3D Shifted hyperbola

Table of Contents

Author: Abakumov Ivan

Publication date: 30th December 2016

Define working folder, add links to Library and SeisLab

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

Part I: Model and acquisition

In this test I will use model 63. Model 63 consists of the constant velocity part ($v_0 = 1500\ m/s, z \leq 250\ m$) simulating the water layer, and the constant-gradient velocity part ($v = v_0 + \kappa(z - z_0), z_0 = 250\ m, \kappa = 0.5\ s^{-1}, z > 250\ m$) simulating the sedimentary layer. The reflector simulates the top of the salt body. The reflector is described by the fourth order polynomial function of lateral coordinates. The black line indicates the trajectory of the central ray. The depth of the NIP point is approximately equal to $1.0\ km$. Traveltimes in this model computed numerically with a very high precision.

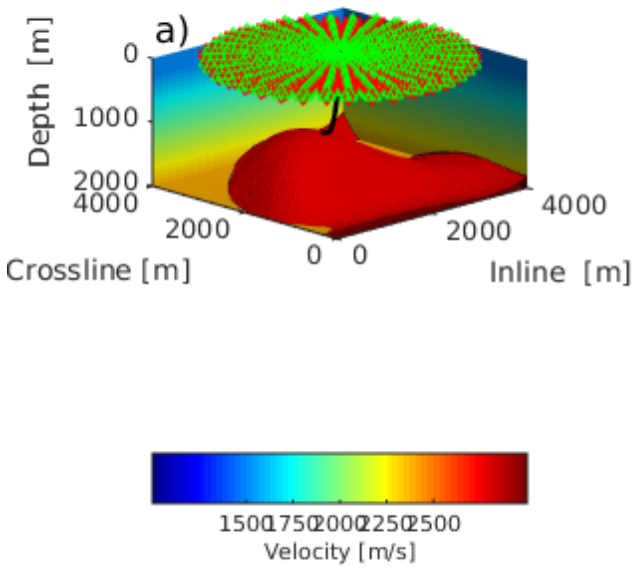
```
model = 63;
acquisition = 4;

Get_model_parameters;
Get_model_acquisition_geometry;

hx = (Xg(1, :) - Xs(1,:))/2;
hy = (Xg(2, :) - Xs(2,:))/2;
hh = sqrt(hx.^2 + hy.^2);
```

```
of_CMP = offset;
az_CMP = azimuth;
```

Part I: Plot acquisition geometry

```
[XX, YY] = meshgrid(G.xx, G.yy);
[ZZ, ind] = Get_model_surface(XX,YY,model);

Gold = oldGrid(G);

velmod = zeros(G.nx, G.ny, G.nz);

for k=1:25
    velmod(:,:,k) = 1500;
end
for k=26:201
    velmod(:,:,k) = velmod(:,:,k-1) + G.dz*0.5;
end
velmod_no_salt = velmod;
for i=1:G.nx
    for j=1:G.ny
        top = round(ZZ(i,j)/G.dz)+1;
        velmod(i,j,top:end) = 4400;
    end
end
[sXX, sYY, sZZ] = meshgrid(G.xx, G.yy, G.zz);

% Create central ray

X0 = [ 2000; 2000; 0];

[~, xref ]  = Get_model_exact_traveltime(X0, X0, 63);

tti  = FSM3D(Gold, xref, velmod_no_salt);
xi = x2grid(X0(1), G.x0, G.dx, G.nx);
yi = x2grid(X0(2), G.y0, G.dy, G.ny);
zi = x2grid(X0(3), G.z0, G.dz, G.nz);

dt = tti(xi,yi,zi)/G.nx/1.002;
ray = zeros(3, G.nx+1);
ray(:,1) = X0;

for i=1:G.nx
    xi = x2grid(ray(1,i), G.x0, G.dx, G.nx);
    yi = x2grid(ray(2,i), G.y0, G.dy, G.ny);
    zi = x2grid(ray(3,i), G.z0, G.dz, G.nz);
    pp(1,1) = (tti(xi+1,yi,zi) - tti(xi,yi,zi))/G.dx;
    pp(2,1) = (tti(xi,yi+1,zi) - tti(xi,yi,zi))/G.dy;
    pp(3,1) = (tti(xi,yi,zi+1) - tti(xi,yi,zi))/G.dz;
    ray(:,i+1) = ray(:,i) - pp*dt/(norm(pp))^2;
end
ray(:,end) = xref;
```

```
% Plot 3D velocity cube

figure(1)
subplot(3,2,[1,3,5])

xs = 4000;
ys = 4000;
zs = 2000;
h=slice(sXX,sYY,sZZ,velmod,xs,ys,zs);
set(h,'FaceColor','interp','EdgeColor','none','DiffuseStrength',.8)
xlabel('Inline  [m]');
ylabel('Crossline [m]');
zlabel('Depth  [m]');
colormap('jet')
caxis([1000 3000])
camlight(100,100)
lighting gouraud
hold on
p = patch(isosurface(sXX,sYY,sZZ,velmod,4200));
isonormals(sXX,sYY,sZZ,velmod,p)
p.FaceColor = 'red';
p.EdgeColor = 'none';
daspect([1,1,1])
view(-44,16);
hold on
plot3(ray(1,:),ray(2,:),ray(3,:),'-black', 'LineWidth',2  );
plot3(X0(1),X0(2),X0(3),'*black' );
plot3(xref(1), xref(2), xref(3), '.black','LineWidth',2 )
axis tight
set(gca, 'ZDir', 'reverse')
axis([0 4000 0 4000 0 2000])
c = colorbar('southoutside', 'Ticks', [1500,1750 2000, 2250, 2500]);
c.Label.String = 'Velocity [m/s]';
plot3(Xs(1,:),Xs(2,:),Xs(3,:), 'rv');
plot3(Xg(1,:),Xg(2,:),Xg(3,:), 'g^');
plot3(X0(1),X0(2),X0(3), 'b*');
text(1000,5000,'a)','FontSize',14,'Color','black')
```



Part I: Download traveltimes

```
tti_ex_CMP = MLD([mlibfolder '/CRS/models/model_' num2str(model) '_traveltimes_for

tti_ex_CMP    = reshape(tti_ex_CMP,length(az_CMP),length(of_CMP));
hx    = reshape(hx,length(az_CMP),length(of_CMP));
hy    = reshape(hy,length(az_CMP),length(of_CMP));
hh    = reshape(hh,length(az_CMP),length(of_CMP));
```

Part I: Find true stacking parameters: VNMO, TP

```
tind = 1:2;
VNMO = zeros(1,length(azimuth));
TP   = zeros(1,length(azimuth));

t0 = tti_ex_CMP(1,1);
v0 = 1500;

for az = 1:length(azimuth)
    % take minimum possible aperture
    tex = tti_ex_CMP(az, tind);

    % NMO ellipse (VNMO)
```

```
    f_e = @(vNMO)(sum((tex-sqrt(t0^2 + 4*offset(tind).^2/vNMO^2)).^2));
    VNMO(az) = fminbnd(f_e, 1000, 3000);

    % Shifted hyperbola (1 parameter - TP)
    f_o = @(tp)(sum((tex-sqrt(tp^2 + 4*offset(tind).^2/v0^2) + (tp-t0)).^2));
    TP(az) = fminbnd(f_o, 0, 2*t0);
end
```

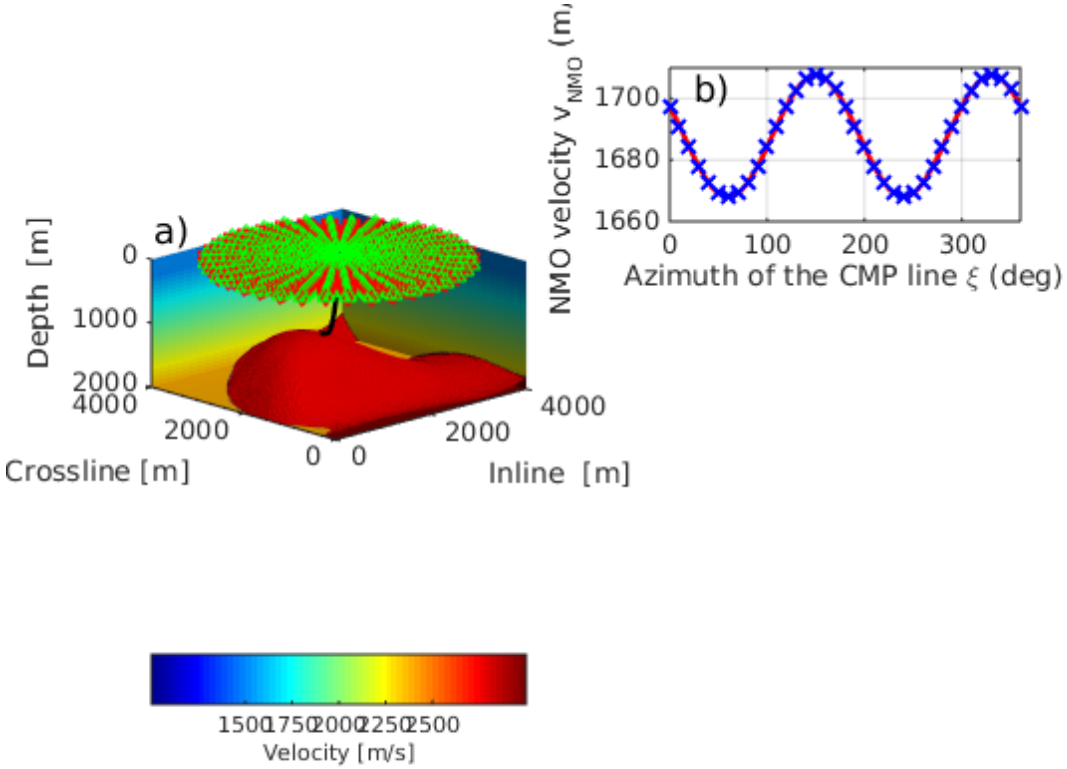
Part I: Find true stacking parameters: TP', VA, S

```
tind = 1:3;
TAVA = zeros(2,length(azimuth));

for az = 1:length(azimuth)
    % take minimum possible aperture
    tex = tti_ex_CMP(az, tind);

    % Shifted hyperbola (2 parameters - TP and VA)
    f_a = @(tava)(sum((tex-sqrt(tava(1)^2 + 4*offset(tind).^2/tava(2)^2) + (tava(1
    TAVA(:,az) = fminsearch(f_a, [t0;  v0]);
end

TA = TAVA(1,:);
VA = TAVA(2,:);

STA = t0./TA;
SVA = (VA./VNMO).^2;
S = (STA + SVA)/2;
```

Part I: Plot true stacking parameters: VNMO

```
x = 0:10:360;
y = VNMO;
yu = max(y);
yl = min(y);
yr = (yu-yl);                                % Range of 'y'
yz = y-yu+(yr/2);
zx = x(yz .* circshift(yz,[0 1]) <= 0);      % Find zero-crossings
per = 2*mean(diff(zx));                      % Estimate period
ym = mean(y);                                % Estimate offset

fit = @(b,x)  b(1).*(sin(2*pi*x./b(2) + 2*pi/b(3))) + b(4);    % Function to fit
fcn = @(b) sum((fit(b,x) - y).^2);                              % Least-Squares co
s = fminsearch(fcn, [yr;  per;  -1;  ym]);

xp = linspace(min(x),max(x));

subplot(3,2,2)
plot(xp,fit(s,xp), '-r', 'Linewidth', 2)
hold on
plot(x,y,'xb',  'Linewidth', 2)
```

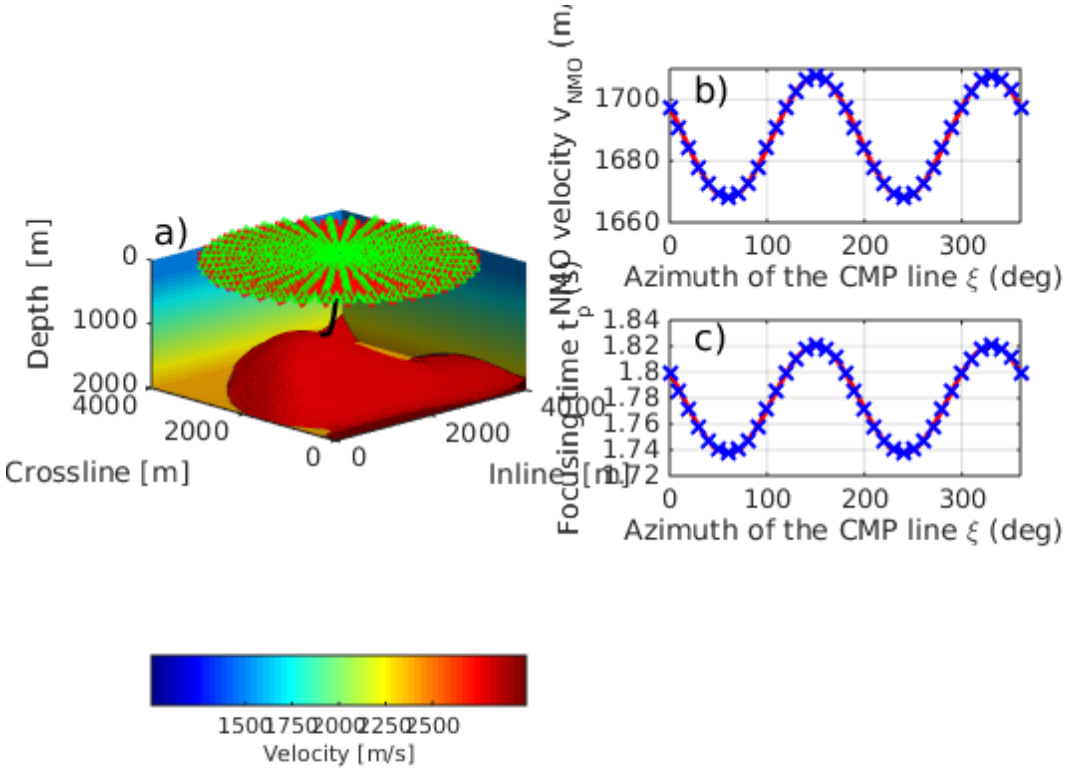
```
axis([0,360,1660,1710])
xlabel('Azimuth of the CMP line \xi (deg)')
ylabel('NMO velocity v_{NMO} (m/s)')
text(25,1702,'b)','FontSize',14,'Color','black')
grid
```



Part I: Plot true stacking parameters: TP

```
x = 0:10:360;
y = TP;
yu = max(y);
yl = min(y);
yr = (yu-yl);                                   % Range of 'y'
yz = y-yu+(yr/2);
zx = x(yz .* circshift(yz,[0 1]) <= 0);         % Find zero-crossings
per = 2*mean(diff(zx));                         % Estimate period
ym = mean(y);                                   % Estimate offset

fit = @(b,x)  b(1).*(sin(2*pi*x./b(2) + 2*pi/b(3))) + b(4);    % Function to fit
fcn = @(b) sum((fit(b,x) - y).^2);                              % Least-Squares co
s = fminsearch(fcn, [yr;  per;  -1;  ym]);

xp = linspace(min(x),max(x));

subplot(3,2,4)
plot(xp,fit(s,xp), '-r', 'Linewidth', 2)
```

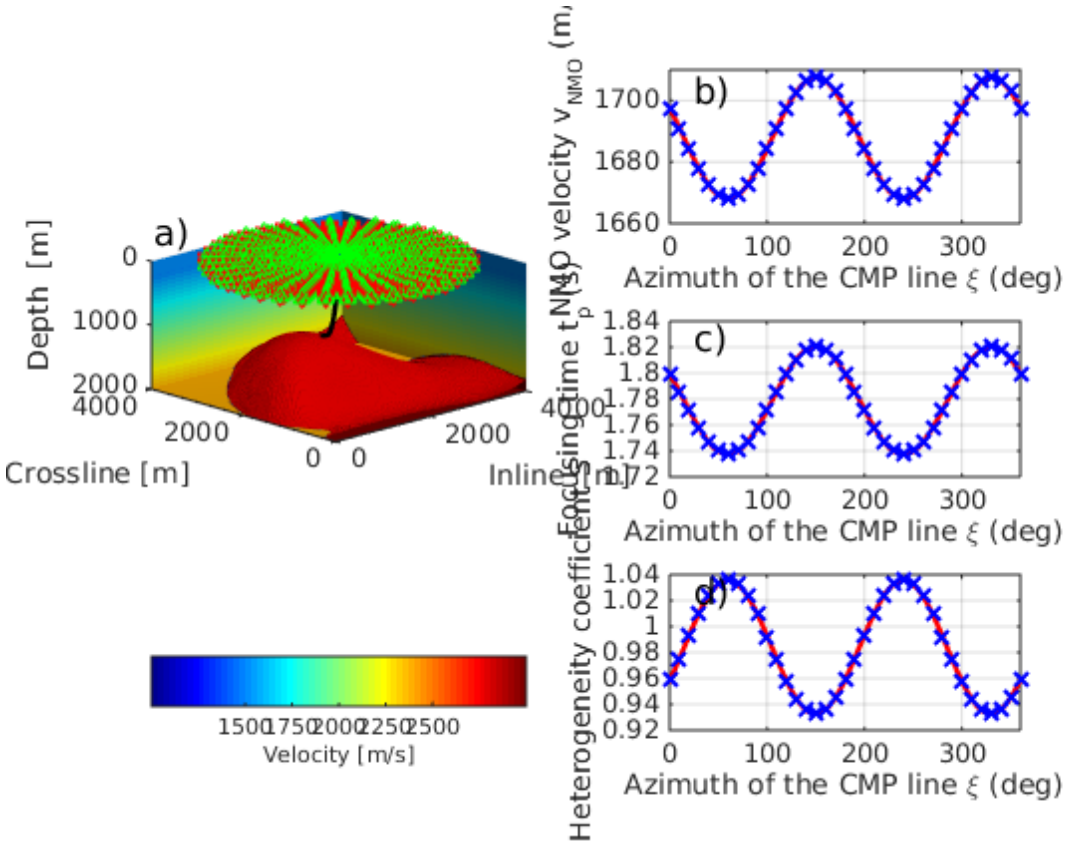
```
hold on
plot(x,y,'xb',  'Linewidth', 2)
axis([0,360,1.72,1.84])
xlabel('Azimuth of the CMP line \xi (deg)')
ylabel('Focusing time t_p (s)')
text(25,1.825,'c)','FontSize',14,'Color','black')
grid
```



Part I: Plot true stacking parameters: S

```
x = 0:10:360;
y = S;
yu = max(y);
yl = min(y);
yr = (yu-yl);                                % Range of ‘y’
yz = y-yu+(yr/2);
zx = x(yz .* circshift(yz,[0 1]) <= 0);      % Find zero-crossings
per = 2*mean(diff(zx));                      % Estimate period
ym = mean(y);                                % Estimate offset

fit = @(b,x)  b(1).*(sin(2*pi*x./b(2) + 2*pi/b(3))) + b(4);    % Function to fit
fcn = @(b) sum((fit(b,x) - y).^2);                              % Least-Squares co
s = fminsearch(fcn, [yr;  per;  -1;  ym]);

xp = linspace(min(x),max(x));
```

```
subplot(3,2,6)
plot(xp,fit(s,xp), '-r', 'Linewidth', 2)
hold on
plot(x,y,'xb',  'Linewidth', 2)
axis([0,360,0.92,1.04])
xlabel('Azimuth of the CMP line \xi (deg)')
ylabel('Heterogeneity coefficient S')
text(25,1.025,'d)','FontSize',14,'Color','black')
grid
```



Part II: Find errors of 3D moveout approximations

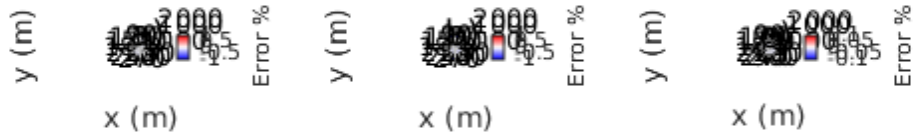
```
tind = 1:11;

ERR_e = zeros(length(azimuth),length(offset(tind)));
ERR_o = zeros(length(azimuth),length(offset(tind)));
ERR_a = zeros(length(azimuth),length(offset(tind)));

for az = 1:length(azimuth)
    tex = tti_ex_CMP(az, tind);

    % NMO ellipse (VNMO)
    e_e = @(VNMO)(tex-sqrt(t0^2 + 4*offset(tind).^2/VNMO^2));
    ERR_e(az, :) = e_e(VNMO(az));
```

```
    % Shifted hyperbola (1 parameter - TP)
    e_o = @(tp)(tex-sqrt(tp^2 + 4*offset(tind).^2/v0^2) + (tp-t0));
    ERR_o(az, :) = e_o(TP(:,az));

    % Shifted hyperbola (2 parameters - TP and VA)
    e_a = @(tpva)(tex-sqrt(tpva(1)^2 + 4*offset(tind).^2/tpva(2)^2) + (tpva(1)-t0)
    ERR_a(az, :) = e_a(TAVA(:,az));
end
```

Part II: Plot errors of 3D moveout approximations

```
[OF, AZ] = meshgrid(offset(tind),azimuth);
XX = 2*OF.*cos(AZ);
YY = 2*OF.*sin(AZ);

figure(2)
subplot(1,3,1)
polar(azimuth,2000*ones(size(azimuth)), '-black');
hold on
contourf(XX,YY,ERR_e*100,5);
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')
text(-2000,2000,'a)','FontSize',14,'Color','black')
c = colorbar('Ticks', [-1,-0.5 0, 0.5, 1]);
c.Label.String = 'Error %';
caxis([-1 1])
colormap(makeColorMap([0 0 1],[1 1 1],[1 0 0],100));
xlabel('x (m)')
ylabel('y (m)')

subplot(1,3,2)
polar(azimuth,2000*ones(size(azimuth)), '-black');
hold on
contourf(XX,YY,ERR_o*100,5);
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')
text(-2000,2000,'b)','FontSize',14,'Color','black')
c = colorbar('Ticks', [-1,-0.5 0, 0.5, 1]);
```

```
c.Label.String = 'Error %';
caxis([-1 1])
colormap(makeColorMap([0 0 1],[1 1 1],[1 0 0],100));
xlabel('x (m)')
ylabel('y (m)')

subplot(1,3,3)
polar(azimuth,2000*ones(size(azimuth)), '-black');
hold on
contourf(XX,YY,ERR_a*100,5);
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')
text(-2000,2000,'c)','FontSize',14,'Color','black')
c = colorbar('Ticks', [-0.1,-0.05 0, 0.05, 0.1]);
c.Label.String = 'Error %';
caxis([-0.1 0.1])
colormap(makeColorMap([0 0 1],[1 1 1],[1 0 0],100));
xlabel('x (m)')
ylabel('y (m)')
```



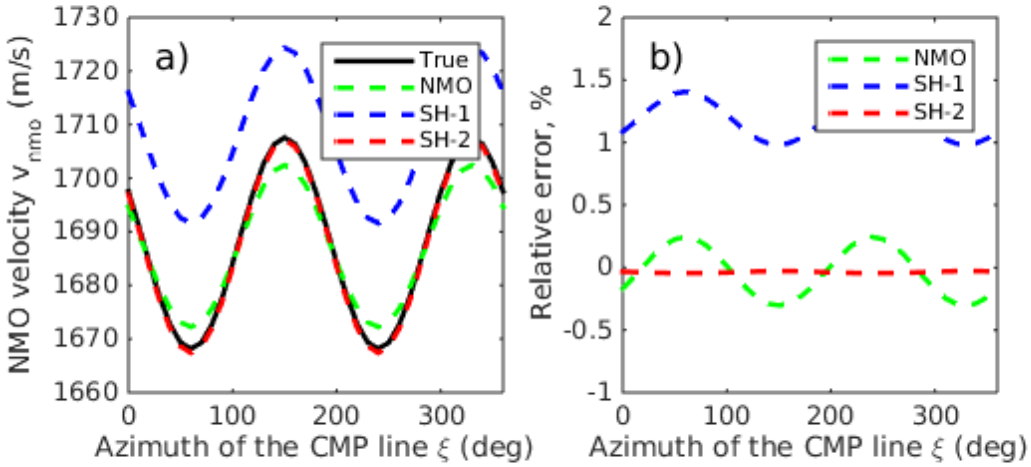
Part III: Find best-fit stacking parameters

```
tind = 1:11;

VNMObf = zeros(1,length(azimuth));
TPbf   = zeros(1,length(azimuth));
TPVAbf = zeros(2,length(azimuth));

for az = 1:length(azimuth)
    tex = tti_ex_CMP(az, tind);

    % NMO ellipse (VNMO)
    f_e = @(vNMO)(sum((tex-sqrt(t0^2 + 4*offset(tind).^2/vNMO^2)).^2));
    e_e = @(VNMO)(tex-sqrt(t0^2 + 4*offset(tind).^2/VNMO^2));
    VNMObf(az) = fminbnd(f_e, 1000, 3000);

    % Shifted hyperbola (1 parameter - TP)
    f_o = @(tp)(sum((tex-sqrt(tp^2 + 4*offset(tind).^2/v0^2) + (tp-t0)).^2));
    e_o = @(tp)(tex-sqrt(tp^2 + 4*offset(tind).^2/v0^2) + (tp-t0));
    TPbf(az) = fminbnd(f_o, 0, 2*t0);

    % Shifted hyperbola (2 parameters - TP and VA)
    f_a = @(tpva)(sum((tex-sqrt(tpva(1)^2 + 4*offset(tind).^2/tpva(2)^2) + (tpva(1
    e_a = @(tpva)(tex-sqrt(tpva(1)^2 + 4*offset(tind).^2/tpva(2)^2) + (tpva(1)-t0)
    TPVAbf(:,az) = fminsearch(f_a, [t0;  v0]);
end
TAbf = TPVAbf(1,:);
VAbf = TPVAbf(2,:);
```

Part III: Plot best-fit stacking parameters

```
VNMO_e = VNMObf;
VNMO_o = sqrt(TPbf/t0)*v0;
VNMO_a = sqrt(TAbf/t0).*VAbf;

figure(3)
az = azimuth/pi*180;
subplot(1,2,1)
plot(az, VNMO, '-k',    'Linewidth', 2);
hold on
plot(az, VNMO_e, '--g',    'Linewidth', 2);
plot(az, VNMO_o, '--b',    'Linewidth', 2);
plot(az, VNMO_a, '--r',    'Linewidth', 2);
xlabel('Direction of the profile \xi (deg)')
ylabel('Focusing time t_p (s)')
legend('True', 'NMO', 'SH-1', 'SH-2');
axis('square')
axis([0 360 1660 1730])
xlabel('Azimuth of the CMP line \xi (deg)')
ylabel('NMO velocity v_{nmo} (m/s)')
text(25,1722,'a)','FontSize',14,'Color','black')
```

```
subplot(1,2,2)
plot(az, (VNMO_e-VNMO)./VNMO*100, '--g',    'Linewidth', 2);
hold on
plot(az, (VNMO_o-VNMO)./VNMO*100, '--b',    'Linewidth', 2);
plot(az, (VNMO_a-VNMO)./VNMO*100, '--r',    'Linewidth', 2);
xlabel('Direction of the profile \xi (deg)')
ylabel('Focusing time t_p (s)')
legend('NMO', 'SH-1', 'SH-2');
axis('square')
axis([0 360 -1 2])
xlabel('Azimuth of the CMP line \xi (deg)')
ylabel('Relative error, %')
text(25,1.66,'b)','FontSize',14,'Color','black')
```



Published with MATLAB® R2014b