Inversion Homework #3

Professor H. Yao & H. Zhang

Jintao Li

SA20007037

E-mail: lijintao@mail.ustc.edu.cn

November 13, 2020

Chapter 2

Exercise 1

A seismic profiling experiment is performed where the first arrival times of seismic energy from a mid-crustal refractor are observed at distances (in kilometers) of

$$\mathbf{x} = \begin{bmatrix} 6.0000 \\ 10.1333 \\ 14.2667 \\ 18.4000 \\ 22.5333 \\ 26.6667 \end{bmatrix} \tag{1}$$

from the source, and are found to be (in seconds after the source origin time)

$$\mathbf{t} = \begin{bmatrix} 3.4935 \\ 4.2853 \\ 5.1374 \\ 5.8181 \\ 6.8632 \\ 8.1841 \end{bmatrix} . \tag{2}$$

These vectors can also be found in the MATLAB data file **profile.mat**. A two-layer, flat Earth structure gives the mathematical model

$$t_i = t_0 + s_2 x_i, \tag{3}$$

where the intercept time, t_0, depends on the thickness and slowness of the upper layer, and s_2 is the slowness of the lower layer. The estimated noise in the first arrival time measurements is believed to be independent and normally distributed with expected value 0 and standard deviation $\sigma = 0.1s$.

matlab code: prepare profile.mat

```
x = [6.0000 10.1333 14.2667 18.4000 22.5333 26.6667]';
t = [3.4935 4.2853 5.1374 5.8181 6.8632 8.1841]';
save("profile.mat", "x", "t")
```

(a) Find the least squares solution for the model parameters t_0 and s_2. Plot the data, the fitted model, and the residuals.

Solution:

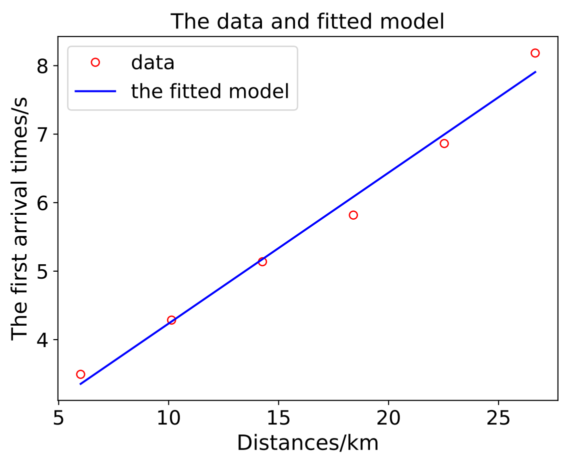
$$\mathbf{m}_{L_2} = \begin{bmatrix} t_0 \\ s_2 \end{bmatrix} = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{t} = \begin{bmatrix} 2.0323 \\ 0.2203 \end{bmatrix}, \tag{4}$$

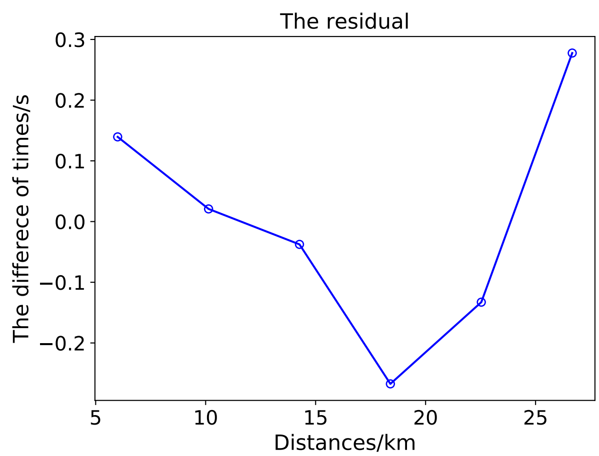
where

$$\mathbf{G} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ 1 & x_4 \\ 1 & x_5 \\ 1 & x_6 \end{bmatrix} . \tag{5}$$

matlab code

```
clear;clc;close all;
load("profile.mat")

G = [ones(6, 1), x];
m_L2 = inv(G' * G) * G' * t;
r = t - G * m_L2;

% plot the data, the fitted model
figure(1)
plot(x, t, "ro");
hold on;
t_m = G * m_L2;
plot(x, t_m, "-b");
xlabel("Distances/km");
ylabel("The first arrival times/s")
legend(["data", "the fitted model"], "Location", "southeast")

% plot the residual
figure(2)
plot(x, r, "-o");
xlabel("Distances/km");
ylabel("The differece of times/s")
title("The residual")
```





(b) Calculate and comment on the model parameter correlation matrix (e.g., 2.43). How are the correlations manifested in the general appearance of the error ellipsoid in (t_0, s_0) space?

Solution:

$$\mathbf{Cov}(\mathbf{m}_{L_2}) = \sigma^2(\mathbf{G}^T\mathbf{G})^{-1} \tag{6}$$

$$\rho_{m_i,m_j} = \frac{\mathrm{Cov}\,(m_i, m_j)}{\sqrt{\mathrm{Var}\,(m_i)\cdot \mathrm{Var}\,(m_j)}} \tag{7}$$

So,

$$\rho = \begin{bmatrix} 1 & -0.9179 \\ -0.9179 & 1 \end{bmatrix} \tag{8}$$

matlab code

```
%% 1-b
Sigma = 0.1;
C = Sigma^2 .* inv(G' * G);
rho = C(1,2) ./ sqrt(C(1,1) .* C(2,2));
correlation_matrix = [1 rho; rho 1];
```

(c) Plot the error ellipsoid in the (t_0, s_2) plane and calculate conservative 95%confidence intervals for t_0 and s_2 for the appropriate value of Δ^2. Hint: The following MATLAB function will plot a two-dimensional covariance ellipse about the model parameters, where C is the covariance matrix, DELTA2 is Δ^2, and m is the 2-vector of model parameters.

matlab code

```
%set the number of points on the ellipse to generate and plot
function plot_ellipse(DELTA2,C,m)
n=100;
%construct a vector of n equally-spaced angles from (0,2*pi)
theta=linspace(0,2*pi,n)';
%corresponding unit vector
xhat=[cos(theta),sin(theta)];
Cinv=inv(C);
%preallocate output array
r=zeros(n,2);
for i=1:n
    %store each (x,y) pair on the confidence ellipse
    %in the corresponding row of r
    r(i,:)=sqrt(DELTA2/(xhat(i,:)*Cinv*xhat(i,:)'))*xhat(i,:);
end
plot(m(1)+r(:,1), m(2)+r(:,2));
axis equal
```

Solution:

The degrees of freedom is 2, so $\Delta^2 = 5.99$. And the eigenvalues of $\mathbf{C}^{-1}$ is

$$[\lambda_1, \lambda_2] \approx [90, 190470]. \tag{9}$$

So,

$$\sqrt{F^{-1}_{\chi^2,3}(0.95)}\left[1/\sqrt{\lambda_1}, 1/\sqrt{\lambda_2}\right] \approx [238, 1068.1] \tag{10}$$

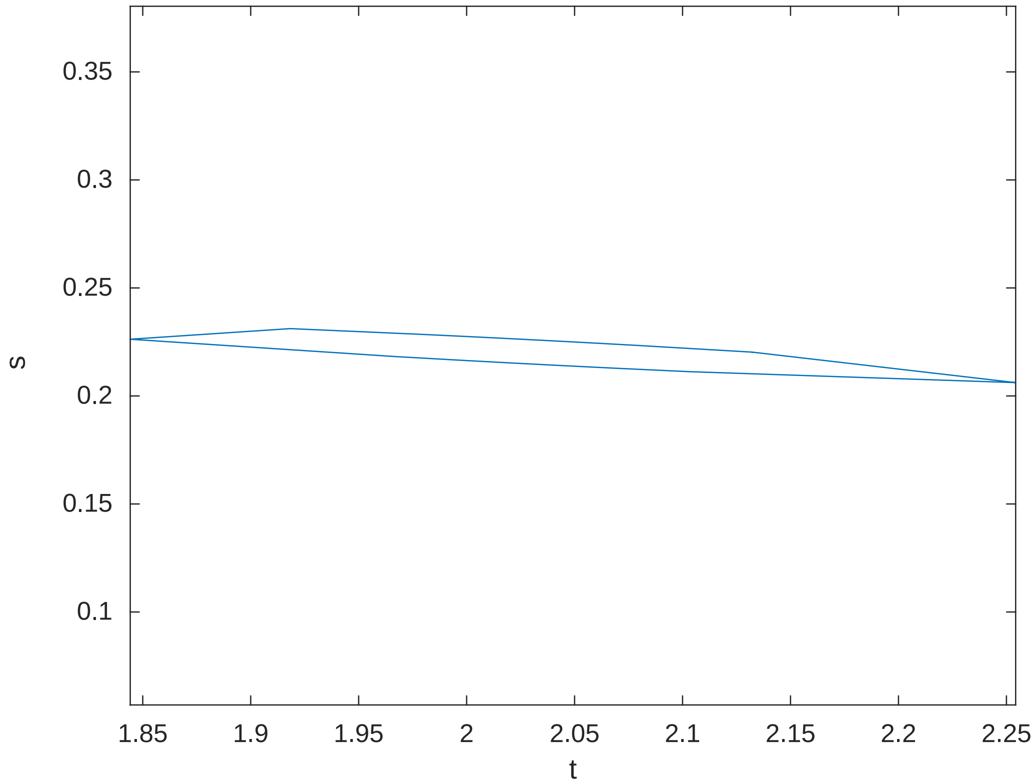
i.e.

$$[t_0, s_2] = [2.0323 \pm 238, 0.2203 \pm 1068.1] \tag{11}$$

The error ellipsoid in the (t_0, s_2) plane:

matlab code

```
%% 1-c
DeltaS = 5.99;
plot_ellipse(DeltaS, C, m_L2)
xlabel("t_0 / s")
ylabel("s_2 / km")
lambda = eig(inv(C));
intv = sqrt(DeltaS) .* lambda .^ 0.5;
```



(d) Evaluate the p-value for this model. You may find the library function **chi2cdf** to be useful here.

Solution:

The χ^2 value for this regression is :

$$\chi^2_{obs} = \sum_{i=1}^{m} (t_i - (\mathbf{Gm}_{L_2})_i)^2/\sigma_i^2. \tag{12}$$

And using matlab **chi2cdf** to calculate the p-value, which is 8.7992×10^{-4}
matlab code

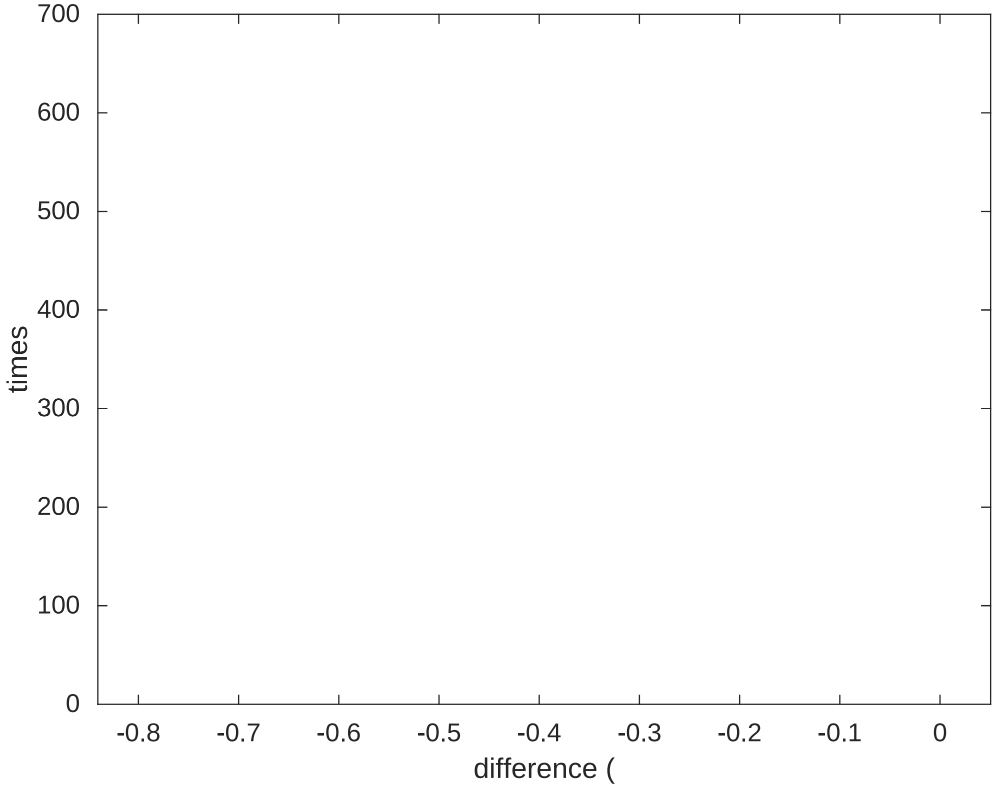
```
%% 1-d
chiValue = sum(r.^2 ./ Sigma^2);
p = chi2cdf(chiValue, 4, "upper");
```

(e) Evaluate the value of χ^2 for 1000 Monte Carlo simulations using the data prediction from your model perturbed by noise that is consistent with the data assumptions. Compare a histogram of these χ^2 values with the theoretical χ^2 distribution for the correct number of degrees of freedom. You may find the library function **chi2pdf** to be useful here.

Solution:

matlab code

```
%% 1-e
chi_sim = zeros(1, 1000);
p_sim = zeros(1, 1000);
for i = 1:1:1000
    noise = 0.1*randn(6, 1);
    t_noise = t + noise;
    m_noise = inv(G' * G) * G' * t_noise;
    r = t_noise - G * m_noise;
    chiv = sum(r .^2 ./ Sigma^2);
    chi_sim(i) = chi2cdf(chiv, 4);
    p_sim(i) = chi2cdf(chiv, 4,"upper");
end

histogram(chi_sim-chi);
```

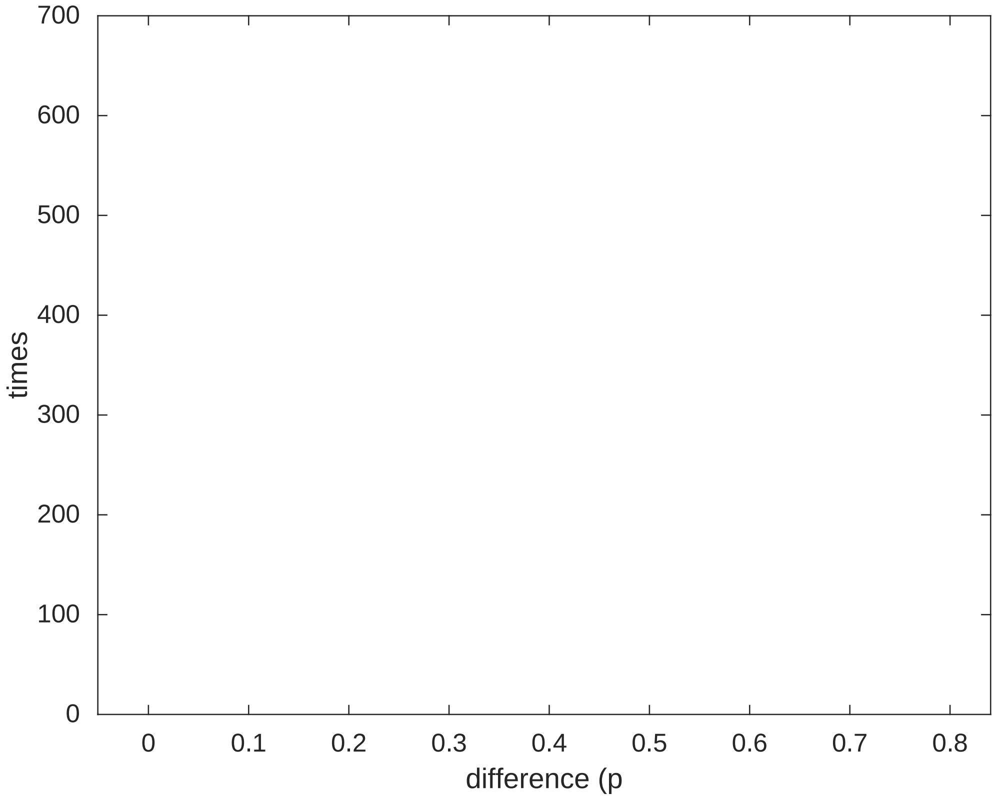


(f) Are your p-value and Monte Carlo χ^2 distribution consistent with the theoretical modeling and the data set? If not, explain what is wrong.

Solution:

matlab code

```
%% 1-f
histogram(p_sim-p);
```



(g) Use IRLS to find 1-norm estimates for t_0 and s_2. Plot the data predictions from your model relative to the true data and compare with (a).

Solution:

We set the iteration tolerance $\tau = 0.0001$ and $\epsilon = 0.0001$.
matlab code: create IRLS.m, which performs IRLS.

```
function m = IRLS(t, G)

condition = true;
threshold = 0.0001;
epsilon = 0.0001;

m0 = inv(G' * G) * G' * t;

r = abs(t - G * m0);
r(r < epsilon) = epsilon;
R = diag(r.^-1);

i = 0;
while condition
    i = i+1;
    m1 = (G' * R * G) \ (G' * R * t);
```

```
    condition = (norm(m1 - m0) ./ (1 + norm(m1))) > threshold;
    m0 = m1;
    r = abs(t - G * m0);
    r(r < epsilon) = epsilon;
    R = diag(r.^-1);
end

m = m0;
```

matlab code: find 1-norm estimates

```
load("profile.mat")
G = [ones(6, 1), x];
m_L2 = inv(G' * G) * G' * t;

% plot 2-norm solution
figure(1)
plot(x, t, "ro");
hold on;
t_m = G * m_L2;
plot(x, t_m, "-b");
hold on;

% calculate 1-norm solution and plot it
m_L1 = IRLS(t, G);
plot(x, G * m1, "--g");

xlabel("Distances/km");
ylabel("The first arrival times/s")
legend(["data", "L2 solution", "IRLS"], "Location", "southeast")
```

The comparision:

(h) Use Monte Carlo error propagation and IRLS to estimate symmetric 95% confidence intervals on the 1-norm solution for t_0 and s_2.

Solution:

The 95% confidence intervals are given by

$$\mathbf{m}_{L_1} \pm 1.96 \, \mathrm{diag} \left(\mathrm{Cov}\left(\mathbf{m}_{L_1}\right)\right)^{1/2}. \tag{13}$$

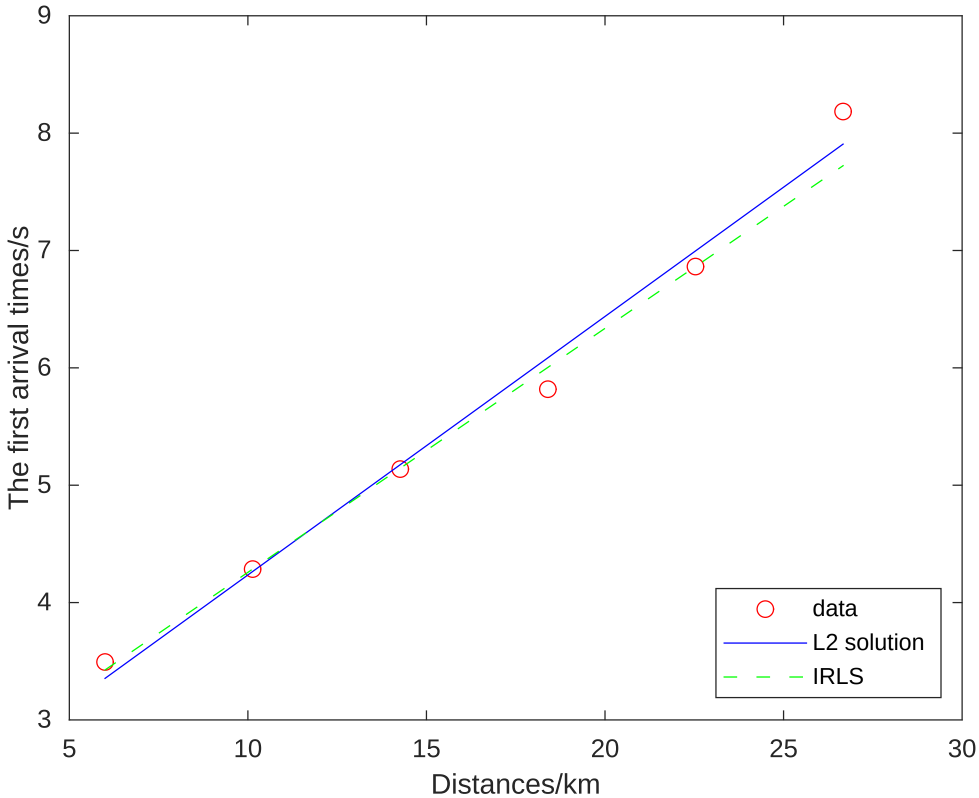
And

$$\mathrm{Cov}(\mathbf{m}_{L_1}) = \frac{\mathbf{A}^T\mathbf{A}}{q}. \tag{14}$$

Calculate it by MATLAB, $[t_0, s_2] = [2.1786 \pm 0.2996, 0.2079 \pm 0.0201]$

matlab code

```
%% 1-h
q = 1000;
m_all = zeros(q, 2);
A = zeros(q, 2);

for i = 1:1:q
    noise = Sigma .* randn(6, 1);
```




```
    t_noise = t + noise;
    m_all(i, :) = (IRLS(t_noise, G))';
end

A = m_all - mean(m_all);

CovML1 = A' * A ./ q;

conf = 1.96 .* diag(CovML1) .^ 0.5;
```

(i) Examining the contributions from each of the data points to the 1-norm misfit measure, can you make a case that any of the data points are statistical outliers?

Solution:

Exercise 2

In this chapter we have largely assumed that the data errors are independent. Suppose instead that the data errors have an MVN distribution with expected value $\mathbf{0}$ and a covariance matrix $\mathbf{C}_D$. It can be shown that the likelihood function is then

$$L(\mathbf{m} \mid \mathbf{d}) = \frac{1}{(2\pi)^{m/2}} \frac{1}{\sqrt{\det\left(\mathbf{C}_D\right)}} e^{-(\mathbf{Gm}-\mathbf{d})^T \mathbf{C}_D^{-1} (\mathbf{Gm}-\mathbf{d})/2}. \tag{15}$$

(a) Show that the maximum likelihood estimate can be obtained by solving the minimization problem,

$$\min(\mathbf{Gm}-\mathbf{d})^T\mathbf{C}_D^{-1}(\mathbf{Gm}-\mathbf{d}). \tag{16}$$

Solution:

$$\begin{aligned} \max\ L(\mathbf{m}|\mathbf{d}) &= \max\ e^{-(\mathbf{Gm}-\mathbf{d})^T\mathbf{C}_D^{-1}(\mathbf{Gm}-\mathbf{d})/2} \\ &= \max\ -(\mathbf{Gm}-\mathbf{d})^T\mathbf{C}_D^{-1}(\mathbf{Gm}-\mathbf{d})/2 \\ &= \min\ (\mathbf{Gm}-\mathbf{d})^T\mathbf{C}_D^{-1}(\mathbf{Gm}-\mathbf{d}). \end{aligned} \tag{17}$$

(b) Show that (2.111) can be solved using the system of equations

$$\mathbf{G}^T\mathbf{C}_D^{-1}\mathbf{Gm} = \mathbf{G}^T\mathbf{C}_D^{-1}\mathbf{d}. \tag{18}$$

Solution:

To find a solution $\mathbf{m}$ satisfing (2.111), we can leverage its derivative is equal to 0, i.e.

$$\begin{aligned} &F = (\mathbf{Gm}-\mathbf{d})^T\mathbf{C}_D^{-1}(\mathbf{Gm}-\mathbf{d}) \\ &\frac{\partial F}{\partial \mathbf{m}} = 0 \\ \Rightarrow &\mathbf{G}^T\mathbf{C}_D^{-1}(\mathbf{Gm}-\mathbf{d}) + (\mathbf{Gm}-\mathbf{d})^T\mathbf{C}_D^{-1}\mathbf{G} = 0 \\ \Rightarrow &\mathbf{G}^T\mathbf{C}_D^{-1}(\mathbf{Gm}-\mathbf{d}) = -(\mathbf{G}^T\mathbf{C}_D^{-1}(\mathbf{Gm}-\mathbf{d}))^T \\ \Rightarrow &\mathbf{G}^T\mathbf{C}_D^{-1}(\mathbf{Gm}-\mathbf{d}) = 0 \\ i.e.& \\ &\mathbf{G}^T\mathbf{C}_D^{-1}\mathbf{Gm} = \mathbf{G}^T\mathbf{C}_D^{-1}\mathbf{d} \end{aligned} \tag{19}$$

Note that : $\mathbf{C}^T = \mathbf{C}$, so $(\mathbf{C}^{-1})^T = \mathbf{C}^{-1}$

(c) Show that (2.111) is equivalent to the linear least squares problem

$$\min\left\|\mathbf{C}_D^{-1/2}\mathbf{Gm} - \mathbf{C}_D^{-1/2}\mathbf{d}\right\|_2, \tag{20}$$

where $\mathbf{C}_D^{-1/2}$ is the matrix square root of $\mathbf{C}_D^{-1}$.

Solution:

From the normal equations (A.73), a linear least squares problem can convert to solve:

$$\mathbf{G}^T\mathbf{Gm} = \mathbf{G}^T\mathbf{d}. \tag{21}$$

So, the linear least squares problem

$$\min\left\|\mathbf{C}_D^{-1/2}\mathbf{Gm} - \mathbf{C}_D^{-1/2}\mathbf{d}\right\|_2, \tag{22}$$

can convert to solve :

$$
\begin{aligned}
&(\mathbf{C}_D^{-\frac{1}{2}}\mathbf{G})^T(\mathbf{C}_D^{-\frac{1}{2}}\mathbf{G})\mathbf{m} = (\mathbf{C}_D^{-\frac{1}{2}}\mathbf{G})^T\mathbf{C}_D^{-\frac{1}{2}}\mathbf{d} \\
\Rightarrow &\mathbf{G}^T\mathbf{C}_D^{-\frac{1}{2}}\mathbf{C}_D^{-\frac{1}{2}}\mathbf{G}\mathbf{m} = \mathbf{G}^T\mathbf{C}_D^{-\frac{1}{2}}\mathbf{C}_D^{-\frac{1}{2}}\mathbf{d} \\
\Rightarrow &\mathbf{G}^T\mathbf{C}_D^{-1}\mathbf{G}\mathbf{m} = \mathbf{G}^T\mathbf{C}_D^{-1}\mathbf{d}
\end{aligned}
\tag{23}
$$

The two problem can convert to solve a same equation, so the two problem are equivalent.

(d) The Cholesky factorization of $\mathbf{C}_D^{-1}$ can also be used instead of the matrix square root. Show that (2.111) is equivalent to the linear least squares problem

$$
\min \|\mathbf{RGm} - \mathbf{Rd}\|_2 \tag{24}
$$

where $\mathbf{R}$ is the Cholesky factor of $\mathbf{C}_D^{-1}$.

Solution:

Exercise 5

Use linear regression to fit a polynomial of the form

$$
y_i = a_0 + a_1 x_i + a_2 x_i^2 + \ldots + a_{19} x_i^{19} \tag{25}
$$

to the noise-free data points

$$
(x_i, y_i) = (-0.95, -0.95), (-0.85, -0.85), \ldots, (0.95, 0.95) \tag{26}
$$

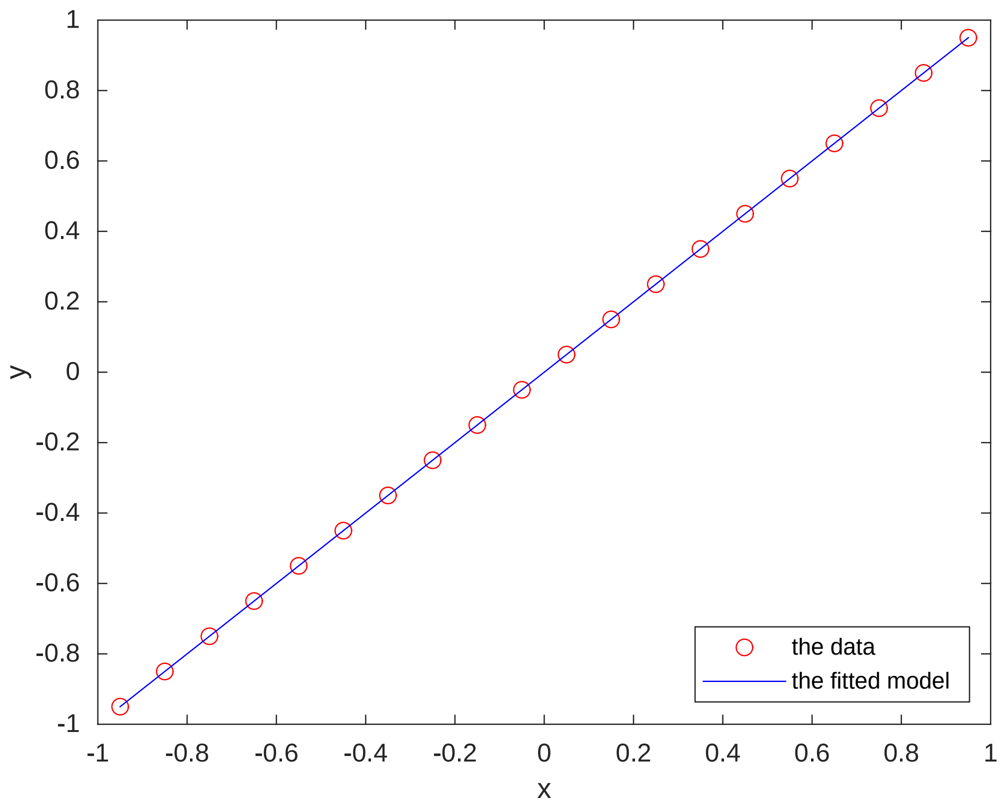
Use the normal equations to solve the least squares problem.
Plot the data and your fitted model, and list the parameters, a_i, obtained in your regression. Clearly, the correct solution has $a_1 = 1$, and all other $a_i = 0$. Explain why your answer differs.

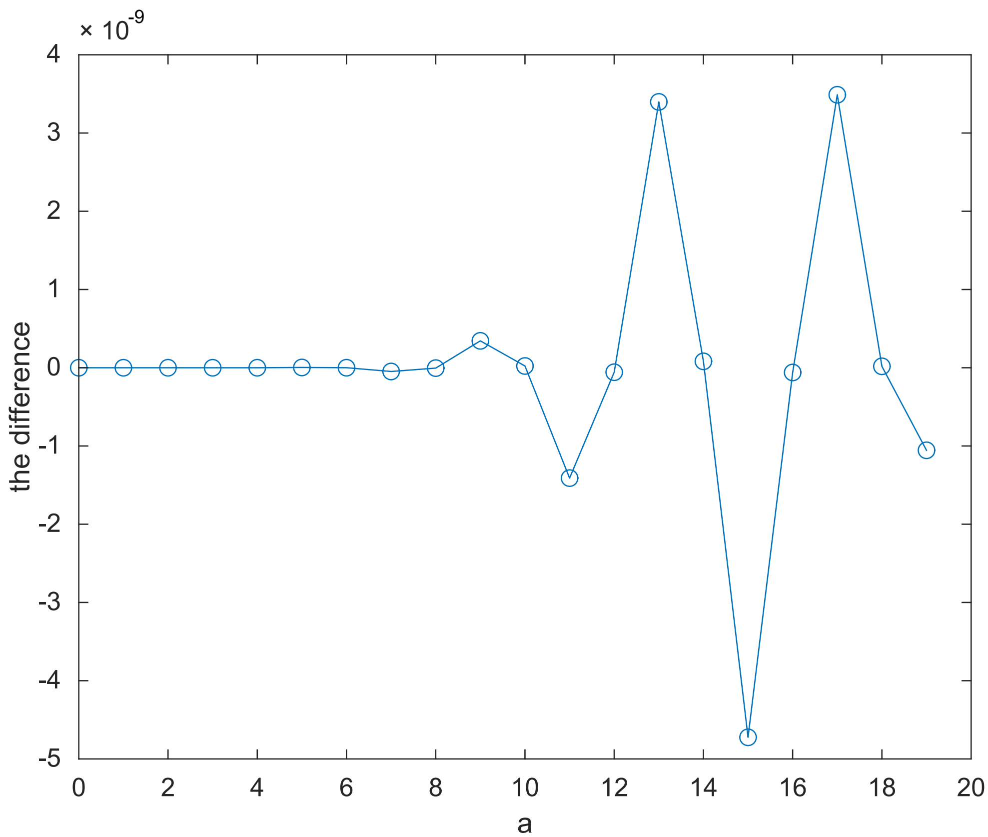
Solution:

matlab code

```
clear;clc;close all;

x = (-0.95:0.1:0.95)';
y = x;

G = zeros(20,20);
for i=1:20
    G(:, i) = x.^(i-1);
end

m = G \ y;
m_true = zeros(20, 1);
m_true(2) = 1;
y_pre = G*m;

figure(1)
plot(x, y, "ro")
hold on;
plot(x, y_pre, "-b")
```

```
xlabel("x")
ylabel("y")
legend(["the data", "the fitted model"], "Location", "southeast")

figure(2)
r = m - m_true;
a = 0:1:19;
plot(a, r, "-o")
xlabel("a")
ylabel("the difference")
```




× 10^-9
4
3
2
1
0
-1
-2
-3
-4
-5
the difference
0
2
4
6
8
10
12
14
16
18
20
a