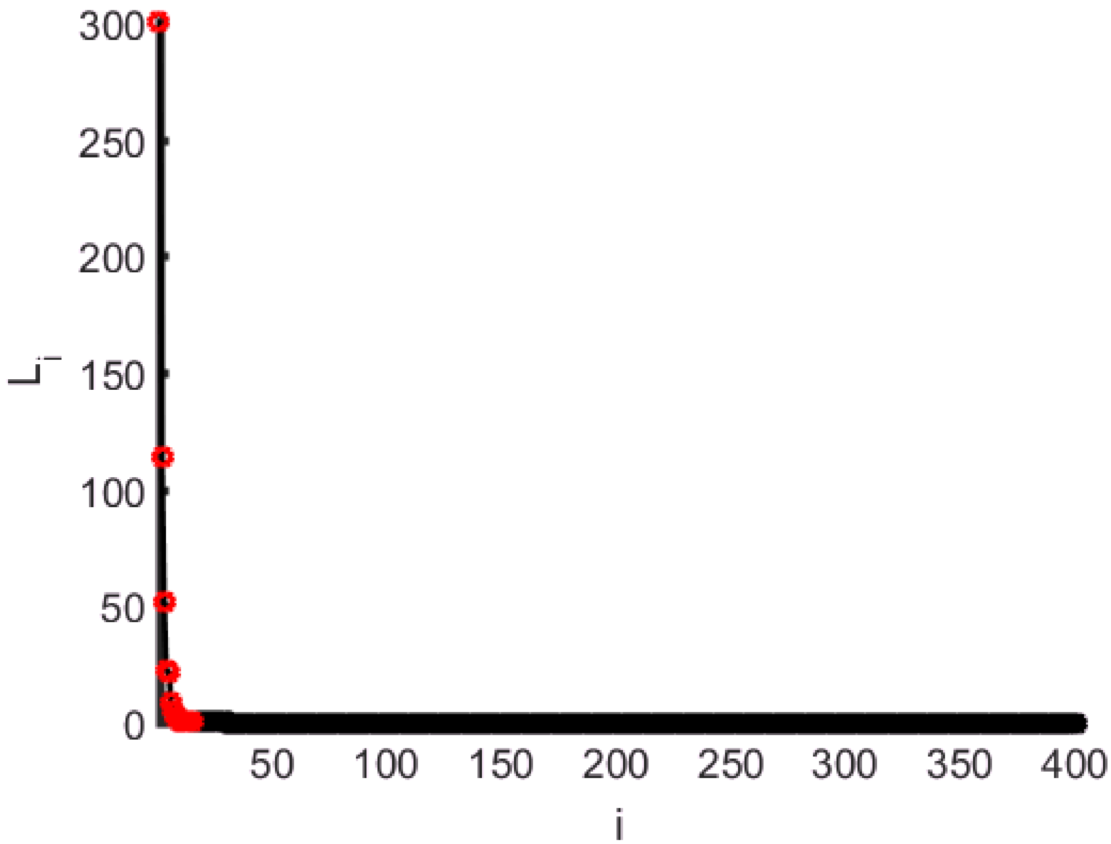
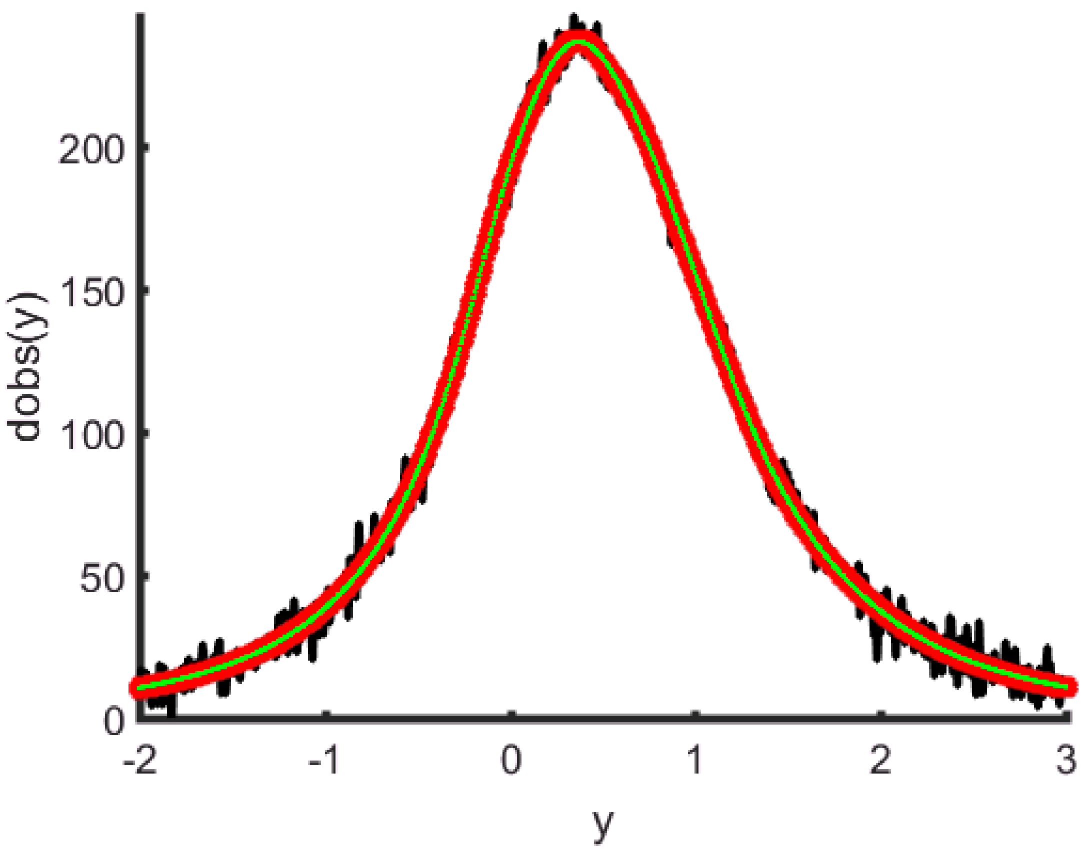
```
% gda07_06
% illustration of natural and non-negative least squares
% with gravity problem
% supports Figure 7.7

clear all;

% model is two-dimensional grid of pixels, each of which represents mass
% coorinate system, x is down, y is right, origin in upper-left
Nx = 20;
Ny = 20;
M=Nx*Ny;

% model x coordinate
xmin = 0.0;
xmax = 1.0;
Lx = xmax-xmin;
Dx = (xmax-xmin)/(Nx-1);
x = xmin + Dx*[0:Nx-1]';

% model y coordinate
ymin = 0.0;
ymax = 1.0;
Ly = ymax-ymin;
Dy = (ymax-ymin)/(Ny-1);
y = ymin + Dy*[0:Ny-1]';

% gravity data are on top surface, extending past edges of model by several
% model widths
N=30*Ny;
xo = -0.5; % observations well above surface of model
yomin=ymin-2*Ly;
yomax=ymax+2*Ly;
Lyo = (yomax-yomin);
Dyo = (yomax-yomin)/(N-1);
yo = yomin + Dyo*[0:N-1]';

% data kernel for vertical component of gravity
g = 1; % gravitational constant (not a relistic value)
G=zeros(N,M);
jofq=zeros(Nx,1); % backpointer to x(j) value of model parameters
kofq=zeros(Ny,1); % backpointer to y(k) value of model parameters
for i = [1:N]  % loop over observation points
    % loop over masses
    q = 0; % counts model parameters
    for j = [1:Nx] % x
    for k = [1:Ny] % y
        q = q+1;
        jofq(q)=j; kofq(q)=k;
        D2 = (x(j)-xo)^2 + (y(k)-yo(i))^2; % distance squared
        ct = (x(j)-xo) / sqrt(D2); % cosine of vertical angle
        G(i,q) = ct*g/D2;
    end
    end
end

% true model parameters; just something cooked up to look interesting
mxytrue = ([1:Nx]-Nx/2)'*ones(1,Ny)/Nx * ones(Nx,1)*([1:Ny]-Ny/2)/Ny;
mxytrue = mxytrue - min(min(mxytrue));
```

```
mxytrue = (mxytrue / max(max(mxytrue))) + random('uniform',0,0.1,Nx,Ny);
% convert to vector
mtrue=zeros(M,1);
for q = [1:M]
    mtrue(q) = mxytrue( jofq(q), kofq(q) );
end
% convert back to image as a check
% mxytrue2=zeros(Nx,Ny);
% for q = [1:M]
%     mxytrue2( jofq(q), kofq(q) ) = mtrue(q);
% end
% gda_draw(' ',mxytrue,' ',mxytrue2);

% create and plot synthetic data
sigmad=5;
dobs = G*mtrue + random('Normal',0.0,sigmad,N,1);
figure(2);
clf;
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on;
axis( [yomin, yomax, 0, max(dobs) ] );
plot( yo, dobs, 'k-', 'LineWidth', 3 );
xlabel('y');
ylabel('dobs(y)');

% calculate and plot singular values
[U, L, V] = svd(G);
lambda = diag(L);
p=15;
lambdap=lambda(1:p);
figure(3);
clf;
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on;
axis( [1, length(lambda), 0 max(lambda) ] );
plot( [1:length(lambda)]', lambda, 'k-', 'LineWidth', 2 );
plot( [1:length(lambda)]', lambda, 'ko', 'LineWidth', 2 );
plot( [1:p]', lambdap, 'ro', 'LineWidth', 2 );
xlabel('i');
ylabel('L_i');
```

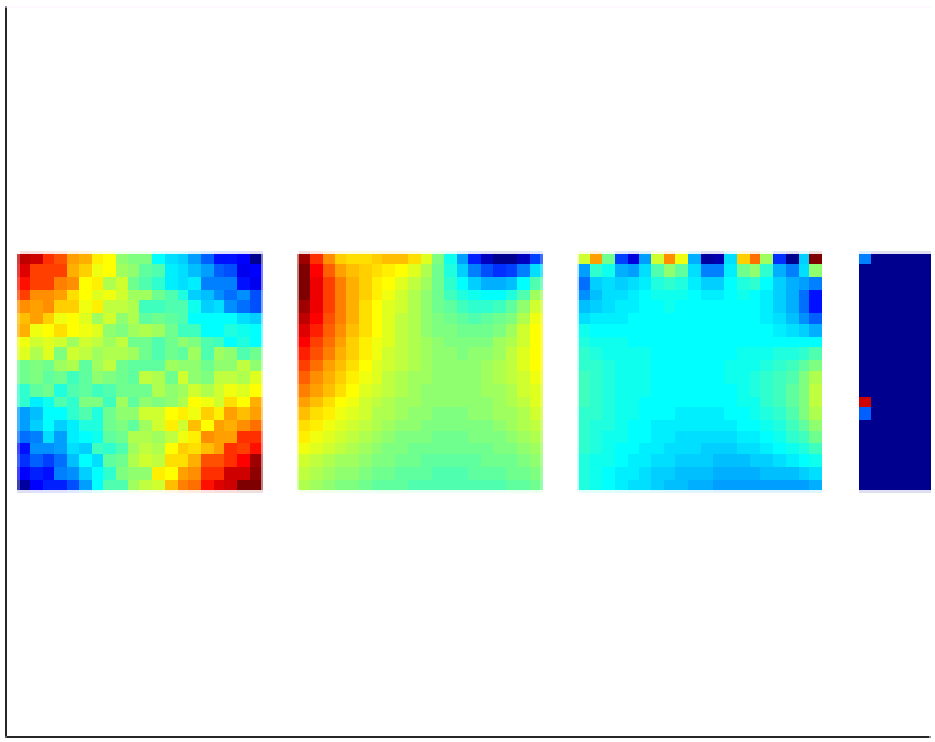



```
% Figure. Singular valuee. The cut-off p=15 was selected.

% natural solution
Up=U(:,1:p);
Vp=V(:,1:p);
lambdap=lambda(1:p);
mestN = Vp*((Up'*dobs)./lambdap);
% convert to image
mxyestN=zeros(Nx,Ny);
for q = [1:M]
    mxyestN( jofq(q), kofq(q) ) = mestN(q);
end

% natural solution, smaller p
p=4;
Up=U(:,1:p);
Vp=V(:,1:p);
lambdap=lambda(1:p);
mestN2 = Vp*((Up'*dobs)./lambdap);
% convert to image
mxyestN2=zeros(Nx,Ny);
for q = [1:M]
    mxyestN2( jofq(q), kofq(q) ) = mestN2(q);
end

% predicted data
dpreN = G*mestN;
figure(2);
plot( yo, dpreN, 'ro', 'LineWidth', 2 );
eN = dobs-dpreN;
EN=eN'*eN;
```

```
% non-negative splution
mestNN = lsqnonneg(G,dobs);
% convert to image
mxyestNN=zeros(Nx,Ny);
for q = [1:M]
    mxyestNN( jofq(q), kofq(q) ) = mestNN(q);
end

% predicted data
dpreNN = G*mestNN;
figure(2);
plot( yo, dpreNN, 'g.', 'LineWidth', 2 );
```




```
% Figure 7.7 (B) The N = 600 gravitational force observations
% dobs (black curve) and the gravitational force predicted by the natural solution
% (red curve, p = 15) and nonnegative least squares (green curve). (C) True model.
% (D) Natural estimate of model, with p = 4. (E) Natural estimate of model, with p = 15.
% (F) Nonnegative estimate of model. MatLab script gda07_06.

eNN = dobs-dpreNN;
ENN=eNN'*eNN;

gda_draw(' ',mxytrue,' ',mxyestN2,' ',mxyestN,' ',mxyestNN);
```



```
% Figure 7.7 (C) True model (D) Natural estimate of model, with p = 4. (E) Natural estimate
% of model, with p = 15 (F) Nonnegative estimate of model. MatLab script gda07_06.

fprintf('Total Error: Natural %f   Non-Negative %f\n',EN,ENN);
```

```
Total Error: Natural 15272.903468   Non-Negative 15351.197038
```