



# Inversion Homework #3

Professor H. Yao & H. Zhang

Jintao Li

SA20007037

E-mail: [lijintao@mail.ustc.edu.cn](mailto: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

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

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

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

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

$$t_i = t_0 + s_2 x_i, \quad (3)$$

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**

```
1 x = [6.0000 10.1333 14.2667 18.4000 22.5333 26.6667]';
2 t = [3.4935 4.2853 5.1374 5.8181 6.8632 8.1841]';
3 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:**

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

where

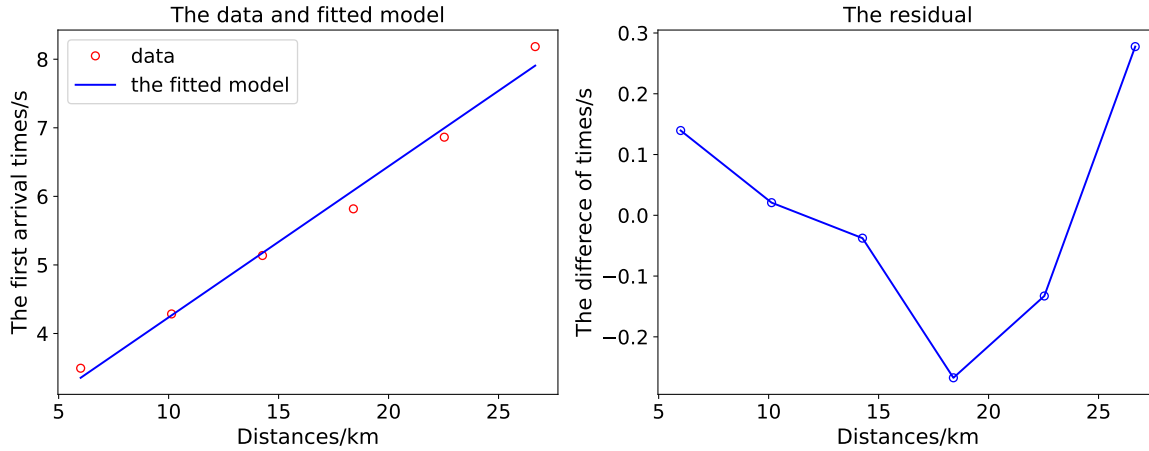
$$\mathbf{G} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ 1 & x_4 \\ 1 & x_5 \\ 1 & x_6 \end{bmatrix}. \quad (5)$$

**matlab code**

```

1 clear;clc;close all;
2 load("profile.mat")
3
4 G = [ones(6, 1), x];
5 m_L2 = inv(G' * G) * G' * t;
6 r = t - G * m_L2;
7
8 % plot the data, the fitted model
9 figure(1)
10 plot(x, t, "ro");
11 hold on;
12 t_m = G * m_L2;
13 plot(x, t_m, "-b");
14 xlabel("Distances/km");
15 ylabel("The first arrival times/s")
16 legend(["data", "the fitted model"], "Location", "southeast")
17
18 % plot the residual
19 figure(2)
20 plot(x, r, "-o");
21 xlabel("Distances/km");
22 ylabel("The difference of times/s")
23 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:**

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

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

So,

$$\rho = \begin{bmatrix} 1 & -0.9179 \\ -0.9179 & 1 \end{bmatrix} \quad (8)$$

**matlab code**

```

1 %% 1-b
2 Sigma = 0.1;
3 C = Sigma^2 .* inv(G' * G);
4 rho = C(1,2) ./ sqrt(C(1,1) .* C(2,2));
5 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  $\Delta^2$ . Hint: The following MATLAB function will plot a two-dimensional covariance ellipse about the model parameters, where  $C$  is the covariance matrix, DELTA2 is  $\Delta^2$ , and  $m$  is the 2-vector of model parameters.

**matlab code**

```

1 %set the number of points on the ellipse to generate and plot
2 function plot_ellipse(DELTA2,C,m)
3 n=100;
4 %construct a vector of n equally-spaced angles from (0,2*pi)
5 theta=linspace(0,2*pi,n)';
6 %corresponding unit vector
7 xhat=[cos(theta),sin(theta)];
8 Cinv=inv(C);
9 %preallocate output array
10 r=zeros(n,2);
11 for i=1:n
12     %store each (x,y) pair on the confidence ellipse
13     %in the corresponding row of r
14     r(i,:)=sqrt(DELTA2/(xhat(i,:)*Cinv*xhat(i,:)'))*xhat(i,:);
15 end
16 plot(m(1)+r(:,1), m(2)+r(:,2));
17 axis equal

```

**Solution:**

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

$$[\lambda_1, \lambda_2] \approx [90, 190470]. \quad (9)$$

So,

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

i.e.

$$[t_0, s_2] = [2.0323 \pm 238, 0.2203 \pm 1068.1] \quad (11)$$

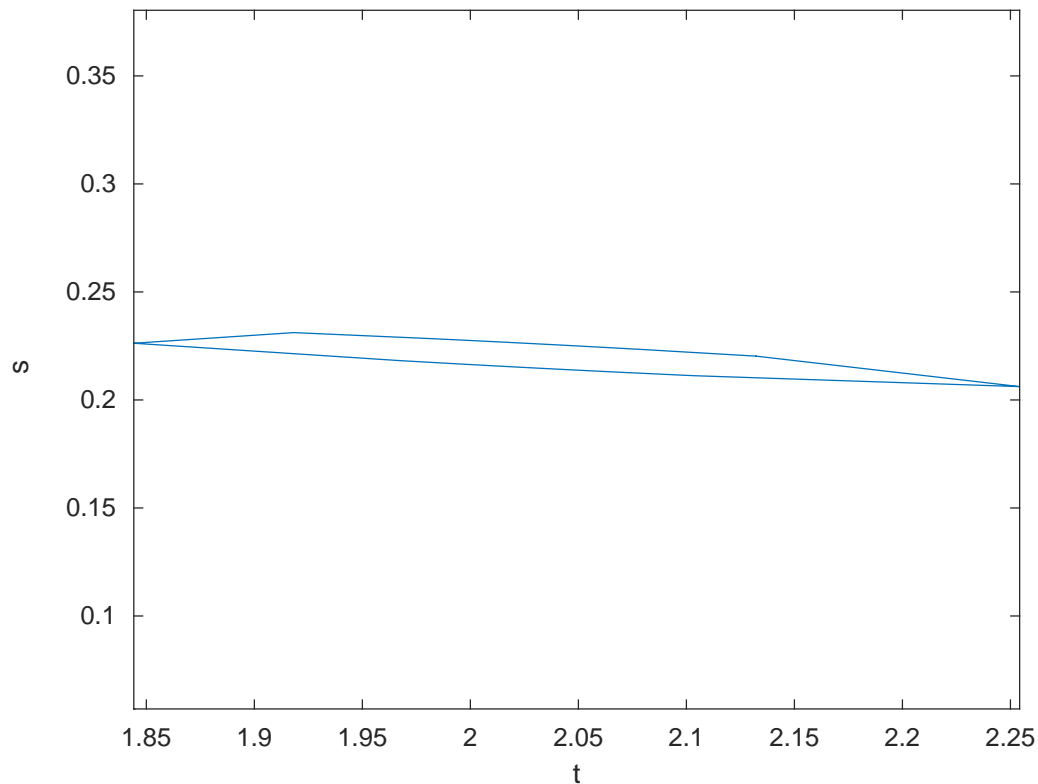
The error ellipsoid in the  $(t_0, s_2)$  plane:

**matlab code**

```

1 %% 1-c
2 DeltaS = 5.99;
3 plot_ellipse(DeltaS, C, m_L2)
4 xlabel("t_0 / s")
5 ylabel("s_2 / km")
6 lambda = eig(inv(C));
7 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  $\chi^2$  value for this regression is :

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

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

**matlab code**

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

(e) Evaluate the value of  $\chi^2$  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  $\chi^2$  values with the theoretical  $\chi^2$  distribution for the correct number of degrees of freedom. You may find the library function **chi2pdf** to be useful here.

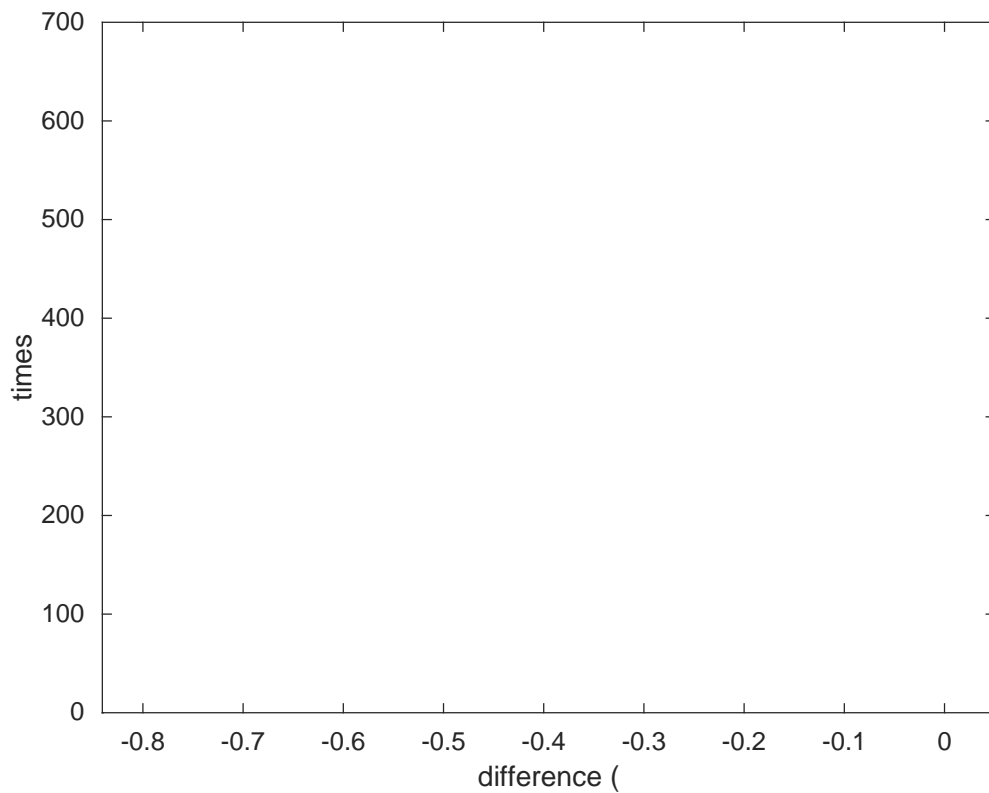
**Solution:**

## matlab code

```

1 %% 1-e
2 chi_sim = zeros(1, 1000);
3 p_sim = zeros(1, 1000);
4 for i = 1:1:1000
5     noise = 0.1*randn(6, 1);
6     t_noise = t + noise;
7     m_noise = inv(G' * G) * G' * t_noise;
8     r = t_noise - G * m_noise;
9     chiv = sum(r.^2 ./ Sigma^2);
10    chi_sim(i) = chi2cdf(chiv, 4);
11    p_sim(i) = chi2cdf(chiv, 4, "upper");
12 end
13
14 histogram(chi_sim-chi);

```



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

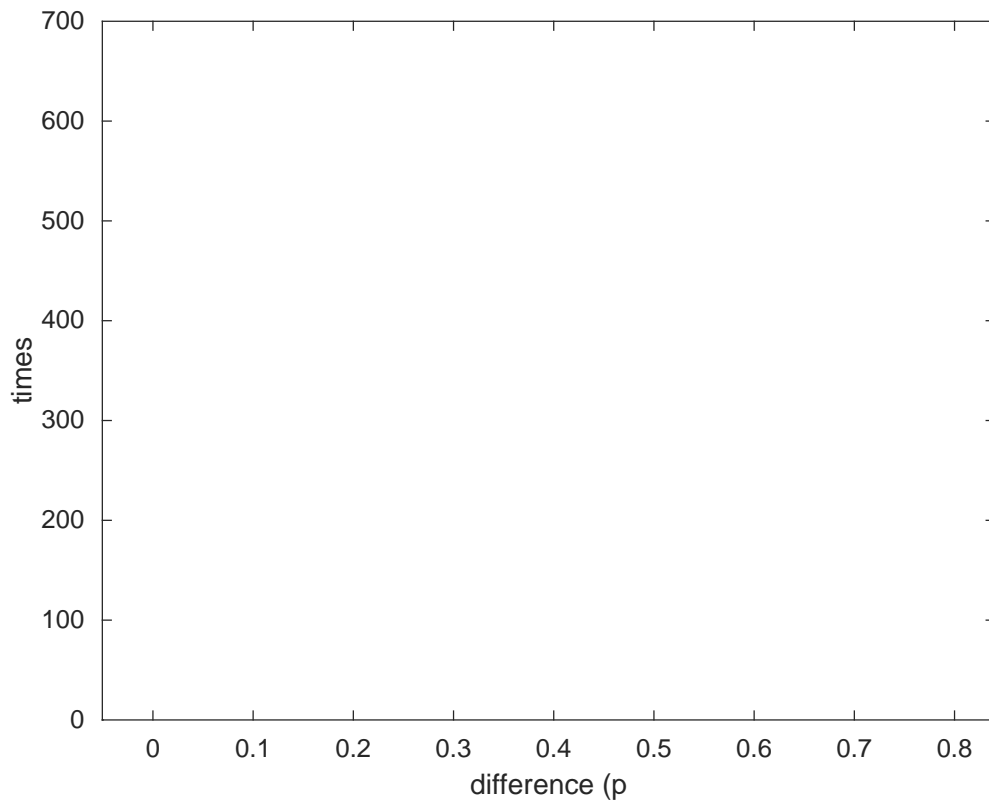
**Solution:**

## matlab code

```

1 %% 1-f
2 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.**

```

1 function m = IRLS(t, G)
2
3 condition = true;
4 threshold = 0.0001;
5 epsilon = 0.0001;
6
7 m0 = inv(G' * G) * G' * t;
8
9 r = abs(t - G * m0);
10 r(r < epsilon) = epsilon;
11 R = diag(r.^-1);
12
13 i = 0;
14 while condition
15     i = i+1;
16     m1 = (G' * R * G) \ (G' * R * t);

```

```

17     condition = (norm(m1 - m0) ./ (1 + norm(m1))) > threshold;
18     m0 = m1;
19     r = abs(t - G * m0);
20     r(r < epsilon) = epsilon;
21     R = diag(r.^-1);
22 end
23
24 m = m0;

```

matlab code: find 1-norm estimates

```

1 load("profile.mat")
2 G = [ones(6, 1), x];
3 m_L2 = inv(G' * G) * G' * t;
4
5 % plot 2-norm solution
6 figure(1)
7 plot(x, t, "ro");
8 hold on;
9 t_m = G * m_L2;
10 plot(x, t_m, "-b");
11 hold on;
12
13 % calculate 1-norm solution and plot it
14 m_L1 = IRLS(t, G);
15 plot(x, G * m1, "--g");
16
17 xlabel("Distances/km");
18 ylabel("The first arrival times/s")
19 legend(["data", "L2 solution", "IRLS", "Location", "southeast"])

```

The comparison:

(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

$$\mathbf{m}_{L_1} \pm 1.96 \text{diag}(\text{Cov}(\mathbf{m}_{L_1}))^{1/2}. \quad (13)$$

And

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

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

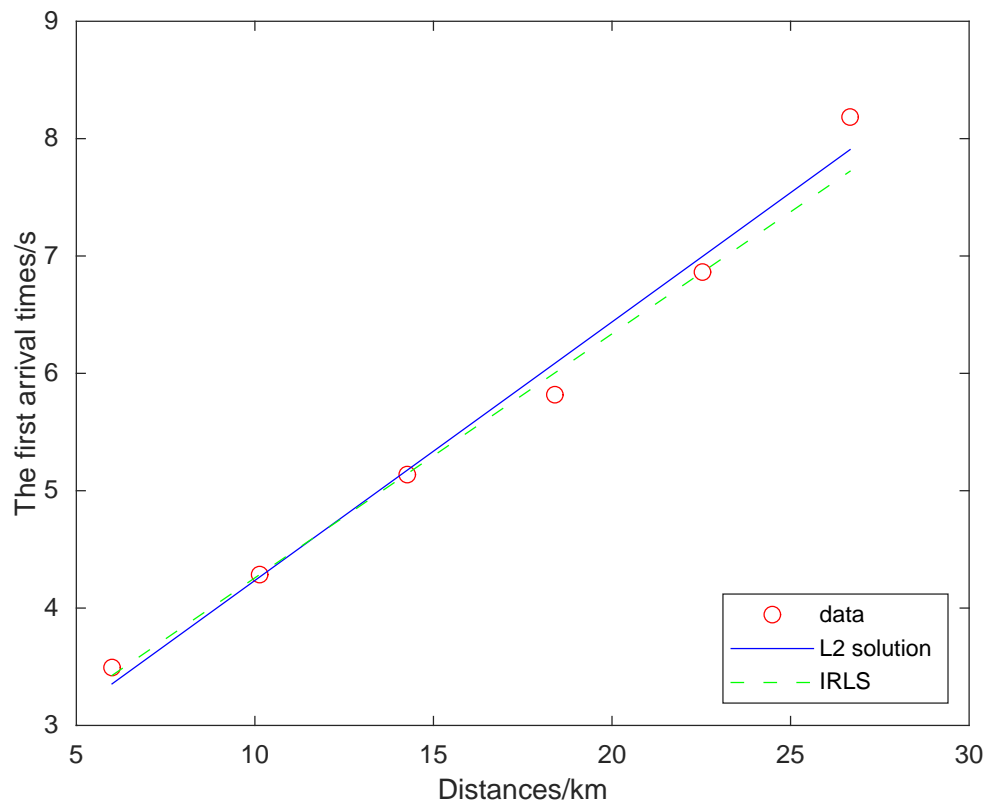
matlab code

```

1 %% 1-h
2 q = 1000;
3 m_all = zeros(q, 2);
4 A = zeros(q, 2);
5
6 for i = 1:1:q
7     noise = Sigma .* randn(6, 1);

```





```

8     t_noise = t + noise;
9     m_all(i, :) = (IRLS(t_noise, G))';
10 end
11
12 A = m_all - mean(m_all);
13
14 CovML1 = A' * A ./ q;
15
16 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

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

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

$$\min(\mathbf{G}\mathbf{m} - \mathbf{d})^T \mathbf{C}_D^{-1} (\mathbf{G}\mathbf{m} - \mathbf{d}). \quad (16)$$

**Solution:**

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

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

$$\mathbf{G}^T \mathbf{C}_D^{-1} \mathbf{G} \mathbf{m} = \mathbf{G}^T \mathbf{C}_D^{-1} \mathbf{d}. \quad (18)$$

**Solution:**

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

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

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

$$\min \left\| \mathbf{C}_D^{-1/2} \mathbf{G} \mathbf{m} - \mathbf{C}_D^{-1/2} \mathbf{d} \right\|_2, \quad (20)$$

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:

$$\mathbf{G}^T \mathbf{G} \mathbf{m} = \mathbf{G}^T \mathbf{d}. \quad (21)$$

So, the linear least squares problem

$$\min \left\| \mathbf{C}_D^{-1/2} \mathbf{G} \mathbf{m} - \mathbf{C}_D^{-1/2} \mathbf{d} \right\|_2, \quad (22)$$

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

$$\min \|\mathbf{R}\mathbf{G}\mathbf{m} - \mathbf{R}\mathbf{d}\|_2 \tag{24}$$

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

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

to the noise-free data points

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

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

```

1 clear;clc;close all;
2
3 x = (-0.95:0.1:0.95)';
4 y = x;
5
6 G = zeros(20,20);
7 for i=1:20
8     G(:, i) = x.^(i-1);
9 end
10
11 m = G \ y;
12 m_true = zeros(20, 1);
13 m_true(2) = 1;
14 y_pre = G*m;
15
16 figure(1)
17 plot(x, y, "ro")
18 hold on;
19 plot(x, y_pre, "-b")

```

```
20 xlabel("x")
21 ylabel("y")
22 legend(["the data", "the fitted model"], "Location", "southeast")
23
24 figure(2)
25 r = m - m_true;
26 a = 0:1:19;
27 plot(a, r, "-o")
28 xlabel("a")
29 ylabel("the difference")
```

