Technische Universität Berlin **Wintersemester 2018/2019**
Fakultät II – Mathematik und Naturwissenschaften
Institut für Mathematik
Reinhold Schneider, Benjamin Kutschan

Numerische Mathematik I

5. PA

Programmieraufgabe 1: (Abgabe in den Rechnersprechstunden bis zum 25. Januar 2019, Terminbuchung bis 21. Dezember 2018 möglich)
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 (20. Dezember) beruhen. 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 bei Bedarf die Gelegenheiten, auch schon vor dem Abgabetermin Hilfestellung zu erhalten.

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 $j = 0, ..., n-1$. Die Funktion g wird durch die Funktion

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

interpoliert, falls $d = DFT(f)$.
Falls n gerade ist, wrid 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.
Welches dieser beiden trigonometrischen Polynome approximiert die Funktion $x \mapsto \cos x$ besser? Plotten Sie beide Fälle. Tipp:

```
fh = @(x) f(x,y)
```

Nutzen Sie `interpol.m` von der Webseite, um Polynominterpolation mit Hilfe ihrer DFT auszuprobieren. Sie können als Anzahl der Stützstellen dort nur Zweierpotenzen angeben. `interpol.m` kann auch dafür dienen, die Korrektheit ihres Programmes zu überprüfen. Im Idealfall sollte sich das Interpolationspolynom immer näher an die angegebene Funktion anschmiegen. Probieren Sie zum Beispiel $x \mapsto |x|$.
Wer die FFT noch schneller machen möchte, sollte sich Gedanken über die Berechnung der Einheitswurzeln machen. Das geht deutlich schneller, wenn man auf trigonometrische Funktionen verzichtet (diese benötigen deutlich mehr Rechenzeit als zum Beispiel eine Multiplikation).