Universität Tübingen
Mathematisches Institut
Prof. Dr. Christian Lubich

Tübingen, den 17.11.2010

6. Übungsblatt zur Numerik

Aufgabe 20: Stellen Sie für eine äquidistante Zerlegung $x_j = x_0 + jh$ $(j = 0, 1, \ldots, n)$ das Gleichungssystem für den kubischen Spline s mit

$$s(x_j) = 0 \qquad \text{für } j = 0, \ldots, n$$
$$s'(x_0) = 1 \quad s'(x_n) = 0$$

auf. Zeigen Sie, dass die Steigungen $v_j = s'(x_j)$ mit wachsendem j rasch abfallen.

Interpretation: Störungen in den Ableitungen am Rand wirken sich im interpolierenden Spline auf Intervallen weg von x_0 kaum aus.

Aufgabe 21:

(a) Berechnen Sie mit dem Newton-Schema das Interpolationspolynom $p(x)$ zu folgenden Daten:

x_j	0	1	2	3	4
y_j	-1	-1	-7	-7	35

Berechnen Sie mit diesem Newton-Schema auch alle Ableitungen von p an der Stelle $x = -1$.

(b) Stellen Sie das Polynom $p(x) = x^3 + 2x^2 + x + 3$ mit Hilfe des Newtontableaus in der Form $p(x) = a(x-1)^3 + b(x-1)^2 + c(x-1) + d$ dar.

Aufgabe 22: Der kubische Spline minimiert $\int_a^b [s''(x)]^2 dx$. Minimiert man allgemeiner

$$\int_a^b [s''(x)]^2 dx + \lambda^2 \int_a^b [s'(x)]^2 dx$$

erhält man einen Spline, bei dem zusätzlich die Länge minimiert wird. Dies entspricht physikalisch einem Balken unter Zug. Für einen derartigen Spline ergibt sich der Ansatz

$$s_i(x) = a_i + b_i x + c_i e^{\lambda x} + d_i e^{-\lambda x}.$$

Erklären Sie den Ansatz und definieren Sie natürliche, periodische und eingespannte Splines unter Zug.

Programmieraufgabe 4: Schreiben Sie eine Funktion `newton_koeff.m` zur Berechnung der Koeffizienten des Interpolationspolynoms in Newton-Darstellung durch dividierte Differenzen für gegebene $(x_i, f_i),\ i = 0, \ldots, n$. Schreiben Sie zudem eine Funktion `newton_horner.m`, welche das Polynom an einer Stelle mit Hilfe des Hornerschemas auswertet, indem Sie die Stützstellen und die Koeffizienten als Vektor übergeben.
Testen Sie Ihre Funktionen mit den Daten

$$x_k = \frac{10k}{n} - 5, \qquad f_k = \frac{1}{x_k^2 + 1}$$
$$x_k = 5 \cdot \cos\left(\frac{2k+1}{2n+2}\pi\right), \qquad f_k = \frac{1}{x_k^2 + 1}$$

für $k = 0, \dots, n$ und $n = 4, 6, 8, 10$, und plotten Sie die Interpolationspolynome mittels Ihrer Funktionen im Intervall $[-5, 5]$, realisiert in einem Hauptprogramm `p04.m`. Lagern Sie die Auswertung von f in eine Funktion `f_p45.m` aus.

Programmieraufgabe 5: Realisieren Sie eine Funktion `tschebyscheff_koeff.m`, die Ihnen die Koeffizienten der Tschebyscheff-Darstellung liefert, und eine Funktion `tschebyscheff_clenshaw.m`, welche das Polynom durch den Clenshaw-Algorithmus auswertet.
Testen Sie Ihre Funktionen mit den Daten aus Programmieraufgabe 4 und plotten Sie die Interpolationspolynome mittels Ihrer Funktionen im Intervall $[-5, 5]$, realisiert in einem Hauptprogramm `p05.m`. Lagern Sie die Auswertung von f in eine Funktion `f_p45.m` aus.

Programmieraufgabe 6: Diese Programmieraufgabe geht nicht in die Gesamtpunktzahl der Programmieraufgaben ein, sondern bietet die Möglichkeit, extra Punkte zu sammeln!
Schreiben Sie eine Funktion `adaptint2.m` entsprechend zur Programmieraufgabe 3, welche nicht rekursiv vorgeht, sondern das Vorgehen der Vorlesung realisiert, in welcher Sie drei Vektoren *Gitter, Fehler_Gitter, Approx_Gitter* erzeugen, die das adaptive Gitter, die Fehler und die Approximationen auf den einzelnen Teilintervallen speichern. Kontrollieren Sie mit `sum(Fehler_Gitter)>tol` den Fortgang Ihres Programms. Der Rückgabewert sei `sum(Approx_Gitter)`. Die Approximation des Integrals auf einem Teilintervall durch die Simpsonregel und die Abschätzung des Fehlers sollen durch eine Funktion `SimpsonErr.m` geliefert werden. Dabei wird der Fehler durch die absolute Differenz der Simpsonregel und der zweifach angewendeten Trapezregel abgeschätzt. Vergleichen Sie die Anzahl der Teilintervalle bei gleicher Toleranz mit der Anzahl der Implementierung der Programmieraufgabe 3. Falls Sie Interesse haben, realisieren Sie die Konvergenzbeschleinigung, indem Sie S_n' berechnen.

Tipp: Der Aufruf `[M,k] = max (v)` liefert Ihnen nicht nur den betraglich maximalen Wert des Vektors v, sondern auch den entsprechenden Index.

Besprechung in den Übungen am 24.11.2010
Abgabe der Programmieraufgabe bis zum 30.11.2010
Klausurtermin: Montag der 31.01.2011 von 16-18 Uhr