

Praktikum zu Numerik 2

Blatt 3

(Abgabe: 21. Juni 2017)

Aufgabe 6. (8 Punkte)

Der MATLAB-Befehl `plot(X, Y, 'r-*')` stellt einen durch die Vektoren X and Y definierten Polygonzug graphisch dar. Ist $X = [x_0, x_1, \dots, x_n]^T$ und $Y = [f(x_0), f(x_1), \dots, f(x_n)]^T$, 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

$$(a) \quad x_i = i/n, \quad (b) \quad x_i = (i/n)^4$$

für $i = 0, 1, \dots, n$ 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.

Aufgabe 7. (8 Punkte)

- (1) Schreiben Sie eine Funktion, die mittels Basistransformation die komplexe Fourier-Interpolationsaufgabe für einen gegebenen Vektor y_j , $j = 0, 1, \dots, n$ (mit gleichverteilten Stützpunkten $x_i = 2\pi i/n$, $i = 0, 1, \dots, n$) löst.
- (2) Schreiben Sie weiters eine Funktion `complexCoeff2realCoeff(beta)`, die die komplexen Koeffizienten für ein reelles Signal in die `sin/cos`-Koeffizienten der reellen Aufgabe umrechnet. Hierbei sollte die Funktion prüfen ob die Koeffizienten zu einem reellen Signal passen - Achtung vor Rundungsfehlern, eine völlig exakte übereinstimmung ist nicht zu erwarten! Schreiben Sie eine Routine `[Tx] = evalrealFourier(a, b, x)`, die die trigonometrische Interpolation, die durch die Koeffizienten `a` und `b` gegeben ist, an den Punkten in `x` auswertet und die Funktionswerte `T(x)` im Vektor `Tx` speichert.

Hinweis: Matlab kann automatisch die reellen Zahlen verlassen und ins Komplexe übergehen (z.B. `sqrt(-1)`). Nutzen Sie für die imaginäre Einheit "1i". Außerdem sind die Befehle: `real()`, `imag()` und `conj()` hilfreich.

Hinweis zur Abgabe: Laden Sie bitte die Vorlage auf der Seite der Vorlesung herunter und entpacken Sie diese. Benennen Sie den entpackten Ordner 'Bn-Gruppenname' um in 'Bn-(IhreNamen)'. Arbeiten Sie in diesem Ordner. Speichern Sie die Skripte, in denen die für die Aufgabe relevanten Befehle/Funktionsaufrufe stehen, als 'Aufgabe[n].m'. Verpacken Sie den Ordner 'Bn-(IhreNamen)', (z.B. mit 7Zip) und schicken Sie den verpackten Ordner an `numerik_prakt@mathematik.uni-freiburg.de`.