Albert-Ludwigs-Universität Freiburg
Abteilung für Angewandte Mathematik
Prof. Dr. Patrick Dondl
Dr. Martin Jesenko

Sommersemester 2019

Praktikum zu Numerik 2
Blatt 3
(Abgabe: 7. Juni 2019)

Aufgabe 6. (8 Punkte)

Schreiben Sie eine Hilfsfunktion `X = tscheby(a,b,n)`, die die Tschebyscheff-Knoten $a \leq t_0 < t_1 < \ldots < t_n \leq b$ auf dem Intervall $[a, b]$ liefert. Dann verwenden Sie die Funktion `neville` aus Aufgabe 4, um das Interpolationspolynom der Funktion $f(x) = (1 + 25x^2)^{-1}$ an den Punkten $x_a = \pi/8$ und $x_b = \pi/4$ für $n = 1, 2, 4, 8, 16, 32$ auszuwerten, allerdings dieses Mal bezüglich der Tschebyscheff-Knoten $-1 \leq t_0 < t_1 < \ldots < t_n \leq 1$. Vergleichen Sie die Resultate mit denen bezüglich der äquidistanten Stützstellen aus Aufgabe 4.

Aufgabe 7. (8 Punkte)

Der `MATLAB`-Befehl `plot(X,Y,'r-*')` stellt einen durch die Vektoren X and Y definierten Polygonzug graphisch dar. Ist $X = (x_0, x_1, \ldots, x_n)^T$ und $Y = (f(x_0), f(x_1), \ldots, f(x_n))^T$, so wird eine stetige, stückweise lineare Interpolation der Funktion f dargestellt. Die Darstellung des Graphen kann mit dem Optionalen Argument `r-*` in Farbe, Liniendarstellung und Markierung verändert werden. Weitere nützliche Befehle sind: `hold on, hold off, axis, xlabel, ylabel, legend`

(1) Illustrieren Sie graphisch die stückweise lineare Approximation der Funktionen $f(x) = x^{1/2}$ auf dem Intervall $[0, 1]$ mit den Gitterpunkten

$$\text{(a)} \quad x_i = i/n, \qquad \text{(b)} \quad x_i = (i/n)^4$$

für $i = 0, 1, \ldots, n$ und $n = 2, 4, 8, 16$ indem Sie diese mit der Darstellung von f auf einem sehr feinen Gitter vergleichen.

(2) Schreiben Sie eine Routine `CubicSplineNatural( X, Y )` zur Berechnung eines interpolierenden kubischen Splines mit natürlichen Randbedingungen (ohne die Matlab-eigene Funktion `spline` zu nutzen). Speichern Sie den Spline mithilfe der Matlab-Funktion `mkpp`, als „piecewise polynomial“ vom Typ `struct`, sodass er mithilfe von `ppval` ausgewertet werden kann. Für die Funktionsweise die Dokumentation der beiden Befehle anschauen. Testen Sie die Routine mit den Partitionierungen aus (1) für die Funktion $f(x) = \sin(2\pi x)$. Erzeugen Sie jeweils aussagekräftige Graphiken und speichern Sie diese mit Hilfe des Kommandos `savefig` (siehe `MATLAB`-Dokumentation) ab.