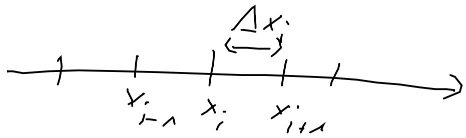
Fortsetzung Fouriertransformation

Erster Schritt zur Umsetzung diskretisieren:

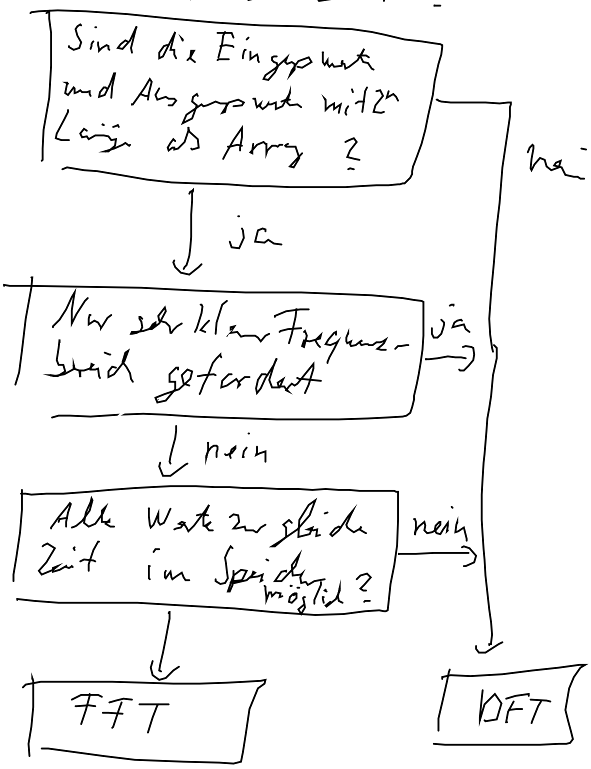
$$f(k) = \int_{-\infty}^{\infty} dx\, e^{-ikx} f(x) \simeq \sum_i \Delta x_i\, e^{-i k_j x_i} f(x_i)$$

Intervalle

muss nicht äquidistant sein



FFT oder DFT?



FFT unterstützt 2^n Eingangs- und Ausgangsarrays.

DFT unterstützt beliebigen Frequenzbereich

FFT benötigt kompletten Eingangsarray, DFT kann mitlaufend berechnet werden.

FFT Algorithmus

Hauptidee: Beschleunigen des Algorithmus durch schon berechnete Zwischenwerte.

Zunächst weiter diskretisieren. Annahme $k_j = \Delta k\, j$ und $x_i = i\,\Delta x$ äquidistant. mit $\Delta k \cdot \Delta x = \frac{2\pi}{N}$ ← Anzahl der Punkte

$$f(k_j) = \sum_i \Delta x\, e^{-i\frac{2\pi}{N} i j} f(x_i) = \Delta x \sum_i \left(e^{-i\frac{2\pi}{N}}\right)^{ij} f(x_i)$$

$$= \Delta x \sum_{i=0}^{N-1} w_N^{ij} f(x_i) \qquad \text{mit } w_N = e^{-i\frac{2\pi}{N}}$$

Man kann jetzt zeigen, dass die Fourier Transformation von N Elementen auf zwei FT mit je $\frac{N}{2}$ Elementen zurückgeführt werden kann.

a) und zwar für gerade Elemente!

$$f(k_{2j}) = \Delta x \sum_{i=0}^{2n-1} w_N^{i\cdot 2j} f(x_i) = \Delta x \sum_{i=0}^{n-1} \Big(\underbrace{w_N^{2ij} f(x_i)}_{\text{erste Hälfte}} + \underbrace{w_N^{2(i+n)j} f(x_{i+n})}_{\text{zweite Hälfte}}\Big)$$

$$= \Delta x \sum_{i=0}^{N-1} \Big(w_N^{2ij} \big(f(x_i) + f(x_{i+n})\big)\Big)$$

w_N^{2ij}, da

Beweis: $w_N^{2\ell(i+n)} = w_N^{2\ell n}\, w_N^{2\ell i}$

$= w_N^{2\ell i}$, da $w_N^{2n\ell} = e^{-i\frac{2\pi}{N}2\ell n} = e^{i2\pi\ell} = 1$

Definiere $y_i = f(x_i) + f(x_{i+n})$

und bemerke $w_N^2 = w_n$

Dann ist $f(k_{2j}) = \sum_{i=0}^{n-1} y_i\, w_n^{ij}$

b) und für ungerade Elemente

$$f(k_{2j+1}) = \Delta x \sum_{i=0}^{2n-1} f(x_i)\, w_N^{(2j+1)i} = \sum_{i=0}^{2n-1} \Delta x \Big[\underbrace{\big(f(x_i)}_{\text{erste Hälfte}} - \underbrace{f(x_{i+n})\big)}_{\text{zweite Hälfte}} w_N^i\Big] w_N^{2ij}$$

w_N^{2in}

Definiere $y_{n+i} = \big(f(x_i) - f(x_{i+n})\big)\, w_N^i$

$$f(k_{2j+1}) = \sum_{i=0}^{n-1} y_{i+n}\, w_n^{ij}$$

Die Schritte a) und b) lassen sich falls N eine Potenz von 2 ist, wiederholen!

Jeder Rechenschritt benötigt $\frac{N}{2}$ komplexe Multiplikation.
also Rechenaufwand $(\frac{N}{2} \log_2 N)$.

$\Rightarrow$ Rekursive Implementation der FFT

Schnelle Implementation für FFT auf CPU ist FFTW
fastest Fourier Transform of the West (de facto Standard)

FFTWSSI (parallelisiert)

<u>DFT Algorithmus</u> (Tipps zur Implementation in der Praxis)

<u>Vorteil:</u> die Ausgangsfkt muss nicht gespeichert werden
(Mitlaufen lassen des Algorithmus)
· Spezieller Spektralbereich nur interessant.

z.B. $[\omega_{start}, \omega_{end}]$ z.B. mit M Punkten

Also $\omega_{start} + l\,\Delta\omega$

Dann hilft der folgende Trick (da exp Fkt teuer!)

$$\text{OmegaStart}[l] = \exp(i(\omega_{start} + l\Delta\omega)t_0)$$

z.B. $\text{OmegaStart}[0] = \exp(i(\omega_{start})t_0)$

$\text{OmegaStartSchritt} = \exp(i\Delta\omega t_0)$

Verwende: $\text{OmegaStart}[l+1] = \text{OmegaStartSchritt} \times \text{OmegaStart}[l]$

Analog: baue ein Array:

$$\text{OmegaStep}[l] = \exp(i(\omega_{start} + l\Delta\omega)\Delta t)$$

Sowie ein Array für die aktuelle Zeit
$\text{OmegaCurrent}[l] = \text{OmegaStart}[l]$ initialisiert.

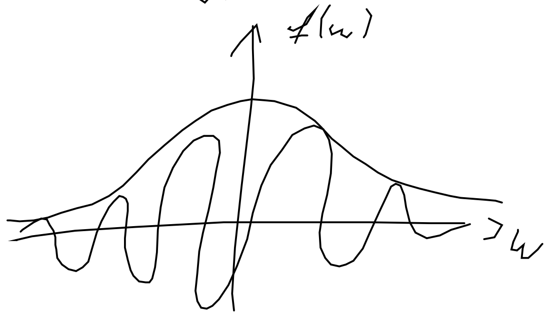
Von $t \to t + \Delta t$:

$\text{OmegaCurrent}[l] *= \text{OmegaStep}[l]$

Bracketing DFT

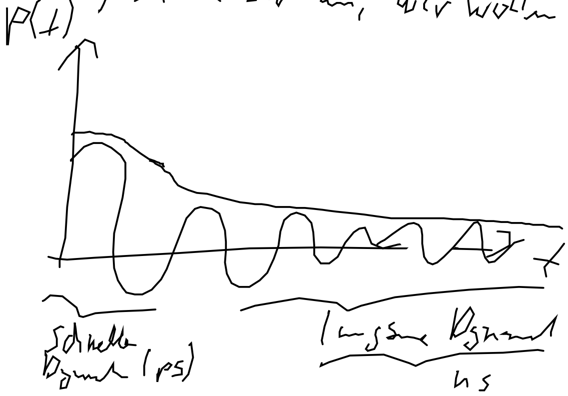
DFT[l] += OmegaConst[l] × f(t) · Δt

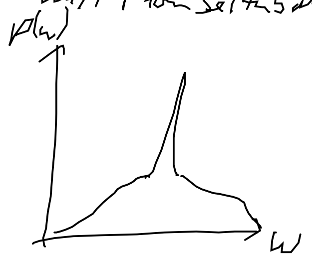
Bemerkung: Typisches Verhalten



VI.15 Harmonic Inversion

Nehmen wir an, wir wollen ein Spektrum berechnen





: Hier ist die Rechnung sehr aufwendig, es wäre sehr nachteilig die Dynamik bis in den Nanosekundenbereich zu berechnen

$$P(\omega) = \int_0^T p(t)\, e^{i\omega t}$$ für Fourier Transformation

Mit der maximalen Auflösung $\Delta E \sim \frac{\hbar}{T}$

Ziel: Harmonic Inversion Algorithmus, der es erlaubt nur einen kleinen Bereich der Zeitdynamik zu berechnen und trotzdem das Spektrum zu erhalten.

Wir folgen Mandelshtam und Taylor, J. Chem. Phys. 107 (17) (1997). Der Algorithmus ist die Basis der Library „harminv".

Filter-Diagonalisierungs Methode für harmonische Inversion

Idee: Wir tun so, als ob es ein fiktives Quantensystem gäbe, das unsere Korrelation $C(t)$ beschreibt.

Also betrachte $C(t) := \varphi(t)$ und wir diskretisieren:

$c_n := C(t_n) := C(n\tau)$, wir tun so als gäbe es

$\hat{U}$ und Ω und $|\phi_0\rangle$, so dass

$$c_n = \langle\phi_0|\underbrace{\hat{U}^n}_{\text{Zeitpropagator } e^{-i\tau\Omega}}|\phi_0\rangle = \langle\phi_0|e^{-in\tau\hat{\Omega}}|\phi_0\rangle$$

Wobei der Zeitpropagator bzgl. Eigenbasis $|\Psi_k\rangle$

$$\hat{U} = \sum_{k=1}^{K} u_k |\Psi_k\rangle\langle\Psi_k| \text{ und der Eigenwert } u_k = e^{-i\tau\omega_k}$$

Wir kombinieren die beiden Gleichungen

$$c_n = \sum_{k=1}^{K} \underbrace{|\langle\phi_0|\Psi_k\rangle|^2}_{d_k} u_k^n = \sum_{k=1}^{K} d_k\, e^{-in\tau\omega_k}$$

Idee: Jedes Signal lässt sich durch abfallende Exponentiale beschreiben. $\Leftarrow$ Ziel von Harmonic Inversion ist es d_k und ω_k zu extrahieren.

Für eindeutige Frequenzen kann man $-\frac{\pi}{\tau} < \mathrm{Re}\,\omega_k < \frac{\pi}{\tau}$ wählen.

$\Rightarrow$ Nyquist Rate, sowie $\mathrm{Im}\,\omega_k \leq O(N\tau)$ (sollte innerhalb des Intervalls substantiell zerfallen)

$\Rightarrow$ Wie können wir die Frequenz ω_k sehen?

$\Rightarrow$ Diagonalisieren Ω oder $\hat{U}$ diagonalisieren.

Anmerkung: Das wäre möglich?! Wähle vollständige Basis $|\chi_j\rangle$ ($j \in \{1,\dots,k\}$)

Dann ist

$$U_{jj'} = \langle \chi_j | \hat{U} | \chi_{j'}\rangle \qquad S_{jj'} = \langle \chi_j | \chi_{j'}\rangle$$

führt das zum verallgemeinerten Eigenwertproblem (Achtung die Vektoren sind nicht orthogonal)

$$\underline{\underline{U}} \cdot \underline{B}_k = u_k \, \underline{\underline{S}} \cdot \underline{B}_k$$

↳ $\underline{B}_k$ sind die Eigenvektoren, Darstellung der Spaltenvektoren als

$$|\psi_k\rangle = \sum_{j=1}^{k} B_{jk} |\chi_j\rangle$$

bzgl. der Basis $|\chi_j\rangle$

wobei $d_k = |\langle \phi_0 | \psi_k\rangle|^2 = |\sum_{j=1}^{k} B_{jk} \langle \phi_0 | \chi_j\rangle|^2$

Problem: Was ist die Basis die wir verwenden?

Erstelle die ⇒ Krylov Basis zu U!

$$|\Phi_n\rangle = U^n |\phi_0\rangle \quad \text{mit } n = 0, 1, \dots, \mu.$$

Für $\mu = k-1$ ist die Basis vollständig, für $\mu > k-1$ übervollständig.

Die Matrixelemente haben damit direkte Beziehung zu c_i:

Wie sieht $U_{jj'}$ in der Basis aus:

$$\langle \Phi_n | \hat{U} | \Phi_{n'}\rangle = \langle \Phi_0 | (U^n)(U^{n'+1}) | \phi_0\rangle = \langle \phi_0 | U^{n+n'+1} | \phi_0\rangle$$

Summen hinschreiben $= c_{n+n'+1}$

$\langle \Phi_n | \Phi_{n'} \rangle = c_{n+n'}$ $\quad \langle \Phi_n | \bar{\Phi}_0 \rangle = c_n$

Das Eigenwertproblem zur Lösung hat die Form:

$$\sum_{n=0}^{M} c_{n+n'+1} B_{nk} = u_k \sum_{n=0}^{M} c_{n+n'} B_{nk}$$ mit den Eigenwert. $u_k = e^{-i\tau\omega_k}$

$$d_k = \left(\sum_{n=0}^{M} B_{nk} c_n \right)^2$$