

VSTUPNÍ DATA

```
%<<zdroj_ai_fii_fi_omi>>

function [Zdroj,Ai,fii,fi,omi] = zdroj_ai_fii_fi_omi(ZZdroj,ivar)

    Zdroj=ZZdroj(ivar,:);

    Ai=[200 150 170];
    fii=[0 pi 0];
    fi=[300 600 440];
    omi=2*pi*fi;

    % ZADANE HODNOTY AMPLITUDY TLAKU ZDROJE
    % ZADANE HODNOTY FAZE ZDROJE
    % ZADANE HODNOTY FREKVENCE ZDROJU
    % ZADANE HODNOTY UHLOVE FREKVENCE ZDROJU

% nalezeni indexu vsech pozic v Zdroj, kde je hodnota nenulova
adr=find(Zdroj ~=0);
Zdroj=Zdroj(adr);
Ai=Ai(adr);
fii=fii(adr);
omi=omi(adr);
fi=fi(adr);
```

DEFINICE PROSTŘEDÍ

```
%<<mriz>>

function [Lmax,tmax,DATA]=mriz()

% souradnice uzlovych bodu site
mriz=[

1.00      0.00   0.00   0.00
2.00      0.20   0.00   0.00
... (pro byly přehlednost vynechány řádky)
960.00    5.80   6.00   0.00
961.00    6.00   6.00   0.00
```

```

];

uzel=[mriz(:,1), mriz(:,2), mriz(:,3), mriz(:,4)]; % deklarace promenne uzel: poradi, X, Y, Z

T=((800-300)/6.0)*(6.0-uzel(:,2))+300; % teplota prostredi T [°C], linearni funkce, okrajove hodnoty: 800°C, 300°C
ro=101325./(287.05*(273.16+T)); % hustota vzduchu ro=p0/RT [kg/m^3]
c0=331.5+0.61*T; % rychlost vzduchu c0 [m/s] např. dle Merhauta

DATA=[uzel T ro c0]; % deklarace promenne DATA: poradi, X, Y, Z, teplota, hustota, rychlost v uzlu

deltax=max(DATA(:,2)); % maximalni vzdalenost na ose x
deltay=max(DATA(:,3)); % maximalni vzdalenost na ose y
Lmax=sqrt(deltax.^2+deltay.^2); % maximalni dimenze definovaneho prostoru (delka uhlopricky v rovine xy)
tmax=Lmax/min(DATA(:,7)); % maximalni doba sireni vlny od zdroje na hranici definovaneho prostoru

```

HLAVNÍ FUNKCE

```

%<<main_function>>

clear all, close all % pripraveni programu

zzdroj=[473 481 489] % ZADANE POLOHY ZDROJU

[Lmax,tmax,DATA]=mriz; % nacteni max dimenze, max casu, uzlu, teploty, ro, c0
ii=1:length(zzdroj(:,1)); % promenna
[Zdroj,Ai,fii,fi,omi]=zdroj_ai_fii_fi_omi(zzdroj,ii); % nacteni vstupnich dat
lambdamax=DATA(1,end)/max(fi); % max vlnova delka lambda = c0/f

[BP]=BP(DATA,Zdroj,fi); % nacteni uzlu lezicich v blizkem poli

fig1=1;fig2=2;fig3=3;fig4=4;fig5=5;fig6=6;fig7=7;fig8=8; % definovani promennych pro tvorbu obrazku
fig9=9;fig10=10;fig11=11;fig12=12;fig13=13;fig14=14;fig15=15;

% komplexni intenzita

```

```

for ii=ii
    Ic=Ic_intenzita(DATA,Zdroj,Ai,fii,omi); % vypocet komplexni intenzity (volani funkce)
end

    KRESLI_IcXY_intenzita(ZZdroj,DATA,Ic,fig1); % zobrazeni komplexni inenzity (volani funkce), graf

% okamzita intenzita, akusticky tlak, akusticka rychlost
    PA11=[]; VA11=[]; % vytvoreni promennych pro ukladani hodnot v danyh casech vypoctu
    p=1; % vytvoreni promenne poradi subplotu

for time=tmax/8:tmax/8:tmax % frekvence vypoctu

    It=It_intenzita(DATA,Zdroj,Ai,fii,omi,time); % vypocet okamzite intenzity (volani funkce)

    KRESLI_ItXY_intenzita_sub(DATA,It,Zdroj,p,fig4); % zobrazeni okamzite inenzity (volani funkce), graf

    [P,V]=p_v_tlak_rychlost(DATA,Zdroj,Ai,fii,omi,time); % vypocet tlaku a rychlosti (volani funkce)
    PA11=[PA11 P(:,2)]; % zapis hodnot tlaku v danyh casech
    VA11=[VA11 V(:,2)]; % zapis hodnot rychlosti v danyh casech

    p=p+1; % nacistani poradi subplotu
end

eval(['print -dtiff -r800 obr4_' num2str(ZZdroj(:,1)) '_'... % prikaz k ulozeni grafu
      num2str(ZZdroj(:,2)) '_' num2str(ZZdroj(:,3)) ])

for time=tmax/8:tmax/8:tmax % frekvence vypoctu

    It=It_intenzita(DATA,Zdroj,Ai,fii,omi,time); % vypocet okamzite intenzity (volani funkce)

    KRESLI_ItXY_intenzita(DATA,It,Zdroj,BP,fig5); % zobrazeni okamzite inenzity (volani funkce), graf

    fig5=fig5+1; % nacistani poradi subplotu

eval(['print -dtiff -r800 obr5_' num2str(ZZdroj(:,1)) '_'... % prikaz k ulozeni grafu

```

```

    num2str(ZZdroj(:,2)) '_' num2str(ZZdroj(:,3)) '_' num2str(fig5-5) ])
end

KRESLI_tlak(DATA,PA11,fig13); % zobrazeni hladin akustickeho tlaku (volani funkce), graf

eval(['print -dtiff -r800 Obr6_' num2str(ZZdroj(:,1)) '_'... % prikaz k ulozeni grafu
      num2str(ZZdroj(:,2)) '_' num2str(ZZdroj(:,3)) ])

KRESLI_rychlost_3D(DATA,VA11,fig14); % zobrazeni efektivni akusticke rychlosti (volani funkce), 3D graf

eval(['print -dtiff -r800 Obr7_' num2str(ZZdroj(:,1)) '_'... % prikazy k ulozeni grafu
      num2str(ZZdroj(:,2)) '_' num2str(ZZdroj(:,3)) ])

KRESLI_rychlost_2D(DATA,VA11,fig15); % zobrazeni vektoru akusticke rychlosti (volani funkce), 2D graf

eval(['print -dtiff -r800 Obr8_' num2str(ZZdroj(:,1)) '_'... % prikazy k ulozeni grafu
      num2str(ZZdroj(:,2)) '_' num2str(ZZdroj(:,3)) ])

pause % pozastaveni programu

close all % ukonceni programu

```

VÝPOČET KOMPLEXNÍ INTENZITY

```

%<<Ic_intenzita>>

function [Ic] = Ic_intenzita(uzel,Zdroj,Ai,fii,omi)

    for bod=1:length(uzel(:,1)) % cyklus pres vsechny uzly site
        if sum(ismember(bod,Zdroj))>0, continue, end % vynechani zdroju

%cast nezohlednujici prekryvani vice zdroju
        I1=0; % vytvoreni promenne
        for i=1:length(Zdroj) % cyklus pres zdroje

```

```

dx=uze1(bod,2)-uze1(Zdroj(i),2); % x-vzdalenost uzlu od zdroje
dy=uze1(bod,3)-uze1(Zdroj(i),3); % y-vzdalenost uzlu od zdroje
dz=uze1(bod,4)-uze1(Zdroj(i),4); % z-vzdalenost uzlu od zdroje
r=[dx;dy;dz]; % polohovy vektor od zdroje do uzlu
R=sqrt(dx^2+dy^2+dz^2); % velikost polohoveho vektoru

k=omi(i)/uze1(i,7); % vlnove cislo k=omi/c0

ksi1=-1/(2*i*uze1(i,6)*omi(i))*Ai(i)^2/R^4*r*(1-1i*k*R); % komplexni intenzita, prvni cast
I1=I1+ksi1; % ulozeni hodnot prvni casti komplexni intenzity v jednotlivych uzlech
end

%cast zohlednujici prekryvani vice zdroju
I2=0; % vytvoreni promenne
t=0;
for N=1:length(Zdroj) % cyklus pres zdroje, prvni
    dxN=uze1(bod,2)-uze1(Zdroj(N),2); % x-vzdalenost uzlu od zdroje
    dyN=uze1(bod,3)-uze1(Zdroj(N),3); % y-vzdalenost uzlu od zdroje
    dzN=uze1(bod,4)-uze1(Zdroj(N),4); % z-vzdalenost uzlu od zdroje
    RN=sqrt(dxN^2+dyN^2+dzN^2); % velikost polohoveho vektoru

    kN=omi(N)/uze1(N,7); % vlnove cislo k=omi/c0
    betaN=omi(N)*t-kN*RN+fii(N); % uhel od roviny XY v danem case

for M=1:length(Zdroj) % cyklus pres zdroje, druhy
    if M==N,continue,end
    dxM=uze1(bod,2)-uze1(Zdroj(M),2); % x-vzdalenost uzlu od zdroje
    dyM=uze1(bod,3)-uze1(Zdroj(M),3); % y-vzdalenost uzlu od zdroje
    dzM=uze1(bod,4)-uze1(Zdroj(M),4); % z-vzdalenost uzlu od zdroje
    RM=sqrt(dxM^2+dyM^2+dzM^2); % velikost polohoveho vektoru
    rM=[dxM;dyM;dzM]; % polohovy vektor od zdroje do uzlu

    kM=omi(M)/uze1(M,7); % vlnove cislo k=omi/c0
    betaM=kM*RM-fii(M)-omi(M)*t; % uhel od roviny XY v danem case
end

```

```

ksi2=-Ai(N)*Ai(M)/(2*RN*RM^3*1i*omi(M)*uzel(M,6))*... % komplexni intenzita, druha cast
(1-1i*kM*RM)*rM*exp(1i*betaN)*exp(1i*betaM);
I2=I2+ksi2; % ulozeni hodnot prvni casti komplexni intenzity v jednotlivych uzlech
end
end
% vysledna hodnota
Ic(bod,:)= [bod (I1+I2)']; % ulozeni vyslednych hodnot komplexni intenzity v jednotlivych uzlech
end

```

VÝPOČET OKAMŽITÉ INTENZITY

```

%<<It_intenzita>>

function [It] = It_intenzita(uzel,Zdroj,Ai,fii,omi,time)

for bod=1:length(uzel(:,1)) % cyklus pres vsechny uzly site
if sum(ismember(bod,Zdroj))>0, continue, end % vynechani zdroju

%cast nezohlednujici prekryvani vice zdroju
I1=0; % vytvoreni promenne
for i=1:length(Zdroj) % cyklus pres zdroje
dx=uzel(bod,2)-uzel(Zdroj(i),2); % x-vzdalenost uzlu od zdroje
dy=uzel(bod,3)-uzel(Zdroj(i),3); % y-vzdalenost uzlu od zdroje
dz=uzel(bod,4)-uzel(Zdroj(i),4); % z-vzdalenost uzlu od zdroje
R=sqrt(dx^2+dy^2+dz^2); % velikost polohoveho vektoru
r=[dx;dy;dz]; % polohovy vektor od zdroje do uzlu

k=omi(i)/uzel(i,7); % vlnove cislo k=omi/c0
beta=k*R-fii(i)-omi(i)*time; % uhel od roviny XY v danem case

ksi1=(k*R*cos(beta)^2-cos(beta)*sin(beta))*Ai(i)^2/... % okamzita intenzita, prvni cast
(omi(i)*uzel(i,6)*R^4)*r;
I1=I1+ksi1; % ulozeni hodnot prvni casti okamzite intenzity v jednotlivych uzlech
end
end

```

```

end

%cast zohlednujici prekryvani vice zdroju
I2=0;

for N=1:length(Zdroj)
    dxN=uze1(bod,2)-uze1(Zdroj(N),2);
    dyN=uze1(bod,3)-uze1(Zdroj(N),3);
    dzN=uze1(bod,4)-uze1(Zdroj(N),4);
    RN=sqrt(dxN^2+dyN^2+dzN^2);

    kN=omi(N)/uze1(N,7);
    betaN=kN*RN-fii(N)-omi(N)*time;

    for M=1:length(Zdroj)
        if M==N,continue,end
        dxM=uze1(bod,2)-uze1(Zdroj(M),2);
        dyM=uze1(bod,3)-uze1(Zdroj(M),3);
        dzM=uze1(bod,4)-uze1(Zdroj(M),4);
        RM=sqrt(dxM^2+dyM^2+dzM^2);
        rM=[dxM;dyM;dzM];

        kM=omi(M)/uze1(M,7);
        betaM=kM*RM-fii(M)-omi(M)*time;

        ksi2=Ai(N)*Ai(M)/(RN*RM^3*omi(M)*uze1(M,6))*rM*...
            (kM*RM*cos(betaN)*cos(betaM)-sin(betaM)*cos(betaN));
        I2=I2+ksi2;
    end
end

% vysledna hodnota
It(bod,:)= [bod (I1+I2)'];
end

```

% vytvoreni promenne
 % cyklus pres zdroje, prvni
 % x-vzdalenost uzlu od zdroje
 % y-vzdalenost uzlu od zdroje
 % z-vzdalenost uzlu od zdroje
 % velikost polohoveho vektoru
 % vlnove cislo $k=omi/c0$
 % uhel od roviny XY v danem case
 % cyklus pres zdroje, prvni
 % x-vzdalenost uzlu od zdroje
 % y-vzdalenost uzlu od zdroje
 % z-vzdalenost uzlu od zdroje
 % velikost polohoveho vektoru
 % polohovy vektor od zdroje do uzl
 % vlnove cislo $k=omi/c0$
 % uhel od roviny XY v danem case
 % okamzita intenzita, druha cast
 % ulozeni hodnot prvni casti okamzite intenzity v jednotlivych uzlech
 % ulozeni vyslednych hodnot komplexni intenzity v jednotlivych uzlech

VÝPOČET TLAKU A RYCHLOSTI

```
%<<p_v_tlak_rychlost>>

function [P,V] = p_v_tlak_rychlost(uzel,Zdroj,Ai,fii,omi,time)

for bod=1:length(uzel(:,1))                                % cyklus pres vsechny uzly site
    if sum(ismember(bod,Zdroj))>0, continue, end         % vynechani zdroju

    p=0; v=0;                                             % vytvoreni promennych

    for i=1:length(Zdroj)                                  % cyklus pres zdroje
        dx=uzel(bod,2)-uzel(Zdroj(i),2);                % x-vzdalenost uzlu od zdroje
        dy=uzel(bod,3)-uzel(Zdroj(i),3);                % y-vzdalenost uzlu od zdroje
        dz=uzel(bod,4)-uzel(Zdroj(i),4);                % z-vzdalenost uzlu od zdroje
        r=[dx;dy;dz];                                     % polohovy vektor od zdroje do uzlu
        R=sqrt(dx^2+dy^2+dz^2);                           % velikost polohoveho vektoru

        k=omi(i)/uzel(i,7);                               % vlnove cislo k=omi/c0
        beta=k*R-fii(i)-omi(i)*time;                    % uhel od roviny XY v danem case

        p=p+Ai(i)/R*cos(beta);                           % akusticky tlak
        v=v+Ai(i)/(omi(i)*uzel(i,6)*R^3)*(k*R*cos(beta)-sin(beta))*r; % akusticka rychlost
    end

    P(bod,:)= [bod p];                                   % ulozeni hodnot tlaku v jednotlivych uzlech
    V(bod,:)= [bod v(1)+1i*v(2)];                        % ulozeni vektoru rychlosti v jednotlivych uzlech
end
```

GRAF KOMPLEXNÍ INTENZITY

```
%<<KRESLI_IcXY_intenzita>>
```



```

function [ ] = KRESLI_IcXY_intenzita(ZZdroj,uzel,Ic,fig1)

figure(fig1),clf % cilovy obrazek

Ixi=power(abs(real(Ic(:,2))),0.25).*sign(real(Ic(:,2))); % koncove x-souradnice vektoru okamzite akusticke intenzity
Iyi=power(abs(real(Ic(:,3))),0.25).*sign(real(Ic(:,3))); % koncove y-souradnice vektoru okamzite akusticke intenzity
% velikost vektoru je normovana umocnenim na 1/4
h=quiver(uzel(:,2),uzel(:,3),Ixi,Iyi); % zobrazeni vektoru aktivni akusticke intenzity
set(h,'color','r') % parametry grafu

title('Aktivní akustická intenzita I(r) [W/m^2]') % popisky
xlabel(' x [m]')
ylabel(' y [m]')
xlim ([-0.1 6.1]) % interval zobrazovanych x-souradnic
ylim ([-0.1 6.1]) % interval zobrazovanych y-souradnic

eval(['print -dtiff -r800 Obr1_' num2str(ZZdroj(:,1)) '_' ... % prikaz k ulozeni grafu
num2str(ZZdroj(:,2)) '_' num2str(ZZdroj(:,3)) ])

figure(fig1+1) % cilovy obrazek

Qxi=power(abs(imag(Ic(:,2))),0.25).*sign(imag(Ic(:,2))); % koncove x-souradnice vektoru okamzite akusticke intenzity
Qyi=power(abs(imag(Ic(:,3))),0.25).*sign(imag(Ic(:,3))); % koncove y-souradnice vektoru okamzite akusticke intenzity

m=quiver(uzel(:,2),uzel(:,3),Qxi,Qyi); % zobrazeni vektoru reaktivni akusticke intenzity
set(m,'color','b') % parametry grafu

title('Reaktivní akustická intenzita Q(r) [W/m^2]') % popisky
xlabel(' x [m]')
ylabel(' y [m]')
xlim ([-0.1 6.1]) % interval zobrazovanych x-souradnic
ylim ([-0.1 6.1]) % interval zobrazovanych y-souradnic

eval(['print -dtiff -r800 Obr2_' num2str(ZZdroj(:,1)) '_'... % prikaz k ulozeni grafu
num2str(ZZdroj(:,2)) '_' num2str(ZZdroj(:,3)) ])

```

```

figure(fig1+2) % cilovy obrazek
n=quiver(uzel(:,2),uzel(:,3),Ixi+Qxi,Iyi+Qyi); % zobrazeni vektoru komplexni akusticke intenzity
set(n, 'color', 'm') % paramtery grafu

title('komplexní akustická intenzita  $I_c(r)$  [w/m2]) % popisky
xlabel(' x [m]')
ylabel(' y [m]')
xlim ([-0.1 6.1]) % interval zobrazovanych x-souradnic
ylim ([-0.1 6.1]) % interval zobrazovanych y-souradnic

eval(['print -dtiff -r800 obr3_' num2str(Zdroj(:,1)) '_'... % prikaz k ulozeni grafu
num2str(Zdroj(:,2)) '_' num2str(Zdroj(:,3)) ])

```

GRAF OKAMŽITÉ INTENZITY

```

%<<KRESLI_ItXY_intenzita_sub>>

function [ ] = KRESLI_ItXY_intenzita_sub(uzel,It,Zdroj,p,fig4)

figure(fig4) % cilovy obrazek

Itxi=power(abs(It(:,2)),0.25).*sign(It(:,2)); % koncove x-souradnice vektoru okamzite akusticke intenzity
Ityi=power(abs(It(:,3)),0.25).*sign(It(:,3)); % koncove y-souradnice vektoru okamzite akusticke intenzity
% velikost vektoru je normovana umocnenim na 1/4

subplot(2,4,p);
quiver(uzel(:,2),uzel(:,3),Itxi,Ityi, 'Linewidth',0.5) % zobrazeni vektoru okamzite akusticke intenzity
hold on

for i=1:length(Zdroj) % cyklus pres vsechny zdroje
bod=Zdroj(i);
Z=plot(uzel(bod,2),uzel(bod,3), 'om'); % zvyrazneni polohy zdroju v grafu fialove
set(Z, 'Linewidth',0.5) % parametry zobrazeni zdroju
end

```

```

axis equal
xlim ([-0.1 6.1])
ylim ([-0.1 6.1])
xlabel(['\bf time' num2str(p) '/8' '\rm          x [m]'])
ylabel(' y [m]')

subplot(2,4,1);
title(['          '... % popisek obrazku
'okamžitá akustická intenzita It(r,t) [J/m^2]'])

```

```

%<<KRESLI_ItXY_intenzita>>

function [ ] = KRESLI_ItXY_intenzita(uzel,It,Zdroj,BP,fig5)

figure(fig5)

Itxi=power(abs(It(:,2)),0.25).*sign(It(:,2));
Ityi=power(abs(It(:,3)),0.25).*sign(It(:,3));

quiver(uzel(:,2),uzel(:,3),Itxi,Ityi, 'Linewidth',0.5)
hold on

for i=1:length(Zdroj)
    bod=Zdroj(i);
    Z=plot(uzel(bod,2),uzel(bod,3), 'om');
    set(Z, 'Linewidth',0.5)
end

axis equal
xlim([uzel(min(BP),2)-1 uzel(max(BP),2)+1])
ylim([uzel(min(BP),3)-1 uzel(max(BP),3)+1])
xlabel(['\bf time' num2str(fig5-4) '/8' '\rm          x [m]'])

```

```
ylabel(' y [m]')
title('Okamžitá akustická intenzita It(r,t) [J/m^2]')
```

GRAF HLADINY AKUSTICKÉHO TLAKU

```
%<<KRESLI_tlak>
```

```
function [xii,yii,zii,xi,yi,zi]= KRESLI_tlak(DATA,PA11,fig13)
```

```
figure(fig13)
```

```
% cílový obrázek
```

```
Pef=max(PA11(),[],2)./(2^(1/2));
```

```
% hodnota efektivního akustického tlaku v uzlech site
```

```
adr=find(Pef==0);
```

```
% ošetření uzlu zdroje, ve kterých je hodnota nulová
```

```
Pef(adr)=Pef(adr+1);
```

```
PefdB=20*log10(Pef/20e-6);
```

```
% hodnota hladiny akustického tlaku
```

```
Ndata=[DATA(:,2),DATA(:,3),PefdB];
```

```
% uložení hodnot hladiny akustického tlaku v jednotlivých uzlech
```

```
xii=unique(Ndata(:,1));
```

```
% x-souradnice uzlu
```

```
yii=unique(Ndata(:,2));
```

```
% y-souradnice uzlu
```

```
zii=[];
```

```
% vytvoření proměnné pro z-souradnici zastupující v tomto grafu PefdB
```

```
for k1=1:length(xii)
```

```
% zápis hodnot PefdB v závislosti na souřadnicích xy do souřadnice z
```

```
for k2=1:length(yii)
```

```
adr=find(xii(k1,1)==Ndata(:,1) & yii(k2,1)==Ndata(:,2));
```

```
zii(k1,k2)=Ndata(adr,3);
```

```
end
```

```
end
```

```
pocet=50;
```

```
% vytvoření formy grafu
```

```
[xi,yi]=meshgrid(xii(1):(xii(end)-xii(1))/pocet:xii(end),...
```

```
yii(1):(yii(end)-yii(1))/pocet:yii(end));
```

```
zi=griddata(xii,yii,zii',xi,yi);
```

```

m1=max(max(zi)); % maximalni zobrazovana hodnota hladiny akustickeho tlaku
m2=min(min(zi)); % minimalni zobrazovana hodnota hladiny akustickeho tlaku
surf(xi,yi,zi); % zobrazeni hodnot hladiny akustickeho tlaku, 3D graf
colorbar; % legenda

caxis(gca); % parametry os
title(['Hladina akustického tlaku L <'... % popisky
num2str(m2,3) ' ; ' num2str(m1,3) '> [dB]']);
xlabel('x [m]')
ylabel('y [m]')
zlabel('L [dB]')

```

GRAF AKUSTICKÉ RYHLOSTI

```
%<<KRESLI_rychlost_3D>>
```

```
function []= KRESLI_rychlost_3D(DATA,VA11,fig14)
```

```

figure(fig14) % cilovy obrazek

maxVA11=max(VA11(), [],2); % amplituda akusticke rychlosti v uzlech site
Vmodul=(real(maxVA11).^2+imag(maxVA11).^2).^(1/2); % velikost vektoru akusticke rychlosti v uzlech site
Vef=Vmodul./(2^(1/2)); % hodnota efektivni akusticke rychlosti v uzlech site

adr=find(Vef==0); % osetreni uzlu zdroju, ve kterych je hodnota nulova
Vef(adr)=Vef(adr+1);

Ndata=[DATA(:,2),DATA(:,3),real(Vef)]; % ulozeni hodnot efektivni akusticke rychlosti v jednotlivych uzlech
xii=unique(Ndata(:,1)); % x-souradnice uzlu
yii=unique(Ndata(:,2)); % y-souradnice uzlu
zii=[]; % vytvoreni promenne pro z-souradnici zastupujici v tomto grafu Vef

for k1=1:length(xii) % zapis hodnot PefdB v zavislosti na souradnicich xy do souradnice z

```

```

for k2=1:length(yii)
    adr=find(xii(k1,1)==Ndata(:,1) & yii(k2,1)==Ndata(:,2));
    zii(k1,k2)=Ndata(adr,3);
end
end

pocet=50; % vytvoreni formy grafu
[xi,yi]=meshgrid(xii(1):(xii(end)-xii(1))/pocet:xii(end),...
    yii(1):(yii(end)-yii(1))/pocet:yii(end));

zi=griddata(xii,yii,zii',xi,yi);

m1=max(max(zi)); % maximalni zobrazovana hodnota efektivni akusticke rychlosti
m2=min(min(zi)); % minimalni zobrazovana hodnota efektivni akusticke rychlosti
surf(xi,yi,zi); % zobrazeni hodnot efektivni akusticke rychlosti, 3D graf
colorbar; % legenda

caxis(gca); % parametry os
title(['Akustická rychlost v(r) <'... % popisky
    num2str(m2,3) ' ; ' num2str(m1,3) '> [m/s]' ]);

xlabel('x [m]')
ylabel('y [m]')
zlabel('v [m/s]')

```

```
%<<KRESLI_rychlost_2D>>
```

```
function []= KRESLI_rychlost_2D(DATA,VA11,fig15)
```

```

figure(fig15) % cilovy obrazek

Vef=max(VA11(), [], 2)./(2^(1/2)); % hodnota efektivni akusticke rychlosti v uzlech site
Vxi=power(abs(real(Vef(:))),0.25).*sign(real(Vef(:))); % koncove x-souradnice vektoru akusticke rychlosti
Vyi=power(abs(imag(Vef(:))),0.25).*sign(imag(Vef(:))); % koncove y-souradnice vektoru akusticke rychlosti

```

```

q=quiver(DATA(:,2),DATA(:,3),vxi,vyi);
set(q,'color','g')

xlim ([-0.1 6.1])
ylim ([-0.1 6.1])
title('Akustická rychlost v(r) [m/s]')
xlabel(' x [m]')
ylabel(' y [m]')

% velikost vektoru je normovana umocnenim na 1/4
% zobrazeni vektoru akusticke rychlosti, graf
% parametry grafu

% interval zobrazovanych x-souradnic
% interval zobrazovanych y-souradnic
% popisky

```

DEFINICE BLÍZKÉHO POLE

```

%<<BP>>

function [BP] = BP(UZEL,zdroj,fi)

BP=[];
DATA=[];

% vytvoreni promennych

for i=1:length(zdroj)
    % cyklus pres zdroje
    c0=UZEL(zdroj(i),7);
    % rychlost v uzlu zdroje
    r=c0/2/pi/fi(i);
    % rameno vymezejici hranici blizkeho pole zdroje

    for ii=1:length(UZEL(:,1))
        % cyklus pres vsechny uzly site
        if UZEL((ii),2)>(-r+UZEL(zdroj(i),2))...
            &UZEL((ii),2)<(r+UZEL(zdroj(i),2))...
            &UZEL((ii),3)>(-r+UZEL(zdroj(i),3))...
            &UZEL((ii),3)<(r+UZEL(zdroj(i),3))
            % nalezeni uzlu lezicich v blizkem poli zdroje

            BP=[BP ii];
        end
    end
end

% ulozeni souradnic uzlu lezicich v blizkem poli zdroje
end

```