Trigonometrische Interpolation

Für $n = 2^\ell$ können die Koeffzienten des trigonometrischen Polynoms

$$p(x) = c_m \cos(mx) + \sum_{|k|<m} c_k \mathrm{e}^{\mathrm{i}kx}, \quad m = n/2\,,$$

das die Daten

$$f_j = f(x_j), \quad x_j = 2\pi j/n,\, j = 0, \ldots, n-1\,,$$

interpoliert, mit der inversen schnellen Fourier-Transformation berechnet werden:

$$(c_0, \ldots, c_m, c_{-m+1}, \ldots, c_{-1}) = \mathrm{IFFT}(f)\,.$$

Beweis:

zusätzlicher Kosinus-Term $\rightsquigarrow$ gerade Anzahl der Daten
(notwendig für die schnelle Fourier-Transformation)
definiere

$$(c_0, \ldots, c_m, c_{-m+1}, \ldots, c_{-1}) = \tilde{c} = \mathrm{IFFT}(f) = \frac{1}{n} W_n^* f$$

mit der Fourier-Matrix W_n
(Indizierung von $\tilde{c}$ und f von 0 bis $n-1$)
$W_n/\sqrt{n}$ unitär bzw. Definition der inversen diskreten Fourier-Transformation $\implies$

$$f = W_n \tilde{c} \quad \Leftrightarrow \quad f_j = \tilde{p}(x_j) = \sum_{k=0}^{n-1} \tilde{c}_k \mathrm{e}^{\mathrm{i}kx_j} \,,$$

d.h. $\tilde{p}$ erfüllt die Interpolationsbedingungen ersetze Terme bei $\tilde{p}$, ohne Verletzung der Interpolationsbedingungen

- $k = m$:
$$\mathrm{e}^{\mathrm{i}mx_j} = (-1)^j = \cos(mx_j)\,,$$
da $mx_j = (n/2)(2\pi j/n) = \pi j$ und $\mathrm{e}^{\mathrm{i}\pi} = -1$
- $k = m+1, \ldots, n-1$:
$\tilde{c}_k = c_{k-n}$ und
$$\mathrm{e}^{\mathrm{i}kx_j} = \mathrm{e}^{\mathrm{i}(k-n)x_j}\,,$$
da $-nx_j = -2\pi j$ und $\mathrm{e}^{2\pi\mathrm{i}} = 1$

$\Longrightarrow$
$$\tilde{p}(x_j) = \sum_{k=0}^{m-1} c_k \mathrm{e}^{\mathrm{i}kx_j} + c_m \cos(mx_j) + \underbrace{\sum_{k=m+1}^{n-1} c_{k-n}\mathrm{e}^{\mathrm{i}(k-n)x_j}}_{\sum_{k=-m+1}^{-1} c_k \mathrm{e}^{\mathrm{i}kx_j}}$$

Beispiel:

Interpolation der Daten

$$f = (0.2, -6, -0.2, 10, 0.2, -6, -0.2, 10)^{\mathrm{t}}$$

an den Stellen $x_j = 2\pi j/8$, $j = 0, \ldots, 7$, durch ein trigonometrisches Polynom

inverse diskrete Fourier-Transformation $\rightsquigarrow$

$$\tilde{c} = \mathrm{IFFT}(f) = (1, 0, 0.1 + 4\mathrm{i}, 0, -1, 0, 0.1 - 4\mathrm{i}, 0)^{\mathrm{t}}$$

Umindizierung ($\tilde{c}_k = c_{k-8}, k = 5, 6, 7$) $\rightsquigarrow$

$$c = (c_{-3}, \ldots, c_4)^{\mathrm{t}} = (0, 0.1 - 4\mathrm{i}, 0, 1, 0, 0.1 + 4\mathrm{i}, 0, -1)^{\mathrm{t}}$$

interpolierendes trigonometrisches Polynom

$$p(x) = (0.1 - 4\mathrm{i})\mathrm{e}^{-2\mathrm{i}x} + 1 + (0.1 + 4\mathrm{i})e^{2\mathrm{i}x} - \cos(4x)$$

Daten f reell $\implies$ p reell

$$(0.1 - 4\mathrm{i})\mathrm{e}^{-2\mathrm{i}x} + (0.1 + 4\mathrm{i})\mathrm{e}^{2\mathrm{i}x} = 0.2\cos(2x) - 8\sin(2x)$$

Beispiel:

Die trigonometrische Interpolation in Verbindung mit der diskreten Fourier-Transformation kann zum Ausblenden hochfrequenter Störungen in Signalen verwendet werden.

interpoliere die Daten

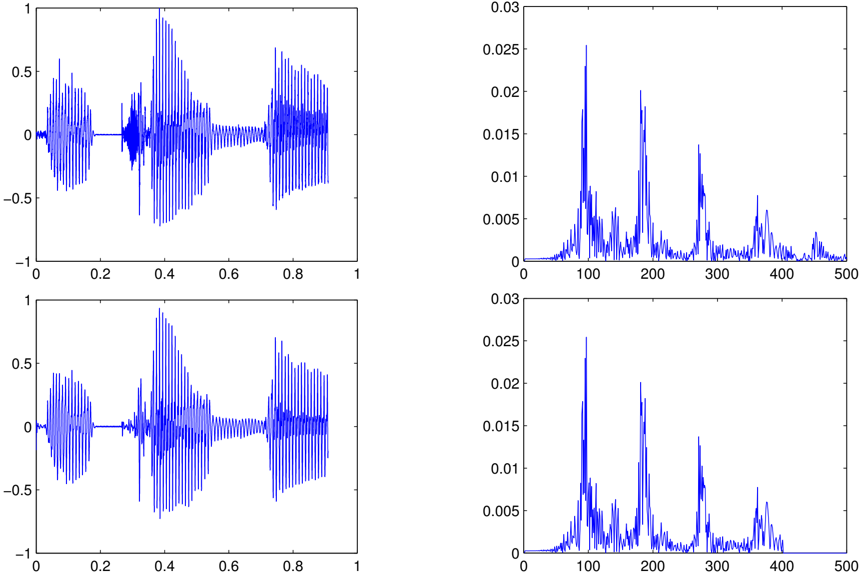
$$f_j \approx f(x_j), \quad x_j = \frac{2\pi j}{n}, \quad 0 \leq j < n = 2^\ell,$$

mit einem trigonometrischen Polynom

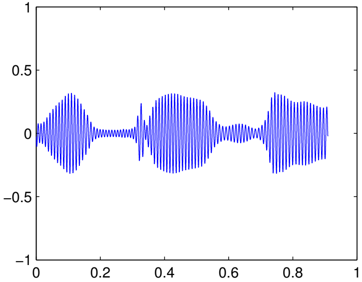
$$p(x) = c_m \cos(mx) + \sum_{|k|<m} c_k \mathrm{e}^{\mathrm{i}kx}, \quad m = n/2$$

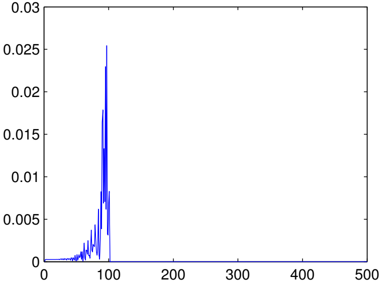
Tiefpass mit Bandbreite M:
setze alle Koeffizienten c_k mit $|k| > M$ null
$\rightsquigarrow$ Unterdrückung von Störungen für hinreichend kleines M

Sprachsignal $f(x)$ 500 der 40000 Amplituden $|\mathrm{Re}(c_k)|$



Glättungseffekt für die Bandbreite $M = 400$





Glättungseffekt für die Bandbreite $M = 100$

kleine Bandbreite $\rightsquigarrow$ unerwünschter Genauigkeitsverlust

Bei der Implementierung ist zu beachten, dass die inverse diskrete Fourier-Transformation den permutierten Koeffizientenvektor $\tilde{c}$ berechnet:

$$(\tilde{c}_0, \ldots, \tilde{c}_{n-1}) = (c_0, \ldots, c_m, c_{-m+1}, \ldots, c_{-1})$$

Programmsegment für einen Fourier-Filter:

IFFT: $f_j \to \tilde{c}_k$
$c_{M+1} = \tilde{c}_{M+1}, \ldots, \tilde{c}_{n-1-M} = c_{-M-1}$ auf null setzen
FFT: $\tilde{c}_k \to p(x_j)$

Hochpass:
Nullsetzen der unteren Koeffizienten