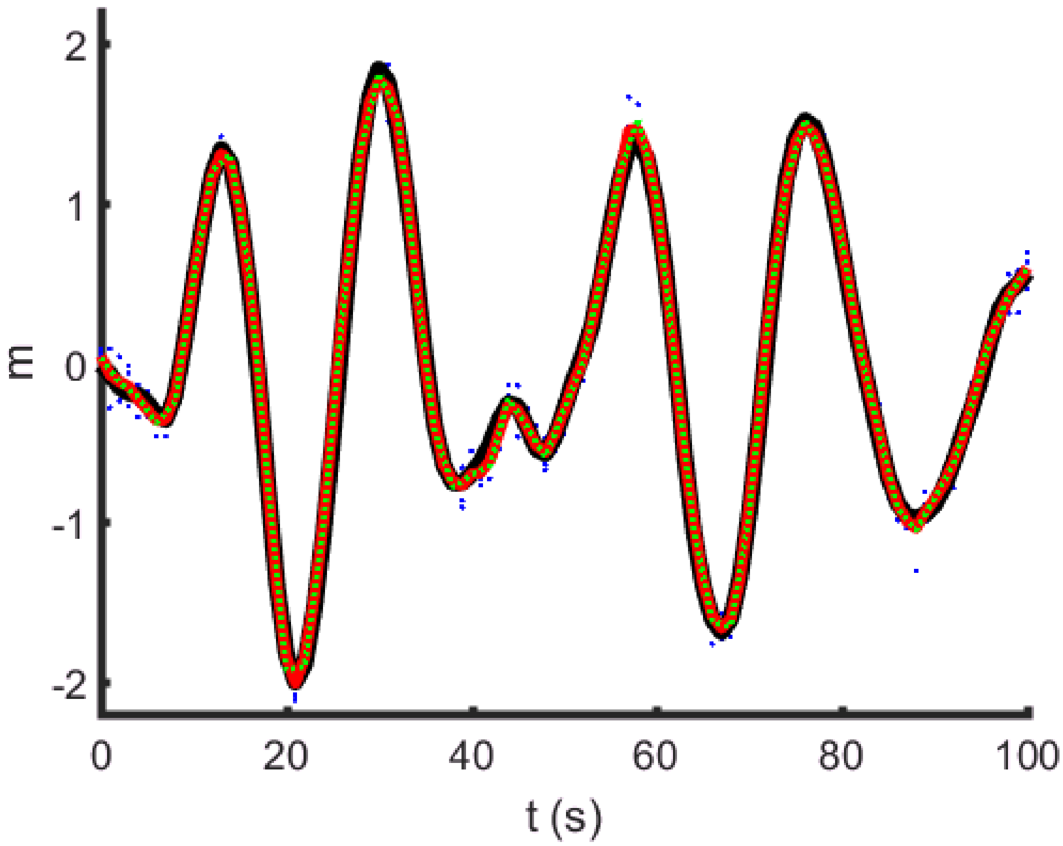
```
% gda05_19
% supports Figure 5.19
% example of F test for problem with prior information

clear all;

% dense set of model parameters
M1=101; % must be odd

% densely sampled time t variable
Dt1=1;
t1 = Dt1*[0:M1-1]';

% sparse set of model parameters
M2=(M1+1)/2;

% sparsely sampled time t variable
Dt2=2*Dt1;
t2 = Dt2*[0:M2-1]';

% matrix D2S takes dense to sparse model parameters by decimation
% m2 = D2S*m1
D2S = zeros(M2, M1);
j=1;
for i=[1:M2]
    D2S(i,j)=1.0;
    j=j+2;
end

% matrix S2D takes sparse to dense model parameters by linear interpolation
% m1 = S2D*m2
S2D = zeros(M1, M2);
j=1;
for i=[1:2:M1]
    S2D(i,j)=1;
    j=j+1;
end
j=1;
for i=[2:2:M1-1]
    S2D(i,j)=0.5;
    S2D(i,j+1)=0.5;
    j=j+1;
end

% make a densely sampled wiggly curve m(t) by starting out with random noise
% and bandpass filtering it around a narrow range of angular
% frquencies centered on w0.  The bandpass filtering is
% accomplished by the gda_chebyshevfilt() function, which
% though not mentioned in the text (sorry about that) is pretty
% easy to use and has many other applications.  Its arguments are
% output_timeseries = gda_chebyshevfilt( input_timeseries, Dt, f1, f2 )
% where Dt is the sampling of the timeseries (say in s) and
% (f1,f2) are the (low,high) size of the frequency band (say in Hz)
n = 10;
A = 2.0;
w0 = n*pi/t1(end);
Dw = w0/10;
mtrue1 = gda_chebyshevfilt( random('Normal',0,1,M1,1), Dt1, (w0-Dw)/(2*pi), (w0+Dw)/(2*pi) );
mtrue1 = A*mtrue1/max(abs(mtrue1));
```

```
% data kernel is just 3 independent observations of same point
N = 3*M1;
G1 = [eye(M1,M1); eye(M1,M1); eye(M1,M1)];
G2 = [S2D; S2D; S2D];
dtrue = G1*mtrue1;

% observed data is noisy
sigma_d = 0.1;
dobs = dtrue + random('Normal',0,sigma_d,N,1);
sqrtcovdi = eye(N,N)/sigma_d;

% prior information: second derivative close to zero
K1 = M1-2;
H1 = (1/Dt1^2)*toeplitz( [1; zeros(M1-3,1)], [1, -2, 1, zeros(1,M1-3) ] );
h1 = zeros(M1-2,1);
K2 = M2-2;
H2 = (1/Dt2^2)*toeplitz( [1; zeros(M2-3,1)], [1, -2, 1, zeros(1,M2-3) ] );
h2 = zeros(M2-2,1);

% prior covariance of model parameters
sigma_mdd = (w0^2)*A/sqrt(2);
sqrtcovhi1 = eye(K1,K1)/sigma_mdd;
sqrtcovhi2 = eye(K2,K2)/sigma_mdd;

% overall least squares equation and its solution
% dense problem
F1 = [ sqrtcovdi*G1; sqrtcovhi1*H1 ];
f1 = [ sqrtcovdi*dobs; sqrtcovhi1*h1 ];
FTFinv1 = inv(F1'*F1);
mest1 = FTFinv1*(F1'*f1);
% sparse problem
F2 = [ sqrtcovdi*G2; sqrtcovhi2*H2 ];
f2 = [ sqrtcovdi*dobs; sqrtcovhi2*h2 ];
FTFinv2 = inv(F2'*F2);
mest2 = FTFinv2*(F2'*f2);

% plot m
figure(1);
clf;
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on;
axis( [0, t1(end), -1.1*A, 1.1*A] );
plot( t1, dobs(1:M1), 'b.', 'LineWidth', 2 );
plot( t1, dobs(M1+1:2*M1), 'b.', 'LineWidth', 2 );
plot( t1, dobs(2*M1+1:3*M1), 'b.', 'LineWidth', 2 );
plot( t1, mtrue1, 'k-', 'LineWidth', 6 );
plot( t1, mest1, 'r-', 'LineWidth', 4 );
plot( t2, mest2, 'g:', 'LineWidth', 2 );
xlabel('t (s)');
ylabel('m');
```




```
% Figure. Model parameters m(t) (solid curve) and data d(t) (dots) are functions of
% an independent variable t.  The inverse problem is use use the prior information
% of smoothness to reconstruct the curve.  AN F-test is used to evaluate whether
% a finely-parameterized model is a better fit than a corsely-parameterized one.

% error associated with dense problem
e1 = sqrtcovdi*(G1*mest1-dobs);
l1 = sqrtcovhi1*(H1*mest1-h1);
E1 = e1'*e1;
L1 = l1'*l1;
P1 = E1+L1;
vP1 = N+K1-M1;
vE1 = vP1*N/(N+K1);
vL1 = vP1*K1/(N+K1);

% error associated with sparse problem
e2 = sqrtcovdi*(G2*mest2-dobs);
l2 = sqrtcovhi2*(H2*mest2-h2);
E2 = e2'*e2;
L2 = l2'*l2;
P2 = E2+L2;
vP2 = N+K2-M2;
vE2 = vP2*N/(N+K2);
vL2 = vP2*K2/(N+K2);

% F ratio
vA=vP1;
vB=vP2;
PhiA = P1;
PhiB = P2;
Fobs = (PhiA/vA) / (PhiB/vB);
```

```
if( Fobs<1 )
    Fobs=1/Fobs;
end
disp(sprintf('1/F %f F %f', 1/Fobs, Fobs));
```

```
1/F 0.954740 F 1.047405
```

```
Pval = 1 - abs(fcdf(1/Fobs,vA,vB)-fcdf(Fobs,vA,vB));
Pleft = fcdf(1/Fobs,vA,vB);
Pright = 1-fcdf(Fobs,vA,vB);
disp(sprintf('P(F<%f) = %f', 1/Fobs, Pleft));
```

```
P(F<0.954740) = 0.344053
```

```
disp(sprintf('P(F>%f) = %f', Fobs, Pright));
```

```
P(F>1.047405) = 0.344053
```

```
disp(sprintf('P(F<%f or F>%f) = %f', 1/Fobs, Fobs, Pval));
```

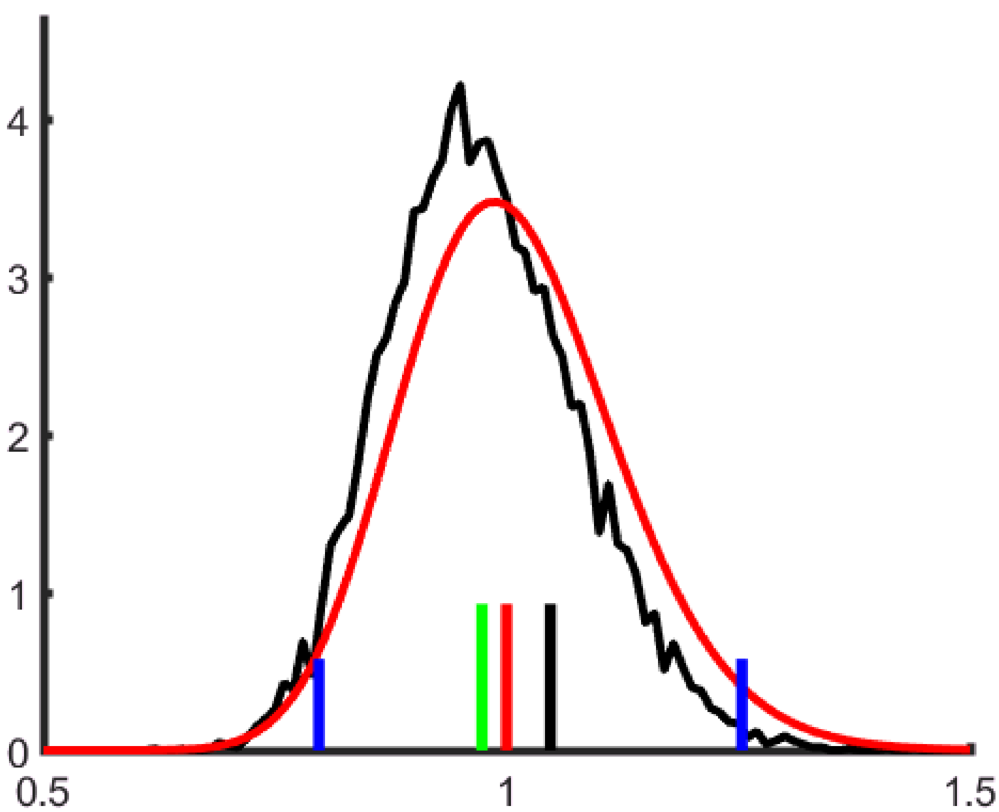
```
P(F<0.954740 or F>1.047405) = 0.688107
```

```
if( (Pleft+Pright)<0.05 )
    fprintf('Null Hypothesis can be rejected to 95%% confidence\n');
else
    fprintf('Null Hypothesis cannot be rejected to 95%% confidence\n');
end
```

```
Null Hypothesis cannot be rejected to 95% confidence
```

```

% now do a whole lot of the same problems
% with different realizations of observational noise
Nreal = 10000;
FFvec = zeros(Nreal,1);
for ireal = [1:Nreal]
    % add new noise to data
 dobs = dtrue + random('Normal',0,sigma_d,N,1);

    % dense problem
 f1 = [ sqrtcovdi*dobs; sqrtcovhi1*h1 ];
 mest1 = FTFinv1*(F1'*f1);
 e1 = sqrtcovdi*(G1*mest1-dobs);
 l1 = sqrtcovhi1*(H1*mest1-h1);
 PP1 = (e1'*e1+l1'*l1);

    % add new noise to data
 dobs = dtrue + random('Normal',0,sigma_d,N,1);
    % sparse problem
 f2 = [ sqrtcovdi*dobs; sqrtcovhi2*h2 ];
 mest2 = FTFinv2*(F2'*f2);
 e2 = sqrtcovdi*(G2*mest2-dobs);
 l2 = sqrtcovhi2*(H2*mest2-h2);
 PP2 = (e2'*e2+l2'*l2);

    % tabulate F ratio
    FFvec(ireal) = (PP1/vP1) / (PP2/vP2);
end
```

```
FFvec = sort(FFvec);
FFmean = mean(FFvec);

% histogram and empirical p.d.f.
Nbins = 101;
binmin = 0.5;
binmax = 1.5;
Dbin = (binmax-binmin)/(Nbins-1);
bins = binmin + Dbin*[0:Nbins]';
FFhist = hist( FFvec, bins );
FFhist = FFhist/(Dbin*sum(FFhist));

% plot pdf's
figure(2);
clf;
set(gca,'LineWidth',3);
set(gca,'FontSize',14);
hold on;
top = 1.1*max(FFhist);
axis( [binmin, binmax, 0, top ] );
plot( bins,FFhist, 'k-', 'LineWidth', 3 );
plot( bins,fpdf(bins,vP1,vP2), 'r-', 'LineWidth', 3 );
plot( [Fobs,Fobs]', [0, top/5]', 'k-', 'LineWidth', 4 );
plot( [FFmean,FFmean]', [0, top/5]', 'g-', 'LineWidth', 4 );
plot( [finv(0.5,vP1,vP2),finv(0.5,vP1,vP2)]', [0, top/5]', 'r-', 'LineWidth', 4 );
plot( [finv(0.025,vP1,vP2),finv(0.025,vP1,vP2)]', [0, top/8]', 'b-', 'LineWidth', 4 );
plot( [finv(0.975,vP1,vP2),finv(0.975,vP1,vP2)]', [0, top/8]', 'b-', 'LineWidth', 4 );
```




```
% Figure 5.19  The F-distribution (red curve) for the second example of Section 5.7, has v_A=3
% and v_B=301 degrees of freedom, and mean (red bar) and 95% confidence interval (blue bars) a
% The value F^est=0.969 (black bar) estimated from the data (treated as one realization of the
```

```
% problem) lies within the interval, indicating that the null hypothesis cannot be rejected.
```