Projekce do bází a Fourierova řada

Jan Černocký, Michal Fapšo FIT VUT Brno

1 Malé opáčko - báze a promítání do nich

Při promítání vektoru do báze použijeme skalární násobení vektorů, abychom dostali skalár, který nám určuje souřadnici vektoru v dané bázi.

Poznámka: pozor při kopírování z pdf do matlabu. Některé znaky se mohou zkopírovat nesprávně (např. '). Kopírujte proto raději ze zdrojového textu v latexu.

Příklad 1.

$\mathbf{x} = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \quad \mathbf{b_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \mathbf{b_2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

```
x=[2;3]; b1=[1;0]; b2=[0;1];
figure();
subplot(121);
hold on;
plotv(x,'b-'); % modra
plotv(b1,'r-'); % cervena
plotv(b2,'r-');

% promitnuti do bazi a vypocteni promitnutych souradnic
x1=x'*b1
x2=x'*b2
cx1=b1.*x1;
cx2=b2.*x2;

p1=plot(cx1(1),cx1(2),'MarkerFaceColor',[0 0 1],'Marker','square', 'Color',[0 0 1]);
set(p1,'Clipping','off')
p2=plot(cx2(1),cx2(2),'MarkerFaceColor',[0 0 1],'Marker','square', 'Color',[0 0 1]);
set(p2,'Clipping','off')
daspect([1 1 1]); % zachovani pomeru vsech os v plotu

% do dalsiho grafu vektor z promitnutych souradnic zesyntetizujeme
subplot(122);
hold on;
% zobrazime obe baze
plotv(b1,'r-'); % cervena
plotv(b2,'r-');
% secteme promitnute souradnice a tak dostaneme puvodni vektor x
synt_x=cx1+cx2;
plotv(synt_x,'g-'); % zelena
daspect([1 1 1]); % zachovani pomeru vsech os v plotu
hold off;
```

Pozn.: Clipping jsme museli vypnout, aby body cx1 a cx2 nebyly oříznuty z obrázku. Funkcí `get(objekt)` *získáte seznam parametrů objektu, které se dají funkcí* `set(objekt, ...)` *nastavit.*

Příklad 2.

Teď si zkusíme něco méně triviálního. Pootočíme souřadnou soustavu:

$$\mathbf{x} = \begin{bmatrix} 2 & 3 \end{bmatrix} \quad \mathbf{b_1} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \quad \mathbf{b_2} = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

```
x=[2;3]; b1=[1/sqrt(2);1/sqrt(2)]; b2=[-1/sqrt(2);1/sqrt(2)];
...
zbytek je stejny jako v predchazejicim prikladu
```

Úkoly:

1. Ověřte v obou případech, že báze jsou ortonormální.

Příklad 3. ...vícerozměrný prostor

Mějme 8-rozměrný vektor $\mathbf{x} = [3\ 2\ 1\ 0\ 1\ 2\ 3\ 4]^T$ a báze $\mathbf{b_1} = \sqrt{\frac{1}{8}}[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^T$, $\mathbf{b_2} = [b_{21}\ b_{22}\ b_{23}\ \ldots\ b_{28}]^T$ $\sqrt{\frac{2}{8}}\cos(2\pi/8n)$

```
n = 0:7; x = [3;2;1;0;1;2;3;4];
subplot(2,3,1); stem(n,x); grid
```

Promítnutí do báze $\mathbf{b_1}$. Pro výukové účely nejdříve vektor $\mathbf{x}$ s bází vynásobíme prvek po prvku, čímž dostaneme vektor, který si zobrazíme. Potom sečteme jeho prvky, čímž dostaneme výsledek skalárního součinu vektoru s bází.

```
% generovani 1. baze
b1 = ones(8,1) * sqrt(1/8);
subplot(2,3,2); stem (n,b1); grid
% vypocet souradnic vektoru x v bazi b1
xb1 = b1 .* x;
subplot(2,3,3); stem (n,xb1); grid
sum(xb1)
```

Promítnutí do báze $\mathbf{b_2}$:

```
% generovani 2. baze
b2 = sqrt(2/8) * cos (2*pi*1/8 * n)'; % potrebujeme sloupcovy vektor
subplot(2,3,5); stem (n,b2); grid
% vypocet souradnic vektoru x v bazi b2
xb2 = b2 .* x;
subplot(2,3,6); stem (n,xb2); grid
sum(xb2)
```

Úkoly:

1. Ověřte v obou případech, že báze jsou ortonormální.

Příklad 4. ...a teď to zkusíme s funkcemi

Podobnost cosinusovky a stejnosměrného signálu: $x(t) = \cos(2\pi t)$, $b(t) = 1$, $\int_0^1 x(t)b(t)dt = ...$

```
krok = 0.01;
t = 0:krok:1;
x = cos(2*pi*t);
subplot(1,3,1); plot(t,x); title('x(t)'); grid
b = ones(1,length(t));
subplot(1,3,2); plot(t,b); title('b(t)'); grid
xb = x.*b;
subplot(1,3,3); plot(t,xb); title('x(t)b(t)'); grid
sum(xb) * krok    % jakoze integral
% vybarveni, abychom videli co se integruje
a1=xb; a1(find(a1<0))=0;
a2=xb; a2(find(a2>0))=0;
hold on;
area(t,a1,'FaceColor',[0 0 1]);
area(t,a2,'FaceColor',[1 0 0]);
```

Pozn.: Výsledek není přesně 0, ale 0.01 protože cosinus začíná i končí hodnotou 1, ale hodnota -1 je tam pouze jednou.

Podobnost cosinusovky s cosinusovkou:

$$x(t) = \cos(2\pi t), \quad b(t) = \sqrt{2}\cos(2\pi t), \quad \int_0^1 x(t)b(t)dt = ... \tag{1}$$

```
...
b = sqrt(2)*cos(2*pi*t);
...
```

Podobnost cosinusovky s 2-krát rychlejší cosinusovkou:

$$x(t) = \cos(2\pi t), \quad b(t) = \sqrt{2}\cos(4\pi t), \quad \int_0^1 x(t)b(t)dt = ... \tag{2}$$

```
...
b = sqrt(2)*cos(4*pi*t);
...
```

Podobnost cosinusovky a sinusovky:

$$x(t) = \cos(2\pi t), \quad b(t) = \sqrt{2}sin(2\pi t), \quad \int_0^1 x(t)b(t)dt = ... \tag{3}$$

```
...
b = sqrt(2)*sin(2*pi*t);
...
```

Úkoly:

1. Zhodnoťte, zda výsledky odpovídají intuici “je to kladné, když je to podobné”.
2. ověřte opět ortonormalitu. Pomůcka: norma funkce $||b(t)|| = \sqrt{\int |b(t)|^2 dt}$

Příklad 5. ...směs sinusovky a cosinusovky

Podobnost cosinusovky se sinusovkou vyšla nulová, takže se nepodobají. Zkusíme tedy teď podobnost se sinusovkou i cosinusovkou:

$$x(t) = \cos(2\pi t - \pi/5) \tag{4}$$

$$b_1(t) = \sqrt{2}\cos(2\pi t) \quad \int_0^1 x(t)b_1(t)dt = ... \tag{5}$$

$$b_2(t) = \sqrt{2}\sin(2\pi t) \quad \int_0^1 x(t)b_1(t)dt = ... \tag{6}$$

Souřadnice (koeficienty) funkce x pro obě báze vypočteme:

$$cb_1 = \int x(t)b_1(t)dt \tag{7}$$

$$cb_2 = \int x(t)b_2(t)dt \tag{8}$$

```
krok = 0.01;
t = 0:krok:1;
x = cos(2*pi*t-pi/5);
b1 = sqrt(2)*cos(2*pi*t);
b2 = sqrt(2)*sin(2*pi*t);

% zobrazeni x, b1, b2
close all; % uzavreni vsech oken, at mame uklizeno
subplot(2,3,1); plot(t,x); title('x(t)'); grid
subplot(2,3,2); plot(t,b1); title('b1(t)'); grid
subplot(2,3,5); plot(t,b2); title('b2(t)'); grid

% promitnuti do bazi
xb1 = x.*b1; % promitnuti funkce x do baze b1
xb2 = x.*b2; % promitnuti funkce x do baze b2

% vykresleni promitnuti
subplot(2,3,3); plot(t,xb1); title('x(t)b1(t)'); grid
subplot(2,3,6); plot(t,xb2); title('x(t)b2(t)'); grid

% vypocet integralu
cb1=sum(xb1) * krok
cb2=sum(xb2) * krok

% zpetne zesyntetizovani puvodni funkce z vypoctenych koeficientu cb1 a cb2
figure()
plot(t,x,t,(cb1*b1 + cb2*b2));
```

Získali jsme tedy 2 koeficienty, které nám udávají souřadnice funkce $x(t)$ v bázovém prostoru daném funkcemi $b_1(t)$, $b_2(t)$. Jaký je závěr?

1. pro $b_1(t)$: je to podobné.
2. pro $b_2(t)$: je to také podobné, ale o něco méně.

Když jsme potom tyto souřadnice zpátky promítli přes tyto báze, dostali jsme původní funkci.

Tyto souřadnice nám však určují pouze amplitudu sinusu a consinusu. Jak tedy můžeme jejich spojením získat zadanou funkci x, posunutý sinus? Když totiž sčítáme sinus a cosinus, přičemž každý vynásobíme nějakou "vahou", výsledná funkce bude různě posunutý sinus (nebo cosinus) s různou amplitudou. Vahami pro sinus a cosinus tedy určujeme nejen amplitudu výsledné funkce, ale také její posunutí.

Příklad 6.

Teď si ukážeme, jako se to dělá elegantněji pomocí komplexních exponenciál. Báze bude:

$$b_1(t) = e^{j\omega_1 t} = \cos\omega_1 t + j\sin\omega_1 t \tag{9}$$

Koeficient, se kterým budeme do této báze promítat, vypočteme:

$$c_1 = \int x(t) b_1^\star(t) dt \tag{10}$$

(báze musí být komplexně sdružená). Když však vynásobíme koeficient c_1 s bází b_1, nedostaneme ještě původný signál. Výsledný signál bude komplexní. Přidáme tedy ještě jednu bázi, která bude s b_1 komplexně sdružená:

$$b_{-1}(t) = e^{-j\omega_1 t} = \cos\omega_1 t - j\sin\omega_1 t \tag{11}$$

Její koeficient pak bude:

$$c_{-1} = \int x(t) b_{-1}^\star(t) dt \tag{12}$$

Poznámka: více se o komplexních exponenciálách dozvíme v následující kapitole o Fourierově řadě.

```
krok = 0.01;
t = 0:krok:1;
x = sin(2*pi*t-pi/5);

% baze pomoci komplexnich exponencial
b1 = exp(j*2*pi*t);
bm1 = exp(-j*2*pi*t);

% vypocet koeficientu
c1 = x*b1'*krok
cm1 = x*bm1'*krok

% baze pomoci sinusu a cosinusu
b1 = cos(2*pi*t) + j*sin(2*pi*t);
bm1 = cos(2*pi*t) - j*sin(2*pi*t);

% vypocet koeficientu
c1 = x*b1'*krok
cm1 = x*bm1'*krok
```

2 Fourierova řada

FŘ vyjadřuje libovolný periodický signál jako sumu komplexních exponenciál:

$$x(t) = \sum_{k=-\infty}^{+\infty} c_k e^{jk\omega_1 t}, \tag{13}$$

kde $\omega_1 = \frac{2\pi}{T_1}$ je základní kruhová frekvence signálu a c_k jsou koeficienty FŘ. Ty se spočítají:

$$c_k = \frac{1}{T_1} \int_{T_1} x(t) e^{-jk\omega_1 t} dt. \tag{14}$$

Pro hloubavé: c_k je tedy komplexní číslo a můžeme ho např. pro k=1 rozepsat na reálnou a imaginární složku, $c_1 = c_1^{\cos} - jc_1^{\sin}$ a pro k=-1 na $c_{-1} = c_1^{\cos} + jc_1^{\sin}$. c_1 a c_{-1} jsou komplexně sdružené, tedy se liší pouze v znaménku imaginární složky. Když sečteme komplexní exponenciály pro k=±1, z rovnice (13) dostaneme:

$$x(t) = (c_1^{\cos} - jc_1^{\sin})e^{j\omega_1 t} + (c_1^{\cos} + jc_1^{\sin})e^{-j\omega_1 t}, \tag{15}$$

pak rozepíšeme komplexní exponenciálu podle rovnice (9) a (11):

$$x(t) = (c_1^{\cos} - jc_1^{\sin})(\cos\omega_1 t + j\sin\omega_1 t) + (c_1^{\cos} + jc_1^{\sin})(\cos\omega_1 t - j\sin\omega_1 t) \tag{16}$$

po roznásobení nám zůstane:

$$x(t) = 2c_1^{\cos}\cos\omega_1 t + 2c_1^{\sin}\sin\omega_1 t \tag{17}$$

Znovu jsme tedy dostali, podobně jako v příkladu 5, směs cosinusovky a sinusovky s vahami $c_1^{\cos}$ (reálná složka) a $c_1^{\sin}$ (imaginární složka), kterou dokážeme popsat libovolně posunutou sinusovku nebo cosinusovku s libovolnou amplitudou. To samé platí i pro $k > 1$ a $k < -1$ s tím, že k musíme přidat všude, k výrazu $\omega_1 t$. Pro k=0 nám pro $x(t)$ zůstane pouze reálná část koeficientu c_0, udávající stejnosměrnou složku výsledného signálu $x(t)$.

V těchto laboratořích se budeme zabývat FŘ velmi typických signálů: periodických sledů obdélníkových impulsů. Takový signál je nadefinován jako

$$x(t) = \begin{cases} D & \text{pro} \quad -\frac{\vartheta}{2} \le t \le \frac{\vartheta}{2} \\ 0 & \text{pro} \quad -\frac{T_1}{2} \le t < -\frac{\vartheta}{2} \quad \text{a} \quad \frac{\vartheta}{2} < t \le \frac{T_1}{2} \end{cases}$$

s periodou T_1. Z přednášek víme, že jeho koeficienty FŘ se dají spočítat analyticky:

$$c_k = D\frac{\vartheta}{T_1}\text{sinc}\left(\frac{\vartheta}{2}k\omega_1\right). \tag{18}$$

Koeficienty Fourierovy řady budeme počítat třemi způsoby:

1. Budeme pouze zobrazovat koeficienty vypočtené podle teoretického vztahu.
2. Vygenerujeme ručně funkce $e^{-jk\omega_1 t}$ (komplexně sdružené báze) a budeme do nich promítat signál - násobit a integorvat (čti sčítat) přes jednu periodu.
3. Použijeme funkci `fft` a ukážeme si, že sice koeficienty počítá, ale s jistými problémy,

Příklad 7.

Vygenerujte sled obdélníkových impulsů $x(t)$ s parametry: $D = 6$, $T_1 = 1$ μs, $\vartheta = 0.25$ μs. Zobrazte jednu periodu a několik period.

Jako při každé práci se signály se spojitým čase je nutné zvolit krok, nastavíme jej např. tak, abychom na jednu periodu měli 200 bodů:

```
T1 = 1e-6; theta=0.25e-6; D=6;
krok = T1/200;
t = -T1/2:krok:T1/2; x = zeros(size(t));
x (find(t > -theta/2 & t < theta/2)) = D; plot (t,x);
```

Pokud se chceme podívat na několik period, je to možné např. pomocí:

```
x5 = [x x x x x]; t5 = [t-2*T1 t-T1 t t+T1 t+2*T1];
plot (t5,x5);
```

My ale budeme pracovat pouze s jednou periodou.

Příklad 8.

Vypočtěte koecificienty c_k podle vzorečku (18) pro $k \in [-20, 20]$. Pak zobrazte jejich modul a argument na příslušných kruhových frekvencích:

```
omega1 = 2*pi / T1;
k = -20:20;
ck = ...
figure(1);
subplot (211); stem (k*omega1,abs(ck));
subplot (212); stem (k*omega1,angle(ck));
```

!!! Pozor. Pokud budete používat funkci `sinc`, musíte dělit argument hodnotou π, protože Matlab ji definuje jako $\mathrm{sinc}(x) = \frac{\sin \pi x}{\pi x}$, my definujeme: $\mathrm{sinc}(x) = \frac{\sin x}{x}$. !!!

Úkoly:

1. Doplňte výpočet koeficientů a zobrazte je.
2. Zkuste do obrázku modulů "dostat" také průběh pomocné funkce $\left|\frac{\vartheta}{T_1}\mathrm{sinc}\left(\frac{\vartheta}{2}\omega\right)\right|$ - můžete využít např. funkcí `hold on` a `hold off`.
3. Zkontrolujte, zda první dotek pomocné funkce s nulou leží skutečně na kruhové frekvenci $\omega_a = \frac{2\pi}{\vartheta}$.

Příklad 9.

Vypočtěte koeficienty c_k "ručně" pomocí generování komplexních exponenciál $e^{-jk\omega_1 t}$. Pokuste se použít skalární součin spíše než cyklus. Pokud dostanete exponenciálu na frekvenci $k\omega_1$ třeba do vektoru `ee`, mohl by integrál z rovnice (14) vypadat třeba takhle:

```
c = 1 / T1 * x * ee.' * krok;
```

Úkoly:

1. Všechny takto vypočtené koeficienty pro $k \in [-20, 20]$ uložte do vektoru, např. `ckr` (jako 'ručně'). Pak zobrazte a srovnejte `ck` a `ckr`.

Úkoly pro hloubavé:

1. Boxík v Matlabu implementuje výpočet každého koeficientu jako skalární součin. `x` i `ee` jsou řádkové vektory, takže aby to fungovalo, je potřeba udělat z `ee` vektor sloupcový. Proč je transposice zapsána jako `ee.'` a ne jen jako `ee'` ? Zkuste prostudovat `help punct`.
2. Co by muselo být v `ee`, aby stačilo použít `ee'`?

Příklad 10.

Zkuste vypočítat koeficienty FŘ pomocí rychlé Fourierovy transformace (fast Fourier transform – FFT). O FFT se dozvíme později, teď jen pár řádek z Matlabu:

```
X = fft(x); N = length(X);
ckf = X / N;
omegaf = (0:(N-1)) * omega1;
figure(3)
subplot (211); stem (omegaf,abs(ckf));
subplot (212); stem (omegaf,unwrap(angle(ckf))); grid;
```

Úkoly:

1. Moduly vypadají dobře, ale ve střední části (zoom) zjistíte, že se každý čtvrtý koeficient už nerovná nule. Pokuste se vysvětlit, co se stalo.
2. Co se stalo s fází ?
3. Proč je v zobrazení fáze použita funkce `unwrap` a co se zobrazí bez ní ?

Příklad 11.

Ukažte, jak se obdélníkový signál skládá z jednotlivých komponentů $c_k e^{jk\omega_1 t}$. Začněte stejnosměrnou složkou c_0 a pak postupně přidávejte pro $k = \pm 1, \quad k = \pm 2, \ldots$. Pro kažý krok vykreslete komplexní exponenciály, jejich součet a sumu všeho. Výsledkem by mělo být něco podobného slajdům 37 a 38 v přednášce o FŘ.

Příklad 12. ...analýza slunečních erupcií (nepovinné)

```
% nacteni a zobrazeni dat
load sunspot.dat
year=sunspot(:,1);
relNums=sunspot(:,2);
plot(year,relNums)

% zazoomovani na prvnich 50 let
plot(year(1:50),relNums(1:50),'b.-');

% Fourierova transformace
Y = fft(relNums);
Y(1)=[]; % prvni prvek je pouze suma vektoru relNums, odstranime ho
plot(abs(Y));

% zobrazeni vykonu na frekvencich
n=length(Y);
power = abs(Y(1:floor(n/2))).^2; % staci nam pouze polovina hodnot Y
nyquist = 1/2;
freq = (1:n/2)/(n/2)*nyquist;
plot(freq,power)
xlabel('cycles/year')
title('Periodogram')

% x-ova osa v rokoch
period=1./freq;
plot(period,power);
axis([0 40 0 2e+7]);
ylabel('Power');
xlabel('Period (Years/Cycle)');

% najdenie frekvencie s maximalnym vykonom
hold on;
index=find(power==max(power));
mainPeriodStr=num2str(period(index));
plot(period(index),power(index),'r.', 'MarkerSize',25);
text(period(index)+2,power(index),['Period = ',mainPeriodStr]);
hold off;
```

Zdroj: http://www.mathworks.com/products/matlab/demos.html?\file=/products/demos/shipping/matlab/sunspots.html

Příklad 13. ...analýza zašuměného signálu (nepovinné)

```
t = 0:.001:.25;
x = sin(2*pi*50*t) + sin(2*pi*120*t);
plot(t,x);

y = x + 2*randn(size(t)); % zasumeny signal
plot(y(1:50))

% Fourierova transformace
Y = fft(y,256);

Pyy = Y.*conj(Y)/256;
f = 1000/256*(0:127); % staci nam prvnich 127 hodnot, zbytek je symetricky
plot(f,Pyy(1:128))
title('Power spectral density')
xlabel('Frequency (Hz)')

f(find(Pyy(1:128)>30))   % vypis frekvence s vykonem nad 30
```

Zdroj: http://www.mathworks.com/products/matlab/demos.html?\file=/products/demos/shipping/matlab/fftdemo.html

Řešení příkladů

Př.2. Úk.1. Ověření ortonormality:

```
b1'*b2
%... kdyz je vysledek = 0, jsou kolme
sqrt(sum(b1.^2))
sqrt(sum(b2.^2))
%... kdyz je vysledek = 1, jsou jednotkove
```

Př.3. Úk.1. Stejně jako v Př.2. Úk.1.

Př.4. Úk.2. Výpočet absolutní hodnoty funkce:

```
sqrt(sum(b.^2)*krok)
```

Př.8. Úk.1. Výpočet c_k:

```
ck = D * theta / T1 * sinc (theta /2 * k * omega1 / pi); % /pi kvuli matlabu
```

Př.8. Úk.2. Vykreslení pomocné funkce:

```
omega = min(k*omega1):omega1/100:max(k*omega1);
pom = abs( D * theta / T1 * sinc (theta /2 * omega / pi)); % /pi kvuli matlabu
subplot (211); hold on; plot (omega,pom,'r--'); hold off;
```

Př.8. Úk.3. Kontrola, zda dotek funkce lezi v $\frac{2\pi}{\theta} = 8 \times 10^6\pi = 25.133 \times 10^6$ - jo. Pomocí lupy se dá přiblížit místo prvního doteku pomocné funkce s x-ovou osou.

Př.9. "Ruční" výpočet koeficientů - generováním komplexních exponenciál:

```
ckr = zeros(size(k)); gj0 = find (k==0);
for ii = k,
  ee = exp(-j * ii * omega1 * t); plot3(t,real(ee),imag(ee)); %pause
  % do integralu je ovsem potreba dostat dt - to je krok
  ckr(gj0 + ii) = 1 / T1 * x * ee' * krok;
end
figure(2)
subplot (211); stem (k*omega1,abs(ckr));
subplot (212); stem (k*omega1,angle(ckr));
```

Př.9. Úk.1. Srovnání `ck` a `ckr`:

```
ck - ckr   % rozdíl by měl být malý
```

Př.9. Úk.pro hloubavé 1. Znak `'` za vektorem tento vektor ztransponuje a zároveň komplexní čísla ve vektoru převede na komplexně sdružená (viz. `help punct`). Operátor .' pouze transponuje.

Př.9. Úk.pro hloubavé 2. V ee by musela být přímo báze, ne její komplexní sdružení. Takže `ee = exp(j * ii * omega1 * t)`

Př.10. Úk.1. Dochází k aliasingu a do spektra proniká jeho kopie ze vzorkovací frekvence. Více v přednáškách o vzorkování a DFT.

Př.10. Úk.2. S fází se dějí tyto věci:

1. Protože je signál oproti originálu posunutý o půl periody, k argumentu každého koeficientu se tedy přičítá $-k\omega_1\tau$, což je $-k\pi$.
2. Matlab nerespektuje naše poučky o kráse a používá π nebo $-\pi$, jak se mu zachce.
3. Pro nulové koeficienty $|c_k| = 0$ nedává fázi nula.

Př.10. Úk.3. Funkce `unwrap` mění skoky větší nebo rovny π na jejich 2π násobek.

Př.11. Skládání obdélníkového signálu z komplexních exponenciál:

```
% zacneme pro ss. slozku
gj0 = find (k==0);
yy = ck(gj0) * ones(size(t));
figure(4);
plot (t,yy);
% ted pro normalni k-cka:
for ii=1:20,
  % udelame a nakreslime  exponencialy
  e = ck(gj0+ii) * exp(j*ii*omega1*t);
  me = ck(gj0-ii) * exp(-j*ii*omega1*t);
  subplot(311); plot3(t,real(e),imag(e),'r-',t,real(me),imag(me),'b--');
  % secteme je a bude cosinusovka
  thiscos = e + me;
  subplot(312); plot(t,thiscos);
  % a pricteme k signalu, ktery skladame
  yy = yy+thiscos;
  subplot(313); plot(t,yy); pause
end
```