

Numerisches Programmieren, Übungen

5. Übungsblatt: Diskrete Fourier-Transformation

1) Eigenschaften der diskreten Fourier-Transformation (DFT)

Wie in der Vorlesung ist die diskrete Fourier-Transformation von komplexen Eingabedaten $v = (v_0, v_1, \dots, v_{n-1})^T$ definiert als

$$c_k = (DFT(v))_k := \frac{1}{n} \sum_{j=0}^{n-1} v_j \cdot \bar{\omega}^{jk} \quad k = 0, 1, \dots, n-1 \quad (1)$$

mit $\omega = \exp(i \frac{2\pi}{n})$.

Analog ist die inverse diskrete Fourier-Transformation als Auswertung des trigonometrischen Interpolationspolynoms an den Interpolationspunkten definiert:

$$v_l = (IDFT(c))_l := \sum_{k=0}^{n-1} c_k \cdot \omega^{kl} \quad l = 0, 1, \dots, n-1. \quad (2)$$

- i) Zeigen Sie, dass gilt: $DFT(v) = \frac{1}{n} \overline{IDFT(\bar{v})}$.
- ii) Zeigen Sie, dass gilt: $DFT(v + u) = DFT(v) + DFT(u)$.
- iii) Zeigen Sie, dass folgende Beziehungen in Bezug auf ω gelten:

$$\sum_{k=0}^{n-1} \bar{\omega}^{kl} = \begin{cases} n, & \text{für } l = 0 \\ 0, & \text{für } l = 1, \dots, n-1 \end{cases}$$
$$\sum_{k=0}^{n-1} \omega^{kl} \bar{\omega}^{kj} = \begin{cases} n, & \text{für } l = j \\ 0, & \text{für } l \neq j \end{cases} .$$

- iv) Zeigen Sie, dass DFT und $IDFT$ tatsächlich Umkehroperationen sind, indem Sie mit Hilfe der Definitionen (1) und (2) nachweisen:

$$IDFT(DFT(v))_l = v_l$$

v) Berechnen Sie die diskrete Fouriertransformation DFT für folgende drei Vektoren $a, b, c \in \mathbb{C}^n$!

$$a = (a_0, a_1, \dots, a_{n-1})^T = (1, 0, 0, \dots, 0, 0)^T,$$

$$b = (b_0, b_1, \dots, b_{n-1})^T = (i, i, i, \dots, i, i)^T,$$

$$c = (c_0, c_1, \dots, c_{n-1})^T = (1, -1, 1, -1, \dots)^T \text{ für gerades } n!$$

2) Frequenzanalyse

Eine von vielen Anwendungen der DFT ist die Transformation eines diskreten Signals $s \in \mathbb{C}^n$ aus dem Wertebereich in den Spektralbereich. Mathematisch kann man das auch als Übergang von der normalen Basis B des \mathbb{C}^n zu einer alternativen Basis $\tilde{B} = \{b_0, \dots, b_{n-1}\}$ betrachten. Die neuen Basisvektoren von \tilde{B} sind

$$b_l = \begin{pmatrix} \exp(i \frac{0\pi l}{n}) \\ \exp(i \frac{2\pi l}{n}) \\ \vdots \\ \exp(i \frac{2(n-1)\pi l}{n}) \end{pmatrix} \in \mathbb{C}^n \quad l = 0, \dots, n-1. \quad (3)$$

Die Basisvektoren b_l aus (3) sind diskrete Varianten von (komplexwertigen) Schwingungen mit Frequenz l . Man kann folgende Gleichheit herleiten:

$$\sum_{l=0}^{n-1} (DFT(s))_l \cdot b_l = s.$$

Das bedeutet, dass die DFT die Koeffizienten des Vektors s bezüglich der Basis \tilde{B} liefert. Somit beschreibt die DFT, wie sich das Signal s durch Linearkombination der Schwingungen b_0, \dots, b_{n-1} zusammensetzen lässt. Dabei geben die Beträge der Komponenten der DFT den Anteil der jeweiligen Schwingung an.

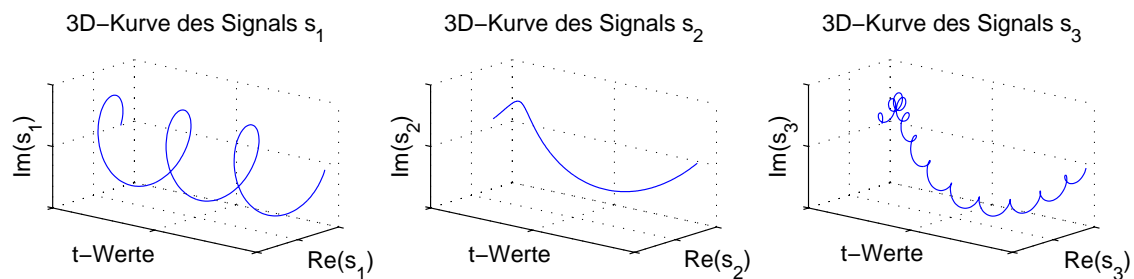


Abbildung 1: Komplexe Signale

Ordnen Sie anhand dieser Information den Signalen s_1 bis s_3 (Abb. 1) die Frequenzspektren f_1 bis f_3 (Abb. 2) zu! Dabei sind die Signale zum besseren Verständnis kontinuierlich dargestellt (z.B. $s_1 = \exp(3it)$, $t \in [0; 2\pi]$), wohingegen die Frequenzspektren den Betrag der (komplexen) Koeffizienten der DFT mit $n = 21$ gesampelten Werten des jeweiligen Signals beschreiben.

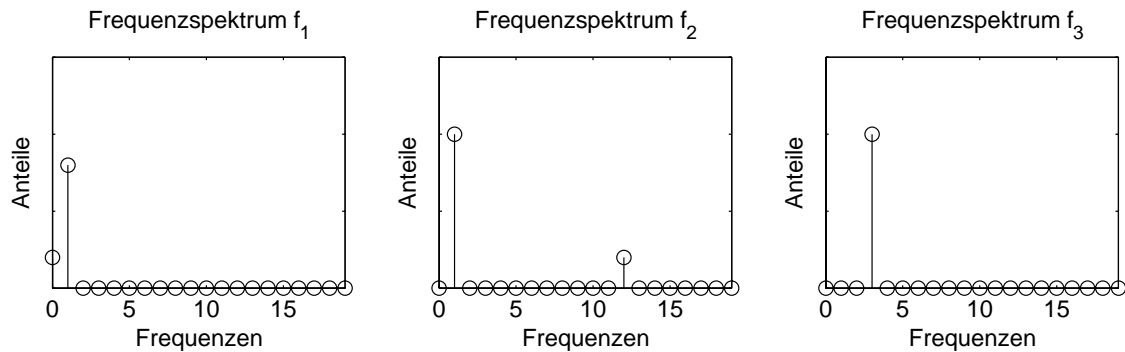


Abbildung 2: Frequenzspektren

3) Schnelle Fourier-Transformation (FFT)

In dieser Aufgabe wollen wir die schnelle Fouriertransformation (FFT) im Detail nachvollziehen und mit der direkten Formel der inversen diskreten Fourier-Transformation (*IDFT*) vergleichen. Der rekursive FFT-Algorithmus lautet:

```

FUNCTION [v[0], ..., v[n-1]] = IDFT(c[0], ..., c[n-1], n)
if n==1
    v[0] = c[0];
else
    m = n/2;
    z1 = IDFT(c[0], c[2], ..., c[n-2], n/2);
    z2 = IDFT(c[1], c[3], ..., c[n-1], n/2);
    omega = exp(2*pi*i/n);
    for j=0, ..., m-1
        v[j] = z1[j] + omega^j * z2[j];
        v[m+j] = z1[j] - omega^j * z2[j];
    end
end
return;

```

Der Algorithmus kann wie in der Vorlesung durch eine Sortierphase und eine Kombinationsphase (sukzessive Anwendung des Butterfly-Operators) visualisiert werden (vgl. Abb. 3 und Abb. 4).

Der Einfachheit halber soll nun $n = 4$ sein, d.h. wir wollen mit der *IDFT* aus den vier Koeffizienten c_0, c_1, c_2, c_3 die zugehörigen vier Funktionswerte v_0, v_1, v_2, v_3 berechnen.

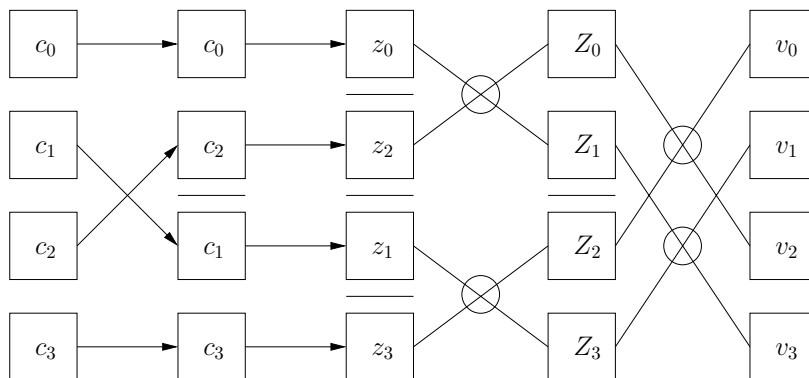


Abbildung 3: Schematischer Ablauf des FFT-Algorithmus.

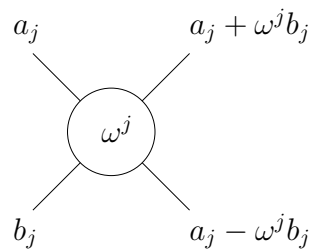


Abbildung 4: Der Butterfly-Operator.

Die direkte Formel dazu lautet ja

$$v_j = \sum_{k=0}^{n-1} c_k \omega^{jk}, \quad j = 0, \dots, n-1. \quad (4)$$

- i) Berechnen Sie die v_j zunächst nach der direkten Formel (4)!
- ii) Verwenden Sie nun den FFT-Algorithmus, um zu zeigen, dass man damit tatsächlich dasselbe Ergebnis wie in i) erhält!