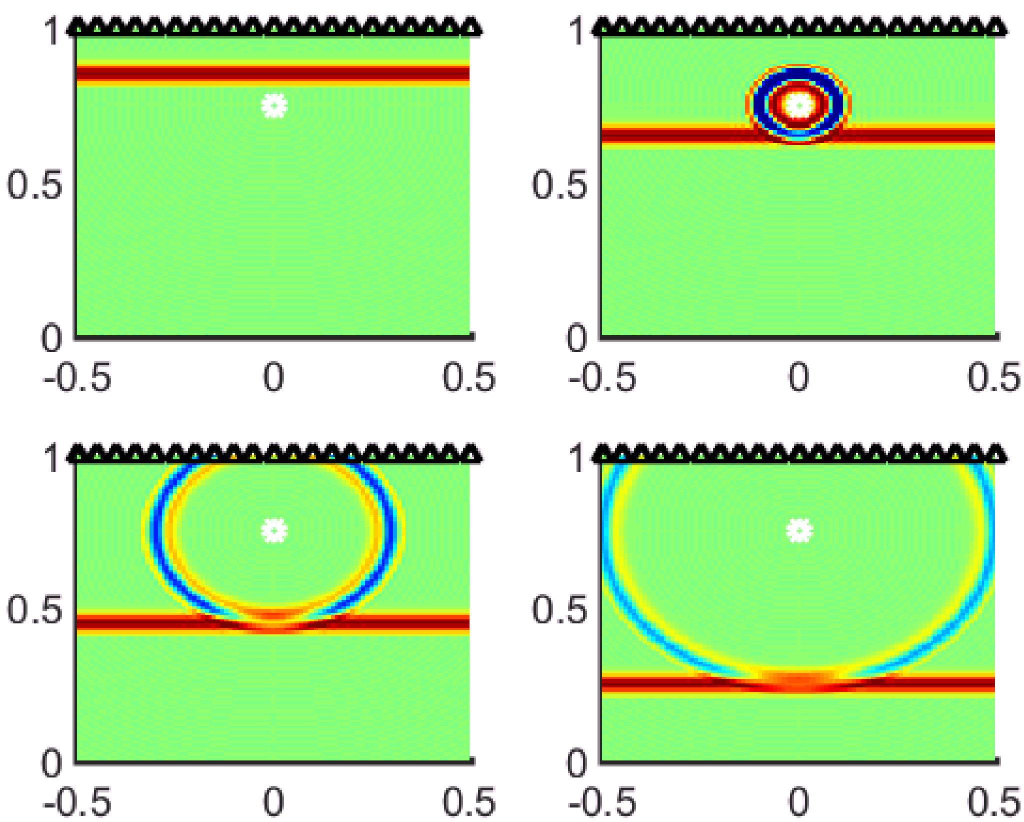
```
% gda13_05
% Supports Figures 13.4 and 13.5
% seismic migration in a medium with a uniform
% reference slowness s0, and with a reference
% (unperturbed) wavefield that is a downgoing
% plane wave.

% Note: This script uses a function gda_delay()
% that is not desribed in the text (sorry about
% that) that delays a time series by a given amount.
% It is roughly equivalent to the MatLab function
% circshift(), except that it allows delays that
% are non-integral multiples of the sampling interval
% Dt.  It uses a Fourier technique.

clear all;

% reference slowness
s0 = 1;

% axes setup.  Note: Although the calulations
% are for a plane wave that starts at z=0 and
% advances in the positive z direction, I plot
% eveything upside down so that it looks "downgoing"
% (Sorry about that).

% independent variable x
xmin = -0.5;
xmax =  0.5;
Nx = 101;
Dx = (xmax-xmin)/(Nx-1);
x = xmin + Dx*[0:Nx-1]';

% independent variable z
zmin = 0;
zmax = 1;
Nz = 101;
Dz = (zmax-zmin)/(Nz-1);
z = zmin + Dz*[0:Nz-1]';

% independent variable t
% sampling Dt depends on sampling Dz
Dt = s0*Dz;
Nt = 512;
tmin = 0;
t = tmin + Dt*[0:Nt-1]';
tmax = t(end);

% in order to prevent singularities, geometrical
% spreading 1/R is replaced with 1/(R+epsi)
epsi = 4*Dx;

% source time function is a Gaussian pulse
sigmat = 2*Dt;
t0 = 2*sigmat;
p = exp( -((t-t0).^2)/(2*sigmat*sigmat) );
pdd = [0;diff(p,2);0]; % first derivative
pdd = (0.1/(4*pi))*pdd/max(pdd); % second derivative
```

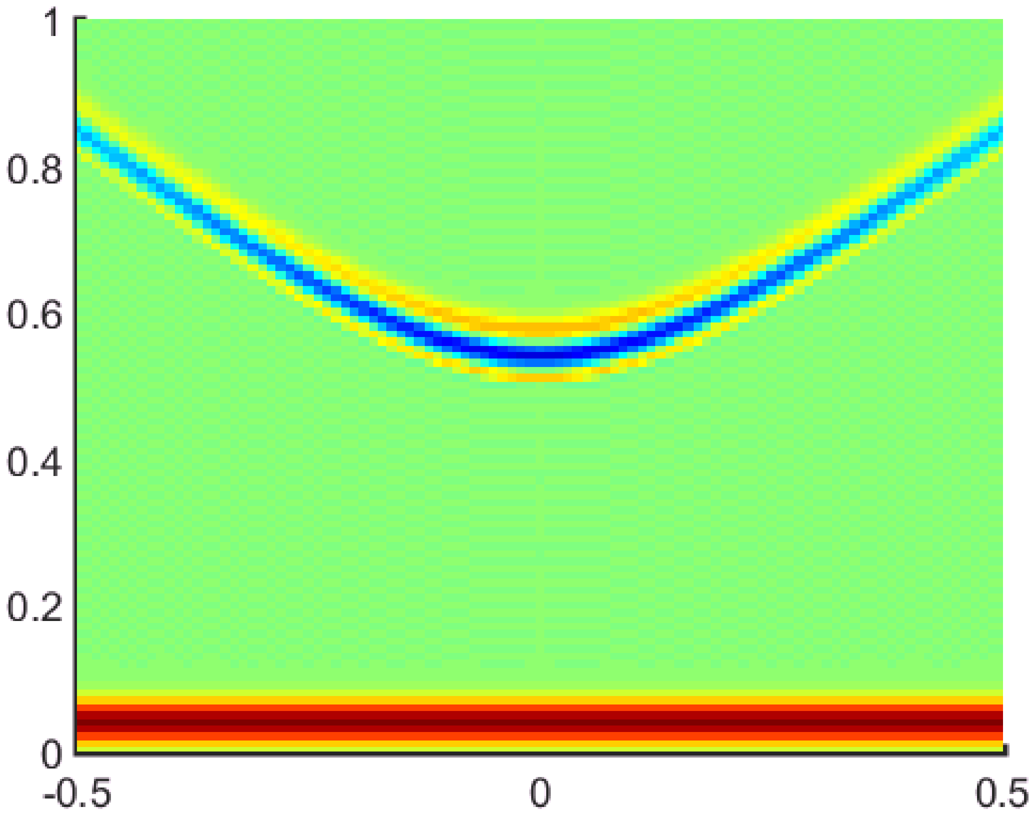
```
% unperturbed plane wave pulse on the grid
u0 = zeros(Nz,Nx,Nt);
for iz = [1:Nz]
    tau0 = s0*(z(iz)-zmin); % phase delay for a plane wave
    d=gda_delay(p,Dt,tau0);
    for ix = [1:Nx]
        % note that pwave wave experiences no geometerical spreading
        u0(iz,ix,:)=d;
    end
end

% array of stations (receivers), all at zmin
istas=[1:5:Nx]';

% point scatterer
xs = 0;
zs = 0.25;

% scattered wavefield
tau0 = s0*(zs-zmin); % delay of planewave down to scatterer
du = zeros(Nz,Nx,Nt);
for iz = [1:Nz]
    for ix = [1:Nx]
        R=sqrt(((x(ix)-xs)^2)+((z(iz)-zs)^2));
        Ri = 1/(R+epsi); % fudged geometrical spreading
        dtau = s0*R; % delay of scattered wave
        tau1 = dtau+tau0; % total delay
        d=gda_delay(pdd,Dt,tau1);
        du(iz,ix,:)=d*(4*pi*Ri);
    end
end

% plot wavefield
figure(1);
clf;
subplot(2,2,1);
itf = 20;
it = itf;
set(gca,'LineWidth',3);
hold on;
set(gca,'FontSize',14);
axis( [xmin, xmax, zmin, zmax] );
colormap('jet');
imagesc([xmin, xmax], [zmax, zmin], u0(:,:,it)+du(:,:,it), [-1, 1]);
plot(xs,zmax-zs,'wo','LineWidth',3);
plot(x(istas), zmax-0, 'k^', 'LineWidth', 2 );
subplot(2,2,2);
it = itf*2;
set(gca,'LineWidth',3);
hold on;
set(gca,'FontSize',14);
axis( [xmin, xmax, zmin, zmax] );
colormap('jet');
imagesc([xmin, xmax], [zmax, zmin], u0(:,:,it)+du(:,:,it), [-1, 1]);
plot(xs,zmax-zs,'wo','LineWidth',3);
plot(x(istas), zmax-0, 'k^', 'LineWidth', 2 );
subplot(2,2,3);
it = itf*3;
set(gca,'LineWidth',3);
hold on;
set(gca,'FontSize',14);
```

```
axis( [xmin, xmax, zmin, zmax] );
colormap('jet');
imagesc([xmin, xmax], [zmax, zmin], u0(:,:,it)+du(:,:,it), [-1, 1]);
plot(xs,zmax-zs,'wo','LineWidth',3);
plot(x(istas), zmax-0, 'k^', 'LineWidth', 2 );
subplot(2,2,4);
it = itf*4;
set(gca,'LineWidth',3);
hold on;
set(gca,'FontSize',14);
axis( [xmin, xmax, zmin, zmax] );
colormap('jet');
imagesc([xmin, xmax], [zmax, zmin], u0(:,:,it)+du(:,:,it), [-1, 1]);
plot(xs,zmax-zs,'wo','LineWidth',3);
plot(x(istas), zmax-0, 'k^', 'LineWidth', 2 );
```



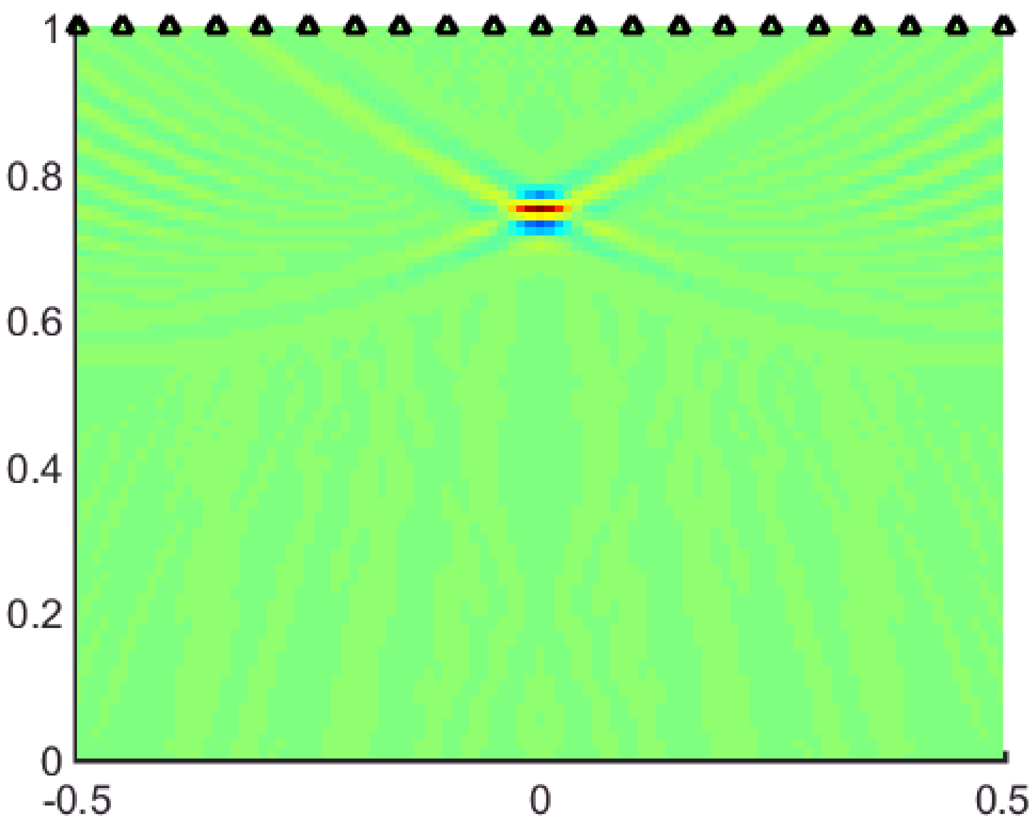

```
% Figure 13.4 (A)-(D) Sequence of four time snapshots of a down-going
% incident plane wave (red band) interacting with a point-like heterogeneity
% (white circle) and generating a scattered spherical wave. The wave field is
% observed by an array of receivers (triangles) on the Earth’s surface.
% MatLab script gda_13.05.

% plot record section on surface
figure(2);
itp = 101;
tp = Dt*(itp-1);
clf;
set(gca,'LineWidth',3);
hold on;
set(gca,'FontSize',14);
axis( [xmin, xmax, 0, tp] );
```

```
colormap('jet');
imagesc([xmin, xmax], [0, tp], squeeze(u0(1,:,1:itp)+du(1,:,1:itp))', [-1, 1]);
```




```
% Figure 13.5 (A) Seismic refection data recorded by the array of receivers
% in Fig. 13.??, containing incident and scattered waves.

% back-propagate du
dubT = zeros(Nz,Nx,Nt);
for ixr = istas' % for each receiver on surface
    xr = x(ixr); % receiver position
    zr = z(1);
    dub = zeros(Nz,Nx,Nt); % wavefield at receiver
    a = squeeze(du(1,ixr,:)); % reduce to vector
    for iz = [1:Nz] % back-propagate to each gridpoint
    for ix = [1:Nx]
        R=sqrt(((x(ix)-xr)^2)+((z(iz)-zr)^2)); % distance
        Ri = 1/(R+epsi); % fudged geometrical spreading
        taub = -s0*R; % time advance
        d=gda_delay(a,Dt,taub);
        dub(iz,ix,:)=d*(4*pi*Ri); % for this observation
    end
    end
    dubT=dubT+dub; % stack all observations
end

% correlate
c = zeros(Nz,Nx);
for iz = [1:Nz] % each gridpoint
for ix = [1:Nx]
    a = squeeze(u0(iz,ix,:)); % reduce to vector
    add = [0;diff(a,2);0]; % 2nd derivative
```

```
    b = squeeze(dubT(iz,ix,:)); % reduce to vector
    c(iz,ix)=add'*b; % correlate (= dot product)
end
end
cmax = max(max(c));

% plot correlation
figure(3);
clf;
set(gca,'LineWidth',3);
hold on;
set(gca,'FontSize',14);
axis( [xmin, xmax, zmin, zmax] );
colormap('jet');
imagesc([xmin, xmax], [zmax, zmin], c/cmax, [-1, 1]);
plot(x(istas), zmax-0, 'k^', 'LineWidth', 2 );
```




```
% Figure 13.5B. The data from (A) is migrated to produce the correlation
% C(x,z), which is a proxy for the estimated heterogeneity m^est (x,z).
% The correlation is at the location of the true heterogeneity (compare
% with Fig. 13.04). MatLab script gda_13.05.
```