Prof. Dr. Karsten Urban
Dipl.-Math. oec. Klaus Stolle
Institut für Numerische Mathematik
Universität Ulm

WiSe 2015/2016
Blatt 2
23.10.2015

Angewandte Numerik 2

Abgabetermin: Freitag, 30.10.2015, **vor der Übung**

Für dieses Übungsblatt gibt es 26 Theorie- und 16 Matlab-Punkte, sowie 8 Theorie- und 8 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 50-Prozent-Grenzen liegen aktuell (inklusive Blatt 2) bei 24,5 Theoriepunkten und 16 Matlabpunkten.

Aufgabe 6 *(Bernstein-Polynome)* (1T+2T+2T+2T* Punkte)

Berechnen Sie mit möglichst geringem Aufwand

a) das konstante Bernstein-Polynom (also das Bernstein-Polynom vom Grad 0),

b) die linearen Bernstein-Polynome (also die Bernstein-Polynome vom Grad 1),

c) die quadratischen Bernstein-Polynome und

d) die kubischen Bernstein-Polynome

bezüglich des Intervalls $[0, 1]$. Geben Sie jeweils die von Ihnen verwendete Formel an.

Aufgabe 7 *(Darstellung der Bernstein-Polynome)* (8M* Punkte)

Schreiben Sie ein Matlabskript `bernsteinPolynome`, mit dem Sie die Bernstein-Polynome vom Grad 0 bis zum Grad n plotten können. Wählen Sie eine effiziente Implementierung. Testen Sie Ihr Matlabskript mit $n = 8$.

Aufgabe 8 *(Programmieraufgabe: Bézier-Kurven im $\mathbb{R}^2$)* (5M+5M+5M+1M+4T+2T Punkte)

a) Schreiben Sie eine Matlabfunktion `C = bezier(P,t)`, die die Bézier-Kurve zu den Kontrollpunkten `P` an den Stellen `t` auswertet. `P` ist dabei der Vektor $(\boldsymbol{P}_0, \ldots, \boldsymbol{P}_n)$ der $n+1$ Kontrollpunkte $\boldsymbol{P}_j \in \mathbb{R}^2$, also eine Matrix $\in \mathbb{R}^{2\times(n+1)}$. `t` ist ein Vektor $(t_1, \ldots, t_m)$. Ihre Funktion soll die Werte $\boldsymbol{C}(t_j) \in \mathbb{R}^2$, $j = 1, \ldots, m$ der Bézier-Kurve in der Matrix $\mathtt{C} = (\boldsymbol{C}(t_1), \ldots, \boldsymbol{C}(t_m)) \in \mathbb{R}^{2\times(m+1)}$ zurück geben.

b) Schreiben Sie ein Matlabskript `bezierKurve`, bei dessen Aufruf der Benutzer aufgefordert wird, die Kontrollpunkte $\boldsymbol{P}_0, \ldots, \boldsymbol{P}_n$ zu markieren. Bei der Eingabe soll Ihr Skript das zugehörige Kontrollpolygon zeichnen. Nach Eingabeende soll die Bézier-Kurve zu den eingegebenen Kontrollpunkten geplottet werden.
Hinweis: Sie können sich beim Einlesen der Kontrollpunkte am Matlabskript `drawSpline` von Aufgabe 5 orientieren.

c) Schreiben Sie eine Matlabfunktion `C0 = deCasteljauVisual(P,t0)`, die den Punkt `C0` der Bézier-Kurve zu den Kontrollpunkten `P` (wie in Aufgabenteil a)) an der Stelle `t0` $\in [0, 1]$ mit dem De-Casteljau-

Algorithmus berechnet. Ihre Funktion soll den De-Casteljau-Algorithmus verdeutlichen und dazu die Polygonzüge durch die Punkte $\boldsymbol{P}_{k,j}(t_0)$ mit $k = 1, \ldots, n$ und $j = 0, \ldots, n-k$ zeichnen sowie die Ecken $\boldsymbol{P}_{k,j}(t_0)$ durch kleine Kreise markieren.

d) Ergänzen Sie Ihr Matlabskript `bezierKurve`, so dass Sie Ihre Matlabfunktion `deCasteljauVisual` testen können.

e) Erläutern Sie, wie man mit Hilfe des De-Casteljau-Algorithmus eine Bézier-Kurve berechnen kann. Wählen Sie für Ihre Erläuterungen ein einfaches Beispiel mit 4 Kontrollpunkten $P_j \in \mathbb{R}^2,\ (j = 0, ..., 3)$.

f) Haben Sie Ihre Matlabfunktion `bezier` aus Aufgabenteil a) effizient implementiert? Begründen Sie Ihre Antwort.

Aufgabe 9 *(Interpolation mit quadratischen B-Splines)* (6T+4T+3T+2T Punkte)

Gegeben seien die folgenden Messwerte

x_i	0	1	2	3
y_i	2	3	3	5

Interpolieren Sie diese durch einen quadratischen B-Spline:

a) Berechnen Sie hierzu zunächst alle B-Spline-Basisfunktionen $N_j^p(t)$ vom Grad $p \in 0, 1, 2$.
Hinweis: Konstruieren Sie sich eine Knotenfolge $\mathcal{T} = \{t_0, \ldots, t_7\}$, in der die Werte $x_0 = 0$ und $x_3 = 3$ jeweils dreifach vorkommen.

b) Berechnen Sie aus den Interpolationsbedingungen $s(x_i) = y_i$ ein (unterbestimmtes) lineares Gleichungssystem zur Bestimmung der Koeffizienten c_j des quadratischen B-Splines $s(x) = \sum_{j=0}^{4} c_j N_j^2(x)$.

c) Bestimmen Sie den frei wählbaren Parameter aus einer zusätzlichen Bedingung, beispielsweise aus der Randbedingung $s'(0) = 0$.

d) Geben Sie den interpolierenden quadratischen B-Spline s explizit an.

Aufgabe 10 *(Wahr oder falsch?)* (6T* Punkte)

Gegeben sie an, ob die folgenden Aussagen wahr oder falsch sind. Begründen Sie Ihre Antwort.

a) Wenn eine Kurve in der xy-Ebene in der Parameterdarstellung gegeben ist, dann kann man besonders leicht feststellen, ob ein gegebener Punkt $(\hat{x}, \hat{y})^T$ auf dieser Kurve liegt.

b) $f(x, y) = x^2 + \frac{y^2}{2} - 1 = 0$ und $\boldsymbol{C}(t) = (cos(2\pi t), \sqrt{2}\, sin(2\pi t)),\ 0 \le t \le 1$ stellen die selbe Kurve dar.

c) Die Bernstein-Polynome haben einen lokalen Träger.

d) Die B-Spline-Basisfunktionen haben einen lokalen Träger.

e) Die Auswertung der B-Spline-Basisfunktionen mit der Rekursionsformel ist stabil.

f) Jeder kubische Spline kann als Linearkombination der B-Spline-Basisfunktionen vom Grad 3 dargestellt werden.

Hinweise:
Die Programmieraufgaben sind in Matlab zu erstellen. Senden Sie alle Files in einer E-mail mit dem Betreff **Loesung-Blatt02** an **angewandte.numerik@uni-ulm.de** (Abgabetermin jeweils wie beim Theorieteil). Drucken Sie zusätzlich allen Programmcode sowie die Ergebnisse aus und geben Sie diese vor der Übung ab. Der Source Code sollte strukturiert und, wenn nötig, dokumentiert sein.