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

WiSe 2017/2018
Blatt 01
20.10.2017

# Angewandte Numerik 2

**Besprechung** in den Tutorien in der Woche vom 23.10.2017 bis 27.10.2017

Für dieses Übungsblatt gibt es 16 Theorie- und 16 Matlab-Punkte, sowie 7 Theorie- und keine Matlab-Zusatzpunkte. Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.
Die 60-Prozent-Grenzen liegen aktuell (inklusive Blatt 01) bei 9,6 Theoriepunkten und 9,6 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 reellen Parameter $\alpha$, $\beta$, $\gamma$, $\delta$ und $\epsilon$ so, dass $s$ ein kubischer Spline bezüglich der vorgegebenen Knoten $t_i \in \mathcal{T} = \{-1, 0, 1, 2\}$ ist.

**Aufgabe 2** *(*Aufstellen des linearen Gleichungssystem zur Berechnung kubischer Splines) (3T+3T+3T Punkte)

Es sei $f : [0, 4] \to \mathbb{R}$ zweimal stetig differenzierbar und die Stützstellen $t_i := i, \ i = 0, \ldots, 4$ 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 3** *(*Berechnung eines kubischen Splines mit vollständigen Randbedingungen) (7T* Punkte)

Bestimmen Sie unter Verwendung vollständiger Randbedingungen den kubischen Spline $s$, welcher die Funktion $f(x) = \frac{1}{2\pi} \sin(\pi x)$ in den Punkten $t_0 = -1$, $t_1 = 0$, $t_2 = 1$ und $t_3 = 2$ interpoliert.
**Hinweis:** Zur Lösung des auftretenden linearen Gleichungssystems können Sie Matlab verwenden.

**Aufgabe 4** *(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 $t_i$, $i = 0, ..., n+1$ mit $t_i < t_{i+1}$ ist,
- `y` ein Vektor $(y_0, ..., y_{n+1})^T$ von Werten $y_i$ an den Knoten $t_i$, $i = 0, ..., n+1$ 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.

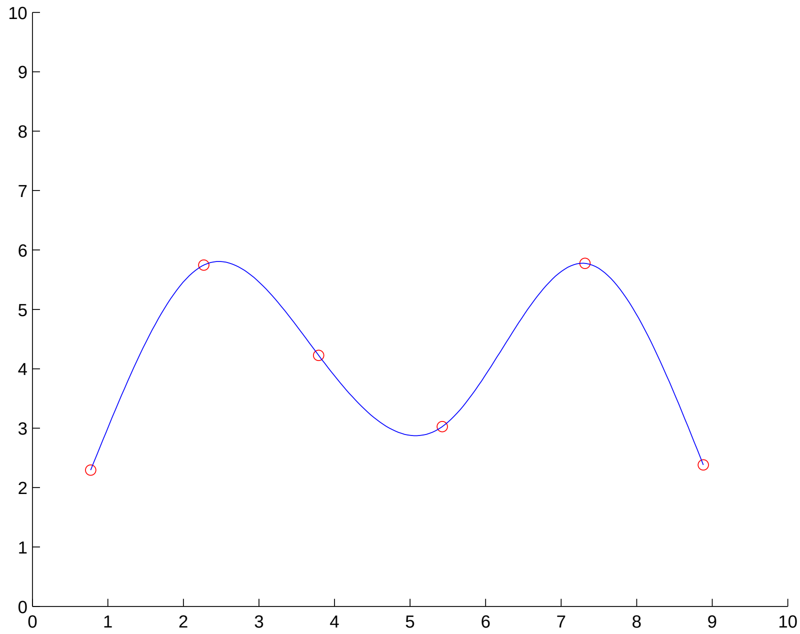
Ihre Matlabfunktion `C = cubicSplineCoeff(t,y,randBed,param)` soll die Koeffizienten des kubischen Splines $s$ mit $s_i(x) = a_i(x - t_i)^3 + b_i(x - t_i)^2 + c_i(x - t_i) + d_i$, $i = 0, ..., n$, der die Paare $(t_i, y_i)$, $i = 0, ..., n+1$ interpoliert, berechnen und sie in der Koeffizientenmatrix

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

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 $t_0 \le x_j \le t_{n+1}$ 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. Der Source Code muss strukturiert und dokumentiert sein. Senden Sie **spätestens 24 Stunden vor Ihrem Tutorium** alle Matlab-Files und alle Ergebnisse in einer E-mail mit dem Betreff **Loesung-Blatt01** an **angewandte.numerik@uni-ulm.de**.