht interpoliert, sondern über Federn mit den entsprechenden Kurvenpunkten  $\mathbf{C}(u_i)$  verknüpft.

Wie wir in Abschnitt 6.1 gesehen haben, kann durch Kennzeichnung von freien Knoten  $\mathbf{S}_i$  des  $G^1$ - und des  $G^2$ -Kurvennetzes die entsprechende Position variiert werden. Um für ein Kurvennetz einen Glättungsansatz wie in obigem Kurvenfall machen zu können, werden alle Knoten des Kurvennetzes als frei markiert. Bei der Optimierung werden die Ecken des Polyeders als Aufmaßpunkte verwendet. Der zu dem Aufmaßpunkt  $\mathbf{V}_i$  gehörige Knoten  $\mathbf{S}_i$  des Kurvennetzes wird bei der Optimierung variiert, wobei  $\mathbf{S}_i$  über eine Federkonstante  $c_i$  an die Ecke  $\mathbf{V}_i$  angebunden wird. Dies geschieht über Hinzunahme des Termes

$$\sum_{i=1}^N c_i \|\mathbf{S}_i - \mathbf{V}_i\|^2$$

zu dem für die Optimierung des Kurvennetzes angesetzten Funktional.

Das optimierte Kurvennetz kann mit den in Abschnitt 4.2 und in Abschnitt 5.2 beschriebenen Methoden zum Ausfüllen des  $G^1$ - bzw. des  $G^2$ -Kurvennetzes zu einer glatten Fläche fortgesetzt werden.

Im Unterschied zu dem Approximationsansatz bei Kurven verhalten sich die Kurvennetze „steifer“. Dies ist auf die Tatsache zurückzuführen, daß die Ableitungen der einzelnen Kurven  $\mathbf{C}_{ij}$  des Kurvennetzes in den Eckpunkten nicht beliebig variieren, sondern durch die Fixierung der lokalen Parameter bereits in ihrer Größenordnung festgelegt sind.

## 6.3 Interpolation gegebener Bézierflächen

Das Füllen von  $n$ -eckigen Löchern, die von Polynomflächen berandet sind und das Erzeugen von Blendflächen zwischen zwei oder mehreren Polynomflächen sind Probleme, die häufig im Umfeld der Modellierung von Freiformflächen auftreten (vgl. [Hah89, GZ94, Gre94a]).

Wir betrachten diese Probleme als Teil der folgenden Fragestellung, wie sie sich für Flächen über Polyedern stellt:

**Gegeben:** Ein Polyeder mit beliebigen  $n$ -eckigen, konvexen Seitenflächen, bei dem zu einigen der drei- oder viereckigen Seitenflächen schon TP-Bézier bzw. Dreiecks-Bézierflächen definiert sind.

Wir setzen hierbei voraus, daß die gegebenen Flächenstücke sinnvoll gewählt sind, d.h. daß die entsprechenden Eckpunkte des Polyeders von diesen Flächenstücken interpoliert werden und daß die Flächenstücke die Topologie der entsprechenden Seitenflächen erhalten und gegebenenfalls glatt aneinander anschließen.

**Gesucht:** Zu allen Seitenflächen des Polyeders, für die bislang keine Flächen definiert sind, sollen Teilflächen bestimmt werden, so daß eine global glatte (tangentialstetige oder  $G^1$  stetige) Fläche entsteht.

Durch diese Formulierung ist die obigen Problemstellung mit den Kurvennetzmethoden lösbar.

Es ist klar, daß nicht beliebige Flächen vorgegeben sein können, wenn die  $G^1$ - und  $G^2$ -Kurvennetzmethoden zur Interpolation eingesetzt werden sollen. Als Einschränkung muß gefordert werden:

**$G^1$ -Kurvennetzmethode:** Alle gegebenen Flächen über drei- bzw. viereckigen Seitenflächen des Polyeders sind kubische Dreiecks-Bézierflächen bzw. biquadratische TP-Bézierflächen.

**$G^2$ -Kurvennetzmethode:** Die gegebenen Flächen über drei- bzw. viereckigen Seitenflächen des Polyeders sind Dreiecks-Bézierflächen vom Grad vier bzw. bikubische TP-Bézierflächen.

Diese Einschränkungen garantieren, daß im Falle der  $G^1$ -Kurvennetzmethode die durch die bereits vorhandenen Flächenstücke gegebenen Randkurven und Querableitungsvektorfelder maximal kubisch bzw. quadratisch sind. Damit können die Randkurven der gegebenen Flächen als Kurven des  $G^1$ -Kurvennetzes dargestellt und die modifizierte Shirman-Séquin Methode zum Ausfüllen des Polyeders verwendet werden (vgl. Abschnitt 4.2).

Für die  $G^2$ -Kurvennetzmethode gilt ähnliches. Hier darf der Grad der Randkurven nicht mehr als fünf, der Grad der Querableitungsvektorfelder nicht mehr als drei betragen. Dies wird durch Einhalten der obigen Anforderungen an die vorgegebenen Flächen gewährleistet.

Natürlich dürfen die durch die initialen Flächen gegebenen Positions-, Normalen- und gegebenenfalls Krümmungsdaten in den entsprechenden Ecken des Polyeders nicht verändert

werden. Die zugehörigen partiellen Ableitungen werden also gemäß der durch die Flächen bekannten Normalen- und Krümmungsdaten fixiert.

## 6.4 Ergebnisse

In dem abschließende Abschnitt dieses Kapitels über Anwendungen der  $G^1$ - und der  $G^2$ -Kurvennetzmethodene geben wir einige Beispiele zu den verschiedenen Erweiterungen, wie sie in den vorherigen Abschnitten beschrieben wurden.

Abbildung 6.3 links zeigt ein Beispiel zur Approximation von Polyedern. Hier wurde die  $G^2$ -Kurvennetzmethodene mit einer globalen Federkonstanten von 0.7 eingesetzt. Es zeigt sich, daß die Polyederecken noch gut approximiert werden (rechter Teil). Der visuelle Eindruck der Fläche und der Krümmungsverlauf hat sich im Vergleich zur Interpolation (linker Teil) deutlich verbessert.

Die rechte Seite von Abbildung 6.3 zeigt drei Beispiele zur Interpolation gegebener Bézierflächen. Bei der Hausecke sind insgesamt zehn biquadratische TP-Bézierflächen vorgegeben. Die Füllfläche besteht aus einer Interpolationsfläche mit einem freien Knoten, die mit der  $G^1$ -Kurvennetzmethodene generiert wurde. Die Vorgaben bei der Kofferecke sind dreimal drei Dreiecks-Bézierflächen vom Grad 3, die die entsprechenden Ebenen parametrisieren. Zu Berechnung der Interpolationsfläche mit der  $G^1$ -Kurvennetzmethodene werden vier freie Knoten eingesetzt. Die Blendfläche zu den drei Rohrenden schließlich wurde mit der  $G^2$ -Kurvennetzmethodene bestimmt, die über fünf freie Knoten verfügt. Vorgegeben sind hierbei dreimal vier biquadratische TP-Bézierflächen.

Als letztes Anwendungsbeispiel kommen wir nochmals auf den Gesichtsdatensatz aus Abbildung 5.15 auf Seite 99 zurück. Abbildung 6.4 zeigt das Ergebnis der Approximation des Polyeders. Verwendet wird hierbei die  $G^2$ -Kurvennetzmethodene mit einer globalen Federkonstante von 0.7. Die resultierende Approximationsfläche hat einen wesentlich weicheren Verlauf als die entsprechenden Interpolationsflächen. Die Probleme im Mund- und Augenbereich, die bei den Interpolationsflächen auftreten sind kaum mehr erkennbar, die Hautpustel über dem Auge ist nur noch schwach ausgeprägt.

Tabellen B.12 und B.13 im Anhang B gibt Auskunft über die verwendeten Parameter und über die Laufzeiten der einzelnen Verfahren.



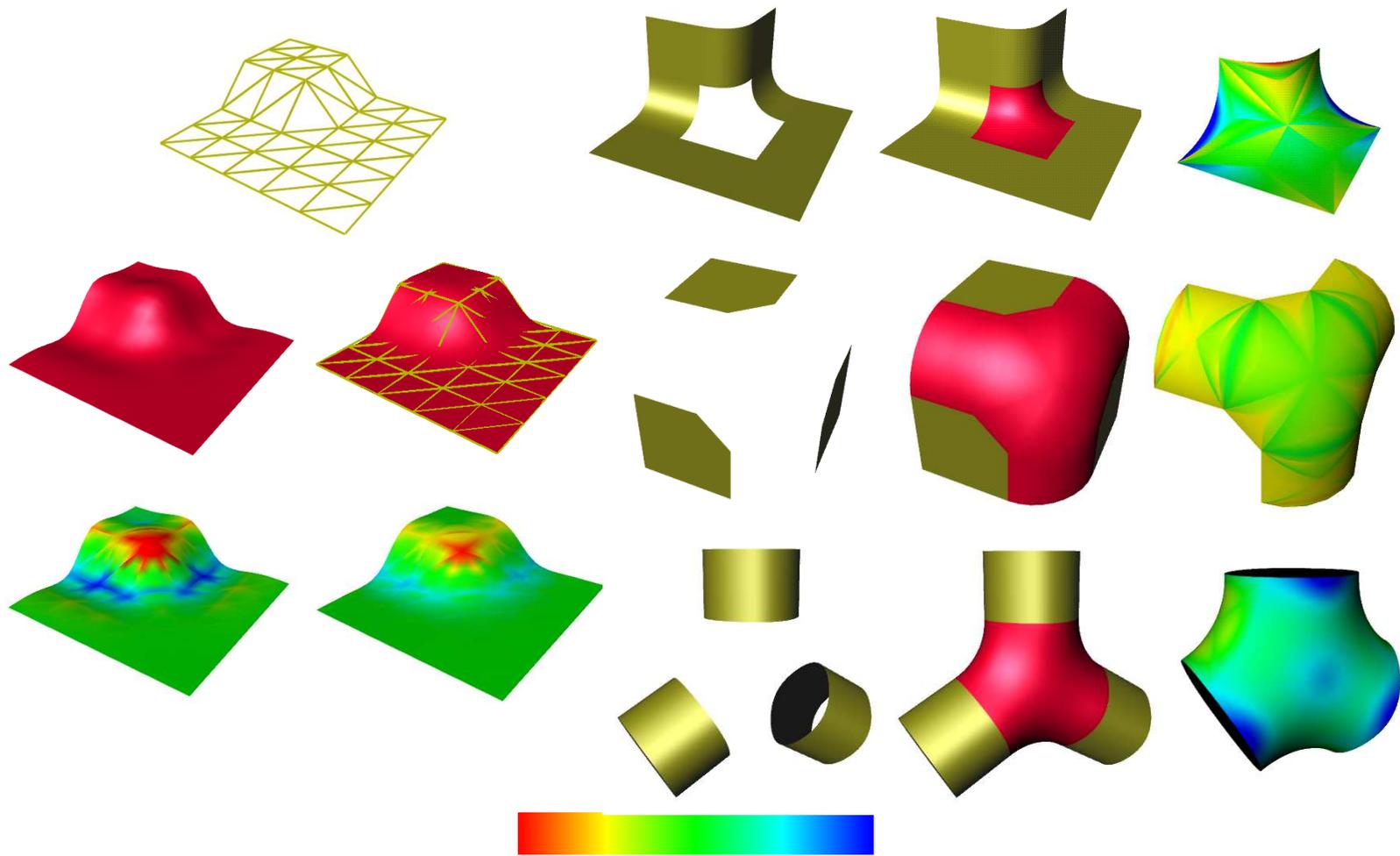


Abbildung 6.3: Beispiel zu den Anwendungen: Links ein Approximationsbeispiel mit globaler Federkonstante 0.7 (rechter Teil) im Vergleich mit der Interpolation desselben Polyeders (rechter Teil) ( $G^2$ -Kurvennetzmethode); Rechts drei Beispiele zur Interpolation gegebener Bézierflächen: Eine Hausecke ( $G^1$ -Kurvennetzmethode), eine Kofferecke ( $G^1$ -Kurvennetzmethode) und eine Blendfläche zwischen drei Rohrenden ( $G^2$ -Kurvennetzmethode).



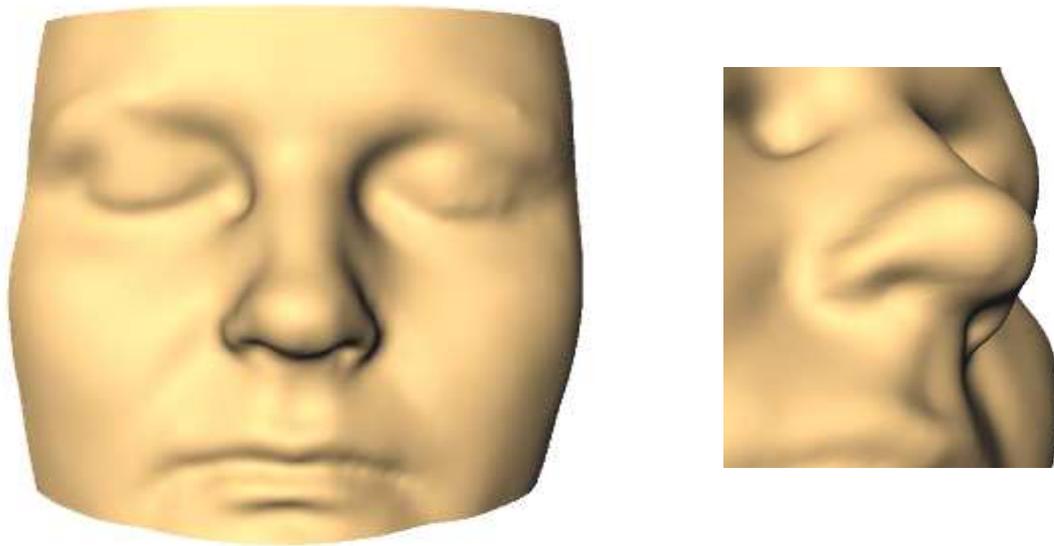


Abbildung 6.4: Die  $G^2$ -Kurvennetzmethode als Approximationsansatz angewendet auf die Gesichtsdaten (links) und die entsprechende Nahaufnahme des Nasenbereichs (rechts).



# Kapitel 7

## Implementation

In dem letzten Kapitel dieser Arbeit wird auf die Implementation der Verfahren, die in den vorangegangenen Kapiteln beschrieben wurden, eingegangen.

Ausnahmslos alle Methoden der Kapitel 2, 3, 4, 5 und 6 wurden in eine bestehende, objektorientierte Spline-Tool-Bibliothek unter C++ integriert.<sup>1</sup> Diese Spline-Tool-Bibliothek wird am Institut für Mathematische Maschinen und Datenverarbeitung (IMMD) der Universität Erlangen, Lehrstuhl für Graphische Datenverarbeitung, weiterentwickelt und gepflegt. Sie dient als gemeinsame Plattform für neuen Methoden, die am Lehrstuhl im Bereich der Geometrischen Modellierung entwickelt und/oder getestet werden (vgl. [SKKG94a, SKKG94b, GKP<sup>+</sup>95]).

In Abschnitt 7.1 werden ganz kurz die wesentlichen Merkmale des objektorientierten Designs angesprochen. Eine Übersicht über die Klassenhierarchie mit den wichtigsten Klassen der Spline-Tool Bibliothek findet sich in Abschnitt 7.2. Einige ausgewählte Klassen, die für die Implementierung der in dieser Arbeit beschriebenen Algorithmen von grundlegender Bedeutung sind, werden in Abschnitt 7.3 kurz erläutert.

### 7.1 Objektorientiertes Design

Die wesentliche Aufgabe beim Design objektorientierter Programme und Bibliotheken besteht in der Festlegung der Operationen, die auf die Objekte einer Klasse angewendet werden können. Von außen betrachtet, werden die Objekte einer Klasse durch diese Operationen bzw. Methoden eindeutig beschrieben. Interne Strukturen sind für den Benutzer nicht zugänglich (Datenkapselung). Durch Ableiten können die Methoden einer Klasse auf eine Subklasse übertragen werden. Hierbei kann eine abgeleitete Klasse durch Überschreiben der Methode ihrer Basisklasse eine andere Implementierung der Methode verwenden, um die geforderte Funktionalität zur Verfügung zu stellen. Durch Einführung abstrakter Basisklassen ist es möglich gemeinsame Schnittstellen für eine Gruppe von Klassen zu schaffen, wodurch auch eine einfache Referenzierung von mehreren, verschiedenen Objek-

---

<sup>1</sup>Eine gewisse Ausnahme sind hierbei die funktionalen MNN-Verfahren (vgl. Abschnitt 2.1.2 auf Seite 21). Diese werden nach einer nicht aufwärts kompatiblen Änderung in der aktuellen Bibliothek nicht mehr unterstützt.

ten von abgeleiteten Klassen ohne explizite Typumwandlung möglich ist (Polymorphismus, virtuelle Methoden).

## 7.2 Die Spline-Tool Bibliothek

Abbildung 7.1 zeigt einen Auszug aus der Klassenhierarchie der Bibliothek in dem alle für die Implementation der einzelnen Verfahren wichtigen Klassen dargestellt sind. Eine der wesentlichen Ideen bei der Konzeptionierung dieser Bibliothek ist die Ausnutzung der Vererbung zur einfachen Einbindung neuer Algorithmen, sowie die Strukturierungsmöglichkeit durch Abbildung der verschiedenen Flächentypen auf die Klassenhierarchie. Dadurch wird es möglich durch Überschreiben weniger Basismethoden die volle Funktionalität, wie sie in der Basisklasse zur Verfügung gestellt wird, zu erhalten. Für die verschiedenen Flächenklassen wird dies dadurch möglich, daß die abstrakte Basisklasse `ParamSurface` zur Repräsentation parametrischer Flächen alleine unter Verwendung der Punktauswertung alle differentialgeometrischen Größen einer Fläche  $\mathbf{S}$  für einen konkreten Parameter  $\mathbf{u}$  numerisch bestimmen kann (vgl. [SKKG94a, SKKG94b]).

Dies mag, je nach Auswertungsalgorithmus für die Fläche, nicht besonders effizient sein, doch es befreit den Benutzer von der Implementierung der Methoden zur Berechnung dieser geometrischen Größen. Durch einfaches Überschreiben von Methoden kann jederzeit ein effizienterer Algorithmus zur Bestimmung der differentialgeometrischen Größen einer Fläche eingebracht werden.

Differentialgeometrisch gesehen, ist eine Fläche dadurch charakterisiert, daß sie *lokal* parametrisierbar ist (vgl. Abschnitt 1.2 auf Seite 6). Dieser Sachverhalt wird durch die Klasse `ObmCompositeParamSurf` wiedergespiegelt. Beide Klassen, `ParamSurface` und `ObmCompositeParamSurf` sind von der abstrakten Basisklasse `ObmBaseSurface` abgeleitet. Dies dient der einheitlichen Referenzierung von Objekten verschiedener Flächenklassen. Um eine möglichst große Flexibilität zu erhalten, wird zur Repräsentation der einzelnen Flächenstücke in `ObmCompositeParamSurf` eine Liste von Referenzen auf Objekte vom Typ `ObmBaseSurface` verwaltet. Damit kann prinzipiell eine beliebige Hierarchietiefe von Flächenstücken aufgebaut werden: Ein Flächenstück, welches selbst wiederum aus mehreren Teilflächen besteht (beispielsweise besteht eine Shirman-Séquin Interpolationsfläche aus Teilflächen, die selbst aus je drei Dreiecks-Bézierflächen bestehen; vgl. Abschnitt 2.2.1).

Die verschiedenen Klassen in denen Verfahren zur Lösung des funktionalen Interpolationsproblems implementiert sind, sind von der Basisklasse `InterpolSurface` abgeleitet (vgl. Abschnitt 2.1 auf Seite 17 und Kapitel 3 auf Seite 37). In dieser Basisklasse werden grundlegende Algorithmen, wie Methoden zur Verwaltung der Datenpunkte und der Datenwerte, zur Verfügung gestellt.

Zur Darstellung von Kurven und Flächen benutzt die Spline-Tool Bibliothek `Open InventorTM` (vgl. [Wer94]), eine objektorientierte Bibliothek zur Erstellung von 3D Anwendungsprogrammen. `Open InventorTM` baut im wesentlichen auf der `Open GLTM` Bibliothek auf (vgl. [NDW93]).

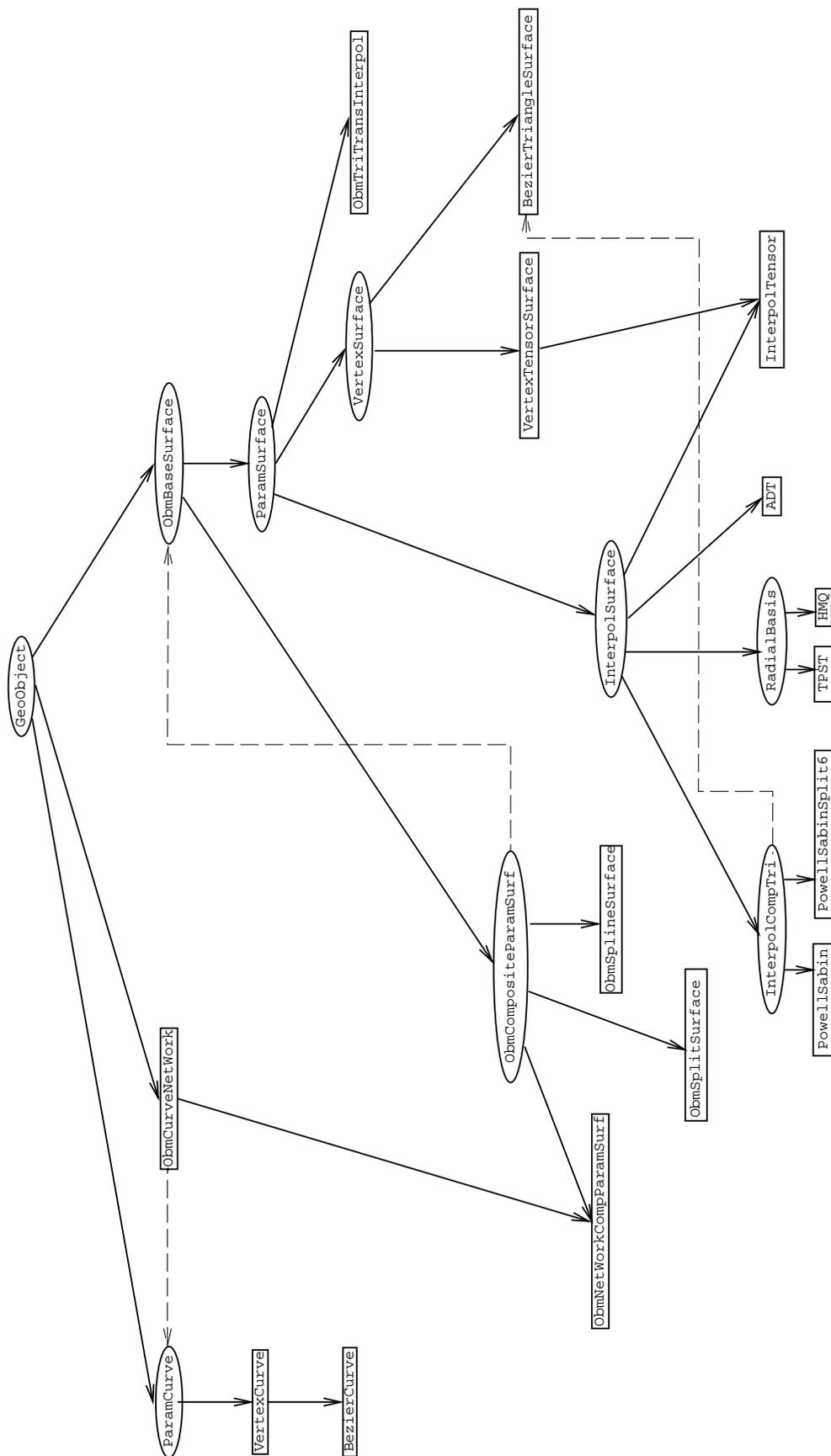


Abbildung 7.1: Auszug aus der Klassenhierarchie: Abstrakte Basisklassen sind durch Ovale gekennzeichnet. Ableitungen werden durch dicke, durchgezogene Linien, wichtige Einbeziehungen von Klassen als Variable bzw. als Listen in andere Klassen werden durch dünne, gestrichelte Linien dargestellt.

Es hat sich gezeigt, daß das Konzept einer Spline-Tool Bibliothek sehr gut für das schnelle Einbauen und Testen von neuen Algorithmen geeignet ist. Hierbei wird extensiv von der Vererbungshierarchie Gebrauch gemacht: Es mußten nur an wenigen Stellen Algorithmen zur Bestimmung von geometrischen Eigenschaften von Flächen (Punktauswertung) integriert werden. Die Einbindung von Klassen zur Durchführung von numerischer Integration, zur Bestimmung von Optimierungsgrößen für Basisflächen (Bézier- und B-Splineflächen) und zur Behandlung von linearen Gleichungssystemen hat den Aufwand zur Umsetzung der eigentlichen Algorithmen deutlich reduziert.

Die in dieser Arbeit vorgestellten Algorithmen wurden über den Verlauf der letzten Jahre entwickelt und in die Spline-Tool Bibliothek integriert. Dies führt ganz zwangsläufig dazu, daß gewisse Teilbereiche der Bibliothek stark verwachsen sind. Ein weiteres Problem zeigt sich bei der Anwendung größerer Datensätze. Hier mußten zunächst einige Engpässe bei der Visualisierung der Flächen überwunden werden, da die Spline-Tool Bibliothek ursprünglich nicht für solche großen Beispiele konzipiert war.

### 7.3 Spezielle Klassen

Im folgenden werden einige wichtige Klasse, die zur Umsetzung der verschiedenen Methoden in die Spline-Tool Bibliothek integriert wurden, kurz erklärt.

#### `ObmCurveNetWork`

Die Klasse `ObmCurveNetWork` beschreibt ein Kurvennetz. Wir gehen dabei davon aus, daß ein Kurvennetz als Teil einer Zweischrittmethode zur Generierung von Interpolationsflächen verwendet wird. Daher enthält diese Klasse das zu interpolierende Polyeder als Variable. Die eigentlichen Kurven werden als eine Liste von Referenzen auf den Basistyp der parametrischen Kurven, `ParamCurve` verwaltet. Zusätzlich wird in einer Liste von Nebenbedingungen festgelegt, welche geometrischen Eigenschaften, wie „freie Knoten“ oder Interpolation gegebener Normalen in einer Ecke, das Kurvennetz haben soll. Private Daten, die die einzelnen Verfahren zur Berechnung der Kurvennetze benötigen, werden in einer dafür vorgesehenen Struktur gespeichert. Diese private Struktur wird in `ObmCurveNetWork` referenziert.

Das Kurvennetz wird über die Methode `constructNet` der Klasse `ObmCurveNetWork` mit dem gewünschten Verfahren entsprechend der gesetzten Parameter bestimmt.

#### `ObmNetWorkCompParamSurf` :

Viele in den vorherigen Kapiteln beschriebenen Verfahren zur Lösung des parametrischen Interpolationsproblems basieren auf den zwei Schritten: Generierung eines Kurvennetzes und Interpolation des Kurvennetzes durch geeignete Flächenstücke. Die Klasse `ObmNetWorkCompParamSurf` repräsentiert eine solche stückweise parametrische Fläche, die auf einem Kurvennetz aufbaut.

Im wesentlichen vereinigt sie die Funktionalität der Kurvennetzklasse `ObmCurveNetWork` und der Klasse `ObmCompositeParamSurf`, die zur Darstellung von stückweisen parametrischen Flächen benutzt wird. Zusätzlich wird eine Liste von geometrische Nebenbedingungen verwaltet, die angeben, ob beispielsweise ein Approximationsansatz gemacht werden soll oder ob vordefinierte Bézierflächen interpoliert werden müssen. Eine Referenz auf eine private Datenstruktur dient dem Zugriff auf Parameter, die von den einzelnen Verfahren zur Bestimmung der Fläche benötigt werden.

Die Methode `constructSurface` berechnet die Interpolationsfläche und ggf. das Kurvennetz anhand der gegebenen, privaten Daten. Erwähnenswert ist hier, daß diese Klasse eine private Methode enthält, die die geometrischen Nebenbedingungen des Kurvennetzes mit den geometrischen Nebenbedingungen für die Flächen synchronisiert.

#### `ObmSplitSurface`:

In dieser Klasse werden die Splitflächen, z.B. die  $3 \times 4$  Splitfläche nach Shirman und Séquin, beschrieben.

Diese Flächen sind immer über  $n$ -eckigen Polygonen definiert. Wir gehen davon aus, daß zu einer Splitfläche Randkurven über dem Polygon und/oder weitere Kontrollpunkte oder -vektoren gegeben sind, die die Splitfläche eindeutig festlegen. Die Randkurven und die zusätzlichen Kontrollpunkte werden in dieser Klasse verwaltet. Parameter, die von den einzelnen Verfahren zur Bestimmung der Fläche benötigt werden, werden in einer privaten Datenstruktur gehalten, die in dieser Klasse referenziert wird.

Auch diese Klasse besitzt eine zentrale Methode, um die Splitfläche entsprechend der gesetzten Parameter zu berechnen.

#### `ObmSplineSurface`:

Dies ist die Basisklasse zur Generierung von Splineflächen nach dem Verfahren von Loop oder Peters. Da diese Verfahren nicht zu den Zwischrittmethoden gehören, wird diese Klasse direkt von `ObmCompositeParamSurf` und nicht von `ObmNetWorkCompParamSurf` abgeleitet.

Die Klasse `ObmSplineSurface` besitzt private Variablen zur Verwaltung der verschiedenen Polyeder, die zur Konstruktion einer Splinefläche notwendig sind: Das Kontrollpolyeder, das verfeinerte Polyeder und gegebenenfalls das zur Interpolation verwendete, transformierte Polyeder. Private Daten werden in einer eigens dafür definierten Struktur gespeichert, die in dieser Klasse referenziert wird.

Methoden zur Generierung der verschiedenen Polyeder aus dem gegebenen Kontrollpolyeder werden zur Verfügung gestellt. Ebenso gibt es eine Methode, die Anhand der bereitgestellten privaten Daten, die lokale Bézier-Repräsentation der resultierenden Splinefläche bestimmt.

**InterpolTensor:**

Das in Abschnitt 3 vorgestellte Verfahren zur Lösung des funktionalen Interpolationsproblems beruht auf dem Ansatz einer TP-B-Spline Fläche. Die Klasse `InterpolTensor` ist daher sowohl von `VertexTensorSurface` (Beschreibung allgemeiner Tensorflächen, die auf regulären Kontrollnetzen aufbauen) und von der Basisklasse `InterpolSurface` abgeleitet.

Neben den von den Basisklassen geerbten Klassenvariablen werden weitere Informationen über die Unterteilung in  $4 \times 4$ -Blöcke benachbarter Knoten, sowie über das zu verwendende Optimierungsfunktional benötigt.

Über die Methode `constructSurface` wird anhand der gegebenen Informationen die TP-B-Splinefläche berechnet.

# Zusammenfassung und Ausblick

In dieser Arbeit wurden verschiedene bekannte Methoden zur Lösung des *funktionalen* und des *parametrischen* Interpolationsproblems beschrieben. Im funktionalen wie im parametrischen Fall zeigt sich, daß viele der mit den bekannten, heuristischen Methoden generierten Interpolationsflächen unerwünschte Falten oder Beulen aufweisen (vgl. auch [Man92, LMD92]). Optimierende Verfahren hingegen sind z.T. extrem zeitaufwendig.

Die in dieser Arbeit vorgestellten neuen Verfahren zur Lösung der Interpolationsprobleme bauen auf der Optimierung der *freien* Parameter, d.h. der Parameter die nicht durch Stetigkeits- oder Interpolationsbedingungen fixiert sind, auf. Um den Rechenaufwand für die Optimierung so gering wie möglich zu halten, werden ausschließlich *quadratische* Funktionale zur Optimierung eingesetzt.

Die in Kapitel 3 beschriebene neue Methode zur Lösung des funktionalen Interpolationsproblems beruht auf dem Ansatz einer TP-B-Spline Fläche mit weit mehr Freiheitsgraden, als dies zur Einhaltung der Interpolationsbedingungen nötig wäre. Mit einem *Ritz-Galerkin*-Ansatz werden diese überzähligen Freiheitsgrade dann zur Optimierung der Interpolationsfläche verwendet. Diese Methode generiert Interpolationsflächen von sehr guter Qualität, d.h. mit sehr gleichmäßigem Krümmungsverhalten.

In Kapitel 4 wurde ein neues Verfahren zur Lösung des parametrischen Interpolationsproblems basierend auf einem  $G^1$ -Kurvennetzansatz vorgestellt. Die hierbei gewählte Repräsentation des Kurvennetzes erlaubt es, daß ein Teil der unbestimmten Kurvennetzparameter unter Verwendung eines quadratischen Funktionals optimiert werden. Zum Ausfüllen der Kurvennetze wird beim  $G^1$ -Kurvennetz eine Variante des *Shirman-Séquin* Verfahrens verwendet, bei welcher ein Teil der freien Parameter durch geometrieabhängige Heuristiken bestimmt werden. Zudem ermöglicht eine verallgemeinerte Repräsentation der Splitfläche die Optimierung von Querableitungsvektoren entlang der Randkurven einer Splitfläche.

Die neue  $G^2$ -Kurvennetzmethode zur Generierung von Interpolationsflächen wurde in Abschnitt 5 diskutiert. Die Repräsentation des  $G^2$ -Kurvennetzes erlaubt ebenfalls die Optimierung eines Teils der Kurvennetzparameter. Die Öffnungen des  $G^2$ -Kurvennetzes werden mit einer einzigen Bézierfläche vom Grad 7 ausgefüllt. In diesem Fall können die drei innersten Bézier-Kontrollpunkte durch Ansatz eines quadratischen Funktionals optimiert werden.

Zudem wurde gezeigt, daß die beiden Schritte zur Optimierung des Kurvennetzes und der Flächenstücke für die  $G^1$ - und die  $G^2$ -Kurvennetzmethode auch zu einem *simultanen* Optimierungsschritt zusammengefaßt werden können.

Im Vergleich mit den bekannten Verfahren zur Lösung des parametrischen Interpolati-

onsproblems zeigt sich, daß die Qualität der Interpolationsflächen im Vergleich mit nicht optimierenden Methoden deutlich verbessert werden kann. Gegenüber der Methode von Moreton und Séquin ist andererseits ein deutlich günstigeres Laufzeitverhalten zu erkennen. Probleme können allerdings bei der Lösbarkeit der Matrizen, die sich aus den Optimierungsproblemen ableiten, ergeben. Dies gilt allerdings nur bei Ansatz von Funktionalen höherer Ordnung für das Kurvennetz und bei der simultanen Optimierung unter Verwendung des Laplace- und des Laplace-Beltrami Funktional, sowie des Funktional dritter Ordnung.

Die Repräsentation des  $G^1$ - und des  $G^2$ -Kurvennetzes erlaubt die Einbeziehung der Kurvennetzknotten in die Optimierung. Dadurch eröffnen sich eine Reihe von Anwendungen dieser Interpolationsmethoden, wie die Erweiterung auf Polyeder mit konvexen,  $n$ -eckigen Seitenflächen und die Approximation von Polyederecken. Die Interpolation von Bézierflächen, die zu ausgewählten Seiten des Polyeders gegeben sind, ist eine weitere Anwendungsmöglichkeit. Letztere ermöglicht die Behandlung von Blendingproblemen beliebiger Topologie, sowie das Ausfüllen von  $n$ -eckigen Löchern.

Der Einsatz des *datenabhängigen* Thin-Plate Funktional hat sich bei dem Ritz-Galerkin Ansatz zur Lösung des funktionalen Interpolationsproblems als sehr erfolgreich erwiesen. Im parametrischen Fall werden die verschiedenen datenabhängigen Funktionale mit unterschiedlichem Erfolg zur Optimierung der Kurvennetze und der Flächen verwendet. Eine generelle Empfehlung, in welcher Situation welches Funktional mit Sicherheit erfolgreich eingesetzt werden kann, ist nicht möglich. Die Beispiele belegen jedoch, daß im Vergleich zu dem Verfahren von Shirman und Séquin stets eine deutliche Verbesserung der Flächengüte erzielt werden kann, die dadurch meist sehr nahe an die Qualität der Moreton-Séquin Interpolationsflächen heranreicht.

## Ausblick

Trotz der Fortschritte bei der Behandlung des funktionalen und des parametrischen Interpolationsproblems, bleiben eine Reihe von offenen Problemen, die Gegenstand weiterer Forschung sein sollten.

Bei der momentanen Wahl der Knotenvektoren beim Ritz-Galerkin Ansatz haben *lokale* Verdichtungen der Datenpunkte einen *globalen* Einfluß. Der Einsatz hierarchischer Flächenstrukturen, wie der hierarchischen B-Splines von Forsey (vgl. [For90]), sollte in diesem Zusammenhang untersucht werden.

Die heuristischen Methoden zur Bestimmung lokaler Parameter für die  $G^1$ - und die  $G^2$ -Kurvennetzmethod arbeiten im allgemeinen gut. Dennoch sollten in diesem Zusammenhang weitere Methoden, z.B. Optimierungsansätze, erforscht werden.

Weitere Untersuchungen im Bereich der Repräsentation von Flächen mit beliebiger Topologie sind notwendig. Es zeigt sich, daß die Verfahren nach Peters und Loop (vgl. [Pet93, Pet94, Loo94a]) bei der Bestimmung der lokalen Bézier-Repräsentationen der (Interpolations)Flächen sehr effizient sind. Diese Methoden verhalten sich aber bei unregelmäßigen Polyedern zu „steif“ und generieren zu viele Bézierflächen.

# Anhang A

## Mathematischer Anhang

### A.1 Globale Lösbarkeit für den Ritz-Galerkin-Ansatz

In diesem Abschnitt wird die Lösbarkeit des globalen Interpolationsproblems bei dem Ritz-Galerkin-Ansatz bewiesen (vgl. Abschnitt 3 auf Seite 37).

**Satz A.1.1** *Sind alle lokalen Interpolationsprobleme  $(\mathcal{G}_I, F_I)$ ,  $I = (4k, 4l)$ ,  $F_I$  wie in Abschnitt 3.1 auf Seite 38, lösbar (d.h. für beliebige Wahl der Werte  $f_i$ ,  $i \in F_{(4k, 4l)}$ ), so besitzt auch das globale Interpolations eine Lösung.*

*Beweis.* Nach Voraussetzung ist das lokale Interpolationsproblem  $(\mathcal{G}_{(0,0)}, F_{(0,0)})$  lösbar. Da in unserem Fall  $\mathcal{G}_I \cap \mathcal{G}_J = \emptyset$  für  $I \neq J$  bleiben die schon bestimmten Kontrollpunkte  $b_J$ ,  $J \in \mathcal{G}_{(0,0)}$  bei Betrachtung aller weiteren Interpolationsprobleme unverändert. Für  $\mathcal{G}_{(4,0)}$  wird nun das Interpolationsproblem bzgl. der Datenwerte:

$$f_i^1 = \begin{cases} 0 & \text{if } i \in F_{(0,0)}, \\ f_i & \text{sonst} \end{cases},$$

betrachtet. Damit bleibt die Interpolationsbedingung auf den Datenpunkten  $\mathbf{u}_i$ ,  $i \in F_{(0,0)}$  weiterhin erfüllt. Sind alle lokalen Interpolationsprobleme gelöst worden, so interpoliert die resultierende TP-B-Splinefläche in alle Datenpunkte  $\mathbf{u}_i$  die entsprechenden Datenwerte  $f_i$ .

■

### A.2 Lösbarkeit der Kurvennetz-Optimierungsprobleme

Im folgenden werden einige Aussagen über die Lösbarkeit des Variationsproblems für das kubische  $G^1$ - und für das quintische  $G^2$ -Kurvennetz gemacht (vgl. Abschnitt 4.1.3 auf Seite 56 und 5.1.3 auf Seite 74).

### A.2.1 Lösbarkeit für das $G^1$ -Kurvennetz

Zunächst zum kubischen  $G^1$ -Kurvennetz. Das aus dem Variationsansatzes bzgl. des Funktionals  $\sigma$  abgeleitete Gleichungssystem ist genau dann regulär, wenn zum homogenen Problem nur die triviale Lösung existiert. Damit muß das Kurvennetz  $\mathbf{N}^H$  bestehend aus Kurven der Form:

$$\mathbf{C}_{ij}^H(u) = (a_{ij}S_{ui} + b_{ij}S_{vi}) \hat{\mathbf{w}}_i H_1^3(u) + (a_{ji}S_{uj} + b_{ji}S_{vj}) \hat{\mathbf{w}}_j H_1^3(1-u), \quad (\text{A.1})$$

und  $\sigma(\mathbf{N}^H) = 0$  überall verschwinden (vgl. Gleichung (4.6) auf Seite 55).

Wir verwenden im weiteren die Abkürzung  $A_{ij} = a_{ij}S_{ui} + b_{ij}S_{vi}$ . Zunächst treffen wir einige Aussagen über eine einzelne Kurve  $\mathbf{C}_{ij}^H$  des Kurvennetzes:

**Lemma A.2.1** (i) Für das simple Funktionals  $\sigma$  gilt (vgl. Gleichung (4.7) auf Seite 56):

$$\sigma(\mathbf{C}_{ij}^H) = 0 \iff \begin{cases} \mathbf{C}_{ij}^H \equiv 0 & \text{falls } \alpha \neq 0 \text{ oder } \beta \neq 0 \\ A_{ij}\hat{\mathbf{w}}_i = A_{ji}\hat{\mathbf{w}}_j & \text{falls } \alpha = \beta = 0 \end{cases}.$$

(ii) Wird das Laplace-Beltrami Funktional  $\sigma$  angesetzt, so gilt (vgl. Gleichung (4.8) auf Seite 57):

$$\sigma(\mathbf{C}_{ij}^H) = 0 \iff \mathbf{C}_{ij}^H \equiv 0.$$

(iii) Für das Funktional dritter Ordnung  $\sigma$  gilt (vgl. Gleichung (4.9) auf Seite 57):

$$\sigma(\mathbf{C}_{ij}^H) = 0 \iff \begin{cases} \mathbf{C}_{ij}^H \equiv 0 & \text{falls die Referenzkurve } \mathbf{C}_{ij}^0 \text{ nicht linear ist} \\ A_{ij}\hat{\mathbf{w}}_i = A_{ji}\hat{\mathbf{w}}_j & \text{sonst} \end{cases}.$$

*Beweis.*

zu (i): Für  $\alpha > 0$  oder  $\beta > 0$  folgt aus  $\sigma(\mathbf{C}_{ij}^H) = 0$  sofort  $\mathbf{C}_{ij}^{H''} \equiv 0$  bzw.  $\mathbf{C}_{ij}^{H'} \equiv 0$ . Damit muß  $\mathbf{C}_{ij}^H$  ein konstantes oder ein lineares Polynom sein. Da  $\mathbf{C}_{ij}^H$  für die Parameter  $u = 0$  und  $u = 1$  verschwindet, folgt  $\mathbf{C}_{ij}^H \equiv 0$ .

Analog wie eben erkennt man, daß im Falle  $\alpha = \beta = 0$   $\mathbf{C}_{ij}^H$  ein quadratisches Polynom ist. Ein Koeffizientenvergleich für  $t^3$  ergibt:  $A_{ij}\hat{\mathbf{w}}_i - A_{ji}\hat{\mathbf{w}}_j = 0$ .

zu (ii) Aus  $\sigma(\mathbf{C}_{ij}^H) = 0$  folgt sofort:  $\Delta_{\mathbf{C}_{ij}^0}(\mathbf{C}_{ij}^H) = 0$ . Für den Laplace-Beltrami Operator gilt:

$$\Delta_{\mathbf{C}_{ij}^0}(\mathbf{C}_{ij}^H) = \frac{1}{\sqrt{\langle \mathbf{C}_{ij}^{0'} | \mathbf{C}_{ij}^{0'} \rangle}} \left( \frac{\mathbf{C}_{ij}^{H'}}{\sqrt{\langle \mathbf{C}_{ij}^{0'} | \mathbf{C}_{ij}^{0'} \rangle}} \right)'. \quad (\text{A.2})$$

Somit gibt es einen konstanten Vektor  $\vec{\mathbf{k}}$  mit:  $\frac{\mathbf{C}_{ij}^{H'}}{\sqrt{\langle \mathbf{C}_{ij}^{0'} | \mathbf{C}_{ij}^{0'} \rangle}} = \vec{\mathbf{k}}$ .

Aufgrund des strikt positiven Nenners  $\sqrt{\langle \mathbf{C}_{ij}^{0'} | \mathbf{C}_{ij}^{0'} \rangle}$  müssen die einzelnen Komponenten von  $\mathbf{C}_{ij}^H$  monoton sind. Da  $\mathbf{C}_{ij}^H$  für  $u = 0$  und  $u = 1$  verschwindet, folgt:  $\mathbf{C}_{ij}^H \equiv 0$ .

zu (iii): Das Funktional dritter Ordnung ergibt sich für die Kurve  $\mathbf{C}_{ij}$  zu

$$\sigma(\mathbf{C}_{ij}) = \int \frac{1}{\sqrt{\langle \mathbf{C}_{ij}^{0'} | \mathbf{C}_{ij}^{0'} \rangle}} \left\langle \frac{d}{du} \left( \Delta_{\mathbf{C}_{ij}^0}(\mathbf{C}_{ij}) \right) \mid \frac{d}{du} \left( \Delta_{\mathbf{C}_{ij}^0}(\mathbf{C}_{ij}) \right) \right\rangle ds,$$

wobei  $\Delta_{\mathbf{C}_{ij}^0}$  den Laplace-Beltrami Operator bzgl. derselben Referenzkurve  $\mathbf{C}_{ij}^0$  bezeichnet. Für  $\sigma(\mathbf{C}_{ij}^H) = 0$  erhalten wir damit  $\Delta_{\mathbf{C}_{ij}^0}(\mathbf{C}_{ij}^H) = \vec{\mathbf{k}}$  für einen konstanten Vektor  $\vec{\mathbf{k}}$ . Eingesetzt ergibt sich

$$\frac{\mathbf{C}_{ij}^{H''}}{\langle \mathbf{C}_{ij}^{0'} | \mathbf{C}_{ij}^{0'} \rangle} - \frac{\mathbf{C}_{ij}^{H'} \langle \mathbf{C}_{ij}^{0''} | \mathbf{C}_{ij}^{0'} \rangle}{\langle \mathbf{C}_{ij}^{0'} | \mathbf{C}_{ij}^{0'} \rangle^2} = \vec{\mathbf{k}}. \quad (\text{A.3})$$

In unserem Falle ist die Referenzkurve  $\mathbf{C}_{ij}^0$  kubisch. Bringen wir die beiden Summanden in (A.3) auf den Hauptnenner, so ergibt sich der Grad des Nennerpolynoms *exakt* zu  $4(d^0 - 1)$ ,  $d^0 = \deg(\mathbf{C}_{ij}^0)$ . Der Grad des Zählerpolynoms hingegen ist *maximal*  $2d^0 + d - 3$ ,  $d^0 = \deg(\mathbf{C}_{ij}^0)$ ,  $d = \deg(\mathbf{C}_{ij}^H)$ . Da  $d \leq 3$  kann die linke Seite von (A.3) nur für  $d^0 = 1$  konstant sein.

Damit ist  $\mathbf{C}_{ij}^0$  linear und aus  $\sigma(\mathbf{C}_{ij}^H) = 0$  folgt sofort, daß  $\mathbf{C}_{ij}^H$  ein quadratisches Polynom ist mit  $A_{ij}\hat{\mathbf{w}}_i = A_{ji}\hat{\mathbf{w}}_j$ . Für  $\vec{\mathbf{k}} = 0$  ist das Problem analog zu dem beim Laplace-Beltrami Funktional. ■

Mit Hilfe der eben bewiesenen Eigenschaften für einzelne Kurven sind folgende Aussagen über die Lösbarkeit des aus dem Variationsansatzes bzgl. des Funktionals  $\sigma$  abgeleiteten Gleichungssystems möglich:

**Satz A.2.2** *Das Optimierungsproblem unter Verwendung des einfachen Funktionals mit  $\alpha > 0$  oder  $\beta > 0$  (vgl. Gleichung (4.7) auf Seite 56) und unter Ansatz des Laplace-Beltrami Funktionals (vgl. Gleichung (4.8) auf Seite 57) ist stets eindeutig lösbar.*

*Beweis.* Folgt unmittelbar aus Lemma A.2.1 (i) und (ii), da an jeder Ecke  $\mathbf{V}_i$  mindestens zwei Kurven  $\mathbf{C}_{ij_1}^H$  und  $\mathbf{C}_{ij_1}^H$  anliegen mit linear unabhängigen Vektoren  $(a_{ij_1}, b_{ij_1}), (a_{ij_2}, b_{ij_2})$ . ■

Eine vollständige Charakterisierung der Lösbarkeit bei dem einfachen Funktional mit  $\alpha = \beta = 0$  und des Funktionals dritter Ordnung ist schwierig. Wir geben im folgenden eine *hinreichende* Bedingung für die Lösbarkeit an. Dazu zunächst folgendes Lemma.

**Bezeichnung/Lemma A.2.3** (i) *Gehen von der Ecke  $\mathbf{V}_i$  zwei Kurven des Kurvennetzes  $\mathbf{C}_{ij_k}, k = 1, 2$  aus mit linear unabhängigen Parameterrichtungen  $\vec{\mathbf{t}}_{ij_k}, k = 1, 2$  und mit*

- (a)  $\hat{\mathbf{w}}_i \neq \hat{\mathbf{w}}_{j_k}, k = 1, 2$  für das einfache Funktional mit  $\alpha = \beta = 0$  bzw.
- (b)  $\hat{\mathbf{w}}_i \neq \hat{\mathbf{w}}_{j_k}$  oder  $\mathbf{C}_{ij_k}^0$  nicht linear  $k = 1, 2$  für das Funktional dritter Ordnung,

so ist  $S_{ui} = S_{vi} = 0$ . In diesem Fall bezeichnen wir  $\mathbf{V}_i$  als stabil.

(ii) *Ist die Ecke  $\mathbf{V}_i$  stabil, so gilt für jede Kurve  $\mathbf{C}_{ij}^H$  an dieser Ecke  $A_{ij} = A_{ji} = 0$  unabhängig von dem angesetzten Funktional.*

*Beweis.*

zu (i): Aus Lemma A.2.1 folgt unter den gegebenen Bedingungen stets  $A_{ij} = A_{ji} = 0$ . Da  $\vec{t}_{i_jk}, k = 1, 2$  linear unabhängig sind folgt aus  $A_{ij_1} = A_{ij_2} = 0$  die Behauptung.

zu (ii): Ist  $\mathbf{V}_i$  stabil, so gilt  $A_{ij} = 0$  für jede Kurve  $\mathbf{C}_{ij}^H$  bei  $\mathbf{V}_i$ . Da  $\mathbf{C}_{ij}^H$  entweder verschwindet oder der Bedingung  $A_{ij}\hat{\mathbf{w}}_i = A_{ji}\hat{\mathbf{w}}_j$  genügt, gilt immer  $\mathbf{C}_{ij}^H \equiv 0$ . ■

Man erkennt sofort, daß das aus der Optimierung resultierende Gleichungssystem genau dann regulär ist, wenn alle Ecken  $\mathbf{V}_i$  stabil sind.

Mit Hilfe des eben bewiesenen Lemmas kann folgende hinreichende geometrische Bedingung für die Lösbarkeit bei Verwendung des einfachen Funktional mit  $\alpha = \beta = 0$  bzw. des Funktional dritter Ordnung angegeben werden.

**Satz A.2.4** *Vorausgesetzt das durch  $S_u = S_v = 0$  gegebene Kurvennetz ist winkel-regulär (vgl. Bezeichnung 2.2.1 auf Seite 25).*

*Ist das lineare Gleichungssystem für das Optimierungsproblem mit dem einfachen Funktional mit  $\alpha = \beta = 0$  bzw. mit dem Funktional dritter Ordnung singular, so gibt es eine Sequenz  $\mathcal{S} = (\mathbf{V}_{i_k})_{k=0}^l$  von nicht stabilen Ecken mit*

- die Kante  $\overline{\mathbf{V}_{i_j}\mathbf{V}_{i_{j+1}}}$  hat die Eigenschaft (ia) bzw. (ib) aus Lemma A.2.3,
- $\mathcal{S}$  ist geschlossen oder  $\mathbf{V}_{i_0}$  und  $\mathbf{V}_{i_l}$  liegen auf dem Rand des Polyeders.

*Beweis.* Jedes nicht triviale Kurvennetz  $\mathbf{N}^H$  mit  $\sigma(\mathbf{N}^H) = 0$  ist dadurch gekennzeichnet, daß nicht alle partiellen Ableitungen  $S_u, S_v$  verschwinden. Damit gibt es mindestens eine nicht stabile Ecke  $\mathbf{V}_{i_0}$ . Da das Kurvennetz winkel-regulär ist, gibt es bei  $\mathbf{V}_{i_0}$  eine Kurve  $\mathbf{C}_{i_0i_1}^H \not\equiv 0$ . Aus (ii) von Lemma A.2.3 folgt, daß  $\mathbf{V}_{i_1}$  auch nicht stabil ist. Liegt  $\mathbf{V}_{i_1}$  nicht auf dem Rand, so folgt aus dem Winkelregularität, daß es noch eine weitere Kante  $\overline{\mathbf{V}_{i_1}\mathbf{V}_{i_2}}$  bei  $\mathbf{V}_{i_1}$  gibt mit  $\mathbf{C}_{i_1i_2}^H \not\equiv 0$ . Dieses Argument läßt sich fortsetzen, solange nur Punkte betrachtet werden, die nicht auf dem Rand liegen. Liegt  $\mathbf{V}_{i_1}$  auf dem Rand des Polyeders, so kann  $\mathbf{C}_{i_0i_1}^H$  die einzige Kurve  $\not\equiv 0$  bei  $\mathbf{V}_{i_1}$  sein („T“-Form). Damit gibt es eine Eckensequenz  $(\mathbf{V}_{i+k})_{k=0}^l$  mit den obigen Eigenschaften. ■

Somit sind unter Umständen Probleme, bei denen alle Ecken  $\mathbf{V}_i$  in einer Hyperebene des  $\mathbb{R}^3$  liegen, mit dem einfachen Funktional für  $\alpha = \beta = 0$  oder mit dem Funktional dritter Ordnung nicht lösbar. In diesem Fall sind die Vektoren  $\hat{\mathbf{w}}_i$  alle gleich. Sind die Ecken allerdings mehr zufällig verteilt, so treten in der Praxis keine Schwierigkeiten auf.

## A.2.2 Lösbarkeit für das $G^2$ Kurvennetz

Die Lösbarkeit des Optimierungsproblems beim  $G^2$  Kurvennetzes beruht auf der Betrachtung der entsprechenden homogenen Formen des Kurvennetzes und des Querableitungsnetzes  $\mathbf{N}^H$  bzw.  $\vec{\mathbf{Q}}^H$ .

Eine Kurve  $\mathbf{C}_{ij}^H$  des homogenen Kurvennetzes und eine Querableitung  $\vec{\mathbf{q}}_{ij}^H$  des homogenen Querableitungsnetzes haben die Form:

$$\begin{aligned}\mathbf{C}_{ij}^H(u) &= A_{ij}\hat{\mathbf{w}}_i H_1^5(u) + B_{ij}\hat{\mathbf{w}}_i H_2^5(u) + B_{ji}\hat{\mathbf{w}}_j H_2^5(1-u) + A_{ji}\hat{\mathbf{w}}_j H_2^5(1-u), \\ \mathbf{q}_{ij}^H(u) &= E_{ij}\hat{\mathbf{w}}_i H_0^3(u) + F_{ij}\hat{\mathbf{w}}_i H_1^3(u) + F_{ji}\hat{\mathbf{w}}_j H_1^3(1-u) + E_{ji}\hat{\mathbf{w}}_j H_0^3(1-u),\end{aligned}$$

mit

$$\begin{aligned}A_{ij} &= a_{ij}S_{ui} + b_{ij}S_{vi}, & B_{ij} &= c_{ij}S_{ui} + d_{ij}S_{vi} + (a_{ij}, b_{ij}) \begin{pmatrix} S_{uvi} & S_{uvi} \\ S_{uvi} & S_{vvi} \end{pmatrix} \begin{pmatrix} a_{ij} \\ b_{ij} \end{pmatrix}, \\ E_{ij} &= e_{ij}S_{ui} + f_{ij}S_{vi}, \text{ und} & F_{ij} &= (e_{ij}, f_{ij}) \begin{pmatrix} S_{uvi} & S_{uvi} \\ S_{uvi} & S_{vvi} \end{pmatrix} \begin{pmatrix} a_{ij} \\ b_{ij} \end{pmatrix}.\end{aligned}$$

Zunächst ein Lemma das, in analoger Weise zu Lemma A.2.1, eine Aussage über eine einzelne Kurve  $\mathbf{C}_{ij}^H$  und über ein einzelnes Querableitungsvektorfeld  $\vec{\mathbf{q}}_{ij}^H$  macht:

**Lemma A.2.5** *Betrachten wir zunächst die Kurve  $\mathbf{C}_{ij}^H$  des homogenen Kurvennetzes.*

(i) *Für das einfache Funktional  $\sigma$  (vgl. Gleichung (4.7) auf Seite 56) gilt:*

$$\begin{aligned}\sigma(\mathbf{C}_{ij}^H) &= 0 \\ \iff &\begin{cases} \mathbf{C}_{ij}^H \equiv 0 & \text{falls } \alpha \neq 0 \text{ oder } \beta \neq 0 \\ -2A_{ij}\hat{\mathbf{w}}_i = -2A_{ji}\hat{\mathbf{w}}_j = B_{ij}\hat{\mathbf{w}}_i = B_{ji}\hat{\mathbf{w}}_j & \text{falls } \alpha = \beta = 0 \end{cases}. \quad (\text{A.4})\end{aligned}$$

(ii) *Wird das Laplace-Beltrami Funktional  $\sigma$  (vgl. Gleichung (4.8) auf Seite 57) angesetzt, so gilt:*

$$\sigma(\mathbf{C}_{ij}^H) = 0 \iff \mathbf{C}_{ij}^H \equiv 0.$$

(iii) *Für das Funktional dritter Ordnung  $\sigma$  (vgl. Gleichung (4.9) auf Seite 57) gilt:*

$$\begin{aligned}\sigma(\mathbf{C}_{ij}^H) &= 0 \\ \iff &\begin{cases} \mathbf{C}_{ij}^H \equiv 0 & \text{falls } \mathbf{C}_{ij}^0 \text{ nicht linear ist} \\ -2A_{ij}\hat{\mathbf{w}}_i = -2A_{ji}\hat{\mathbf{w}}_j = B_{ij}\hat{\mathbf{w}}_i = B_{ji}\hat{\mathbf{w}}_j & \text{sonst} \end{cases}. \quad (\text{A.5})\end{aligned}$$

Für das Querableitungsvektorfeld  $\vec{\mathbf{q}}_{ij}^H$  des homogenen Querableitungsnetzes erhalten wir:

(iv) *Für das einfache Funktional  $\sigma$  gilt (vgl. Gleichung (5.8) auf Seite 75)*

$$\sigma(\vec{\mathbf{q}}_{ij}^H) = 0 \implies \begin{cases} F_{ij} = F_{ji} = 0 & \text{falls } \alpha \neq 0 \text{ oder } \beta \neq 0 \\ F_{ij}\hat{\mathbf{w}}_i = -F_{ji}\hat{\mathbf{w}}_j = E_{ji}\hat{\mathbf{w}}_j - E_{ij}\hat{\mathbf{w}}_i & \text{falls } \alpha = \beta = 0 \end{cases}$$

(v) *Für das auf dem Operator  $L_{\vec{\mathbf{q}}}$  aufbauende Funktional  $\sigma$  (vgl. Gleichung (5.10) auf Seite 75) gilt*

$$\sigma(\vec{\mathbf{q}}_{ij}^H) = 0 \implies F_{ij} = F_{ji} = 0.$$

*Beweis.* Der Beweis der Aussagen (i)–(iii) für die Kurve  $\mathbf{C}_{ij}^H$  ist sehr ähnlich zu den entsprechenden Beweisen bei Lemma A.2.1. Wir werden deshalb nur auf die wesentlichen Unterschiede eingehen.

zu (i): Für  $\alpha \neq 0$  oder  $\beta \neq 0$  ist der Beweis völlig analog zu dem von Lemma A.2.1 (i) .

Für  $\alpha = \beta = 0$  muß  $\mathbf{C}_{ij}^H$  ein quadratisches Polynom sein. Damit folgt unmittelbar:

$$\begin{aligned} \mathbf{C}_{ij}^{H''}(u) &= \text{const} = \mathbf{C}_{ij}^{H'}(1) - \mathbf{C}_{ij}^{H'}(0) \\ \implies B_{ij}\hat{\mathbf{w}}_i &= B_{ji}\hat{\mathbf{w}}_j = -(A_{ji}\hat{\mathbf{w}}_j + A_{ij}\hat{\mathbf{w}}_i). \end{aligned} \quad (\text{A.6})$$

Da  $\mathbf{C}_{ij}^H(0) = \mathbf{C}_{ij}^H(1) = 0$  gilt zudem:

$$\mathbf{C}_{ij}^{H'}(0) = -\mathbf{C}_{ij}^{H'}(1) \iff A_{ij}\hat{\mathbf{w}}_i = A_{ji}\hat{\mathbf{w}}_j.$$

Zusammen mit Gleichung (A.6) sind dies drei unabhängige, notwendige Bedingungen an die Koeffizienten des quintischen Polynoms  $\mathbf{C}_{ij}^H$ . Diese Bedingungen sind auch hinreichend um den Grad von  $\mathbf{C}_{ij}^H$  auf zwei zu beschränken.

zu (ii): Wie im Fall des  $G^1$ -Kurvennetzes erhalten wir folgende Bedingung (vgl. Lemma A.2.1 (A.2)). Es existiert ein konstanter Vektor  $\vec{\mathbf{k}}$  mit:  $\frac{\mathbf{C}_{ij}^{H'}}{\sqrt{\langle \mathbf{C}_{ij}^{0'} | \mathbf{C}_{ij}^{0'} \rangle}} = \vec{\mathbf{k}}$ .

Der Nenner  $\sqrt{\langle \mathbf{C}_{ij}^{0'} | \mathbf{C}_{ij}^{0'} \rangle}$  ist strikt positiv, womit die Komponenten von  $\mathbf{C}_{ij}^H$  monoton sind. Da  $\mathbf{C}_{ij}^H(0) = \mathbf{C}_{ij}^H(1) = 0$  folgt unmittelbar:  $\mathbf{C}_{ij}^H \equiv 0$ .

zu (iii): Wie beim  $G^1$ -Kurvennetz erhalten wir für das Funktional dritter Ordnung  $\sigma$  angewendet auf die Kurve  $\mathbf{C}_{ij}^H$  (vgl. Gleichung Gleichung (A.3) auf Seite 121):

$$\sigma(\mathbf{C}_{ij}^H) \equiv 0 \iff \Delta_{\mathbf{C}_{ij}^0}(\mathbf{C}_{ij}^H) = \vec{\mathbf{k}} \quad \text{mit konstantem Vektor } \vec{\mathbf{k}}.$$

Durch Auswertung von  $\Delta_{\mathbf{C}_{ij}^0}(\mathbf{C}_{ij}^H)$  bei  $u = 0$  bzw.  $u = 1$  ergibt sich ein Vektor in Richtung von  $\hat{\mathbf{w}}_i$  bzw.  $\hat{\mathbf{w}}_j$ . Damit folgt für  $\hat{\mathbf{w}}_i \neq \hat{\mathbf{w}}_j$  :  $\vec{\mathbf{k}} = 0$ . Das betrachtete Problem ist also äquivalent zu  $\Delta_{\mathbf{C}_{ij}^0}(\mathbf{C}_{ij}^H) \equiv 0$  wodurch mit (ii) folgt:  $\mathbf{C}_{ij}^H \equiv 0$ .

Ist  $\hat{\mathbf{w}}_i = \hat{\mathbf{w}}_j$ , so ergibt ein Gradvergleich des Zähler- und des Nennerpolynoms in (A.3), daß  $\mathbf{C}_{ij}^0$  entweder linear oder quadratisch sein kann. Für lineares  $\mathbf{C}_{ij}^0$  haben wir dieselbe Situation wie in (i)  $\alpha = \beta = 0$ , d.h. es muß gelten:  $\mathbf{C}_{ij}^{H''} = 0$ . Ist  $\mathbf{C}_{ij}^0$  quadratisch, so gilt aufgrund von  $\mathbf{C}_{ij}^{0'}(0) \parallel \hat{\mathbf{w}}_i$  und  $\mathbf{C}_{ij}^{0'}(1) \parallel \hat{\mathbf{w}}_j$ :  $\mathbf{C}_{ij}^{0''} \parallel \hat{\mathbf{w}}_i$ , womit  $\mathbf{C}_{ij}^0$  in der Ebene  $(\mathbf{V}_i, \hat{\mathbf{w}}_i) = (\mathbf{V}_j, \hat{\mathbf{w}}_j)$  laufen muß. Alle in Abschnitt 5.1.2 auf Seite 74 verwendeten Methoden zur Bestimmung der lokalen Parameter liefern in diesem Falle aber  $\mathbf{C}_{ij}^0(u) = (1-u)\mathbf{V}_i + u\mathbf{V}_j$ . Damit kann  $\mathbf{C}_{ij}^0$  also nicht tatsächlich vom Grad 2, muß also linear sein.

Nun zu den Beweisen der Aussagen über das Vektorfeld  $\vec{\mathbf{q}}_{ij}^H$  des homogenen Querableitungsnetzes  $\vec{\mathbf{Q}}^H$ :

zu (iv): Ist  $\sigma(\vec{\mathbf{q}}_{ij}^H) = 0$  für  $\alpha \neq 0$  oder  $\beta \neq 0$ , so ist  $\vec{\mathbf{q}}_{ij}^H(u)$  ein konstantes oder ein verschwindendes Vektorfeld. In beiden Fällen ist damit:  $F_{ij}\hat{\mathbf{w}}_i = F_{ji}\hat{\mathbf{w}}_j = 0$ , also  $F_{ij} = F_{ji} = 0$ .

Für  $\alpha = \beta = 0$  ist  $\vec{\mathbf{q}}_{ij}^H$  ein lineares Vektorfeld, woraus unmittelbar folgt, daß die Ableitung von  $\vec{\mathbf{q}}_{ij}^H$  durch  $E_{ji}\hat{\mathbf{w}}_j - E_{ij}\hat{\mathbf{w}}_i$  bestimmt ist. Damit gilt:  $F_{ij}\hat{\mathbf{w}}_i = -F_{ji}\hat{\mathbf{w}}_j = E_{ji}\hat{\mathbf{w}}_j - E_{ij}\hat{\mathbf{w}}_i$ . zu (v): Für das Funktional  $\sigma$  basierend auf dem Operator  $L_{\vec{\mathbf{q}}_{ij}^0}$  folgt aus  $\sigma(\vec{\mathbf{q}}_{ij}) = 0$  (vgl. Gleichung (5.10) auf Seite 75):

$$L_{\vec{\mathbf{q}}_{ij}^0}(\vec{\mathbf{q}}_{ij}^H) = 0 \iff \vec{\mathbf{q}}_{ij}^{H'} \langle \vec{\mathbf{q}}_{ij}^0 \mid \vec{\mathbf{q}}_{ij}^0 \rangle = \vec{\mathbf{q}}_{ij}^H \langle \vec{\mathbf{q}}_{ij}^{0'} \mid \vec{\mathbf{q}}_{ij}^0 \rangle.$$

Aufgrund der Wahl von  $\vec{\mathbf{q}}_{ij}^0$  gilt:  $\vec{\mathbf{q}}_{ij}^0(0) \in \text{span}\{\hat{\mathbf{u}}_i, \hat{\mathbf{v}}_i\}$  und  $\vec{\mathbf{q}}_{ij}^{0'}(0) \parallel \hat{\mathbf{w}}_i$ . Damit ist für  $u = 0$   $\langle \vec{\mathbf{q}}_{ij}^{0'} \mid \vec{\mathbf{q}}_{ij}^0 \rangle = 0$ . Somit muß für  $u = 0$  auch  $\vec{\mathbf{q}}_{ij}^{H'}$  verschwinden, also:  $F_{ij} = 0$ . Analog folgt  $F_{ji} = 0$ . ■

Zur Lösbarkeit des aus dem Optimierungsproblems abgeleiteten linearen Gleichungssystems kann für das  $G^2$  Kurvennetz die zu Satz A.2.2 analoge Aussage getroffen werden:

**Satz A.2.6** *Das Optimierungsproblem für das  $G^2$  Kurvennetz unter Ansatz des einfachen Funktionals mit  $\alpha > 0$  oder  $\beta > 0$  zur Bewertung der Kurven und Querableitungen (vgl. Gleichung (4.7) auf Seite 56 und (5.8) auf Seite 75) und unter Ansatz des Laplace-Beltrami Funktionals und des Funktionals basierend auf dem Operator  $L_{\vec{\mathbf{q}}_{ij}^0}$  (vgl. Gleichung (4.8) auf Seite 57 und (5.10) auf Seite 75) ist stets eindeutig lösbar.*

*Beweis.* Da an jeder Ecke  $\mathbf{V}_i$  mindestens zwei Kurven  $\mathbf{C}_{ij_k}^H$  mit linear unabhängigen Vektoren  $(a_{ij_1}, b_{ij_1}), (a_{ij_2}, b_{ij_2})$  anliegen, verschwinden aufgrund der Voraussetzung an das Funktional die partiellen Ableitungen  $S_{ui}, S_{vi}$ . Für die partiellen Ableitungen  $S_{uui}, S_{uvi}$  und  $S_{vvi}$  ergeben sich mit  $S_{ui} = S_{vi} = 0$  durch

$$B_{ij_k} = 0 \iff (a_{ij_k}, b_{ij_k}) \begin{pmatrix} S_{uui} & S_{uvi} \\ S_{uvi} & S_{vvi} \end{pmatrix} \begin{pmatrix} a_{ij_k} \\ b_{ij_k} \end{pmatrix} = 0, \quad k = 1, 2,$$

$$F_{ij_k} = 0 \iff (e_{ij_k}, f_{ij_k}) \begin{pmatrix} S_{uui} & S_{uvi} \\ S_{uvi} & S_{vvi} \end{pmatrix} \begin{pmatrix} a_{ij_k} \\ b_{ij_k} \end{pmatrix} = 0, \quad k = 1, 2,$$

vier unabhängige Bedingungen. Damit müssen auch die zweiten partiellen Ableitungen allesamt verschwinden. Das homogene Problem hat also nur die triviale Lösung. ■

Durch analoge Schlußweise wie im Beweis für Satz A.2.6 können hinreichende Aussagen über die Lösbarkeit des Optimierungsproblems unter Verwendung des einfachen Funktionals mit  $\alpha = \beta = 0$  und des Funktionals dritter Ordnung für das  $G^1$ -Kurvennetz (vgl. Lemma A.2.3 und Satz A.2.4) auf das  $G^2$  Kurvennetz getroffen werden.

## A.3 Transformation der inneren Produktmatrix

In diesem Abschnitt beweisen wir folgende Transformationseigenschaft von dem inneren Produkt  $\langle \cdot \mid \cdot \rangle_\sigma$  bzgl. eines quadratischen Funktionals  $\sigma$  (vgl. Abschnitt 4.2).

**Satz A.3.1** Betrachten wir den linearen Funktionenraum  $\mathcal{B} = \text{span}\{B_1, \dots, B_n\}$  und einen Unterraum  $\mathcal{C} = \text{span}\{C_1, \dots, C_k\}$ ,  $k \leq n$  von  $\mathcal{B}$ .

Sei  $T = (t_{ij})_{i,j}$  die Transformationsmatrix für die Koeffizienten von Funktionen  $F \in \mathcal{C}$ , d.h.

$$F = \sum_{i=1}^k c_i C_i = \sum_{i=1}^n b_i B_i \quad \text{mit} \quad T \begin{pmatrix} c_0 \\ \vdots \\ c_k \end{pmatrix} = \begin{pmatrix} b_0 \\ \vdots \\ b_n \end{pmatrix}.$$

Dann gilt für die innere Produktmatrix

$$(\langle C_i | C_j \rangle)_{i,j=1,\dots,k} = T^t (\langle B_i | B_j \rangle)_{i,j=1,\dots,n} T. \quad (\text{A.7})$$

*Beweis.*

$$\begin{aligned} \langle C_i | C_j \rangle &= \left\langle \sum_{k=1}^n t_{ki} B_k \mid \sum_{l=1}^n t_{lj} B_l \right\rangle = \sum_{(k,l)}^{(n,n)} t_{ki} t_{lj} \langle B_k | B_l \rangle \\ &= (t_{0i}, \dots, t_{ni}) (\langle B_k | B_l \rangle)_{k,l=1,\dots,n} \begin{pmatrix} t_{0j} \\ \vdots \\ t_{nj} \end{pmatrix}. \end{aligned}$$

■

# Anhang B

## Tabellen

Die nachstehenden Tabellen geben Auskunft über die Parameter sowie über das Laufzeitverhalten<sup>1</sup> der einzelnen Verfahren, wie sie bei den Beispielen in den Abschnitten 3.3 auf Seite 43, 5.4 auf Seite 82 und 6.4 auf Seite 105 ermittelt wurden.

Folgende Begriffe werden in der **Parameter**-Spalte und in der **Laufzeit**-Spalte verwendet:

Parameter-Spalte															
Funktional	<p>Das zur Optimierung des Kurvennetzes bzw. der Fläche verwendete Funktional:</p> <table border="1"> <tr> <td>„ohne“</td> <td>keine Optimierung</td> </tr> <tr> <td>„simple“</td> <td>einfaches Funktional</td> </tr> <tr> <td>„Lapl“</td> <td>Laplace Funktional</td> </tr> <tr> <td>„ThinPl“</td> <td>Thin-Plate Funktional</td> </tr> <tr> <td>„LapBelt“</td> <td>Laplace-Beltrami</td> </tr> <tr> <td>„DatThin“</td> <td>datenabhängiges Thin-Plate Funktional</td> </tr> <tr> <td>„Dritte“</td> <td>Funktional dritter Ordnung</td> </tr> </table> <p>Die Zahl in der Iterationsschritte ist in Klammern angegeben; „S“ bedeutet simultane Optimierung.</p>	„ohne“	keine Optimierung	„simple“	einfaches Funktional	„Lapl“	Laplace Funktional	„ThinPl“	Thin-Plate Funktional	„LapBelt“	Laplace-Beltrami	„DatThin“	datenabhängiges Thin-Plate Funktional	„Dritte“	Funktional dritter Ordnung
„ohne“	keine Optimierung														
„simple“	einfaches Funktional														
„Lapl“	Laplace Funktional														
„ThinPl“	Thin-Plate Funktional														
„LapBelt“	Laplace-Beltrami														
„DatThin“	datenabhängiges Thin-Plate Funktional														
„Dritte“	Funktional dritter Ordnung														
Gitter	Zahl der Gitterlinien bei Arge, Dæhlen und Tveito.														
Global	Der globale Shape-Parameter beim Verfahren nach Nielson.														
Shift	Bestimmung der Ordinatenwerte für nahen Gitterpunkte nach der Shift-Methode														
Tangente	Verfahren zur Bestimmung der Tangentenlänge ( $G^1$ - bzw. $G^2$ -Kurvennetz).														
Unterteilung	Ansatz der Blöcke von $4 \times 4$ Knoten beim Ritz-Galerkin in $x$ - und $y$ -Richtung														
Zweite	Methode zur Bestimmung der lokalen Parameterrichtung der zweiten Ableitung in den Endpunkten ( $G^2$ Kurvennetz).														

<sup>1</sup>Die Beispiele wurden alle auf einer SGI<sup>TM</sup> Indigo mit R4400 Prozessor und 128MB Hauptspeicher gerechnet. Eine Ausnahme bilden die Gesichtsdaten (Tabelle B.10 und B.13), die auf einer SGI Power ONYX mit zwei R8000 Prozessoren und ebenfalls 128 MB Hauptspeicher gerechnet wurden.

Laufzeit-Spalte	
Ableitung	Bestimmung bzw. Abschätzung von Ableitungen.
Init	Initialisierung von Datenstrukturen und evtl. Vorverarbeitungsschritte.
LGS	Lösen des linearen Gleichungssystems zur Optimierung.
Matrix	Aufstellen der Optimierungsmatrix.

## Tabellen zu funktionalen Interpolationsmethoden

Verfahren	Parameter	Laufzeit	Bemerkungen
Thin Plate Spline		Matrix: 4.60 sec	
Arge, Dæhlen und Tveito	Gitter: 75; Shift	Init: 4.87 sec Matrix: 3.09 sec LGS: 189.22 sec Total: 197.18 sec	
Ritz-Galerkin Ansatz	Unterteilung: (8,8); Funktional: ThinPl	Init/Matrix: 3.41 sec LGS: 8.87 sec Total: 12.28 sec	
Ritz-Galerkin Ansatz	Unterteilung: (8,8); Funktional: DatThin	Init/Matrix: 323.43 sec LGS: 30.83 sec Total: 354.26 sec	

Tabelle B.1: Tabelle zu Abbildungen 3.4 auf Seite 47 und 3.5 auf Seite 49; Interpolationsflächen zur Testfunktion  $S_1$  von Lyche und Morken.

Verfahren	Parameter	Laufzeit	Bemerkungen
Thin Plate Spline		Matrix: 14.40 sec	
Arge, Dæhlen und Tveito	Gitter: 75; Shift	Init: 4.83 sec Matrix: 3.06 sec LGS: 227.09 sec Total: 234.98 sec	
Ritz-Galerkin Ansatz	Unterteilung: (8,8); Funktional: ThinPl	Init/Matrix: 5.06 sec LGS: 8.82 sec Total: 13.88 sec	
Ritz-Galerkin Ansatz	Unterteilung: (8,8); Funktional: DatThin	Init/Matrix: 303.73 sec LGS: 41.83 sec Total: 345.56 sec	

Tabelle B.2: Tabelle zu Abbildungen 3.5 auf Seite 49; Interpolationsflächen zur Testfunktion  $S_2$  von Franke.

Verfahren	Parameter	Laufzeit	Bemerkungen
Thin Plate Spline		Matrix: 71.70 sec	
Arge, Dæhlen und Tveito	Gitter: 100; Shift	Init: 6.32 sec Matrix: 4.52 sec LGS: 181.03 sec Total: 191.87 sec	
Ritz-Galerkin Ansatz	Unterteilung: (12,12); Funktional: ThinPl	Init/Matrix: 4.41 sec LGS: 50.86 sec Total: 55.25 sec	
Ritz-Galerkin Ansatz	Unterteilung: (8,8); Funktional: DatThin	Init/Matrix: 733.31 sec LGS: 153.01 sec Total: 886.32 sec	

Tabelle B.3: Tabelle zu Abbildungen 3.6 auf Seite 51; Interpolationsflächen zur Testfunktion  $S_3$  von Ritchie.

## Tabellen zu parametrischen Interpolationsmethoden

Verfahren	Parameter	Laufzeit	Bemerkungen
Moreton und Séquin		Kurven ca 20 min Flächen ca 20 std Total: ca 20.3 std	
Shirman und Séquin		Kurven 0.05 sec Flächen 0.31 sec Total: 0.36 sec	
Nielson	Global: 1.10	Kurven: Matrix 0.01 sec LGS 0.09 sec	Verbesserte Bestimmung der Tangenten (vgl. Abschnitt 2.2.2)
		Flächen: 0.08 sec Total: 0.18 sec	

Tabelle B.4: Tabelle zu Abbildungen 5.8 auf Seite 85; Interpolationsflächen zu den Daten einer Sphäre.

Verfahren	Parameter	Laufzeit	Bemerkungen
Loop		Total: 0.54 sec	Interpolation ges.: 288 Flächen
$G^1$ -Kurvennetz Methode	Kurven:		
	Tangente: EK, KS; Funktional: simple(2)	Init: 0.02 sec Matrix: 0.02 sec LGS: 0.16 sec	
	Flächen:		
	Funktional: ohne	0.31 sec	
		Total: 0.51 sec	
$G^2$ -Kurvennetz Methode	Kurven:		
	Tangente: EK, KS; Zweite: N; Funktional: simple(2)	Init: 0.01 sec Matrix: 0.13 sec LGS: 0.30 sec	
	Flächen:		
	Funktional: ohne	2.24 sec	
		Total: 2.68 sec	

Tabelle B.5: Tabelle zu Abbildungen 5.9 auf Seite 87; Interpolationsflächen zu den Daten einer Sphäre.

Verfahren	Parameter	Laufzeit	Bemerkungen
Moreton und Séquin		Kurven ca 45 min	
		Flächen ca 50 std	
		Total: ca 50.75 std	
Shirman und Séquin		Kurven 0.17 sec	
		Flächen 2.91 sec	
		Total: 3.08 sec	
Nielson	Global: 1	Kurven:	
		Matrix: 0.05 sec LGS: 3.39 sec	Verbesserte Bestimmung der Tangenten (vgl. Abschnitt 2.2.2)
	Flächen:		
	0.90 sec		
		Total: 4.34 sec	

Tabelle B.6: Tabelle zu Abbildungen 5.10 auf Seite 89; Interpolationsflächen zu den Torus-Daten.

Verfahren	Parameter	Laufzeit	Bemerkungen
Loop		Total: 2.23 sec	Interpolation ges.: 2400 Flächen
$G^1$ -Kurvennetz Methode	Kurven:		
	Tangente: EK, KS; Funktional: simple(2)	Init: 0.13 sec Matrix: 0.21 sec LGS: 0.99 sec	
	Flächen:		
	Funktional: Lapl	Init: 0.61 sec Matrix: 6.22 sec LGS: 1.49 sec Total: 9.65 sec	
$G^2$ -Kurvennetz Methode	Kurven:		
	Tangente: EK, KS; Zweite: N; Funktional: simple(2)	Init: 0.14 sec Matrix: 0.79 sec LGS: 1.56 sec	
	Flächen:		
	Funktional: ohne	9.82 sec Total: 12.31 sec	

Tabelle B.7: Tabelle zu Abbildung 5.11 auf Seite 91; Interpolationsflächen zu den Torus-Daten.

Verfahren	Parameter	Laufzeit	Bemerkungen
Shirman und Séquin		Kurven 0.29 sec Flächen 5.49 sec Total: 5.78 sec	
	Kurven:		# OptParam <sup>2</sup> : 382
	Tangente: EK, KS; Funktional: simple(3)	Init: 0.24 sec Matrix: 0.53 sec LGS: 1.37 sec	
$G^1$ -Kurvennetz Methode	Flächen:		# OptParam: 1975
	Funktional: ThinPl(1;S)	Init: 1.13 sec Matrix: 10.45 sec LGS: 17.34 sec Total: 31.06 sec	
$G^2$ -Kurvennetz Methode	Kurven:		# OptParam: 955
	Tangente: EK, KS; Zweite: N; Funktional: LapBelt(3)	Init: 0.25 sec Matrix: 5.49 sec LGS: 3.15 sec	
	Flächen:		# OptParam: 4024
	Funktional: Dritte(1;S)	Init: 4.76 sec Matrix: 184.39 sec LGS: 66.10 sec Total: 266.14 sec	

Tabelle B.8: Tabelle zu Abbildung 5.13 auf Seite 95; Die erste CAD-Fläche.

<sup>2</sup>Zahl der Optimierungsparameter (skalar).

Verfahren	Parameter	Laufzeit	Bemerkungen	
Shirman und Séquin		Kurven 1.19 sec		
		Flächen 23.87 sec		
		Total: 25.06 sec		
$G^1$ -Kurvennetz Methode	Tangente: EK, KS; Funktional: LapBelt(3)	Init: 1.10 sec	# OptParam: 1102	
		Matrix: 6.56 sec		
		LGS: 4.70 sec		
	Funktional: ThinPl(1;S)	Flächen:		# OptParam: 5899
		Init: 0.47 sec		
		Matrix: 32.24 sec		
LGS: 249.26 sec				
		Total: 294.33 sec		
$G^2$ -Kurvennetz Methode	Tangente: EK, KS; Zweite: N; Funktional: LapBelt(3)	Init: 1.10 sec	# OptParam: 2755	
		Matrix: 17.22 sec		
		LGS: 10.33 sec		
	Funktional: ThinPl(1;S)	Flächen:		# OptParam: 12196
		Init: 13.16 sec		
		Matrix: 103.54 sec		
LGS: 901.09 sec				
		Total 1046.44 sec		

Tabelle B.9: Tabelle zu Abbildungen 5.14 auf Seite 97; Die zweite CAD-Fläche.

$G^1$ -Kurvennetz Methode	Tangente: PK, vKS; Funktional: LapBelt(2)	Kurven:		# OptParam: 2992
		Init: 1.51 sec		
		Matrix: 8.60 sec		
	Funktional: ThinPl(1;S)	Flächen:		# OptParam: 29335
		Init: 0.58 sec		
		Matrix: 51.04 sec		
LGS: 420.51 sec				
		Total: 488.73 sec		
$G^2$ -Kurvennetz Methode	Tangente: PK, vKS; Zweite: N; Funktional: LapBelt(3)	Init: 3.03 sec	# OptParam: 7480	
		Matrix: 37.39 sec		
		LGS: 24.87 sec		
	Funktional: DatT-hin(1;S)	Flächen:		# OptParam: 33823
		Init: 49.80 sec		
		Matrix: 939.70 sec		
LGS: 741.53 sec				
		Total 1796.32 sec		

Tabelle B.10: Tabelle zu Abbildungen 5.15 auf Seite 99; Die Gesichtsdaten.

Verfahren	Parameter	Laufzeit	Bemerkungen
$G^2$ -Kurvennetz Methode	Kurven:		
	Tangente: EK, KS; Zweite: N; Funktional: simple(4)	Init: 0.03 sec Matrix: 0.40 sec LGS: 1.01 sec	
	Flächen:		
	Funktional: DatT- hin(1;S)	Init: 1.55 sec Matrix: 26.13 sec LGS: 4.14 sec <hr/> Total 33.26 sec	
$G^2$ -Kurvennetz Methode	Kurven:		
	Tangente: EK, KS; Zweite: N; Funktional: simple(4)	Init: 0.03 sec Matrix: 0.49 sec LGS: 2.04 sec	
	Flächen:		
	Funktional: DatT- hin(1;S)	Init: 1.54 sec Matrix: 26.06 sec LGS: 5.78 sec <hr/> Total 35.94 sec	

Tabelle B.11: Tabelle zu dem Approximationsbeispiel in Abbildungen 6.3 auf Seite 107.

Verfahren	Parameter	Laufzeit	Bemerkungen
Hausecke			
$G^1$ -Kurvennetz Methode	Kurven:		
	Tangente: EK, KS; Funktional: simple(4)	Init: 0.01 sec Matrix: 0.01 sec LGS: 0.06 sec	
	Flächen:		
	Funktional: DatT- hin(1;S)	Init: 0.16 sec Matrix: 1.53 sec LGS: 0.03 sec <hr/> Total 1.80 sec	
Kofferecke			
$G^1$ -Kurvennetz Methode	Kurven:		
	Tangente: EK, KS; Funktional: LapBelt(3)	Init: 0.02 sec Matrix: 0.14 sec LGS: 0.06 sec	
	Flächen:		
	Funktional: Dritte(1;S)	Init: 0.21 sec Matrix: 5.88 sec LGS: 0.95 sec <hr/> Total 7.26 sec	
Blendfläche			
$G^2$ -Kurvennetz Methode	Kurven:		
	Tangente: EK, KS; Zweite: N; Funktional: simple(4)	Init: 0.05 sec Matrix: 0.46 sec LGS: 0.16 sec	
	Flächen:		
	Funktional: Dritte(1;S)	Init: 1.57 sec Matrix: 11.18 sec LGS: 0.80 sec <hr/> Total 14.22 sec	

Tabelle B.12: Tabelle zu den Beispielen in Abbildungen 6.3 auf Seite 107.

$G^2$ -Kurvennetz Methode	Kurven:		
	Tangente: PK, KS; Zweite: N; Federkonst.: 0.7 Funktional: LapBelt(3)	Init: 3.03 sec Matrix: 43.75 sec LGS: 30.78 sec	# OptParam: 11968
	Flächen:		
	Funktional: DatT- hin(1;S)	Init: 50.13 sec Matrix: 986.33 sec LGS: 1386.16 sec <hr/> Total 2500.18 sec	# OptParam: 38311

Tabelle B.13: Tabelle zu Abbildungen 6.4 auf Seite 109; Die approximierten Gesichtsdaten.

# Literaturverzeichnis

- [Alf89] Alfeld, P.: Scattered data interpolations in three or more variables. In: Lyche, T. und L.L. Schumaker, Herausgeber: *Math. Methods in CAGD I*, Seiten 1–33, Boston, 1989. Academic Press.
- [ADT94] Arge, E., M. Dæhlen, und A. Tveito: Approximation of scattered data using smooth grid functions. Technischer Bericht, SINTEF-SI, Oslo, 1994.
- [BBB87] Bartels, R., J. Beatty, und B. Barsky: *An introduction to splines for use in computer graphics and geometric modeling*. Kaufmann, Los Altos, Calif., 1987.
- [Bra92] Braess, D.: *Finite Elemente*. Springer, Berlin-Heidelberg, 1992.
- [CK83] Chiyokura, H. und F. Kimura: Design of solids with free-form surfaces. In: *Proc. SIGGRAPH*, Band 17:3, Seiten 289–298. ACM, 1983.
- [dB78] Boor, C. de: *A practical guide to splines*. Springer, New York, 1978.
- [dBHS87] Boor, C. de, K. Höllig, und M. Sabin: High accuracy geometric hermite interpolation. *Computer-Aided Geom. Design 4* (1987), Seiten 269–278.
- [DeR90] DeRose, T.: Necessary and sufficient conditions for tangent plane continuity of Bézier surfaces. *Computer-Aided Geom. Design 7* (1990), Seiten 165–179.
- [DM92] DeRose, T. und S. Mann: An approximately  $G^1$  cubic surface interpolant. In: Lyche, T. und L.L. Schumaker, Herausgeber: *Math. Methods in CAGD II*, Seiten 185–196. Academic Press, 1992.
- [dC92] Carmo, M. do: *Differentialgeometrie von Kurven und Flächen*. Vieweg, Wiesbaden, 2. Auflage, 1992.
- [Duc77] Duchon, J.: Splines minimizing rotation-invariant seminorms in sobolev spaces. In: Schempp, W und K. Zeller, Herausgeber: *Constructive Theory of Functions of Several Variables*, Band 571 der Reihe *Lecture Notes in Mathematics*, Seiten 85–100. Springer Verlag, Berlin, 1977.
- [EM94] Edelsbrunner, H. und E. P. Mücke: Three-dimensional alpha shapes. *ACM Trans. Graph.* 13, 1 (Januar 1994), Seiten 43–72. ISSN 0730-0301.
- [Far82] Farin, G.: A construction for visual continuity of polynomial surface patches. *Computer Graphics and Image Processing 20* (1982), Seiten 272–282.

- [Far86] Farin, G.: Triangular Bernstein-Bézier patches. *Computer-Aided Geom. Design 3* (1986), Seiten 83–127.
- [Far93] Farin, G.: *Curves and surfaces for computer aided geometric design*. Academic Press, Boston, 2 Auflage, 1993.
- [For90] Forsey, David R.: *Motion control and surface modeling of articulated figures in computer animation*. Ph.d. thesis, University of Waterloo, 1990.
- [Fra82] Franke, R.: Scattered data interpolation: Test of some methods. *Math. Comp.* 38 (1982), Seiten 181–200.
- [FN91] Franke, R. und G.M. Nielson: Scattered data interpolation and application: a tutorial and survey. In: Hagen, H. und D. Roller, Herausgeber: *Geometric Modeling*, Seiten 131–160. Springer, 1991.
- [GZ94] Gregory, J.A. und J. Zhou: Filling polygonal holes with bicubic patches. *Computer-Aided Geom. Design 11* (1994), Seiten 391–410.
- [Gre94a] Greiner, G.: Surface construction based upon variational principles. In: P.J. Laurent, A. LeMéhauté und L.L. Schumaker, Herausgeber: *Curves and Surfaces II*, Boston, 1994. AK Peters.
- [Gre94b] Greiner, G.: Data dependent fairness functionals. Technischer Bericht 94-9, University of Erlangen, 1994.
- [Gre94c] Greiner, G.: Variational design and fairing. In: *Proc. EUROGRAPHICS '94*, Band 13. Eurographics, Blackwell Publishers, 1994.
- [GKP<sup>+</sup>95] Greiner, G., A. Kolb, R. Pfeifle, H.-P. Seidel, M. Encarnaç o, und R. Klein: A platform for visualizing curves and surfaces. *Computer Aided Design* (1995).
- [Hah89] Hahn, J.: Filling polygonal holes with rectangular patches. In: Stra er, W. und H.-P. Seidel, Herausgeber: *Theory and practice of geometric modelling*, Seiten 81–92. Springer, Berlin, 1989.
- [HKD93] Halstead, M., M. Kass, und T. DeRose: Efficient, fair interpolation using Catmull-Clark surfaces. In: *Proc. SIGGRAPH*, Seiten 35–44. ACM, 1993.
- [HDD<sup>+</sup>92] Hoppe, H., T. DeRose, T. Duchamp, J. McDonald, und W. Stuetzle: Surface reconstruction from unorganized points. In: *Proc. SIGGRAPH*, Band 26:2, Seiten 71–78. ACM, 1992.
- [HDD<sup>+</sup>93] Hoppe, H., T. DeRose, T. Duchamp, J. McDonald, und W. Stuetzle: Mesh optimization. In: *Proc. SIGGRAPH*, Seiten 19–26. ACM, 1993.
- [HL92] Hoschek, J. und D. Lasser: *Grundlagen der geometrische Datenverarbeitung*. Teubner, Stuttgart, 1992.
- [Kli78] Klingenberg, W.: *A course in differential geometry*. Springer-Verlag, Berlin-Heidelberg, 1978.

- [KPS95a] Kolb, A., H. Pottmann, und H.-P. Seidel: Surface reconstruction based upon minimum norm networks. In: Dæhlen, M., T. Lyche, und L.L. Schumaker, Herausgeber: *Math. Methods for Curves and Surfaces*, Ulvik, Norway, 1995. Vanderbilt University Press.
- [KPS95b] Kolb, A., H. Pottmann, und H.-P. Seidel: Fair surface reconstruction using quadratic functionals. In: *Proc. EUROGRAPHICS '95*, Band 14, Seiten 469–479. Eurographics, Blackwell Publishers, 1995.
- [KS95] Kolb, A. und H.-P. Seidel: Interpolating scattered data with  $C^2$  surfaces. *Computer Aided Design 27:4* (1995), Seiten 277–282.
- [LH89] Liu, D. und J. Hoschek:  $GC^1$  continuity conditions between adjacent rectangular and triangular Bézier patches. *Computer Aided Design 21* (1989), Seiten 194–200.
- [Loo94a] Loop, C.: A  $G^1$  triangular spline surface of arbitrary topological type. *Computer-Aided Geom. Design 11* (1994), Seiten 303–330.
- [Loo94b] Loop, C.: Smooth spline surfaces over irregular meshes. In: *Proc. SIGGRAPH*, Seiten 303–310. ACM, 1994.
- [LD89] Loop, C. und T. DeRose: A multisided generalization of Bézier-patches. *ACM Trans. Graph. 8:3* (1989), Seiten 204–234.
- [LD90] Loop, C. und T. DeRose: Generalized B-Spline surfaces of arbitrary topology. In: *Proc. SIGGRAPH*, Band 24, Seiten 347–356. ACM, 1990.
- [LMD92] Lounsbery, M., S. Mann, und T. DeRose: Parametric surface interpolation. *IEEE Computer Graphics & Appl. 12* (1992), Seiten 45–52.
- [LM87] Lyche, T. und K. Morken: Knot removal for parametric B-Spline curves and surfaces. *Computer-Aided Geom. Design 4* (1987), Seiten 217–230.
- [Man92] Mann, S.: *Surface approximation using geometric Hermite patches*. PhD thesis, University of Washington, 1992.
- [Mor92] Moreton, H.P.: *Minimum curvature variation curves, networks and surfaces for fair free-form shape design*. PhD thesis, University of California, Berkeley, 1992.
- [MS91] Moreton, H.P. und C.H. Séquin: Surface design with minimum energy networks. In: *Proc. of the Symposium on Solid Modeling Foundations in CAD/CAM Applications*, Seiten 291–301, New York, 1991. ACM.
- [MS92] Moreton, H.P. und C.H. Séquin: Functional optimization for fair surface design. In: *Proc. SIGGRAPH*, Band 26, Seiten 167–176, 1992.
- [NDW93] Neider, J., T. Davis, und M. Woo: *OpenGL Programming Guide*. Addison Wesley, 1993.

- [Nie79] Nielson, G.M.: The side-vertex method for interpolation in triangles. *J. Approx. Th.* 25 (1979), Seiten 318–336.
- [Nie83] Nielson, G.M.: A method for interpolating scattered data based upon a minimum norm network. *Math. Comp.* 40 (1983), Seiten 253–271.
- [Nie87] Nielson, G.M.: A visually continuous transfinite triangular interpolant. In: Farin, G., Herausgeber: *Geometric Modeling: Applications and new trends*, Seiten 235–245. SIAM, Philadelphia, 1987.
- [Nie88] Nielson, G.M.: Interactive surface design using triangular network splines. In: *Proceedings 3rd Int. Conf. on Engineering Graphics and Descriptive Geometry*, Band 2, Seiten 70–77, Vienna, 1988.
- [Nie93] Nielson, G.M.: Scattered data modeling. *IEEE Computer Graphics & Appl.* 13:1 (1993), Seiten 60–70.
- [NF84] Nielson, G.M. und R. Franke: A method for construction of surfaces under tension. *Rocky Mt. J. Math.* 14 (1984), Seiten 203–221.
- [Pet89] Peters, J.: Local interpolation of a cubic mesh by a piecewise [bi]quartic  $C^1$  surface without splitting. Technischer Bericht 89-25, University of Wisconsin, 1989.
- [Pet91] Peters, J.: Smooth interpolation of a mesh of curves. *Constr. Approx.* 7 (1991), Seiten 221–246.
- [Pet93] Peters, J.: Smooth free-form surfaces over irregular meshes generalizing quadratic splines. Technischer Bericht, Purdue University, W-Lafayette, 1993.
- [Pet94] Peters, J.: Constructing  $C^1$  surfaces of arbitrary topology using biquadratic and bicubic splines. In: Sapidis, N.S., Herausgeber: *Designing fair curves and surfaces*, Seiten 277–293. SIAM, 1994.
- [Pip87] Piper, B.: Visually smooth interpolation of triangular Bézier patches. In: Farin, G., Herausgeber: *Geometric Modeling: Applications and new trends*, Seiten 235–245. SIAM, Philadelphia, 1987.
- [Pot90] Pottmann, H.: Smooth curves under tension. *Computer Aided Design* 22 (1990), Seiten 241–245.
- [Pot91] Pottmann, H.: Scattered data interpolation based upon generalized minimum norm networks. *Constr. Approx.* 7 (1991), Seiten 247–256.
- [PS77] Powell, M.J. und M.A. Sabin: Piecewise quadratic approximation on triangles. *ACM Trans. Math. Software* 3 (1977), Seiten 316–325.
- [Rie95] Riepl, A.: Interpolation verteilter Daten mit krümmungsminimierenden Flächen. Diplomarbeit am Lehrstuhl Informatik 9, Graphische Datenverarbeitung, Universität Erlangen, 1995.

- [Rit78] Ritchie, S.I.M.: Surface representation by finite elements. Master's thesis, University of Calgary, 1978.
- [SS76] Schmeißer, G. und H. Schirmeier: *Praktische Mathematik*. de Gruyter, Berlin, New York, 1976.
- [Sch87] Schumaker, L.L.: Triangulation methods. In: Chui, C.K., Herausgeber: *International Workshop on Topics in multivariate approximation, Santiago*, Seiten 219–232. Academic Press, Boston, 1987.
- [Sch93] Schumaker, L.L.: Triangulations in CAGD. *IEEE Computer Graphics & Appl.* 13:1 (1993), Seiten 71–78.
- [Sch68] Schweikert, D.G.: An interpolating curve using a spline under tension. *J. Math. Phys.* 45 (1968), Seiten 312–317.
- [She68] Shepard, D.: A two-dimensional interpolation function for irregular spaced data. In: *Proc. 23<sup>rd</sup> ACM National Conference*, Seiten 517–524, 1968.
- [SS87] Shirman, L.A. und C.H. Séquin: Local surface interpolation with Bézier patches. *Computer-Aided Geom. Design* 4 (1987), Seiten 279–295.
- [SS91] Shirman, L.A. und C.H. Séquin: Local surface interpolation with Bézier patches: errata and improvements. *Computer-Aided Geom. Design* 8 (1991), Seiten 217–221.
- [SKKG94a] Slusallek, Ph., R. Klein, A. Kolb, und G. Greiner: An object-oriented framework for curves and surfaces with applications. In: Laurent, P.J., A. LeMéhauté, und L.L. Schumaker, Herausgeber: *Curves and Surfaces II*, Seiten 457–466, Boston, 1994. AK Peters.
- [SKKG94b] Slusallek, Ph., R. Klein, A. Kolb, und G. Greiner: An object-oriented framework for curves and surfaces. In: *Proc. Fourth Eurographics Workshop on Object-Oriented Graphics*, Seiten 29–39, Sintra, Portugal, 1994.
- [Sto93] Stoer, J.: *Numerische Mathematik I*. Springer, Berlin, 1993.
- [SF73] Strang, G. und G. Fix: *An analysis of the finite element method*. Prentice-Hall, Englewood Cliffs, NJ, 1973.
- [Vel94] Veltkamp, R.C.: Closed object boundaries from scattered points. Lecture Notes in Mathematics, Berlin, 1994.
- [Wei94] Weiß, V.: Erzeugung glatter Flächen durch funktionale Optimierung. Studienarbeit am Lehrstuhl Informatik 9, Graphische Datenverarbeitung, Universität Erlangen, 1994.
- [Wer94] Wernecke, J.: *The Inventor Mentor*. Addison Wesley, 1994.



# Lebenslauf

von

Andreas Kolb

- |                            |  |
|----------------------------|--|
| 4. Juni 1965               | geboren als Sohn der Eltern Annemarie Kolb, geb. Klose und Wilhelm Kolb  |
| Juli 1971 – Juni 1975      | Grundschule Markelfingen   |
| Juli 1975 – Juni 1980      | Tegginger Hauptschule Radolfzell   |
| Juli 1980 – Juni 1982      | Berufsfachschule für Elektrotechnik Konstanz; Abschluß: Mittlere Reife   |
| Juli 1982 – Juni 1985      | Technisches Gymnasium Konstanz; Abschluß: Hochschulreife   |
| Juli 1985 – September 1986 | Wehrdienst   |
| WS 1986/87 – WS 1991/92    | Studium der Mathematik (Diplom) an der Universität Erlangen; Abschluß: Diplom  |
| 1.8.1992 – 31.12.95        | Angestellter am Institut für Mathematische Maschinen und Datenverarbeitung (IMMD) der Universität Erlangen, Lehrstuhl 9 (Graphische Datenverarbeitung) |