

Ein Triangulierungsverfahren zur Approximation mit Dahmen-Micchelli-Seidel-Splines

Dem Fachbereich 11/Mathematik der
Gerhard-Mercator-Universität
Gesamthochschule Duisburg
zur Erlangung des akademischen Grades eines
Dr. rer. nat.

genehmigte Dissertation
von
Markus Hußmann
aus
46487 Wesel

Referent: Prof. Dr. H. H. Gonska
Korreferent: Priv.-Doz. Dr. X. Zhou

Tag der mündlichen Prüfung: 14.04.1999

Inhaltsverzeichnis

1	Einleitung	2
2	Das Approximationsverfahren	6
2.1	DMS-Splines	6
2.2	Approximation mit DMS-Splines	19
2.2.1	Zur Triangulierung	19
2.2.2	Zur Knotenmenge	20
2.2.3	Zu den Gewichten	21
2.3	Beispiele zur Approximation	23
3	Das Triangulierungsverfahren	32
3.1	Theoretische Grundlagen	32
3.2	Bekannte Algorithmen	38
3.2.1	Konvexe Hülle	39
3.2.2	Delaunay-Triangulierung	41
3.2.3	Clustering-Algorithmus	50
3.3	Ein neuer Triangulierungsalgorithmus	52
3.3.1	Algorithmus A	53
3.3.2	Algorithmus B	63
3.3.3	Algorithmus C	70
3.4	Korrektheit	83
3.5	Beispiele zur Triangulierung	86

Kapitel 1

Einleitung

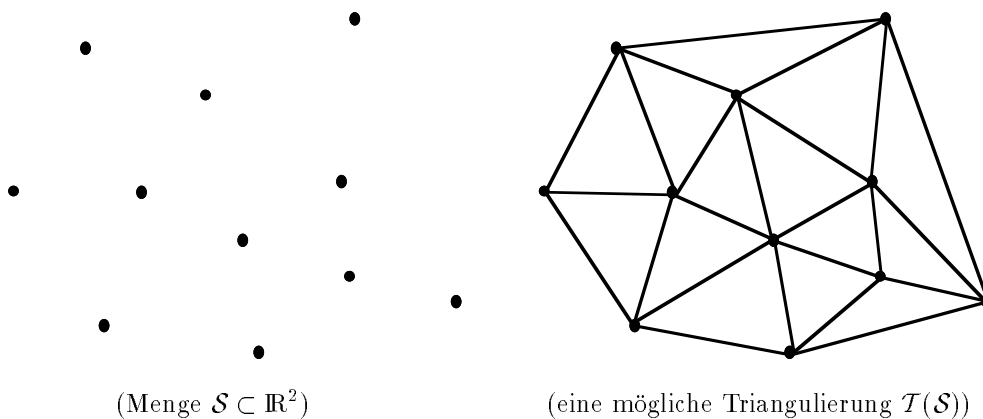
B-Splines spielen in vielen Bereichen der Mathematik eine große Rolle. Die normalisierten B-Splines z.B. bilden die Grundlage für den variationsvermindernden Schoenberg-Spline. Dieser ist aufgrund seiner vielen geometrischen Eigenschaften im Bereich der Anwendungen besonders beliebt, insbesondere durch die diskrete Definition und die lokale Kontrolle. An sinnvollen Verallgemeinerungen ins Multivariate wurde lange Zeit geforscht. W. Dahmen, C. A. Micchelli und H. P. Seidel stellten 1992 in [DMS92] mit dem DMS-Spline eine nach ihnen benannte Verallgemeinerung vor, bei der fast alle Eigenschaften aus dem Univariaten erhalten blieben. Der DMS-Spline ist eine Linearkombination von normalisierten (multivariaten) B-Splines. Zur Definition benötigen wir zunächst eine multivariate Triangulierung und zu jedem Punkt der Triangulierung eine zugeordnete Menge an Knoten. Allein diese Knotenmenge bestimmt den Grad und die Stetigkeit des Splines. Im Bivariaten liegen genau darin die Vorteile gegenüber anderen Flächendarstellungen. Durch die Definition über eine Triangulierung ist man im Gegensatz z.B. zu Tensor-Produkt-Flächen nicht auf ein regelmäßiges Gitter angewiesen. Desweiteren kann man ganz einfach die Stetigkeit in den Übergängen der polynomialen Bereiche durch die zugeordnete Knotenmenge erzielen, was beispielsweise bei Bernstein-Bézier-Flächen über Dreiecken nicht ganz so problemlos funktioniert.

Sei nun das folgende Approximationsproblem gegeben:

- gegeben: $\mathcal{D} = \{\mathbf{d}_i \in \mathbb{R}^2 : i = 1, \dots, l\}$, sowie $f(\mathbf{d}_i) \in \mathbb{R}$
- gesucht: Approximant $F : \mathbb{R}^2 \supset \mathcal{M} \rightarrow \mathbb{R}$ mit $F(\mathbf{d}_i) \approx f(\mathbf{d}_i)$.

Bislang gibt es kaum veröffentlichte Ergebnisse über die Approximation mit DMS-Splines. Wir wollen, was die Problemstellung schon zum Ausdruck bringt, hier den bivariaten Fall betrachten. Dazu benötigen wir auf jeden Fall eine bivariate Triangulierung (vgl. Definition 2.10). Eine solche Triangulierung $\mathcal{T} = \mathcal{T}(\mathcal{S})$

kann man sich als eine Menge von Dreiecken vorstellen, deren Vereinigung \mathcal{M} überdeckt. Hat man eine endliche Menge \mathcal{S} von Punkten im \mathbb{R}^2 , so erhält man eine solche Aufteilung in Dreiecke durch Verbinden der Punkte. Dabei werden so viele Verbindungen gezogen, bis nur noch Dreiecke vorhanden sind. Wichtig dabei ist, daß sich die Verbindungen nicht überschneiden.



Bisherige Triangulierungsverfahren beschäftigen sich damit, zu einer gegebenen Menge an Punkten eine Triangulierung dieser Punktmenge zu finden. Wir haben es noch etwas schwieriger, da uns die Menge \mathcal{S} noch nicht bekannt ist. Wir werden also sowohl eine Menge $\mathcal{S} \subset \mathbb{R}^2$ als auch eine Triangulierung $\mathcal{T}(\mathcal{S})$ bestimmen. Diese Vorgehensweise ergibt sich, wie wir noch sehen werden, aus der Definition eines DMS-Splines und aus der Art des vorgestellten Approximationsverfahrens.

Eine Triangulierung der Datenmenge \mathcal{D} , über der die Funktionswerte zu approximieren sind, dient oft, wie etwa bei Cline und Renka [CIR84a], als Grundlage für ein Interpolationsverfahren. Solche Triangulierungen haben im Vergleich zu denen, die wir bei unserem Verfahren benötigen, verhältnismäßig viele Dreiecke. Im Gegenzug verzichten wir auf die Forderung nach Interpolation und begnügen uns mit einer Least-Square-Approximation der Werte $f(\mathbf{d}_i)$ in den Daten \mathbf{d}_i , ($i = 1, \dots, l$).

Da der triangulierte Bereich die Definitionsmenge des approximierenden DMS-Splines darstellt, sollen alle Daten \mathcal{D} in diesem Bereich liegen. Wie wir bereits erfahren haben, ist der (bivariate) DMS-Spline eine gewichtete Summe von normalisierten (bivariaten) B-Splines. Die Bestimmung der Gewichte erfolgt bei dem vorgeschlagenen Approximationsverfahren mit Hilfe eines Least-Square-Ansatzes. Man wird sehen, daß das Least-Square-Verfahren als notwendige Bedingung zur Terminierung eine bestimmte vom Grad k des Splines abhängige Mindestanzahl $d \in \mathbb{N}$ an Daten $\mathbf{d}_i \in \mathcal{D}$ in jedem Dreieck benötigt. Desweiteren werden die Fehler der Approximation umso geringer, je kleiner die Dreiecke der Triangulierung sind.

Um die Dreiecke klein zu halten, wollen wir neben einer Minimalanzahl d an Daten pro Dreieck auch eine von d abhängige Maximalanzahl fordern.

Zu diesen speziellen Anforderungen an eine Triangulierung sind bisher keine Algorithmen bekannt, obwohl z.B. bei einem von H. J. Wenz vorgeschlagenen Quasiinterpolationsverfahren [Wen93] ähnliche Triangulierungen benötigt werden. Die einzige Arbeit, die auch Triangulierungsverfahren erwähnt, die nicht nur eine gegebene Menge an Punkten triangulieren, sondern diese Menge auch bestimmen, ist von Foley, Dayanand und Zeckzer [FDZ95]. Dort ist zum einen von einem Verfahren die Rede, das eine regelmäßige Triangulierung erzeugt, sowie von einem, das beliebige Formen von Dreiecken zulässt. Foley, Dayanand und Zeckzer benötigen allerdings Triangulierungen, die eine durchschnittliche Anzahl an Daten in den Dreiecken haben. Eine minimale bzw. eine maximale Anzahl kann dort auch nicht garantiert werden, und somit ist das Verfahren hier unbrauchbar.

Aus numerischen Gründen ergibt sich eine weitere Forderung an die Triangulierung. Wait und Mitchell benutzten in [WaM85] eine Abschätzung, nach der der Fehler eines Approximanten F zu einer Funktion f , die über einem Dreieck definiert ist, proportional ist zu $[\sin(\alpha_k)]^{-1}$, wobei α_k der kleinste Innenwinkel des Dreiecks ist. Daher sollen lange spitze Dreiecke nach Möglichkeit vermieden werden. Das Triangulierungsproblem können wir nun zusammenfassen.

Problem 1.1

Gegeben sei eine Menge von Daten $\mathcal{D} = \{\mathbf{d}_i \in \mathbb{R}^2 : i = 1, \dots, l\}$, wobei nicht alle Daten kollinear liegen.

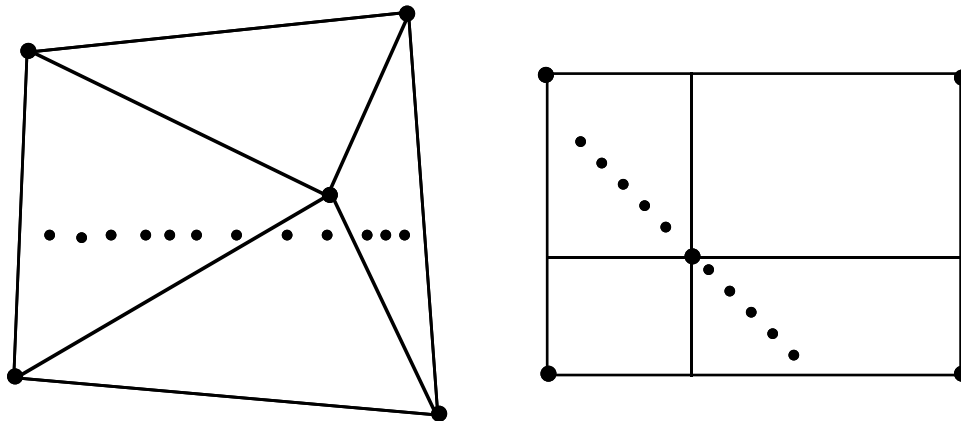
Gesucht ist dann eine Punktmenge $\mathcal{S} = \{\mathbf{t}_i \in \mathbb{R}^2 : i = 1, \dots, n\}$ und eine zugehörige Triangulierung $\mathcal{T} = \mathcal{T}(\mathcal{S})$, die die folgenden Anforderungen erfüllt:

- (A1) *Das triangulierte Gebiet beinhaltet alle Daten.*
- (A2) *Jedes Dreieck der Triangulierung enthält mindestens $d \in \mathbb{N}$ Daten.*
- (A3) *Jedes Dreieck enthält höchstens $3d - 1$ Daten.*
- (A4) *Spitze Dreiecke sollten möglichst vermieden werden.*

In dem folgenden Kapitel widmen wir uns dem Approximationsverfahren. Zunächst wird der bivariate DMS-Spline eingeführt, dann das Approximationsverfahren beschrieben und zuletzt an Beispielen illustriert.

Nach ein paar Grundlagen wird dann im dritten Kapitel ausführlich das neu entwickelte Triangulierungsverfahren beschrieben. Der Algorithmus ist umfangreich und gliedert sich in die drei Hauptalgorithmen A, B und C. Eine Unterteilung in drei Algorithmen ist, wie wir in den Beispielen sehen werden, notwendig, um gute Ergebnisse zu erzielen. Durch Algorithmus A wird eine Triangulierung erzeugt, die eine Minimalanzahl an Daten in den Dreiecken garantiert. Algorithmus B verfeinert die Triangulierung, d.h. vergrößert die Menge \mathcal{S} und somit auch die Anzahl

an Dreiecken der Triangulierung. Die Minimalanzahl an Daten bleibt dabei weiterhin garantiert. Oft liefert Algorithmus B bereits so gute Ergebnisse, daß Algorithmus C gar nicht mehr zu verfeinern braucht. Falls es aber doch notwendig ist, da es noch Dreiecke in der Triangulierung gibt, die mehr als $3d - 1$ Daten beinhalten, verfeinert Algorithmus C die Triangulierung noch einmal. Dies geschieht durch die Berechnung eines Punktes t , der in einem Dreieck mit mindestens $3d$ Daten so plaziert wird, daß in den entstehenden drei Dreiecken je mindestens wieder d Daten liegen. Die Existenz eines solchen Punktes wird durch Satz 3.39 gesichert. Daß dies durchaus nichts Selbstverständliches ist, zeigt schon die Tatsache, daß man ein Viereck mit $4d$ Daten nicht immer in vier Dreiecke mit je d unterteilen kann. Ebenso wenig kann man ein Rechteck durch einen Punkt immer analog in vier Rechtecke unterteilen. Man vergleiche dazu die folgenden Bilder.



An Beispielen wird in Abschnitt 3.5 der gesamte neue Triangulierungsalgorithmus veranschaulicht. Dort kann man sich auch davon überzeugen, daß er in Bezug auf Anforderung (A4), der Vermeidung spitzer Dreiecke, sehr gute Ergebnisse liefert.

Es sei erwähnt, daß alle Implementierungen (Triangulierungs- und Approximationsverfahren) mit Mathematica 2.2.2 auf Power Macintosh 9500/120 vorgenommen wurden.

Schließlich möchte ich ganz besonders Herrn Dr. Wenz danken, für die Anregung zu dieser Arbeit und für die vielen ausführlichen und sehr hilfreichen Diskussionen. Desweiteren gebührt mein Dank Herrn Prof. Dr. Gonska und Herrn Dr. Zhou, die mir ebenso jederzeit und in allen Belangen mit ihrem Rat und ihrem Einsatz zur Seite standen.