

# 作业 Chapter 3

姓名：范潇 学号：2254298 日期：2024 年 4 月 1 日

## 1. (5.2)

我使用 Python 完成了对本题的求解。

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from scipy import interpolate
5 data = [
6     [370, 470, 550, 600, 670, 690, 670, 620, 580, 450, 400, 300, 100, 150,
7      250],
8     [510, 620, 730, 800, 850, 870, 850, 780, 720, 650, 500, 200, 300, 350,
9      320],
10    [650, 760, 880, 970, 1020, 1050, 1020, 830, 800, 700, 300, 500, 550,
11     480, 350],
12    [740, 880, 1080, 1130, 1250, 1280, 1230, 1040, 900, 500, 700, 780, 750,
13     650, 550],
14    [830, 980, 1180, 1320, 1450, 1420, 400, 1300, 700, 900, 850, 810, 380,
15     780, 750],
16    [880, 1060, 1230, 1390, 1500, 1500, 1400, 900, 1100, 1060, 950, 870,
17     900, 936, 950],
18    [910, 1090, 1270, 1500, 1200, 1100, 1350, 1450, 1200, 1150, 1010, 880,
19     1000, 1050, 1100],
20    [950, 1190, 1370, 1500, 1200, 1100, 1550, 1600, 1550, 1380, 1070, 900,
21     1050, 1150, 1200],
22    [1430, 1450, 1460, 1500, 1550, 1600, 1550, 1600, 1600, 1600, 1550, 1500,
23     1500, 1550, 1550],
24    [1420, 1430, 1450, 1480, 1500, 1550, 1510, 1430, 1300, 1200, 980, 850,
25     750, 550, 500],
26    [1380, 1410, 1430, 1450, 1470, 1320, 1280, 1200, 1080, 940, 780, 620,
27     460, 370, 350],
28    [1370, 1390, 1410, 1430, 1440, 1140, 1110, 1050, 950, 820, 690, 540,
29     380, 300, 210],
30    [4800, 1370, 1390, 1400, 1410, 960, 940, 880, 800, 690, 570, 430, 290,
```

```
210, 150]
19 ]
20 data = pd.DataFrame(data, columns=range(0, 6000, 400), index=range(0, 5200,
    400))
21 x = np.arange(0, 4850, 50)
22 y = np.arange(0, 5650, 50)
23 f = interpolate.interp2d(data.columns, data.index, data, kind='linear')
24 z = f(x, y)
25 fig, ax = plt.subplots()
26 contour = ax.contour(x, y, z)
27 ax.clabel(contour, inline=True, fontsize=10)
28 plt.show()
29 z = pd.DataFrame(z, columns=x, index=y)
30 z.to_csv('height.csv')
```

得到的等高图如下图所示，得到的高程数据存储在文件 height.csv 中。

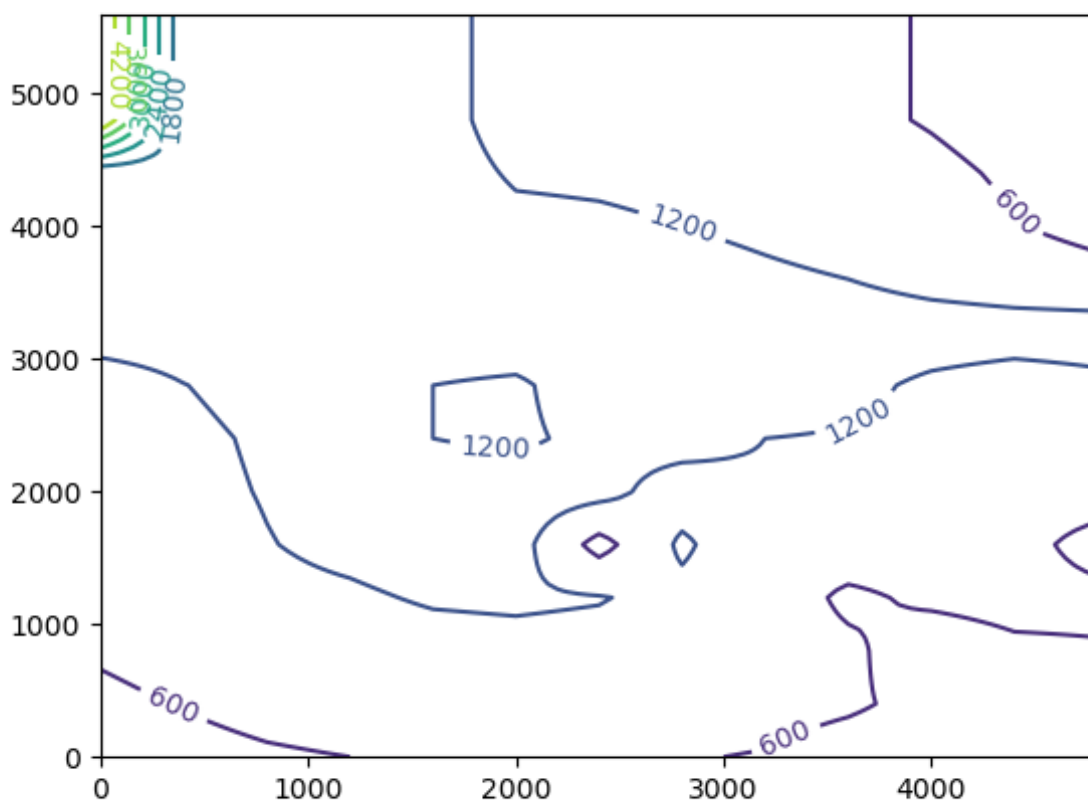


图 1: 等高图

## 2. (5.3)

我使用 Python 完成了对本题的求解。

```
1 import numpy as np
2 from scipy.optimize import curve_fit
3
4 def f(x,a,b):
5     return a*np.e**(b*x)
6
7 x = range(1,9)
8 y = [15.3, 20.5, 27.4, 36.6, 49.1, 65.6, 87.87, 117.6]
9
10 initial_guess = [15,1]
11 fit_params, covariance = curve_fit(f,x,y,initial_guess)
12
13 a_fit,b_fit = fit_params
14 print(a_fit,b_fit)
```

得到的输出为 11.425066497697129 和 0.2914239509980085。

### 3. (5.4)

我使用 Python 完成了对本题的求解。

思路是利用题中所给的数据点（不包括泵水阶段）来拟合出各个时间与水位之间的函数关系，然后通过差分来得到水位差值关于时间的函数，乘以常数后便得到流量的近似函数。

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.interpolate import UnivariateSpline
4 x1 = [0,3316,6635,10619,13937,17921,21240,25223,28543,32284]
5 y1 = [3175,3110,3054,2994,2947,2892,2850,2795,2752,2697]
6 x2 = [39435,43318,44636,49953,53936,57254,60574,64554,68535,71854,75021]
7 y2 = [3550,3445,3350,3260,3167,3087,3012,2927,2842,2767,2697]
8 x3 = [85968,89953,93270]
9 y3 = [3475,3397,3340]
10 spline = UnivariateSpline(x1+x2+x3,y1+y2+y3)
11 x = range(0,89953)
12 y = spline(x)
13 plt.scatter(x = range(len(np.diff(y))),y=np.diff(y))
```

得到的输出如下图所示：

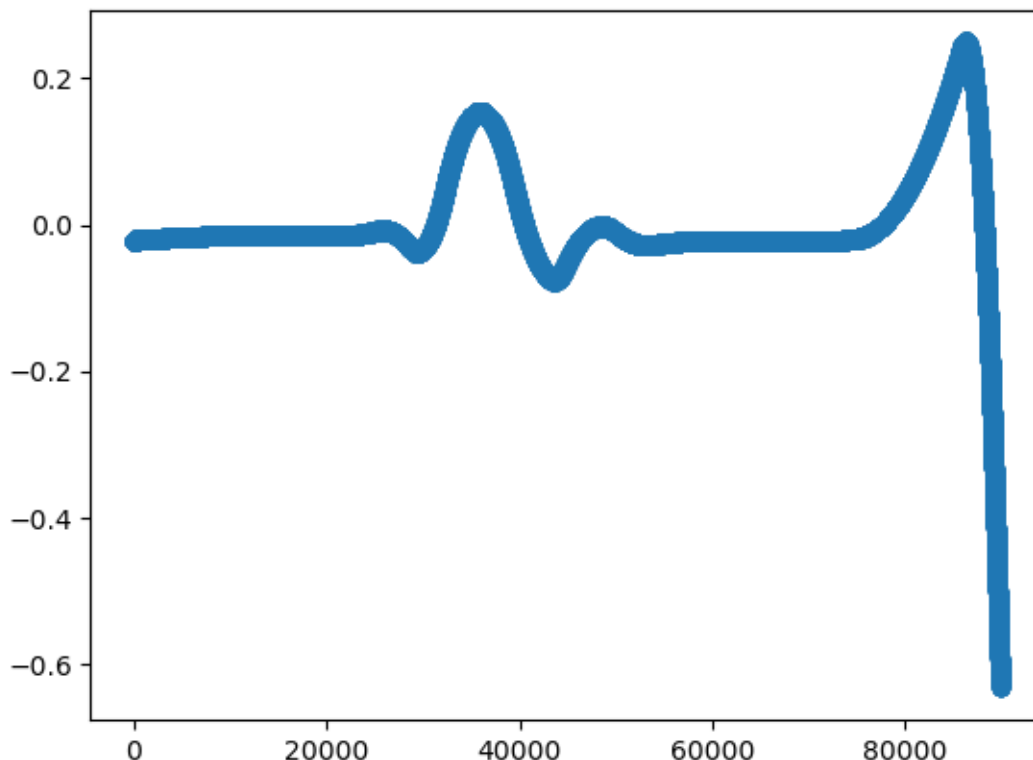


图 2: 时间-水位差值关系

可以看到插值得到的函数在右端偏差较大，但是由于假设（4），只需取  $[0, 86400]$  内的函数即可，而在该范围内的差异偏差较小。

在此基础上，只需乘以常数  $\frac{E^2\pi}{4}$  即可得到流量函数  $f(t)$ 。