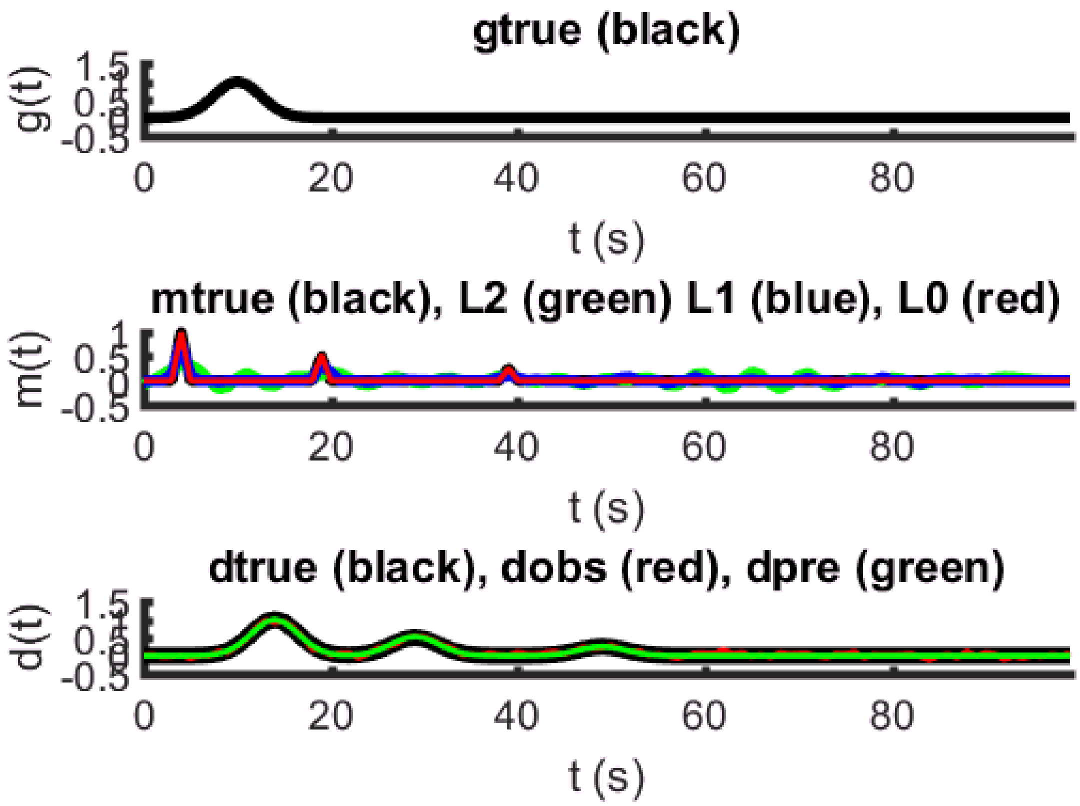
```
clear all;
% gda08_06
% supports Figure 8.6

% An example of solving a minimum length problem under
% the L0 and L1 norms, by the itertive reweighting algorithm
% The example chosen here is
% d(t) = g(t) * m(t) where * is convolution
% The convolution is represented by a Toeplitz matrix G

% time variable t
N=100;
M=N;
t = [0:N-1]';

% a gaussian pulse g(t)
t0 = 10;
s = 2.5;
g = exp( -((t-t0).^2)/(2*s^2) );  % gaussian g(t)

% matrix equivalent to convolution by g
G = toeplitz( g, [g(1), zeros(1,N-1)] );

% true solution is a collection of spikes
mtrue = zeros(N,1);
mtrue(5)=1; mtrue(20)=0.5; mtrue(40)=0.25;

dtrue = G*mtrue; % true data

% observed data is true data plus random noise
sd = 0.05;
dobs = dtrue+random('Normal',0,sd,N,1);

% setup for L1 norm
n = 1;
delta = 1e-5;
gamma = 2;
mu = 0.01;
w = ones(M,1);
for j=[1:10]  % reweighting algorithm
    mest = (G'*G + mu*diag(w))\(G'*dobs);
    dpre = G*mest;
    e = dobs-dpre;
    E = e'*e;
    if( j==1 ) % save L2 solution
        mL2 = mest;
        EL2 = E;
    end
    am = abs(mest); % Frommlet & Nuel's (2016) weight formula
    for k=[1:M]
        if( am(k)>=delta )
           w(k)=(delta^(n-2))*exp(((n-2)/gamma)*log1p((am(k)/delta)^gamma));
        else
           w(k)=(am(k)^(n-2))*exp(((n-2)/gamma)*log1p((delta/am(k))^gamma));
        end
    end
end
mL1 = mest;
EL1 = E;
```

```
% setup for L0 norm (actually 0.1 norm)
n=0.1;
mu = 0.01;
epsi = 1.0e-6;
w = ones(M,1);
for j=[1:10] % reweighting algorithm
    mest = (G'*G + mu*diag(w))\(G'*dobs);
    dpre = G*mest;
    e = dobs-dpre;
    E = e'*e;
    am = abs(mest); % Frommlet & Nuel's (2016) weight formula
    for k=[1:M]
        if( am(k)>=delta )
           w(k)=(delta^(n-2))*exp(((n-2)/gamma)*log1p((am(k)/delta)^gamma));
        else
           w(k)=(am(k)^(n-2))*exp(((n-2)/gamma)*log1p((delta/am(k))^gamma));
        end
    end
end
mL0 = mest;
EL0 = E;

fprintf('EL2/N %f EL1/N %f EL0/N %f sd %f\n', EL2, EL1, EL0, sd );
```

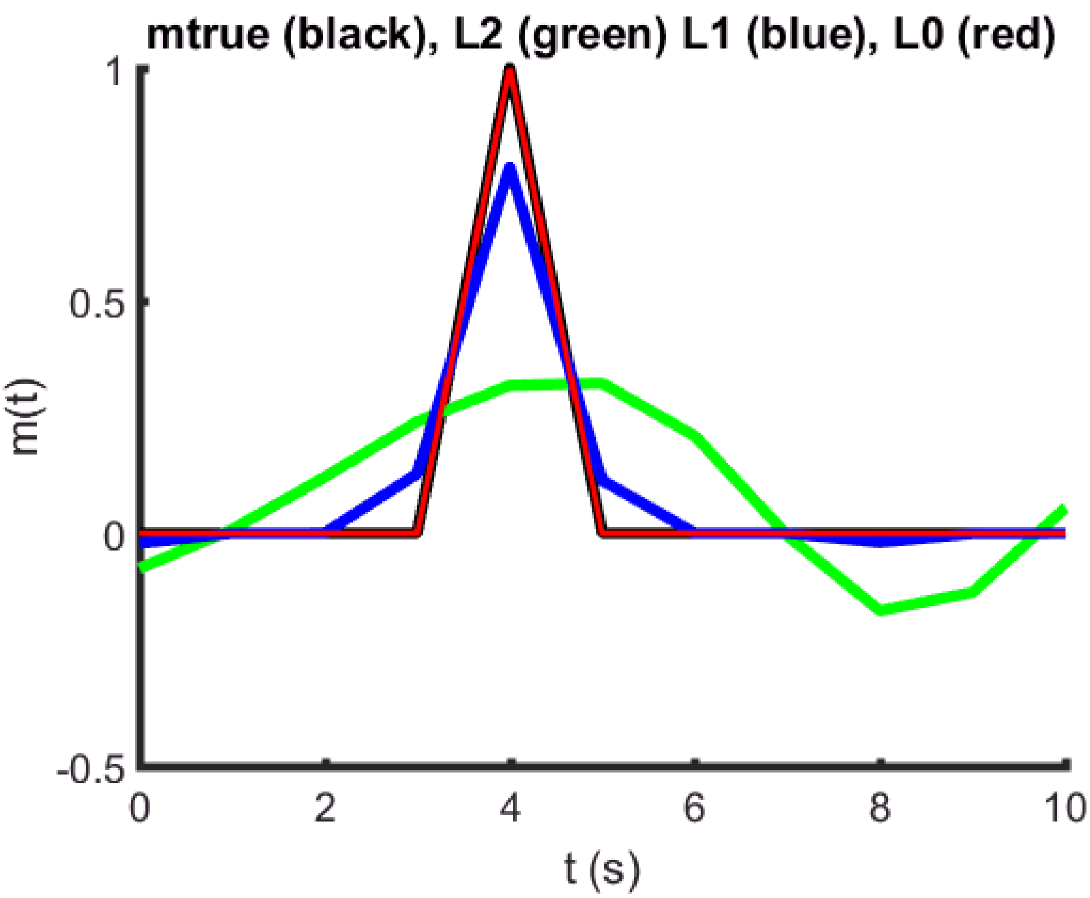
```
EL2/N 0.161551 EL1/N 0.179273 EL0/N 0.284687 sd 0.050000
```

```
figure(1);
clf;
subplot(3,1,1); % plot g(t)
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on;
axis( [0, N-1, -0.5, 1.5] );
xlabel('t (s)');
ylabel('g(t)');
title('gtrue (black)');
plot( t, g, 'k-', 'LineWidth', 4);

subplot(3,1,3); % plot d(t)
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on;
axis( [0, N-1, -0.5, 1.5] );
xlabel('t (s)');
ylabel('d(t)');
title('dtrue (black), dobs (red), dpre (green)');
plot( t, dtrue, 'k-', 'LineWidth', 6);
plot( t, dobs, 'r-', 'LineWidth', 2);
plot( t, dpre, 'g-', 'LineWidth', 2);

subplot(3,1,2); % plot m(t)
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on;
axis( [0, N-1, -0.5, 1.0] );
xlabel('t (s)');
ylabel('m(t)');
title('mtrue (black), L2 (green) L1 (blue), L0 (red)');
```

```
plot( t, mtrue, 'k-', 'LineWidth', 4);
plot( t, mL2, 'g-', 'LineWidth', 4);
plot( t, mL1, 'b-', 'LineWidth', 4);
plot( t, mL0, 'r-', 'LineWidth', 2);
```




```
% Figure 8.6 The filter problem d=g*m with a spiky filter m. (A) The time series g is a
% smooth pulse. (B) The true filter m^true (black) is sparse, with three widely separated
% spikes.  The estimated the L_2 (green), L_1 (blue) and L_0 (red) filters have increasing
% spikiness.  (C) All three filters (green) fit data (black).

figure(2); % close up of one of the spikes
clf;
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on;
axis( [0, 10, -0.5, 1.0] );
xlabel('t (s)');
ylabel('m(t)');
title('mtrue (black), L2 (green) L1 (blue), L0 (red)');
plot( t, mtrue, 'k-', 'LineWidth', 4);
plot( t, mL2, 'g-', 'LineWidth', 4);
plot( t, mL1, 'b-', 'LineWidth', 4);
plot( t, mL0, 'r-', 'LineWidth', 2);
```




```
% Figure 8.6 The filter problem d=g*m with a spiky filter m. (D) Enlargement of the shaded
% region in C. MatLab  gda08_06.
```