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

WiSe 2014/2015
Blatt 1
17.10.2014

Angewandte Numerik 2

Abgabetermin: Freitag, 24.10.2014, **vor der Übung**

Für dieses Übungsblatt gibt es 23 Theorie- und 16 Matlab-Punkte, sowie 10 Theorie- und 0 Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 50-Prozent-Grenzen liegen aktuell bei 11,5 Theoriepunkten und 8 Matlabpunkten.

Aufgabe 1 *(Konstruktion von Splines)* (7T Punkte)

Gegeben sei die Funktion $s : [-1, 2] \to \mathbb{R}$ mit

$$s(x) := \begin{cases} (\alpha x - 1)(x^3 + \beta x), & x \in [-1, 0) \\ \delta x^3 + 2x - \gamma, & x \in [0, 1) \\ \epsilon x^2(x-1) + x + \frac{1}{3}, & x \in [1, 2]. \end{cases}$$

Bestimmen Sie die Parameter α, β, γ, δ und ϵ so, dass s ein kubischer Spline bezüglich der vorgegebenen Knoten $x_i \in \mathcal{T} = \{-1, 0, 1, 2\}$ ist.

Aufgabe 2 *(Eigenschaften natürlicher kubischer Splines)* (7T Punkte)

Auf dem Intervall $[-1, 1]$ seien die Knoten $x_0 = -1$, $x_1 = 0$ und $x_2 = 1$ gegeben. Welche Eigenschaften eines natürlichen kubischen Splines bezüglich der zugehörigen Zerlegung besitzt die folgende Funktion f, welche besitzt sie nicht?

$$f(x) := \begin{cases} (x+1) + (x+1)^3, & -1 \leq x \leq 0, \\ 4 + (x-1) + (x-1)^3, & 0 < x \leq 1. \end{cases}$$

Aufgabe 3 *(Aufstellen des linearen Gleichungssystem zur Bestimmung von Splines)* (3T+3T+3T Punkte)

Es sei $f : [0, 4] \to \mathbb{R}$ zweimal stetig differenzierbar und die Stützstellen $x_i := i$, $i = 0, \ldots, 4$ gegeben. Geben Sie jeweils explizit das lineare Gleichungssysteme zur Bestimmung der Koeffizienten des kubischen Splines s an, der f interpoliert. Und zwar bei

a) vollständigen Randbedingungen: $s'(0) = f'(0)$, $s'(4) = f'(4)$,

b) natürlichen Randbedingungen: $s''(0) = s''(4) = 0$,

c) periodischen Randbedingungen: $s'(0) = s'(4)$, $s''(0) = s''(4)$ (hier gilt zusätzlich $f(0) = f(4)$).

Hinweis: Stellen Sie jeweils nur das lineare Gleichungssystem auf. Sie brauchen dieses nicht explizit zu lösen. Hinweise zur Aufstellung dieser linearen Gleichungssysteme finden Sie im Skript unter „Berechnung kubischer Splines“.

Aufgabe 4 *(Berechnung eines natürlichen kubischen Splines)* (10T* Punkte)

Gegeben sind die fünf Wertepaare $(x_i, f_i)_{i=0,\dots,4}$ als

x_i	0.0	0.25	0.5	0.75	1.0
f_i	1.0	2.0	1.0	0.0	1.0

Bestimmen Sie den interpolierenden kubischen Spline mit natürlichen Randbedingungen und berechnen Sie dessen Wert an der Stelle $x = 0.35$. Wiederum im Skript unter „Berechnung kubischer Splines“ finden Sie die Vorgehensweise, um die Koeffizienten a_j, b_j, c_j und d_j des Splines in der Darstellung

$$s_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$$

zu bestimmen.

Hinweis: Rechnen Sie weitestgehend mit Brüchen.

Aufgabe 5 *(Programmieraufgabe: Kubische Splines)* (8M+5M+3M Punkte)

a) Schreiben Sie eine Matlabfunktion `C = cubicSplineCoeff(t,y,randBed,param)`, wobei

- `t` der Vektor $(t_0, ..., t_{n+1})^T$ der Knoten t_k, $k = 0, ..., n+1$ mit $t_k < t_{k+1}$ ist,
- `y` ein Vektor $(y_0, ..., y_{n+1})^T$ von Werten y_k an den Knoten t_k, $k = 0, ..., n+1$ ist,
- `randBed` die Randbedingungen angibt und die Werte `'nat'` für natürliche Randbedingungen, `'vollst'` für vollständige Randbedingungen und `'per'` für periodische Randbedingungen annehmen kann,
- `param` ein Vektor mit den gegebenenfalls benötigten Werten der Randbedingungen ist.

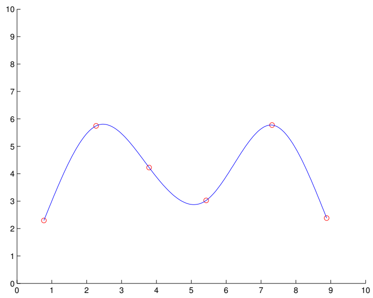
Ihre Matlabfunktion `C = cubicSplineCoeff(t,y,randBed,param)` soll die Koeffizienten des kubischen Splines s mit $s_k(t) = a_k t^3 + b_k t^2 + c_k t + d_k$, $k = 0, ..., n$, der die Paare (t_k, y_k), $k = 0, ..., n+1$ interpoliert, berechnen und sie in der Koeffizientenmatrix

$$C = \begin{pmatrix} d_0 & d_1 & \dots & d_n \\ c_0 & c_1 & \dots & c_n \\ b_0 & b_1 & \dots & b_n \\ a_0 & a_1 & \dots & a_n \end{pmatrix} \in \mathbb{R}^{4\times(n+1)}$$

zurück geben. Sie dürfen das auftretende lineare Gleichungssystem mit dem Matlaboperator \ lösen.

b) Schreiben Sie eine Matlabfunktion `sfx = evalSpline(T,C,x)`, die den durch den Knotenvektor `t` und die Koeffizientenmatrix `C` definierten kubischen Spline an den Punkten $\mathtt{x} = (x_1, x_2, ..., x_m)^T$ mit $t_0 \le x_i \le t_{n+1}$ auswertet.

c) Laden Sie das Matlabskript `drawSpline.m` von der Homepage herunter. Modifizieren Sie es so, dass es einen Spline durch alle vom Benutzer vorgegebenen Punkte zeichnet. Die Zeichnung könnte beispielsweise folgendermaßen aus sehen:



Hinweise:
Die Programmieraufgaben sind in Matlab zu erstellen. Senden Sie alle Files in einer E-mail mit dem Betreff **Loesung-Blatt01** 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 nöbig, dokumentiert sein.