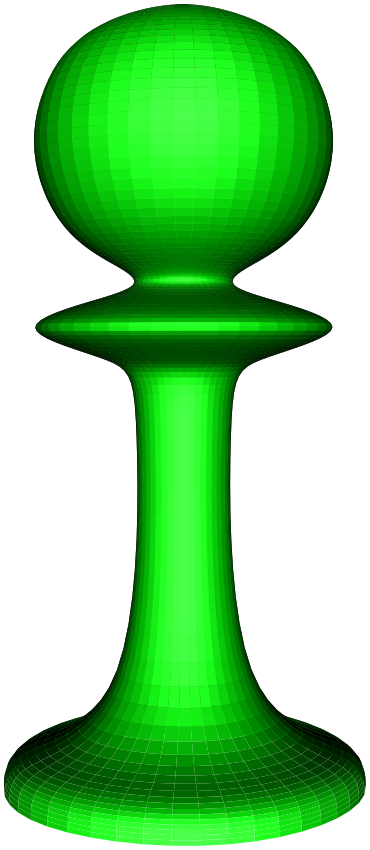
Karlsruher Institut für Technologie

Angewandte Geometrie (CAGD)
Prof. Dr. Hartmut Prautzsch
Dipl.-Ing. Yijun Xu
B.Sc. Louis Lutzweiler

MARS-Basispraktikum SS 2021

Versuch 4

ROTATIONSFLÄCHEN



Inhaltsverzeichnis

1 Aufgabenstellung

Bei der Modellierung von Objekten treten häufig Rotationskörper auf. Beispielsweise können alle Figuren des Schachspiels mit Ausnahme des Springers als Rotationskörper dargestellt werden.

Aufgabe dieses Versuchs ist es, eine Kurve, die in der xz-Ebene liegt, um die z-Achse zu drehen, so dass eine Rotationsfläche erzeugt wird. Die zu drehende Kurve ist in B-Spline-Darstellung gegeben. Die Rotationsfläche soll durch eine Tensorprodukt-Spline-Fläche approximiert werden.

2 Theoretische Grundlagen

2.1 Tensorprodukt-B-Spline-Flächen

Es seien $M_0^m(u), \ldots, M_k^m(u)$ die B-Splines vom Grad m über der Knotenfolge $\mathbf{u} = (u_0, \ldots, u_{m+k+1})$ und $N_0^n(v), \ldots, N_l^n(v)$ die B-Splines vom Grad n über der Knotenfolge $\mathbf{v} = (v_0, \ldots, v_{n+l+1})$. Dann ist eine **Tensorprodukt-Spline-Fläche** in B-Spline-Darstellung, kurz TB-Fläche genannt, gegeben durch

$$\mathbf{r}(u,v) = \sum_{i=0}^{k} \sum_{j=0}^{l} \mathbf{b}_{ij} M_i^m(u) N_j^n(v) \quad .$$

Sie wird nur über dem Intervall $[u_m, u_{k+1}] \times [v_n, v_{l+1}] \subset \mathbb{R}^2$ betrachtet. Die Kontrollpunkte $\mathbf{b}_{ij} \in \mathbb{R}^3$ bilden das Kontrollnetz der TB-Fläche.

2.2 Erzeugung der Rotationsfläche

Aus der Spline-Kurve

$$\mathbf{s}(u) = \sum_{i=0}^{k} \mathbf{d}_i M_i^m(u)$$

mit den Kontrollpunkten $\mathbf{d}_i = (x_i \ 0 \ z_i)^t \in \mathbb{R}^3$, die in der xz-Ebene verläuft, soll durch Drehung um die z-Achse eine Rotationsfläche $\mathbf{r}(u,v)$ erzeugt werden. Da $\mathbf{r}(u,v)$ als TB-Fläche dargestellt werden soll und jede TB-Fläche eine stückweise polynomiale Fläche ist, kann $\mathbf{r}(u,v)$ nicht exakt, sondern nur approximativ berechnet werden. Dazu gehen wir wie folgt vor.

Zuerst geben wir einen Rotationsparameter $R \in \mathbb{N}$ mit $R > 3$ vor. Dann rotieren wir jeden Kontrollpunkt $\mathbf{d}_i$ in R Schritten um die z-Achse: Für jeden Kontrollpunkt $\mathbf{d}_i = (x_i\ 0\ z_i)^t \in \mathbb{R}^3$ werden R Punkte

$$\mathbf{c}_{ij} := \begin{pmatrix} x_i \cos(2\pi j/R) \\ x_i \sin(2\pi j/R) \\ z_i \end{pmatrix}, \quad j = 0, \ldots, R-1$$

berechnet. Die Punkte $\mathbf{c}_{ij}$ liegen für festes i in einer Ebene parallel zur xy-Ebene. Für jedes feste i interpolieren wir die Punkte $\mathbf{c}_{i0}, \ldots, \mathbf{c}_{i,R-1}$ wie in Versuch 2 durch einen geschlossenen, uniformen, kubischen Spline der Form

$$\mathbf{b}_i(v) = \sum_j \mathbf{b}_{ij} N_j^3(v)$$

mit Kontrollpunkten $\mathbf{b}_{ij} \in \mathbb{R}^3$. (Für festes i liegen die $\mathbf{b}_{ij}$ in der Ebene $z = z_i$.)

Die Gleichung der approximierten Rotationsfläche lautet dann

$$\mathbf{r}(u, v) = \sum_{i,j} \mathbf{b}_{ij} M_i^m(u) N_j^3(v) \ .$$

Aufgabe: *Werden die Kontrollpunkte $\mathbf{b}_{ij}$ wie oben dargestellt berechnet, so muss die Interpolationsroutine aus Versuch 2 mehrmals aufgerufen werden. Überlegen Sie sich, wie die $\mathbf{b}_{ij}$ berechnet werden können, so dass die Interpolationsroutine nur einmal aufgerufen werden muss.*

2.3 Knoteneinfügen

Der Knoteneinfüge-Algorithmus für Spline-Kurven ist in Versuch 3 vorgestellt worden. Auch für TB-Flächen existiert ein Knoteneinfüge-Algorithmus. Jedoch muss im Fall von TB-Flächen unterschieden werden, ob ein Knoten in den $\mathbf{u}$-Knotenvektor oder in den $\mathbf{v}$-Knotenvektor eingefügt wird.

Wir betrachten den Fall, dass für eine TB-Fläche

$$\mathbf{r}(u, v) = \sum_{i=0}^{k} \sum_{j=0}^{l} \mathbf{c}_{ij} M_i^m(u) N_j^n(v)$$

ein Knoten $\hat{u}$ in den Knotenvektor $\mathbf{u}$ eingefügt werden soll, so dass aus dem Knotenvektor $\mathbf{u}$ der Knotenvektor $\overline{\mathbf{u}}$ entsteht. Es seien $\overline{M}_0^m(u), \ldots, \overline{M}_{k+1}^m(u)$ die B-Splines vom Grad m über dem Knotenvektor $\overline{\mathbf{u}}$. Dann existieren eindeutig bestimmte Kontrollpunkte $\mathbf{d}_{ij}$ mit

$$\mathbf{r}(u, v) = \sum_{i=0}^{k+1} \sum_{j=0}^{l} \mathbf{d}_{ij} \overline{M}_i^m(u) N_j^n(v) \ . \tag{1}$$

Wie können die $\mathbf{d}_{ij}$ berechnet werden? Für festes j ist

$$\mathbf{s}_j(u) = \sum_{i=0}^{k} \mathbf{c}_{ij} M_i^m(u)$$

eine B-Spline-Kurve vom Grad m über dem Knotenvektor $\mathbf{u}$. Mit dem Knoteneinfüge-Algorithmus aus Versuch 3 wird in $\mathbf{u}$ der Knoten $\hat{u}$ eingefügt, und wir erhalten Kontrollpunkte $\mathbf{d}_{0j}, \ldots, \mathbf{d}_{k+1,j}$. Wegen

$$\mathbf{s}_j(u) = \sum_{i=0}^{k+1} \mathbf{d}_{ij} \overline{M}_i^m(u)$$

gilt für diese Kontrollpunkte Gleichung (1).

Fazit: Um einen Knoten in die $\mathbf{u}$-Knotenfolge einer TB-Fläche einzufügen, muss der Knoteneinfüge-Algorithmus für B-Spline-Kurven auf jede Spalte $\mathbf{c}_{0j}, \ldots, \mathbf{c}_{kj}$ des Kontrollnetzes der TB-Fläche angewendet werden, insgesamt also $l + 1$-mal.

Einfache Aufgabe: *Wie sieht der Algorithmus für* **mehrfaches** *simultanes Knoteneinfügen in die $\mathbf{u}$-Knotenfolge aus? Wie funktioniert der Algorithmus für das Knoteneinfügen in die $\mathbf{v}$-Knotenfolge?*

2.4 Bézier-Darstellung einer TB-Fläche

Viele Visualisierungsprogramme, so auch das von uns verwendete `geomview`, können keine Flächen in B-Spline-Darstellung darstellen. Um unsere Rotationsfläche darstellen zu können, müssen wir also die Segmente der TB-Flächen in Bézier-Darstellung konvertieren können. Das Schreiben von Konvertierungsprogrammen ist ein trauriges, aber unvermeidliches Los der Informatiker und Informatikerinnen. Trotzdem müssen wir nun dieses Problem anpacken.

Eine TB-Fläche

$$\mathbf{r}(u,v) = \sum_{i=0}^{k} \sum_{j=0}^{l} \mathbf{c}_{ij} M_i^m(u) N_j^n(v)$$

ist über jedem Intervall $[u_a, u_{a+1}] \times [v_b, v_{b+1}]$ mit $u_a < u_{a+1}$ und $v_b < v_{b+1}$ ihres Parameterbereichs eine polynomiale Fläche vom Bigrad (m,n). Da die Bernstein-Polynome

$$B_i^n(t) := \binom{n}{i} t^i (1-t)^{n-i}$$

für $i = 0, \ldots, n$ eine Basis für den Vektorraum aller Polynome vom Grad $\leq n$ bilden, kann die TB-Fläche in jedem Parameterbereich $[u_a, u_{a+1}] \times [v_b, v_{b+1}]$ als polynomiale Tensorprodukt-Fläche bezüglich der Bernstein-Polynome dargestellt werden:

$$\mathbf{r}(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} \mathbf{b}_{ij} B_i^m \left(\frac{u-u_a}{u_{a+1}-u_a} \right) B_j^n \left(\frac{v-v_b}{v_{b+1}-v_b} \right) \quad .$$

Aufgabe: *Zeigen Sie: Sind $u_{i+1} = u_{i+m}$ und $u_{i+m+1} = u_{i+2m}$ zwei m-fache Knoten, so gilt für $j = 0, \ldots, m$ im Intervall $[u_{i+m}, u_{i+m+1}]$ die Gleichung*

$$M_{i+j}^m(u) = B_j^m \left(\frac{u-u_{i+m}}{u_{i+m+1}-u_{i+m}} \right) \quad .$$

Folgerung: Um die Bézier-Darstellung einer TB-Fläche für jedes Intervall $[u_a, u_{a+1}] \times [v_b, v_{b+1}]$ zu berechnen, müssen wir in den **u**-Knotenvektor und in den **v**-Knotenvektor solange Knoten einfügen, bis jeder innere Knoten die Vielfachheit m beziehungsweise n hat.

Diese Konstruktion beschreiben wir nun für die **u**-Knotenfolge genauer. Wie im Fall von B-Spline-Kurven heißt u_i innerer Knoten, falls $m \leq i \leq k+1$ gilt. Tritt ein innerer Knoten $\hat{u}$ mit der Vielfachheit p in der Knotenfolge **u** auf, so wird er $(m-p)$-mal in **u** eingefügt. Nach diesem Einfügevorgang hat also jeder innere Knoten die Vielfachheit m in der neu konstruierten Knotenfolge $\overline{\mathbf{u}}$.

Für die **v**-Knotenfolge muss analog vorgegangen werden.

2.5 Weitere einfache Tensorprodukt-Flächen

Eine Tensorprodukt-Fläche $\mathbf{r}(u,v)$ entsteht, indem eine Kurve $\mathbf{r}(u,0)$ durch den Raum bewegt und verändert wird. Im Falle von Rotationsflächen wurde die Meridiankurve $\mathbf{s}(u)$ um eine Achse rotiert. Schiebflächen sind ebenfalls als Tensorprodukt-Flächen darstellbar. Hierbei wird eine Kurve $\mathbf{s}(u)$ längs einer zweiten Kurve $\mathbf{t}(v)$ verschoben.

Aufgabe: *Seien $\mathbf{s}(u)$, $\mathbf{t}(v)$ Spline-Kurven. Wie lautet die Tensorprodukt-B-Spline-Darstellung der aus ihnen erzeugten Schiebfläche?*

Ist die Erzeugende $\mathbf{s}(u)$ der Schiebfläche eine Gerade, ergibt sich ein allgemeiner Zylinder. Durch Verbiegen eines Blechs können solche Flächen entstehen.

Ein verallgemeinerter Kegel besteht aus allen Geraden, die eine gegebene ebene Leitkurve $\mathbf{t}(v)$ und eine (nicht in der entsprechenden Ebene gelegene) Spitze **s** treffen.

Aufgabe: *Wie lautet die Bézier-Darstellung eines allgemeinen Kegels (Zylinders)? Stellen Sie diese Flächen auf dem Rechner dar.*

3 Praktischer Teil

3.1 Programmierung

Die Methode *generate_rotation_surface* der Klasse *spline* soll eine Rotationsfläche erzeugen. Dazu müssen jeder Kontrollpunkt des Eingabesplines rotiert werden und diese Punkte durch einen periodischen Spline interpoliert werden. Dann müssen diese Splines zu einer Fläche zusammengesetzt werden.

In der Methode *to_bezier_patches* soll nun eine Spline-Fläche vom Grad $m \times n$ durch mehrere Bézier-Flächen beschrieben werden. Dazu müssen die Vielfachheiten der Knoten angepasst werden. Anschließend entsprechen $(m+1) \times (n+1)$ große Gruppen von Kontrollpunkten der Spline-Fläche den Kontrollpunkten einer Bézierfläche. Beachten Sie, dass sich benachbarte Bézierflächen die Kontrollpunkte an den Rändern teilen.

3.2 Durchführung

1. Lesen Sie sich diese Versuchsbeschreibung gründlich durch.
2. Programmieren Sie die Funktion *generate_rotation_surface* in der Klasse *spline*. Diese soll aus einem Spline eine Rotationsfläche in Spline-Repräsentation generieren.
3. Programmieren Sie die Methode *to_bezier_patches* der Klasse *spline_surface*.
4. Schauen Sie sich Ihr Ergebnis an, indem Sie `geomview surfaces.off` in der Kommandozeile eingeben.
5. Führen Sie die Ergebnisse dem Tutor vor.

Literatur

[BOE] *W. Boehm, G. Farin, J. Kahmann*, A survey of curve and surface methods in CAGD, Computer Aided Geometric Design 1, July 1984.

[FAR] *G. Farin*, Curves and Surfaces for Computer Aided Design, Second Edition, Academic Press.

[HOS] *J. Hoschek, D. Lasser*, Grundlagen der geometrischen Datenverarbeitung, Teubner Verlag.

[PRA] *H. Prautzsch, W. Boehm, M. Paluszny*, Bézier- and B-Spline Techniques, 2002, Springer-Verlag, Berlin.