```
% gda12_02
% deconvolution example, d=g*m and m=ginv*d
% Supports Figure 12.3

% this uses generalized least squares
% the data equation is G m = rhs
% the constraint equation is H m = h
% the combined equations are Fm = f
% solution uses the bicg() solver set up to
% solve FTF = FT f, but in a clever way so that
% FTF, G and f are never explicitly

% however, the code also implements the brute
% force solution m = (FTF)\(FT f) inside if statements
% for test purposes

clear all;

clear g H;
global g H;

D=load('../data/airgun.txt');
% digitized airgun pulse, data after:
% Smith, SG,
% Measurement of Airgun Waveforms,
% Geophys. J. R. astr. Soc. (1975) 42, 273-280.

t=D(:,1);  % time
d=D(:,2);  % this is the filter for which we seek an inverse filter
Mo = length(t);
tmin = t(1);
Dt = t(2)-t(1);

% pad with zeros
M=2*Mo;
t = tmin+Dt*[0:M-1];
g = [ d', zeros(1,M-Mo) ]';
tmax = t(M);

% pad with zeros
M=length(t);
N=M; % inverse filter same length as filter

figure(1);
clf;

% plot g
subplot(3,1,1);
set(gca,'LineWidth',2);
hold on;
gmax = max(abs(g));
axis( [tmin, tmax, -gmax, gmax] );
plot( t, g, 'k-', 'LineWidth', 3 );
xlabel('time, t');
ylabel('g(t)');

% don't need G explicitly, but here it is for test purposes
test=1;
if( test )
```

```
    G=zeros(N,M);
    for j=[1:M]
        G(j:N,j)=g(1:N-j+1);
    end
end

% set up H and h
K=2*M;
L=N+K;
H=spalloc(K,M,3*L);

% two sets of constaint equations, curvature and length
% epsilon parameters adjusted empirically
% curvature
e=0.001*max(abs(g));
for j = [2:M]
    H(j,j-1)=e;
end
for j = [1:M]
    H(j,j)=-2*e;
end
for j = [1:M-1]
     H(j,j+1)=e;
end
% damp size, especially towards the end
e2=0.1*max(abs(g));
for j = [1:M]
    H(j+M,j)=e2*sqrt(j);
end

% rhs of data equaltion is a spike delayed by Nd samples
% (a little delay sometimes leads to a better result)
rhs=zeros(N,1);
Nd=20;
rhs(Nd+1)=1;

% rhs of constraint equation is zero
h=zeros(K,1);

if( test ) % don't need F, but here it is for test purposes
    F=[ G', H' ]';
end

if( test ) % don't need f, but here it is for test purposes
    f=zeros(L,1);
    f(1:N)=rhs;
    f(N+1:N+K)=h;
end

if( test ) % don't need brute force solution, but here it is for test purposes
    ginvbf = (F'*F)\(F'*f);
end

% don't need GT*rhs, but here it is for test purposes
if( test )
    Gtrhs = G'*rhs;
end

% set up F'f = GT rhs + HT h
% note that GT v is similar to the convolution G v
% except that g is time-reversed and position of
```

```
% results in convolution is shifted to the bottom
temp=conv(flipud(g),rhs);
FTfa = temp(N:N+M-1);
FTfb = H'*h;
FTf=FTfa+FTfb;

% solve, filterfun does the (FT F v) multiplication cleverly
% so that G, GTG, HTH and f never computed
ginv=bicg(@filterfun,FTf,1e-10,3*L);
```

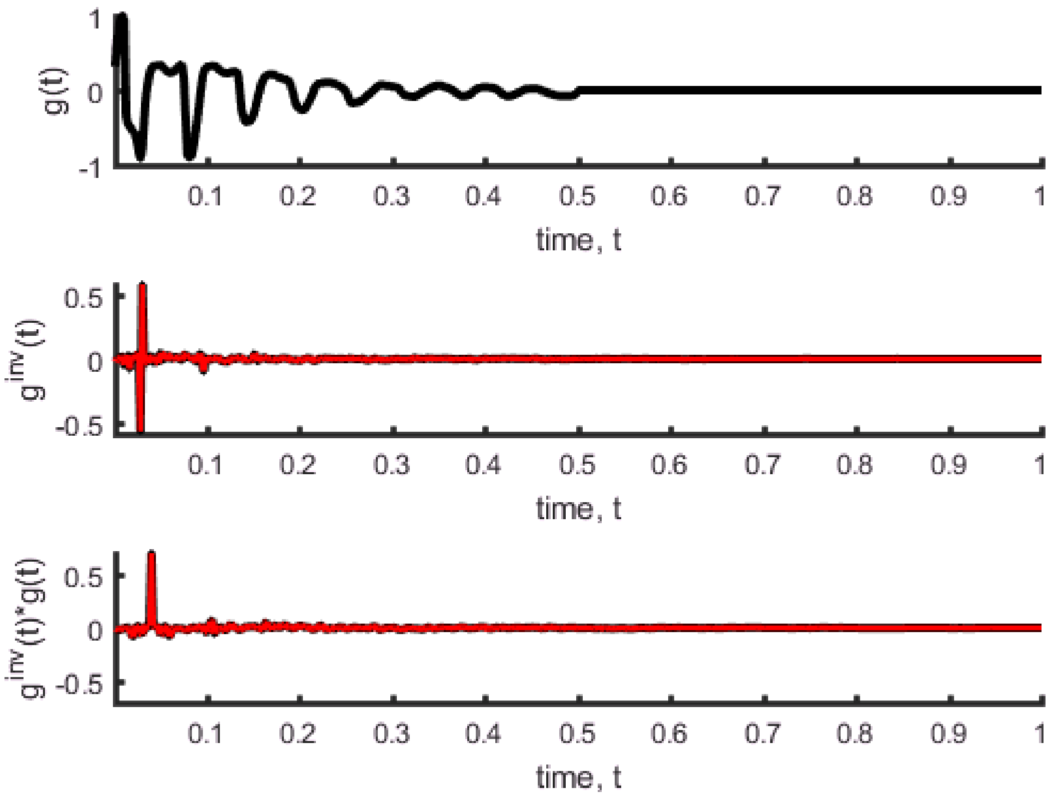
```
bicg converged at iteration 247 to a solution with relative residual 9.6e-11.
```

```

% don't need G*ginv, but here it is for test purposes
if( test )
    Gginv = G*ginvbf;
end

% plot inverse filter
subplot(3,1,2);
set(gca,'LineWidth',2);
hold on;
ginvmax = max(abs(ginv));
axis( [tmin, tmax, -ginvmax, ginvmax] );
plot( t, ginv, 'k-', 'LineWidth', 3 );
if( test ) % plot brute force result
    plot( t, ginvbf, 'r-', 'LineWidth', 2 );
end
xlabel('time, t');
ylabel('g^{inv}(t)');

% plot convolution of inverse filter and filter
c = conv( ginv, g );
c = c(1:N);
subplot(3,1,3);
set(gca,'LineWidth',2);
hold on;
cmax = max(abs(c));
axis( [tmin, tmax, -cmax, cmax] );
plot( t, c, 'k-', 'LineWidth', 3 );
if( test ) % plot brute force result
    plot( t, Gginv, 'r-', 'LineWidth', 2 );
end
xlabel('time, t');
ylabel('g^{inv}(t)*g(t)');
```




```
% Figure 12.3 (A) An airgun signal g(t), after Smith (1975). Ideally, the inverse filter ginv(
% when convolved with g(t) should produce the spike δ(t - t0), centered at time t0. (B) Estimat
% of the inverse filter ginv(t) for t0 = 0.04, computed via generalized least squares with pri
% information on solution size and smoothness. (C) The convolution of g(t) with the estimated
% ginv(t). While not a perfect spike, the result is significantly spikier than the airgun sign
% g(t). MatLab script gda12_02.

% error
E = (rhs-c)'*(rhs-c);
fprintf('damping factors %f %f   prediction error %f\n', e, e2, E );
```

```
damping factors 0.001000 0.100000   prediction error 0.164462
```

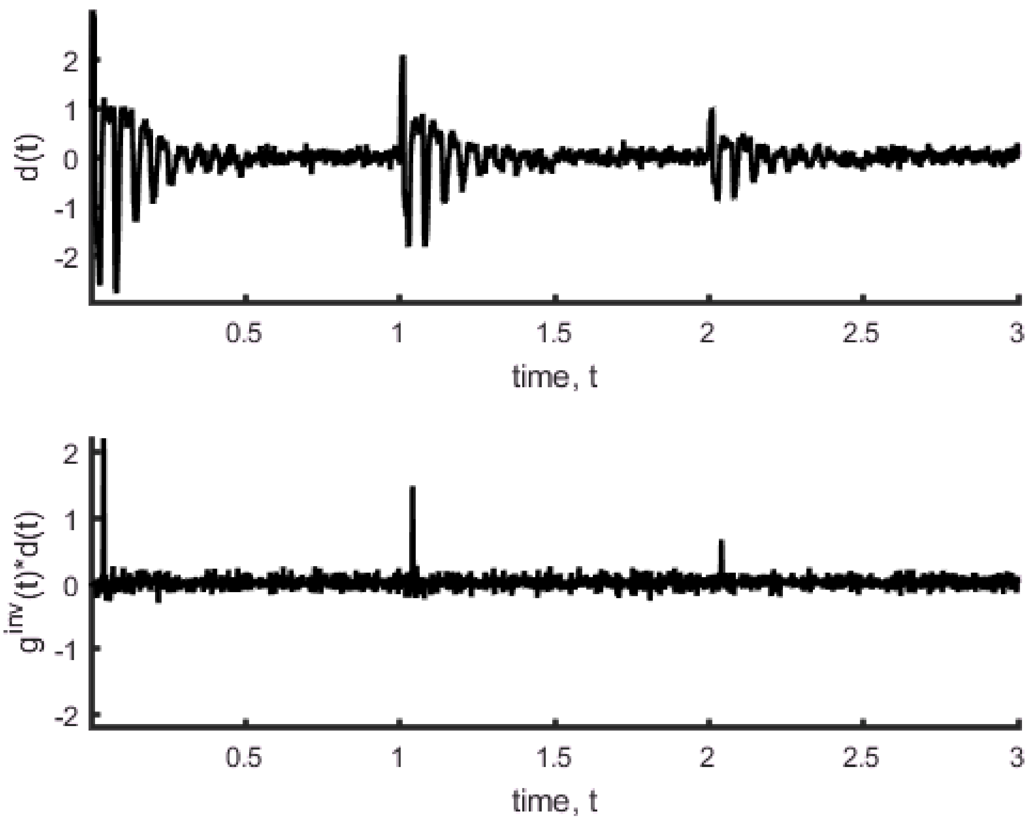
```
% as an example, deconvolve three pulses with additive noise
figure(2);
sd=0.1;
glong = [3*g', 2*g', g']' + random('Normal',0,sd, 3*M, 1);
clong = conv( ginv, glong );
clong = clong(1:3*N);

figure(2)
clf;

% plot data
subplot(2,1,1);
set(gca,'LineWidth',2);
hold on;
glongmax = max(abs(glong));
axis( [tmin, tmin+3*tmax, -glongmax, glongmax] );
```

```
plot( tmin+Dt*[1:3*M]', glong, 'k-', 'LineWidth', 2 );
xlabel('time, t');
ylabel('d(t)');

% plot deconvolved data
subplot(2,1,2);
set(gca,'LineWidth',2);
hold on;
clongmax = max(abs(clong));
axis( [tmin, tmin+3*tmax, -clongmax, clongmax] );
plot( tmin+Dt*[1:3*M]', clong, 'k-', 'LineWidth', 2 );
xlabel('time, t');
ylabel('g^{inv}(t)*d(t)');
```




```
% Figure. (A) sequence of three airgun pulses. (B) The signal after application
% of the inverse filter has three distinct spikes. MatLab script gda12_02.
```