## Contents



## load precomputed data

```
load data1
t=data1(:,1);
y=data1(:,2);% + 30*randn(length(t),1);
sigma=data1(:,3);
```

## Para otros datos.

load dataCO2.txt

## Se asignan variables a las columnas de los datos:

t = dataCO2(:,1); y = dataCO2(:,2); sigma = dataCO2(:,3); N = length(t);

```
%use all data
% N = length(t);
```

## use first 10 data

```
N = 10;
t=t(1:N);
y=y(1:N);
sigma=sigma(1:N);
%
disp('displaying t,y,sigma')
[t , y , sigma]

%add an outlier
%y(4)=y(4)-300;
```

```
displaying t,y,sigma

ans =

          1.00         109.38           8.00
          2.00         187.54           8.00
          3.00         267.53           8.00
          4.00         331.88           8.00
```

```
         5.00          386.05            8.00
         6.00          428.43            8.00
         7.00          452.16            8.00
         8.00          498.15            8.00
         9.00          512.35            8.00
        10.00          512.98            8.00
```

## build the parabolic system matrix

```
G = [ ones(N,1) , t , -1/2*t.*t ];
```

## apply the weighting

```
yw = y./sigma;

Gw = G./[sigma,sigma,sigma];
```

## solve for the least-squares solution

```
disp('Least-squares solution')
m = Gw\yw
```

```
Least-squares solution

m =

        16.42
        96.97
         9.41
```

## get the covariance matrix

```
ginv = inv(Gw'*Gw)*Gw';

disp(['Covariance matrix'])
covm = ginv*ginv'
```

```
Covariance matrix

covm =

        88.53         -33.60           -5.33
       -33.60          15.44            2.67
        -5.33           2.67            0.48
```

## get the 1.96-sigma (95%) conf intervals

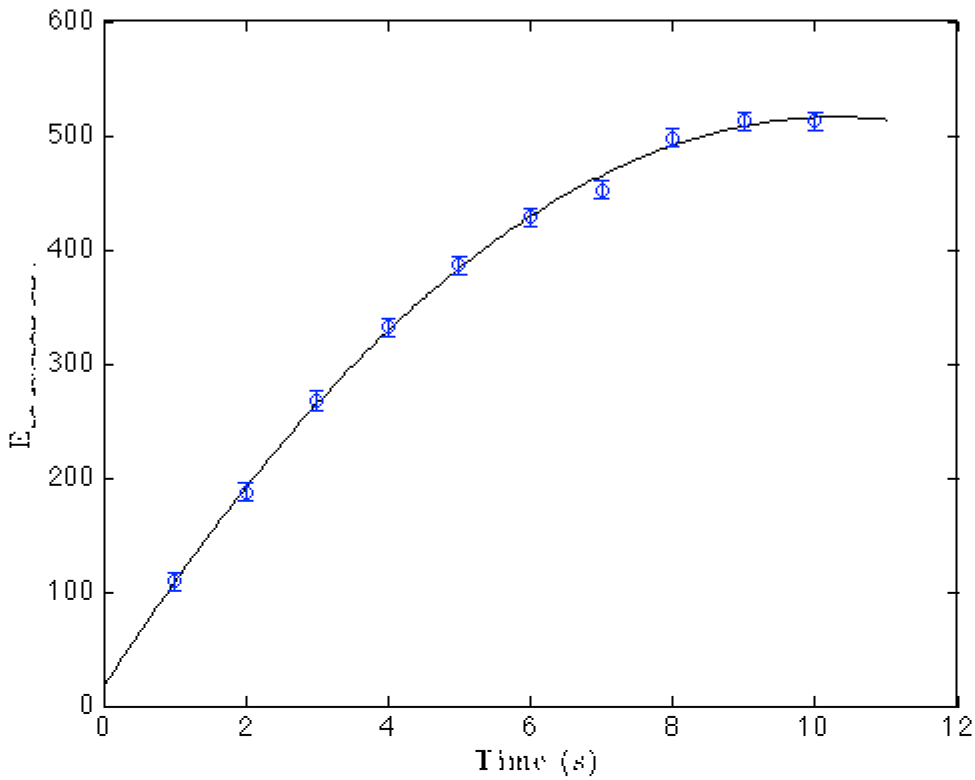
```
disp('95% parameter confidence intervals (m-, mest, m+)')
del = 1.96*sqrt(diag(covm));
[m-del , m , m+del]

dof = N-3;
disp(['Chi-square misfit for ',num2str(dof),' dof'])
chi2 = norm((y - G*m)./sigma)^2


%find the p-value for this data set
%degrees of freedom
disp(['chi-square p-value'])
p = 1-chi2cdf(chi2,dof)


%find the parameter correlations
s=sqrt(diag(covm))
disp(['correlation matrix'])
r = covm./(s*s')
```

```
95% parameter confidence intervals (m-, mest, m+)

ans =

         -2.02         16.42         34.86
         89.27         96.97        104.67
          8.04          9.41         10.77

Chi-square misfit for 7 dof

chi2 =

          4.20

chi-square p-value

p =

          0.76


s =

          9.41
          3.93
          0.70

correlation matrix

r =

          1.00         -0.91         -0.81
         -0.91          1.00          0.97
         -0.81          0.97          1.00
```

## Plot the data and model

```
xx=min(t)-1:0.05:max(t)+1;
mm=m(1)+m(2)*xx-0.5*m(3)*xx.^2;

figure(1)
bookfonts;
plot(xx,mm,'k');
hold on
% plot(t,y,'*')
errorbar(t,y,sigma,'o');
xlabel('Time (s)','interpreter','latex');
ylabel('Elevation (m)','interpreter','latex');
disp(['displaying data and model fit'])
hold off
% print -dpdf parabfig1.pdf
```

```
displaying data and model fit
```




## Monte Carlo Section

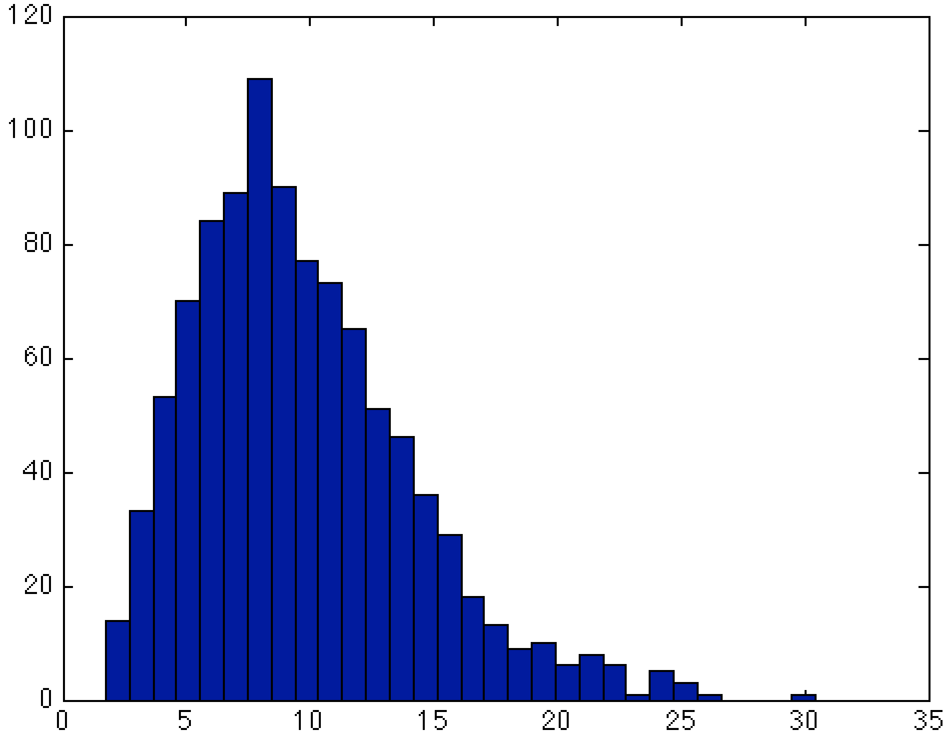
```
y0 = G*m;

for nreal = 1:1000
%
```

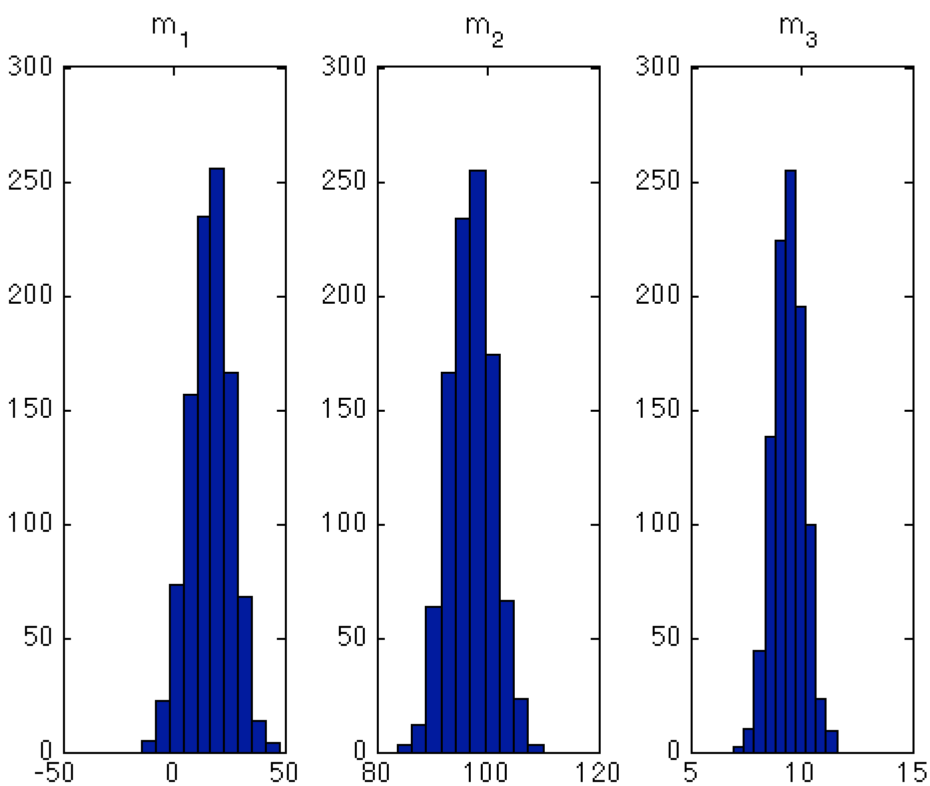
```
%generate a trial data set of perturbed, weighted data
%
  ytrial = y0+sigma.*randn(N,1);
  ywtrial=ytrial./sigma;
  mmc(nreal,:)=(Gw\ywtrial)';
  chimc(nreal)= norm((ytrial-y0)./sigma)^2;
end

figure(2)
bookfonts;
hist(chimc,30);
%title(['1000 Monte-Carlo Chi-square Values'])
disp('Displaying 1000 Monte-Carlo Chi-square Values')
%print -deps parabfig2.eps

figure(3)

subplot(1,3,1)
bookfonts;
hist(mmc(:,1))
title(['m_1'])

subplot(1,3,2)
bookfonts;
hist(mmc(:,2))
title(['m_2'])

subplot(1,3,3)
bookfonts;
hist(mmc(:,3))
title(['m_3'])
%print -deps parabfig3.eps

figure(4)
subplot(1,3,1)
bookfonts;
plot(mmc(:,1),mmc(:,2),'k*')
xlabel('M_1')
ylabel('M_2')
subplot(1,3,2)
bookfonts;
plot(mmc(:,1),mmc(:,3),'k*')
xlabel('M_1')
ylabel('M_3')
subplot(1,3,3)
bookfonts;
plot(mmc(:,2),mmc(:,3),'k*')
xlabel('M_2')
ylabel('M_3')
disp('Displaying 1000 Monte-Carlo models')
%
% Plot the ellipses.
%
figure(5);
clf;
%
```

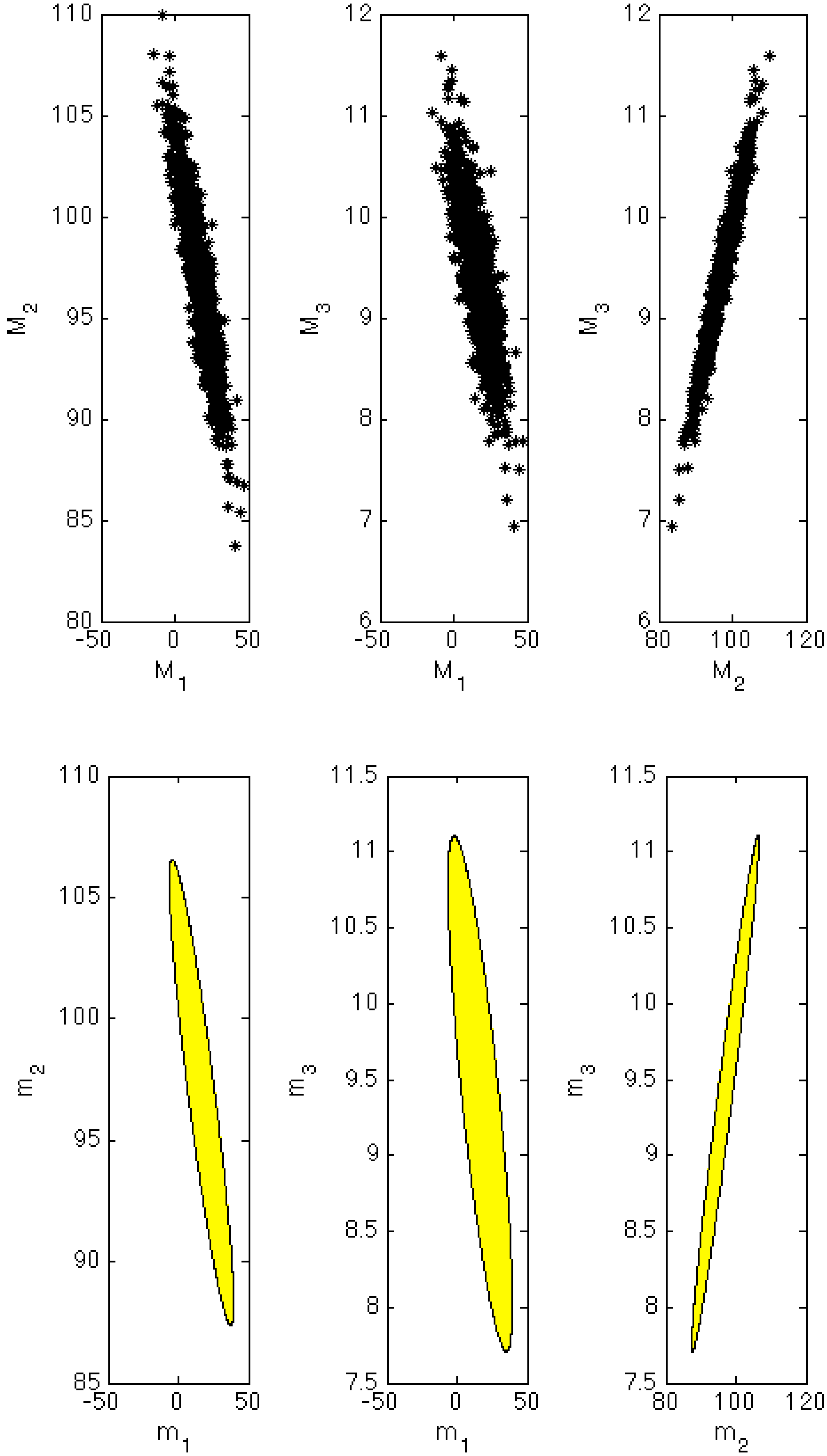
```
% Do the m1, m2 ellipsoid.
%
C=covm([1:2],[1:2]);
[u,lam]=eig(inv(C));
deltachisq=chi2inv(0.95,2);
delta=sqrt(deltachisq);
%generate a vector of angles from 0 to 2*pi
theta=(0:.01:2*pi)';
%calculate the x component of the ellipsoid for all angles
r=zeros(length(theta),2);
r(:,1)=(delta/sqrt(lam(1,1)))*u(1,1)*cos(theta) +...
    (delta/sqrt(lam(2,2)))*u(1,2)*sin(theta);
%calculate the y component of the ellipsoid for all angles
r(:,2)=(delta/sqrt(lam(1,1)))*u(2,1)*cos(theta) +...
    (delta/sqrt(lam(2,2)))*u(2,2)*sin(theta);
%plot(x,y), adding in the model parameters
subplot(1,3,1)
bookfonts;
plot(m(1)+r(:,1),m(2)+r(:,2),'k');
fill(m(1)+r(:,1),m(2)+r(:,2),'y');
%axis([-50 50 85 110]);
xlabel('m_{1}');
ylabel('m_{2}');
m1max=max(r(:,1));
m1min=min(r(:,1));
m2max=max(r(:,2));
m2min=min(r(:,2));
%
% Do the m1, m3 ellipsoid.
%
C=covm([1,3],[1,3]);
[u,lam]=eig(inv(C));
deltachisq=chi2inv(0.95,2);
delta=sqrt(deltachisq);
%calculate the x component of the ellipsoid for all angles
r(:,1)=(delta/sqrt(lam(1,1)))*u(1,1)*cos(theta) +...
    (delta/sqrt(lam(2,2)))*u(1,2)*sin(theta);
%calculate the y component of the ellipsoid for all angles
r(:,2)=(delta/sqrt(lam(1,1)))*u(2,1)*cos(theta) +...
    (delta/sqrt(lam(2,2)))*u(2,2)*sin(theta);
%plot(x,y), adding in the model parameters
subplot(1,3,2)
bookfonts;
plot(m(1)+r(:,1),m(3)+r(:,2),'k');
fill(m(1)+r(:,1),m(3)+r(:,2),'y');
% axis([-50 50 7 12]);
xlabel('m_{1}');
ylabel('m_{3}');
m1max=max([m1max; r(:,1)]);
m1min=min([m1min; r(:,1)]);
m3max=max(r(:,2));
m3min=min(r(:,2));

%
% Do the m1, m3 ellipsoid.
%
```

```
C=covm([2,3],[2,3]);
[u,lam]=eig(inv(C));
deltachisq=chi2inv(0.95,2);
delta=sqrt(deltachisq);
%calculate the x component of the ellipsoid for all angles
r(:,1)=(delta/sqrt(lam(1,1)))*u(1,1)*cos(theta) +...
    (delta/sqrt(lam(2,2)))*u(1,2)*sin(theta);
%calculate the y component of the ellipsoid for all angles
r(:,2)=(delta/sqrt(lam(1,1)))*u(2,1)*cos(theta) +...
    (delta/sqrt(lam(2,2)))*u(2,2)*sin(theta);
%plot(x,y), adding in the model parameters
subplot(1,3,3)
bookfonts;
plot(m(2)+r(:,1),m(3)+r(:,2),'k');
fill(m(2)+r(:,1),m(3)+r(:,2),'y');
% axis([80 120 7 12]);
xlabel('m_{2}');
ylabel('m_{3}');
%print -deps parabfig5.eps

m2max=max([m2max; r(:,1)]);
m2min=min([m2min; r(:,1)]);
m3max=max([m3max; r(:,2)]);
m3min=min([m3min; r(:,2)]);

covm
[u,lam]=eig(inv(covm))
```

```
Displaying 1000 Monte-Carlo Chi-square Values
Displaying 1000 Monte-Carlo models

covm =

         88.53        -33.60         -5.33
        -33.60         15.44          2.67
         -5.33          2.67          0.48


u =

          0.93          0.37         -0.03
         -0.36          0.90         -0.23
         -0.06          0.23          0.97


lam =

          0.01             0             0
             0          0.40             0
             0             0        104.71
```


120
100
80
60
40
20
0
0
5
10
15
20
25
30
35


m1
m2
m3
300
250
200
150
100
50
0
-50
0
50
80
100
120
5
10
15



M2
M1
M3
M1
M3
M2
m2
m1
m3
m1
m3
m2