

```

% gda03_06
% least squares example with
% sparse matrices and
% bicg() solver
% reconstruct function using noisy observations of its
% derivative and integral

% Supports Section 3.4

clear all;

% need these two commands to use bconj() biconjugate gradient solver
clear G;
global G;

% true data is a sine wave
M=101;
Dz = 1.0;
z = Dz*[0:M-1]'; % auxilliary variable
zmax = max(z);
mtrue = sin( 4*pi*z/zmax );

% let us suppose that we observe both the derivative and intergal of
% the function

% first derivative, each row looks like (1/Dz)*[ 0 ... 0 0 -1 1 0 0 ... 0 ]
N = (M-1) + M;
MAXEL = 2*M+M*M;
G = spalloc( N, M, MAXEL );
for i = [1:M-1]
    G(i,i) = -1/Dz;
    G(i,i+1) = 1/Dz;
end

% integral, each row looks like Dz*[ 1 1 1 ... 1 0 0 0 0 ... 0]
for i = [1:M]
    for j = [1:i]
        G(i+M-1,j)=Dz;
    end
end

% synthetic data, including a little random noise
sigmad = 0.3;
dobs = G*mtrue + random('Normal',0,sigmad,N,1);

% least squares solution #1, using normal matrix arithmetic
mest1 = (G'*G)\(G'*dobs);

% least squares solution #2, using bicg()
tol = 1e-6;
maxit = 3*N;
mest2 = bicg( @leastsquaresfcn, G'*dobs, tol, maxit );

```

bicg converged at iteration 20 to a solution with relative residual 7.2e-07.

```

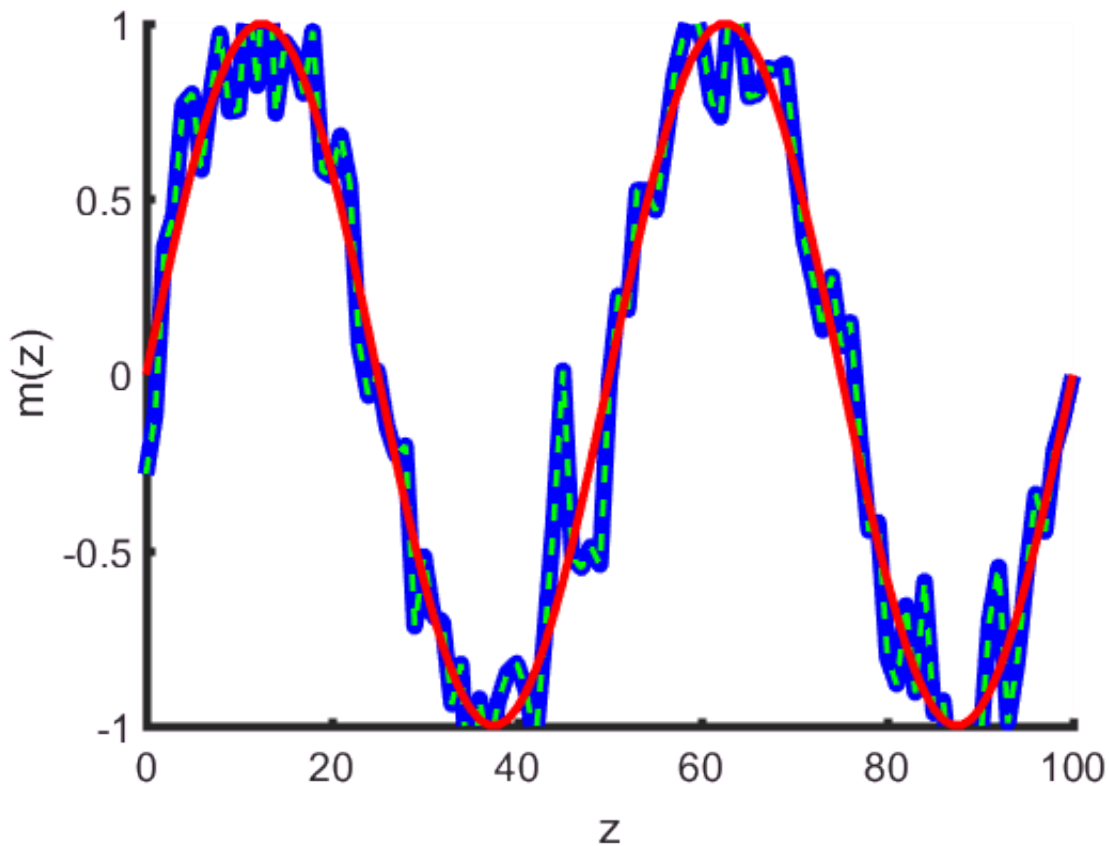
% plot
figure(1);
clf;

```

```

set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on;
axis( [0, zmax, -1, 1] );
plot( z, mest1, 'b-', 'LineWidth', 6 );
plot( z, mest2, 'g--', 'LineWidth', 2 );
plot( z, mtrue, 'r-', 'LineWidth', 3 );
xlabel('z');
ylabel('m(z)');

```



```

% Figure. Reconstruction of a function from noisy observations of its intergal and its
% derivative (which corresponds to a sparse data kernel). True model is a sine wave.
% The standard solver (blue) and bicg() solver with sparse matrices (green) give the
% same solution.

```