Figure 313

Table of Contents

Check interpretation of coefficients in anisotropic CRS formula

Author: Abakumov Ivan **Publication date**: 31st August 2016

Define working folder, add links to Library and SeisLab

```
clear; close all; clc;
mlibfolder = '/home/zmaw/u250128/Desktop/MLIB';
path(path, mlibfolder);
addmypath;
```

Introduction

```
% Note:
% Accuracy of traveltime 10e-12
% Accuracy of attributes 10e-10

    model = 64:64
    acquisition = 4;


model =

    64
```

Get model parameters

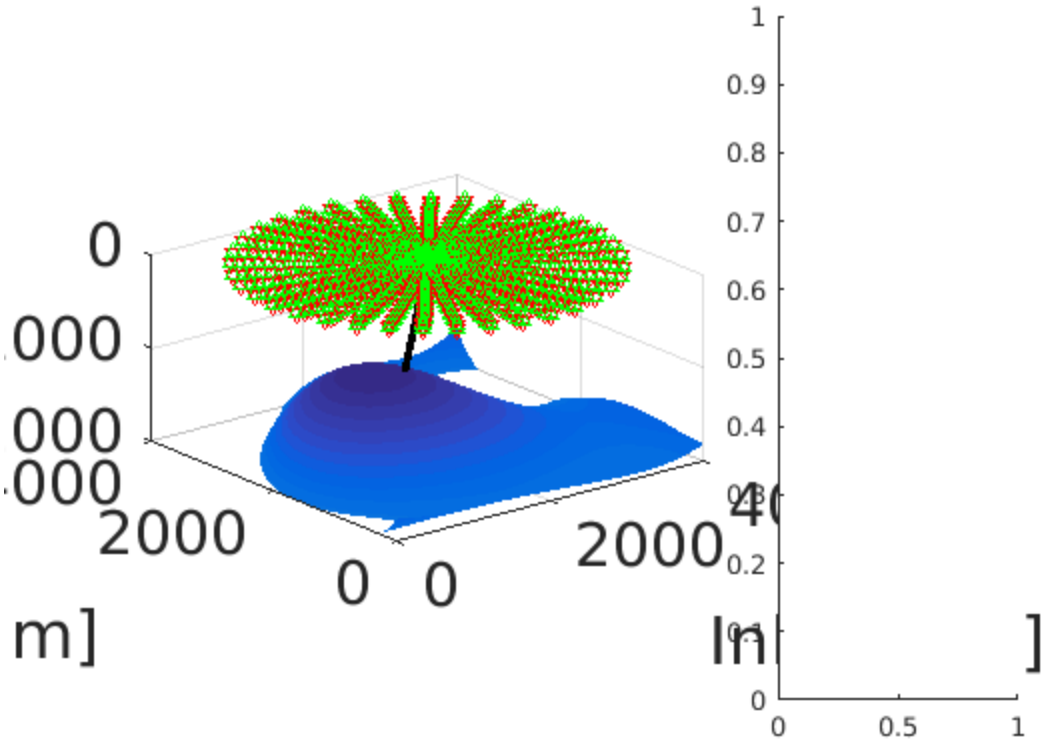
```
    Get_model_parameters;
```

Get acquisition geometry

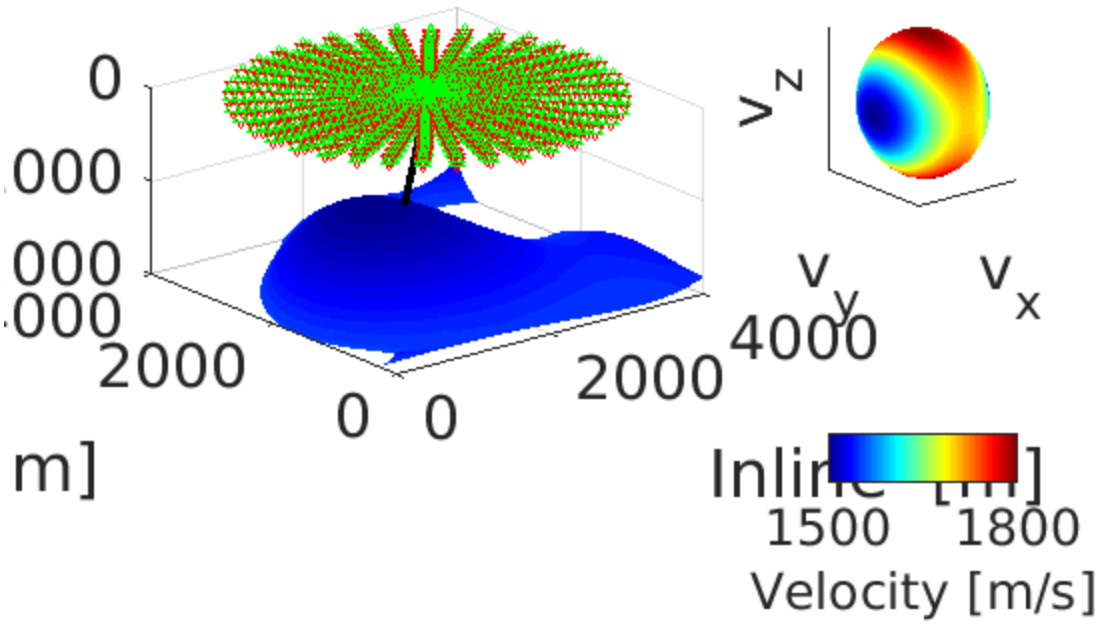
```
Get_model_acquisition_geometry;
```

Plot acquisition geometry

```
X0 = [ 2000; 2000; 0];
[~, xref ]  = Get_model_exact_traveltime(X0, X0, 64);

figure(1)
subplot(1,3,[1 2])
[XX, YY] = meshgrid(G.xx, G.yy);
[ZZ, ind] = Get_model_surface(XX,YY,model);
h = surf(XX,YY,ZZ);

 set(h,'FaceColor','interp','EdgeColor','none','DiffuseStrength',.8)
hold on
plot3(Xs(1,:),Xs(2,:),Xs(3,:), 'rv');
plot3(Xg(1,:),Xg(2,:),Xg(3,:), 'g^');
plot3(X0(1),X0(2),X0(3), 'b*');

 plot3(linspace(X0(1),xref(1),200),linspace(X0(2),xref(2),200),linspace(X0(3),xref
black', 'LineWidth',3);
axis([G.x0, G.mx, G.y0, G.my, G.z0, G.mz]);
xlabel('Inline  [m]', 'Fontsize', 24);
ylabel('Crossline [m]', 'Fontsize', 24);
zlabel('Depth  [m]', 'Fontsize', 24);
set(gca,'xtick',[0 2000 4000])
set(gca,'ytick',[0 2000 4000])
set(gca,'ztick',[0 1000 2000])
view(-39,19);
set(gca, 'ZDir', 'reverse')
set(gca,'FontSize',24)
axis('equal')
axis([0 4000 0 4000 0 2000])
subplot(1,3,3)
```




```
   [x, y, z] = ellipsoid(0,0,0,sqrt(A11),sqrt(A22),sqrt(A33),30);
   subplot(1,3,3)
   c = sqrt(x.^2 + y.^2 + z.^2);
   surf(x, y, z,c)
   xlabel('v_x', 'Fontsize', 24);
   ylabel('v_y', 'Fontsize', 24);
   zlabel('v_z', 'Fontsize', 24);
   c = colorbar('southoutside', 'Ticks', [1500,1800], 'Fontsize',
20);
   c.Label.String = 'Velocity [m/s]';
   set(gca,'xtick',[])
   set(gca,'ytick',[])
   set(gca,'ztick',[])
   axis equal
   view(-39,19);
   %colormap(makeColorMap([0 0 1],[1 1 1],[1 0 0],100));
   colormap('jet')
   shading interp
```



Find stacking parameters

w3, M3 and N3 are stacking parameters. They could be transformed to wavefield attributes

- V - phase velocity
- alpha, beta - phase angles
- theta, phi - group angles

Case 1. Isotropic homogeneous overburden:

```
model = 61;
[ t0, w3, M3, N3 ] = Get_model_stacking_parameters( X0, model );

V0 = 2/norm(w3);

[ alpha, beta, KNIP, KN ] = my_A3P(V0, w3, M3, N3);

KNIP = KNIP
KN = KN
clear alpha beta KNIP KN v0 w3 M3 N3 model
```

KNIP =

```
   1.0e-03 *

    0.8573    0.0000   -0.0000
   -0.0000    0.8573    0.0000
   -0.0000    0.0000    0.0000


KN =

   1.0e-03 *

    0.3485   -0.0077    0.0000
   -0.0077    0.3873    0.0000
   -0.0000    0.0000    0.0000
```

Case 2. Isotropic inhomogeneous overburden:

```
    model = 63;

    CRS_param = MLD([mlibfolder '/CRS/models/model_'
 num2str(model) '_CRS_param.mat']);

    X0 = CRS_param.x0;
    t0 = CRS_param.t0;
    v0 = CRS_param.v0;
    w3 = CRS_param.w;
    M3 = CRS_param.M;
    N3 = CRS_param.N;

    V0 = 2/norm(w3);

    [ alpha, beta, KNIP, KN ] = my_A3P(V0, w3, M3, N3);

    KNIP = KNIP
    KN = KN


KNIP =

   1.0e-03 *

    0.7597   -0.0000   -0.0000
   -0.0000    0.7676   -0.0000
   -0.0000   -0.0000   -0.0000


KN =

   1.0e-03 *

    0.2660   -0.0060    0.0000
```

```
   -0.0060    0.3176    0.0000
   -0.0000   -0.0000   -0.0000
```

Case 3. Anisotropic homogeneous overburden:

```
    model = 64;
    [ t0, w3, M3, N3 ] = Get_model_stacking_parameters( X0, model );

    V0 = 2/norm(w3);

    [ alpha, beta, KNIP, KN ] = my_A3P(V0, w3, M3, N3);

    KNIP = KNIP
    KN = KN


KNIP =

    0.0011    0.0001   -0.0001
    0.0001    0.0010   -0.0000
   -0.0001   -0.0000    0.0000


KN =

   1.0e-03 *

    0.3637    0.0084   -0.0244
    0.0084    0.4106   -0.0093
   -0.0244   -0.0093    0.0018


    theta = atan( sqrt((A11*cos(beta))^2 + (A22*sin(beta))^2)/
A33*tan(alpha));
    phi = atan(A22/A11*tan(beta));

    v = 1/sqrt((sin(theta)*cos(phi))^2/A11 + (sin(theta)*sin(phi))^2/
A22 + (cos(theta))^2/A33);

    vacq = v*([sin(theta)*cos(phi); sin(theta)*sin(phi); cos(theta)]);

    R = Get_Rmatrix_3x3(alpha, beta);

    vwoc = R'*vacq

    KNIP*vwoc;
    KN*vwoc
    KNIP'*vwoc
    KN'*vwoc

    w = w3(1:2, 1);
    N = N3(1:2, 1:2);
    M = M3(1:2, 1:2);
```

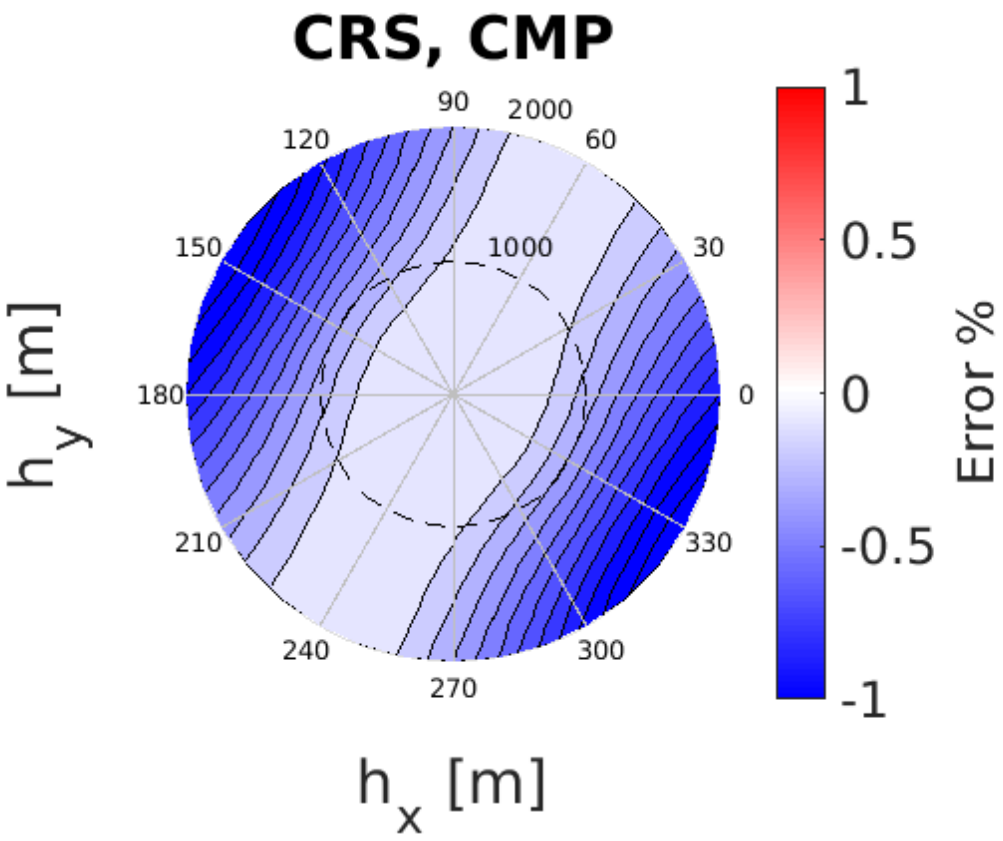
```
vwoc =

   1.0e+03 *

    0.1188
    0.0378
    1.7843


ans =

   1.0e-04 *

    0.3991
   -0.7623
   -0.1058


ans =

   1.0e-05 *

    0.0163
   -0.1028
   -0.0501


ans =

   1.0e-05 *

   -0.2553
    0.0257
   -0.9377
```

CRS, n-CRS and i-CRS errors for CMP and ZO acquisition

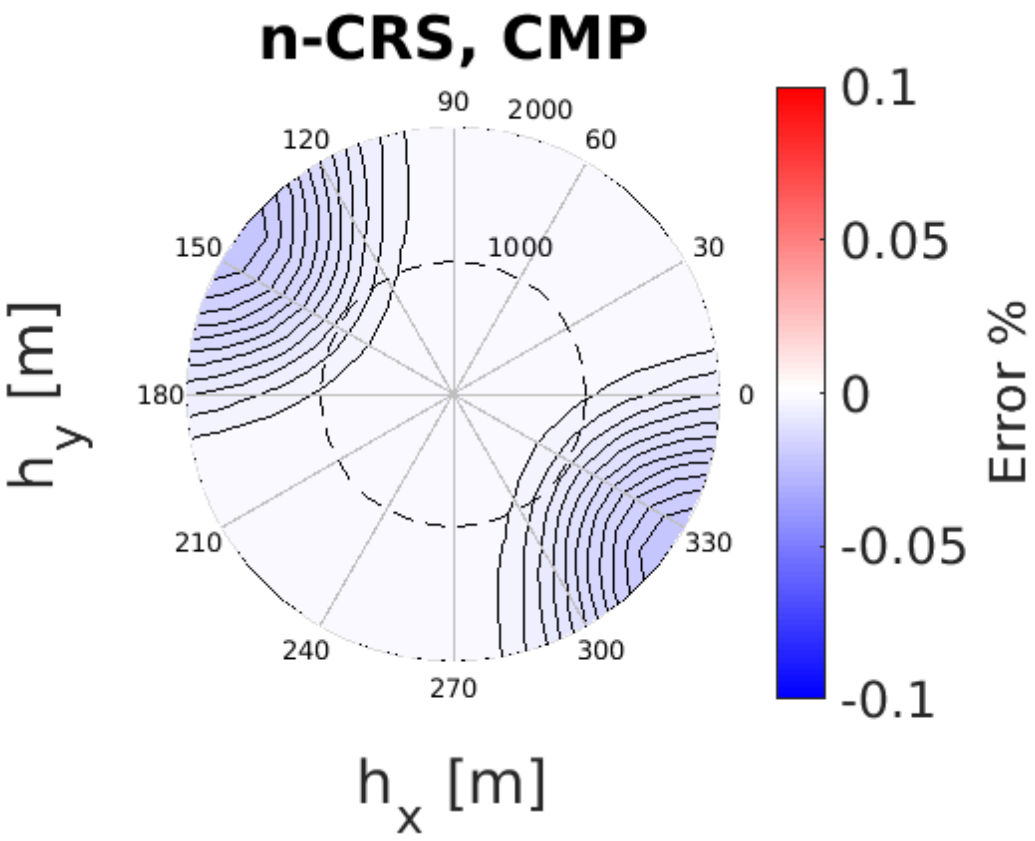
```
    % Load CMP acquisition
    acquisition = 4;
    Get_model_acquisition_geometry;

    % Compute exact traveltimes

    %tti_ex_CMP = Get_model_exact_traveltime(Xs, Xg, model);
    %save([mlibfolder '/CRS/models/model_' num2str(model)
'_traveltimes_for_acq_' num2str(acquisition) '.mat'], 'tti_ex_CMP');
    tti_ex_CMP = MLD([mlibfolder '/CRS/models/model_'
num2str(model) '_traveltimes_for_acq_' num2str(acquisition) '.mat']);
```

```
    % Compute traveltime approximations

    HH = (Xg(1:2, :) - Xs(1:2,:))/2;
    MM = (Xg(1:2, :) + Xs(1:2,:))/2;
    MM(1,:) = MM(1,:) - X0(1);
    MM(2,:) = MM(2,:) - X0(2);
    of_CMP = offset;
    az_CMP = azimuth;
    [OF, AZ] = meshgrid(of_CMP,az_CMP);
    XX = OF.*cos(AZ);
    YY = OF.*sin(AZ);

    tti_crs_CMP = Get_traveltime_3D_CRS (MM, HH, t0, w, M, N);
    tti_ncrs_CMP = Get_traveltime_3D_nCRS(MM, HH, t0, w, M, N);
    tti_icrs_CMP = Get_traveltime_3D_iCRS_el_LIA(MM, HH, t0, V0, w, M,
 N, 3);

    dt_crs_CMP    = reshape((tti_crs_CMP  -tti_ex_CMP)./
tti_ex_CMP*100,length(az_CMP),length(of_CMP));
    dt_ncrs_CMP   = reshape((tti_ncrs_CMP -tti_ex_CMP)./
tti_ex_CMP*100,length(az_CMP),length(of_CMP));
    dt_icrs_CMP   = reshape((tti_icrs_CMP(3,:) -tti_ex_CMP)./
tti_ex_CMP*100,length(az_CMP),length(of_CMP));

    % Load ZO acquisition
    acquisition = 5;
    Get_model_acquisition_geometry;

    % Compute exact traveltimes

    %tti_ex_ZO = Get_model_exact_traveltime(Xs, Xg, model);
    %save([mlibfolder '/CRS/models/model_' num2str(model)
 '_traveltimes_for_acq_' num2str(acquisition) '.mat'], 'tti_ex_ZO');
    tti_ex_ZO = MLD([mlibfolder '/CRS/models/model_'
 num2str(model) '_traveltimes_for_acq_' num2str(acquisition) '.mat']);

    % Compute traveltime approximations

    HH = (Xg(1:2, :) - Xs(1:2,:))/2;
    MM = (Xg(1:2, :) + Xs(1:2,:))/2;
    MM(1,:) = MM(1,:) - X0(1);
    MM(2,:) = MM(2,:) - X0(2);
    of_ZO = offset;
    az_ZO = azimuth;

    tti_crs_ZO = Get_traveltime_3D_CRS (MM, HH, t0, w, M, N);
    tti_ncrs_ZO = Get_traveltime_3D_nCRS(MM, HH, t0, w, M, N);
    tti_icrs_ZO = Get_traveltime_3D_iCRS_el_LIA(MM, HH, t0, V0, w, M,
 N, 3);

    dt_crs_ZO    = reshape((tti_crs_ZO  -tti_ex_ZO)./
tti_ex_ZO*100,length(az_ZO),length(of_ZO));
    dt_ncrs_ZO   = reshape((tti_ncrs_ZO -tti_ex_ZO)./
tti_ex_ZO*100,length(az_ZO),length(of_ZO));
```

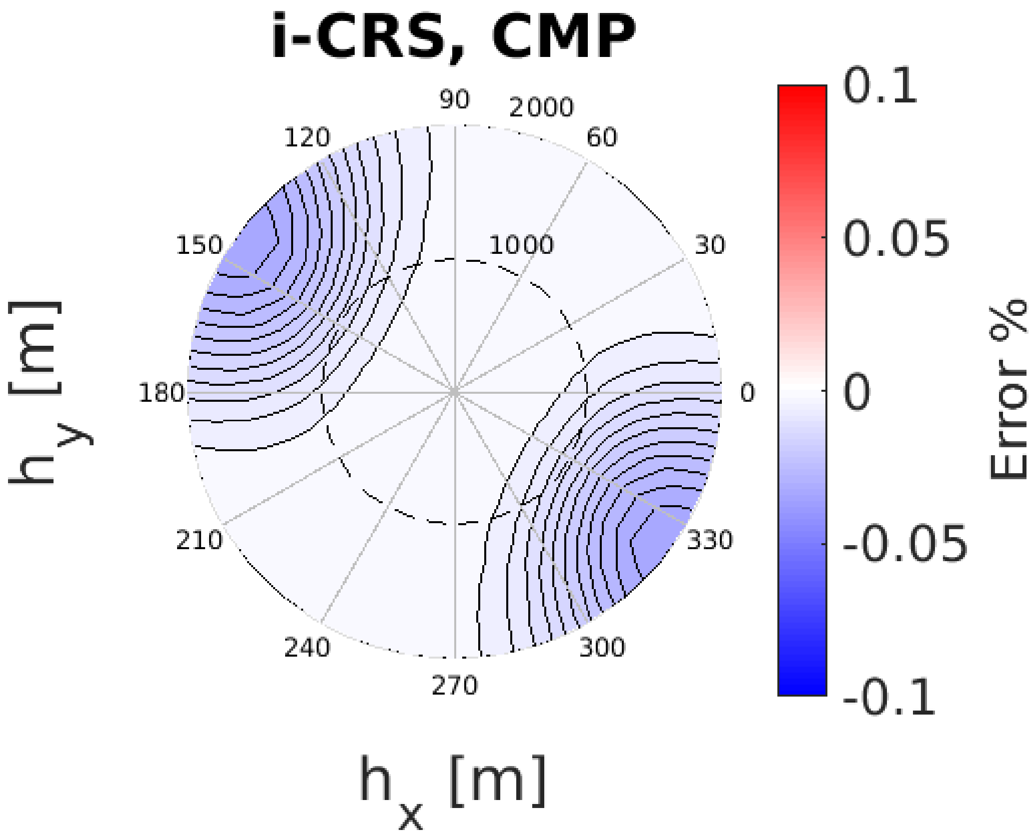
```
    dt_icrs_ZO   = reshape((tti_icrs_ZO(3,:) -tti_ex_ZO)./
tti_ex_ZO*100,length(az_ZO),length(of_ZO));


    [OF, AZ] = meshgrid(of_CMP,az_CMP);
    XX = OF.*cos(AZ);
    YY = OF.*sin(AZ);

    figure(21)
    %subplot(2,3,1)
    h = polar(azimuth,2000*ones(size(azimuth)), '-black');
    hold on
    contourf(XX,YY,dt_crs_CMP,11);
    for a = 0:pi/6:pi
        hold on
        x1 = (-2000:10:2000)*cos(a);
        x2 = (-2000:10:2000)*sin(a);
        plot(x1,x2,'Color',[0.75 0.75 0.75])
    end
    polar(azimuth,1000*ones(size(azimuth)),'--black');
    text(250,1100,'1000')
    c = colorbar('Ticks', [-1,-0.5 0, 0.5, 1], 'Fontsize', 20);
    c.Label.String = 'Error %';
    caxis([-1 1])
    colormap(makeColorMap([0 0 1],[1 1 1],[1 0 0],100));
    xlabel('h_x [m]','Fontsize', 24)
    ylabel('h_y [m]', 'Fontsize', 24)
    title('CRS, CMP', 'Fontsize', 24)
```




```
figure(22)
%subplot(2,3,2)
h = polar(azimuth,2000*ones(size(azimuth)), '-black');
hold on
contourf(XX,YY,dt_ncrs_CMP,11);
for a = 0:pi/6:pi
    hold on
    x1 = (-2000:10:2000)*cos(a);
    x2 = (-2000:10:2000)*sin(a);
    plot(x1,x2,'Color',[0.75 0.75 0.75])
end
polar(azimuth,1000*ones(size(azimuth)),'--black');
text(250,1100,'1000')
c = colorbar('Ticks', [-0.1,-0.05 0, 0.05, 0.1], 'Fontsize', 20);
c.Label.String = 'Error %';
caxis([-0.1 0.1])
colormap(makeColorMap([0 0 1],[1 1 1],[1 0 0],100));
xlabel('h_x [m]','Fontsize', 24)
ylabel('h_y [m]', 'Fontsize', 24)
title('n-CRS, CMP', 'Fontsize', 24)
```



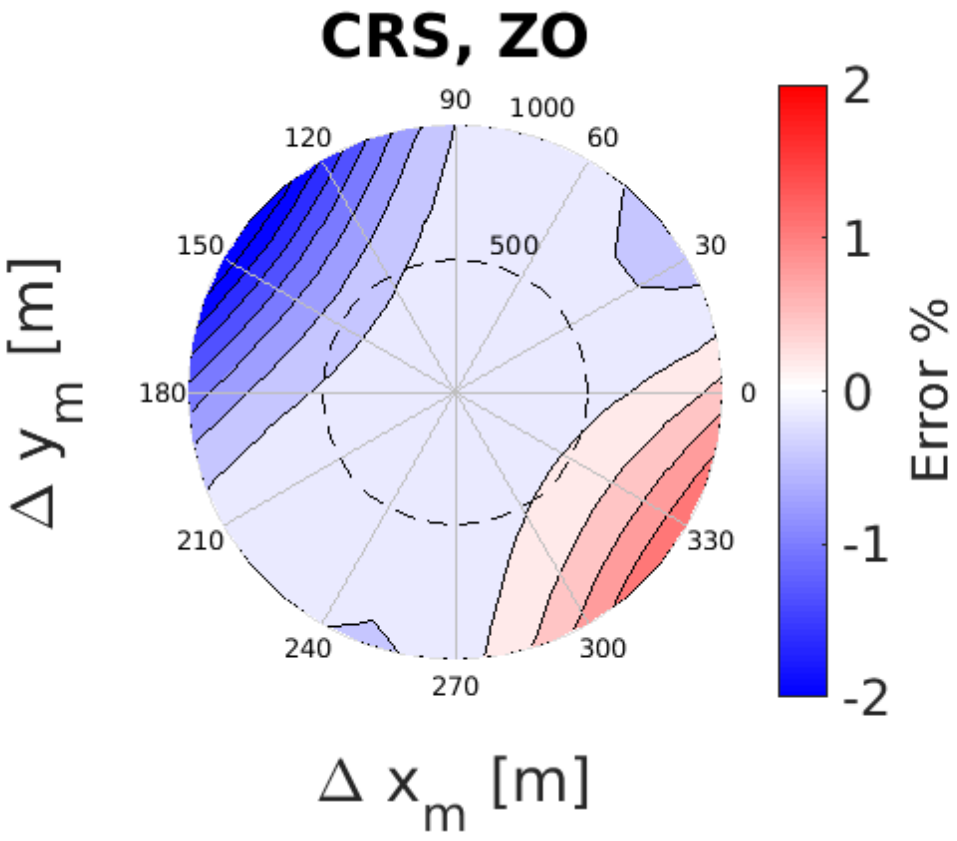

```
figure(23)
%subplot(2,3,3)
h = polar(azimuth,2000*ones(size(azimuth)), '-black');
hold on
contourf(XX,YY,dt_icrs_CMP,11);
for a = 0:pi/6:pi
    hold on
    x1 = (-2000:10:2000)*cos(a);
    x2 = (-2000:10:2000)*sin(a);
    plot(x1,x2,'Color',[0.75 0.75 0.75])
end
polar(azimuth,1000*ones(size(azimuth)),'--black');
text(250,1100,'1000')
c = colorbar('Ticks', [-0.1,-0.05 0, 0.05, 0.1], 'Fontsize', 20);
c.Label.String = 'Error %';
caxis([-0.1 0.1])
colormap(makeColorMap([0 0 1],[1 1 1],[1 0 0],100));
xlabel('h_x [m]','Fontsize', 24)
ylabel('h_y [m]', 'Fontsize', 24)
title('i-CRS, CMP', 'Fontsize', 24)
```



%%%

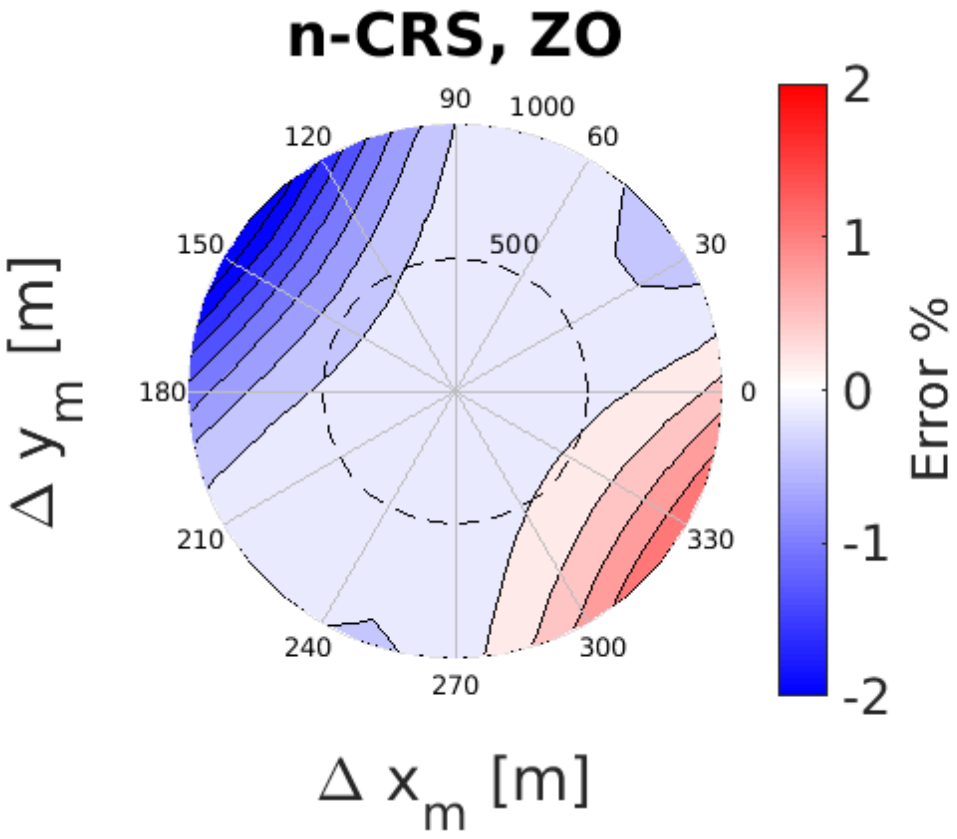
```
[OF, AZ] = meshgrid(of_ZO,az_ZO);
XX = OF.*cos(AZ);
YY = OF.*sin(AZ);

figure(24)
%subplot(2,3,4)
h = polar(azimuth,1000*ones(size(azimuth)), '-black');
hold on
contourf(XX,YY,dt_crs_ZO,11);
%
for a = 0:pi/6:pi
    hold on
    x1 = (-1000:10:1000)*cos(a);
    x2 = (-1000:10:1000)*sin(a);
    plot(x1,x2,'Color',[0.75 0.75 0.75])
end
polar(azimuth,500*ones(size(azimuth)),'--black');
text(125,550,'500')
c = colorbar('Ticks', [-2,-1 0, 1, 2], 'Fontsize', 20);
```

```
c.Label.String = 'Error %';
caxis([-2 2])
colormap(makeColorMap([0 0 1],[1 1 1],[1 0 0],100));
xlabel('\Delta x_m [m]','Fontsize', 24)
ylabel('\Delta y_m [m]', 'Fontsize', 24)
title('CRS, ZO', 'Fontsize', 24)
```

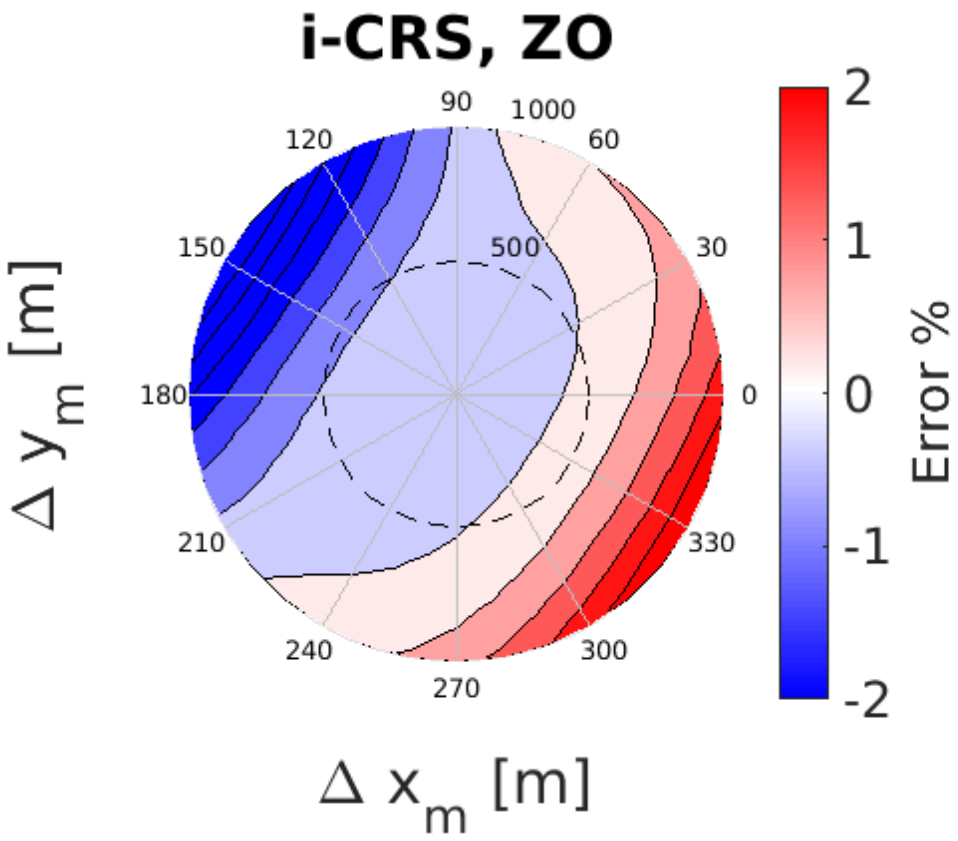



```
figure(25)
%subplot(2,3,5)
h = polar(azimuth,1000*ones(size(azimuth)), '-black');
hold on
contourf(XX,YY,dt_ncrs_ZO,11);
%
for a = 0:pi/6:pi
    hold on
    x1 = (-1000:10:1000)*cos(a);
    x2 = (-1000:10:1000)*sin(a);
    plot(x1,x2,'Color',[0.75 0.75 0.75])
end
polar(azimuth,500*ones(size(azimuth)),'--black');
text(125,550,'500')
c = colorbar('Ticks', [-2,-1 0, 1, 2], 'Fontsize', 20);
c.Label.String = 'Error %';
caxis([-2 2])
colormap(makeColorMap([0 0 1],[1 1 1],[1 0 0],100));
xlabel('\Delta x_m [m]','Fontsize', 24)
```

```
ylabel('\Delta y_m [m]', 'Fontsize', 24)
title('n-CRS, ZO', 'Fontsize', 24)
```



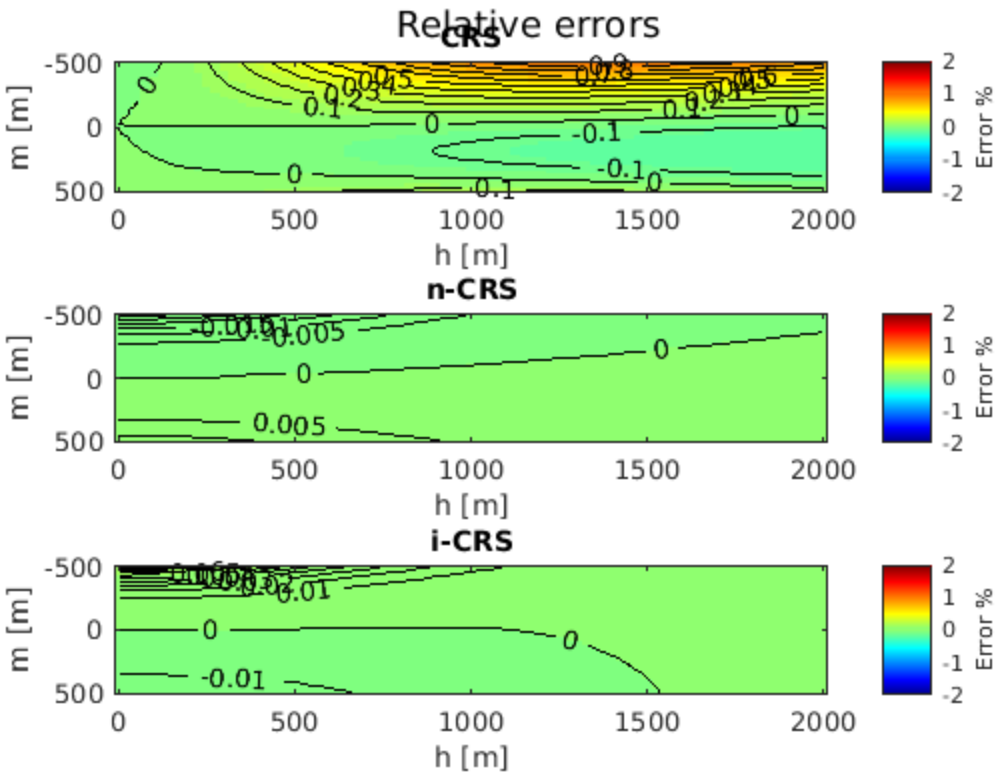

```
figure(26)
%subplot(2,3,6)
h = polar(azimuth,1000*ones(size(azimuth)), '-black');
hold on
contourf(XX,YY,dt_icrs_ZO,11);
%
for a = 0:pi/6:pi
    hold on
    x1 = (-1000:10:1000)*cos(a);
    x2 = (-1000:10:1000)*sin(a);
    plot(x1,x2,'Color',[0.75 0.75 0.75])
end
polar(azimuth,500*ones(size(azimuth)),'--black');
text(125,550,'500')
c = colorbar('Ticks', [-2,-1 0, 1, 2], 'Fontsize', 20);
c.Label.String = 'Error %';
caxis([-2 2])
colormap(makeColorMap([0 0 1],[1 1 1],[1 0 0],100));
xlabel('\Delta x_m [m]','Fontsize', 24)
ylabel('\Delta y_m [m]', 'Fontsize', 24)
title('i-CRS, ZO', 'Fontsize', 24)
```



CRS, n-CRS and i-CRS errors for 2D line

```
% Load CMP acquisition
acquisition = 2;
Get_model_acquisition_geometry;

% Compute exact traveltimes

%tti_ex_2D = Get_model_exact_traveltime(Xs, Xg, model);
%save([mlibfolder '/CRS/models/model_' num2str(model)
 '_traveltimes_for_acq_' num2str(acquisition) '.mat'], 'tti_ex_2D');
tti_ex_2D = MLD([mlibfolder '/CRS/models/model_'
 num2str(model) '_traveltimes_for_acq_' num2str(acquisition) '.mat']);

% Compute traveltime approximations

HH = (Xg(1:2, :) - Xs(1:2,:))/2;
MM = (Xg(1:2, :) + Xs(1:2,:))/2;
MM(1,:) = MM(1,:) - X0(1);
MM(2,:) = MM(2,:) - X0(2);

tti_crs_2D = Get_traveltime_3D_CRS (MM, HH, t0, w, M, N);
tti_ncrs_2D = Get_traveltime_3D_nCRS(MM, HH, t0, w, M, N);
tti_icrs_2D = Get_traveltime_3D_iCRS_el_LIA(MM, HH, t0, V0, w, M, N,
 3);
```

```
texac = reshape(tti_ex_2D,81,41);
tcrs  = reshape(tti_crs_2D,81,41);
tncrs = reshape(tti_ncrs_2D,81,41);
ticrs = reshape(tti_icrs_2D(end,:),81,41);

m = -500:25:500;
h = 0:25:2000;

figure(3)
suptitle('Relative errors')
%
subplot(3,1,1)
data = (tcrs - texac)./texac*100;
imagesc(h,m',flipud(data'))
xlabel('h [m]');
ylabel('m [m]');
title('CRS')
colorbar
colormap('Jet');
caxis([-2 2])
hold on
contour(h,m',flipud(data'),'color','k','ShowText','on')
c = colorbar('Ticks', [-2,-1 0, 1, 2]);
c.Label.String = 'Error %';

subplot(3,1,2)
data = (tncrs - texac)./texac*100;
imagesc(h,m',flipud(data'))
xlabel('h [m]');
ylabel('m [m]');
title('n-CRS')
colorbar
colormap('Jet');
caxis([-2 2])
hold on
contour(h,m',flipud(data'),'color','k','ShowText','on')
c = colorbar('Ticks', [-2,-1 0, 1, 2]);
c.Label.String = 'Error %';

subplot(3,1,3)
data = (ticrs - texac)./texac*100;
imagesc(h,m',flipud(data'))
xlabel('h [m]');
ylabel('m [m]');
title('i-CRS')
colorbar
colormap('Jet');
caxis([-2 2])
hold on
contour(h,m',flipud(data'),'color','k','ShowText','on')
c = colorbar('Ticks', [-2,-1 0, 1, 2]);
c.Label.String = 'Error %';
```



Published with MATLAB® R2016b