Technische Universität München
Institut für Informatik
Prof. Dr. Hans-Joachim Bungartz
Hayden Liu Weng
Sebastian Wolf
Michael Obersteiner

WiSe 2021/22

Numerisches Programmieren, Übungen

Musterlösung 5. Übungsblatt: Diskrete Fourier-Transformation, FFT

0) Komplexe Zahlen

In diesem Übungsblatt wird Rechnen mit komplexen Zahlen gebraucht. Hier finden Sie einige der wichtigsten Eigenschaften komplexer Zahlen ($z \in \mathbb{C}$):

- $z = x + iy, \quad x = \mathrm{Re}(z),\ y = \mathrm{Im}(z)$
- $\overline{z} = x - iy$ (konjugiert Komplexes), $\quad \overline{z \cdot w} = \overline{z} \cdot \overline{w}$
- $|z| = \sqrt{\mathrm{Re}(z)^2 + \mathrm{Im}(z)^2} \ = \ \sqrt{x^2 + y^2} = \sqrt{z \cdot \overline{z}}$
- $e^{it} = \cos(t) + i \sin(t)$,
- $e^z = e^{x+iy} = e^x e^{iy} = e^x \left(\cos(y) + i \sin(y)\right)$.
- e^{it} durchläuft den Einheitskreis gegen den Uhrzeigersinn (beginnend bei (1,0)). Außerdem ist die Funktion 2π-periodisch. Daher gilt:

$$\begin{aligned} e^{i\cdot 0} \ = \ e^{i\cdot 2k\pi} \ &= \ 1 \in \mathbb{R},\ k \in \mathbb{Z} \\ e^{-i\cdot\pi} \ &= \ -1 \in \mathbb{R}. \end{aligned}$$

Zum Beispiel:

$$z = 3 + 4i, \qquad \overline{z} = 3 - 4i, \qquad |z| = \sqrt{3^2 + 4^2} = \sqrt{(3+4i)(3-4i)} = 5.$$

$$e^{i\pi/3} = \cos\left(\frac{\pi}{3}\right) + i \cdot \sin\left(\frac{\pi}{3}\right) = 0.5 + i \cdot \frac{\sqrt{3}}{2}$$

Aufgabe:
wie viele Lösungen hat die Gleichung

$$\omega^3 = 1$$

a) wenn $\omega \in \mathbb{R}$?

b) wenn $\omega \in \mathbb{C}$?

Lösung: $\omega^3 = 1$ ist äquivalent zu

$$(\omega - 1)(\omega^2 + \omega + 1) = 0$$

In $\mathbb{R}$ - nur eine Lösung, weil die Diskriminante vom $\omega^2 + \omega + 1 = 0$ negativ ist. In $\mathbb{C}$ - drei:

$$\omega_1 = \exp(i \cdot 0), \quad \omega_2 = \exp(i \cdot 2\pi/3), \quad \omega_3 = \exp(i \cdot 4\pi/3)$$

1) Frequenzanalyse

Eine von vielen Anwendungen der DFT ist die Transformation eines diskreten Signals $s \in \mathbb{C}^n$ aus dem Wertebereich (z.B. räumliche oder zeitliche Domäne) in den Spektralbereich (z.B. Frequenzraum).

Dargestellt wird der Spektralbereich für gewöhnlich durch das Frequenzspektrum. Es gibt an, wie hoch die jeweiligen Anteile der Grundfrequenzen im Ausgangssignal sind. Die k-te Grundfrequenz ist durch $e^{i\frac{2\pi}{n}k}$ definiert.

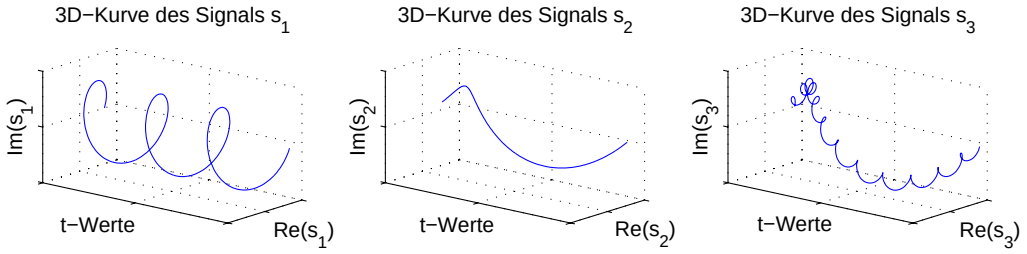


Abbildung 1: Komplexe Signale

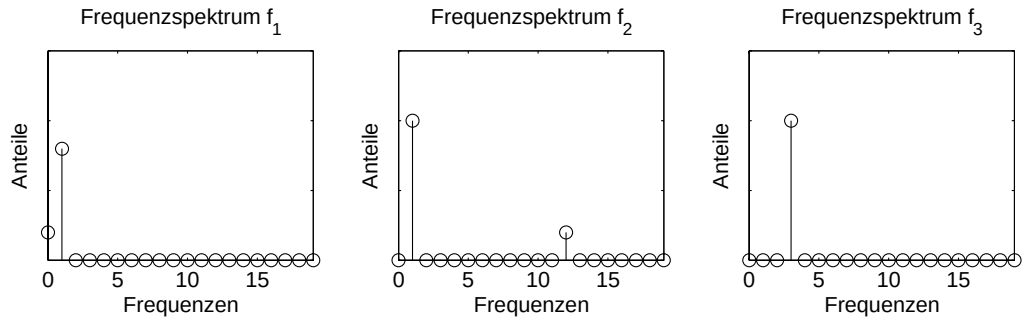


Abbildung 2: Frequenzspektren der Signale

Ordnen Sie den Signalen s_1 bis s_3 (Abbildung 1) die Frequenzspektren f_1 bis f_3 (Abbildung 2) zu! Begründen Sie darüber hinaus Ihre Entscheidung!

Dabei sind die Signale zum besseren Verständnis kontinuierlich dargestellt (z.B. $s_1 = e^{3it},\ t \in [0; 2\pi]$), wohingegen die Frequenzspektren den Betrag der (komplexen) Koeffizienten der DFT mit $n = 21$ gesampelten Werten des jeweiligen Signals beschreiben.

Lösung: Die Signale s_1, s_2, s_3 sind folgendermassen mit den Frequenzspektren verknüpft:

$$\begin{aligned} s_1 &= e^{3it} &&\leftrightarrow\ f_3 \\ s_2 &= 0.2\ i + 0.8\ e^{it} &&\leftrightarrow\ f_1 \\ s_3 &= e^{it} + 0.2\ e^{12it} &&\leftrightarrow\ f_2, \end{aligned}$$

wobei $t \in [0; 2\pi]$.
Zusätzliche Begründungen:

s_1 einzelne gleichmässige mittelfrequentierte Schwingung $\rightarrow f_3$

s_2 einzelne langsame Schwingung + leichte Verschiebung nach oben $\rightarrow f_1$

s_3 dominante langsame Schwingung + kleine hochfrequentierte Schwingung $\rightarrow f_2$

2) Diskrete Fourier-Transformation (DFT)

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

$$c_k \;=\; \Big(\mathrm{DFT}(v)\Big)_k \;:=\; \tfrac{1}{n}\sum_{j=0}^{n-1} v_j \cdot \overline{\omega}^{jk} \qquad k = 0, 1, \ldots, n-1 \tag{1}$$

mit $\omega = \exp(i\,\frac{2\pi}{n})$. Man kann die Gleichung (1) auch als Matrix-Vektor Multiplikation darstellen:

$$\begin{pmatrix} c_0 \\ c_1 \\ \ldots \\ c_{n-1} \end{pmatrix} = c = M_{\mathrm{DFT},n} \cdot v = \frac{1}{n} \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \overline{\omega}^1 & \overline{\omega}^2 & \ldots & \overline{\omega}^{(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \overline{\omega}^{n-1} & \overline{\omega}^{2(n-1)} & \ldots & \overline{\omega}^{(n-1)(n-1)} \end{pmatrix} \cdot \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{pmatrix} \tag{2}$$

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

$$v_l \;=\; \Big(\mathrm{IDFT}(c)\Big)_l \;:=\; \sum_{k=0}^{n-1} c_k \cdot \omega^{kl} \qquad l = 0, 1, \ldots, n-1. \tag{3}$$

und als Matrix-Vektor Multiplikation:

$$\begin{pmatrix} v_0 \\ v_1 \\ \ldots \\ v_{n-1} \end{pmatrix} = v = M_{\mathrm{IDFT},n} \cdot c = \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \omega^1 & \omega^2 & \ldots & \omega^{(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \ldots & \omega^{(n-1)(n-1)} \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{pmatrix} \tag{4}$$

a) Berechnen Sie $\mathrm{DFT}(\begin{pmatrix} 1 & 0 & -1 \end{pmatrix}^T)$!

b) Zeigen Sie, dass $\omega^n = 1$.

c) Zeigen Sie, dass gilt: $\mathrm{DFT}(v) = \frac{1}{n}\,\overline{\mathrm{IDFT}(\overline{v})}$.

d) Zeigen Sie, dass gilt: $\mathrm{DFT}(v + u) = \mathrm{DFT}(v) + \mathrm{DFT}(u)$.

e) Zusatzaufgabe: Jetzt möchten wir zeigen, dass DFT und IDFT tatsächlich Umkehroperationen sind:

$$\mathrm{IDFT}(\mathrm{DFT}(v))_l = v_l.$$

Das entspricht die folgende Matrix-Gleichung:

$$M_{\mathrm{IDFT}} \cdot M_{\mathrm{DFT}} = \mathbb{1}, \tag{5}$$

wobei $\mathbb{1}$ die Einheitsmatrix ist.

i) Stellen Sie die Gleichung (5) für $n = 3$ auf!

ii) Zeigen Sie, dass folgende Beziehungen in Bezug auf ω gelten (für allgemeine n):

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

Hinweis: Verwenden Sie die Formel für eine geometrische Summe (für $z \neq 1$):

$$1 + z + z^2 + \ldots + z^{n-1} = \frac{1 - z^n}{1 - z}.$$

iii) Zeigen Sie dass die aufgestellte Gleichung (5) gilt.

Lösung:

a) Wir haben

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \overline{\omega}^1 & \overline{\omega}^2 \\ 1 & \overline{\omega}^2 & \overline{\omega}^4 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 0 \\ 1 - \overline{\omega}^2 \\ 1 - \overline{\omega}^4 \end{pmatrix},$$

mit

$$\begin{aligned} \omega &= \exp\left(\tfrac{2\pi i}{3}\right) = \cos\left(\tfrac{2\pi}{3}\right) + i \sin\left(\tfrac{2\pi}{3}\right) = -\tfrac{1}{2} + i\tfrac{\sqrt{3}}{2}, \\ \overline{\omega} &= \exp\left(\tfrac{-2\pi i}{3}\right) = \cos\left(\tfrac{-2\pi}{3}\right) + i \sin\left(\tfrac{-2\pi}{3}\right) = -\tfrac{1}{2} - i\tfrac{\sqrt{3}}{2}, \\ \overline{\omega}^2 &= \exp\left(\tfrac{-4\pi i}{3}\right) = \omega, \\ \overline{\omega}^4 &= \omega^2 = \overline{\omega}. \end{aligned}$$

Das ergibt dann

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 0 \\ 1 - \overline{\omega}^2 \\ 1 - \overline{\omega}^4 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 0 \\ 1 - \omega \\ 1 - \overline{\omega} \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 0 \\ 1 - \left(-\frac{1}{2} + i\frac{\sqrt{3}}{2}\right) \\ 1 - \left(-\frac{1}{2} - i\frac{\sqrt{3}}{2}\right) \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{1}{2} - i\frac{\sqrt{3}}{6} \\ \frac{1}{2} + i\frac{\sqrt{3}}{6} \end{pmatrix}.$$

b) $\omega^n = \exp\left(\frac{2\pi i n}{n}\right) = \exp(2\pi i) = \exp(0) = 1.$

c) Wir betrachten den Ausdruck auf der rechten Seite:

$$\begin{aligned} \left(\frac{1}{n}\,\overline{\mathrm{IDFT}(\overline{v})}\right)_l &= \frac{1}{n}\overline{\sum_{k=0}^{n-1} \overline{v_k}\ \omega^{kl}} = \frac{1}{n}\sum_{k=0}^{n-1} \overline{\overline{v_k}\ \omega^{kl}} = \frac{1}{n}\sum_{k=0}^{n-1} \overline{\overline{v_k}}\ \overline{\omega}^{kl} \\ &= \frac{1}{n}\sum_{k=0}^{n-1} v_k\ \overline{\omega}^{kl} = \mathrm{DFT}(v)_l, \quad l = 0, \ldots, n-1. \end{aligned}$$

d) Es gilt für alle $k = 0, \ldots, n-1$:

$$\mathrm{DFT}(v+u)_k \;=\; \frac{1}{n}\sum_{j=0}^{n-1}(v+u)_j\;\overline{\omega}^{jk} \;=\; \frac{1}{n}\sum_{j=0}^{n-1} v_j\;\overline{\omega}^{jk} + \frac{1}{n}\sum_{k=0}^{n-1} u_j\;\overline{\omega}^{kl} \;=\; \mathrm{DFT}(v)_k + \mathrm{DFT}(u)_k.$$

e) i)

$$\frac{1}{3}\cdot\begin{pmatrix}1 & 1 & 1\\ 1 & \overline{\omega}^1 & \overline{\omega}^2\\ 1 & \overline{\omega}^2 & \overline{\omega}^4\end{pmatrix}\cdot\begin{pmatrix}1 & 1 & 1\\ 1 & \omega^1 & \omega^2\\ 1 & \omega^2 & \omega^4\end{pmatrix} \stackrel{?}{=} \begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix} \tag{6}$$

ii) Mit Hilfe der geometrischen Summenformel

$$\sum_{j=0}^{n} z^j \;=\; \frac{1-z^{n+1}}{1-z}, \quad \text{für } z \neq 1$$

berechnen wir mit $z = \overline{\omega}^l$:

$$\text{für } l = 0: \qquad \sum_{k=0}^{n-1}\overline{\omega}^{0k} \;=\; \sum_{k=0}^{n-1} e^0 \;=\; n$$

$$\text{für } l = 1, \ldots, n-1: \qquad \sum_{k=0}^{n-1}\overline{\omega}^{kl} \;=\; \frac{1-(\overline{\omega}^l)^n}{1-\overline{\omega}^l} \;=\; \frac{1-e^{-i\frac{2\pi}{n}nl}}{1-e^{-i\frac{2\pi}{n}l}} \;=\; 0.$$

Außerdem erhalten wir mit dem gleichen Trick:

$$\sum_{k=0}^{n-1}\omega^{kl}\;\overline{\omega}^{kj} \;=\; \sum_{k=0}^{n-1}\omega^{k(l-j)} \;=\; \sum_{k=0}^{n-1} e^{i\frac{2\pi}{n}k(l-j)} \;=\; \begin{cases} n, & \text{für } l = j\\ 0, & \text{für } l \neq j\end{cases}.$$

iii) Wir berechnen mit den Ergebnissen aus c):

$$\frac{1}{3}\begin{pmatrix}1 & 1 & 1\\ 1 & \overline{\omega}^1 & \overline{\omega}^2\\ 1 & \overline{\omega}^2 & \overline{\omega}^4\end{pmatrix}\cdot\begin{pmatrix}1 & 1 & 1\\ 1 & \omega^1 & \omega^2\\ 1 & \omega^2 & \omega^4\end{pmatrix} =$$

$$= \frac{1}{3}\begin{pmatrix}3 & 1+\omega+\omega^2 & 1+\omega^2+\omega^4\\ 1+\overline{\omega}+\overline{\omega}^2 & 1+\overline{\omega}\omega+\overline{\omega}^2\omega^2 & 1+\overline{\omega}\omega^2+\overline{\omega}^2\omega^4\\ 1+\overline{\omega}^2+\overline{\omega}^4 & 1+\overline{\omega}^2\omega+\overline{\omega}^4\omega^2 & 1+\overline{\omega}^2\omega^2+\overline{\omega}^4\omega^4\end{pmatrix} \stackrel{ii)}{=} \frac{1}{3}\begin{pmatrix}3 & 0 & 0\\ 0 & 3 & 0\\ 0 & 0 & 3\end{pmatrix} = \mathbb{1}$$

3) Schnelle (Inverse) Fourier-Transformation (FFT)

In dieser Aufgabe wollen wir die schnelle Variante der inversen Fouriertransformation (IFFT) im Detail nachvollziehen und mit der direkten Formel der inversen diskreten Fourier-Transformation (IDFT) vergleichen (der Fall von FFT und DFT ist ähnlich). Eine mögliche Implementierung der IFFT-Algorithmus lautet:

```
0  FUNCTION [v[0],...,v[n-1]]=IFFT(c[0],...,c[n-1],n)
   if n==1
      v[0] = c[0];
   else
      m = n/2;
5     z1 = IFFT(c[0],c[2],...,c[n-2],n/2);
      z2 = IFFT(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];
10       v[m+j] = z1[j] - omega^j * z2[j];
      end
   end
   return v;
```

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).

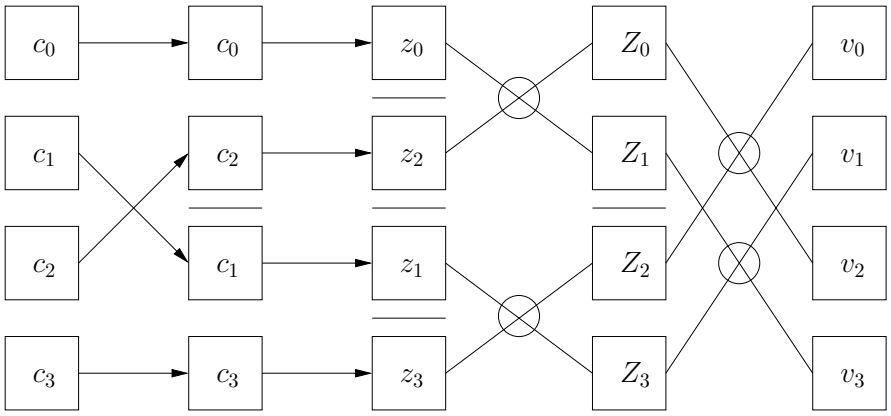


Abbildung 3: Schematischer Ablauf des FFT-Algorithmus.

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.

a) Berechnen Sie die v_j zunächst nach der direkten Formel IDFT(v)!

b) Verwenden Sie nun den IFFT-Algorithmus, um zu zeigen, dass man damit tatsächlich dasselbe Ergebnis wie in a) erhält!

Lösung:

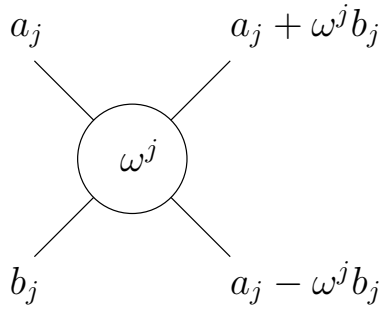


Abbildung 4: Der Butterfly-Operator.

a) Mit der direkten Formel (4) für $n = 4$ und $\omega = e^{i\frac{2\pi}{n}}$ erhalten wir:

$$\begin{aligned}
v_0 &= c_0\omega^{0\cdot 0} + c_1\omega^{0\cdot 1} + c_2\omega^{0\cdot 2} + c_3\omega^{0\cdot 3} = c_0\omega^0 + c_1\omega^0 + c_2\omega^0 + c_3\omega^0 \\
v_1 &= c_0\omega^{1\cdot 0} + c_1\omega^{1\cdot 1} + c_2\omega^{1\cdot 2} + c_3\omega^{1\cdot 3} = c_0\omega^0 + c_1\omega^1 + c_2\omega^2 + c_3\omega^3 \\
v_2 &= c_0\omega^{2\cdot 0} + c_1\omega^{2\cdot 1} + c_2\omega^{2\cdot 2} + c_3\omega^{2\cdot 3} = c_0\omega^0 + c_1\omega^2 + c_2\omega^4 + c_3\omega^6 \\
v_3 &= c_0\omega^{3\cdot 0} + c_1\omega^{3\cdot 1} + c_2\omega^{3\cdot 2} + c_3\omega^{3\cdot 3} = c_0\omega^0 + c_1\omega^3 + c_2\omega^6 + c_3\omega^9.
\end{aligned}$$

Aufgrund der Eigenschaften der Exponentialfunktion (2π-Periodizität etc., vgl. Aufg. 1)) gilt: $\omega^4 = e^{i2\pi} = 1,\ \omega^2 = e^{i\pi} = -1$. Daher erhalten wir insgesamt:

$$\begin{aligned}
v_0 &= c_0 + c_1 + c_2 + c_3 \\
v_1 &= c_0 + c_1\omega - c_2 - c_3\omega \\
v_2 &= c_0 - c_1 + c_2 - c_3 \\
v_3 &= c_0 - c_1\omega - c_2 + c_3\omega.
\end{aligned}$$

b) Da die Sortierphase schon in Abb. 3 erledigt ist, müssen wir hier lediglich noch die Kombinationsphase mit dem Butterfly-Operator durchführen.
Dafür gilt:

$$\begin{aligned}
Z_0 &= z_0 + \omega^0 z_2 \\
Z_1 &= z_0 - \omega^0 z_2 \\
Z_2 &= z_1 + \omega^0 z_3 \\
Z_3 &= z_1 - \omega^0 z_3
\end{aligned}$$

mit $\omega = e^{i\frac{2\pi}{2}}$. Zu beachten ist hierbei die Abhängigkeit des ω von n. In der ersten Kombinationsrunde ergeben IDFT-Lösungen der Größe 2, also $n = 2$. Damit wenden wir nun den Butterfly ein zweites Mal an und erhalten

$$\begin{aligned}
v_0 &= Z_0 + \omega^0 Z_2 = z_0 + z_2 + (z_1 + z_3) = c_0 + c_1 + c_2 + c_3 \\
v_1 &= Z_1 + \omega^1 Z_3 = z_0 - z_2 + \omega\,(z_1 - z_3) = c_0 + c_1\omega - c_2 - c_3\omega \\
v_2 &= Z_0 - \omega^0 Z_2 = z_0 + z_2 - (z_1 + z_3) = c_0 - c_1 + c_2 - c_3 \\
v_3 &= Z_1 - \omega^1 Z_3 = z_0 - z_2 - \omega\,(z_1 - z_3) = c_0 - c_1\omega - c_2 + c_3\omega
\end{aligned}$$

mit $\omega = e^{i\frac{2\pi}{4}}$, was tatsächlich identisch zum Ergebnis der direkten Formel aus a) ist.

4) Zusatzaufgabe: Sound-Effekte oder: Wozu brauchen wir das überhaupt?

Neben zahlreichen anderen Anwendungen spielt (schnelle) Fourier-Transformation eine wichtige Rolle bei Sound-Effekten, die wir hier am Beispiel eines einfachen Spieleszenarios behandeln wollen. Betritt man z. B. in einem Spiel eine Kathedrale, so sollten die Schritte oder auch der Sound von Schüssen von den Wänden zurückhallen um ein realistischeres Erlebnis für den Spieler zu schaffen. Zuerst sollten wir uns deshalb überlegen, wie so ein Zurückhallen überhaupt gemessen werden kann, bevor wir unsere Soundsamples (Schritte, Schüsse, etc.) verarbeiten.

Beim Zurückhallen wird zu jedem Geräusch ein bestimmter Anteil nach einer bestimmten Zeit »zurückgeschickt«. Um diese Anteile festzustellen, könnte man in einer Halle einen einzelnen Impuls aussenden - z. B. durch das Klatschen in die Hände, durch einen Impuls der im Computer erzeugt wurde, durch einen lauten Knall o. ä. Methoden. Direkt nach dem Impuls wird der Rückhall über ein Mikrofon gemessen.

Dieses Ergebnis speichern wir uns in dem Vektor $\kappa = (\kappa_0, \ldots, \kappa_{T-1})$ und nutzen dies um ein allgemeines Signal wie z. B. das Geräusch unserer Fußschritte mit einem ähnlichem Hall zu versehen. Dies lässt sich mit der sogenannten Faltung (Symbol $*$) berechnen:

Für ein Eingangssignal $E(t)$ ergibt sich das Ausgangssignal $A(t)$ wie folgt:

$$A(t) = (E * K)(t) = \sum_{i_t=0}^{T-1} E(i_t) \cdot K(t - i_t)$$

K wird Kernelfunktion genannt. Sie beschreibt, welche Signale des Eingangssignal E zum Zeitpunkt t ausgegeben werden ($A(t)$). Sie gibt zu diskreten Abtaststellen $x \in [-k; k-1]$ die dazugehörige Komponente des Vektors κ zurück (Teilaufgabe c)). Welche das sein wird, werden wir im Laufe des Aufgabenblattes feststellen. Desweiteren legen wir fest, dass $K(x) = 0$ für alle $x \notin [-k; k-1]$ und $E(t) = 0$ für alle $t \notin [0; T-1]$.

Wie man sehr schnell sieht, ist der Rechenaufwand linear für jeden Samplingpunkt zum Zeitpunkt t und sogar quardatisch für alle Samplingpunkte unseres Echointervalls $[0; T-1]$. Eine Möglichkeit wäre, das Echo nur für bestimmte Kernelelemente zu aktivieren, in der Kernelfunktion also > 0 zu setzen (die anderen Elemente des Vektors κ würde man dann auf 0 setzen), um damit die teure quadratische Auswertung zu "mildern". Allerdings wollen wir einen realistischen Sound erzeugen und gehen davon aus, dass der Kernelvektor voll besetzt ist.

Um aus dem Dilemma der quadratischen Kosten zu entkommen ermöglicht uns die FFT einen Ausweg: *Die Faltung im Frequenzraum.* Diese erlaubt es uns nach der Transformation in den Frequenzraum (im Weiteren mit F bezeichnet bzw. mit F^{-1} für die Rücktransformation) die Faltung komponentenweise ($\cdot$) zu vollziehen:

$$(E * K) = F^{-1}(F(E * K)) = F^{-1}(F(E) \cdot F(K))$$

a) Überlegen Sie sich, wie ein Algorithmus für eine solche Verarbeitung eines Soundsignals aussehen müsste. Erstellen Sie eine Skizze mit Eingabesignalen, Kernel, Faltung und dem Ausgabesignal.

b) Zeigen Sie, dass die Faltung mit $\kappa_a = \delta_{a,0}$ das Eingangssignal E unverändert zurück gibt.

c) Zeigen Sie, dass die Faltung mit $\kappa_a = \delta_{a,c}$ das Eingangssignal $E(-c)$ für eine Konstante c zurück gibt. Was bedeutet das für den Faltungskern in ihrem skizzierten Algorithmus?

Lösung:

a) In der beiliegenden Grafik ist der Algorithmus schematisch aufgezeichnet.

Für ein beliebiges Eingabesignal ("Incoming Sound Signal" in der Grafik) soll ein Echo durch eine Faltung mit einem Kernel erzeugt werden.

Wie in der schematischen Zeichnung des Faltungskerns ("Kernel") zu sehen ist, gibt es hierbei einen Identitätspeak ("Identity with height = 1"), der alleine angewandt das ursprüngliche Signal wiedergeben (siehe Teilaufgabe b)) würde. In der Zeichnung wurden desweiteren 2 Echoanhebungen ("Echos") eingeführt, die nicht direkt das Echo zurück geben sondern lediglich einen gewissen verzerrten Anteil vom ursprünglichen Signal (daher Höhe > 1).

Um unseren Algorithmus zu vereinfachen nehmen wir an, dass es nur ein Fenster mit doppelter Kernel-Größe gibt ("Filter window"). Der Kernel (und damit das Filter-Fenster) ist in der Praxis kleiner als $[0, T-1]$. Zum Einen führt das zu einer reduzierten Laufzeit (die Faltung findet nur für die Dauer des Filter-Fensters statt), zum Anderen ist es realistisch anzunehmen, dass die Dauer, in der das Echo auftreten kann, kleiner ist als die Spielzeit des Samples, auf das das Echo angewandt werden soll. Verdoppelt man die Fenstergröße, so müssen wir auch die Kernelgröße verdoppeln um eine komponentenweise Faltung im Frequenzraum zu ermöglichen. Zu guter Letzt überführen wir den Kernel schon jetzt in den Frequenzraum, da er sich im weiteren Verlauf der Signalverarbeitung nicht mehr verändert.

i) Signaleingang:
Die Signalverarbeitung erfolgt mit dem Auffüllen des oberen halben Fensterbereichs.

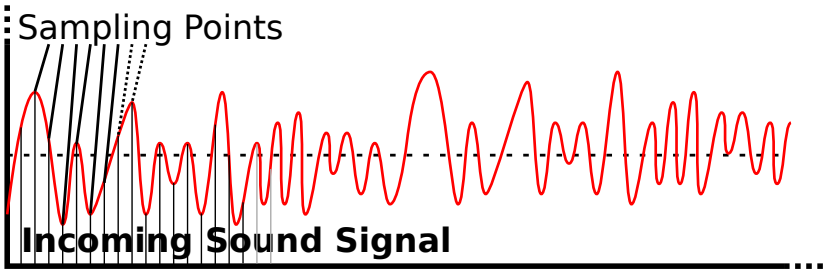
ii) Transformation in den Frequenzraum:
Nachdem ein (weiteres) halbes Fenster an Daten eingelesen wurde und damit die obere Fensterhälfte mit den neuen Daten des Eingabesignals gefüllt ist, verwenden wir unsere FFT um die Signaldaten in den Frequenzraum zu übertragen.

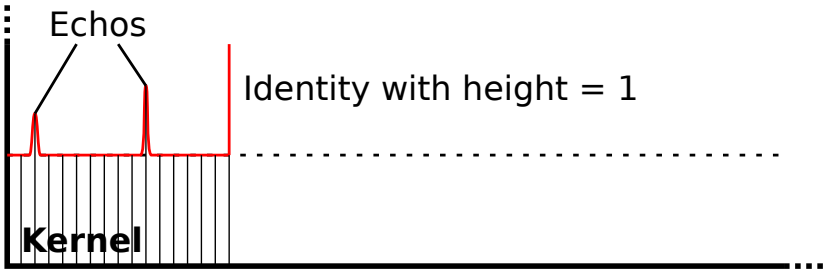
iii) Filter:
Nun können wir komponentenweise den Filter anwenden. Dies benötigt genau soviel Operationen wie das Fenster groß ist, also linear viele Operationen!

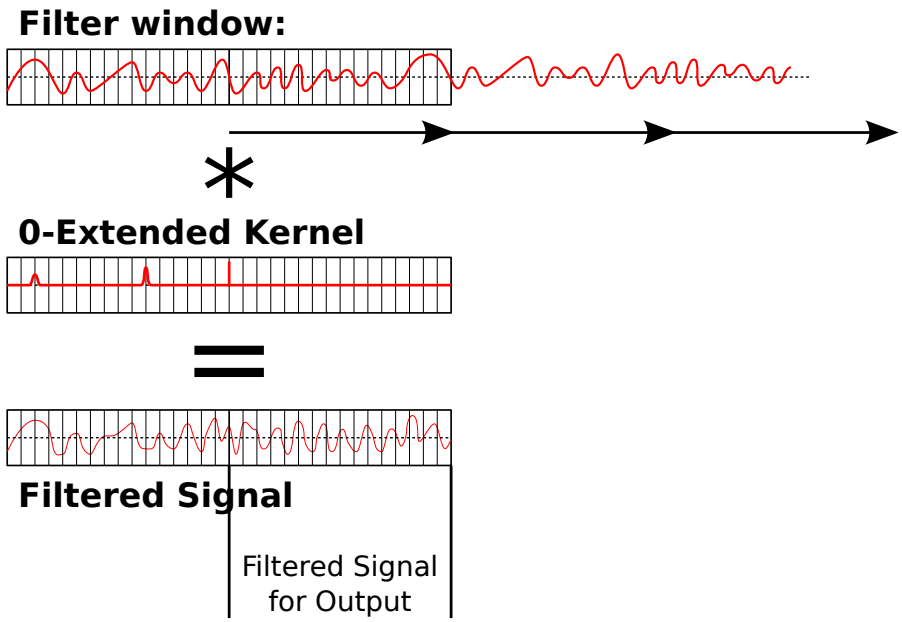
iv) Rücktransformation:
Nach dem Filtern transformieren wir die gefilterten Daten wieder in den Ortsraum (iFFT) und finden in der oberen Hälfte des Fensters unser gewünschtes gefiltertes Signal.

v) Verschieben des Fensters:
Als letzter Schritt muss das Fenster "verschoben" werden was wir einfach dadurch erreichen können indem wir den oberen Teil des Fensters in den unteren Bereich kopieren. Solange noch Eingabedaten vorliegen, wird das Signal ab Schritt 1 weiterverarbeitet.

Ein Beispielprogramm findet sich auf der Website zu dieser Vorlesung (numpro_echo).







b)

$$A(t) = \sum_{i_t=0}^{T-1} I(i_t) \cdot K(t - i_t)$$

Wegen $\kappa_a = \delta_{a,0}$ muss auch $t - i_t = 0$ und $t = i_t$ gelten. Damit erhalten wir:

$$A(t) = \sum_{i_t=0}^{T-1} I(i_t) \cdot \delta_{t-i_t,0} = I(t)$$

c) Zur Veranschaulichung betrachten wir für $c = -6$ den Identitätspeak des Kernel an der Stelle -6. Veranschaulicht würde das bedeuten, dass das Signal um 6 Samplings versetzt ausgegeben wird (was sich allerdings später als falsch herausstellt!)

Mit

$$A(t) = \sum_{i_t=0}^{T-1} E(i_t) \cdot K(t - i_t)$$

und $\kappa_a = \delta_{a,c}$ erhalten wir somit

$$A(t) = \sum_{i_t=0}^{T-1} E(i_t) \cdot \delta_{t-i_t,c}$$

und mit $t - i_t = c \Leftrightarrow i_t = t - c$

$$A(t) = E(t - c)$$

Was bedeutet das nun für unseren Kernel? Statt dass wir z. B. $c = -6$ einsetzen und damit eine um 6 Zeitschritte versetzte Ausgabe erwarten, bekommen wir das Signal zurück, dass 6 Abtastpunkte in der Zukunft liegt. D. h., dass wir unseren Kernel vor der Transformation in den Frequenzraum an der Achse $x = 0$ spiegeln müssen.

Für Interessierte sei hier auf die Vorlesung AWR 1 verwiesen bei der noch schnellere Methoden (z. B. komponentenweise Faltung mit reelen Zahlen statt komplexen) gezeigt werden: `http://www5.in.tum.de/wiki/index.php/AlgorithmsofScientificComputing-Summer11`.