



MARS-Basispraktikum

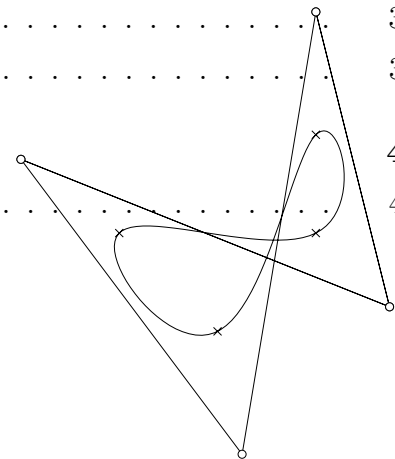
WS 2022

Versuch 2

PERIODISCHE INTERPOLATION

Inhaltsverzeichnis

1	Allgemeines	1
2	Theoretische Grundlagen	1
2.1	Periodische uniforme Spline-Kurven	2
2.2	Auflösung „fast tridiagonaler“ Gleichungssysteme	3
2.3	Approximation von Kreisen	3
3	Praktischer Teil	4
3.1	Programmierung	4



1 Allgemeines

Durchführung

1. Lesen Sie sich diese Versuchsbeschreibung gründlich durch.
2. Programmieren Sie die Methode *interpolate_cubic_periodic* in der Datei `spline.py` sowie *solve_almost_tridiagonal_equation* in der Datei `utils.py`.
3. Implementieren Sie die Funktionen *unit_circle_points* und *calculate_circle_deviation* in der Datei `periodic_interpolation.py`. Aktiviert den zweiten Teil der Aufgabe durch Entfernen der Kommentarzeichen.
4. Führen Sie die Ergebnisse dem Tutor vor.

2 Theoretische Grundlagen

Häufig sind für Modellierzwecke geschlossene Kurven erforderlich, die mit den bisher vorgestellten Algorithmen nicht konstruiert werden können. In diesem Versuch soll nun ein Verfahren vorgestellt

werden, mit dem Punkte $\mathbf{p}_i = (x_i, y_i)$ in der Ebene ($i = 0, \dots, m$) durch eine geschlossene Spline-Kurve dritten Grades interpoliert werden können. Der Anfangspunkt \mathbf{p}_0 und der Endpunkt \mathbf{p}_m der Kurve sollen hierbei zusammenfallen und die rechts- bzw. linksseitigen ersten und zweiten Ableitungen in diesen Punkten sollen übereinstimmen.

Ein Teil der Algorithmen aus Aufgabe 1 kann weiterhin eingesetzt werden. Die grundlegenden Begriffe werden nur kurz zur Vereinheitlichung der Bezeichnungen vorgestellt, ansonsten sei auf die Vorlesung und die am Ende angegebene Literatur verwiesen.

2.1 Periodische uniforme Spline-Kurven

In Aufgabe 1 wurden allgemeine Spline-Kurven eingeführt. Um die Berechnungen für periodische Splines zu vereinfachen, werden wir uns hier auf so genannte uniforme Spline-Kurven dritten Grades beschränken.

Eine *uniforme Spline-Kurve* ist eine Kurve

$$\mathbf{s}(t) = \sum_{i=0}^m \mathbf{d}_i N_i^n(t) \quad (1)$$

über dem Knotenvektor $t_k = k$, $k = 0, \dots, m+n+1$. Durch Anwendung des Algorithmus von de Boor ergeben sich in den Knotenpunkten t_k einer kubischen uniformen Spline-Kurve die folgenden Werte:

$$\mathbf{s}(t_k) = \sum_i \mathbf{d}_i N_i^3(t_k) = \frac{1}{6}(\mathbf{d}_{k-3} + 4\mathbf{d}_{k-2} + \mathbf{d}_{k-1}). \quad (2)$$

Für eine periodische uniforme Spline-Kurve über dem Intervall $[t_n, t_{m+n})$ werden die Kontrollpunkte $\mathbf{d}_0, \dots, \mathbf{d}_{m-1}$ periodisch fortgesetzt, so dass

$$\mathbf{d}_m = \mathbf{d}_0, \quad \mathbf{d}_{m+1} = \mathbf{d}_1, \dots, \mathbf{d}_{m+n-1} = \mathbf{d}_{n-1}. \quad (3)$$

Damit ist gewährleistet, dass die Kurve in den Knoten t_n und t_{m+n} bis einschließlich zur $n-1$ -ten Ableitung übereinstimmt. Für kubische uniforme Spline-Kurven ($n = 3$) gilt nämlich:

$$\begin{aligned} \frac{1}{6}(\mathbf{d}_0 + 4\mathbf{d}_1 + \mathbf{d}_2) &= \mathbf{s}(t_3) = \mathbf{s}(t_{m+3}) = \frac{1}{6}(\mathbf{d}_m + 4\mathbf{d}_{m+1} + \mathbf{d}_{m+2}) \\ \frac{1}{2}(-\mathbf{d}_0 + \mathbf{d}_2) &= \mathbf{s}'(t_3) = \mathbf{s}'(t_{m+3}) = \frac{1}{2}(-\mathbf{d}_m + \mathbf{d}_{m+2}) \\ (\mathbf{d}_0 - 2\mathbf{d}_1 + \mathbf{d}_2) &= \mathbf{s}''(t_3) = \mathbf{s}''(t_{m+3}) = (\mathbf{d}_m - 2\mathbf{d}_{m+1} + \mathbf{d}_{m+2}). \end{aligned}$$

Bei weiterer zyklischer Fortsetzung des Kontrollpolygons (und analog des Knotenvektors) ergibt sich $\mathbf{s}(t) = \mathbf{s}(t + t_m)$.

Zur Interpolation einer gegebenen Punktmenge $\mathbf{p}_0, \dots, \mathbf{p}_m = \mathbf{p}_0$ müssen diesen zunächst Parameter $t_k = k$ zugewiesen werden, so dass

$$\mathbf{s}(t_k) = \frac{1}{6}(\mathbf{d}_{k-3} + 4\mathbf{d}_{k-2} + \mathbf{d}_{k-1}) = \mathbf{p}_{k-3}, \quad k = 3, \dots, m+2. \quad (4)$$

Dies führt zu folgendem LGS zur Bestimmung der Kontrollpunkte:

$$\frac{1}{6} \begin{pmatrix} 4 & 1 & & & & 1 \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 4 & 1 \\ & & & & 1 & 4 & 1 \\ 1 & & & & & 1 & 4 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{d}_1^t \\ \vdots \\ \mathbf{d}_m^t \end{pmatrix} = \begin{pmatrix} \mathbf{p}_0^t \\ \mathbf{p}_1^t \\ \vdots \\ \mathbf{p}_{m-1}^t \end{pmatrix}. \quad (5)$$

Dieses „fast tridiagonale“ Gleichungssystem lässt sich ähnlich wie in Aufgabe 1 lösen (vergleiche nächsten Abschnitt) und man erhält unter Berücksichtigung der Periodizitätsbedingung (3) die Kontrollpunkte einer periodischen uniformen kubischen Spline-Kurve, die die gegebenen Punkte interpoliert.

2.2 Auflösung „fast tridiagonaler“ Gleichungssysteme

Gegeben sei das lineare Gleichungssystem $A\mathbf{x} = \mathbf{d}$,

$$\begin{pmatrix} b_1 & c_1 & & & a_1 \\ a_2 & b_2 & c_2 & & 0 \\ & a_3 & b_3 & c_3 & \\ & & \ddots & \ddots & \ddots \\ & 0 & & a_{n-1} & b_{n-1} & c_{n-1} \\ c_n & & & a_n & b_n \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix}. \quad (6)$$

Dabei gilt: $a_{ij} = 0$ für $|i - j| > 1$ und $(i, j) \neq (1, n)$ bzw. $(n, 1)$. Die ersten $n - 1$ Zeilen definieren ein tridiagonales Gleichungssystem wie in Aufgabe 1, dessen rechte Seite jedoch noch von x_n abhängt. Es wird wie dort gelöst und zuletzt x_n aus der letzten Gleichung ermittelt.

Es ergibt sich folgender Algorithmus zur Lösung des Gleichungssystems (6):

$$\left. \begin{aligned} v_0 &:= y_0 := 0, \quad s_0 := 1 \\ z_k &:= \frac{1}{b_k + a_k v_{k-1}} \\ v_k &:= -z_k \cdot c_k \\ y_k &:= z_k \cdot (d_k - a_k y_{k-1}) \\ s_k &:= -a_k \cdot s_{k-1} \cdot z_k \end{aligned} \right\} \quad k = 1, \dots, n-1$$

$$\left. \begin{aligned} t_n &:= 1, \quad w_n := 0 \\ t_k &:= v_k \cdot t_{k+1} + s_k \\ w_k &:= v_k \cdot w_{k+1} + y_k \end{aligned} \right\} \quad k = n-1, n-2, \dots, 1$$

$$\begin{aligned} c_n(t_1 \cdot x_n + w_1) + a_n(t_{n-1} \cdot x_n + w_{n-1}) + b_n \cdot x_n &= d_n \\ x_k &:= t_k \cdot x_n + w_k \quad k = n-1, n-2, \dots, 1. \end{aligned}$$

Aufgabe: Im Falle uniformer periodischer B-Spline-Interpolation lässt sich der Algorithmus stark vereinfachen. Wo lassen sich Rechenschritte und Variablen einsparen?

2.3 Approximation von Kreisen

Der Algorithmus zur Interpolation geschlossener Kurven kann zur Approximation von Kreisen eingesetzt werden. Kreise sind polynomial nicht exakt darstellbar und können somit auch mit B-Splines nur näherungsweise dargestellt werden. Dagegen kann mit Hilfe von NURBS (= Non Uniform Rational B-Splines) eine exakte rationale Darstellung von Kreissegmenten gegeben werden.

Für eine möglichst gute Approximation werden die Symmetrieeigenschaften des Kreises ausgenutzt und die zu interpolierenden Punkte gleichmäßig auf dem Kreis verteilt gewählt.

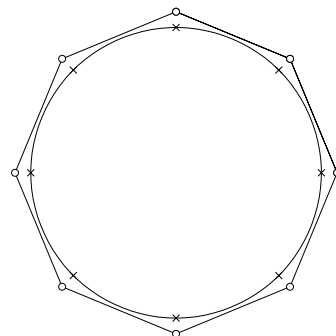
Aufgabe: Wählen Sie vier bzw. acht Punkte gleichmäßig auf dem Einheitskreis verteilt. Interpolieren Sie die Punkte jeweils mit einer geschlossenen B-Spline-Kurve. Wie könnte in diesem Fall ein diskretes Fehlermaß für die Güte der Approximation aussehen? Berechnen Sie in beiden Fällen eine Art mittleren Fehlers und eine Maximalabweichung! Was ergibt sich bei beliebiger Verteilung der zu interpolierenden Punkte?

3 Praktischer Teil

3.1 Programmierung

In der Klasse *spline* fehlt die Methode *interpolate_cubic_periodic*. Sie soll eine Liste an Punkten entgegennehmen und diese mit einem uniformen periodischen Spline von Grad drei interpolieren. Dazu muss die Funktion *solve_almost_tridiagonal_equation* in der Datei `utils.py` implementiert werden. Der interpolierende Spline sollte die Form der Ziffer 8 annehmen.

Als nächstes soll der Einheitskreis durch einen Spline approximiert werden. Implementieren Sie dazu die Funktion *unit_circle_points*, die gleichmäßige verteilte Punkte auf dem Einheitskreis erzeugt. Um die Güte der Approximation zu messen soll die Funktion *calculate_circle_deviation* implementiert werden. Berücksichtigen sie dabei Ihre Überlegungen zur obigen Aufgabe.



Literatur

- [FAR] *G. Farin*, Curves and Surfaces for Computer Aided Design, Second Edition 1990, Academic Press.
- [HOS] *J. Hoschek, D. Lasser*, Grundlagen der geometrischen Datenverarbeitung, 2. Auflage 1992, Teubner Verlag, Stuttgart.
- [PRA] *H. Prautzsch, W. Boehm, M. Paluszny*, Bézier- and B-Spline Techniques, 2002, Springer-Verlag, Berlin.