```
% gda12_05
% Supports Figure 12.7
% a simple tomography example with a 2D model m(x,y)
% the imaged region is a rectangle, the rays are straight lines
% sources and receivers are colocated on three of the four
% edges of the rectangle, a situation which is perhaps analagous
% to seismic tomography problems

clear all;

clear G epsilon;
global G epsilon;

figure(1);
clf;

% read true model from a file
Sbig=double(imread('../data/magma.tif','tif'));
Sbig = -(Sbig-128);
[Nybig, Nxbig] = size(Sbig);

% decimate the model
idec=4;
Nx=floor(Nxbig/idec);
Ny=floor(Nybig/idec);
S=Sbig(1:idec:Nybig-1, 1:idec:Nxbig-1);

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

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

% rearrage model into vector
% order or model parameters in vector: index = (ix-1)*Ny + iy
mtrue = zeros(Nx*Ny,1);
rowindex = zeros(Nx*Ny,1);
colindex = zeros(Nx*Ny,1);
for ix = [1:Nx]
for iy = [1:Ny]
    q = (ix-1)*Ny + iy;
    mtrue(q)=S(ix,iy);
    rowindex(q)=ix;
    colindex(q)=iy;
end
end

% plot true model
subplot(2,2,1);
set(gca, 'YDir', 'reverse' );
set(gca, 'LineWidth', 3 );
```

```
set(gca, 'FontSize', 14 );
colormap('jet');
hold on;
axis( [xmin, xmax, ymin, ymax] );
caxis( [-128, 128] );
image( [xmin, xmax], [ymin, ymax], S, 'CDataMapping', 'scaled');
hold on;
xlabel('y');
ylabel('x');
title( 'true model' );

% construct sources and receivers on 3 sides of the model
Nside = 25;
Ns = 3*Nside;
xs = zeros(Ns, 2);
% side 1
xs(1:Nside,1)=xmin+dx/10;
xs(1:Nside,2)=ymin+Dy*[1:Nside]'/(Nside+1);
% side 2
xs(1*Nside+1:2*Nside,1)=xmin+Dx*[1:Nside]'/(Nside+1);
xs(1*Nside+1:2*Nside,2)=ymin+dy/10;
% side 3
xs(2*Nside+1:3*Nside,1)=xmin+Dx*[1:Nside]'/(Nside+1);
xs(2*Nside+1:3*Nside,2)=ymax-dy/10;

% plot receivers
plot( xs(:,2), xs(:,1), 'k+' );

% rays connect all source/receiver pairs
N = Ns*(Ns-1)/2;
ray = zeros(N,4);
k=1;
for i = [1:Ns]
    for j = [i+1:Ns]
        ray(k,1) = xs(i,1);
        ray(k,2) = xs(i,2);
        ray(k,3) = xs(j,1);
        ray(k,4) = xs(j,2);
        k=k+1;
    end
end

% plot true model and rays
subplot(2,2,2)
set(gca, 'YDir', 'reverse' );
set(gca, 'LineWidth', 3 );
set(gca, 'FontSize', 14 );
colormap('jet');
axis( [xmin, xmax, ymin, ymax] );
hold on;
caxis( [-128, 128] );
image( [xmin, xmax], [ymin, ymax], S, 'CDataMapping', 'scaled');
xlabel('y');
ylabel('x');
title( 'true model with rays' );
plot( xs(:,2), xs(:,1), 'k+' );
% plot every kRay ray, else diagram would be black
kRay=20;
for k = [1:kRay:N]
    plot( [ray(k,2) ray(k,4)], [ray(k,1), ray(k,3)], 'k-', 'LineWidth', 2 );
end
```

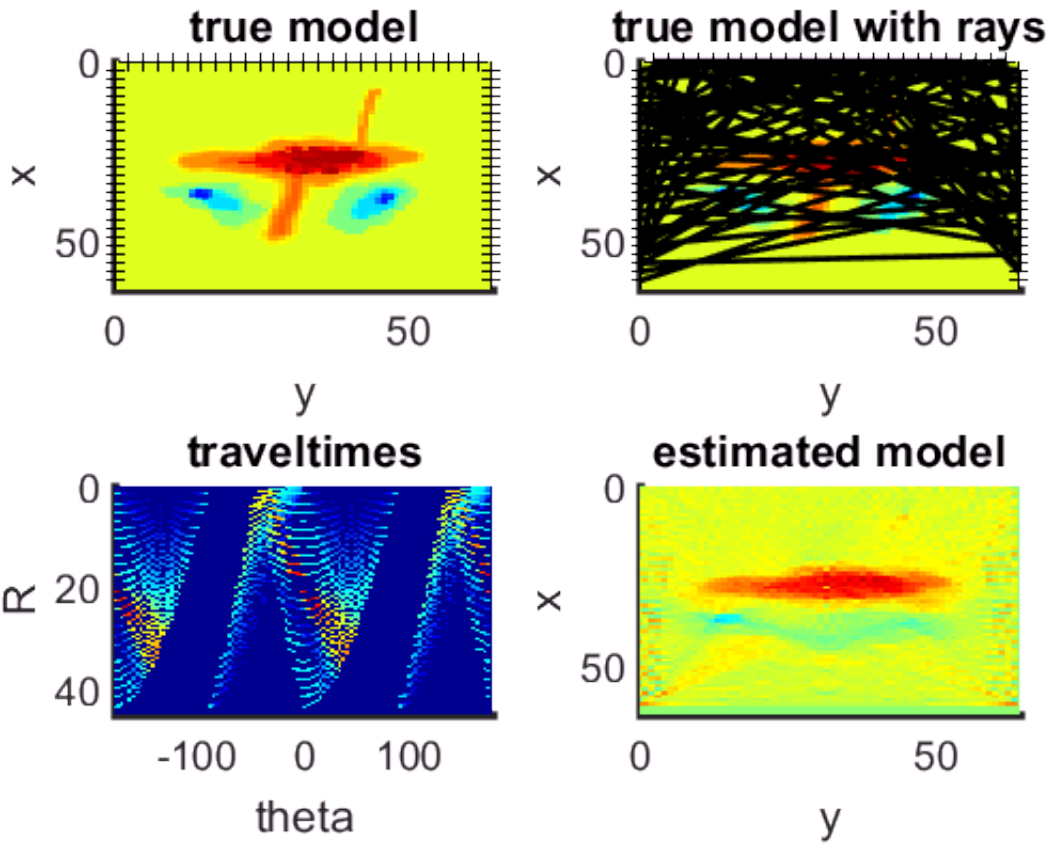
```
% build data kernel
% order or model parameters in vector: index = (ix-1)*Ny + iy
M = Nx*Ny;
G=spalloc(N,M,2*N*Nx);
Nr = 2000;
bins=[1:Nx*Ny];
for k = [1:N]
    x1 = ray(k,1);
    y1 = ray(k,2);
    x2 = ray(k,3);
    y2 = ray(k,4);
    r = sqrt( (x2-x1)^2 + (y2-y1)^2 );
    dr = r/Nr;
% I use a sloppy way of computing the length of the ray
% in each pixel.  I subdivide the ray into Nr pieces, and
% assign each piece to exactly one pixel, the one its
% closest to

if( 0 ) % slow but sure way
    for i = [1:Nr]
        x = x1 + (x2-x1)*i/Nr;
        y = y1 + (y2-y1)*i/Nr;
        ix = 1+floor( (Nx-1)*(x-xmin)/Dx );
        iy = 1+floor( (Ny-1)*(y-ymin)/Dy );
        q = (ix-1)*Ny + iy;
        G(k,q) = G(k,q) + dr;
    end
else % faster way, or so we hope
    % calculate the array indices of all the ray pieces
    xv = x1 + (x2-x1)*[1:Nr]'/Nr;
    yv = y1 + (y2-y1)*[1:Nr]'/Nr;
    ixv = 1+floor( (Nx-1)*(xv-xmin)/Dx );
    iyv = 1+floor( (Ny-1)*(yv-ymin)/Dy );
    qv = (ixv-1)*Ny + iyv;
    % now count of the ray segments in each pixel of the
    % image, and use the count to increment the appropriate
    % element of G.  The use of the hist() function to do
    % the counting is a bit weird, but it seems to work
    count=hist(qv,bins);
    icount = find( count~=0 );
    G(k,icount) = G(k,icount) + count(icount)*dr;
end
end

% compute true travel times and plot them
d = G*mtrue;

% plot true travel times on a 2D diagram
% arranged by the rays's closest approach to the origin
% and its orientation

% want to plot traveltimes on NR by Ntheta grid
% independent variable R
NR = 100;
Rmin = 0.0;
Rmax = sqrt((Dx/2)^2+(Dy/2)^2);
dR = (Rmax-Rmin)/(NR-1);

% independent variable theta
Ntheta = 100;
```

```
thetamin = -180.0;
thetamax = 180.0;
dtheta = (thetamax-thetamin)/(Ntheta-1);

% build traveltime grid for plotting purposes
TT = zeros(NR,Ntheta);
s = zeros(N,1);
R = zeros(N,1);
theta = zeros(N,1);
for k = [1:N]
    x1 = ray(k,1);
    y1 = ray(k,2);
    x2 = ray(k,3);
    y2 = ray(k,4);
    % compute closest approach to origin and angle
    xd = (x2-x1);
    yd = (y2-y1);
    s(k) = -(x1*xd+y1*yd)/(xd^2+yd^2);
    xs = x1+xd*s(k);
    ys = y1+yd*s(k);
    R(k) = sqrt( xs^2 + ys^2 );
    theta(k) = (180/pi)*atan2(ys,xs);
    iR = 1+round((R(k)-Rmin)/dR);
    if( iR<1 )
        iR=1;
    elseif ( iR>NR )
        ir=NR;
    end
    itheta = 1+round((theta(k)-thetamin)/dtheta);
    if( itheta<1 )
        itheta=1;
    elseif ( itheta>Ntheta )
        itheta=Ntheta;
    end
    TT(iR,itheta)=d(k);
    % out in data for reverse orientation of ray, for symmetery
    theta2 = (180/pi)*atan2(-ys,-xs);
    itheta = 1+round((theta2-thetamin)/dtheta);
    if( itheta<1 )
        itheta=1;
    elseif ( itheta>Ntheta )
        itheta=Ntheta;
    end
    TT(iR,itheta)=d(k);

end

% plot TT(R,theta)
subplot(2,2,3);
set(gca, 'LineWidth', 3 );
set(gca, 'FontSize', 14 );
colormap('jet');
hold on;
axis( [thetamin, thetamax, Rmin, Rmax]);
imagesc( [thetamin, thetamax], [Rmin, Rmax], TT);
set(gca, 'YDir', 'reverse' );
xlabel('theta');
ylabel('R');
title( 'traveltimes' );

% inversion of data by damped least squares
```

```
epsilon = 1.0e-2*max(max(abs(G)));
mest=bicg(@dlsfun,G'*d,1e-5,3*M);
```

```
bicg converged at iteration 150 to a solution with relative residual 9.1e-06.
```

```
% rearrange m back into image
Sest = zeros(Nx,Ny);
for k = [1:M]
    ix=rowindex(k);
    iy=colindex(k);
    Sest(ix,iy)=mest(k);
end

% plot estimated image
subplot(2,2,4);
set(gca, 'YDir', 'reverse' );
hold on;
set(gca, 'LineWidth', 3 );
set(gca, 'FontSize', 14 );
colormap('jet');
axis( [xmin, xmax, ymin, ymax] );
caxis( [-128, 128] );
image( [xmin, xmax], [ymin, ymax], Sest, 'CDataMapping', 'scaled');
xlabel('y');
ylabel('x');
title( 'estimated model' );
```




```
% Figure 12.7 Acoustic tomography problem with deficient distribution of rays. (A) True image.
% paths (only a few percent of rays are shown, else the image would be black). (C) Travel time
% organized by the distance r and angle θ of the ray from the midpoint of the image. (D) Recon
```

```
% image. See text for further discussion. MatLab script gda12_05.
```