

1. Programmierprojekt zu Algorithmen der Numerischen Mathematik

(Anwendungen der FFT)

- a) Implementieren Sie die schnelle Fourier-Transformation (ohne Verwendung von `fft` und `ifft`). Sie dürfen annehmen, dass die Länge des Eingabevektors eine Zweierpotenz ist.
Hinweis: Implementieren Sie die schnelle Fourier-Transformation rekursiv (d.h. Ihre Funktion ruft sich selbst wieder auf).
- b) Verwenden Sie Ihren Code aus Teil a) (oder die Matlab-Funktionen `fft` und `ifft`) um eine Funktion `interpolation(f,t,N)` zu implementieren, die für eine Zweierpotenz N die trigonometrische Interpolation f_N einer Funktion f in $t \in \mathbb{R}$ auswertet. Zeichnen Sie hiermit verschiedene Interpolationen der Funktionen g und h aus Aufgabe 2 (s. Blatt 1).
- c) Die zweidimensionale diskrete Fouriertransformation $\mathcal{F}(X) = \hat{X}$ von $X = (x_{m,n}) \in \mathbb{C}^{M \times N}$ ist durch

$$\hat{x}_{k,l} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{m,n} w_M^{mk} w_N^{nl}$$

und die Faltung $X * Y \in \mathbb{C}^{M \times N}$ durch

$$(X * Y)_{k,l} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{m,n} y_{k-m,l-n}$$

($k = 0, \dots, M-1, l = 0, \dots, N-1$) gegeben. Wie im eindimensionalen Fall gilt

$$X * Y = \mathcal{F}^{-1}(\mathcal{F}(X) \cdot \mathcal{F}(Y)),$$

wobei die Multiplikation auf der rechten Seite wiederum komponentenweise zu verstehen ist und

$$\mathcal{F}^{-1}(X) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x_{m,n} w_M^{-mk} w_N^{-nl}$$

gilt. Realisieren Sie die zweidimensionale FFT, indem Sie Ihren Code aus Teil a) (oder die Matlab-Funktionen `fft` bzw. `ifft`) zunächst auf die Zeilen und dann auf die Spalten von X anwenden und berechnen Sie effizient die Faltung der Matrix `double(imread('Ausgangsmatrix.jpg'))` und der Matrix, welche in der Datei `Faltungsmatrix.asc` gespeichert ist. Das Ergebnis C lässt sich mit dem Befehl `imwrite(C/255,'Ergebnismatrix.jpg')`; wieder als Bilddatei speichern. Die Bilddatei und die Faltungsmatrix finden Sie auf der Übungshomepage.

- d) Verwenden Sie Ihren Code aus Teil a) (oder die Matlab-Funktionen `fft` und `ifft`) um den schnellen LGS-Löser für zyklische Matrizen aus Aufgabe 10 (s. Blatt 2) zu implementieren.

- e) (BONUS) Verwenden Sie Ihren Code aus Teil a) (oder die Matlab-Funktionen `fft` und `ifft`) um den Algorithmus der Vorlesung zum Glätten von Daten und zur Differentiation gestörter Daten zu implementieren. Als Eingabefunktion b verwenden Sie (zum Beispiel)

```
N=256; x=(2*pi/N)*[0:N-1]'; f=sin(x)+0.2*sin(3*x)-0.2*cos(6*x);  
e=0.1*randn(N,1); b=f+e;
```

wichtiger Hinweis: Die in Matlab eingebauten Funktionen `fft` und `ifft` weichen etwas von der in der Vorlesung verwendeten Konvention nach. Informieren Sie sich hierüber vor Verwendung der eingebauten Funktionen in der ausführlichen Product-Help!

Besprechung der Programmieraufgabe in den Übungen am 08.05.2013.
Jede Teilaufgabe ist ein Kreuz wert.