

Seminář 6

Fourierova řada

Součet funkcí

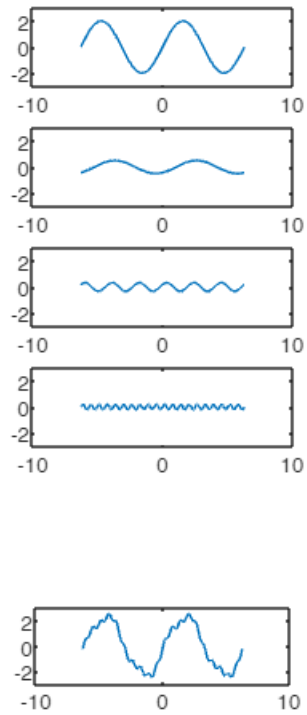
```
n = 300; % pocet bodu

% vytvoření vektoru o n hodnotách pravidelně rozmístěné mezi -2pi a 2pi
x = linspace(-2*pi,2*pi,n)';

% jednotlivé funkce
y1s = 2*sin(x);
y2s = sin(x-pi/3)/2;
y3s = sin(3*x+pi/5)/3;
y4s = sin(10*x)/5;

y1c = 2*cos(x);
y2c = cos(x-pi/3)/2;
y3c = cos(3*x+pi/5)/3;
y4c = cos(10*x)/5;

figure,
subplot(6,1,1), plot(x,y1s);
axis equal
ylim([-3,3]);
xlim([-10,10]);
subplot(6,1,2), plot(x,y2s);
axis equal
ylim([-3,3]);
xlim([-10,10]);
subplot(6,1,3), plot(x,y3s);
axis equal
ylim([-3,3]);
xlim([-10,10]);
subplot(6,1,4), plot(x,y4s);
axis equal
ylim([-3,3]);
xlim([-10,10]);
subplot(6,1,6), plot(x,y1s+y2s+y3s+y4s);
axis equal
ylim([-3,3]);
xlim([-10,10]);
```



Animace

jednotlivých funkcí

```
nt = 50; %pocet bodu na kruznici
theta = linspace(-pi,pi,nt)'; % diskretni body na kruznici

% Kruznice
kx1 = 2*cos(theta);
ky1 = 2*sin(theta);

kx2 = cos(theta-pi/3)/2;
ky2 = sin(theta-pi/3)/2;

kx3 = cos(3*theta+pi/5)/3;
ky3 = sin(3*theta+pi/5)/3;

kx4 = cos(10*theta)/5;
ky4 = sin(10*theta)/5;

Lx=length(x);
Lw=1; %sirka cary
Fs=12;
```

```

for i=1:Lx

    f1=figure (2); clf;

    sp1=subplot(1,2,1);
    % Funkce 1
    plot(kx1,ky1,'LineWidth',Lw,'Color','b'); hold on; grid on;
    line([0 y1c(i)],[0 y1s(i)],'Color','b','LineWidth',Lw,'LineSmoothing','on');

    set(sp1,'Position',[0.0400    0.1800    0.4    0.677]);
    xlim([-2.5 2.5]); ylim([-2.5 2.5])

    line(y1c(i),y1s(i),10,'LineStyle','-','MarkerSize',8,'MarkerFaceColor','b','color','b')

    [xf1, yf1] = ds2nfu(sp1,y1c(i),y1s(i));

    % Funkce 2
    plot(kx2,ky2,'LineWidth',Lw,'Color','r'); hold on;
    line([0 y2c(i)],[0 y2s(i)],'Color','r','LineWidth',Lw,'LineSmoothing','on');

    line(y2c(i),y2s(i),10,'LineStyle','-','MarkerSize',8,'MarkerFaceColor','r','color','r')

    [xf2, yf2] = ds2nfu(sp1,y2c(i),y2s(i));

    % Funkce 3
    plot(kx3,ky3,'LineWidth',Lw,'Color','g'); hold on;
    line([0 y3c(i)],[0 y3s(i)],'Color','g','LineWidth',Lw,'LineSmoothing','on');

    line(y3c(i),y3s(i),10,'LineStyle','-','MarkerSize',8,'MarkerFaceColor','g','color','g')

    [xf3, yf3] = ds2nfu(sp1,y3c(i),y3s(i));

    % Funkce 4
    plot(kx4,ky4,'LineWidth',Lw,'Color','m'); hold on;
    line([0 y4c(i)],[0 y4s(i)],'Color','m','LineWidth',Lw,'LineSmoothing','on');

    line(y4c(i),y4s(i),10,'LineStyle','-','MarkerSize',8,'MarkerFaceColor','m','color','m')

    [xf4, yf4] = ds2nfu(sp1,y4c(i),y4s(i));

    sp2=subplot(1,2,2);
    % Funkce 1

    plot(x(1:i),y1s(1:i),'LineWidth',Lw,'Color','b'); hold on; grid on;

```

```

ylim([-2.5 2.5]); xlim([-10 10])
set(sp2,'Position',[0.48    0.178200    0.49    0.680]);

[xg1, yg1] = ds2nfu(x(i),y1s(i));
annotation('line',[xf1 xg1],[yf1 yg1],'color','b','LineStyle','--','LineWidth',Lw);

% Funkce 2
plot(sp2,x(1:i),y2s(1:i),'g','LineWidth',Lw,'Color','r'); hold on; grid on;

[xg2, yg2] = ds2nfu(x(i),y2s(i));
annotation('line',[xf2 xg2],[yf2 yg2],'color','r','LineStyle','--','LineWidth',Lw);

% Funkce 3
plot(sp2,x(1:i),y3s(1:i),'g','LineWidth',Lw,'Color','g'); hold on; grid on;

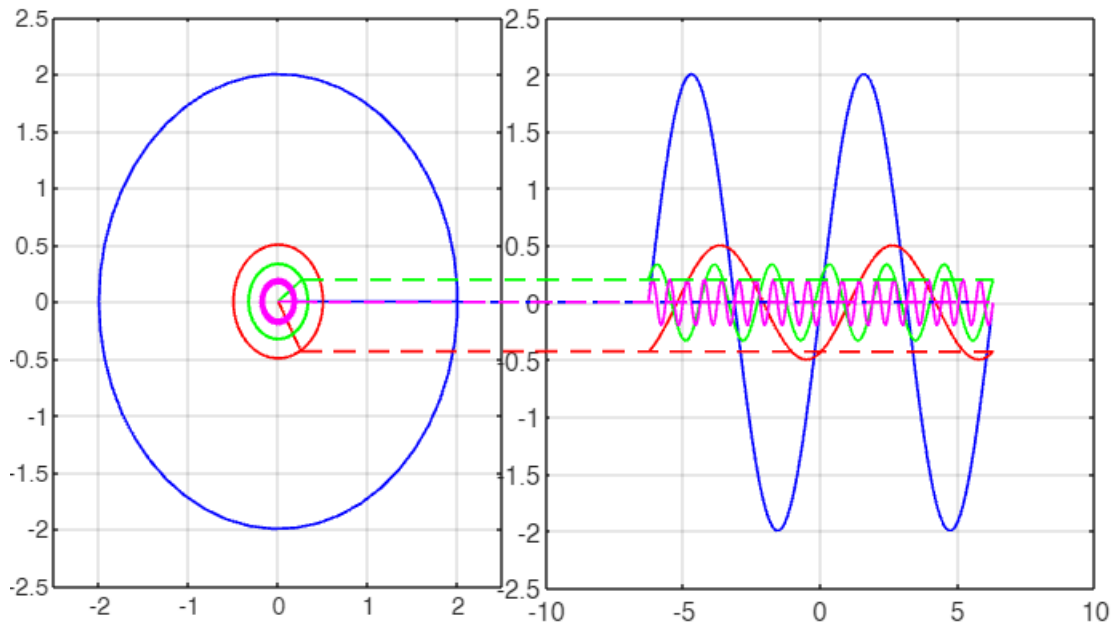
[xg3, yg3] = ds2nfu(x(i),y3s(i));
annotation('line',[xf3 xg3],[yf3 yg3],'color','g','LineStyle','--','LineWidth',Lw);

% Funkce 4
plot(sp2,x(1:i),y4s(1:i),'g','LineWidth',Lw,'Color','m'); hold on; grid on;

[xg4, yg4] = ds2nfu(x(i),y4s(i));
annotation('line',[xf4 xg4],[yf4 yg4],'color','m','LineStyle','--','LineWidth',Lw);

pause(0.01);
end

```



Animace dohromady

Funkce jsou počítány

```
for i=1:length(x)
    f1=figure (2); clf;

    sp1=subplot(1,2,1);
    % Funkce 1
    plot(kx1,ky1,'LineWidth',Lw,'Color','b'); hold on; grid on;
    line([0 y1c(i)],[0 y1s(i)],'Color','b','LineWidth',Lw,'LineSmoothing','on');

    set(sp1,'Position',[0.0400    0.1800    0.4    0.677]);
    xlim([-4 4]); ylim([-4 4])

    line(y1c(i),y1s(i),10,'LineStyle','-','MarkerSize',8,'MarkerFaceColor','b','color','b')

    % Funkce 2
    plot(kx2+y1c(i),ky2+y1s(i),'LineWidth',Lw,'Color','r'); hold on;
    line(y1c(i)+[0 y2c(i)],y1s(i)+[0
y2s(i)],'Color','r','LineWidth',Lw,'LineSmoothing','on');

    % Funkce 3
    plot(kx3+y1c(i)+y2c(i),ky3+y1s(i)+y2s(i),'LineWidth',Lw,'Color','g'); hold on;
    line(y1c(i)+y2c(i)+[0 y3c(i)],y1s(i)+y2s(i)+[0 y3s(i)],'Color','g','LineWidth',Lw,
    ...
        'LineSmoothing','on');

    % Funkce 4
    plot(kx4+y1c(i)+y2c(i)+y3c(i),ky4+y1s(i)+y2s(i)
+y3s(i),'LineWidth',Lw,'Color','m');
    hold on;
    line(y1c(i)+y2c(i)+y3c(i)+[0 y4c(i)],y1s(i)+y2s(i)+y3s(i)+[0
y4s(i)],'Color','m',...
        'LineWidth',Lw,'LineSmoothing','on');

    [xf4, yf4] = ds2nfu(y1c(i)+y2c(i)+y3c(i)+y4c(i),y1s(i)+y2s(i)+y3s(i)+y4s(i));

    sp2=subplot(1,2,2);
    % dohromady

    plot(x(1:i),y1s(1:i)+y2s(1:i)+y3s(1:i)+y4s(1:i),'LineWidth',Lw,'Color','m');
    hold on; grid on;

    ylim([-4 4]); xlim([-10 10])
    set(sp2,'Position',[0.48    0.178200    0.49    0.680]);

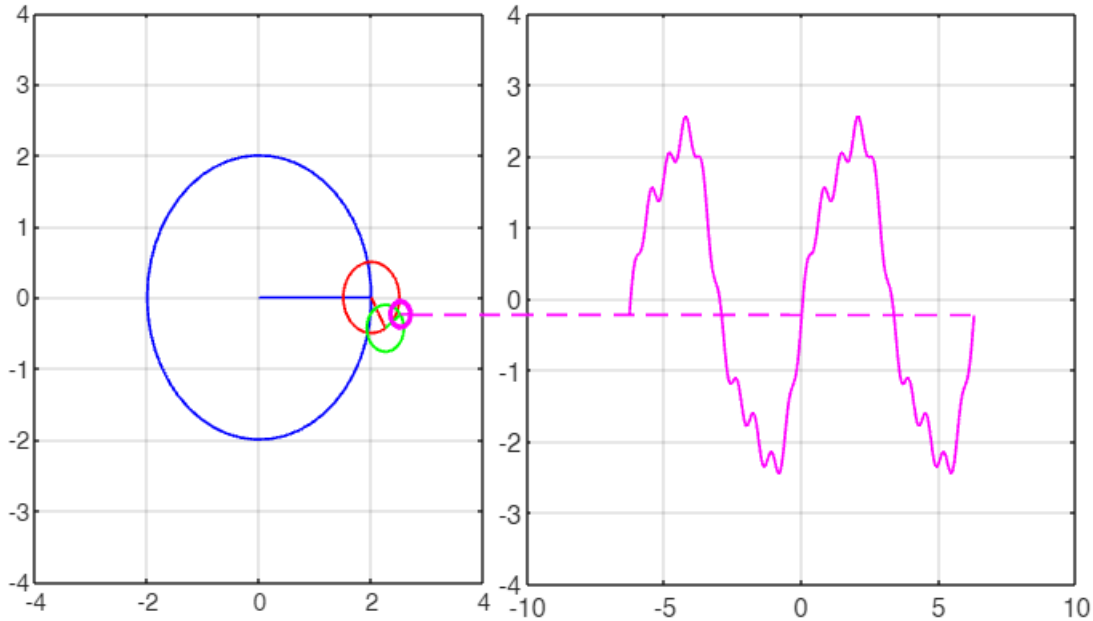
    [xgt, ygt] = ds2nfu(x(i),y1s(i)+y2s(i)+y3s(i)+y4s(i));
```

```

annotation('line',[xf4 xgt],[yf4 ygt],'color','m','LineStyle','--','LineWidth',Lw);

pause(0.01);
end

```



Vytvoření fourierovy řady

Funkce: Fseries.m and Fseriesval.m

$$y = a_0 / 2 + \text{Sum}_k [a_k \cos(kx) + b_k \sin(kx)]$$

Příklad 1

```

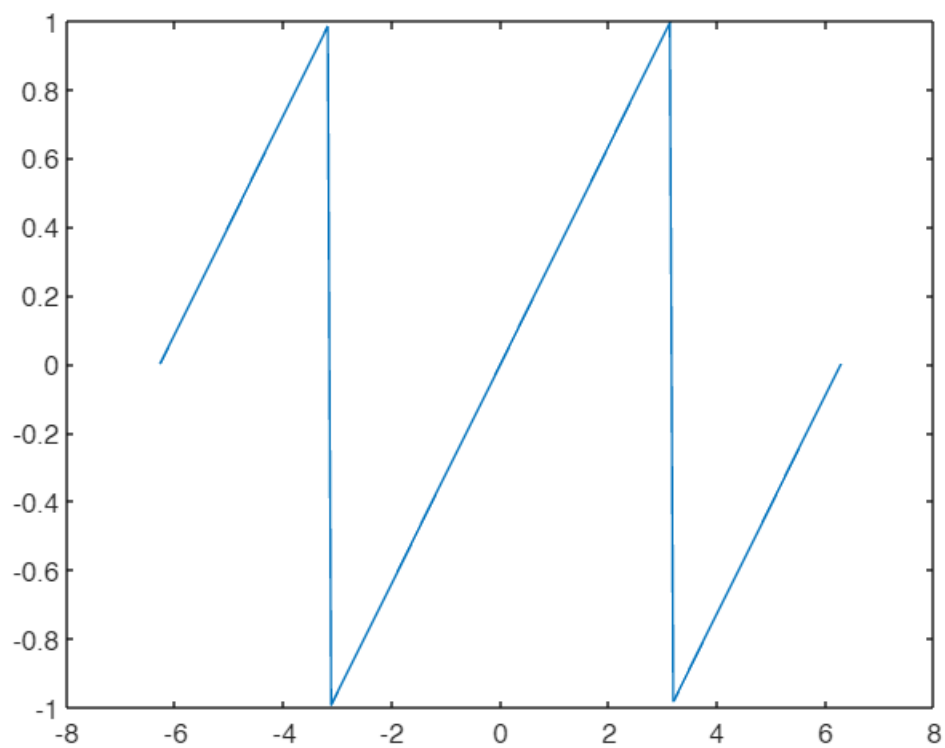
% Generovani dat

% vytvoreni vektoru o 50 hodnotach pravidelne rozmistene mezi -2pi a 2pi
x = linspace(-2*pi,2*pi,200)';
% vypocet funkcnich hodnot pomoci funkce sawtooth
y = sawtooth(x + pi);

% alternativni vypocet bez pouziti funkce sawtooth
%y = mod(x + pi,2*pi) - pi;

figure, plot(x,y);

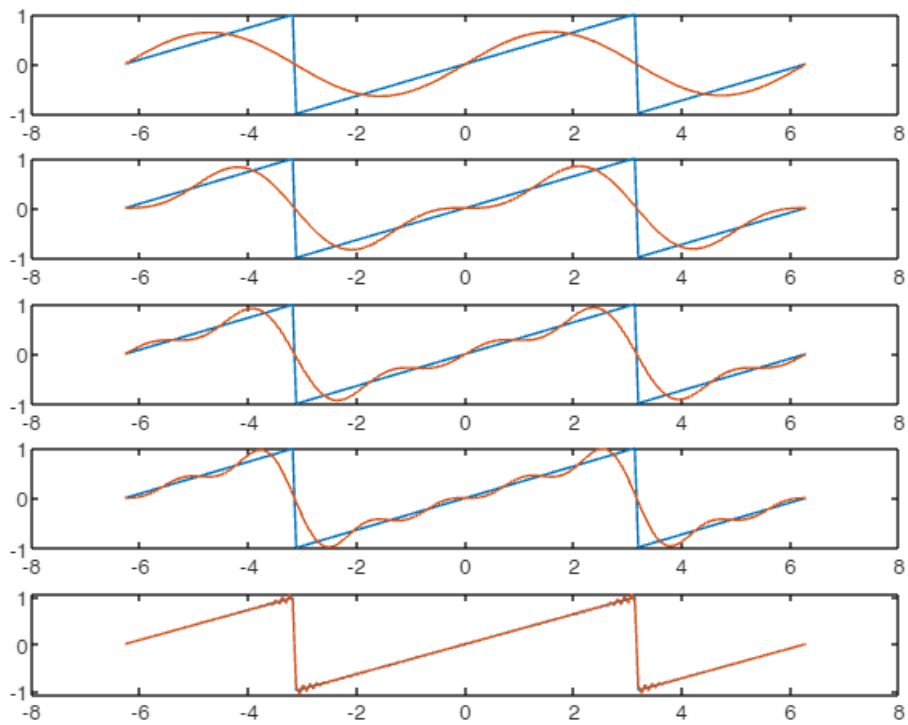
```



Rozvoj pomocí sinu a cosinu

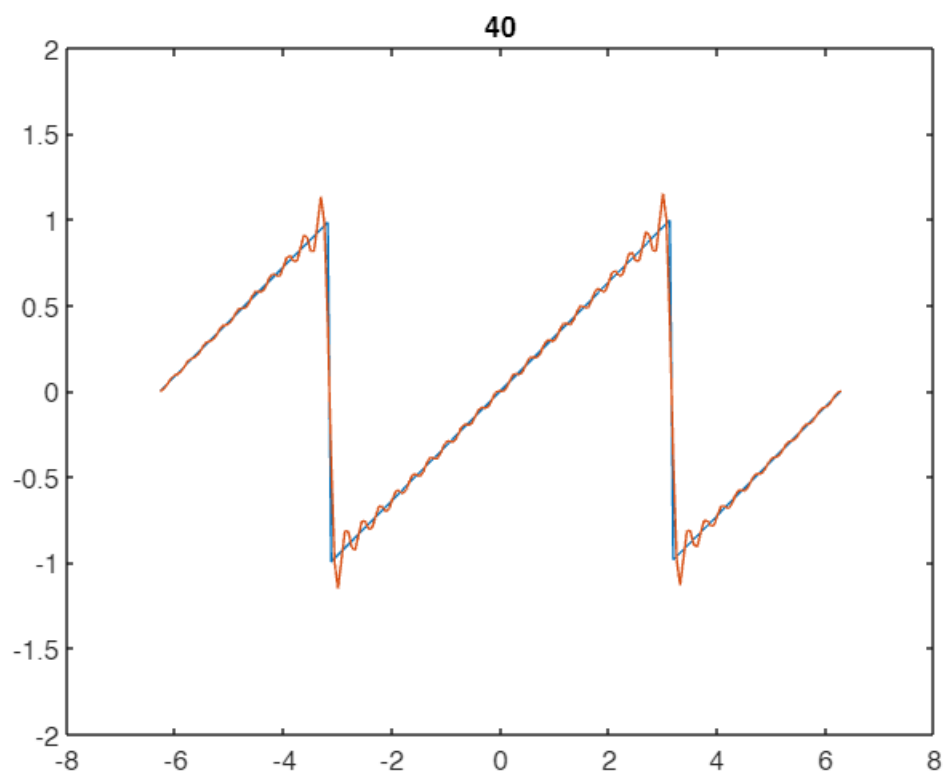
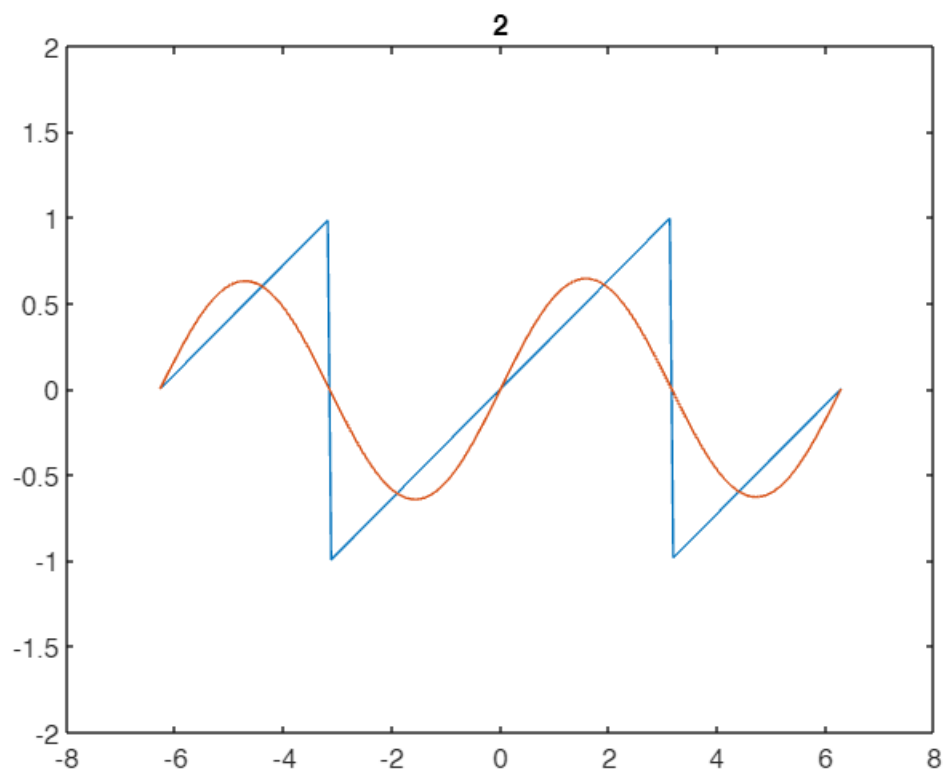
```
% rady
n = [2, 4, 6, 8, 90];

figure,
% pro vsechny rady spocitame rozvoj pomoci Fseries
for i = 1 : size(n,2)
    [~,b,yfit] = Fseries(x,y,n(i));
    subplot(size(n,2),1,i)
    plot(x,y,x,yfit);
end
```



Animace

```
% pro vsechny rady spocitame rozvoj pomoci Fseries
figure,
for i = 2 : 2 : 40
    [~,b,yfit] = Fseries(x,y,i);
    pause(0.3);
    plot(x,y,x,yfit);
    title(num2str(i));
    ylim([-2,2]);
end
```

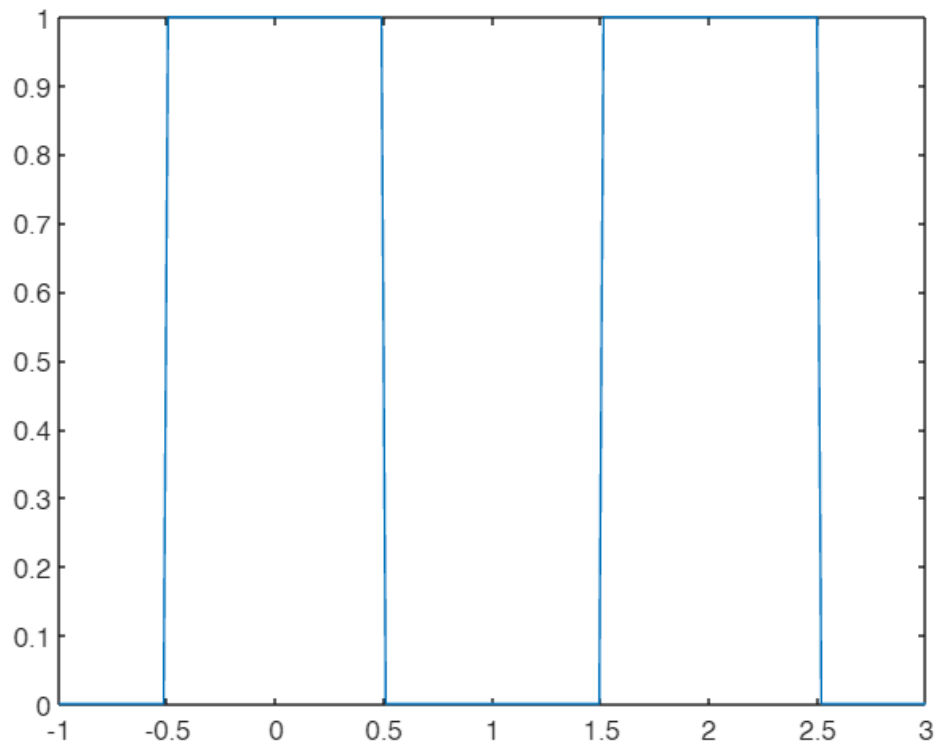
Příklad 2

% Generovani dat

```

x = linspace(-1,3,200)';
y = double(or(and(x < 2.5, x >= 1.5),and(x< 0.5, x >= -0.5)));% vypocet funkcnich
hodnot
figure, plot(x,y);

```



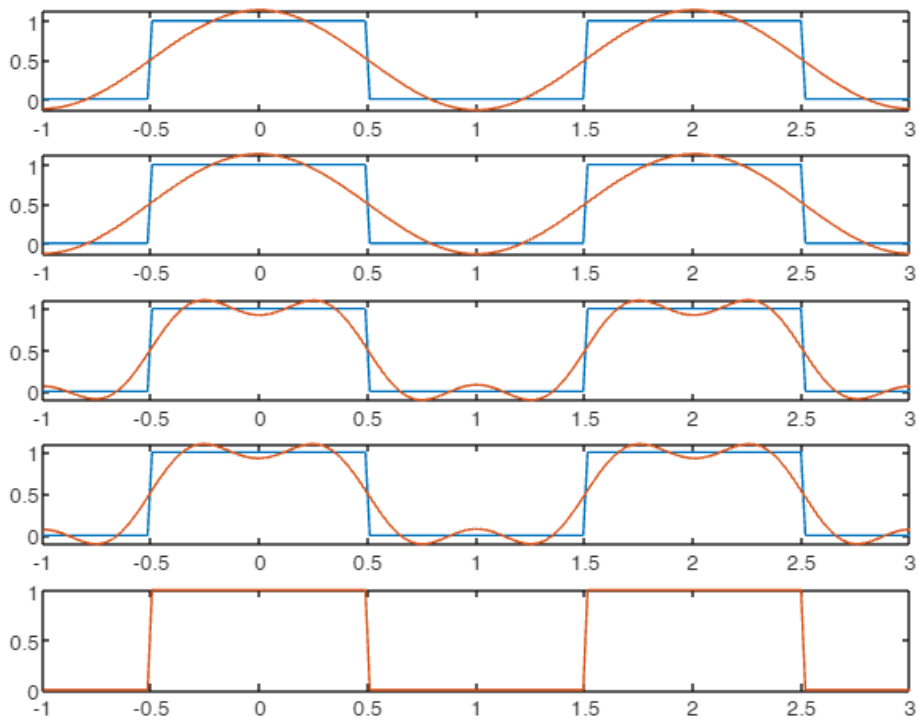
Rozvoj pomoci sinu a cosinu

```

% rady
n = [2, 4, 6, 8, 99];

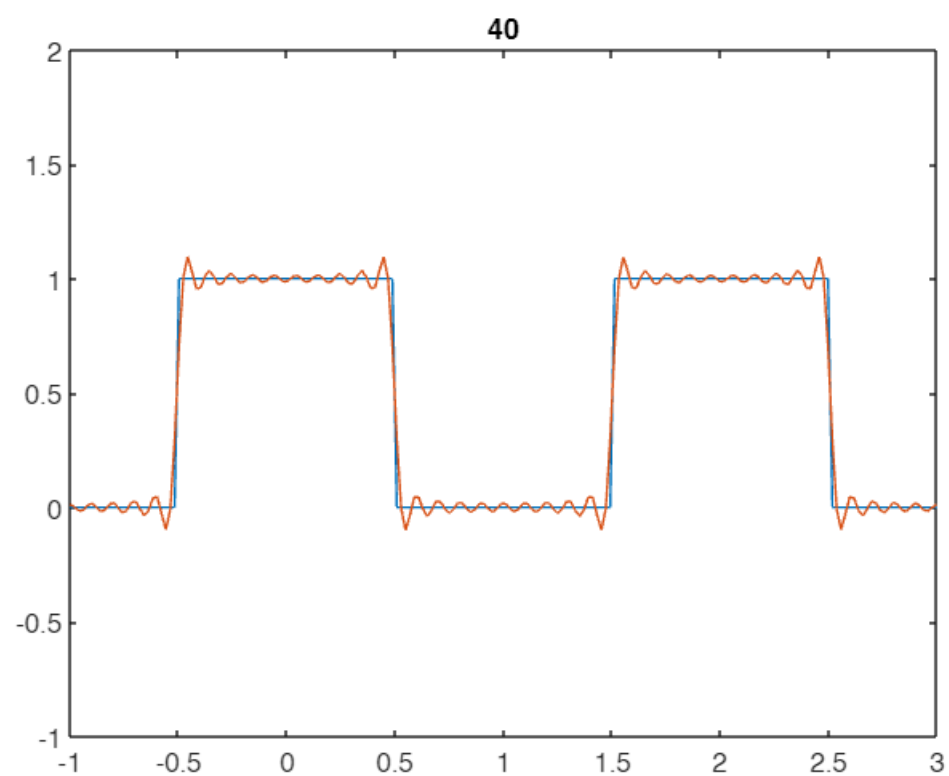
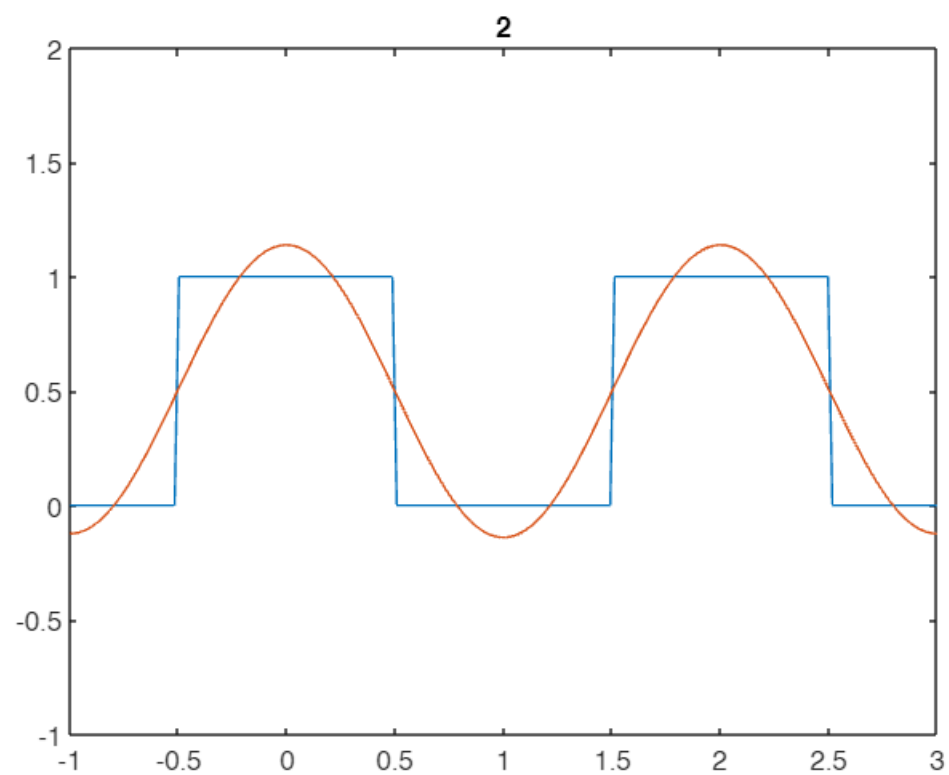
figure,
% pro vsechny rady spocitame rozvoj pomoci Fseries
for i = 1 : size(n,2)
    [~,b,yfit] = Fseries(x,y,n(i));
    subplot(size(n,2),1,i)
    plot(x,y,x,yfit);
end

```



Animace

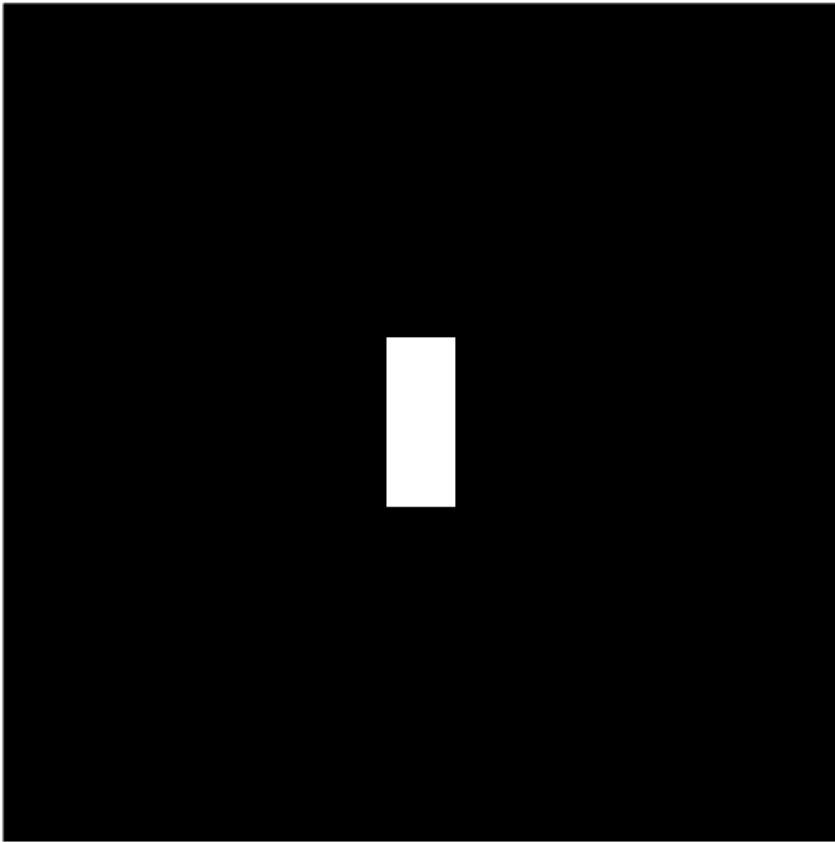
```
figure,
for i = 2 : 2 : 40
    [~,b,yfit] = Fseries(x,y,i);
    pause(0.3);
    plot(x,y,x,yfit);
    title(num2str(i));
    ylim([-1,2]);
end
```



Frekvenční doména

Vytvoření jednoduchého obrázku

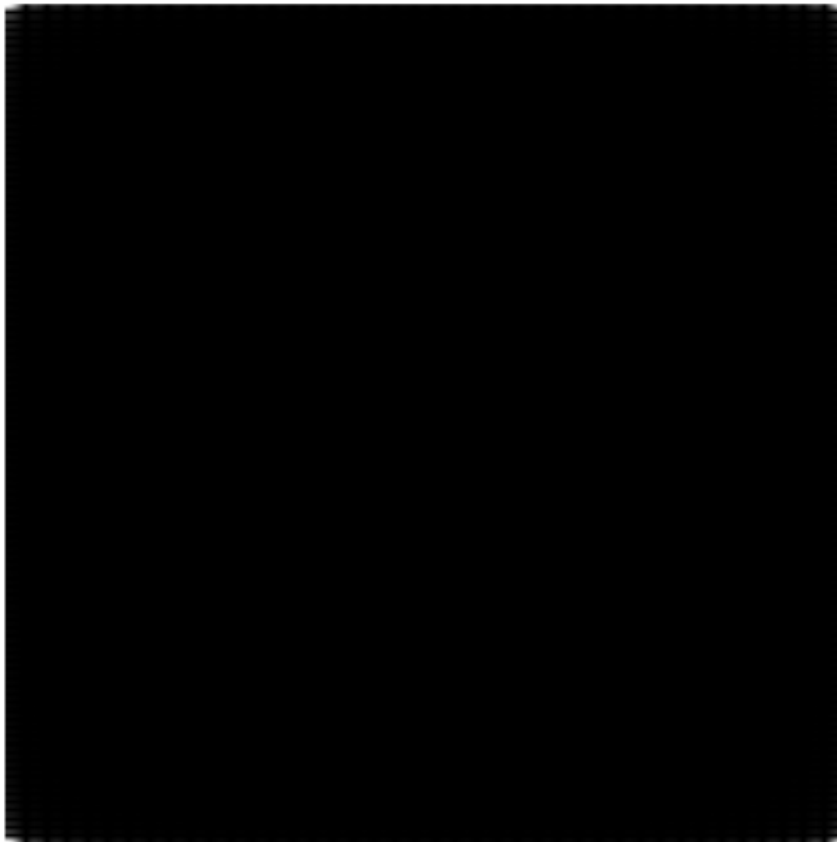
```
f = zeros(500);  
f(200:300,230:270)=1;  
figure, imshow(f);
```



Fourierovo spektrum

Převod do frekvencni domeny - funkce `fft2()` (Fast fourier transform)

```
F = fft2(f);  
figure('name','Frekvencni domena'), imshow(abs(F),[]);
```



DC koeficient

```
dc_koef = F(1,1);  
display(dc_koef);
```

```
dc_koef = 4141
```

```
[m,n]=size(F);  
  
% DC koef. vydeleny poctem pixelu  
dc_koef_norm = F(1,1)/(m*n);  
display(dc_koef_norm);
```

```
dc_koef_norm = 0.0166
```

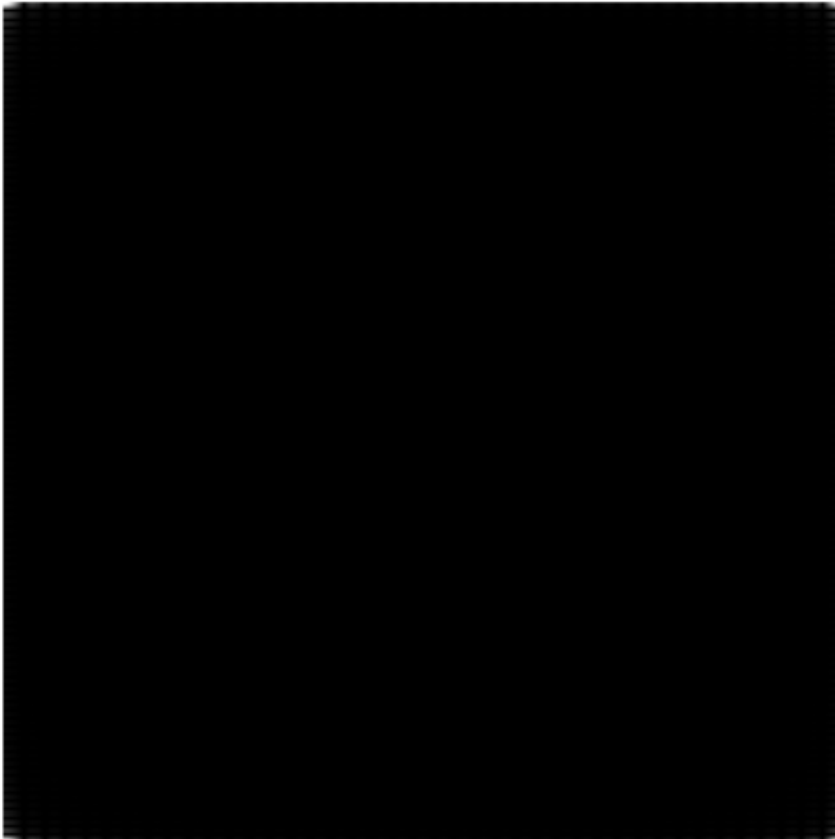
```
% Prumerna intenzita v obrazku  
prum_int = sum(sum(f))/(m*n);  
display(prum_int);
```

```
prum_int = 0.0166
```

Spektrum

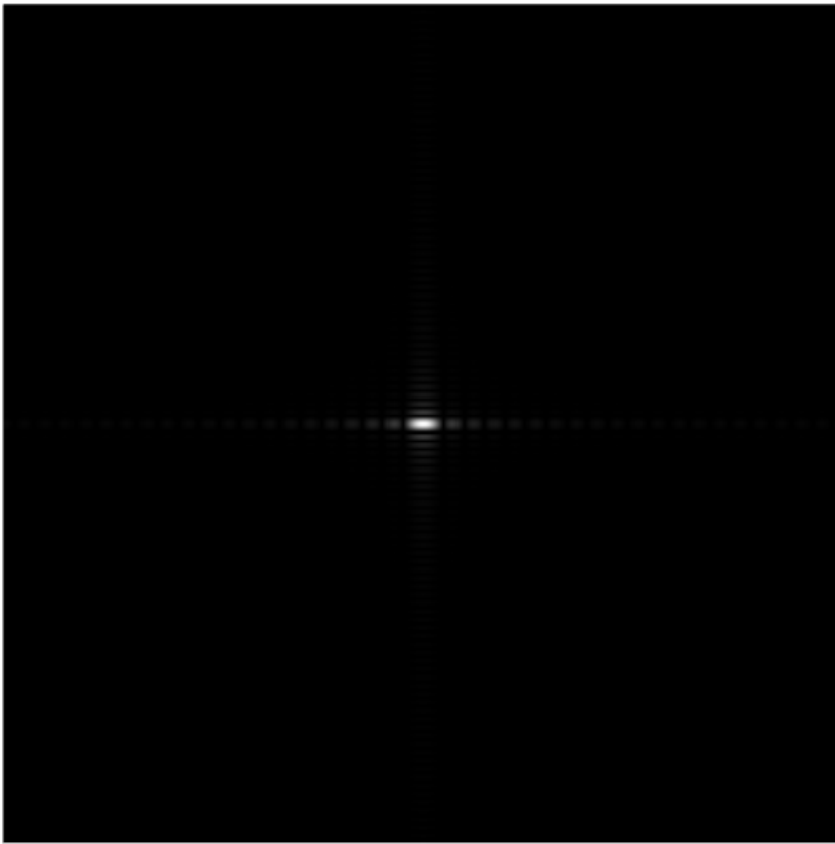
Můžeme získat použitím absolutní hodnoty.

```
S = abs(F);  
figure('name', 'Foufierofo spektrum'), imshow(S,[]);
```



Centorvané fourierovo spektrum

```
Sc = fftshift(S);  
figure('name', 'Centrovane foufierofo spektrum'), imshow(Sc,[]);
```



Funkce `fftshift()`

Provede se posun o půl periody doprava a dolů (tedy jako by se přehodil 1. a 4. kvadrant a 2. a 3).

```
a = [1 2;
      3 4];
fftshift(a)
```

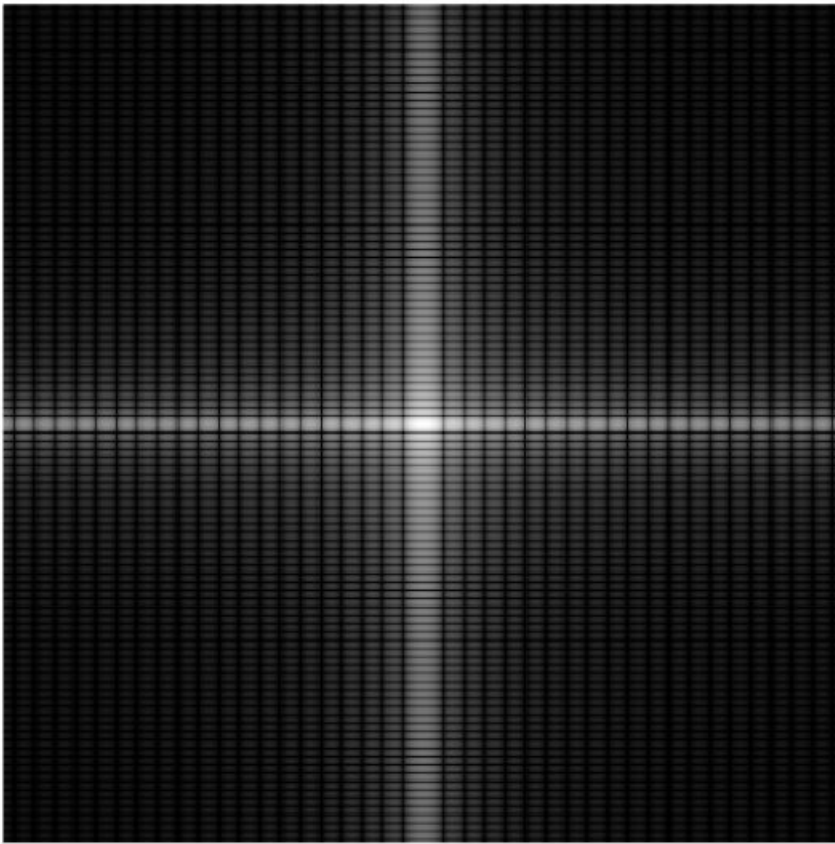
```
ans = 2×2
      4      3
      2      1
```

Fourierovo spektrum + logaritmická transformace

```
maximalni = max(max(Sc));
display(maximalni);
```

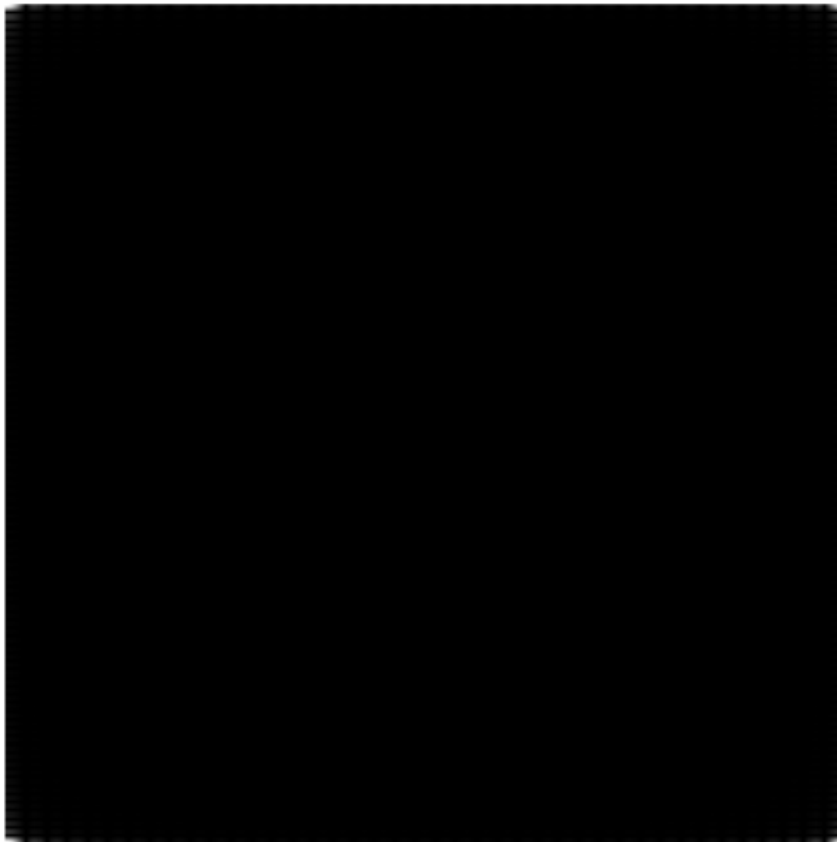
```
maximalni = 4141
```

```
Sc1 = log(1+Sc);
figure('name', 'Centrovane foufierovo spektrum - logaritmicka transformace'),
imshow(Sc1,[]);
```

Necentrovane fourierovo spektrum

```
S1 = ifftshift(Sc);  
figure('name', 'Foufieroovo spektrum'), imshow(S1,[]);
```



Funkce `ifftshift()`

```
M = [1 2 3;  
     4 5 6;  
     7 8 9];  
M2 = fftshift(M);  
  
fftshift(M2)
```

```
ans = 3x3  
     5     6     4  
     8     9     7  
     2     3     1
```

```
ifftshift(M2)
```

```
ans = 3x3  
     1     2     3  
     4     5     6  
     7     8     9
```

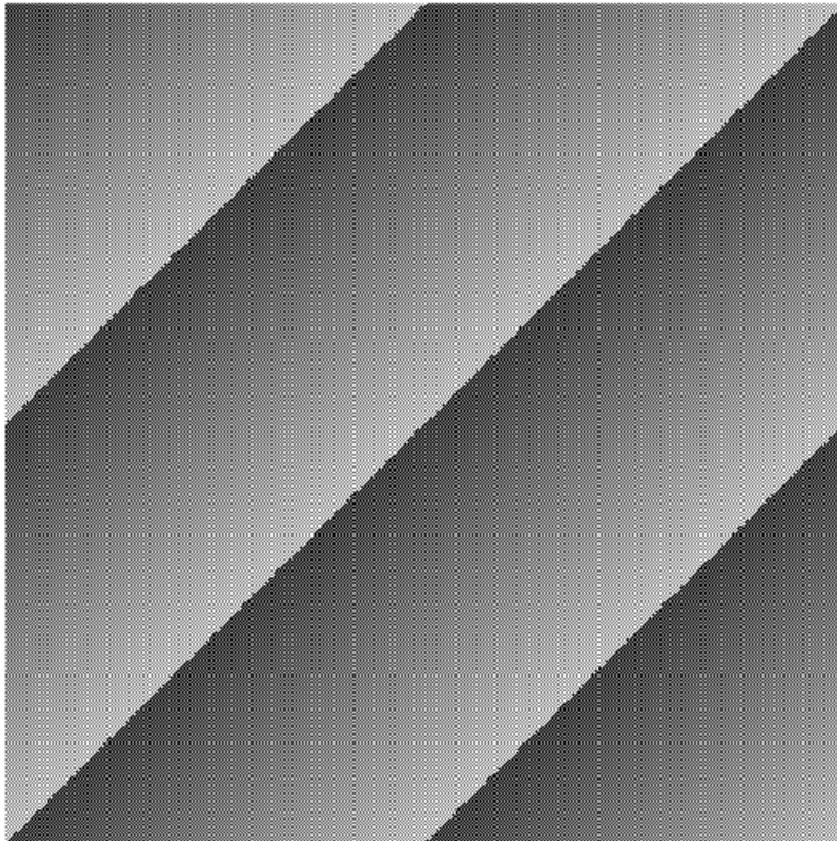
Fáze

```

R = real(F);
I = imag(F);
phi = atan2(I,R);
%aphi = angle(F);

figure, imshow(phi,[]);

```



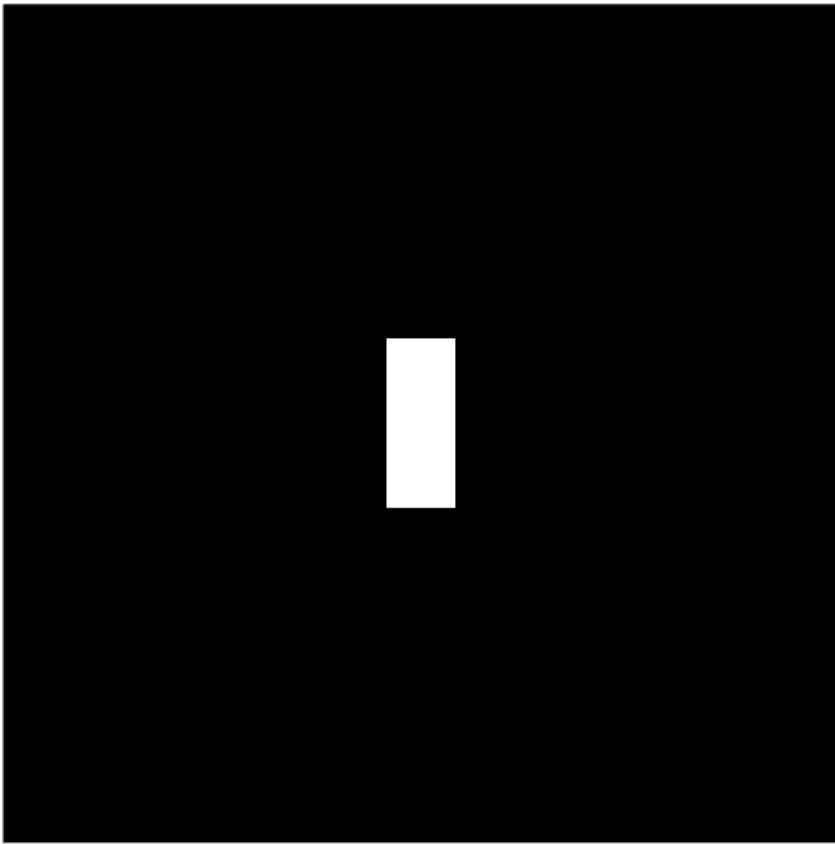
Převod do prostorové domény

```

f = ifft2(F);
f = real(f);

figure('name', 'Prostorova domena'), imshow(f,[]);

```

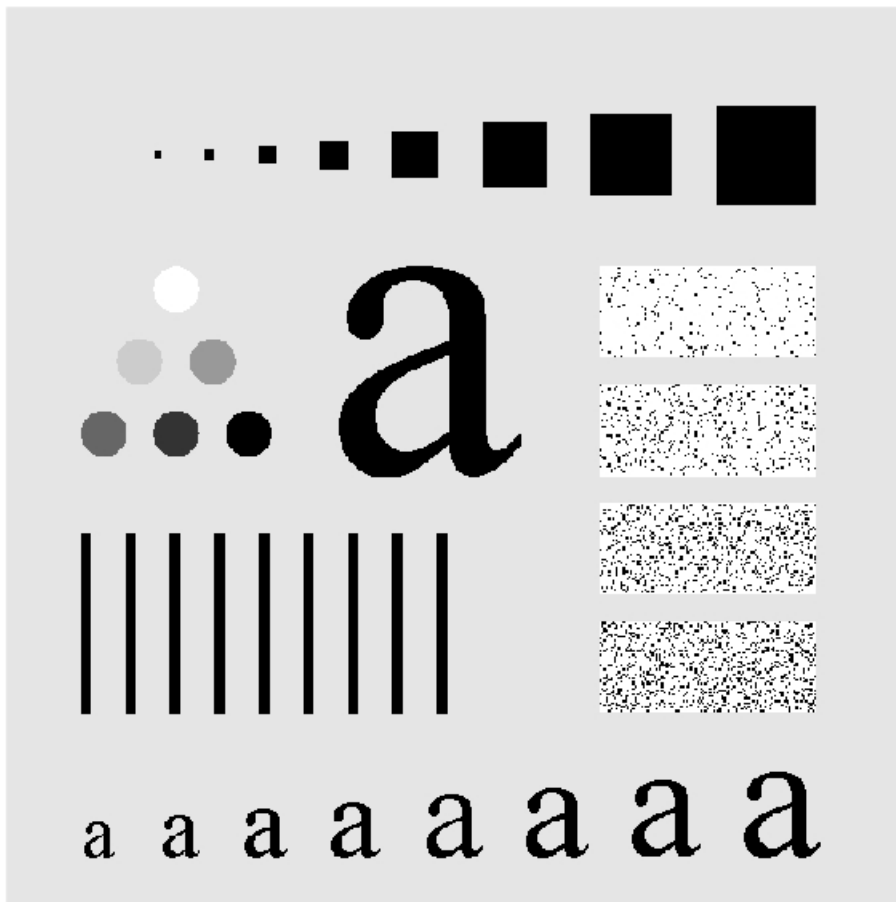


Úkol 1

Porovnejte fázi a spektrum dvou obrázků, které obsahují stejný bílý obdélník, jen posunutý.

Filtrování ve frekvenční doméně

```
f=rgb2gray(imread('a.png'));  
f = double(f);  
figure, imshow(f,[]);
```

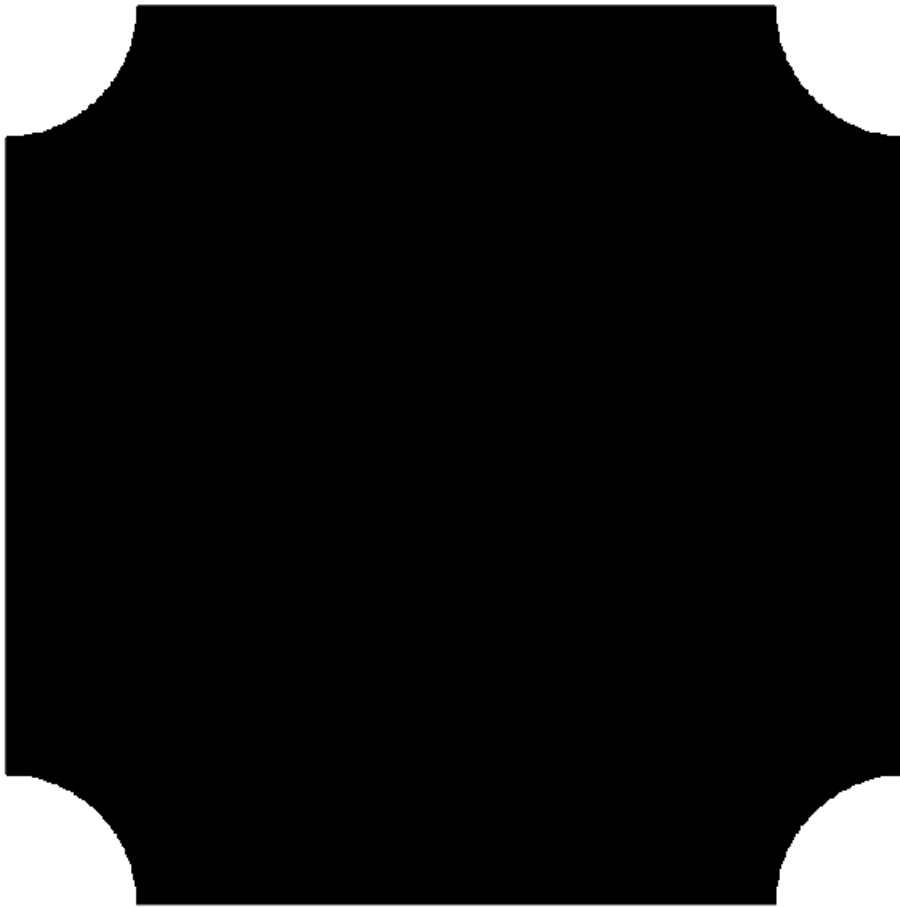


Idealní LP filtr

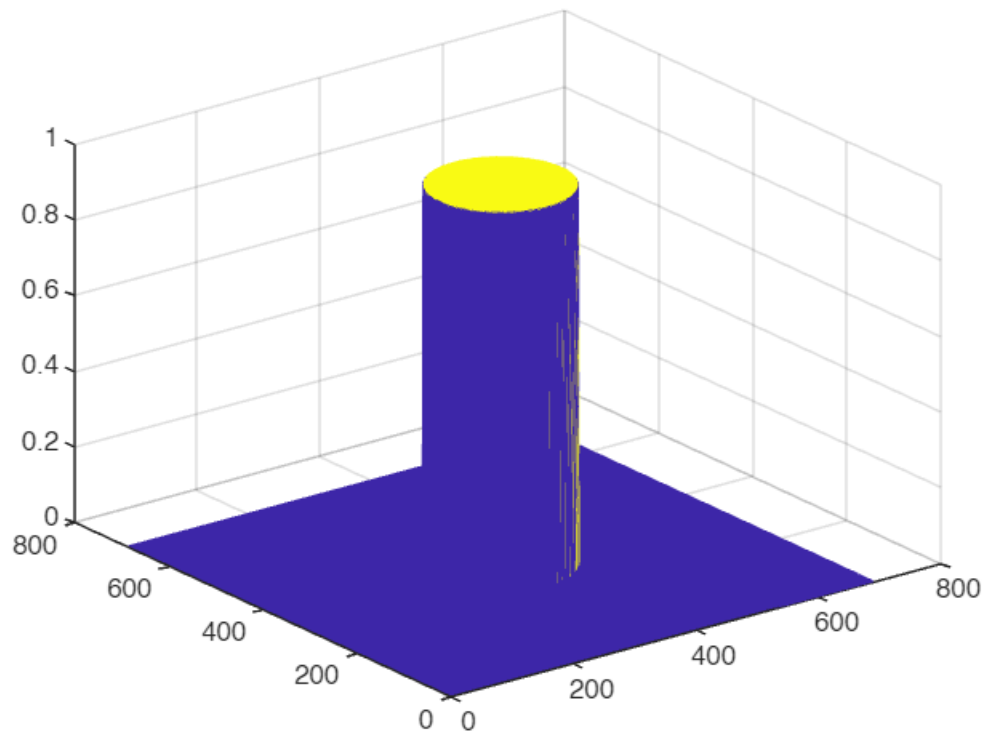
```
[m,n] = size(f);

r=100;
H_ideal = lpfilter('ideal' , m, n, r) ;

figure, imshow(H_ideal,[]);
```



```
[xx,yy] = meshgrid(1:m,1:n);  
figure, mesh(xx,yy,fftshift(H_ideal));
```

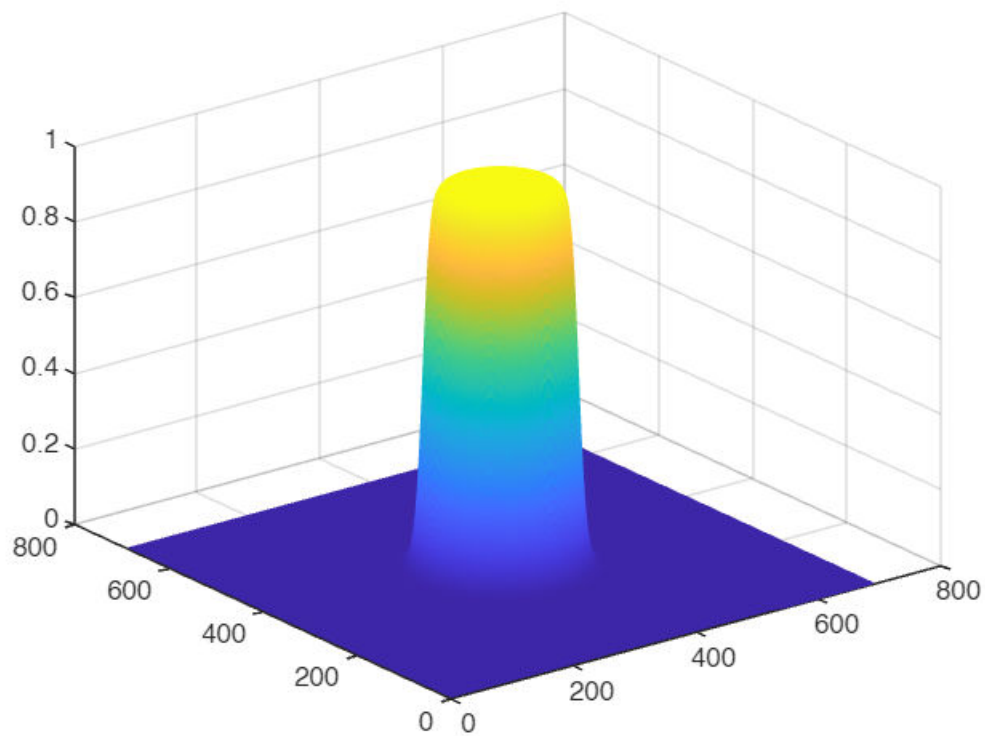


Butterworth LP filter

```
[m,n] = size(f);  
  
r=100;  
rad=10;  
H_btw = lpfilter('btw' , m, n, r,rad) ;  
  
figure, imshow(H_btw,[]);
```



```
[xx,yy] = meshgrid(1:m,1:n);  
figure, mesh(xx,yy,fftshift(H_btw));
```

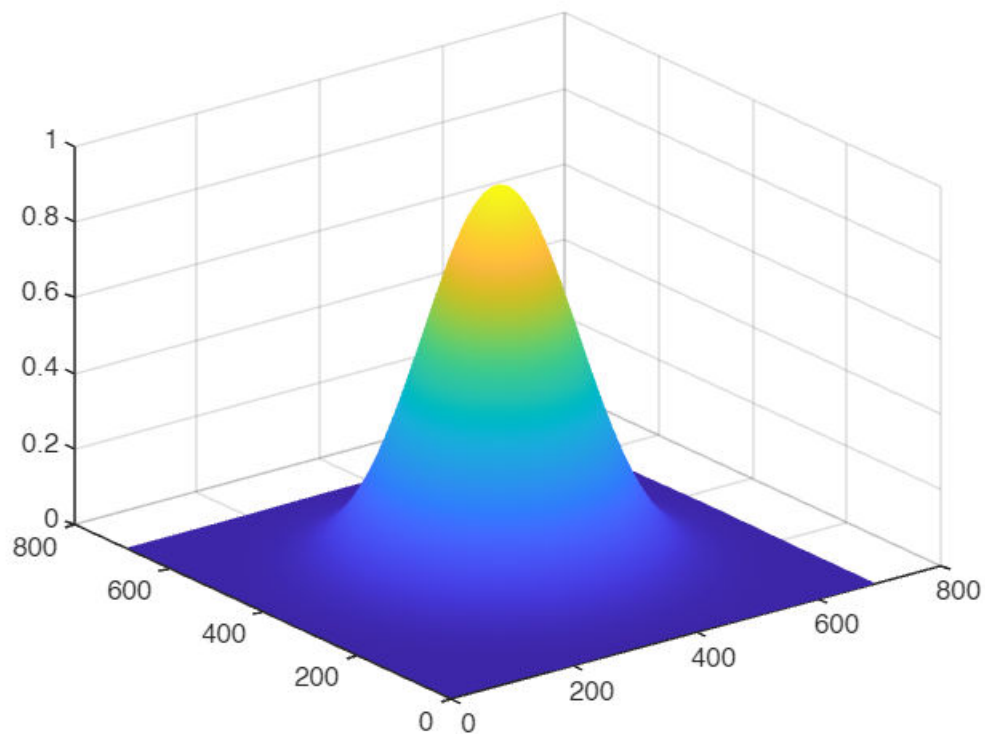



Gaussian LP filtr

```
[m,n] = size(f);  
  
r=100;  
H_gauss = lpfilter('gaussian' , m, n, r) ;  
  
figure, imshow(H_gauss,[]);
```



```
[xx,yy] = meshgrid(1:m,1:n);  
figure, mesh(xx,yy,fftshift(H_gauss));
```



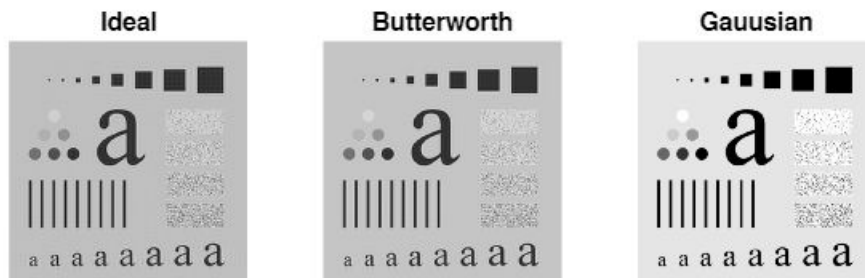
Aplikace filtrů

```
F = fft2(f);
% fitrovani
G = H_ideal.*F;
G2 = H_btw.*F;
G3 = H_gauss.*F;

% prevod zpet
f21=ifft2(G);
f22=ifft2(G2);
f23=ifft2(G3);

f21 = real(f21);
f22 = real(f22);
f23 = real(f23);

figure,
subplot(1,3,1), imshow(f21,[]);
title('Ideal')
subplot(1,3,2), imshow(f22,[]);
title('Butterworth')
subplot(1,3,3), imshow(f23,[]);
title('Gaussian')
```



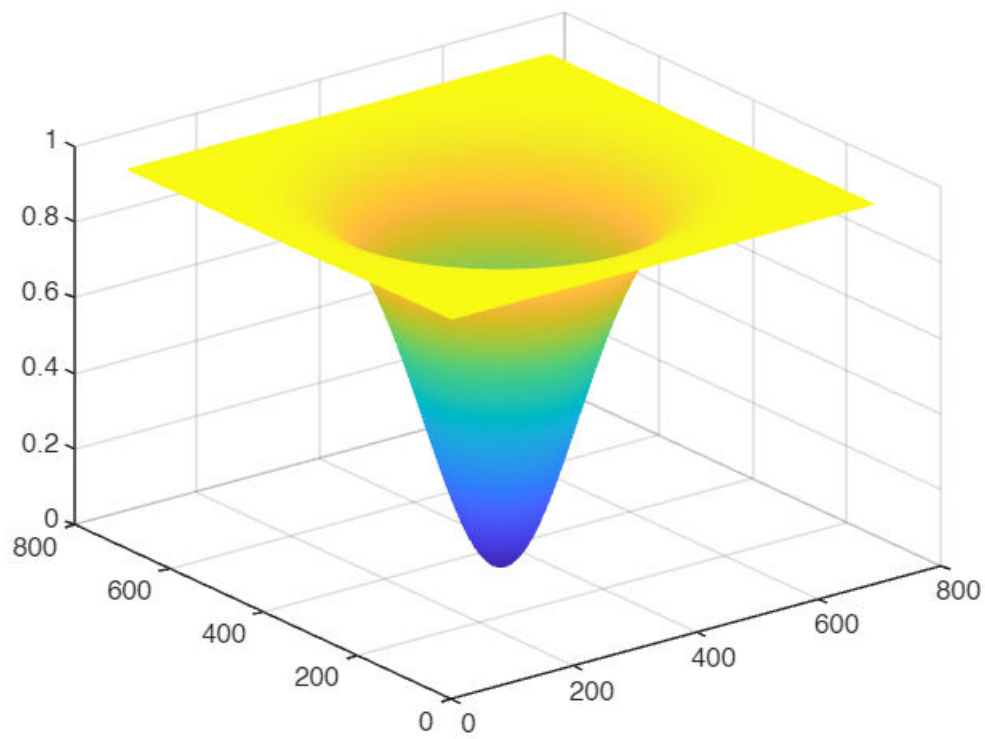
HP filtry

```
H_ideal_hp = hpfilter('ideal' , m, n, r) ;
H_btw_hp = hpfilter('btw' , m, n, r) ;
H_gauss_hp = hpfilter('gaussian' , m, n, r) ;

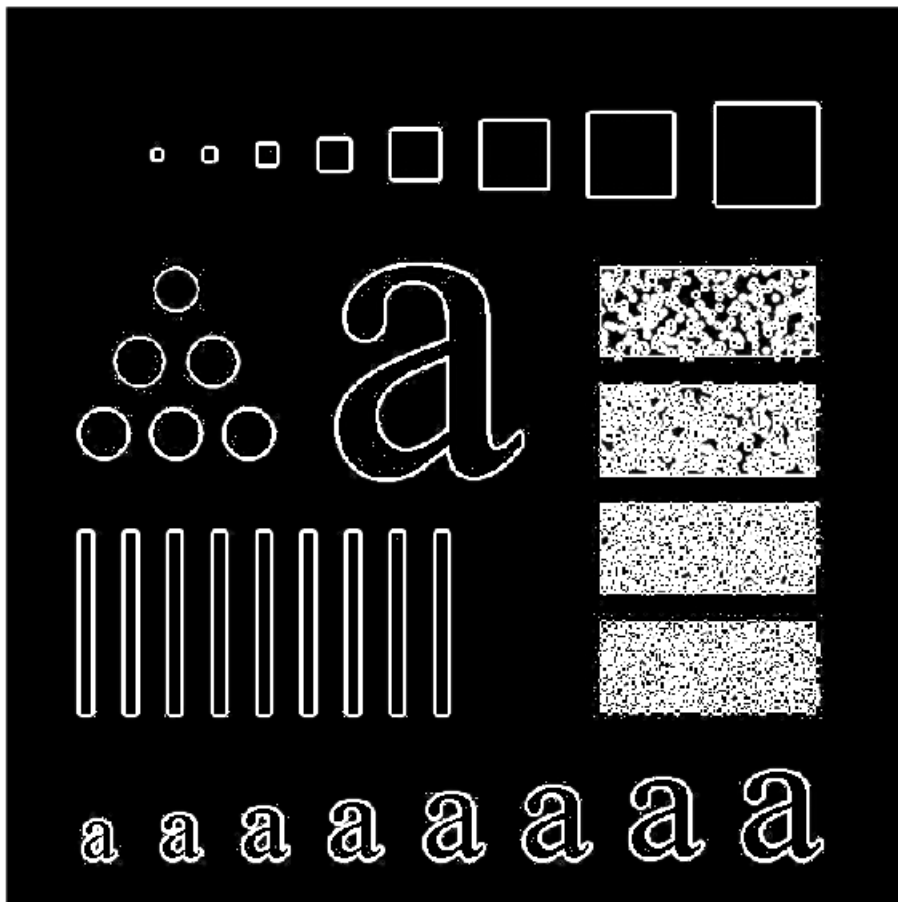
figure, imshow(H_gauss_hp,[]);
```



```
[xx,yy] = meshgrid(1:m,1:n);  
figure, mesh(xx,yy,fftshift(H_gauss_hp));
```

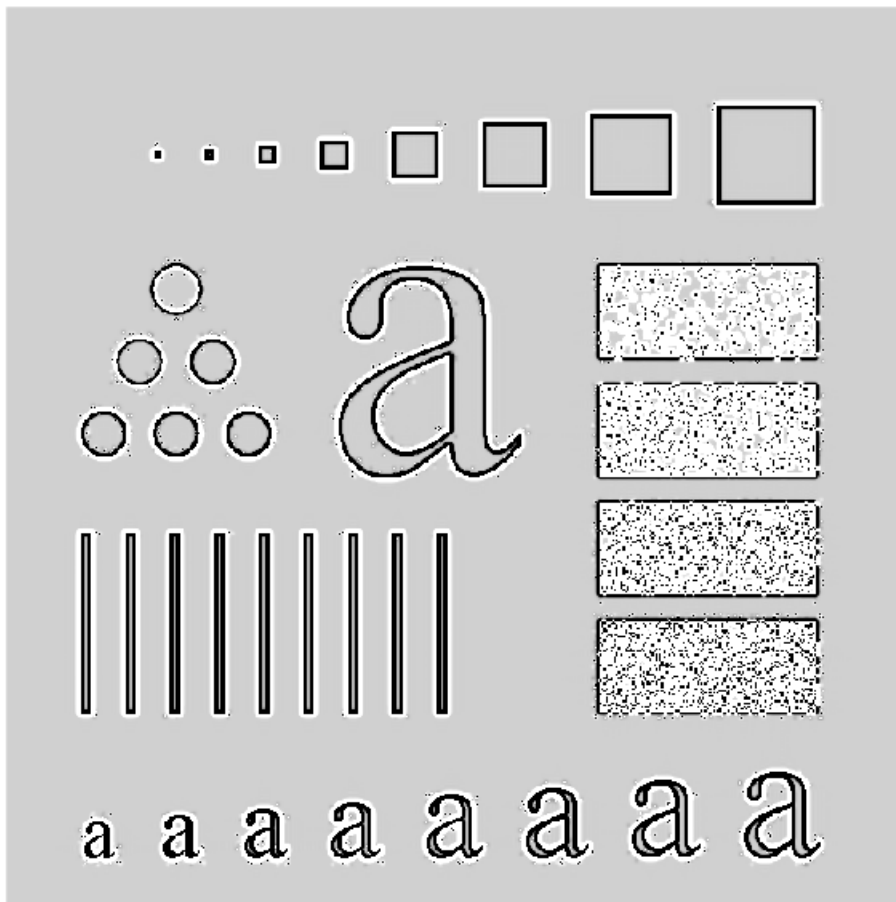


```
F = fft2(f);  
G = H_gauss_hp.*F;f2=ifft2(G);  
f2 = real(f2);  
  
figure,imshow(f2);
```



Odstranili jsme DC koeficient, tedy i informaci o průměrné hodnotě v obrázku.

```
f3 = (F(1,1)/(m*n))/255 + f2;  
figure,imshow(f3);
```

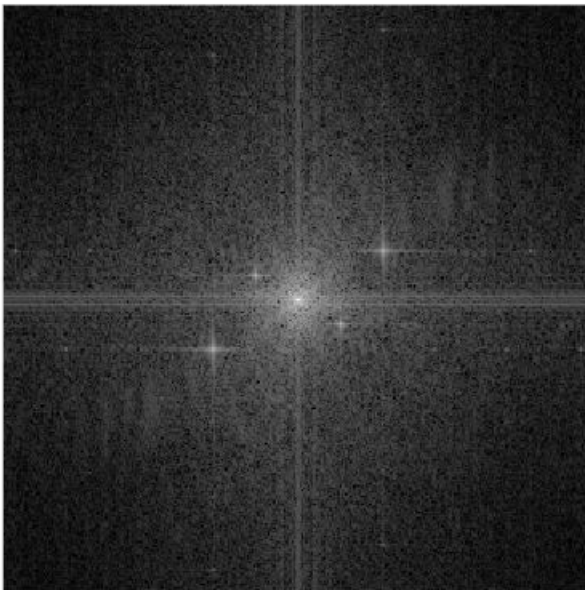


Ukázka šumu - periodický šum

```
f = im2double(rgb2gray(imread('ClownOrig.jpg')));  
[M,N] = size(f);  
figure, imshow(f);
```




```
F = fft2(f);  
S = fftshift(log(1+abs(F)));  
figure, imshow(S,[]);
```



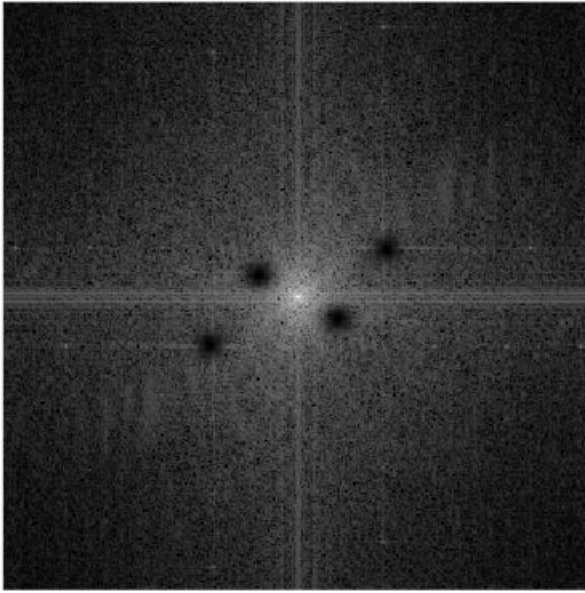
Např. notch zde:

```
C1 = [136 127;  
      123 191];  
H1 = cnotch('gaussian','reject', M,N,C1,5);  
  
imshow(H1,[]);
```



Aplikace

```
P1 = fftshift(H1).*S;  
imshow(P1,[]);
```



```
F = fft2(f, size(H1, 1), size(H1, 2));  
G = H1.*F;  
g1 = real(ifft2(G));  
g1 = g1(1:size(f, 1), 1:size(f, 2));
```

```
figure,  
subplot(1,2,1), imshow(f,[]);  
title('Original')  
subplot(1,2,2), imshow(g1,[]);  
title('Upraveny')
```

Original



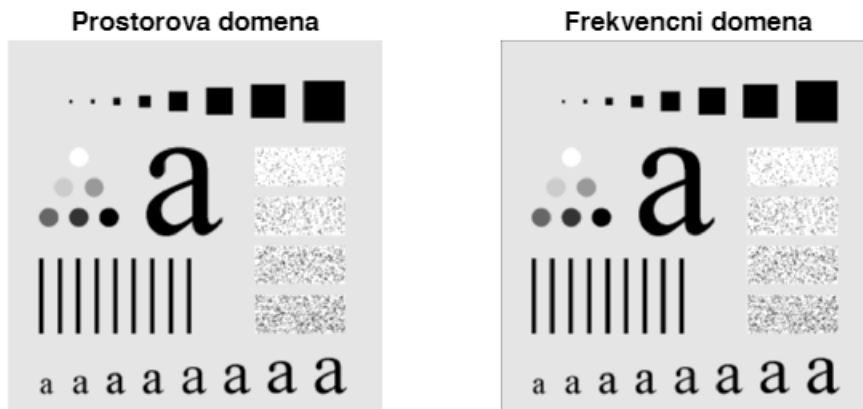
Upraveny



Vytvoření filtru z filtru v prostorové doméně

```
f=rgb2gray(imread('a.png'));  
f = im2double(f);  
[m,n] = size(f);  
  
% konvolucni maska  
h = fspecial("average",5);  
  
F = fft2(f);  
  
% vytvoreni masky ve frekvencni domene o velikosti stejne jako je obrazek  
H = freqz2(h, m, n);  
  
H1 = ifftshift(H);  
  
% fitrovani  
G = H1.*F;  
  
% prevod zpet  
gf=ifft2(G);  
  
gf = real(gf);  
  
gp = imfilter(f,h);
```

```
figure,
subplot(1,2,1), imshow(gf);
title('Prostorova domena')
subplot(1,2,2), imshow(gp,[]);
title('Frekvencni domena')
```



```
d = abs(gp(5:end-5,5:end-5) - gf(5:end-5,5:end-5));

maximalni_rozdil = max(d(:)); % maximalni rozdil

display(maximalni_rozdil);
```

```
maximalni_rozdil = 1.6653e-15
```

```
minimalni_rozdil = min (d(:));
display(minimalni_rozdil);
```

```
minimalni_rozdil = 0
```

Porovnání doby výpočtu

```
f=rgb2gray(imread('a.png'));
[m,n] = size(f);
f = double(f);
```

```

velikost = 95;
h = fspecial("average", velikost);
H = freqz2(h, m, n);
H1 = ifftshift(H);

```

Prostorová

```

tic();
gp = imfilter(f, h, 'conv', 'same');
toc()

```

Elapsed time is 0.052551 seconds.

```
%figure, imshow(gp,[]);
```

Frekvenční

```

tic();
F = fft2(f);
G = H1.*F;
gf=real(ifft2(G));
toc()

```

Elapsed time is 0.037096 seconds.

```
%figure, imshow(gf,[]);
```

Alias

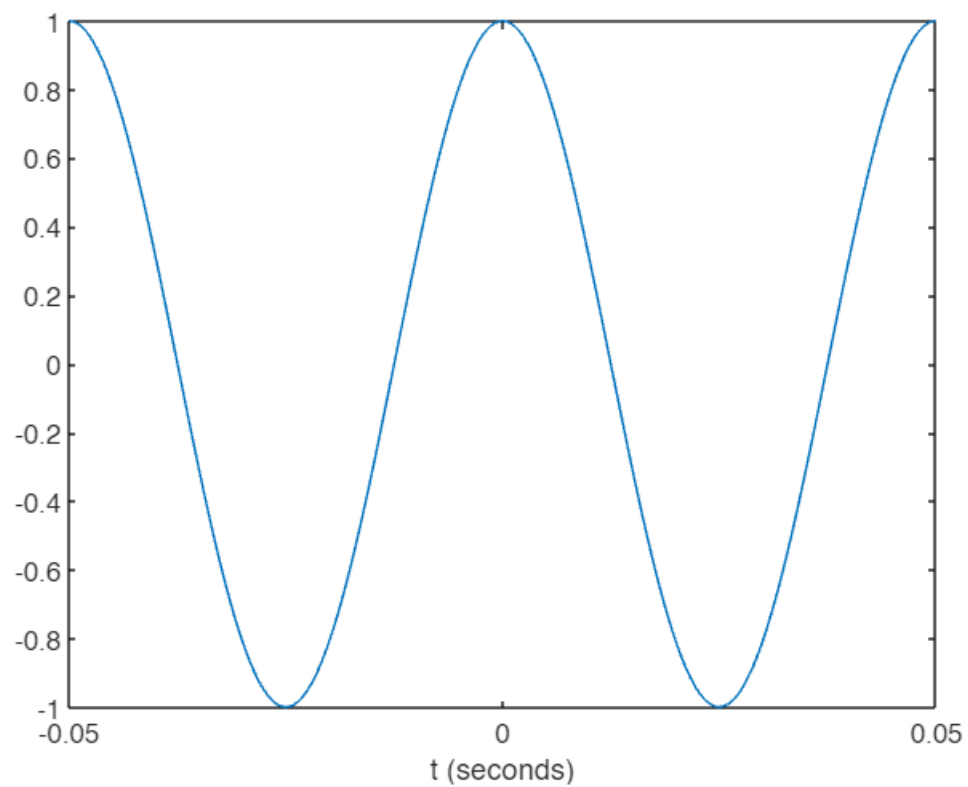
Příklad 1

```

f = 20;
tmin = -0.05;
tmax = 0.05;
t = linspace(tmin, tmax, 400);
x_c = cos(2*pi*f * t);

figure,
plot(t, x_c)
xlabel('t (seconds)');

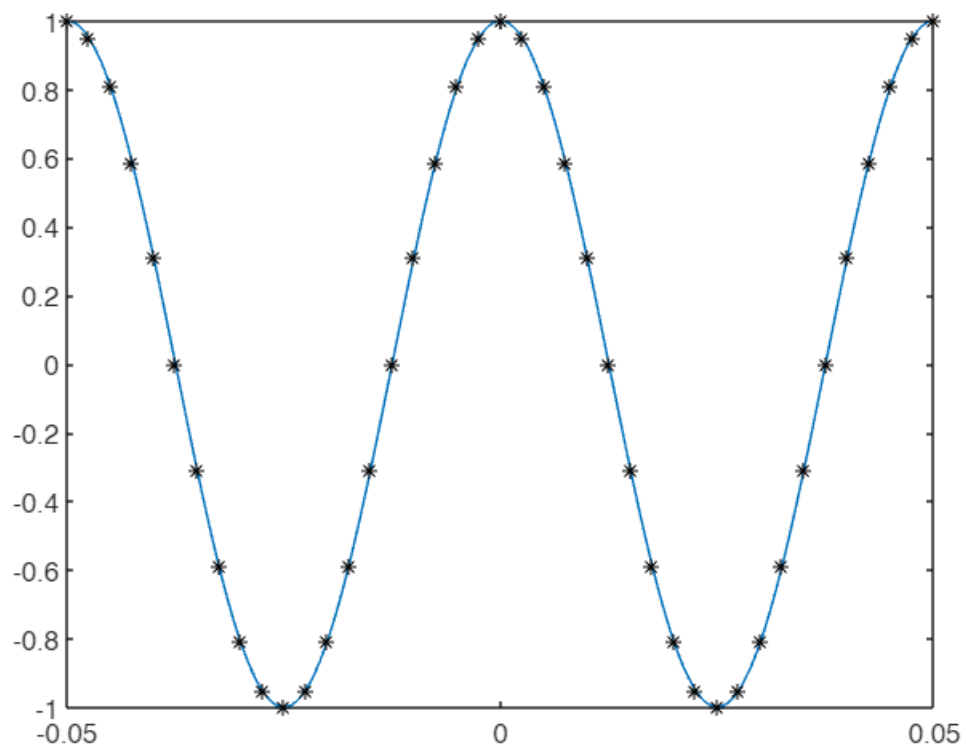
```



Vzorkování s frekvencí 1/400

```
T = 1/400;
nmin = ceil(tmin / T);
nmax = floor(tmax / T);
n = nmin:nmax;
x1 = cos(2*pi*f * n*T);

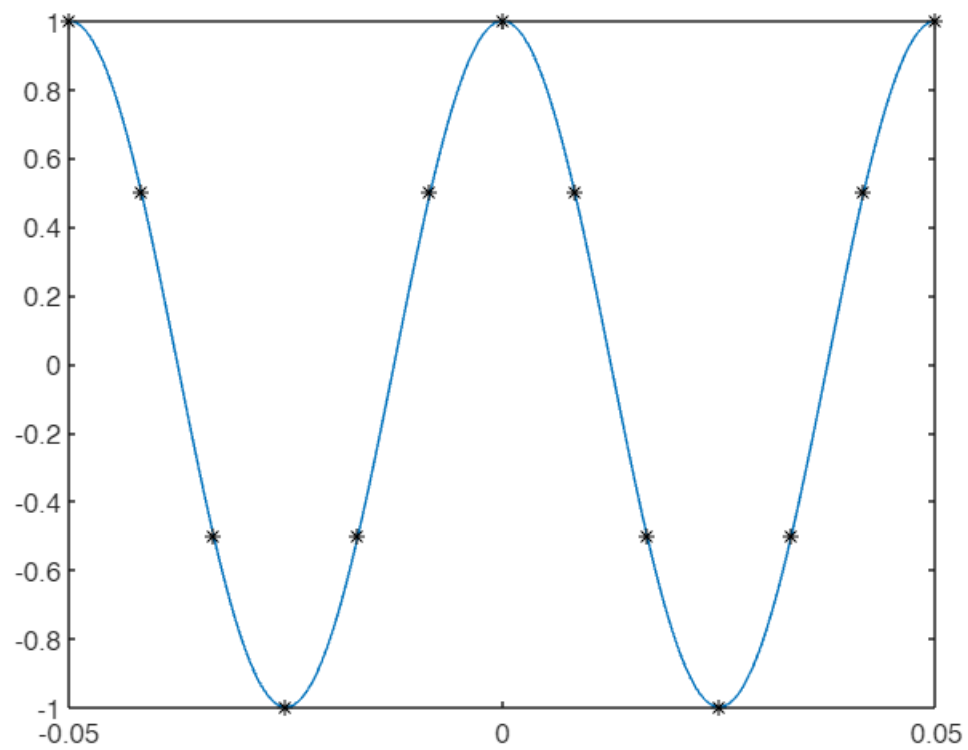
figure,
plot(t, x_c)
hold on
plot(n*T, x1, '*k')
hold off
```



Vzorkování s frekvencí 1/120

```
T = 1/120;
nmin = ceil(tmin / T);
nmax = floor(tmax / T);
n = nmin:nmax;
x1 = cos(2*pi*f * n*T);
```

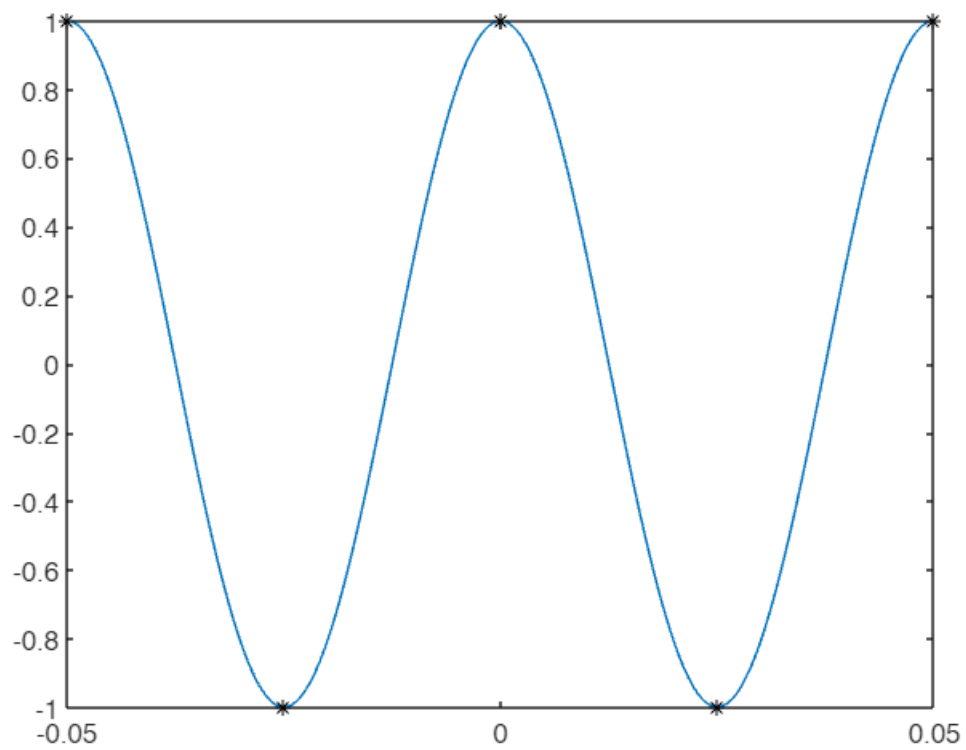
```
figure,
plot(t, x_c)
hold on
plot(n*T, x1, '*k')
hold off
```

Vzorkování s frekvencí 1/40

```
T = 1/40;
nmin = ceil(tmin / T);
nmax = floor(tmax / T);
n = nmin:nmax;
x1 = cos(2*pi*f * n*T);
```

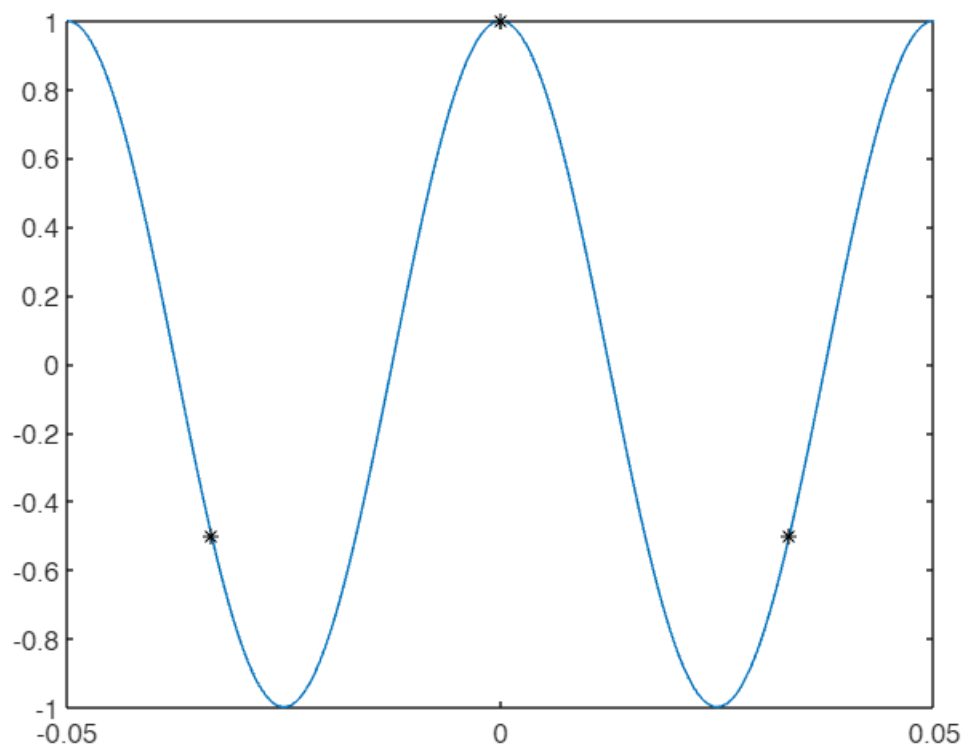
```
figure,
plot(t, x_c)
hold on
plot(n*T, x1, '*k')
hold off
```



Vzorkování s frekvencí 1/30

```
T = 1/30;
nmin = ceil(tmin / T);
nmax = floor(tmax / T);
n = nmin:nmax;
x1 = cos(2*pi*f * n*T);
```

```
figure,
plot(t, x_c)
hold on
plot(n*T, x1, '*k')
hold off
```

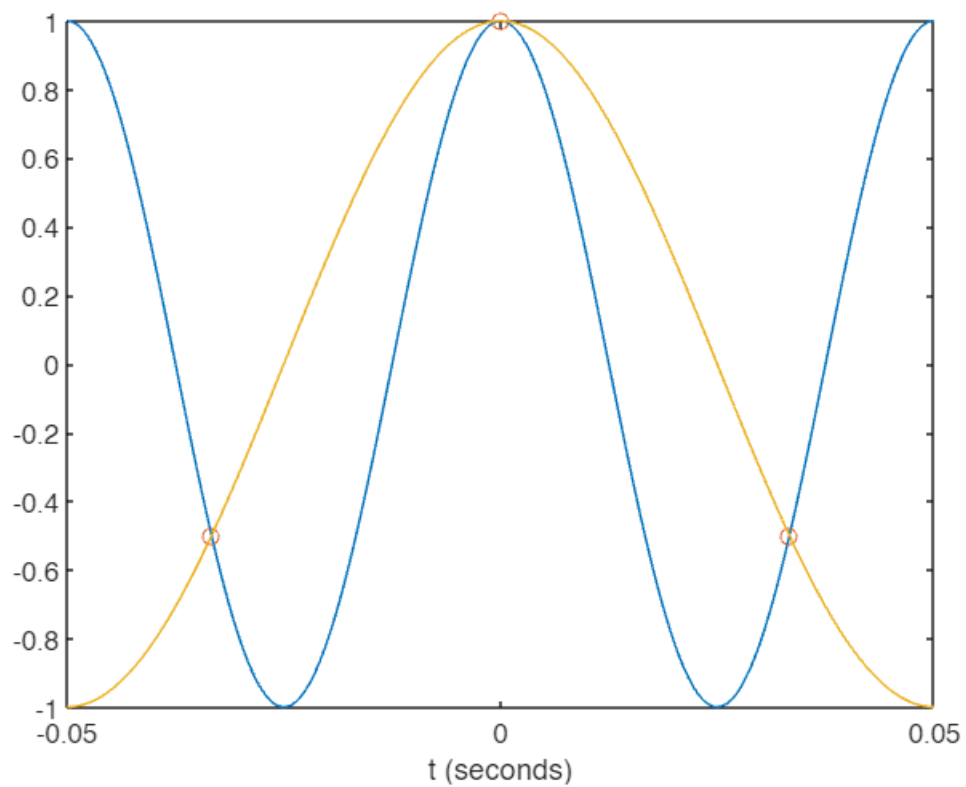


```
f = 20;
tmin = -0.05;
tmax = 0.05;
t = linspace(tmin, tmax, 400);
x_c = cos(2*pi*f * t);
```

```
figure,
plot(t,x_c)
xlabel('t (seconds)');
```

```
T = 1/30;
x_c = cos(2*pi*10 * t);
nmin = ceil(tmin / T);
nmax = floor(tmax / T);
n = nmin:nmax;
x1 = cos(2*pi*f * n*T);
```

```
hold on
plot(n*T, x1, 'o')
plot(t, x_c)
hold off
```

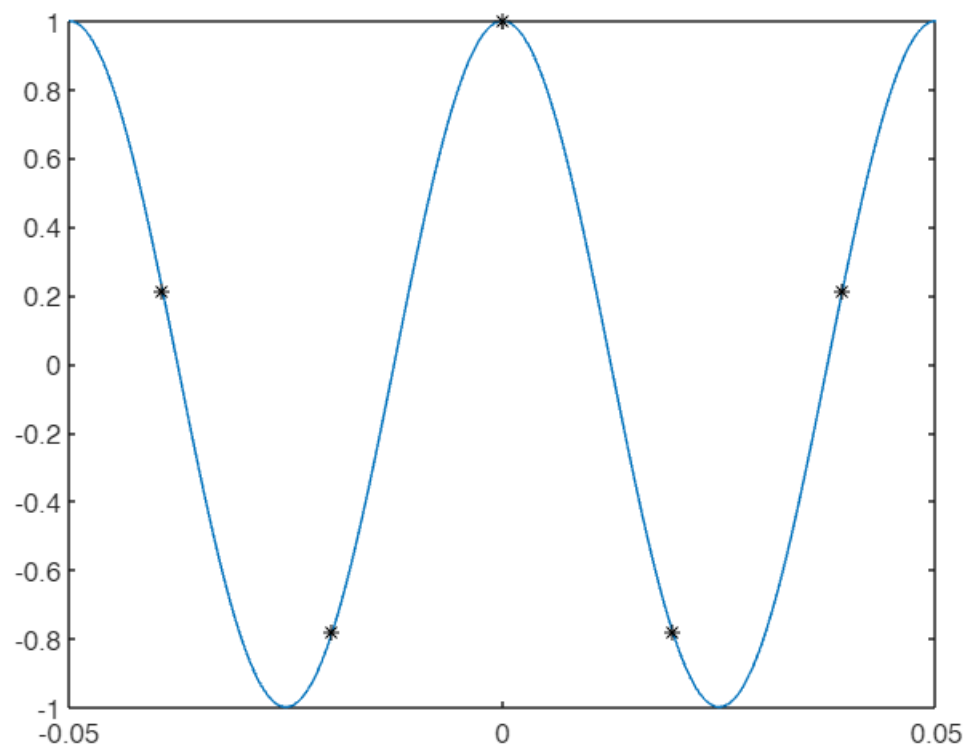


```
f = 20;
tmin = -0.05;
tmax = 0.05;
t = linspace(tmin, tmax, 400);
x_c = cos(2*pi*f * t);

frekv = 51;
T = 1/frekv;

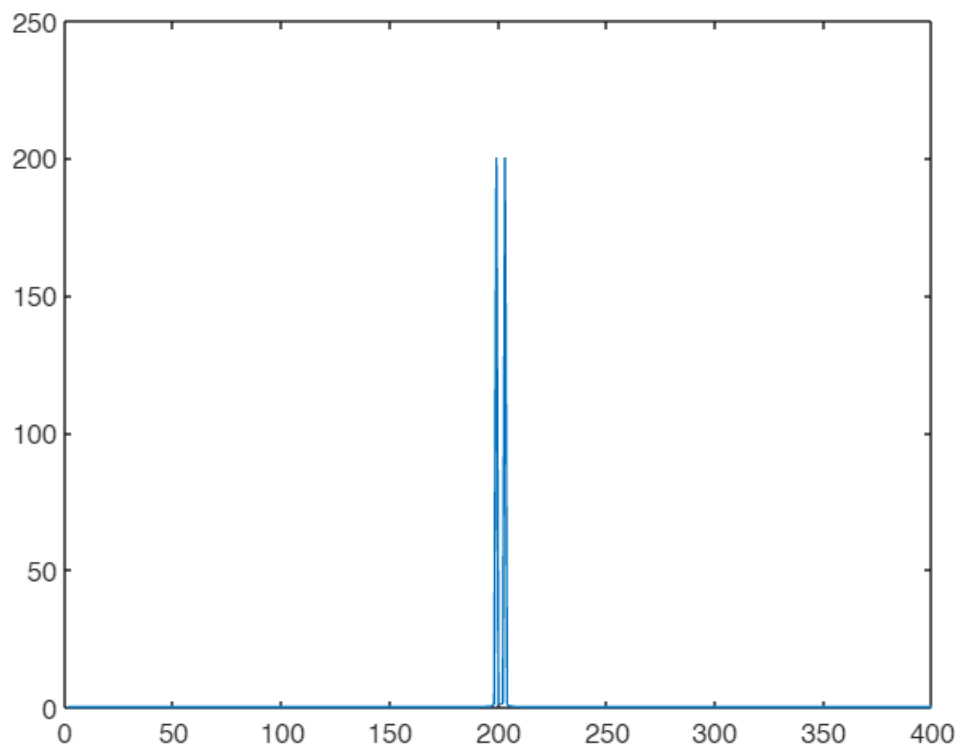
nmin = ceil(tmin / T);
nmax = floor(tmax / T);
n = nmin:nmax;
x1 = cos(2*pi*f * n*T);

figure,
plot(t, x_c)
hold on
plot(n*T, x1, '*k')
hold off
```



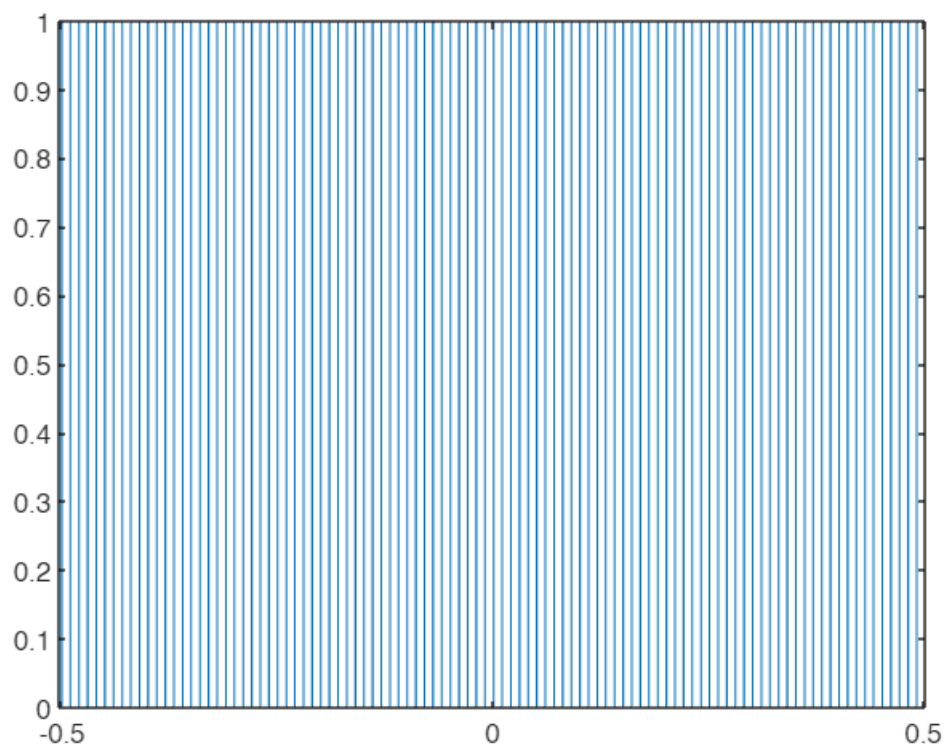
Funkce převedená do frekvenční domény

```
F = fft(x_c);  
F_sp = abs(F);  
  
figure,  
plot(fftshift(F_sp));
```



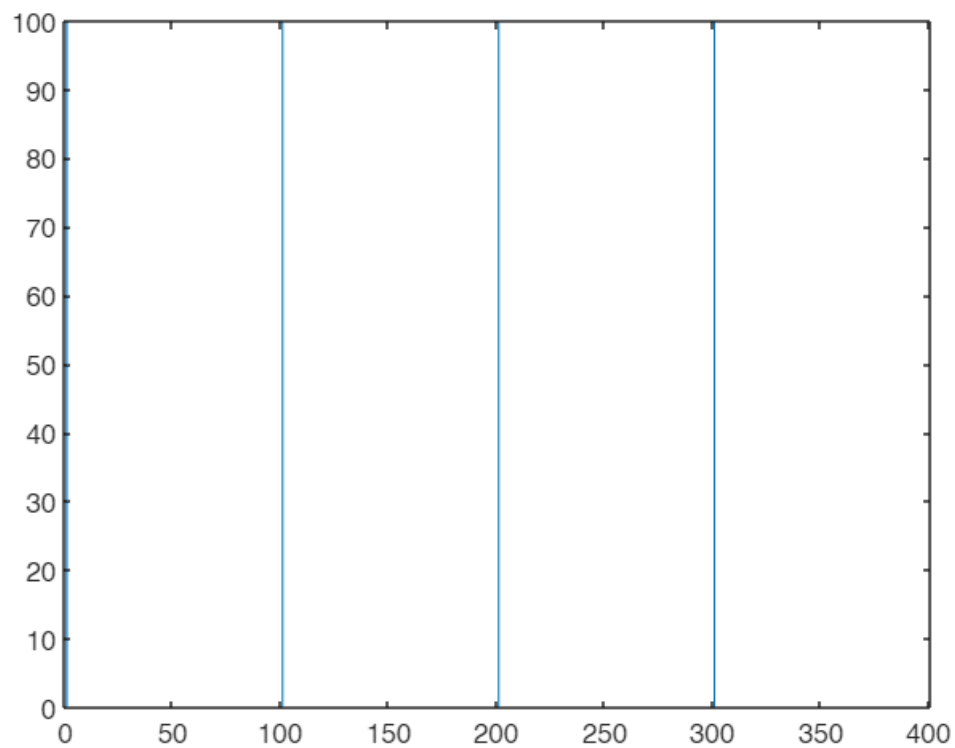
Diracův pulz

```
f = 4;  
%f = 10;  
tmin = -0.5;  
tmax = 0.5;  
t = linspace(tmin, tmax, 400);  
  
x1 = zeros([1,400]);  
x1(1:f:400) = 1;  
  
figure,  
bar(t, x1);
```



Ve frekvenční doméně

```
F = fft(x1);  
F_sp = abs(F);  
  
figure,  
bar(fftshift(F_sp));
```



Příklad 2 - Časový alias

```
i1 = rgb2gray(imread('Run1.png'));  
i2 = rgb2gray(imread('Run2.png'));  
i3 = rgb2gray(imread('Run3.png'));  
i4 = rgb2gray(imread('Run4.png'));  
i5 = rgb2gray(imread('Run5.png'));  
i6 = rgb2gray(imread('Run6.png'));  
i7 = rgb2gray(imread('Run7.png'));  
i8 = rgb2gray(imread('Run8.png'));
```

```
I = [];  
I = uint8(I);
```

```
% vytvoreni sekvence
```

```
for i = 1 : 8 : 500  
    I(:, :, i) = i1;  
    I(:, :, i+1) = i2;  
    I(:, :, i+2) = i3;  
    I(:, :, i+3) = i4;  
    I(:, :, i+4) = i5;  
    I(:, :, i+5) = i6;  
    I(:, :, i+6) = i7;  
    I(:, :, i+7) = i8;
```

```
end
```



```
figure,  
subplot(1,8,1), imshow(i1);  
subplot(1,8,2), imshow(i2);  
subplot(1,8,3), imshow(i3);  
subplot(1,8,4), imshow(i4);  
subplot(1,8,5), imshow(i5);  
subplot(1,8,6), imshow(i6);  
subplot(1,8,7), imshow(i7);  
subplot(1,8,8), imshow(i8);
```



Krok 1

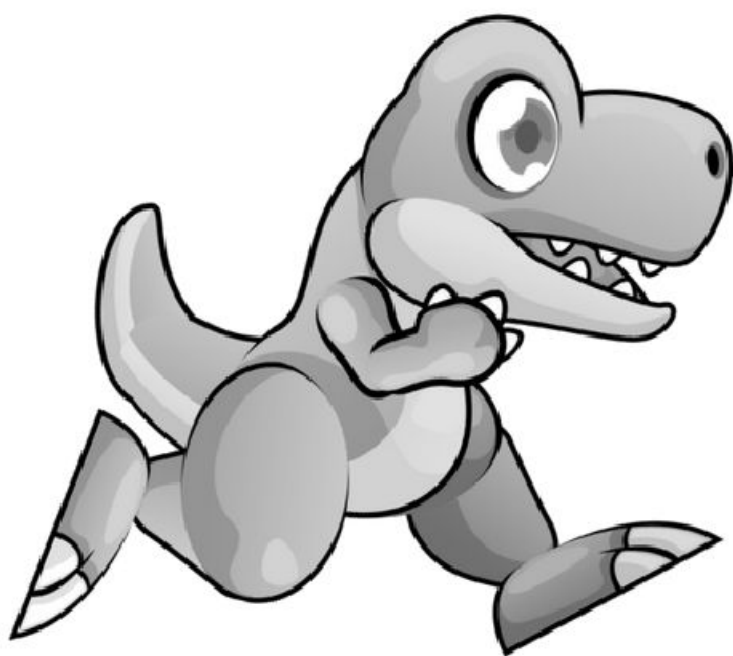
```
figure,  
for i = 1:size(I,3)  
    imshow(I(:,:,i))  
    pause(.1);  
end
```





Krok 4

```
figure,  
for i = 3:4:size(I,3)  
    imshow(I(:,:,i))  
    pause(.1);  
end
```

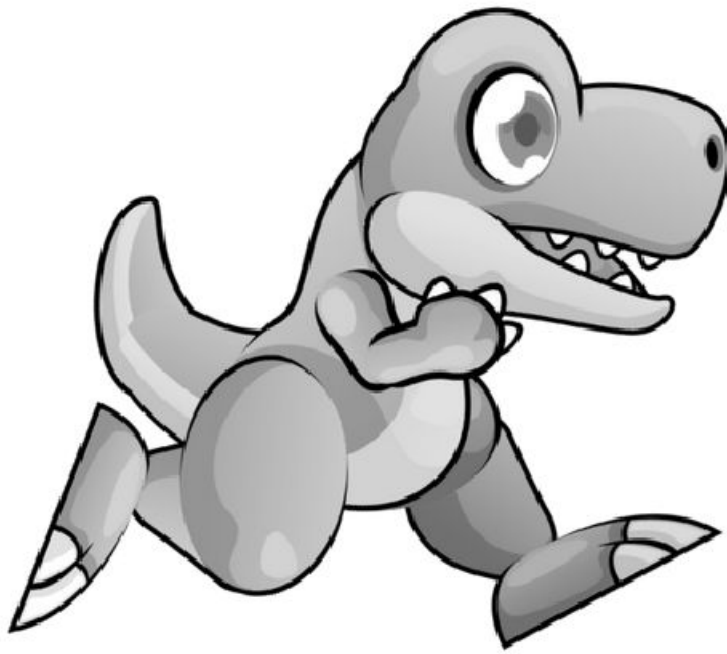




Krok 7

```
figure,  
for i = 1:6:size(I,3)  
    imshow(I(:,:,i))  
    pause(.1);  
end
```





Krok 10

```
figure,  
for i = 1:8:size(I,3)  
    imshow(I(:,:,i))  
    pause(.1);  
end
```





Krok 11

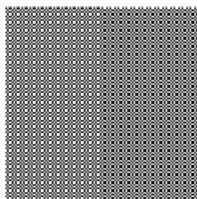
```
figure,  
for i = 1:9:size(I,3)  
    imshow(I(:,:,i))  
    pause(.1);  
end
```





Příklad 3 - šachovnice

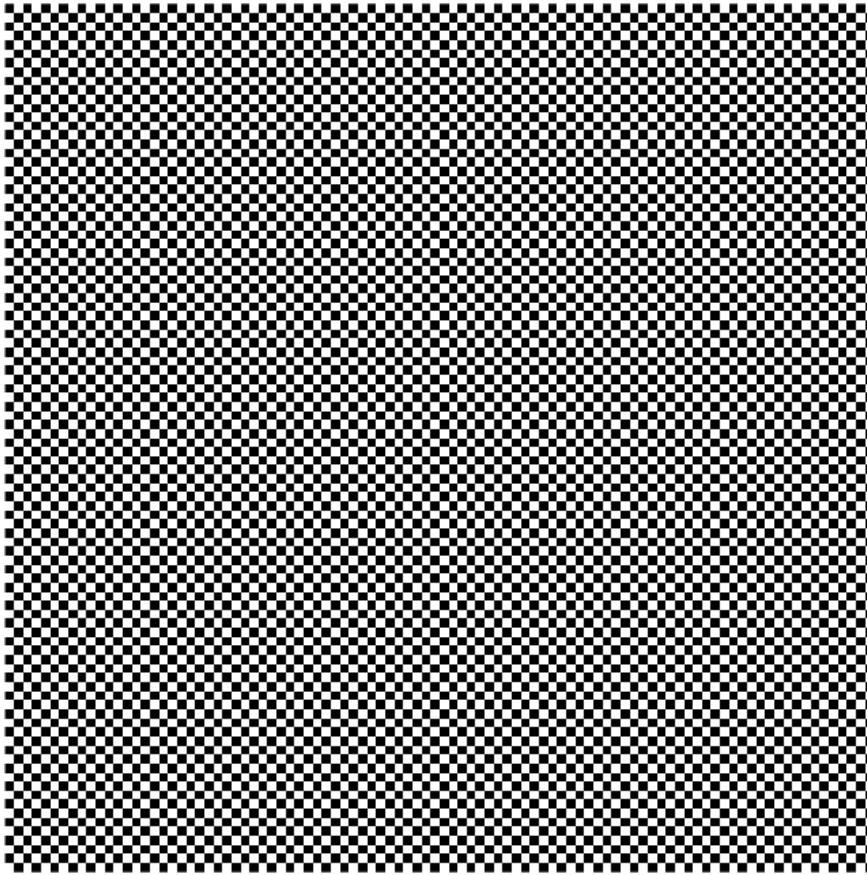
```
% generování obrazku sachovnice  
sachovnice = mat2gray(checkerboard(1,48,48) > 0.5);  
figure,  
imshow(sachovnice);
```



Vzorkování s krokem 1/6

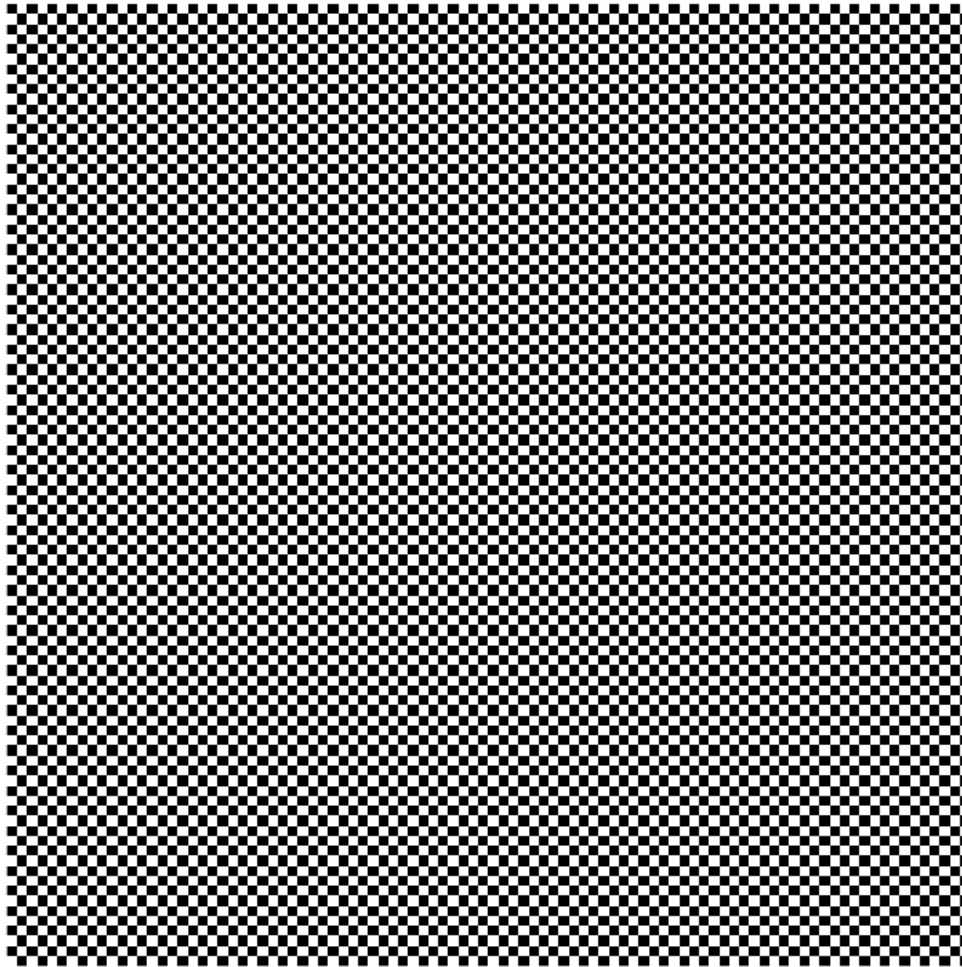
```
s2 = imresize(sachovnice,6,'nearest');  
figure,
```

```
imshow(s2);
```



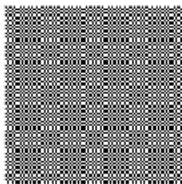
Vzorkování s krokem 1/16

```
s3 = imresize(sachovnice,16,'nearest');  
figure,  
imshow(s3);
```



Vzorkování s krokem 1/0.9174

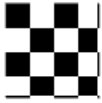
```
s4 = imresize(sachovnice,0.9174,'nearest');  
figure,  
imshow(s4);
```



Vzorkování s krokem 1/0.4798

```
s5 = imresize(sachovnice,0.4798,'nearest');
```

```
figure,  
imshow(s5);
```



Vzorkování s krokem 1/0.5

```
s6 = imresize(sachovnice,0.5,'nearest');  
figure,  
imshow(s6);
```



Příklad 4 - moiré

```
%I = rgb2gray(imread('alias4.png'));  
I = imread('aliasUSAggray.png');  
figure,  
imshow(I);
```



Menší vzorkovací frekvence

```
I05 = I(1:2:end,1:2:end);  
figure,  
imshow(I05);
```



Zvětšení vzorkovaného obrázku

```
I1 = uint8(zeros(size(I)));  
I1(1:2:end,1:2:end)=I05;  
I1(1:2:end,2:2:end)=I05;  
I1(2:2:end,2:2:end)=I05;  
I1(2:2:end,1:2:end)=I05;  
  
figure,imshow(I1);
```




Rozmazání obrázku před vzorkováním (méně detailů)

```
w = 1/9 * [1 1 1;  
          1 1 1;  
          1 1 1];  
Iblur = imfilter(I,w,'corr','same');  
figure,  
imshow(Iblur);
```



Vzorkování rozmazaného obrázku

```
I05blur = Iblur(1:2:end,1:2:end);  
figure,  
imshow(I05blur);
```



Zvětšení vzorkovaného obrázku

```
I1blur = uint8(zeros(size(I)));  
I1blur(1:2:end,1:2:end)=I05blur;  
I1blur(1:2:end,2:2:end)=I05blur;  
I1blur(2:2:end,2:2:end)=I05blur;  
I1blur(2:2:end,1:2:end)=I05blur;  
figure,  
imshow(I1blur);
```



Srovnání

```
figure,  
subplot(1,3,1), imshow(I);  
title('puvodni')  
subplot(1,3,2), imshow(I1);  
title('vzorkovany puvodni')  
subplot(1,3,3), imshow(I1blur);  
title('vzorkovany rozmazany')
```

puvodni



vzorkovany puvodni

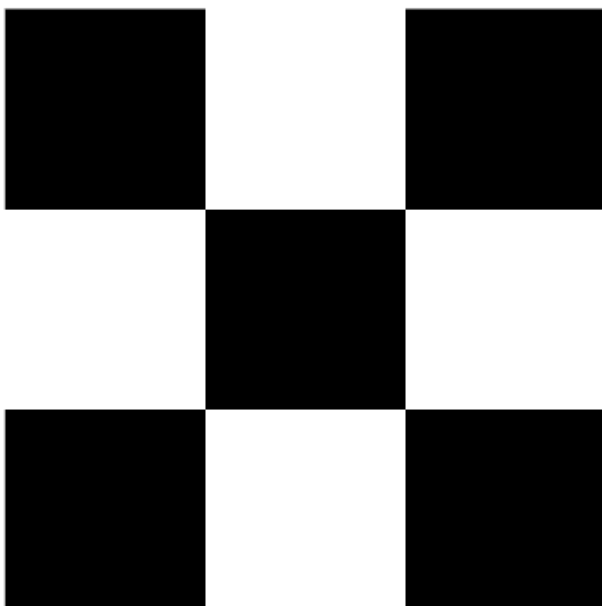


vzorkovany rozmazany



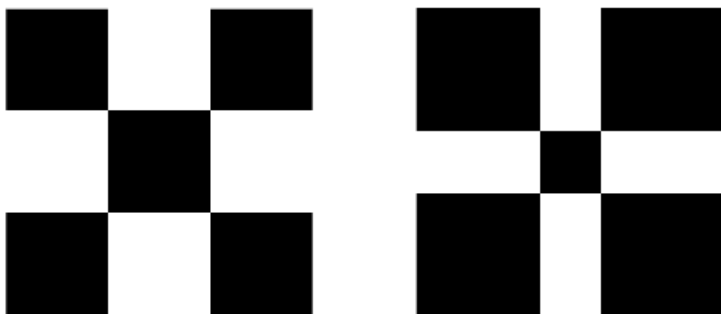
Supersampling

```
C = imread('sachovnice.png');  
figure,  
imshow(C);
```



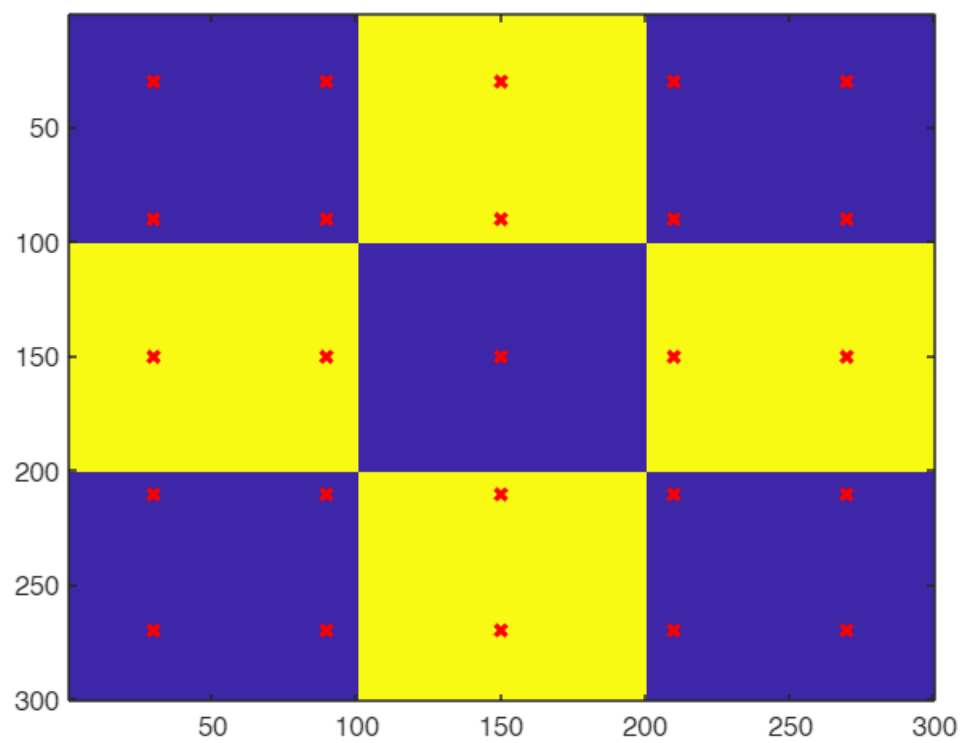
Vzorkování

```
CC = imresize(C, [5 5], 'nearest');  
CCC = imresize(CC, [300 300], 'nearest');  
  
subplot(1,2,1), imshow(C);  
subplot(1,2,2), imshow(CCC);
```



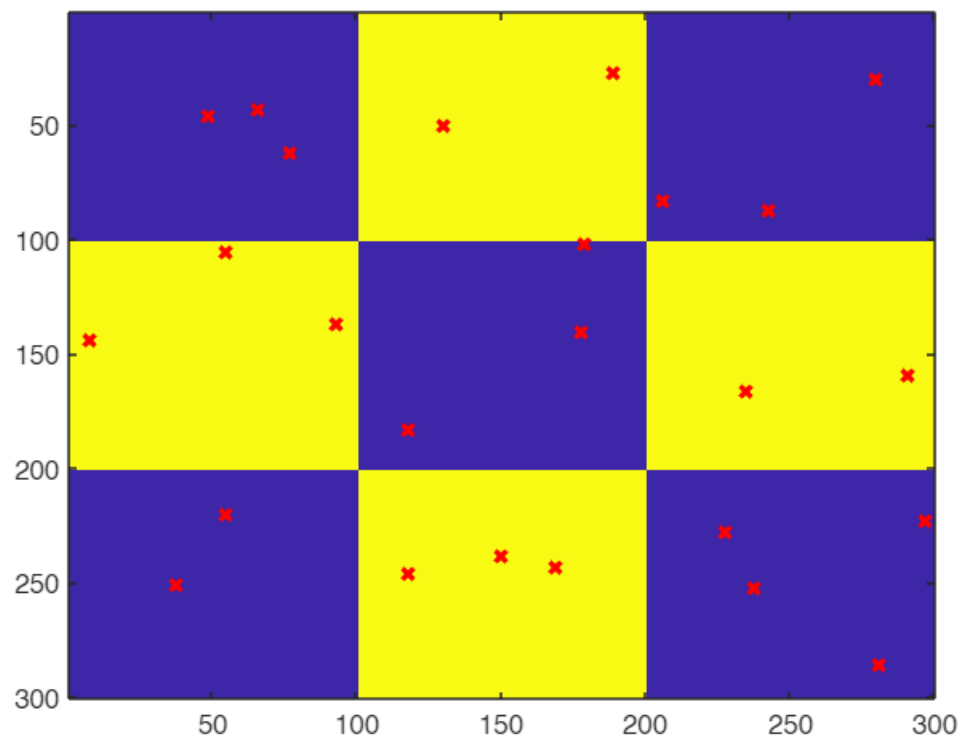
Pravidelne vzorkovani

```
figure,  
[ p_g, v_g ] = Sampling_Grid( C, 20, true );
```



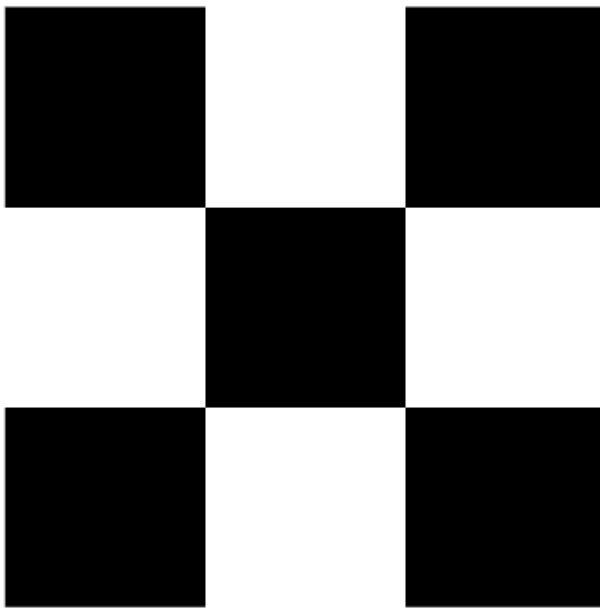
Roztřesené vzorkování

```
figure,  
[ p_j, v_j ] = Sampling_Jittered( C, 20, true );
```

Příklad down sampling na $m_1 \times n_1$ velikost

```
C = imread('sachovnice.png');  
figure,  
imshow(C);
```



```
[m,n] = size(C);
```

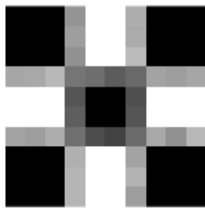
Náhodný výběr

```
m1 = 10;  
n1 = 10;  
  
% velikosti oblasti:  
mm = floor(m/m1);  
nn = floor(n/n1);  
  
k = 100;  
  
vysl = uint8(zeros(m1,n1));
```

Vytvoříme oblasti a z každé náhodně vybereme k vzorků a vezmeme jejich průměr

```
for i = 1:m1  
    for j = 1:n1  
        oblast = C((i-1)*mm + 1 : i*mm, (j-1)*nn + 1 : j*nn);  
        %imshow(oblast);  
        oblast = oblast(:);  
        permutace = randperm(size(oblast,1));  
        oblast = oblast(permutace);  
        vysl(i,j) = mean(oblast(1:k));  
    end  
end
```

```
figure,
imshow(imresize(vysl,10,'nearest')); % zvětšení
```



Pravidelná mřížka

```
C = imread('sachovnice.png');
[m,n] = size(C);

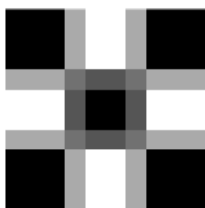
m1 = 10;
n1 = 10;

% velikosti oblasti:
mm = floor(m/m1);
nn = floor(n/n1);

velikost_mrizky = 10;

vysl = uint8(zeros(m1,n1));
% vytvorim oblast
for i = 1:m1
    for j = 1:n1
        oblast = C((i-1)*mm + 1 : i*mm, (j-1)*nn + 1 : j*nn);
        vzorky = [];
        % zde vybírá vzorky vzdalene od sebe dle velikosti mřížky
        for l1 = 1 : velikost_mrizky : mm %mm je velikost oblasti
            for l2 = 1 : velikost_mrizky: nn %nn je velikost oblasti
                vzorky = [vzorky, oblast(l1,l2)]; %kazdy vzorek ukladam do vektoru
            end
        end
        vysl(i,j) = sum(vzorky)/size(vzorky,2); %vypocet prumerne hodnoty vektoru
    end
end

figure,
imshow(imresize(vysl,10,'nearest'));
```



n věží

```
[m,n] = size(C);

%nova velikost
m1 = 100;
n1 = 100;

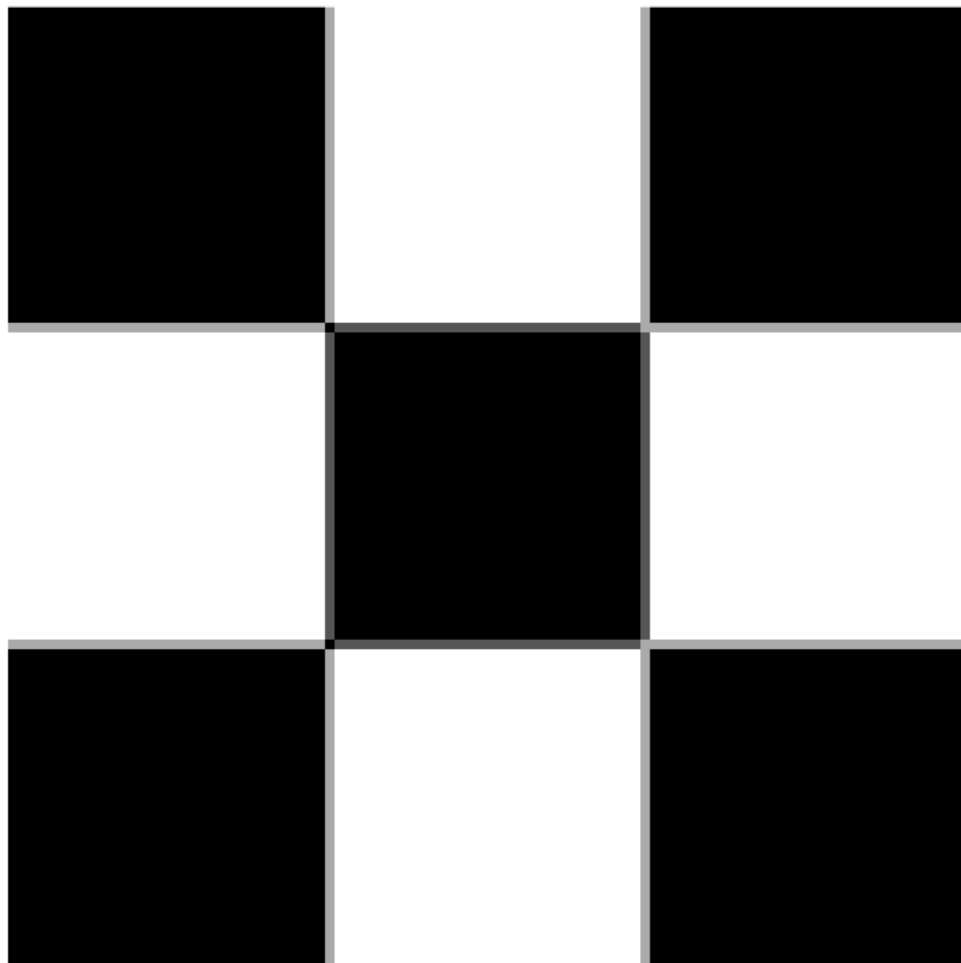
% velikosti oblasti:
mm = floor(m/m1);
nn = floor(n/n1);

vysl = uint8(zeros(m1,n1));

% vytvoreni vseh permutaci
M = eye(mm);
permutace = perms(1:mm);
pocet_permutaci = size(permutace,1);

for i = 1:m1
    for j = 1:n1
        oblast = C((i-1)*mm + 1 : i*mm, (j-1)*nn + 1 : j*nn);
        %randi(pocet_permutaci) vrati nahodne cislo
        vzorky = sum(sum(oblast.*uint8(M(:,permutace(randi(pocet_permutaci),:)))));
        vysl(i,j) = vzorky/mm;
    end
end

figure,
imshow(imresize(vysl,10,'nearest'));
```



Permutace diagonály matice $m \times n$

```
n=4;
M = eye(n);

permutace = perms(1:n);

pocet_permutaci = size(permutace,1);

for i = 1 : pocet_permutaci
    disp(M(:,permutace(i,:)));
end
```

```
0    0    0    1
0    0    1    0
0    1    0    0
1    0    0    0

0    0    1    0
0    0    0    1
0    1    0    0
1    0    0    0
```

0	0	0	1
0	1	0	0
0	0	1	0
1	0	0	0
0	0	1	0
0	1	0	0
0	0	0	1
1	0	0	0
0	1	0	0
0	0	0	1
0	0	1	0
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0
0	0	0	1
0	0	1	0
1	0	0	0
0	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0
0	1	0	0
0	0	0	1
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0
0	0	1	0
0	0	0	1
0	1	0	0
0	0	1	0
1	0	0	0
0	0	0	1
0	1	0	0
0	0	1	0
1	0	0	0
0	0	0	1
0	0	1	0
1	0	0	0
0	1	0	0
0	0	0	1
0	0	0	1
0	0	0	1
0	1	0	0
1	0	0	0
0	0	1	0
0	1	0	0
0	0	0	1
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
1	0	0	0
0	1	0	0
0	0	0	1
0	0	0	1
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
1	0	0	0
0	1	0	0
0	0	0	1
0	0	0	1
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
1	0	0	0
0	1	0	0
0	0	0	1
0	0	0	1
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
1	0	0	0
0	1	0	0
0	0	0	1
0	0	0	1
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
1	0	0	0
0	1	0	0
0	0	0	1
0	0	0	1
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
1	0	0	0
0	1	0	0
0	0	0	1
0	0	0	1
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
1	0	0	0
0	1	0	0
0	0	0	1
0	0	0	1
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
1	0	0	0
0	1	0	0
0	0	0	1
0	0	0	1
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
1	0	0	0
0	1	0	0
0	0	0	1
0	0	0	1
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
1	0	0	0
0	1	0	0
0	0	0	1
0	0	0	1
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
1	0	0	0
0	1	0	0
0	0	0	1
0	0	0	1
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
1	0	0	0
0</			

0	0	1	0
0	0	1	0
1	0	0	0
0	1	0	0
0	0	0	1
0	1	0	0
1	0	0	0
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0
0	0	1	0
0	0	0	1
0	0	0	1
1	0	0	0
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0
0	0	1	0
0	0	0	1
0	1	0	0
1	0	0	0
0	0	0	1
0	1	0	0
0	0	1	0
1	0	0	0
0	0	1	0
0	1	0	0
0	0	0	1
1	0	0	0
0	1	0	0
0	0	0	1
0	0	1	0
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1