```
function y = NonlinearLeastMeanSquaresMatlabForStudents()

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% DETERMINATION OF THE CONSTANT OF GRAVITY
%
% This Matlab-program was developed using "Matlab_R2014b" on a MacBook Pro
% running OS X 10.9.5
%
% written by T. Ihn, D-PHYS ETH Zurich, 5 Oct 2014
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% The data
%

% Importing the data
data = csvread('data_z_t.csv');

% Displaying the data as a table
disp(table(data(:,1),data(:,2),'VariableNames',{'z','t'}))

% Converting the data to SI units:
data(:,1)=data(:,1)/100; % convert from cm to m
data(:,2)=data(:,2)/1000;% convert from ms to s
disp(table(data(:,1),data(:,2),'VariableNames',{'z','t'}))

% Plotting the data
figure(1)
loglog(data(:,1),data(:,2),'bo','MarkerFaceColor','b');
axis([0.05 1.3 0.1 0.6])
xlabel('z (m)');
ylabel('t (s)');
title('Measured data')

%
% Analysis using Model I following the lecture
%

% Defining the mean square error function
function y=MSE1(g,data)
    diff = data(:,2)-sqrt(2*data(:,1)/g);
    y = diff'*diff/length(data);
end

% Plotting the mean square error function
figure(2)
g = linspace(9.759,9.779);
Q = zeros(size(g));
for n=1:length(g),
    Q(n) = MSE1(g(n),data);
end;
plot(g,1e6*Q,'b-');
axis([9.759 9.779 2.85e-1 3.3e-1])
xlabel('g (m/s^2)');
ylabel('Q (ms^2)');
title('Mean square error')
```

```
% Numerical minimzation of the square error

[gg,Qmin]=fminsearch(@(g) MSE1(g,data),9.78);

N = length(data);
sigma2_est = N*Qmin/(N-1);

% Standard error of the parameters
s_sigma2 = N*Qmin*sqrt(2/(N-3))/(N-1);

% for obtaining the standard error in the g estimate we define a function
% for the normalized mean square error minus one
function y=MSE1norm(s,gg,Qmin,data)
    y = (length(data)-1)*(MSE1(gg+s,data)/Qmin-1)-1;
end
% the zero of this function is the desired uncertainty
sg = fzero(@(s) MSE1norm(s,gg,Qmin,data),0.01);
fprintf('\nModel I:\n');
fprintf('g = %1.4f +/- %1.4f m/s2\n',gg,sg);
fprintf('sigma2 = %1.3f +/- %1.3f ms2\n',1e6*sigma2_est,1e6*s_sigma2);

% Plot the result of the fit
figure(1)
hold on
loglog(data(:,1),sqrt(2*data(:,1)/gg),'g-');
hold off
title('Measured data and fit')

%
% Inspecting the residuals of the fit
%
res1 = data(:,2)-sqrt(2*data(:,1)/gg);
figure(3)
plot(data(:,1),1e3*res1,'bo','MarkerFaceColor','b');
axis([0 1.4 -2 2])
xlabel('z (m/s^2)');
ylabel('r (ms)');
title('Fit residuals Model I')
grid

%
% Analysis using Model II following the lecture
%

% Defining the mean square error function
function y=MSE2(parms,data)
    diff = data(:,2)-sqrt(2*(data(:,1)-parms(2))/parms(1));
    y = diff'*diff/length(data);
end

% Plotting the mean square error function
g=linspace(9.796,9.804);
z0=linspace(-0.0022,-0.0018);
Y=zeros(length(z0),length(g));
X=Y;
Q=Y;
for n=1:length(g),
```

```
    for m=1:length(z0);
        Y(m,n)=1e3*z0(m);
        X(m,n)=g(n);
        Q(m,n)=MSE2([g(n) z0(m)],data);
    end
end

figure(4)
pcolor(X,Y,Q)
shading interp
hold on
xlabel('g (m/s^2)')
ylabel('z_0 (mm)')
title('Mean square error')
colorbar

% Numerical minimization of the mean square error
[p,Qmin]=fminsearch(@(parms) MSE2(parms,data),[9.78 -0.001]);
gg = p(1);
zz0= p(2);
sigma2_est = N*Qmin/(N-2);
% Draw contour lines and the minimum in the figure
[C,~]=contour(X,Y,Q,[Qmin*(1+1/(N-2)) Qmin*(1+2/(N-2))],'Linewidth',2,'LineColor','w');
plot(gg,1e3*zz0,'wo','MarkerFaceColor','w')

% We determine the standard error of the parameters from the contour lines
CC = C(:,2:(C(2,1)+1)); % This are the coordinates of the relevant contour line
sg = max(CC(1,:))-gg; % the maximum in g-direction
sz0 = max(CC(2,:))-1e3*zz0; % the maximum in z0-direction
% for determining rho, we extract half of the ellipse...
maxi=find(CC(1,:)==max(CC(1,:)));
mini=find(CC(2,:)==max(CC(2,:)));
range=linspace(mini,maxi,abs(maxi-mini)+1);
CCh=CC(:,range);
% ... and determine the intercept with z0=zz0:
intercept=interp1(CCh(2,:),CCh(1,:),1e3*zz0);
% now we can calculate rho:
rho=-sqrt(1-((intercept-gg)/sg)^2);

s_sigma2 = sigma2_est*sqrt(2/(N-4));
hold off

fprintf('\nModel II:\n');
fprintf('g = %1.4f +/- %1.4f m/s2\n',gg,sg);
fprintf('z0 = %1.2f +/- %1.2f mm\n',1e3*zz0,sz0);
fprintf('rho = %1.2f\n',rho);
fprintf('sigma2 = %1.3f +/- %1.3f ms2\n',1e6*sigma2_est,1e6*s_sigma2);

% Plot the result of the fit
figure(1)
hold on
loglog(data(:,1),sqrt(2*(data(:,1)-zz0)/gg),'r--');
hold off
title('Measured data and fits')

% Inspect the residuals of the fit
res2 = data(:,2)-sqrt(2*(data(:,1)-zz0)/gg);
figure(5)
```

```
plot(data(:,1),1e3*res2,'bo','MarkerFaceColor','b');
axis([0 1.4 -2 2])
xlabel('z (m/s^2)');
ylabel('r (ms)');
title('Fit residuals Model II')
grid

%
% Model I: using a shortcut offered by Matlab
%

% first we define the fitting function:
function y=fit1(g,zdata)
    y = sqrt(2*zdata/g);
end

% then we use lsqcurvefit to do the fitting
[gg, resnorm, res, ~, ~, ~, J] = lsqcurvefit(@fit1,9.81,data(:,1),data(:,2));
N=length(data);
sigma2_est=resnorm/(N-1);

% standard errors of the parameters
[~, R] = qr(J,0);
mse = sum(abs(res).^2)/(size(J,1)-size(J,2));
Rinv = inv(R);
Sigma = Rinv*Rinv'*mse;
se = sqrt(diag(Sigma));
sg=full(se);

s_sigma2 = sigma2_est*sqrt(2/(N-3));

fprintf('\nModel I, Matlab least mean square fit:\n');
fprintf('gg = %1.4f +/- %1.4f m/s2\n',gg,sg);
fprintf('sigma2 = %1.3f +/- %1.3f ms2\n',1e6*sigma2_est,1e6*s_sigma2);

%Inspect the residuals of the fit:
figure(6);
plot(data(:,1),-1e3*res,'bo','MarkerFaceColor','b')
axis([0 1.4 -2 2])
xlabel('z (m/s^2)');
ylabel('r (ms)');
title('Fit residuals Model I, Matlab lsqcurvefit')
grid

%
% Model II: using the shortcut offered by Matlab
%

% first we define the fitting function:
function y=fit2(parms,zdata)
    y = sqrt(2*(zdata-parms(2))/parms(1));
end

% then we use lsqcurvefit to do the fitting
[parms, resnorm, res, ~, ~, ~, J] = lsqcurvefit(@fit2,[9.81 0.001],data(:,1),data(:,2));
gg = parms(1);
zz0 = parms(2);
N=length(data);
```

```
sigma2_est=resnorm/(N-1);

% standard errors of the parameters
[~, R] = qr(J,0);
mse = sum(abs(res).^2)/(size(J,1)-size(J,2));
Rinv = inv(R);
Sigma = Rinv*Rinv'*mse; % here Sigma is a 2x2 matrix
sg = full(sqrt(Sigma(1,1)));
sz0 = full(sqrt(Sigma(2,2)));
rho = full(Sigma(1,2)/sg/sz0);
s_sigma2 = sigma2_est*sqrt(2/(N-4));

% Output the result
fprintf('\nModel II, Matlab least mean square fit:\n');
fprintf('g = %1.4f +/- %1.4f m/s2\n',gg,sg);
fprintf('z0 = %1.2f +/- %1.2f mm\n',1e3*zz0,1e3*sz0);
fprintf('rho = %1.2f\n',rho);
fprintf('sigma2 = %1.3f +/- %1.3f ms2\n',1e6*sigma2_est,1e6*s_sigma2);

%Inspect the residuals of the fit:
figure(7);
plot(data(:,1),-1e3*res,'bo','MarkerFaceColor','b')
axis([0 1.4 -2 2])
xlabel('z (m/s^2)');
ylabel('r (ms)');
title('Fit residuals Model II, Matlab lsqcurvefit')
grid

y = [];
end
```