Numerické metody a programování

Lekce 9

Rychlá Fourierova transformace (FFT)

Fourierova transformace

$$H(f) = \int_{-\infty}^{\infty} h(t)\, e^{-i2\pi f t}\, dt$$

inverzní Fourierova transformace

$$h(t) = \int_{-\infty}^{\infty} H(f)\, e^{i2\pi f t}\, df$$

některé vlastnosti:

- škálování

$$h(at) \Leftrightarrow \frac{1}{|a|} H\left(\frac{f}{a}\right)$$

- posuv

$$h(t - t_0) \Leftrightarrow H(f)\, e^{-i2\pi f t_0}$$

- konvoluce

$g * h \Leftrightarrow G(f) H(f)$, kde $g * h \equiv \int_{-\infty}^{\infty} g(\tau)\, h(t-\tau)\, d\tau$

Vzorkování

vzorkování

$$h_n = h(n\Delta)$$

vzorkovací frekvence: Δ^{-1}

Nyquistova frekvence

$$f_c = \frac{1}{2\Delta}$$

vzorkovací teorém:

pokud $H(f) = 0,\ |f| \geq f_c$ (omezená šířka spektra)

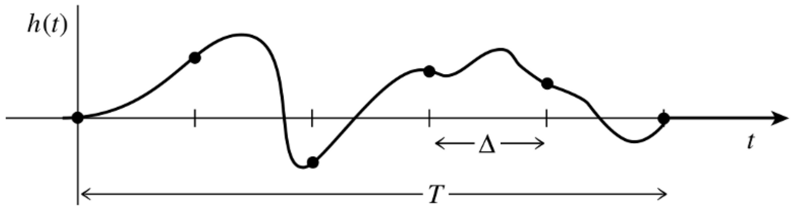
$$h(t) = \Delta \sum_{-\infty}^{\infty} h_n \frac{\sin[2\pi f_c (t - n\Delta)]}{\pi (t - n\Delta)}$$

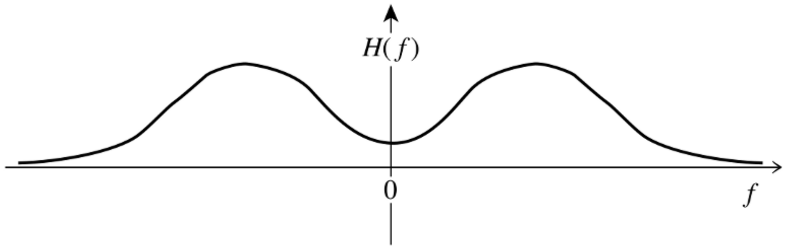
aliasing:

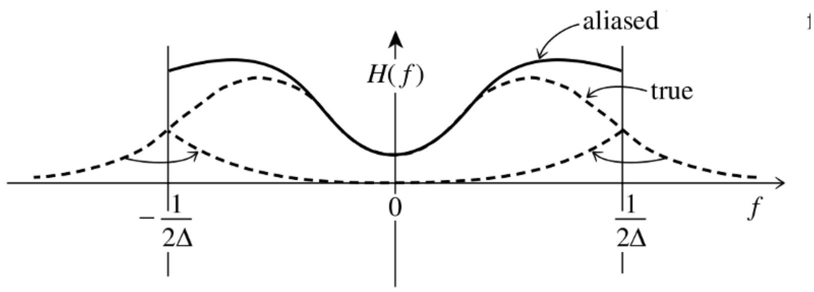
pokud $H(f) \neq 0, \ |f| \geq f_c$, je odpovídající část spektra přesunuta do intervalu $(-f_c, f_c)$

např. funkce $\exp(-i2\pi f_1 t)$ a $\exp(-i2\pi f_2 t)$

vzorkování Δ : stejné vzorky pokud $f_1 - f_2 = k\dfrac{1}{\Delta}$







Diskrétní Fourierova transformace (DFT)

N (sudý) počet vzorků funkce $h(t)$

$$h_k = h(t_k), \quad t_k = k\Delta, \quad k = 0,1,2,\dots,N-1$$

hledáme Fourierovu transformaci v bodech

$$f_n = \frac{n}{N\Delta}, \quad n = -\frac{N}{2},\dots,\frac{N}{2}-1$$

dostaneme

$$H(f_n) = \int_{-\infty}^{\infty} h(t)\, e^{-i2\pi f_n t} dt \approx \Delta \sum_{k=0}^{N-1} h_k\, e^{-i2\pi k n/N}$$

DFT je pak

$$H_n = \sum_{k=0}^{N-1} h_k e^{-i2\pi kn/N}$$

periodicita $H_{-n} = H_{N-n}$ proto se obvykle volí posun spektra: $n = 0, 1, \ldots, N-1$

inverzní DFT

$$h_k = \frac{1}{N} \sum_{n=0}^{N-1} H_n e^{i2\pi kn/N}$$

Rychlá Fourierova transformace (FFT)

maticový zápis DFT

$\boldsymbol{H} = \boldsymbol{W} \cdot \boldsymbol{h}$, kde

$$W_{nk} = w_N^{nk}, \quad w_N = e^{-i2\pi/N}$$

násobení vektoru maticí vyžaduje $O(N^2)$ operací násobení

FFT algoritmus provede transformaci pouze s $O(N \log_2 N)$ operacemi

princip:

DFT délky N lze vyjádřit pomocí dvou DFT poloviční délky (sudé a liché prvky)

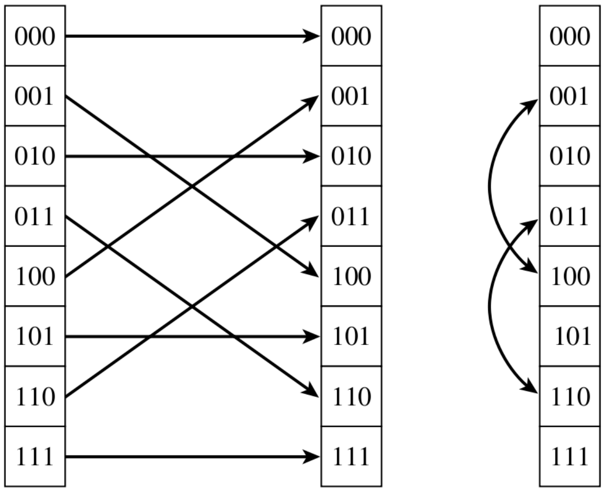
$$\begin{aligned} H_n &= \sum_{k=0}^{N-1} h_k e^{-i2\pi kn/N} \\ &= \sum_{k=0}^{N/2-1} h_{2k} e^{-i2\pi(2k)n/N} + \sum_{k=0}^{N/2-1} h_{2k+1} e^{-i2\pi(2k+1)n/N} \\ &= H_n^e + w_N^n H_n^o \end{aligned}$$

toto dělení lze opakovat až na úroveň DFT délky jedna

FFT algoritmus

- data setříděna v bitově obráceném pořadí (sudost/lichost = poslední dvojková číslice)
- takto uspořádaná data představují transformace délky jedna
- sousední prvky vytvoří transformace délky dva
- toto se opakuje, až vznikne DFT vstupních dat

bitová reverze



např. pro $N=4$

$$\begin{pmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{pmatrix} \rightarrow \begin{pmatrix} H_0^{ee} \\ H_0^{eo} \\ H_0^{oe} \\ H_0^{oo} \end{pmatrix} = \begin{pmatrix} h_0 \\ h_2 \\ h_1 \\ h_3 \end{pmatrix} \rightarrow \begin{pmatrix} H_0^e = H_0^{ee} + w_2^0 H_0^{eo} \\ H_1^e = H_1^{ee} + w_2^1 H_1^{eo} \\ H_0^o = H_0^{oe} + w_2^0 H_0^{oo} \\ H_1^o = H_1^{oe} + w_2^1 H_1^{oo} \end{pmatrix} \rightarrow \begin{pmatrix} H_0 = H_0^e + w_4^0 H_0^o \\ H_1 = H_1^e + w_4^1 H_1^o \\ H_2 = H_2^e + w_4^2 H_2^o \\ H_3 = H_3^e + w_4^3 H_3^o \end{pmatrix}$$

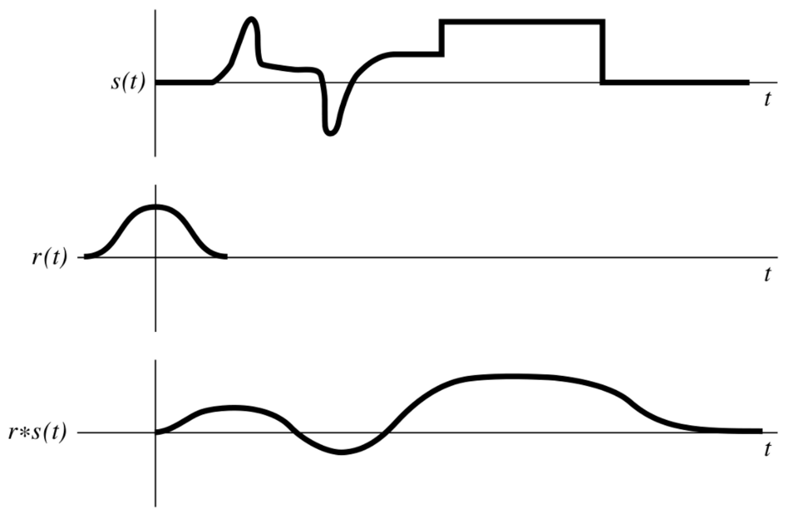
- periodicita FT: $H_1^{ee} = H_0^{ee}$, $H_2^e = H_0^e$ apod.
- celkem $\log_2 N$ úrovní s N operacemi na každou úroveň
- existují zobecnění pro případ kdy N není mocnina 2 (méně efektivní)

FFT ve 2D

$$H_{n_1,n_2} = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} e^{-i2\pi k_1 n_1/N_1} e^{-i2\pi k_2 n_2/N_2} h_{k_1,k_2}$$

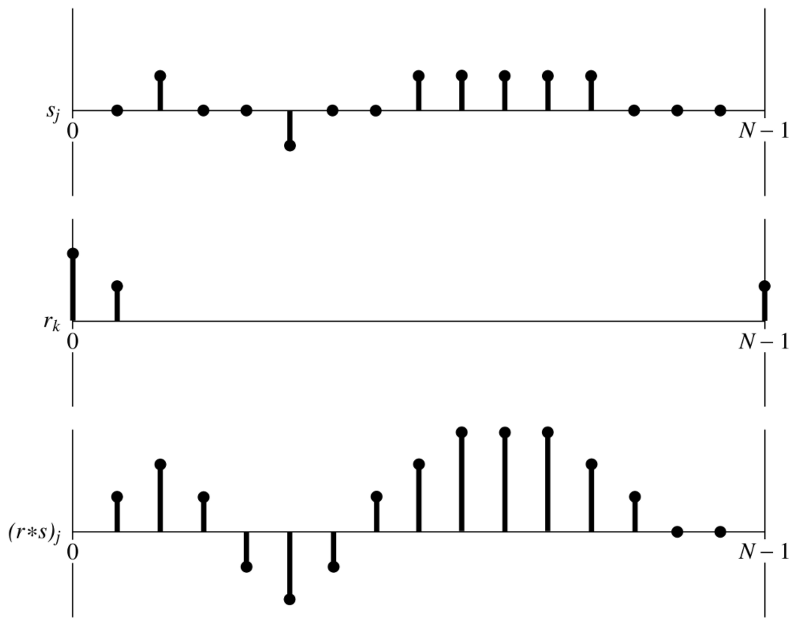
nejprve provedeme 1D FFT přes jeden index, pak 1D FFT přes druhý index

Aplikace FFT: konvoluce



diskrétní konvoluce, odezva h konečné délky N

$$(g*h)_n = \sum_{k=0}^{N-1} g_k h_{n-k}$$



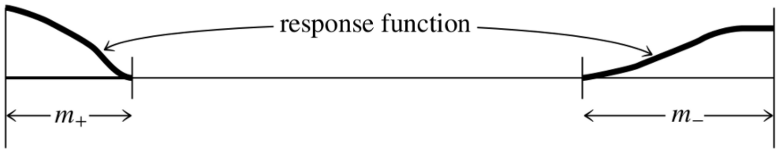
pokud je signál g periodický s periodou N pak

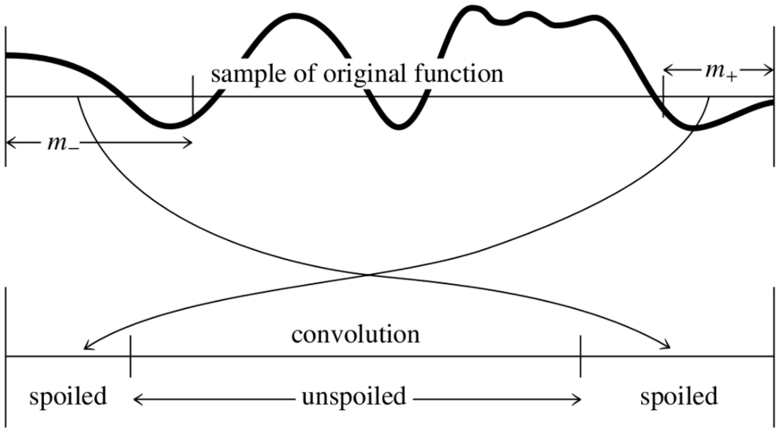
$$\sum_{k=0}^{N-1} g_k h_{n-k} \Leftrightarrow G_n H_n \text{ (konvoluční teorém)}$$

$G_n,\, H_n,\; n=0,1,\ldots,N-1$ jsou DFT hodnot $g_k,\, h_k,\; k=0,1,\ldots,N-1$

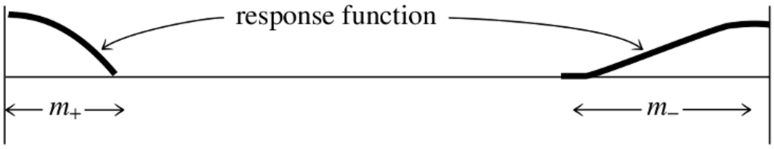
pokud data nejsou periodická projevují se okrajové efekty

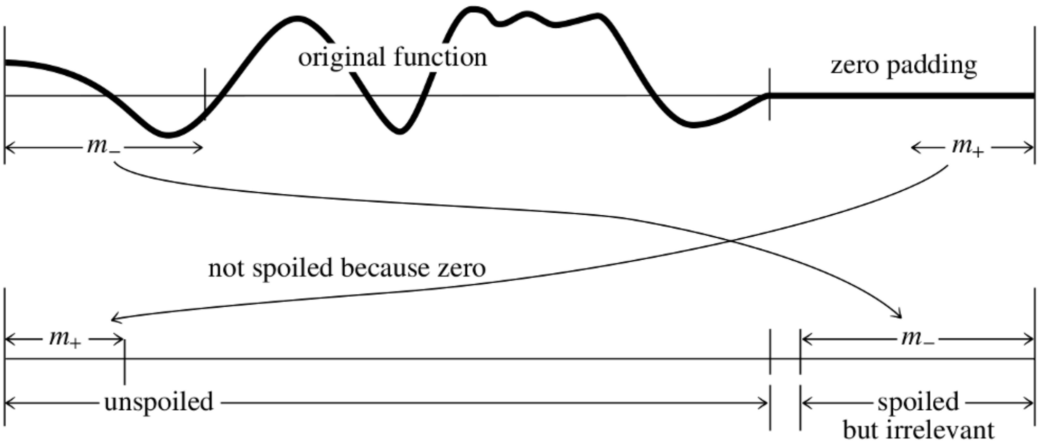
- odezva h obvykle mnohem kratší než N
- prolínání okrajů dat





- problém řeší nastavení funkce nulami





- data se nastaví na jednom z konců počtem nul, který odpovídá delší z polovin odezvy h

algoritmus provádějící konvoluci

- nastavení dat a odezvy nulami na dostatečně velkou délku N
- posun pro FFT
- FFT dat a odezvy $\rightarrow G_n, H_n$
- vytvoření součinu $G_n H_n$
- inverzní FFT

dekonvoluce funguje podobně s tím, že dělíme FFT známé konvoluce FFT odezvy – obdržíme DFT hledané dekonvoluce