武汉大学本科生课程作业

坐标转换与坐标系转换

院　系　测　绘　学　院

学　号　2018302141032

课　程　大地测量学基础

班　级　测绘工程1801班

姓　名　胡　俊　东

二〇二〇年四月

目录

一、程序背景、整体概况及功能简介

1.1 程序背景

在大地测量学中，存在各种不同的坐标系，同时在实际应用时也需要对点的坐标进行转换，故根据《大地测量学基础课程作业-20200303》的要求，结合大地测量学中学习到的知识进行程序设计，开发出一套坐标转换及坐标系转换的程序。

1.2 程序整体概况

该程序在 Microsoft Visual Studio 平台上，使用 C++完成，项目名称为 coordinate_transformation，是一个具有窗体的应用程序。程序运行效果如图。

图 1.1 程序整体界面

1.3 程序功能介绍

1.3.1 主功能按钮

该程序主要功能在右上角。一共有五个功能，即四参数坐标转换，七参数/六参数坐标转换，十三参数坐标转换，地心地固系与大地坐标系转换，地心地固系与

站心坐标系转换，分别对应《大地测量学基础课程作业-20200303》中的五个要求。每一个按钮对应一个需求，简单明了，按下操作按钮后左边的文件显示区会显示出用户导入的文件，同时进度条将根据读入和计算的情况实时显示，完成所有的操作之后，计算出来的结果将显示在右侧的文本框中。

1.3.2 浏览按钮

程序左上角有两个文件选取按钮，通过浏览按钮选取对应文件后，文件的地址会实时显示在浏览按钮左侧的文字框中。

1.3.3 清除按钮

为了避免每一次操作完成后想进行下一次操作必须退出程序重新进入，程序设置了清除按钮。通过清除按钮可以清空所有文本框及回溯进度条，快速开启下一次操作。

1.3.4 进度条

因为特别是在进行地心地固系与大地坐标系转换与地心地固系与站心坐标系转换操作时，需要导入多达 14435 个点，同时对这些点进行计算，所以需要消耗一定的时间，若不进行任何改动，程序看上去就会和卡住一样，很不方便。为了解决这些问题，我在程序中加入进度条控件，按下每一个主操作按钮后，进度条按钮都将实时显示文件的读取情况和计算情况。方便用户查看操作进行情况。

1.3.5 退出按钮

完成所有的之后操作之后，可以点击退出按钮退出程序。

二、四参数坐标转换（小角度）

2.1 算法原理及流程

2.1.1 算法原理

该类型的转换为同一个椭球系统的不同坐标系的转换，亦为二维平面坐标转换，可以推导基本数学模型如下：

$$\begin{bmatrix} X_B \\ Y_B \end{bmatrix} = \begin{bmatrix} X_A \\ Y_A \end{bmatrix} + \begin{bmatrix} 1 & 0 & Y_A & X_A \\ 0 & 1 & -X_A & Y_A \end{bmatrix} \begin{bmatrix} DX \\ DY \\ DR \\ DK \end{bmatrix} \quad (2.1)$$

上式中(X_A, Y_A)为源坐标系坐标，(X_B, Y_B)为目标坐标系坐标，DX、DY为平移参数，DR为旋转参数，DK为尺度因子。不难发现可以将该式子变形成为间接平差的形式，模型如下：

$$\begin{bmatrix} v_A \\ v_B \end{bmatrix} = \begin{bmatrix} 1\,0 & Y_A & X_A \\ 0\,1 & -X_A & Y_A \end{bmatrix} \begin{bmatrix} DX \\ DY \\ DR \\ DK \end{bmatrix} - \begin{bmatrix} X_B - X_A \\ Y_B - Y_A \end{bmatrix} \quad (2.2)$$

因此必要观测量为 4，而要求中给出的参数点也有四个，即得到 8 个参数，故多余观测量为 4，其中的 B，L 矩阵如下

B 矩阵：$\begin{bmatrix} 1 & 0 & Y_A & X_A \\ 0 & 1 & -X_A & Y_A \end{bmatrix}$ (2.3)　　L 矩阵：$\begin{bmatrix} X_B - X_A \\ Y_B - Y_A \end{bmatrix}$ (2.4)

最终带入 4 个转换公共点，得到的 B 矩阵是一个 8×4 的矩阵，而 L 是一个 8×1 的矩阵，根据间接平差的公式，求出 X 矩阵，即可得到要求取的四参数。

$$X = \left(B^T PB\right)^{-1} B^T PL \quad (2.5)$$

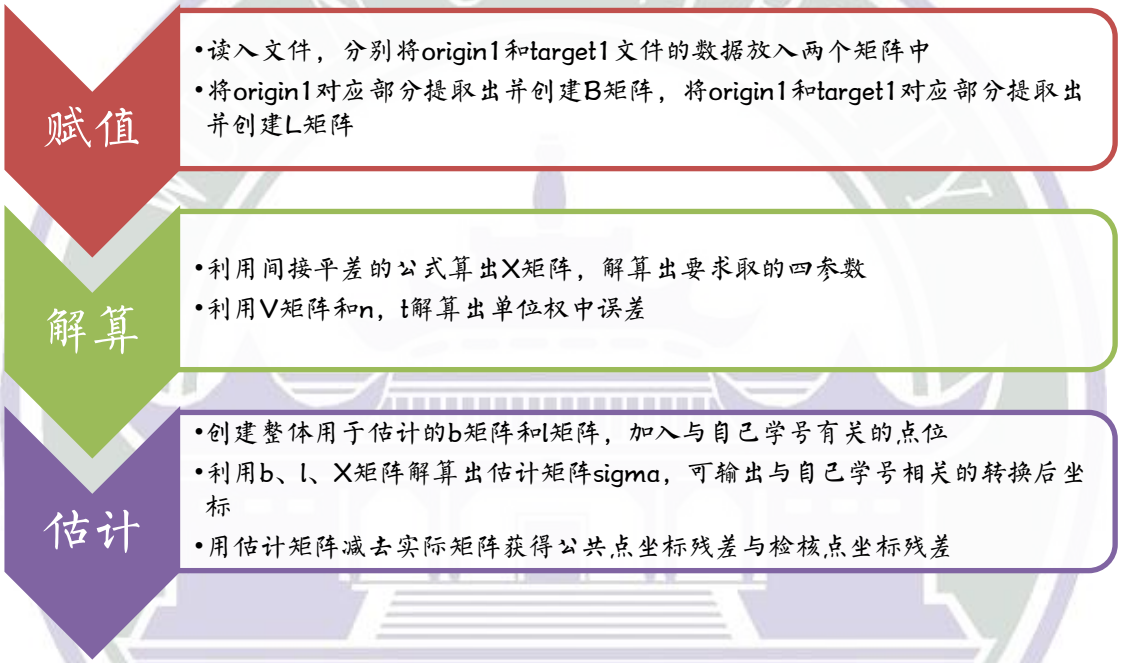
同时运用如下的间接平差单位权中误差公式，利用 V 矩阵和 n，t 即可求得验后

单位权中误差。

$$\sigma=\sqrt{\frac{V^T PV}{n-t}} \quad (2.6)$$

解算完成后，就可以利用得到的四参数开始进行估计，利用 2.1 式，将原来 origin1 文件中的坐标加上有关自己学号的坐标代入计算，即可得到估计矩阵。

2.1.2 算法流程



2.2 关键中间结果及最终结果

2.2.1 程序运行结果

参数值	公共点坐标残差	检核点坐标残差	条件数与冗余度
3682804.259111	−0.002181	−0.053295	n=8，r=4
36390931.706529	0.007126	−0.005840	单位权中误差
−0.008274	−0.000883	−0.098601	0.008868
−0.000033	−0.010639	0.035427	
	0.006881	−0.143980	
	0.007908	0.126331	
	−0.003908	−0.040967	
	−0.004459	0.114554	

2.2.2 特定点运行结果

特定转换点	X	Y	Z
源坐标	36509912.344732	118032.826	0
目的坐标	3796856.072693	36509912.344732	0

三、七参数/六参数坐标转换（小角度）

3.1 算法原理及流程

3.1.1 算法原理

七参数坐标转换模型通常运用于两个不同的三维空间直角坐标系之间转换。我国常用的系统由北京 1954、西安 1980、WCG-84 和 CGCS2000，而这些坐标系统之间的三维坐标转换常用的便是七参数模型。作为一个三维坐标转换模型，其基本的数学模型如下：

$$\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} = \begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 & -Z_A & Y_A & X_A \\ 0 & 1 & 0 & Z_A & 0 & -X_A & Y_A \\ 0 & 0 & 1 & -Y_A & X_A & 0 & Z_A \end{bmatrix} \begin{bmatrix} DX \\ DY \\ DZ \\ RX \\ RY \\ RZ \\ DK \end{bmatrix} \quad (3.1)$$

上式中(X_B, Y_B, Z_B)为目标坐标系坐标，(X_A, Y_A, Z_A)为源坐标系坐标，DX、DY、DZ为平移参数，RX、RY、RZ为旋转参数，Dk为尺度因子。该式子也可以转换为间接平差的模型，变换后的形式如下：

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & -Z_A & Y_A & X_A \\ 0 & 1 & 0 & Z_A & 0 & -X_A & Y_A \\ 0 & 0 & 1 & -Y_A & X_A & 0 & Z_A \end{bmatrix} \begin{bmatrix} DX \\ DY \\ DZ \\ RX \\ RY \\ RZ \\ DK \end{bmatrix} - \begin{bmatrix} X_B - X_A \\ Y_B - Y_A \\ Z_B - Z_A \end{bmatrix} \quad (3.2)$$

七参数模型中通常至少需要知道三个公共已知点，才能解算出转换参数，因为我们用了四组观测值，所以存在多余观测，条件数 n=12，冗余度 r=5。同时我们可以列出 B 矩阵和 L 矩阵，其式如下：

B 矩阵：$$\begin{bmatrix} 1 & 0 & 0 & 0 & -Z_A & Y_A & X_A \\ 0 & 1 & 0 & Z_A & 0 & -X_A & Y_A \\ 0 & 0 & 1 & -Y_A & X_A & 0 & Z_A \end{bmatrix} \quad (3.3)$$

L 矩阵：$$\begin{bmatrix} DX \\ DY \\ DZ \\ RX \\ RY \\ RZ \\ DK \end{bmatrix} \quad (3.4)$$

最终带入 4 个转换公共点，得到的 B 矩阵是一个 12×7 的矩阵，而 L 是一个 7×1 的矩阵，根据间接平差的公式，求出 X 矩阵，即可得到要求取的四参数。

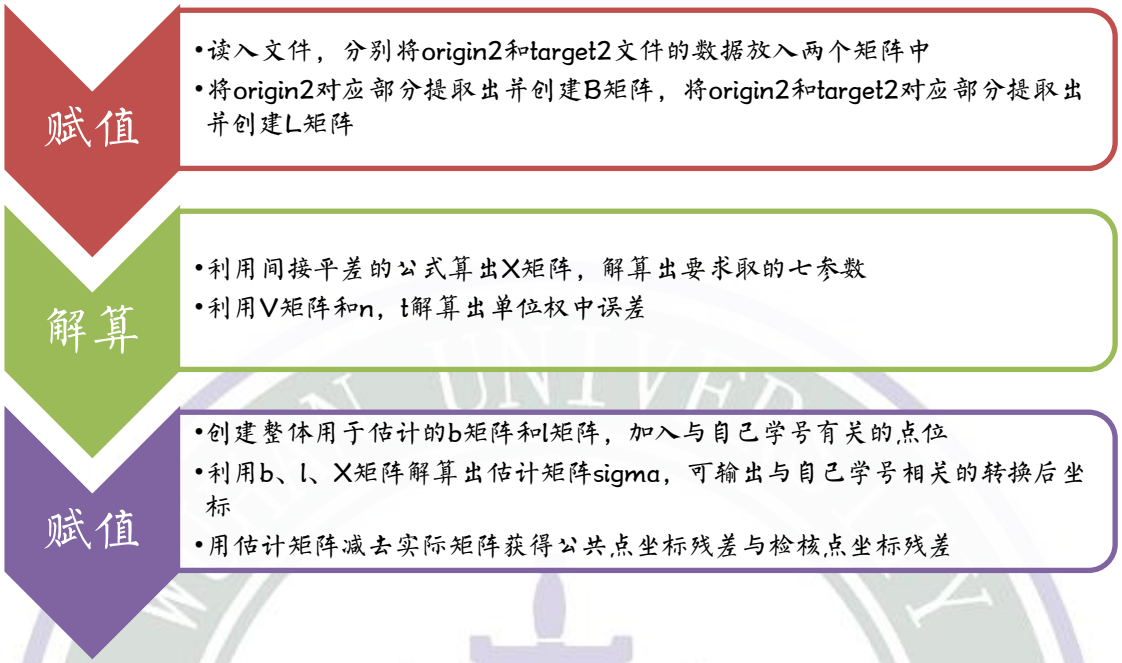
$$X = \left(B^T P B\right)^{-1} B^T P L \quad (3.5)$$

同时运用如下的间接平差单位权中误差公式，利用 V 矩阵和 n，t 即可求得验后单位权中误差。

$$\sigma = \sqrt{\frac{V^T P V}{n - t}} \quad (3.6)$$

解算完成后，就可以利用得到的七参数开始进行估计，利用 3.1 式，将原来 origin2 文件中的坐标加上有关自己学号的坐标代入计算，即可得到估计矩阵。

3.1.2 算法流程



3.2 关键中间结果及最终结果

3.2.1 程序运行结果

参数值	公共点坐标残差	检核点坐标残差	条件数与冗余度
273.189521	−0.007591	0.007620	n=12，r=5
55.158728	0.004946	0.028811	单位权中误差
117.420790	−0.035298	0.046039	0.036074
0.000015	0.006900	0.017410	
0.000016	−0.002123	0.034264	
−0.000022	−0.013493	0.057219	
0.000001	0.028324		
	0.011711		
	0.054729		
	−0.027633		
	−0.014533		
	−0.005938		

3.2.2 特定点运行结果

特定转换点	X	Y	Z
源坐标	2100032.2849	5496032.0138	2894032.6030
目的坐标	−2099928.558202	5496086.257357	2894036.862331

四、十三参数坐标转换（大角度）

4.1 算法原理及流程

4.1.1 算法原理

在测绘数据处理过程中，时常遇到三维基准转换的问题，如在摄影测量、三维激光扫描或是测量机器人自由设站都会遇到大旋转角的三维直角坐标转换的问题。此时需要引进十三参数模型：

$$\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} = k \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} X_A \\ Y_A \\ Z_A \end{bmatrix} + \begin{bmatrix} DX \\ DY \\ DZ \end{bmatrix} \quad (4.1)$$

也可用旋转距阵 M 表示为：

$$M = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \quad (4.2)$$

若 M 满足：

$$MM^T = M^T M = E \quad (4.3)$$

则 M 为正交矩阵，相应的坐标变换为正交变换。若 M 为正交矩阵，则必存在下列条件：

$$\begin{aligned} a_1^2 + a_2^2 + a_3^2 &= 1 \\ b_1^2 + b_2^2 + b_3^2 &= 1 \\ c_1^2 + c_2^2 + c_3^2 &= 1 \\ a_1a_2 + b_1b_2 + c_1c_2 &= 0 \\ a_1a_3 + b_1b_3 + c_1c_3 &= 0 \\ a_2a_3 + b_2b_3 + c_2c_3 &= 0 \end{aligned} \quad (4.4)$$

设未知数为 3 个平移参数、1 个尺度参数、9 个方向余弦参数，则式子（4.1）用泰勒级数展开，可得：

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_0{}^0 \\ Y_0{}^0 \\ Z_0{}^0 \end{bmatrix} + \mu^0 \begin{bmatrix} a_1{}^0 & a_2{}^0 & a_3{}^0 \\ b_1{}^0 & b_2{}^0 & b_3{}^0 \\ c_1{}^0 & c_2{}^0 & c_3{}^0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} dX_0 \\ dY_0 \\ dZ_0 \end{bmatrix} + \begin{bmatrix} a_1{}^0 x_i + a_2{}^0 y_i + a_3{}^0 z_i \\ b_1{}^0 x_i + b_2{}^0 y_i + b_3{}^0 z_i \\ c_1{}^0 x_i + c_2{}^0 y_i + c_3{}^0 z_i \end{bmatrix} d\mu +$$

$$\begin{bmatrix} \mu^0 x_i & \mu^0 y_i & \mu^0 z_i & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu^0 x_i & \mu^0 y_i & \mu^0 z_i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mu^0 x_i & \mu^0 y_i & \mu^0 z_i \end{bmatrix} \begin{bmatrix} da_1 \\ da_2 \\ da_3 \\ db_1 \\ db_2 \\ db_3 \\ dc_1 \\ dc_2 \\ dc_3 \end{bmatrix} \quad (4.5)$$

式中上标为 0 的数为近似值，前标为 d 的值为改正数，可将（4.5）式改写成误差方程的形式，其式子如下：

$$V_i = A_i X + L_i, i = 1, ..., n \quad (4.6)$$

其中的 A，X，L 矩阵的具体形式为：

$$A_i = \begin{bmatrix} 1 & 0 & 0 & a_1{}^0 x_i + a_2{}^0 y_i + a_3{}^0 z_i & \mu^0 x_i & \mu^0 y_i & \mu^0 z_i & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & b_1{}^0 x_i + b_2{}^0 y_i + b_3{}^0 z_i & 0 & 0 & 0 & \mu^0 x_i & \mu^0 y_i & \mu^0 z_i & 0 & 0 & 0 \\ 0 & 0 & 1 & c_1{}^0 x_i + c_2{}^0 y_i + c_3{}^0 z_i & 0 & 0 & 0 & 0 & 0 & 0 & \mu^0 x_i & \mu^0 y_i & \mu^0 z_i \end{bmatrix} \quad (4.7)$$

$$X = \begin{bmatrix} dX_0 & dY_0 & dZ_0 & d\mu & da_1 & da_2 & da_3 & db_1 & db_2 & db_3 & dc_1 & dc_2 & dc_3 \end{bmatrix} \quad (4.8)$$

$$L_i = \begin{bmatrix} X_0{}^0 \\ Y_0{}^0 \\ Y_0{}^0 \end{bmatrix} + \mu^0 \begin{bmatrix} a_1{}^0 & a_2{}^0 & a_3{}^0 \\ b_1{}^0 & b_2{}^0 & b_3{}^0 \\ c_1{}^0 & c_2{}^0 & c_3{}^0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_i - \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_i \quad (4.9)$$

列出误差方程，可以列出条件方程，条件方程的总体形式如下：

$$BX + W = 0 \quad (4.10)$$

其中 X 的含义同上，B 和 W 的表达式如下：

$$B=\begin{bmatrix} 0&0&0&0&2a_1^0&2a_2^0&2a_3^0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&2b_1^0&2b_2^0&2b_3^0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&2c_1^0&2c_2^0&2c_3^0\\ 0&0&0&0&a_2^0&a_1^0&0&b_2^0&b_1^0&0&c_2^0&c_1^0&0\\ 0&0&0&0&a_3^0&0&a_1^0&b_3^0&0&b_1^0&c_3^0&0&c_1^0\\ 0&0&0&0&0&a_3^0&a_2^0&0&b_3^0&b_2^0&0&c_3^0&c_2^0 \end{bmatrix} \quad (4.11)$$

$$W=\begin{bmatrix} a_1^{02}+a_2^{02}+a_3^{02}-1\\ b_1^{02}+b_2^{02}+b_3^{02}-1\\ c_1^{02}+c_2^{02}+c_3^{02}-1\\ a_1^0a_2^0+b_1^0b_2^0+c_1^0c_2^0\\ a_1^0a_3^0+b_1^0b_3^0+c_1^0c_3^0\\ a_2^0a_3^0+b_2^0b_3^0+c_2^0c_3^0 \end{bmatrix} \quad (4.12)$$

此时，若直接利用附有限制条件的间接平差来求解，容易出现奇异阵难以求逆的问题，于是将条件方程化为伪观测方程，则伪观测方程为：

$$V=BX+W \quad (4.13)$$

其中X，B，W的含义同上，将该伪观测方程与原误差方程结合，化为整体的简介平差误差方程，最终形式如下：

$$V_{all}=B_{ll}X-L_{ll} \quad (4.14)$$

其中X矩阵没有变，含义也同上，但是B矩阵和L矩阵是原误差方程和伪观测方程结合而来，其整体形式如下：

$$B_{all}=\begin{bmatrix} A\\ B \end{bmatrix}=\begin{bmatrix} 1&0&0&a_1^0x_i+a_2^0y_i+a_3^0z_i&\mu^0x_i&\mu^0y_i&\mu^0z_i&0&0&0&0&0&0\\ 0&1&0&b_1^0x_i+b_2^0y_i+b_3^0z_i&0&0&0&\mu^0x_i&\mu^0y_i&\mu^0z_i&0&0&0\\ 0&0&1&c_1^0x_i+c_2^0y_i+c_3^0z_i&0&0&0&0&0&0&\mu^0x_i&\mu^0y_i&\mu^0z_i\\ 0&0&0&0&2a_1^0&2a_2^0&2a_3^0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&2b_1^0&2b_2^0&2b_3^0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&2c_1^0&2c_2^0&2c_3^0\\ 0&0&0&0&a_2^0&a_1^0&0&b_2^0&b_1^0&0&c_2^0&c_1^0&0\\ 0&0&0&0&a_3^0&0&a_1^0&b_3^0&0&b_1^0&c_3^0&0&c_1^0\\ 0&0&0&0&0&a_3^0&a_2^0&0&b_3^0&b_2^0&0&c_3^0&c_2^0 \end{bmatrix}$$

(4.15)

$$L_{all}=\begin{bmatrix}L_i\\W\end{bmatrix} \quad (4.16)$$

最终带入 11 个转换公共点，得到的 B 矩阵是一个 39×13 的矩阵，而 L 是一个 39×1 的矩阵，根据间接平差的公式，求出 X 矩阵，即可得到要求取的十三参数。

$$X=\left(B^TPB\right)^{-1}B^TPL \quad (4.17)$$

同时运用如下的间接平差单位权中误差公式，利用 V 矩阵和 n，t 即可求得验后单位权中误差。

$$\sigma=\sqrt{\frac{V^TPV}{n-t}} \quad (4.18)$$

解算完成后，就可以利用得到的十三参数开始进行估计，利用 4.1 式，将原来 origin3 文件中的坐标代入计算，即可得到估计矩阵。

4.1.2 算法流程

完整流程为：

其中循环流程为：

4.2 关键中间结果及最终结果

4.2.1 参数值

参数	求得的值
平移参数 dX0	108525.459447
平移参数 dY0	96612.203369
平移参数 dZ0	101221.058502
尺度参数 μ	0.993905
旋转参数 a1	1.003822
旋转参数 a2	0.000931
旋转参数 a3	−0.000492
旋转参数 b1	0.004978
旋转参数 b2	−0.000441
旋转参数 b3	1.004702
旋转参数 c1	0.003243
旋转参数 c2	0.991146
旋转参数 c3	−0.002215

4.2.2 公共点坐标残差

点号	X	Y	Z
1	4.459447	1.203369	−0.941498
5	−6.832684	4.902316	1.665745
6	1.361924	−3.335160	−11.454767
7	1.961392	−5.145261	3.994937
8	0.832432	3.825235	4.762417
9	−0.330135	0.383759	−1.344421
10	−0.946196	−0.324155	1.575910
11	−5.504749	−6.246897	4.862312
12	−2.021731	1.102460	8.258477
13	−1.854246	2.497170	−1.335670
14	8.874547	1.137163	10.043441

4.2.3 检核点坐标残差

点号	X	Y	Z
15	9.711980	8.695688	−12.150279
16	8.095919	10.987773	−8.229948
17	23.251546	−1.473827	−25.945970
18	22.446154	8.288697	−31.066482
19	21.045622	14.478596	−14.616778
23	45.300930	1.406357	−39.137914

4.2.4 条件数、冗余度与单位权中误差

条件数	冗余度	单位权中误差
33	20	5.867671

五、地心地固系与大地坐标系转换

5.1 算法原理及流程

地心地固系使用 WGS84 椭球，通过查阅资料可知：

a=6378137.0m;

b = 6356752.3142

且 BLH 的相关值皆可以通过公式进行求解：

$$e^2 = \frac{a^2 - b^2}{a^2} \quad (5.1)$$

大地纬度的转换公式为：

$$B = \arctan\left(\frac{Z + e'^2 \times b \times \sin^3\theta}{\sqrt{X^2 + Y^2} - e^2 \times a \times \cos^3\theta}\right) \quad (5.2)$$

大地经度的转换公式为：

$$L = \arctan\frac{Y}{X} \quad (5.3)$$

大地高的转换公式为：

$$H = \arctan\left(\frac{\sqrt{X^2 + Y^2}}{\cos B}\right) - N \quad (5.4)$$

整体的程序运行流程为：

5.2 关键中间结果及最终结果

前十个点的转换后坐标为：

点号	B	L	H
1	0.443969	0.114446	276.591
2	0.44397	0.114445	276.9411
3	0.44397	0.114444	277.2109
4	0.44397	0.114444	277.3834
5	0.44397	0.114444	277.5288
6	0.44397	0.114444	277.5705
7	0.44397	0.114443	277.6646
8	0.443971	0.114443	277.7539
9	0.443971	0.114443	277.793
10	0.443971	0.114443	277.8158

平面（经度-维度）散点图：

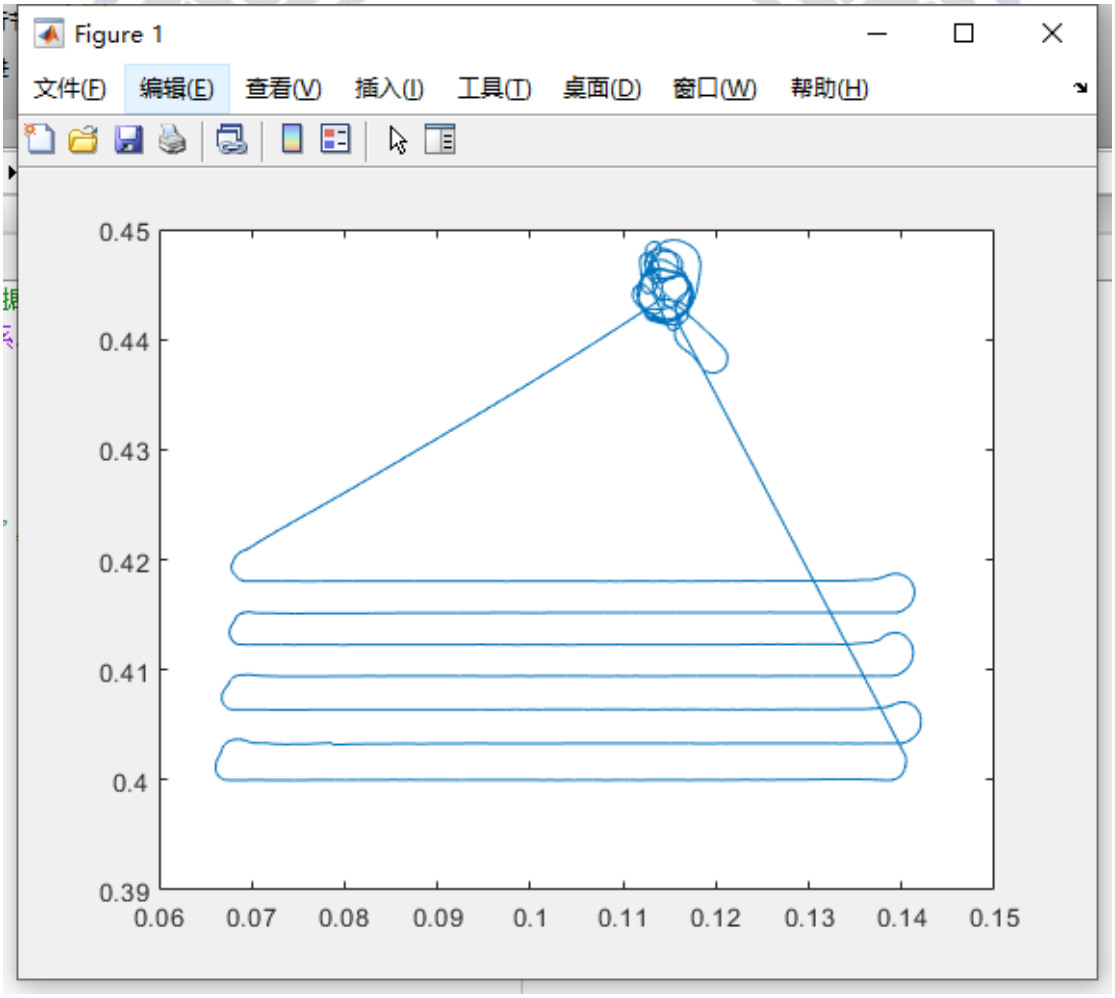


图 5.1 平面（经度-维度）散点图

高程散点图：

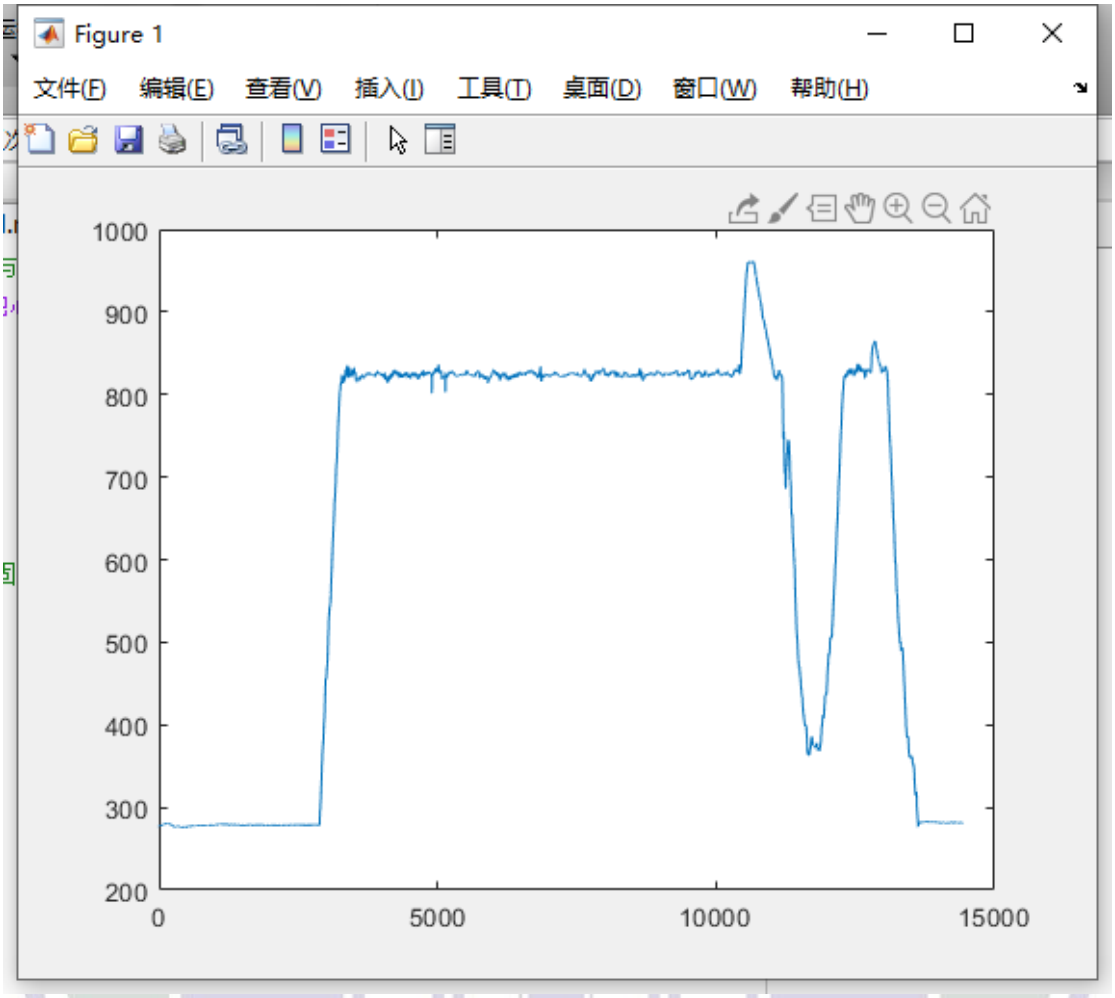


图 5.2 高程散点图

应用猜想：

根据高程散点图，我们不难发现坐标的获取有一个明显的起落过程，同时过程中一部分时间会维持在一个稳定的高度。再观察平面（经度-维度）散点图，中间的运动轨迹非常有规律，就如同组成一个个航带，起末都有一种出发返回的现象。综合考虑，我认为应用场景是在应用无人机进行空中摄影测量时获得的数据。

六、地心地固系与站心坐标系转换

6.1 算法原理及流程

地心地固系使用 WGS84 椭球，通过查阅资料可知：

a=6378137.0m;

b = 6356752.3142

因为已经获得了转换到大地坐标系的算法，可以利用上一步的结果进一步转换，将大地坐标系转换到站心坐标系，有关公式如下：

$$\begin{bmatrix} X_Q \\ Y_Q \\ Z_Q \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} -\sin B\cos L & -\sin L & \cos B\cos L \\ -\sin B\sin L & \cos L & \cos B\sin L \\ \cos B & 0 & \sin B \end{bmatrix} \begin{bmatrix} X^* \\ Y^* \\ Z^* \end{bmatrix}_{PQ} \quad (6.1)$$

其中转换参数的计算公式为：

$$\begin{bmatrix} x^* \\ y^* \\ z^* \end{bmatrix} = \begin{bmatrix} -\sin B\cos L & -\sin B\sin L & \cos B \\ \sin L & \cos L & 0 \\ \cos B\sin L & \cos B\sin L & \sin B \end{bmatrix} \begin{bmatrix} X_Q - X \\ Y_Q - Y \\ Z_Q - Z \end{bmatrix}_{PQ} \quad (6.2)$$

整体程序的运行流程为：

6.2 关键中间结果及最终结果

前十个点的转换后坐标为：

点号	X	Y	Z
1	0	0	0
2	0.096	−0.0558	0.3495
3	0.1257	−0.1234	0.6192
4	0.1557	−0.1578	0.7916
5	0.1504	−0.1994	0.9371
6	0.1589	−0.2179	0.9788
7	0.1654	−0.2442	1.0729
8	0.1835	−0.2619	1.1621
9	0.1863	−0.2733	1.2012
10	0.2035	−0.2745	1.2238

平面散点图：

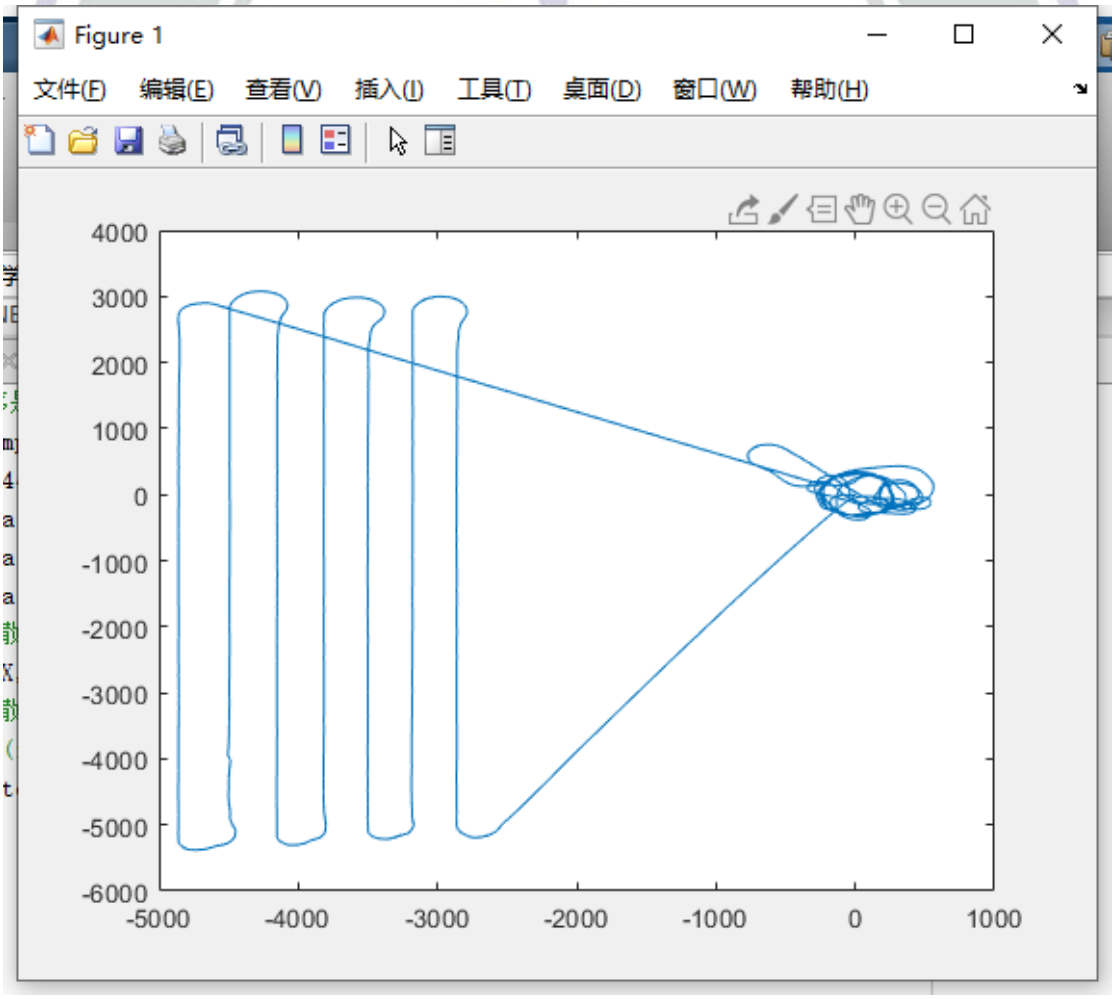


图 6.1 平面散点图

高程散点图：

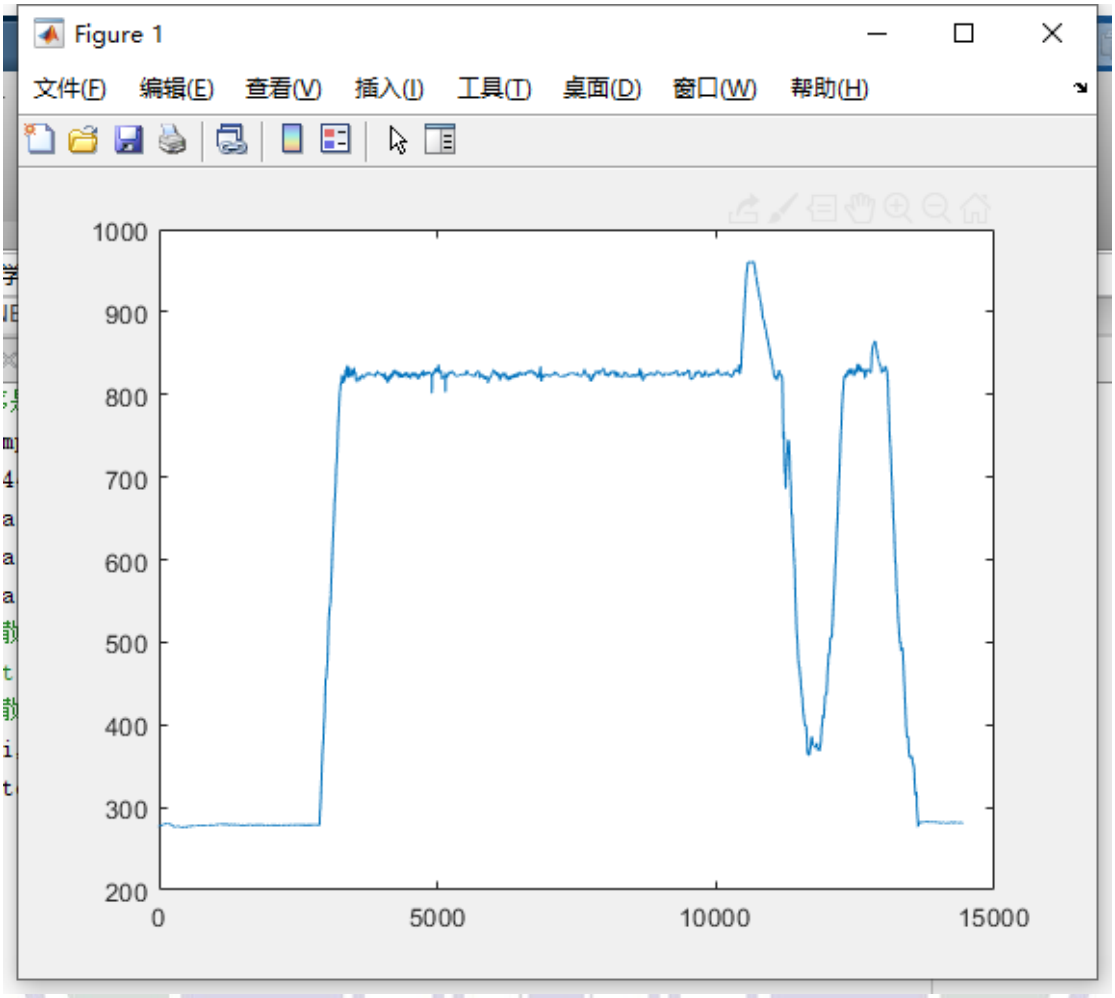


图 6.2 高程散点图

应用场景猜想：

因为与上一个转换的源文件相同，根据高程散点图，我们不难发现坐标的获取有一个明显的起落过程，同时过程中一部分时间会维持在一个稳定的高度。再观察平面（经度-维度）散点图，中间的运动轨迹非常有规律，就如同组成一个个航带，起末都有一种出发返回的现象。综合考虑，我仍认为应用场景是在应用无人机进行空中摄影测量时获得的数据。

七、补充说明

7.1 程序中各个转换运行截图

四参数坐标转换运行结果：

图 7.1 进行四参数转换完成

七参数/六参数坐标转换运行结果：

图 7.2 进行七参数转换完成

十三参数坐标转换运行结果：

图 7.3 进行十三参转换完成

地心地固系与大地坐标系转换运行中：

图 7.4 地心地固系转大地坐标系正在转换

地心地固系与大地坐标系转换运行结果：

图 7.5 地心地固系转大地坐标系转换完成

地心地固系与站心坐标系转换运行中：

图 7.6 地心地固系转站心坐标系正在转换

地心地固系与站心坐标系转换运行结果：

图 7.7 地心地固系转站心坐标系转换完成

7.2 运行辅助说明

因为程序的矩阵运算是依靠 eigen 库实现的，所以若要在其他计算机上实现，需要将本计算机中的 eigen 库添加到路径中，我使用的 eigen 库已打包一并提交，只需要在解决方案管理器中修改附加包含目录，将 eigen 库的文件夹包含进来即可。

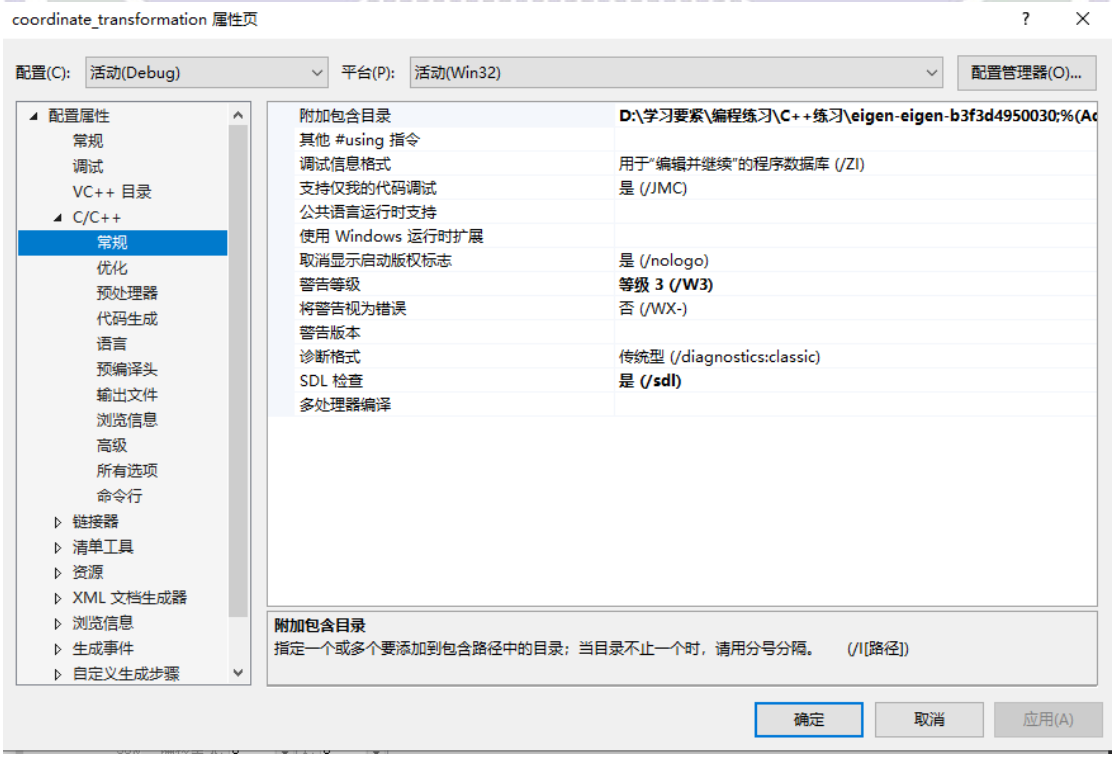


图 7.8 附加包含目录 eigen 库的位置

八、作业感想

8.1 关于 MFC

在完成本个编程作业之前我从来没有运用过用 C++写过 MFC，但是这次想尝试一下，同时也是为了让程序更加人性化，界面更亲和明了，使用了 MFC 程序编写。一路上真的和 C++写控制台有很多不同的地方，因为涉及 CString 的新类型，很多地方要输出就不能简单地 cout 就好，需要运用到 Format 函数进行转换才能放到 edit control 控件里面。因为上个学期我选修了李英冰老师主讲的《网络程序设计》一课，了解一部分 C#的使用，其中控件的使用有很多的相似之处，这次编程才不致于无法完成，因为里面很多思想确实是相通的，帮我少走了很多弯路，让我觉得编程的思想里面真的很多都是相同的，一通百通，编程就是这样，你进门之后就很好走，但是找到门在哪里需要耗费大量的时间，最后做出来的结果还是比较满意的，也算是交上自己满意的答卷吧。

8.2 关于 debug

本次程序的作业里其实前三个功能说白了就是进行平差的过程，我在上一学期学习平差基础的时候编程使用的是 matlab，matlab 有优点就是可以直观而直接地去观察自己赋值完毕的矩阵是什么样子的，debug 就非常方便，而且 matlab 也可以计算接近奇异阵的矩阵的逆，功能真的很强大，以至于我在早期还没有掌握 MFC 里面 debug 的要领时，常常在 matlab 中以相同的步骤，相同的赋值方法进行赋值，然后直接在 matlab 中去寻找哪里有问题，当然这也导致了我的编程量会更多，但是效果还是很好。后来我开始掌握了 C++里面给矩阵 debug 的要领，

通过不同形式创建矩阵，有的形式就可以跟进查看矩阵的问题，在 eigen 库中用不同方式创建矩阵的效果不一样。在本次编程中，十三参中间就曾经出现过 很难发现的 bug，有一个很隐秘的赋值错误，这个错误我整整找了两天时间，才最终找出来，不得不说在编程的过程中需要很细致，不然一个粗心犯的错误很可能需要花掉编这部分三倍以上的时间来修改。

```
MatrixXd X_all(13, 1);//不可以根据去监视查看矩阵的值
Matrix <double,13,1>X_all;//可以根据去监视查看矩阵的值，但是可能有一些意想不到的bug
```

图 8.1 eigen 库中两种不同创建矩阵的利弊

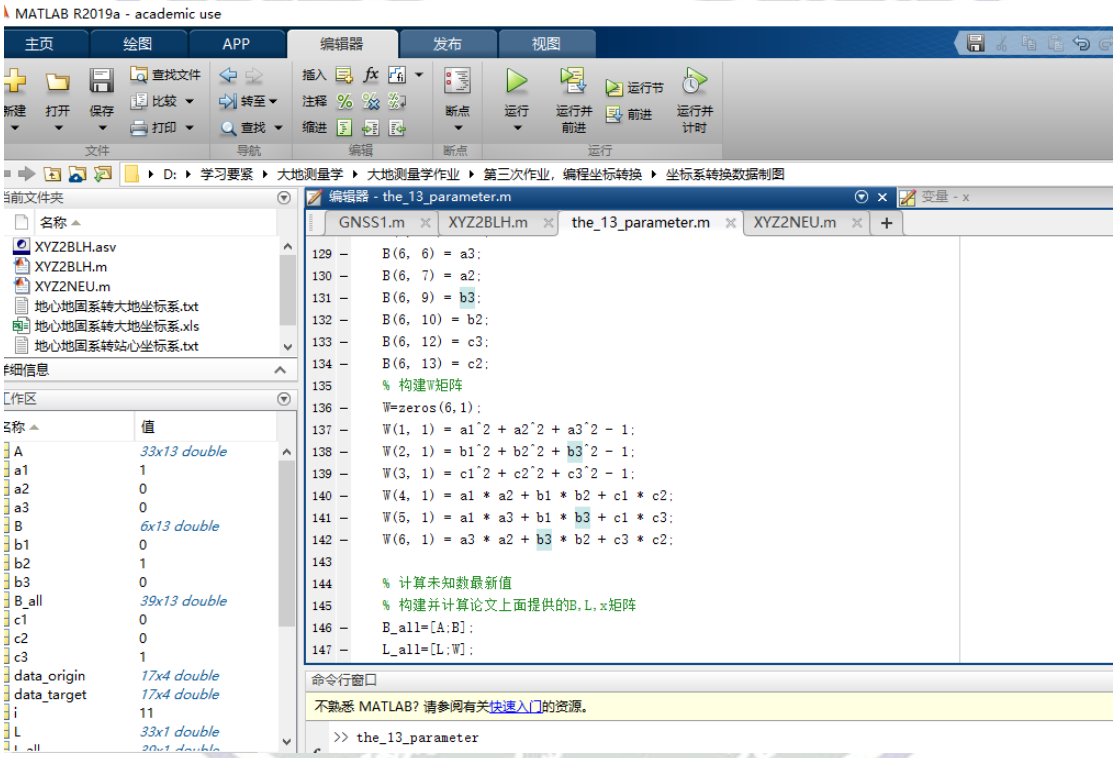


图 8.2 编写十三参时同时写的 matlab 程序

8.3 关于程序整体设计

在编程的过程中，我越来越觉得虽然程序是给计算机来运行，但是最终还是要给人来使用，要为人服务，而这种观点是可以影响到程序整体的布局设计的。例如因为希望程序直观，我采取了更陌生的 MFC 程序，来得到人更愿意接受的窗体，

而不是一个简单的黑底白字的控制台；因为希望能不用因为要读取的目标文件会因为绝对路径的改变而使程序产生 bug，我使用了浏览让用户自行选取文件位置的方法；因为最后两个方法涉及的点数较多，程序需要较长的运行时间，窗体很可能一段时间都不会给用户反馈，这并不是我希望看到的，所以我加入了 process control 控件，实时将程序的运行结果反馈给用户，提供用户的使用体验；因为我不希望每一个功能使用结束后需要关闭程序再重新开始，所以我设计了清除按钮，在每一次转换结束后可以直接清除或者重置，无需重新打开程序即可进行下一次的转换。因为希望尽可能地是程序简洁，每一个转换除开选择文件外，只需要一个按钮就可以自动实现显示导入文件，最终结果等所有功能，增加更多的按钮并不是做不到，但是程序看上去就会很繁琐，程序内部复杂是给开发者的，但是可以做到展现出来是很简洁的。

8.4 关于 mathtype

Mathtype 的使用让我觉得自己的报告更加规范，通过 mathtype 的使用让我对公式的输入有了一个新的方法，虽然用起来确实很繁琐，加个空格还要不断进行数学样式和文本样式的转换，但是一旦习惯之后就会发现还是挺顺手的。同时 mathtype 的使用也让我更加明白了一些其他论文上面公式的写法为什么那么奇怪，例如这次十三参涉及的那篇同济大学的论文，里面表示估计值平方的时候，表示估计值的数字“0”和表示平方的“2”是在一起的，当时给我造成了一定的疑惑，但是现在使用完 mathtype 之后才发现可能就是因为字母右上角只有一个位置，不会堆叠而已。

8.5 结语

本次的程序是我上大学以来写的最大的，代码量最多，持续时间最长的程序，它的代码量达到了 1600 行，再加上我在 matlab 上面用来 debug 的代码，我为了完成程序涉及的代码量高于 2000 行，这是我以前从来没有想到的，但是最后却做到了，对我来说真的意义很大，这个代码我也放到了 github 上面开源，便于备份和共享。我一向认为对自己编程提升最大的一直不是课堂，而是一次次的自己编程和布置的大作业，这次编程编得确实很痛苦，十三参中间的 bug 足足卡了我两天时间，整个过程也是 bug 不断，让我一度怀疑自己到底是在写程序还是在写 bug，但是越艰难，写完之后的成就感就越大，收获也越大。雄关漫道真如铁，而今迈步从头越。从头越，苍山如海，残阳如血。

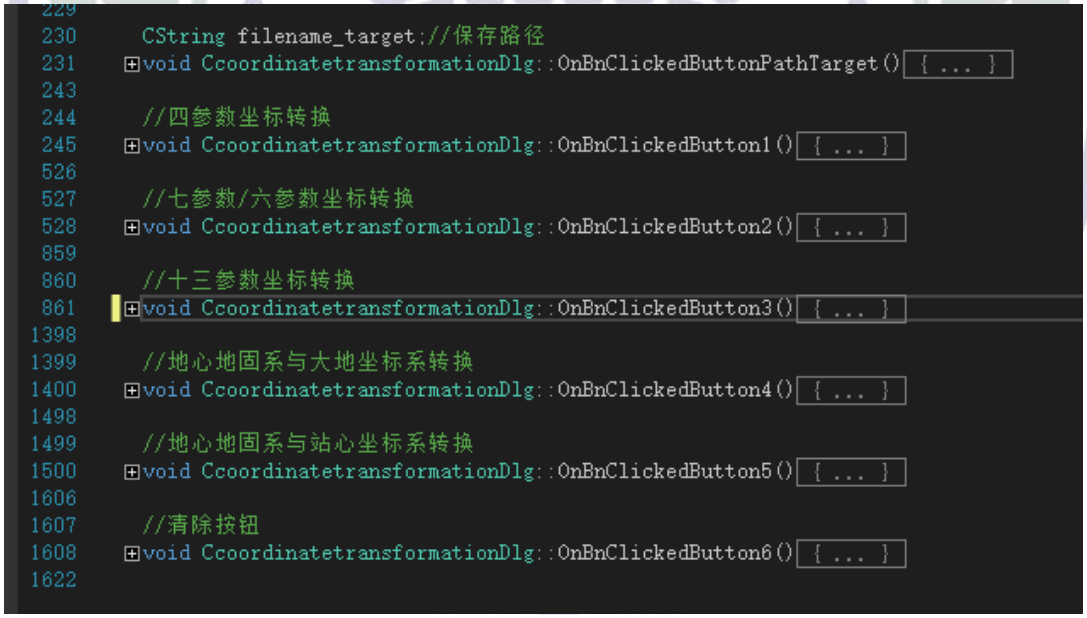


图 8.3 程序整体概括，一共有 1622 行代码

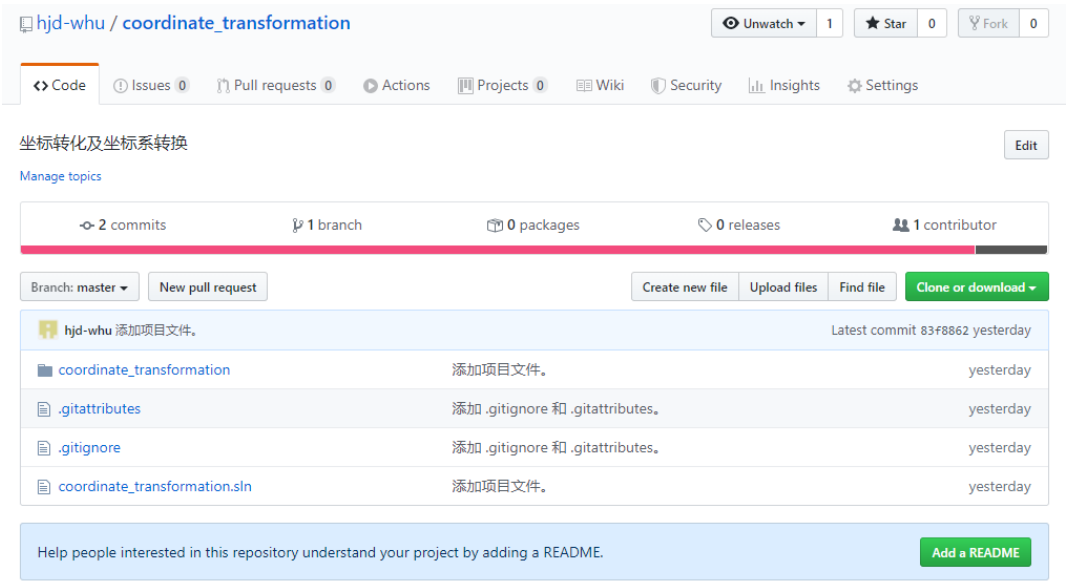


图 8.4 程序在 github 上开源