11. Fourierova transformace, konvoluce, výpočet integrálů pomocí FT

Fourierova transformace je často používaná metoda při řešení specifických úloh. Octave a Matlab mají implementovánu rychlou Fourierovu transformaci, tzv. FFT algoritmus (Fast Fourier Transform). Podmínkou použití FFT jsou ekvidistantně rozdělená data a optimální počet dat rovný 2^n (pokud není počet dat roven 2^n doplní je Octave/Matlab nulami). Při použití Fourierovy transformace přecházíme z časové souřadnice do frekvenční, resp. z přímého prostoru do komplementárního reciprokého.

Mějme n–hodnot nějaké funkce $f(t)$, měřené v ekvidistatních časových intervalech t_i,

$$f(t) = \{f(t_0), f(t_1), \ldots, f(t_{n-1})\} \quad t_n = n\Delta t.$$

Diskrétní Fourierova transformace je definována vztahy

$$F_k = \frac{1}{T}\sum_{r=1}^{n-1} f(t_r)e^{-i\frac{2\pi k}{T}r\Delta t}\Delta t = \frac{1}{n}\sum_{r=1}^{n-1} f(t_r)e^{-i\frac{2\pi kr}{n}},$$

kde F_k reprezentují koeficienty Fourierovy řady funkce $f(t)$ pro frekvence menší než Nyquistova frekvence $\omega = \frac{\pi}{\Delta t}$,

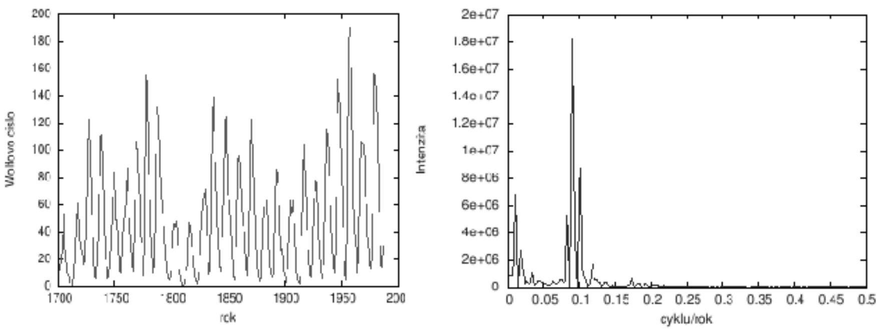
$$\omega_k = \frac{2\pi k}{n\Delta t} = \frac{\pi}{\Delta t} \quad k = 0, 1, ..., \frac{n}{2}$$

Ukažme si aplikaci Fourierovské analýzy při analýze časové série počtu slunečních skvrn na povrchu Slunce, tzv. Wolfova čísla.

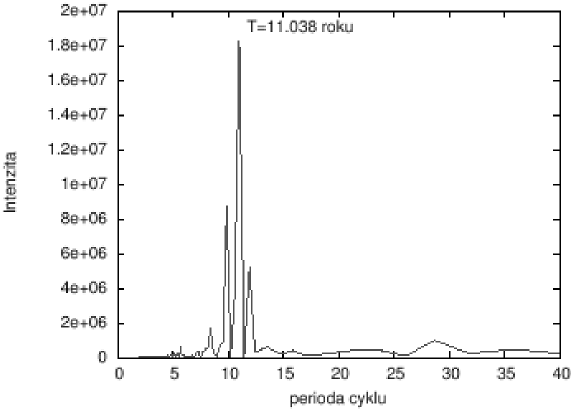
```
octave:1>s=load('sunspot.dat');
octave:2> cas=s(:,1);
octave:3> Wolf=s(:,2);
octave:4> FTWolf=fft(Wolf);
octave:5> FTWolf(1)=[];
     % prvni cislo je suma hodnot v poli
octave:6> n=length(FTWolf);
octave:7> FTWolf2=abs(FTWolf(1:floor(n/2))).^2;
```

```
octave:8> nyquist=1/2;
octave:9> frekv=(1:n/2)/(n/2)*nyquist;
octave:10> out=[frekv' FTWolf2];
octave:11> save -ascii 'periodogram.dat' out
octave:12> perioda=1./frekv;
octave:13> out=[perioda' FTWolf2];
octave:14> save -ascii 'cyklus.dat' out
octave:15> m=find(FTWolf2==max(FTWolf2));
octave:16> perioda(m)
ans =  11.038
```

Na prvním řádku načteme data uložená v souboru `sunspot.dat` do proměnné `s`. První sloupec obsahuje čas, druhý počet slunečních skvrn (Wolfovo číslo). Na čtvrtém řádku provedeme Fourierovu transformaci časové závislosti pomocí funkce `fft` a výsledek uložíme do proměnné `FTWolf`. Na šestém řádku uložíme do proměnné `n` délku pole `FTWolf`. Do proměnné `FTWolf2` uložíme druhou mocninu velikosti prvků (prvky pole `FTWolf` jsou obecně komplexní) z první poloviny pole `FTWolf`, viz dále. Na osmém řádku spočteme Nyquistovu frekvenci (data jsou uložena s Δt=1 (rok)). Na devátém řádku uložíme do vektoru `frekv` frekvence. Dále vytvoříme matici `out`, kterou pak uložíme do souboru `periodogram.dat`. Nyní převedem frekvence na periody (12.řádek), znovu naplní proměnnou `out` a data uložíme do souboru `cyklus.dat`. Na řádku č.15 hledáme index pole odpovídající maximu hodnot vektoru `FTWolf2` a posledním řádku ji zobrazíme. Výsledná hodnota odpovídá délce tzv. slunečního cyklu (11 let). Následující obrázky ukazují grafické znázornění zadaných a spočtených dat:



Obr. 11.36. Počet slunečních skrn (Wolfovo číslo) od roku 1700 (nalevo) a počet cyklů za rok po FFT (vpravo).



Obr. 11.37. Perioda slunečního cyklu získaná z FFT Wolfova čísla.

V předchozím příkladě jsme na řádku číslo 7 použili jen první polovinu hodnot uložených v proměnné `FTWolf`. Tento krok je dán řazením hodnot v poli po provedení příkazu `fft`. Nechť je v poli N hodnot. Na počátku pole jsou uloženy hodnoty Fourierovy transformace (FT) daných hodnot pro nezáporná k $(1 \leq k \leq N/2)$ následováné hodnotami FT pro záporné frekvence $(N/2+1 \leq k \leq N/2)$. Pokud potřebujeme data *přerovnat* tak, aby na začátku pole byla data pro záporné frekvence, použijeme funkci `fftshift`. Inverzní Fourierovou transformaci spočteme příkazem `ifft`.

11.1 Výpočet konvoluce a dekonvoluce pomocí FT

V reálném fyzikálním experimentu je vždy naměřený signál $m(t)$ konvolucí čistého fyzikálního signálu $f(t)$ a instrumentální rozlišovací funkce daného experimentální zařízení $r(t)$,[12]

$$m(t) = f(t) * r(t).$$

Konvoluci můžeme snadno spočítat pomocí FT, kdy platí

$$M(t) = F(t) \cdot R(t),$$

kde M, I a R jsou Fourierovské obrazy m, i a r. Ukažme si výpočet konvoluce a její vliv na měřená data na jednoduchém příkladě. Uvažujme dvě blízké spektrální čáry, které budeme pozorovat pomocí spektrometru. Rozlišení spektrometru budeme simulovat pomocí šířky vstupní clony; pro jednoduchost ji budeme popisovat Gaussovou funkcí s pološířkou σ_w (w od slova window, okno). Profil spektrálních čar budeme také popisovat pomocí Gaussových funkcí. Nejdříve napíšeme funkci, která pro zadané parametry spočte Gaussovu křivku:

```
function y=gauss(x,x0,sigma,A)
% funkce vraci Gaussovu krivku spoctenou pro hodnoty
% v poli x, s polohou maxima v x0, s polosirkou sigma
% a s amplitudou A
   y=A*exp(-(x-x0).^2/2.0/sigma);
```

Zvolíme polohy spektrálních čar, jejich intenzity a pološířky a pološířku "pozorovacího okna", tj. šířku vstupní štěrbiny. Dle vztahu $M(t) = F(t) \cdot R(t)$ spočteme Fouriérův obraz měřeného signálu a pomocí funkcí `ifft` a `fftshift` určíme reálný signál. Ten můžeme nakonec zobrazit pomocí funkce `plot`.

```
x=[-10:0.2:10]; % vektor x-vych souradnic
x1=-2; % poloha 1.spektralni cary
x2=2; % poloha 2.spektralni cary
f=gauss(x,x1,0.1,1)+gauss(x,x2,0.1,1); % cary maji sigma=0.1 a A=1
```

[12] čtenář se jistě setkal i s integrálním zápisem konvoluce

$$m(t) = \int f(y-t)r(t)dy.$$

```
r=gauss(x,0,0.1,1); % instrumentalni rozliseni
M=fft(f).*fft(r); % Fourieruv obraz mereneho signalu
m=fftshift(iff(M)); % "mereny" signalu
plot(x,abs(m));
```

Zkuste si pro různé hodnoty šířky vstupní štěrbiny graficky znázornit funkci $m(t)$. Pro rostoucí hodnoty šířky vstupní štěrbiny už nelze rozeznat dvojici spektrálních čar. V reálném spektrometru ovšem nemůžeme mít šířku vstupní clony nekonečně malou – užší clona znamená menší intenzitu měřeného signálu.

Složitější je výpočet dekonvoluce, kdy potřebujeme zjistit fyzikální signál z naměřeného signálu a známého instrumentálního rozlišení. Tato úloha je však špatně podmíněná, silně závisí např. na poměru signál/šum dat a pod. Podrobný článek o výpočtu dekonvoluce v prostředí Matlab publikoval Doc.Rafaja [Raf1], [Raf2].

11.2 Výpočet integrálů pomocí FFT

Velmi často se ve fyzice setkáváme s výpočtem integrálů typu

$$I(\omega) = \int_a^b f(t)e^{i\omega t}dt \quad I_c(\omega) = \int_a^b f(t)\cos(i\omega t)dt \quad I_s(\omega) = \int_a^b f(t)\sin(i\omega t)dt\,,$$

kde ω je kruhová frekvence. Při výpočtu těchto integrálů pomocí FFT je třeba zohlednit kraje intervalu integrace. Počet dat musí být roven mocnině $N = 2^n$. Pokud tak není, a máme jen M dat, doplníme data nulami,

$$N \geq M + 1, \quad N = 2^n \quad N < M + 1, \quad f_j = 0 \quad M + 1 < j \leq N - 1.$$

Hodnota N určuje i maximální hodnotu frekvence ω, do které můžeme integrovat. Hodnotu integrálu spočteme podle vztahu (podrobnosti viz. Numerical Recipes [NR])

$$\begin{array}{rcl} I(\omega_n) & \approx & \Delta e^{i\omega_n a}\left\{W(\theta)\left[FFT(f_0,\ldots,f_{N-1})\right]_n\right. \\ & + & \alpha_0(\theta)f_0 + \alpha_1(\theta)f_1 + \alpha_3(\theta)f_3 + \cdots + \\ & + & \left. e^{i\omega_n(b-a)}\left[\alpha_0^*(\theta)f_M + \alpha_1^*(\theta)f_{M-1} + \alpha_3^*(\theta)f_{M-3} + \ldots\right]\right\} \\ & & \alpha_1 = \alpha_2 = \alpha_3 = 0, \end{array}$$

kde * značí komplexně združené číslo, $f_j = f(t_j)$ a v lichoběžníkové aproximaci

$$\begin{array}{rcl} W(\theta) & = & \frac{2(1-\cos\theta)}{\theta^2} \approx 1 - \frac{1}{12}\theta^2 + \frac{1}{360}\theta^4 - \frac{1}{20160}\theta^6 \\ \alpha_0 & = & -\frac{1-\cos\theta}{\theta^2} + i\frac{\theta-\sin\theta}{\theta^2} \approx \\ & \approx & -\frac{1}{2} + \frac{1}{24}\theta^2 - \frac{1}{720}\theta^4 + \frac{1}{40320}\theta^6 + i\theta\left(\frac{1}{6} - \frac{1}{120}\theta^2 + \frac{1}{5040}\theta^4 - \frac{1}{362880}\theta^6\right) \end{array}$$