**Technische Universität Berlin** **Wintersemester 2015/2016**
**Fakultät II – Mathematik und Naturwissenschaften**
**Institut für Mathematik**
Reinhold Schneider, Benjamin Kutschan

# Numerische Mathematik I

## 9. Übungsblatt: DFT, FFT

**Hausaufgaben: (Abgabe 13. Januar 14:10 - 14:15 in Raum MA 042)**

**Aufgabe 1:** **(5 Punkte)**

Sei $f = (f_0, ..., f_{n-1}) \in \mathbb{C}^n$ und

$$d_j = (\mathrm{DFT}(f))_j = \frac{1}{n} \sum_{k=0}^{n-1} f_k \cdot e^{-jk\frac{2\pi}{n}\cdot i}$$

Wir nennen $d = (d_0, ..., d_{n-1})$ die Diskrete Fourier-Transformation (DFT) von $f$. Die Inverse DFT ist durch

$$(\mathrm{IDFT}(d))_j := \sum_{k=0}^{n-1} d_k e^{jk\frac{2\pi}{n}\cdot i} = f_j$$

definiert. Bezeichne mit $R : \mathbb{C}^n \to \mathbb{C}^n : (\alpha_0, .., \alpha_{n-1}) \mapsto (\alpha_0, \alpha_{n-1}, \alpha_{n-2} ..., \alpha_1)$ den Operator, der die Reihenfolge der Stützstellen invertiert. Überprüfe, ob

$$\mathrm{DFT}^{-1} = \mathrm{IDFT} = n \cdot \mathrm{DFT} \circ R\,.$$

**Aufgabe 2:** **(5 Punkte)**

Sei durch

$$\langle u, v\rangle := \frac{1}{\pi} \int_0^{2\pi} u(x)v(x)\,\mathrm{d}\,x$$

ein Skalarprodukt auf $L_2$ definiert. Zeige, dass die Funktionen $\{f_j : [0, 2\pi] \to \mathbb{R} : x \mapsto \cos(jx)\}_{j\in\mathbb{N}}$ orthonormal sind. (Hinweis: Ihr könnt den Kosinus als Summe von komplexen Exponentialfunktionen schreiben. Achtung: Nicht durch Null teilen.)

**Aufgabe 3:** **(5 Punkte)**

Sei $n \in \mathbb{N}$ und für $j = 0, ..., n-1$ die Stützstellen $x_j := j\frac{2\pi}{n}$ definiert, die auf dem Intervall $[0, 2\pi]$ äquidistant verteilt sind. Sei $g : [0, 2\pi] \to \mathbb{C}$ eine stetige Funktion. Definiere den Vektor $f = (f_0, ..., f_{n-1}) \in \mathbb{C}^n$ durch $f_j := g(x_j)$ für $j = 0, ..., n-1$ . Zeige, dass $g$ durch die Funktion

$$\sum_{k=0}^{n-1} d_k e^{ikx}$$

interpoliert wird, falls $d = DFT(f)$.
Zeige, dass falls $n$ gerade ist, dann $g$ auch durch die Funktion

$$\sum_{k=0}^{\frac{n}{2}-1} \left( d_k e^{ikx} + d_{n-k-1} e^{-i(k+1)x} \right)$$

interpoliert wird.
Hinweis: Diese Aufgabe und ihr Sinn ist möglicherweise einfacher zu lösen, wenn man sich durch die Programmieraufgabe das nötige Verständnis erarbeitet hat.

**Aufgabe 4:** **(5 Punkte)**

Schreibe in Pseudocode oder Matlab einen rekursiven Algorithmus, der die DFT berechnet und mit $\mathcal{O}(n \log n)$ Operationen auskommt. Hierfür soll das Lemma aus der Vorlesung verwendet werden. Algorithmen, die nicht auf der Vorlesung basieren oder aus anderen Quellen stammen, erzielen hier keine Punkte. Ihr dürft davon ausgehen, dass die Länge der Eingabe eine Zweierpotenz ist. (Hinweis: Aufteilungs- und Beschleunigungssatz)

**Programmieraufgabe 1:** (Abgabe in den Rechnersprechstunden bis zum 21. Januar 2016)
Schreiben Sie ein Octave-Programm

```
d=ffTU(f)
```

das die diskrete Fouriertransformation eines Zeilenvektors berechnet, so wie sie in der Vorlesung definiert wurde. (Tip: ...folglich nicht so wie der Beispielcode auf Wikipedia :) ) Der Algorithmus soll in $\mathcal{O}(n \log n)$ Operationen terminieren, wobei $n$ die Größe der Eingabe `f` ist. Er soll desweiteren auf dem Lemma aus der Vorlesung beruhen, so wie der Algorithmus in der Theorieaufgabe 4. Ihr dürft davon ausgehen, dass die Größe des Eingabevektors eine Potenz von 2 ist. Alternativ dürft ihr auch einen iterativen (d.h. ohne Rekursion) Algorithmus schreiben, was auch in der Praxis von höherem Wert ist.
Dies ist vielleicht die trickreichste Programmieraufgabe in diesem Semester. Nutzt bitte die Gelegenheiten, euch schon vor dem Abgabetermin helfen zu lassen.
Welches der beiden Interpolationspolynome aus Aufgabe 3 approximiert die Funktion $x \mapsto \cos x$ besser?