```
% gda12_13
% Supports Figure 12.19
% earthquake location example
% rectangular earth volume, straight line rays
% data are travel times of P and S waves

% note: the problem is formulated here so that all
% the earthquakes are located simultaneously, with
% a single data kernel.  Thus allows for the possibility
% of later adding constints thet involve several earthquakes

clear all;

clear G epsilon;
global G epsilon;
epsilon=1.0e-6;

% material constants
vpvs = 1.78;
vp=6.5;
vs=vp/vpvs;

% bounds of box
xmin=-10;
xmax=10;
ymin=-10;
ymax=10;
zmin=-10;
zmax=0;

% plot
figure(1);
clf;
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on;
axis( [xmin, xmax, ymin, ymax, zmin, zmax] );

% improvise outline of 3D box
plot3( [xmin,xmin], [ymin,ymin], [zmin,zmax], 'k-', 'LineWidth', 3 );
plot3( [xmin,xmin], [ymin,ymax], [zmin,zmin], 'k-', 'LineWidth', 3 );
plot3( [xmin,xmax], [ymin,ymin], [zmin,zmin], 'k-', 'LineWidth', 3 );
plot3( [xmax,xmax], [ymax,ymax], [zmax,zmin], 'k-', 'LineWidth', 3 );
plot3( [xmax,xmax], [ymax,ymin], [zmax,zmax], 'k-', 'LineWidth', 3 );
plot3( [xmax,xmin], [ymax,ymax], [zmax,zmax], 'k-', 'LineWidth', 3 );
plot3( [xmax,xmin], [ymin,ymin], [zmax,zmax], 'k-', 'LineWidth', 3 );
plot3( [xmax,xmax], [ymin,ymin], [zmax,zmin], 'k-', 'LineWidth', 3 );
plot3( [xmin,xmin], [ymax,ymin], [zmax,zmax], 'k-', 'LineWidth', 3 );
plot3( [xmin,xmin], [ymax,ymax], [zmax,zmin], 'k-', 'LineWidth', 3 );
plot3( [xmax,xmax], [ymax,ymin], [zmin,zmin], 'k-', 'LineWidth', 3 );
plot3( [xmax,xmin], [ymax,ymax], [zmin,zmin], 'k-', 'LineWidth', 3 );
xlabel('x_1');
ylabel('x_2');
zlabel('x_3');

% stations
sxm = [-8, -6, -4, -2, 0, 2, 4, 6, 8]'*ones(1,9);
sym = ones(9,1)*[-8, -6, -4, -2, 0, 2, 4, 6, 8];
sx = sxm(:);
```

```
sy = sym(:);
Ns = length(sx);
sz = zmax*ones(Ns,1);
plot3( sx, sy, sz, 'k+', 'LineWidth', 3 );

% earthquakes
Ne = 30;  % number of earthquakes
M = 4*Ne; % 4 model parameters, x, y, z, and t0, per earthquake
extrue = random('uniform',xmin,xmax,Ne,1);
eytrue = random('uniform',ymin,ymax,Ne,1);
eztrue = random('uniform',zmin,zmax,Ne,1);
t0true = random('uniform',0,0.2,Ne,1);
mtrue = [extrue', eytrue', eztrue', t0true']';

Nr = Ne*Ns;   % number of rays, that is, earthquake-stations pairs
N = 2*Ne*Ns;  % data: P and S wave for each earthquake at each depth
              % P times stored first in data array

% true data
% traveltime = length of ray divided by velocity
% ray is straight line connecting earthquake and station
dtrue=zeros(N,1);
for i = [1:Ns] % loop over stations
for j = [1:Ne] % loop over earthquakes
    dx = mtrue(j)-sx(i);
    dy = mtrue(Ne+j)-sy(i);
    dz = mtrue(2*Ne+j)-sz(i);
    r = sqrt( dx^2 + dy^2 + dz^2 );
    k=(i-1)*Ne+j;
    dtrue(k)=r/vp+mtrue(3*Ne+j);
    dtrue(Nr+k)=r/vs+mtrue(3*Ne+j);
end
end

% observed data is true data plus random noise
sd = 0.1;
dobs=dtrue+random('normal',0,sd,N,1);

% inital guess of earthquake locations is random
mest = [random('uniform',xmin,xmax,1,Ne), random('uniform',ymin,ymax,1,Ne), ...
    random('uniform',zmin+2,zmax-2,1,Ne), random('uniform',-0.1,0.1,1,Ne) ]';

% Geiger's method
for iter=[1:10]

    % data kernel
    G = spalloc(N,M,4*N);
    dpre = zeros(N,1);
    for i = [1:Ns] % loop over stations
    for j = [1:Ne] % loop over earthquakes
        dx = mest(j)-sx(i);
        dy = mest(Ne+j)-sy(i);
        dz = mest(2*Ne+j)-sz(i);
        r = sqrt( dx^2 + dy^2 + dz^2 );
        k=(i-1)*Ne+j;
        dpre(k)=r/vp+mest(3*Ne+j);
        dpre(Nr+k)=r/vs+mest(3*Ne+j);
        G(k,j) = dx/(r*vp);
        G(k,Ne+j) = dy/(r*vp);
        G(k,2*Ne+j) = dz/(r*vp);
        G(k,3*Ne+j) = 1;
```

```
        G(Nr+k,j) = dx/(r*vs);
        G(Nr+k,Ne+j) = dy/(r*vs);
        G(Nr+k,2*Ne+j) = dz/(r*vs);
        G(Nr+k,3*Ne+j) = 1;
    end
    end

    % solve with dampled least squares
    dd = dobs-dpre;
    dm=bicg(@dlsfun,G'*dd,1e-5,3*M);
    mest = mest+dm;

end
```

```
bicg converged at iteration 35 to a solution with relative residual 7.3e-06.
bicg converged at iteration 30 to a solution with relative residual 8.2e-06.
bicg converged at iteration 56 to a solution with relative residual 6.6e-06.
bicg converged at iteration 34 to a solution with relative residual 2.9e-06.
bicg converged at iteration 29 to a solution with relative residual 5.3e-06.
bicg converged at iteration 29 to a solution with relative residual 8e-06.
bicg converged at iteration 30 to a solution with relative residual 5.9e-06.
bicg converged at iteration 35 to a solution with relative residual 8e-06.
bicg converged at iteration 44 to a solution with relative residual 6.8e-06.
bicg converged at iteration 40 to a solution with relative residual 5.1e-06.
```

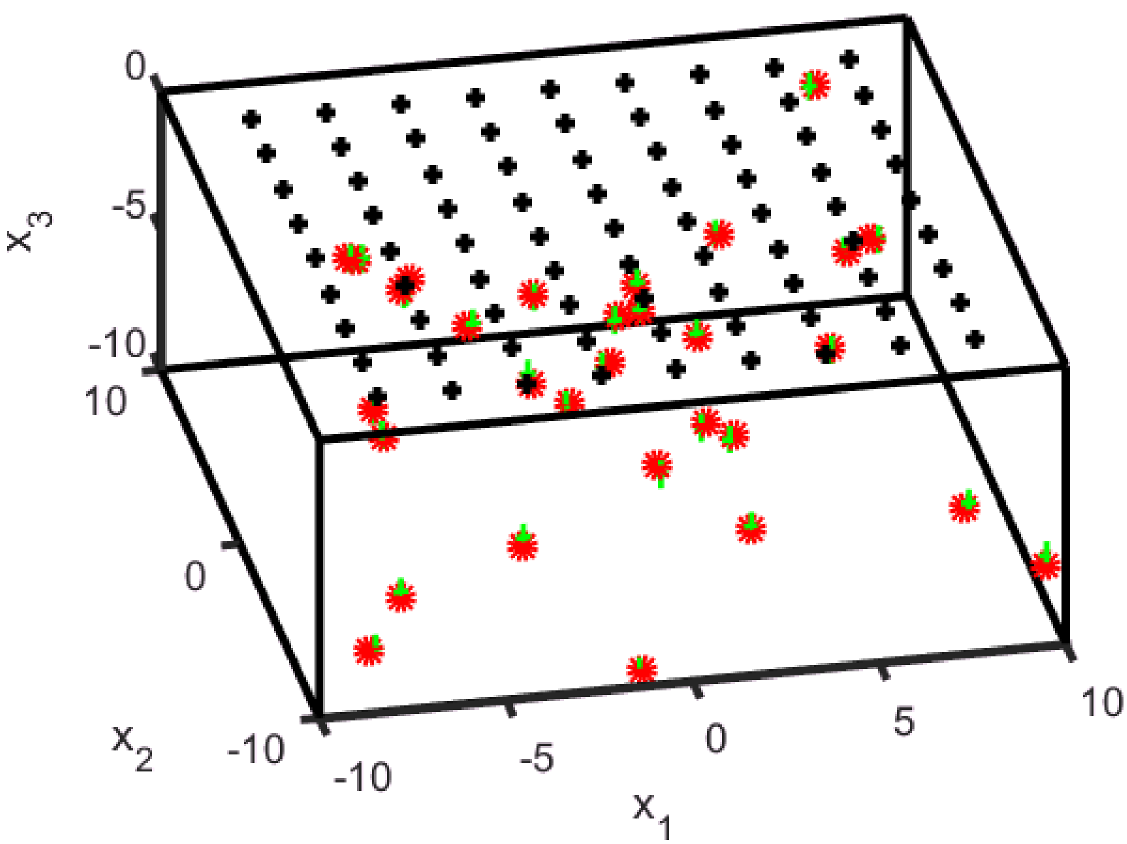
```

% final predicted data
dpre=zeros(N,1);
for i = [1:Ns] % loop over stations
for j = [1:Ne] % loop over earthquakes
    dx = mest(j)-sx(i);
    dy = mest(Ne+j)-sy(i);
    dz = mest(2*Ne+j)-sz(i);
    r = sqrt( dx^2 + dy^2 + dz^2 );
    k=(i-1)*Ne+j;
    dpre(k)=r/vp+mest(3*Ne+j);
    dpre(Nr+k)=r/vs+mest(3*Ne+j);
end
end

% display summary of results
expre = mest(1:Ne);
eypre = mest(Ne+1:2*Ne);
ezpre = mest(2*Ne+1:3*Ne);
t0pre = mest(3*Ne+1:4*Ne);
dd = dobs-dpre;
E = dd'*dd;
fprintf('RMS traveltime error: %f\n', sqrt(E/N) );
```

```
RMS traveltime error: 0.099139
```

```
Emx = (extrue-expre)'*(extrue-expre);
Emy = (eytrue-eypre)'*(eytrue-eypre);
Emz = (eztrue-ezpre)'*(eztrue-ezpre);
Emt = (t0true-t0pre)'*(t0true-t0pre);
fprintf('RMS model misfit: x %f y %f z %f t0 %f\n', sqrt(Emx/Ne), sqrt(Emy/Ne), sqrt(Emz/Ne),
```

```
RMS model misfit: x 0.111472 y 0.106997 z 0.167814 t0 0.029797
```

```
% plot results in 3D space
plot3( extrue, eytrue, eztrue, 'ro', 'LineWidth', 4 );
dx = 0.2;
dy = 0.2;
dz = 0.5;
for i = [1:Ne]
    plot3( [expre(i)-dx, expre(i)+dx]', [eypre(i), eypre(i)]', [ezpre(i), ezpre(i)]', 'g-', 'L
    plot3( [expre(i), expre(i)]', [eypre(i)-dy, eypre(i)+dy]', [ezpre(i), ezpre(i)]', 'g-', 'L
    plot3( [expre(i), expre(i)]', [eypre(i), eypre(i)]', [ezpre(i)-dz, ezpre(i)+dz]', 'g-', 'L
end
view(-12,52);
```




```
% Figure 12.19 Earthquake location example. Arrival times of P and S waves from earthquakes
% (red circles) are recorded on an array of 81 stations (black crosses). The observed arrival
% times include random noise with variance σd2 = (0.1)2s. The estimated locations (green cross
% are computed using Geiger's method. MatLab script gda12_13.
```