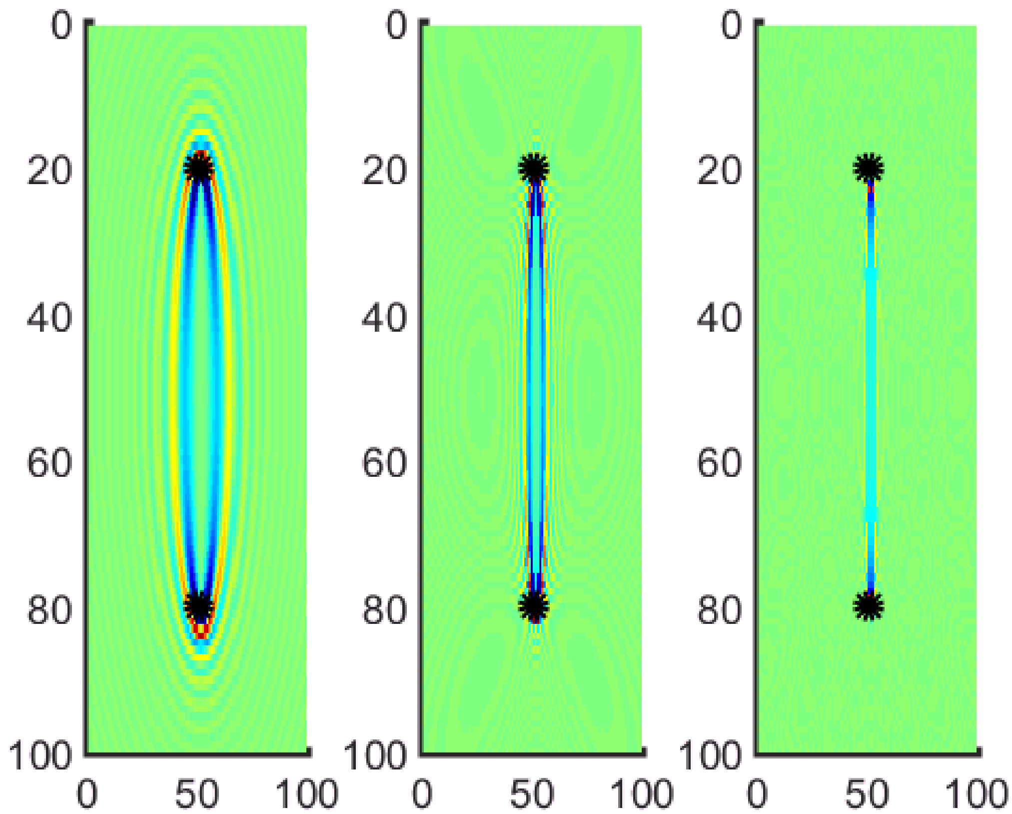
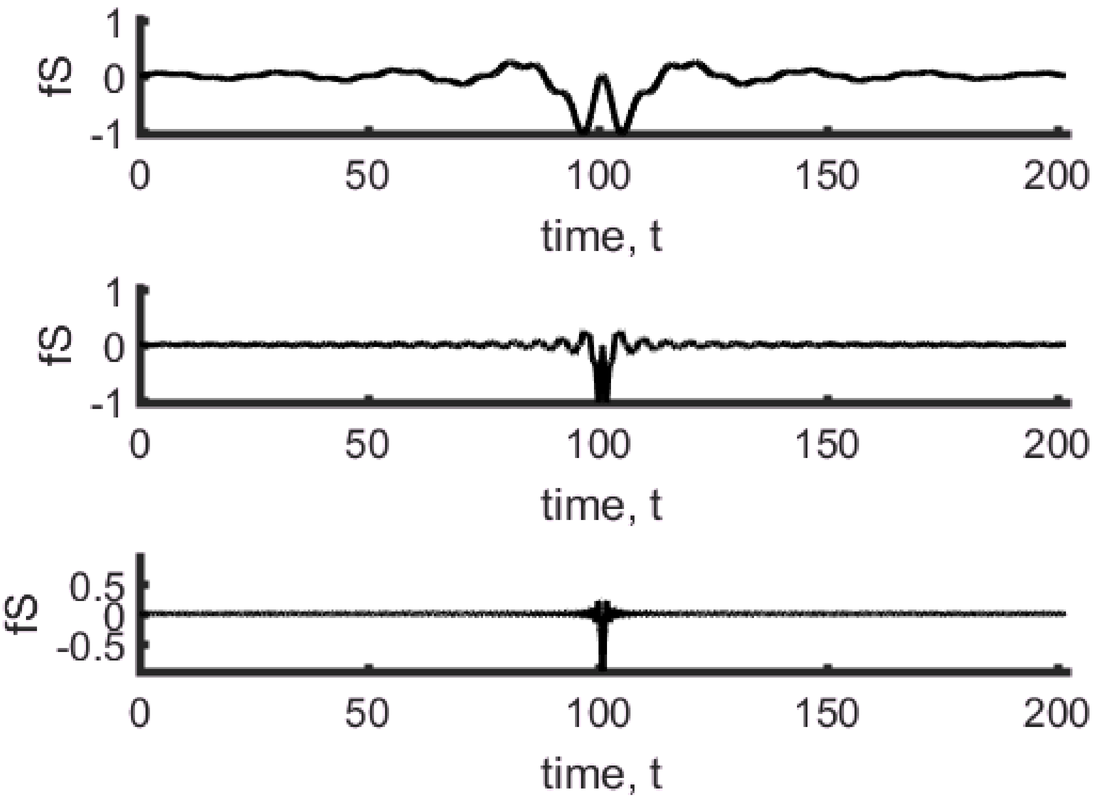
```
% gda13_04
% Supports Figure 13.3
% bannana-doughnut kernel for the acoustic case

clear all;

% images of kernel
figure(1);
clf;

% time series
figure(2);
clf;

% reference slowness
s0 = 1;
m = 1;

% independent variable x
Nx = 101;
Dx = 1;
x = Dx*[0:Nx-1]';
xmin = x(1);
xmax = x(end);

% independent variable y
Ny = 101;
Dy = 1;
y = Dy*[0:Ny-1]';
ymin = y(1);
ymax = y(end);

% independent variable t
SSF = 10;
Nt = 2*Nx*SSF; % number of times depends on number of x's
Dt = 1/SSF;    % time sampling depends on spatial sampling
t = Dt*[0:Nt-1]';
tmin=t(1);
tmax=t(end);

% source time
t0 = tmin + (tmax-tmin)/2;

% define three frequency bands, one for each case
wny = (2*pi)/(2*Dt);
w0 = [0.005*wny; 0.025*wny; 0.1*wny];

% loop over three cases of finite frequencies
for iw=[1:3]

% a band-limited source time function
fS = sinc(2*w0(iw)*(t-t0)) - sinc(0.5*w0(iw)*(t-t0));

% d/dt of source time function
fSd = zeros(Nt,1);
fSd(1:Nt-1) = (fS(2:Nt)-fS(1:Nt-1))/(Dt);

% d2/dt2 of source time function
fSdd = zeros(Nt,1);
```

```
fSdd(2:Nt-1) = (fS(1:Nt-2)-2*fS(2:Nt-1)+fS(3:Nt))/(Dt^2);

% source location
xS = 80;
yS = 51;

% receiver location
xR = 20;
yR = 51;

% Source to Receiver distance and time
RSR = sqrt( (xS-xR)^2 + (yS-yR)^2 );
TSR = s0*RSR;
nSR = floor( TSR/Dt );

% a constant used later
D = - (1/(4*pi*RSR)^2) * Dt * sum( fSd .^ 2 );

d = zeros(Nx,Ny); % the kernel on the (x,y) plane
for ix = [1:Nx] % loop over x
for iy = [1:Ny] % loop over y

    % location of heterogeneity
    xh = x(ix);
    yh = y(iy);

    % Source to heterogenity distance and time
    RSh = sqrt( (xS-xh)^2 + (yS-yh)^2 );
    TSh = s0*RSh;

    % heterogenity to receiver distance and time
    RhR = sqrt( (xR-xh)^2 + (yR-yh)^2 );
    ThR = s0*RhR;

    % some constants
    NShR = floor((TSh+ThR-TSR)/Dt);
    c = 4*pi*RSR * 4*pi*(RSh+Dx) * 4*pi*(RhR+Dx);
    % note, the Dx is a kludge to prevent singularies

    % adjoint field
    a = zeros(Nt,1);
    a(1:Nt-NShR) = fSd(NShR+1:Nt);

    % second derivative of reference field
    b = fSdd;

    % kernel is an inner product of a and b
    d(ix,iy) = (2*s0*D)*Dt*sum( a .* b )/c;

end
end

% plot kernel
figure(1)
subplot(1,3,iw);
colormap('jet');
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on
axis ij
axis( [xmin, xmax, ymin, ymax] );
```

```
dabsmax = max(abs(d(:)))/3;
imagesc(d, [-dabsmax, dabsmax]);
plot( yS, xS, 'ko', 'LineWidth', 4 );
plot( yR, xR, 'ko', 'LineWidth', 4 );

% plot time series
figure(2)
subplot(3,1,iw);
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on
axis( [tmin, tmax, -max(abs(fS)), max(abs(fS))] );
plot( t, fS, 'k-', 'LineWidth', 2 );
xlabel('time, t');
ylabel('fS');

end
```






```
% Figure 13.3 Banana-doughnut kernels (the partial derivative ∂T⁄∂m of finite
% frequency travel time T with respect to model parameter m) for an acousti
% wave propagation problem.  The source x_S and receiver x_R  (back circles)
% are separated by a distance of R=60 km. The perturbation in acoustic velocity
% is δv=(m/2v_0 )  δ(x-x_h ), where v_0=1 km/s is a constant reference velocity
% and x_h is the heterogeneity's location. (A) – (C) Kernels (colors) for a
% suite of x_h's covering the (x,y) plane, for frequency bands centers at
% 0.025,0.125 and 0.500 Hz, respectively. Note that the "banana" shape of the
% kernel narrows with frequency, becoming increasingly ray-like.   (D)-(E)
% Corresponding band-limited source time functions used in the calculation.
% MatLab script gda13_04.m.
```