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

Sommersemester 2019

Praktikum zu Numerik 2
Blatt 4
(Abgabe: 28. Juni 2019)

Aufgabe 8. (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.

Aufgabe 9. (8+4* Punkte) *FFT und Tiefpass*

(1) Implementieren Sie die komplexe Fourier-Synthese und -Analyse als rekursive Funktion und verwenden Sie Ihre Routine, um die Fourier-Transformation der Vektoren $y \in \mathbb{C}^n$ definiert durch $y_j = f_{1,2,3}(2\pi j/n)$, $j = 0, 1, \ldots, n-1$ mit $f_1 = \sin(5x) + 0.5\cos(x)$,

$$f_2(x) = \begin{cases} 1, & x \in [\pi - 1/4, \pi + 1/4] \\ 0, & \text{sonst,} \end{cases}$$

sowie

$$f_3(x) = \begin{cases} 1, & x \in [0, \pi) \\ -1, & \text{sonst} \end{cases}$$

mit $n = 2^s$, $s = 2, \ldots, 5$ zu berechnen. Stellen Sie die komplexen trigonomischen Polynome grafisch dar.

(2) Schreiben Sie einen Tiefpassfilter für die komplexe Fourier-Interpolation einer Funktion f, d.h. eine Routine, die zu hohe Frequenzen aus den Koeffizienten filtert (Achtung: was sind eigentlich die hohen Frequenzen im Signal der komplexen Fourier-Analyse?). Zum Filtern ist es am einfachsten bei einer bestimmten Frequenz abzuschneiden. Testen Sie Ihre Routine an einem glatten, periodischen Signal das mit einem kleinen Zufallsrauschen überlagert wird sowie an den Funktionen aus Teil 1.

Tipp: Das `MATLAB`-Befehl `rand` dürfte von Nutzen sein. Nutzen Sie die `MATLAB`-Darstellung komplexer Zahlen. Kleine komplexe Resultate durch Rundungsfehler können Sie durch Nehmen des Realteils nach der Rücktransformation wieder verschwinden lassen.