Universität Marburg **Sommer-Semester 2009**
Fachbereich Mathematik
Prof. Dr. B. Schmitt, M. Werner

Übungen zur Numerik I
3. Aufgabenblatt

Aufgabe 9 Bei linearer Interpolation der Standardfunktion $f(x) = e^x$ in den Stellen $x_0 = 0, x_1 = 1$ beträgt der Fehler im Intervall $[0,1]$ ungefähr 0.21186. Prüfen Sie mögliche Verbesserungen durch andere lineare Approximationen $p \in \Pi_1$: (4)

a) Berechnen Sie die Interpolierende p_1 für f mit Tschebyscheff-Knoten $x_0 = \frac{1}{4}\left(2-\sqrt{2}\right), x_1 = \frac{1}{4}\left(2+\sqrt{2}\right)$ und den exakten Fehler $\max\{|p_1(x) - f(x)| : x \in [0,1]\}$.

b) Bestimmen Sie die bestapproximierende lineare Funktion $p_1(x) = \hat{a} + \hat{b}x$ und den dazugehörigen Fehler ϵ mit

$$\varepsilon := \max_{x\in[0,1]} |\hat{a} + \hat{b}x - f(x)| \leq \max_{x\in[0,1]} |a + bx - f(x)| \quad \forall a, b \in \mathbb{R}.$$

Hinweis zu b): Die Funktion p_1 ist hier charakterisiert durch die Bedingungen $\varepsilon = f(0) - p_1(0) = p_1(\xi) - f(\xi) = f(1) - p_1(1)$ mit $\xi \in (0,1)$.

Aufgabe 10 Quadratische Splines besitzen nicht so gute Eigenschaften wie die kubischen. Gesucht ist der Spline $s \in S_2^1$ mit den Interpolationsbedingungen $s(x_i) = y_i$, $i = 0, \ldots, n$ in der Beziér-Bernstein-Darstellung mit Koeffizienten $b_0, \ldots, b_{2n}$ bei äquidistantem Gitter $(h_i \equiv h)$. (3)

a) Stellen Sie das (noch unterbestimmte) Gleichungssystem für die Koeffizienten b_{2j-1}, $j = 1, \ldots, n$, auf.

b) Ergänzen Sie dieses System durch die Zusatzbedingung $s'(x_0) = y_0'$ und berechnen Sie explizit den Wert b_{2n-1}. Konstruieren Sie damit einen Datenvektor $y = (y_0, \ldots, y_n)$ mit dem gilt $|b_{2n-1}| \geq n\|y\|_\infty$ bei $y_0' = 0$.

Aufgabe 11 Wenn für die Interpolation keine Ableitungswerte am Rand bekannt sind, kann man beim kubischen Interpolations-Spline die Zahl der Parameter auch durch die folgende KK-Bedingung ("kein Knoten") reduzieren. Dabei fordert man, dass der Spline s in den beiden Randintervallen $[x_0, x_2]$ bzw. $(x_{n-2}, x_n]$ ein Polynom ist, d.h., dass in x_1 und x_{n-1} sogar s''' stetig ist. Das Gitter sei äqudistant. (4)

a) Zeigen Sie, dass die Bedingungen $s'''(x_1-) = s'''(x_1+)$, $s'''(x_{n-1}-) = s'''(x_{n-1}+)$ und

$$a_1 = \frac{4}{3}y_1 - \frac{1}{6}(y_0 + y_2), \quad a_{n-1} = \frac{4}{3}y_{n-1} - \frac{1}{6}(y_{n-2} + y_n),$$

äquivalent sind (es reicht, die Stelle x_1 zu betrachten).

b) Eliminieren Sie damit die Variablen a_1, a_{n-1} aus dem Gleichungssystem (2.2.11) der Vorlesung. Wie groß ist das verbleibende Tridiagonalsystem?

c) Übertragen Sie den Beweis von Satz 2.2.3 und zeigen Sie, dass jetzt gilt

$$\|a\|_\infty = \max\{|a_0|, |a_1|, \ldots, |a_n|\} \leq 5\|y\|_\infty.$$

Aufgabe 12 Schreiben Sie Java-Unterprogramme zur Berechnung und Auswertung des natürlichen kubischen C^2-Interpolationssplines (Randbedingung $s'' = 0$) auf einem äquidistanten Gitter mit Schrittweite h. (5*)

Zunächst sind zu einem Vektor von Funktionswerten $y_0, \ldots, y_n$ die Koeffizienten a_i nach Satz 2.2.3 zu berechnen. Zur Lösung des Gleichungssystems mit Tridiagonalmatrix steht ein Unterprogramm `tgauss` zur Verfügung (s.u.). Für die Auswertung des Splines $s(x)$ ist ein Unterprogramm `double splwert(double x,int n,double[] a)` zu erstellen. Darin ermittelt man zunächst das Intervall mit $x \in (x_m, x_{m+1}]$ und berechnet dann die Bézier-Koeffizienten $b_{3m}, \ldots, b_{3m+3}$ aus a. Die Auswertung $s(x)$ kann mit $y := (x - x_m)/h,\ z := 1 - y$ erfolgen durch $s(x) = (b_0 z + 3b_1 y)z^2 + (3b_2 z + b_3 y)y^2$ (im Fall $m = 0$). Testen Sie die Programme mit $h = 1, n = 8$ und den Daten der Wertetabelle

$x_i =$	0	1	2	3	4	5	6	7	8
$u_i =$	40	240	360	360	200	40	40	160	360
$v_i =$	150	320	320	150	40	150	320	320	150

Damit sind zwei Splinefunktionen $u(x)$ mit $u(x_i) = u_i$ und $v(x)$ mit $v(x_i) = v_i$ zu berechnen. Zeichnen Sie die zweidimensionale Kurve $(u(x), v(x)), x \in [0, 8]$, in einem Graphikfenster. Die Kurve verläuft im Quadrat $[0, 400] \times [0, 400]$.

Auf der Vorlesungsseite befindet sich ein Java-Gerüst `Aufg12.jav0` für die Graphikausgabe. Es enthält außerdem ein Unterprogramm `void tgauss(int n, double[] a)`. Dieses liefert die Komponenten $a[1 \ldots n-1]$ des Koeffizientenvektors zurück, wenn beim Aufruf im gleichen Feld die rechte Seite des Gleichungssystems (2.2.11/12) übergeben wird. Man achte auf korrekte Behandlung der Randbedingungen.

Abgabe: Freitag, 15.05.09, vor der Vorlesung.
Programmieraufgabe eine Woche später.