



河南理工大学  
HENAN POLYTECHNIC UNIVERSITY

## 测绘与国土信息工程学院 课程设计

课程名称: 测量误差与平差基础

开课学期: 2023-2024-1

指导教师: 刘西凤

姓 名: 卢文豪

班 级: 测绘 2204

评语:

成绩:

签字:

日期:

课程设计.....	1
一、课程设计目的与任务.....	3
1.1 课程设计的目的.....	3
1.2 课程设计的任务.....	3
1.3 课程设计的题目.....	3
1.3.1 水准网平差及精度评定程序设计.....	3
1.3.2 导线网平差及精度评定程序设计.....	4
二、课程设计重点与要求.....	6
2.1 课程设计重点.....	6
2.2 课程设计具体内容.....	7
2.3 课程设计的要求.....	7
三、课程设计思路.....	7
3.1 程序设计.....	7
3.2 水准网平差及精度评定.....	8
3.3 导线网平差及精度评定.....	10
四、程序设计流程图.....	14
五、程序源码及说明.....	15
5.1 水准网.....	15
Form1.cs.....	15
5.2 导线网.....	26
Form1.cs.....	26
Algo.cs.....	44
Angle.cs.....	48
AngleEquation.cs.....	50
Edge.cs.....	51
Ellipse.cs.....	52
EquationFormatter.cs.....	53
Point.cs.....	54
六、实例结果分析.....	56
6.1 水准网平差点对比.....	56
6.2 导线网误差椭圆.....	56
七、总结与体会.....	57
八、附件.....	58
8.1 水准网程序界面.....	58
8.2 导线网程序界面.....	60

# 一、课程设计目的与任务

## 1.1 课程设计的目的

误差理论与测量平差是一门理论与实践并重的课程,该课程设计是测量数据处理理论学习的一个重要的实践环节,它是在学生学习了专业基础课“误差理论与测量平差基础”课程后进行的一门实践课程。其目的是增强学生对误差理论与测量平差基础理论的理解,牢固掌握测量平差的基本原理和基本公式,熟悉测量数据处理的基本技能和计算方法,灵活准确地应用于解决各类数据处理的实际问题,并能用所学的计算机理论知识,编制简单的计算程序。

## 1.2 课程设计的任务

(1) 该课的课程设计安排在理论学习结束之后进行的,主要是平面控制网和高程控制网严密平差,时间为一周。

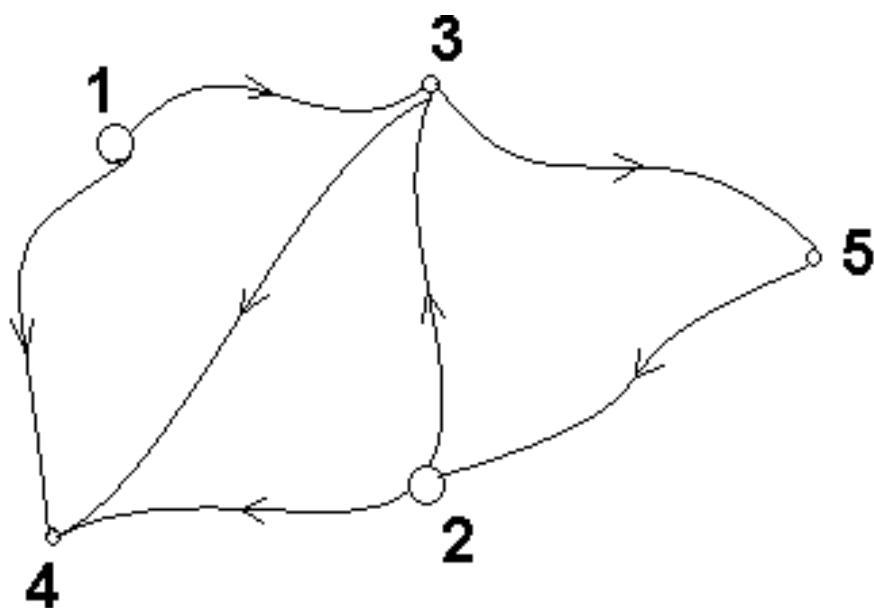
(2) 通过课程设计,培养学生运用本课程基本理论知识和技能,分析和解决本课程范围内的实际工程问题的能力,加深对课程理论的理解与应用。

(3) 在指导老师的指导下,要求每个学生独立完成本课程设计的全部内容。

## 1.3 课程设计的题目

### 1.3.1 水准网平差及精度评定程序设计

如图 1 所示水准网,有 2 个已知点,3 个未知点,7 个测段。各已知数据及观测值见下表一.已知点高程  $H_1=5.016\text{m}$ ,  $H_2=6.016\text{m}$ .



图一 水准网

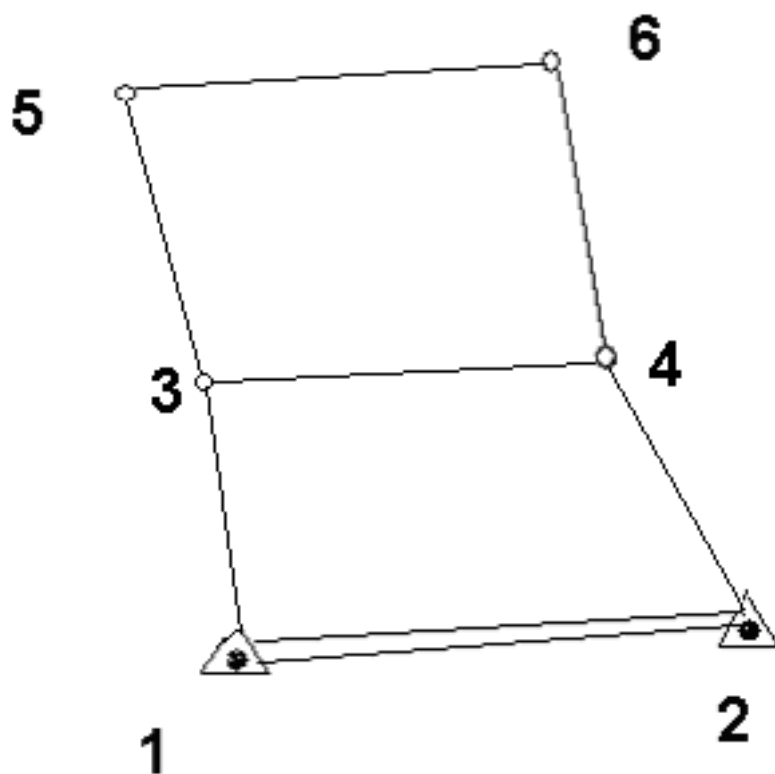
表 1 水准网观测值

端点号	高差观测值	测段距离	序号
1-3	1.359	1.1	1
1-4	2.009	1.7	2
2-3	0.363	2.3	3
2-4	1.012	2.7	4
3-4	0.657	2.4	5
3-5	0.238	1.4	6
5-2	-0.595	2.6	7

求各待定点的高程；3-4 点的高差中误差；3 号点、4 号点的高程中误差。（提示，本网可采用以测段的高差为平差元素，采用间接平差法编写程序计算。）

### 1.3.2 导线网平差及精度评定程序设计

如图 2 所示控制网中，有 2 个已知点，4 个未知点，14 个方向观测值，3 个边长观测值，且方向观测值验前中误差为 1.2 秒，边长观测值固定误差为 0.12 分米，边长观测值比例误差为零。各已知数据、观测值见下表 2、表 3、表 4。



图二 导线网

表 2 已知数据

点号	X (m)	Y (m)
1	121088.500	259894.000
2	127990.100	255874.600

表 3 方向观测值(D.M.S)

测站	照准点	方向值	测站	照准点	方向值
1	2	0.0000	4	6	0.0000
	3	72.10284		3	85.13374
2	4	0.0000		2	217.37126
	1	66.27289	5	6	0.0000
3	1	0.0000		3	79.09487
	4	88.58295	6	4	0.0000
	5	212.10036		5	72.24564

表 4 边长观测值

端点号	边长值	端点号	边长值
4-6	4451.417	2-4	5564.592
6-5	5569.269		

求各待定点的坐标值，评定 4 号点、5 号点的精度，并画出其误差椭圆。

## 二、课程设计重点与要求

### 2.1 课程设计重点

测量平差课程设计要求每一个学生必须遵守课程设计的具体项目的要求，独立完成设计内容，并上交设计报告。在学习知识、培养能力的过程中，树立严谨、求实、勤奋、进取的良好学风。课程设计前学生应认真复习教材有关内容和《误差理论与测量平差基础》课程设计指导书，务必弄清基本概念和本次课程设计的目的、要求及应注意的事项，以保证保质保量的按时完成设计任务。

## 2.2 课程设计具体内容

根据上述的教学目的和任务，本课程设计主要是要求学生完成 2 个综合性的结合生产实践的题目。如目前生产实践中经常用到的水准网严密平差及精度评定，边角网(导线)严密平差及精度评定等内容。重点培养学生正确应用公式、综合分析和解决问题的能力，以及计算机编程能力。

### (1) 水准网平差及精度评定程序设计

根据题目要求，正确应用平差模型列出观测值条件方程、误差方程、法方程和解算法方程，得出平差后的平差值及各待定点的高程平差值，评定各平差值的精度和各高程平差值的精度。

### (2) 导线网平差及精度评定程序设计

选择合适的平差方法，求各待定点的坐标值，评定 1—2 个点的精度，并画出其误差椭圆。

## 2.3 课程设计要求

- (1) 课程设计采用统一封面 A4 纸，左侧装订，由指导老师决定手写或打印。
- (2) 设计方案要保证技术上可行，经济上合理。
- (3) 全部设计内容要求概念清楚，层次分明，论据充分，重点突出。
- (4) 语句简洁通畅，用词准确精练、文字书写工整，标点清晰无误。
- (5) 图表安排合理，统一编号，绘制准确、清晰、美观。

## 三、课程思路

### 3.1 程序设计

#### 1. 界面设计：

主窗口：使用 Windows Forms 应用程序创建一个主窗口，包含菜单栏、状态栏、数据展示区域和结果展示区域。

菜单栏：包含文件、题目、报告、图片、计算和帮助等菜单项，用于实现各种功能。

状态栏：显示当前已知点的信息，如已知点的高程。

数据展示区域：使用 DataGridView 控件显示水准网的数据，包括测段起点、终点、高差、长度等信息。

结果展示区域：使用 RichTextBox 控件显示计算结果，包括各待定点的高程、高差中误差和高程中误差。

## 2. 功能实现：

文件菜单：实现打开数据文件的功能，读取文件中的数据并显示在 DataGridView 控件中。支持多种文件格式，如 CSV、TXT 等。

题目菜单：显示题目描述，让用户了解需要解决的问题。支持题目内容的编辑和保存。

报告菜单：生成平差报告，显示在结果展示区域。支持报告的导出和打印功能。

图片菜单：显示水准网示意图，帮助用户更好地理解问题。支持图片的放大、缩小和保存功能。

计算菜单：实现水准网平差计算的功能，将计算结果显示在结果展示区域。支持多种计算方法和参数设置。

帮助菜单：显示使用说明，帮助用户了解如何使用程序。支持在线帮助和联系开发者功能。

## 3. 数据处理：

使用 MathNet.Numerics.LinearAlgebra 库进行水准网平差计算。根据已知点和路径信息，建立方程组。

## 4. 用户交互：

通过菜单栏和状态栏，用户可以方便地进行文件操作、查看题目描述、生成报告、查看图片、进行计算和查看帮助信息。

通过 DataGridView 控件，用户可以直观地查看水准网的数据。支持数据的排序、筛选和编辑功能。

通过 RichTextBox 控件，用户可以查看计算结果。支持结果的复制、导出和打印功能。

## 5. 程序测试：

在完成程序开发后，进行功能测试，确保每个功能都能正常工作。

进行异常处理，确保程序在遇到错误时能够给出提示，保证程序的稳定性。

进行性能测试，优化程序的性能，提高计算速度和响应速度。

## 6. 程序部署：

将程序打包成安装包，方便用户安装和使用。

# 3.2 水准网平差及精度评定



在图所示水准网中，待定点 5 不是 3 条或 3 条以上水准路线相交的结点，可以把观测高差 3-5 和 5-2 一条水准路线，将最终改正数配给线路 3-5 和 5-2,此处理可以少设一个参数，但不能直接求出待定点 5 高程平差值的中误差。但可以减小一个方程及维度。

1.列出误差方程

设C和D点高程平差值为 $\hat{X}_1$ 和 $\hat{X}_2$ ,相应的近似值取为

$$X_1^0 = H_A + h_1, X_2^0 = H_A + h_2$$

列出观测方程后 ,将有关观测数据代人即得误差方程

$$\begin{aligned} v_1 &= \hat{x}_1 + 0 \\ v_2 &= \hat{x}_2 + 0 \\ v_3 &= \hat{x}_1 - 4 \\ v_4 &= \hat{x}_2 - 3 \\ v_5 &= -\hat{x}_1 + \hat{x}_2 - 7 \\ v'_6 &= -\hat{x}_1 - 2 \end{aligned}$$

式中常数项以 mm 为单位。

2.列出权函数式

3 至 4 间平差 $\hat{h}_5$ 的权函数式为 $\varphi_{\hat{h}_5} = -\hat{x}_1 + \hat{x}_2$

3.组成法方程

以 1km 水准测量的观测高差为单位权观测值,各观测值相互独立, 定权式为  $p=1/s$ ; 得权阵为

$$P = \text{diag} \begin{bmatrix} 0.91 \\ 0.59 \\ 0.43 \\ 0.37 \\ 0.42 \end{bmatrix}$$

由此组成法方程为

$$\begin{bmatrix} 2.01 & -0.42 \\ -0.42 & 1.38 \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} - \begin{bmatrix} -1.72 \\ 4.05 \end{bmatrix} = 0$$

解得

$$\hat{x} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = \begin{bmatrix} -0.258 \\ 2.860 \end{bmatrix} (mm), N_{BB}^{-1} = \begin{bmatrix} 2.01 & -0.42 \\ -0.42 & 1.38 \end{bmatrix}^{-1} = \begin{bmatrix} 0.53 & 0.16 \\ 0.16 & 0.78 \end{bmatrix}$$

4.计算V和 $\hat{L}$

$$\begin{aligned} V &= B\hat{x} - l \\ V_{61} &= [-0.3 \quad 2.9 \quad -4.3 \quad -0.1 \quad -3.9 \quad -1.7]^T (mm) \end{aligned}$$

由  $v'_6 = -1.7$ ,可求得

$$v_6 = \frac{1.4}{4}(-1.7) = -0.6, v_7 = \frac{2.6}{4}(-1.7) = -1.1$$

由此得平差值为

$$\hat{L} = [1.359 \quad 2.012 \quad 0.359 \quad 1.012 \quad 0.653 \quad 0.237 \quad -0.596]^T (\text{m})$$

5.精度评定单位权中误差

$$\hat{\sigma}_0 = \sqrt{\frac{V^T P V}{n - t}} = \sqrt{\frac{19.75}{4}} = 2.2 \text{ mm}$$

C、D点高程中误差

$$\hat{\sigma}_C = 2.2 \sqrt{0.53} = 1.6 \text{ mm} \quad \hat{\sigma}_D = 2.2 \sqrt{0.78} = 1.9 \text{ mm}$$

C至D点高差平差值 $\hat{h}_s$ 的中误差

$$\hat{\sigma}_{\hat{h}_s} = 2.2 \sqrt{0.98} = 2.2 \text{ mm}$$

### 3.3 导线网平差及精度评定

1.计算近似值

连接 1,4 及 4,5 点通过正弦余弦公式算出各边边长

根据已知条件，采用坐标正算和坐标反算计算各点坐标方位角

近似坐标					
点名	X (m)	Y (m)	方向	近似坐标方位角(弧度)	近似边长 (m)
1	121088.500	259894.000	24/12	4.311	7986.718
2	127990.100	255874.600	43/14	3.48	7697.374
3	121565.620	249995.100	46/45	1.826	5987.705
4	122868.420	253699.035	65/13	6.231	6691.317
5	126886.576	258138.295	53/34	4.471	3926.376
6	127175.989	252576.551	31/35	1.891	2769.414

设单位权中误差 $\sigma_0 = 1.2$  秒，则角度观测值的权为

$$P_{\beta_i} = \frac{\sigma_0^2}{\sigma_{\beta}^2} = 1$$

各导线边的权为

$$P_{S_i} = \frac{\sigma_0^2}{\sigma_{S_i}^2} = \frac{1.44}{1.44} = 1 \left( \text{秒}^2 / \text{cm}^2 \right)$$

2.列出方程

$$\hat{\mathbf{L}} = \mathbf{F}(\hat{\mathbf{X}})$$

角度误差方程

距离误差方程

$$\widehat{L}_1 = -(\alpha_{12} - \widehat{\alpha}_{13}) + 360$$

$$\widehat{S}_1 = \sqrt{(\widehat{x}_4 - x_2)^2 + (\widehat{y}_4 - y_2)^2}$$

$$\widehat{L}_2 = \alpha_{21} - \widehat{\alpha}_{24}$$

$$\widehat{S}_2 = \sqrt{(\widehat{x}_6 - \widehat{x}_4)^2 + (\widehat{y}_6 - \widehat{y}_4)^2}$$

$$\widehat{L}_3 = \widehat{\alpha}_{42} - \widehat{\alpha}_{43}$$

$$\widehat{S}_3 = \sqrt{(\widehat{x}_6 - \widehat{x}_5)^2 + (\widehat{y}_6 - \widehat{y}_5)^2}$$

$$\widehat{L}_4 = \widehat{\alpha}_{34} - \widehat{\alpha}_{31}$$

$$\widehat{L}_5 = \widehat{\alpha}_{46} - \widehat{\alpha}_{43}$$

$$\widehat{L}_6 = -(\widehat{\alpha}_{35} - \widehat{\alpha}_{34}) + 360$$

$$\widehat{L}_7 = \widehat{\alpha}_{65} - \widehat{\alpha}_{64}$$

$$\widehat{L}_8 = \widehat{\alpha}_{53} - \widehat{\alpha}_{56}$$

其中j, k 为待定点

$$\widehat{\alpha}_{jk} = \arctan \frac{(Y_k^0 + \widehat{y}_k) - (Y_j^0 + \widehat{y}_j)}{(X_k^0 + \widehat{x}_k) - (X_j^0 + \widehat{x}_j)}$$

对其进行线性化后，可得

$$\widehat{\alpha}_{jk} = \arctan \frac{Y_k^0 - Y_j^0}{X_k^0 - X_j^0} + \left( \frac{\partial \widehat{\alpha}_{jk}}{\partial \widehat{x}_j} \right)_0 \widehat{x}_j + \left( \frac{\partial \widehat{\alpha}_{jk}}{\partial \widehat{y}_j} \right)_0 \widehat{y}_j + \left( \frac{\partial \widehat{\alpha}_{jk}}{\partial \widehat{x}_k} \right)_0 \widehat{x}_k + \left( \frac{\partial \widehat{\alpha}_{jk}}{\partial \widehat{y}_k} \right)_0 \widehat{y}_k$$

得出

$$\delta \alpha''_{jk} = \frac{\rho'' \Delta Y_{jk}^0}{(S_{jk}^0)^2} \widehat{x}_j - \frac{\rho'' \Delta X_{jk}^0}{(S_{jk}^0)^2} \widehat{y}_j - \frac{\rho'' \Delta Y_{jk}^0}{(S_{jk}^0)^2} \widehat{x}_k + \frac{\rho'' \Delta X_{jk}^0}{(S_{jk}^0)^2} \widehat{y}_k \quad (\text{以角度秒为单位})$$

$$L_i + v_i = \hat{\alpha}_{jk} - \hat{\alpha}_{jh}$$

$$\hat{\alpha} = \alpha^0 + \partial \alpha$$

对于方位角  $\alpha$  注意象限角与方位角的关系

距离误差方程

$$\hat{L}_i = L_i + v_i = \sqrt{(\hat{X}_k - \hat{X}_j)^2 + (\hat{Y}_k - \hat{Y}_j)^2}$$

进行线性化

$$L_i + v_i = S_{jk}^0 + \frac{\Delta X_{jk}^0}{S_{jk}^0} (\hat{x}_k - \hat{x}_j) + \frac{\Delta Y_{jk}^0}{S_{jk}^0} (\hat{y}_k - \hat{y}_j)$$

最后得到整体的误差方程

$$L_1 + v_1 = -\frac{\rho'' \Delta Y_{13}^0}{(S_{13}^0)^2} \hat{X}_3 + \frac{\rho'' \Delta X_{13}^0}{(S_{13}^0)^2} \hat{Y}_3 + \alpha_{13}^0 + 360 - \alpha_{12}$$

$$L_2 + v_2 = -\left(-\frac{\rho'' \Delta Y_{24}^0}{(S_{24}^0)^2} \hat{X}_4 + \frac{\rho'' \Delta X_{24}^0}{(S_{24}^0)^2} \hat{Y}_4\right) + \alpha_{21} - \alpha_{24}^0$$

$$L_3 + v_3 = \frac{\rho'' \Delta Y_{42}^0}{(S_{42}^0)^2} \hat{X}_4 - \frac{\rho'' \Delta X_{42}^0}{(S_{42}^0)^2} \hat{Y}_4 - \left(\frac{\rho'' \Delta Y_{43}^0}{(S_{43}^0)^2} \hat{X}_4 - \frac{\rho'' \Delta X_{43}^0}{(S_{43}^0)^2} \hat{Y}_4 - \frac{\rho'' \Delta Y_{43}^0}{(S_{43}^0)^2} \hat{X}_3 + \frac{\rho'' \Delta X_{43}^0}{(S_{43}^0)^2} \hat{Y}_3\right) + \alpha_{42}^0 - \alpha_{43}^0$$

$$L_4 + v_4 = \frac{\rho'' \Delta Y_{34}^0}{(S_{34}^0)^2} \hat{X}_3 - \frac{\rho'' \Delta X_{34}^0}{(S_{34}^0)^2} \hat{Y}_3 - \frac{\rho'' \Delta Y_{34}^0}{(S_{34}^0)^2} \hat{X}_4 + \frac{\rho'' \Delta X_{34}^0}{(S_{34}^0)^2} \hat{Y}_4 - \left(\frac{\rho'' \Delta Y_{31}^0}{(S_{31}^0)^2} \hat{X}_3 - \frac{\rho'' \Delta X_{31}^0}{(S_{31}^0)^2} \hat{Y}_3\right) + \alpha_{34}^0 - \alpha_{31}^0$$

$$L_5 + v_5 = \frac{\rho'' \Delta Y_{46}^0}{(S_{46}^0)^2} \hat{X}_4 - \frac{\rho'' \Delta X_{46}^0}{(S_{46}^0)^2} \hat{Y}_4 - \frac{\rho'' \Delta Y_{46}^0}{(S_{46}^0)^2} \hat{X}_6 + \frac{\rho'' \Delta X_{46}^0}{(S_{46}^0)^2} \hat{Y}_6 - \left(\frac{\rho'' \Delta Y_{43}^0}{(S_{43}^0)^2} \hat{X}_4 - \frac{\rho'' \Delta X_{43}^0}{(S_{43}^0)^2} \hat{Y}_4 - \frac{\rho'' \Delta Y_{43}^0}{(S_{43}^0)^2} \hat{X}_3 + \frac{\rho'' \Delta X_{43}^0}{(S_{43}^0)^2} \hat{Y}_3\right) + \alpha_{46}^0 - \alpha_{43}^0$$

$$L_6 + v_6 = \frac{\rho'' \Delta Y_{34}^0}{(S_{34}^0)^2} \hat{X}_3 - \frac{\rho'' \Delta X_{34}^0}{(S_{34}^0)^2} \hat{Y}_3 - \frac{\rho'' \Delta Y_{34}^0}{(S_{34}^0)^2} \hat{X}_4 + \frac{\rho'' \Delta X_{34}^0}{(S_{34}^0)^2} \hat{Y}_4$$

$$- \left( \frac{\rho'' \Delta Y_{35}^0}{(S_{35}^0)^2} \hat{X}_3 - \frac{\rho'' \Delta X_{35}^0}{(S_{35}^0)^2} \hat{Y}_3 - \frac{\rho'' \Delta Y_{35}^0}{(S_{35}^0)^2} \hat{X}_5 + \frac{\rho'' \Delta X_{35}^0}{(S_{35}^0)^2} \hat{Y}_5 \right) + \alpha_{34}^0 - \alpha_{35}^0 + 360$$

$$L_7 + v_7 = \frac{\rho'' \Delta Y_{65}^0}{(S_{65}^0)^2} \hat{X}_6 - \frac{\rho'' \Delta X_{65}^0}{(S_{65}^0)^2} \hat{Y}_6 - \frac{\rho'' \Delta Y_{65}^0}{(S_{65}^0)^2} \hat{X}_5 + \frac{\rho'' \Delta X_{65}^0}{(S_{65}^0)^2} \hat{Y}_5$$

$$- \left( \frac{\rho'' \Delta Y_{64}^0}{(S_{64}^0)^2} \hat{X}_6 - \frac{\rho'' \Delta X_{64}^0}{(S_{64}^0)^2} \hat{Y}_6 - \frac{\rho'' \Delta Y_{64}^0}{(S_{64}^0)^2} \hat{X}_4 + \frac{\rho'' \Delta X_{64}^0}{(S_{64}^0)^2} \hat{Y}_4 \right) + \alpha_{65}^0 - \alpha_{64}^0$$

$$L_8 + v_8 = \frac{\rho'' \Delta Y_{53}^0}{(S_{53}^0)^2} \hat{X}_5 - \frac{\rho'' \Delta X_{53}^0}{(S_{53}^0)^2} \hat{Y}_5 - \frac{\rho'' \Delta Y_{53}^0}{(S_{53}^0)^2} \hat{X}_3 + \frac{\rho'' \Delta X_{53}^0}{(S_{53}^0)^2} \hat{Y}_3$$

$$- \left( \frac{\rho'' \Delta Y_{56}^0}{(S_{56}^0)^2} \hat{X}_5 - \frac{\rho'' \Delta X_{56}^0}{(S_{56}^0)^2} \hat{Y}_5 - \frac{\rho'' \Delta Y_{56}^0}{(S_{56}^0)^2} \hat{X}_6 + \frac{\rho'' \Delta X_{56}^0}{(S_{56}^0)^2} \hat{Y}_6 \right) + \alpha_{53}^0 - \alpha_{56}^0$$

最终得到:

$$v_1 = -0.276x_1 + 0.048y_1 - 1.47$$

$$v_2 = 0.219x_2 - 0.563y_2 + 0.07$$

$$v_3 = 0.284x_1 + 0.515y_1 - 0.008x_2 - 0.563y_2 + 3.4$$

$$v_4 = 0.391x_1 + 1.274y_2 + 0.008x_2 + 0.563y_2 - 0.399x_3 + 0.712y_3 - 3.4$$

$$v_5 = 0.008x_1 + 0.563y_1 - 0.453x_2 - 0.434y_2 + 0.445x_4 - 0.129y_4$$

$$v_6 = -0.008x_1 - 0.563y_1 - 0.211x_2 + 0.862y_2$$

$$v_7 = -0.399x_1 + 0.712y_1 + 0.604x_3 - 0.403y_3 - 0.206x_4 - 0.308y_4$$

$$v_8 = 0.445x_2 - 0.129y_2 - 0.206x_3 - 0.308y_3 - 0.239x_4 + 0.437y_4$$

由:

$$B^T P B \tilde{x} - B^T P l = 0$$

$$\tilde{x} = (B^T P B)^{-1} B^T P l = N_{BB}^{-1} W$$

解得:

点号	X (m)	Y (m)
3	121565. 636	249995. 122

4	122868.443	253698.946
5	126886.487	258138.106
6	127175.799	252576.496

又有:

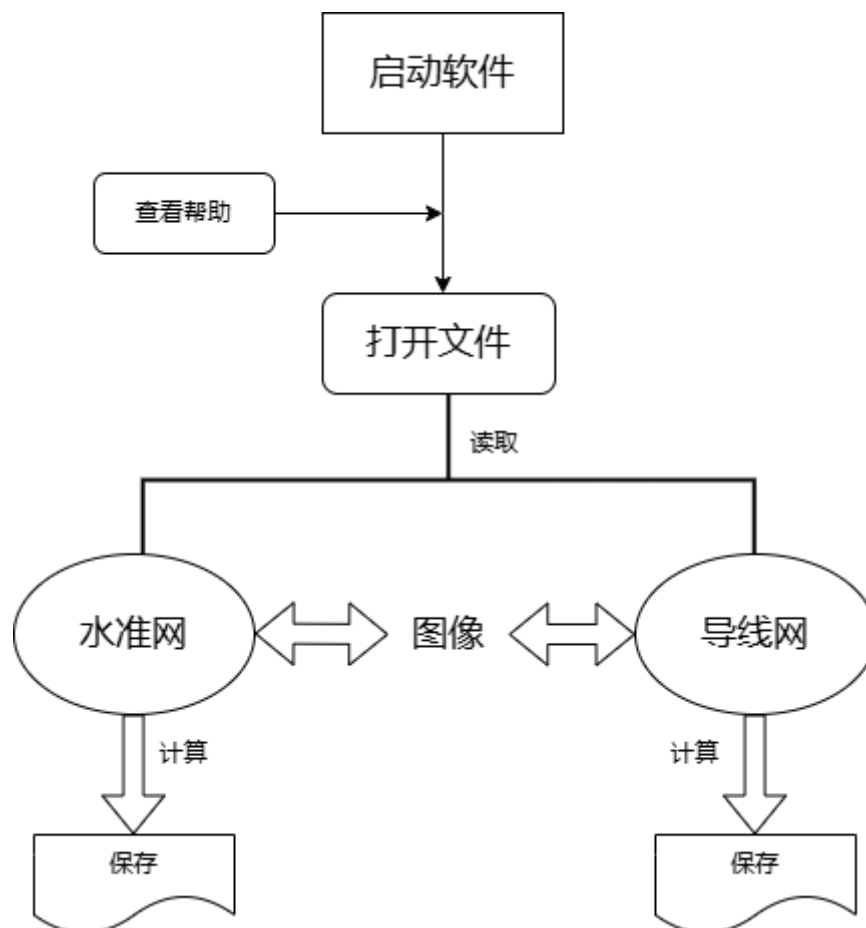
$$\hat{\sigma}_0 = \sqrt{\frac{V^T P V}{n - t}}$$

$$D_{\hat{X}\hat{X}} = \hat{\sigma}_0^2 Q_{\hat{X}\hat{X}} = \hat{\sigma}_0^2 N_{BB}^{-1}$$

得

$$\hat{\sigma}_0 = 13.5\text{mm}$$

#### 四、程序设计流程图



## 五、程序源码及说明

### 5.1 水准网

#### Form1.cs

```
1. using System;
2. using MathNet.Numerics.LinearAlgebra;
3. using MathNet.Numerics.LinearAlgebra.Double;
4. using System;
5. using System.Collections.Generic;
6. using System.Diagnostics;
7. using System.Drawing;
8. using System.Drawing.Imaging;
9. using System.IO;
10. using System.Linq;
11. using System.Text;
12. using System.Windows.Forms;
13. using System.Windows.Forms.DataVisualization.Charting;
14.
15. namespace 水准网平差及精度评定程序
16. {
17.     public partial class Form1 : Form
18.     {
19.         string[] all_lines;
20.         Dictionary<string, string> knownPointsDir = new Dictionary<string
, string>();
21.         double[] L = new double[7];
22.         List<double> yuanheight = new List<double>();
23.         public Form1()
24.         {
25.             InitializeComponent();
26.         }
27.
28.         private void 打开文件
ToolStripMenuItem_Click(object sender, EventArgs e)
29.         {
30.             OpenFileDialog op = new OpenFileDialog();
31.             if (op.ShowDialog() == DialogResult.OK)
32.             {
33.                 all_lines = File.ReadAllLines(op.FileName, Encoding.Defau
lt);
34.             }
35.             else
```

```

36.         {
37.             return;
38.         }
39.     try
40.     {
41.         //切割已知点
42.         string firstLine = all_lines[0];
43.         string[] firstLineParts = firstLine.Split(',');
44.         for (int i = 0; i < firstLineParts.Length - 1; i += 2)
45.         {
46.             if (i + 1 < firstLineParts.Length)
47.             {
48.                 knownPointsDir.Add(firstLineParts[i], firstLinePa
49.                 rts[i + 1]);
50.             }
51.         }
52.         //切割路径数据
53.         for (int i = 2; i < all_lines.Length; i++)
54.         {
55.             string line = all_lines[i];
56.             string[] parts = line.Split(',');
57.             if (parts.Length == 5)
58.             {
59.
60.                 dataGridView1.Rows.Add(parts[0], parts[1], parts[
61.                 2], parts[3], parts[4]);
62.             }
63.             //显示已知点
64.             string knownPoints = null;
65.             foreach (var pair in knownPointsDir)
66.             {
67.                 string temp = pair.Key + ':' + pair.Value + ' ';
68.                 knownPoints += temp;
69.             }
70.             toolStripStatusLabel1.Text = knownPoints;
71.         }
72.     catch
73.     {
74.         MessageBox.Show("文件格式错误!");
75.         toolStripStatusLabel1.Text = "文件格式错误!";
76.         return;
77.     }

```



```

78.         tabControl1.SelectedTab = tabPage1;
79.     }
80.
81.     private void 路径信息
        ToolStripMenuItem_Click(object sender, EventArgs e)
82.     {
83.         pictureBox1.Image = Properties.Resources.shuizhun;
84.         tabControl1.SelectedTab = tabPage3;
85.     }
86.
87.     private void 题目
        ToolStripMenuItem_Click(object sender, EventArgs e)
88.     {
89.
90.         string text = "如图所示水准网，有 2 个已知点，3 个未知点，7 个测段。
            各已知数据及观测值见下表。 \n\n" +
91.             "已知点高程：H1=5.016m，H2=6.016m\n\n" +
92.             "求解： \n" +
93.             "(1) 求各待定点的高程； \n" +
94.             "(2) 3-4 点的高差中误差； \n" +
95.             "(3) 3 号点、4 号点的高程中误差。 \n";
96.
97.         richTextBox1.Text = text;
98.         tabControl1.SelectedTab = tabPage2;
99.
100.    }
101.
102.    private void 报告
        ToolStripMenuItem1_Click(object sender, EventArgs e)
103.    {
104.        string newText = "n=7, t=3, u=3, r=4.方程数为 7。 \n" +
105.            "待定点 5 不是三条及三条以上水准路线交点 \n" +
106.            "把 h6 和 h7 合并为一条水准路线 \n" +
107.            "设 3, 4 点高程依次为 x1, x2" + '\n';
108.
109.        richTextBox1.AppendText(newText);
110.    }
111.
112.    private void 图片
        ToolStripMenuItem_Click(object sender, EventArgs e)
113.    {
114.        pictureBox1.Image = Properties.Resources.func;
115.        tabControl1.SelectedTab = tabPage3;
116.    }

```

```

117.
118.         private void 计算
           ToolStripMenuItem_Click(object sender, EventArgs e)
119.         {
120.             if (knownPointsDir.Count == 0 || dataGridView1.Rows.Count ==
                0)
121.             {
122.                 MessageBox.Show("请先加载文件数据!");
123.                 return;
124.             }
125.
126.             Dictionary<string, double> unknownPointsHeights = new Dictio
                nary<string, double>();
127.
128.             double HA = double.Parse(knownPointsDir.Values.ElementAt(0))
                ;
129.             double HB = double.Parse(knownPointsDir.Values.ElementAt(1))
                ;
130.             List<double> measurements = new List<double>();
131.
132.             foreach (DataGridViewRow row in dataGridView1.Rows)
133.             {
134.                 if (row.IsNewRow) continue;
135.
136.                 double h = double.Parse(row.Cells[2].Value.ToString());
137.                 measurements.Add(h);
138.             }
139.
140.             // 初始值
141.             double x1 = HA + measurements[0];
142.             double x2 = HA + measurements[1];
143.
144.             unknownPointsHeights["x1"] = x1;
145.             unknownPointsHeights["x2"] = x2;
146.             List<double> length = new List<double>();
147.             // 记录长度
148.             foreach (DataGridViewRow row in dataGridView1.Rows)
149.             {
150.                 if (row.IsNewRow) continue;
151.                 double value = double.Parse(row.Cells[3].Value.ToString(
                    ));
152.                 length.Add(value);
153.             }
154.

```

```

155.         List<double> height = new List<double>();
156.         // 记录高差
157.         foreach (DataGridViewRow row in dataGridView1.Rows)
158.         {
159.             if (row.IsNewRow) continue;
160.             double h = double.Parse(row.Cells[2].Value.ToString());
161.             height.Add(Math.Round(h, 3));
162.         }
163.         yuanheight.AddRange(height);
164.         double[] l = new double[] { 0, 0, -4, -3, -7, -2 };
165.
166.         // 计算矩阵P
167.         double[,] P = new double[6, 6];
168.         double l67 = length[5] + length[6];
169.         List<double> length67 = new List<double>(length.GetRange(0,
170.             5));
171.         length67.Add(l67); // 加入 67 总距离
172.         for (int i = 0; i < 6; i++)
173.         {
174.             P[i, i] = Math.Round(1 / length67[i], 3);
175.         }
176.         // 将B和P转换为Matrix, 将l转换为Vector
177.         var B = DenseMatrix.OfArray(new double[,])
178.         {
179.             { 1, 0 },
180.             { 0, 1 },
181.             { 1, 0 },
182.             { 0, 1 },
183.             { -1, 1 },
184.             { -1, 0 },
185.         });
186.         Matrix<double> PMatrix = Matrix<double>.Build.DenseOfArray(P
187.             );
188.         Vector<double> lVector = Vector<double>.Build.Dense(l);
189.         Matrix<double> NBB = B.Transpose().Multiply(PMatrix).Multipl
190.             y(B);
191.         Vector<double> w = B.Transpose().Multiply(PMatrix).Multiply(
192.             lVector);
193.         Vector<double> x = NBB.Inverse().Multiply(w);
194.         double newX1 = x1 + x[0] * 0.001;
195.         double newX2 = x2 + x[1] * 0.001;
196.         Vector<double> v = B.Multiply(x) - lVector;

```

```

195.
196.         double v6 = length[5] / (length[5] + length[6]) * v[5];
197.         double v7 = length[6] / (length[5] + length[6]) * v[5];
198.         double[] newv = new double[7];
199.         for (int i = 0; i < 5; i++)
200.         {
201.             newv[i] = v[i];
202.         }
203.         newv[5] = v6;
204.         newv[6] = v7;
205.
206.         for (int i = 0; i < L.Length; i++)
207.         {
208.             L[i] = height[i] + newv[i] * 0.001;
209.         }
210.         double newX3 = HB - L[6];
211.
212.         richTextBox1.Text += ("Matrix B:\n");
213.         richTextBox1.Text += (MatrixToString(B) + "\n");
214.         richTextBox1.Text += ("Matrix P:\n");
215.         richTextBox1.Text += (MatrixToString(PMatrix) + "\n");
216.         richTextBox1.Text += ("Matrix NBB:\n");
217.         richTextBox1.Text += (MatrixToString(NBB) + "\n");
218.         richTextBox1.Text += ("Vector w:\n");
219.         richTextBox1.Text += (VectorToString(w) + "\n");
220.         richTextBox1.Text += ("Vector x:\n");
221.         richTextBox1.Text += (VectorToString(x) + "\n");
222.         richTextBox1.Text += "\n改正
数: " + string.Join(", ", newv.Select(num => Math.Round(num, 3))) + '\n';
223.         richTextBox1.Text += "改正后高
程: " + string.Join(", ", L.Select(element => Math.Round(element, 3)));
224.         richTextBox1.Text += '\n' + "*****
**" + '\n';
225.         richTextBox1.Text += ("问题一:\n");
226.         richTextBox1.Text += "3号点高
程:" + newX1.ToString("F4") + 'm' + '\n' + "4号点高
程:" + newX2.ToString("F4") + 'm' + '\n' + "5号点高
程:" + newX3.ToString("F4") + 'm' + '\n';
227.         Matrix<double> vmatrix = Matrix<double>.Build.Dense(6, 1);
228.
229.         for (int i = 0; i < v.Count; i++)
230.         {
231.             vmatrix[i, 0] = v[i];
232.         }

```

```

233.         double result = (vmatrix.Transpose().Multiply(PMatrix).Multi
           ply(vmatrix))[0, 0];
234.         double seigema = +Math.Round(Math.Sqrt(result) / 2, 1);
235.         Matrix<double> Qxx = NBB.Inverse();
236.         Matrix<double> QLLgu = B.Multiply(NBB.Inverse()).Multiply(B.
           Transpose());
237.
238.         double segemac = seigema * Math.Sqrt(Qxx[0, 0]);
239.         double segemad = seigema * Math.Sqrt(Qxx[1, 1]);
240.         richTextBox1.Text += "*****" +
           '\n';
241.         richTextBox1.Text += "问题二:\n";
242.         richTextBox1.Text += "seigema:" + seigema + "mm" + '\n';
243.
244.         richTextBox1.Text += "CD 点高程中误差:\n";
245.         richTextBox1.Text += "C:" + Math.Round(segemac, 1).ToString(
           ) + "mm" + '\n';
246.         richTextBox1.Text += "D:" + Math.Round(segemad, 1).ToString(
           ) + "mm" + '\n';
247.         double segemah5 = seigema * Math.Sqrt(QLLgu[4, 4]);
248.         richTextBox1.Text += "*****" +
           '\n'; ;
249.         richTextBox1.Text += "问题三:\n";
250.         richTextBox1.Text += "h5 中误
           差:" + Math.Round(segemah5, 1).ToString() + "mm";
251.         tabControl1.SelectedTab = tabPage2;
252.     }
253.
254.     private void 保存当前报告
           ToolStripMenuItem_Click(object sender, EventArgs e)
255.     {
256.         SaveFileDialog saveFileDialog = new SaveFileDialog();
257.         saveFileDialog.Filter = "文本文件 (*.txt)|*.txt|所有文
           件 (*.*)|*.*";
258.
259.         if (saveFileDialog.ShowDialog() == DialogResult.OK)
260.         {
261.             string reportContent = richTextBox1.Text;
262.             File.WriteAllText(saveFileDialog.FileName, reportContent
           );
263.         }
264.     }
265.

```

```

266.         private void 退出
           ToolStripMenuItem_Click(object sender, EventArgs e)
267.         {
268.             Application.Exit();
269.         }
270.
271.         private void 帮助
           ToolStripMenuItem_Click(object sender, EventArgs e)
272.         {
273.             string helpText = "水准网平差及精度评定程序使用说明:\n" +
274.                               "1. 打开数据文件。-下方会显示已知点信息\n" +
275.                               "2. 查看已知点和路径信息。 \n" +
276.                               "3. 阅读题目描述。 \n" +
277.                               "4. 生成平差过程的矩阵。 \n" +
278.                               "5. 可查看水准网示意图。 \n" +
279.                               "6. 进行水准网平差计算。 \n" +
280.                               "7. 查看计算结果。 \n" +
281.                               "8. 保存报告。 \n" +
282.                               "9. 退出程序。 \n" +
283.                               "按顺序点击按钮即可!!\n" +
284.                               "\n 请注意，数据文件格式必须正确，否则程序将提
           示错误。";
285.
286.             MessageBox.Show(helpText, "帮助
           ", MessageBoxButtons.OK, MessageBoxIcon.Information);
287.         }
288.
289.         private void 清除数据
           ToolStripMenuItem_Click(object sender, EventArgs e)
290.         {
291.             Process.Start(Application.ExecutablePath);
292.             Application.Exit();
293.         }
294.
295.         private void 对比图像
           ToolStripMenuItem_Click(object sender, EventArgs e)
296.         {
297.             if (yuanheight == null || L == null)
298.             {
299.                 MessageBox.Show("请先进行平差计算!");
300.                 return;
301.             }
302.
303.             // 清除旧的数据点

```

```

304.         chart1.Series.Clear();
305.         chart1.ChartAreas.Clear();
306.
307.         // 创建新的 ChartArea
308.         ChartArea chartArea1 = new ChartArea("ChartArea1");
309.         chart1.ChartAreas.Add(chartArea1);
310.
311.         // 创建两个 Series
312.         Series series1 = new Series("Series1");
313.         series1.ChartArea = "ChartArea1";
314.         series1.ChartType = SeriesChartType.Line;
315.
316.         Series series2 = new Series("Series2");
317.         series2.ChartArea = "ChartArea1";
318.         series2.ChartType = SeriesChartType.Line;
319.         // 设置 Series 的颜色
320.         series1.Color = Color.Red;
321.         series2.Color = Color.Blue;
322.         // 添加 Series 到 Chart
323.         chart1.Series.Add(series1);
324.         chart1.Series.Add(series2);
325.
326.         // 添加数据点
327.         for (int i = 0; i < 6; i++)
328.         {
329.             series1.Points.AddXY(i + 1, yuanheight[i]);
330.             series2.Points.AddXY(i + 1, L[i]);
331.         }
332.
333.         // 设置 Y 轴的刻度和标题
334.         chartArea1.AxisY.Title = "整体趋势";
335.         chartArea1.AxisY2.Title = "微小差异";
336.         chartArea1.AxisY.TitleFont = new Font("微软雅黑", 12, FontStyle.Bold);
337.         chartArea1.AxisY2.TitleFont = new Font("微软雅黑", 12, FontStyle.Bold);
338.         // 使用 LINQ 获取最小值和最大值
339.         double minY = yuanheight.Min();
340.         double maxY = yuanheight.Max();
341.
342.         // 设置 Y 轴的显示范围
343.         chartArea1.AxisY.Minimum = minY - 1;
344.         chartArea1.AxisY.Maximum = maxY + 1;
345.         chartArea1.AxisY2.Minimum = minY - 0.01;

```

```

346.         chartArea1.AxisY2.Maximum = maxY + 0.01;
347.
348.         // 将第二个系列的Y轴设置为Y2轴
349.         series2.YAxisType = AxisType.Secondary;
350.
351.         // 更新图表
352.         chart1.Invalidate();
353.         tabControl1.SelectedTab = tabPage4;
354.     }
355.
356.     private void 缩小
        ToolStripMenuItem_Click(object sender, EventArgs e)
357.     {
358.         double width = chart1.ChartAreas[0].AxisX.Maximum - chart1.C
        hartAreas[0].AxisX.Minimum;
359.         double height = chart1.ChartAreas[0].AxisY.Maximum - chart1.
        ChartAreas[0].AxisY.Minimum;
360.         double x_cen = chart1.ChartAreas[0].AxisX.Maximum - width /
        2;
361.         double y_cen = chart1.ChartAreas[0].AxisY.Maximum - height /
        2;
362.         chart1.ChartAreas[0].AxisX.Maximum = x_cen + width / 2 * 1.2
        ;
363.         chart1.ChartAreas[0].AxisX.Minimum = x_cen - width / 2 * 1.2
        ;
364.         chart1.ChartAreas[0].AxisY.Maximum = y_cen + height / 2 * 1.
        2;
365.         chart1.ChartAreas[0].AxisY.Minimum = y_cen - height / 2 * 1.
        2;
366.     }
367.
368.     private void 放大
        ToolStripMenuItem_Click(object sender, EventArgs e)
369.     {
370.         double width = chart1.ChartAreas[0].AxisX.Maximum - chart1.C
        hartAreas[0].AxisX.Minimum;
371.         double height = chart1.ChartAreas[0].AxisY.Maximum - chart1.
        ChartAreas[0].AxisY.Minimum;
372.         double x_cen = chart1.ChartAreas[0].AxisX.Maximum - width /
        2;
373.         double y_cen = chart1.ChartAreas[0].AxisY.Maximum - height /
        2;
374.         chart1.ChartAreas[0].AxisX.Maximum = x_cen + width / 2 * 0.8
        ;

```



```

375.         chart1.ChartAreas[0].AxisX.Minimum = x_cen - width / 2 * 0.8
;
376.         chart1.ChartAreas[0].AxisY.Maximum = y_cen + height / 2 * 0.
8;
377.         chart1.ChartAreas[0].AxisY.Minimum = y_cen - height / 2 * 0.
8;
378.     }
379.
380.     private void 保存对比图像
        ToolStripMenuItem_Click(object sender, EventArgs e)
381.     {
382.         // 创建一个FileDialog 对象来选择保存位置
383.         SaveFileDialog saveDialog = new SaveFileDialog();
384.         saveDialog.Filter = "Bitmap Image (*.bmp)|*.bmp|JPEG Image (
        *.jpg)|*.jpg|PNG Image (*.png)|*.png";
385.         saveDialog.Title = "保存对比图像";
386.         saveDialog.ShowDialog();
387.
388.         if (saveDialog.FileName != "")
389.         {
390.             // 将图表转换为Bitmap 对象
391.             Bitmap bitmap = new Bitmap(chart1.Width, chart1.Height);
392.             Graphics g = Graphics.FromImage(bitmap);
393.             chart1.DrawToBitmap(bitmap, new Rectangle(0, 0, chart1.W
        idth, chart1.Height));
394.             g.Dispose();
395.
396.             // 创建 ImageFormat 对象
397.             ImageFormat format = ImageFormat.Jpeg;
398.
399.             // 保存Bitmap 对象到指定位置
400.             bitmap.Save(saveDialog.FileName, format);
401.         }
402.     }
403.
404.     private static string MatrixToString(Matrix<double> matrix)
405.     {
406.         string result = "";
407.         for (int i = 0; i < matrix.RowCount; i++)
408.         {
409.             for (int j = 0; j < matrix.ColumnCount; j++)
410.             {
411.                 result += Math.Round(matrix[i, j], 2).ToString() + "
";

```

```

412.         }
413.         result += Environment.NewLine;
414.     }
415.     return result;
416. }
417.
418.     private static string VectorToString(Vector<double> vector)
419.     {
420.         string result = "";
421.         for (int i = 0; i < vector.Count; i++)
422.         {
423.             result += Math.Round(vector[i], 2).ToString("F3") + " ";
424.         }
425.         return result;
426.     }
427. }
428. }

```

## 5.2 导线网

### Form1.cs

```

1. using System;
2. using System;
3. using System.Collections.Generic;
4. using System.Drawing;
5. using System.Drawing.Imaging;
6. using System.IO;
7. using System.Numerics;
8. using System.Text;
9. using System.Windows.Forms;
10. using MathNet.Numerics.LinearAlgebra;
11. using IronPython.Hosting;
12. using Microsoft.Scripting.Hosting;
13. using System.Diagnostics;
14.
15. namespace 导线网平差及精度评定程序
16. {
17.     public partial class Form1 : Form
18.     {
19.         List<Point> KnownPoints = new List<Point>();
20.         List<Angle> Angles = new List<Angle>();
21.         List<Edge> Edges = new List<Edge>();
22.         List<Point> UnknownPoints = new List<Point>(new Point[4]);

```

```

23.         List<AngleEquation> equations = new List<AngleEquation>();
24.         Algo Algo = new Algo();
25.         List<Point> updatedPoints = new List<Point>();
26.         double Rho = 206265;
27.         Matrix<Complex> Qxxquan;
28.         double seigemaquan;
29.         public Form1()
30.         {
31.             InitializeComponent();
32.         }
33.
34.
35.         private void 打开文件
ToolStripMenuItem_Click(object sender, EventArgs e)
36.         {
37.             OpenFileDialog openFileDialog = new OpenFileDialog();
38.             openFileDialog.Filter = "文本文件|*.txt";
39.             openFileDialog.Title = "选择文件";
40.
41.             if (openFileDialog.ShowDialog() == DialogResult.OK)
42.             {
43.                 try
44.                 {
45.                     string[] lines = System.IO.File.ReadAllLines(openFile
Dialog.FileName);
46.                     List<string[]> dataSet1 = new List<string[]>();
47.                     List<string[]> dataSet2 = new List<string[]>();
48.                     List<string[]> dataSet3 = new List<string[]>();
49.                     int dataSetIndex = 0;
50.
51.                     foreach (string line in lines)
52.                     {
53.                         // 跳过空行
54.                         if (string.IsNullOrEmpty(line))
55.                         {
56.                             dataSetIndex++;
57.                             continue;
58.                         }
59.
60.                         // 使用逗号分隔每一行
61.                         string[] values = line.Split(',');
62.
63.                         // 检查是否有足够的值
64.                         if (values.Length == 3)

```

```

65.         {
66.             // 根据 dataSetIndex 将数据添加到相应的数据集中
67.             switch (dataSetIndex)
68.             {
69.                 case 0:
70.                     dataSet1.Add(values);
71.                     break;
72.                 case 1:
73.                     dataSet2.Add(values);
74.                     break;
75.                 case 2:
76.                     dataSet3.Add(values);
77.                     break;
78.             }
79.         }
80.     }
81.
82.     // 假设每个数据集的大小相同
83.     int rowCount = Math.Max(dataSet1.Count, Math.Max(data
        Set2.Count, dataSet3.Count));
84.
85.     for (int i = 0; i < rowCount; i++)
86.     {
87.         // 从每个数据集中取出对应行的数据, 如果数据集不够大, 则
        使用空数组
88.         string[] row1 = i < dataSet1.Count ? dataSet1[i]
            : new string[3];
89.         string[] row2 = i < dataSet2.Count ? dataSet2[i]
            : new string[3];
90.         string[] row3 = i < dataSet3.Count ? dataSet3[i]
            : new string[3];
91.
92.         // 将数据添加到 dataGridView1 中
93.         dataGridView1.Rows.Add(row1[0], row1[1], row1[2],
            row2[0], row2[1], row2[2], row3[0], row3[1], row3[2]);
94.     }
95. }
96. catch (Exception ex)
97. {
98.     MessageBox.Show("读取文件时出错: " + ex.Message);
99. }
100. }
101. }
102.

```

```

103.         private void 初始化
           ToolStripMenuItem_Click(object sender, EventArgs e)
104.         {
105.
106.             // 添加已知点
107.             KnownPoints.Add(new Point(1, 121088.500, 259894.000));
108.             KnownPoints.Add(new Point(2, 127990.100, 255874.600));
109.
110.             // 角度列表
111.             string[] angleStrings = {
112.                 "72.10284",
113.                 "66.27289",
114.                 "88.58295",
115.                 "123.11341",
116.                 "85.13374",
117.                 "132.23352",
118.                 "79.09487",
119.                 "72.24564"
120.             };
121.
122.             // 处理角度
123.             for (int i = 0; i < angleStrings.Length; i++)
124.             {
125.                 Angle angle = Angle.Parse(angleStrings[i], i + 1);
126.                 Angles.Add(angle);
127.             }
128.
129.             String sbhjs = null;
130.             foreach (var point in KnownPoints)
131.             {
132.                 sbhjs += ("点号: {point.PointNumber}, 坐
           标: ({point.x}, {point.y})");
133.             }
134.             toolStripStatusLabel1.Text = sbhjs;
135.
136.             Edges.Add(new Edge(4, 6, 4451.417));
137.             Edges.Add(new Edge(2, 4, 5564.592));
138.             Edges.Add(new Edge(6, 5, 5569.269));
139.             StringBuilder sb = new StringBuilder();
140.
141.             // 输出已知点信息
142.             sb.AppendLine("已知点信息:");
143.             foreach (var point in KnownPoints)
144.             {

```

```

145.            sb.AppendLine($"点号: {point.PointNumber}, 坐
            标: ({point.x}, {point.y})");
146.        }
147.        sb.AppendLine();
148.
149.        // 输出角度信息
150.        sb.AppendLine("角度信息:");
151.        foreach (var angle in Angles)
152.        {
153.            double roundedRadians = Math.Round(angle.Radians, 3);
154.            sb.AppendLine($"角度序号: {angle.AngleNumber}, 弧
            度: {roundedRadians}");
155.        }
156.        sb.AppendLine();
157.
158.        // 输出边信息
159.        sb.AppendLine("边信息:");
160.        foreach (var edge in Edges)
161.        {
162.            sb.AppendLine(edge.Start + "-" + edge.End + ": " + edge.
            Length + "m");
163.        }
164.
165.        // 更新富文本框
166.        richTextBox1.Clear();
167.        richTextBox1.AppendText(sb.ToString());
168.        tabControl1.SelectedTab = tabPage2;
169.    }
170.
171.    private void 平差计算
        ToolStripMenuItem_Click(object sender, EventArgs e)
172.    {
173.        richTextBox1.Clear();
174.
175.        // 计算方位角
176.        #region 计算方位角
177.        List<Angle> CalculatedAzimuths = new List<Angle>();
178.        Angle chushifangweijiao = Algo.coordinateBackwardCalculation
            (KnownPoints[0], KnownPoints[1]);
179.        CalculatedAzimuths.Add(chushifangweijiao);
180.
181.        richTextBox1.AppendText("\n");
182.        richTextBox1.AppendText("*****\n
            ");

```

```

183.                richTextBox1.AppendText("\n");
184.
185.                richTextBox1.AppendText("1-2 方位角
为:" + Math.Round(chushifangweijiao.Radians, 3) + "\n");
186.                Angle fangweijiao24 = Algo.CalculateAzimuth(chushifangweijiao, Angles[1]);
187.                CalculatedAzimuths.Add(fangweijiao24);
188.                richTextBox1.AppendText("2-4 方位角
为:" + Math.Round(fangweijiao24.Radians, 3) + "\n");
189.                Angle fangweijiao43 = Algo.CalculateAzimuth(fangweijiao24, Angles[5]);
190.                CalculatedAzimuths.Add(fangweijiao43);
191.                richTextBox1.AppendText("4-3 方位角
为:" + Math.Round(fangweijiao43.Radians, 3) + "\n");
192.                Angle fangweijiao31 = Algo.CalculateAzimuth(fangweijiao43, Angles[2]);
193.                CalculatedAzimuths.Add(fangweijiao31);
194.                richTextBox1.AppendText("3-1 方位角
为:" + Math.Round(fangweijiao31.Radians, 3) + "\n");
195.                Angle fangweijiao46 = Algo.CalculateAzimuth(fangweijiao43, Angles[4]);
196.                CalculatedAzimuths.Add(fangweijiao46);
197.                richTextBox1.AppendText("4-6 方位角
为:" + Math.Round(fangweijiao46.Radians, 3) + "\n");
198.                Angle fangweijiao65 = Algo.CalculateAzimuth(fangweijiao46, Angles[7]);
199.                CalculatedAzimuths.Add(fangweijiao65);
200.                richTextBox1.AppendText("6-5 方位角
为:" + Math.Round(fangweijiao65.Radians, 3) + "\n");
201.                Angle fangweijiao53 = Algo.CalculateAzimuth(fangweijiao65, Angles[6]);
202.                CalculatedAzimuths.Add(fangweijiao53);
203.                richTextBox1.AppendText("5-3 方位角
为:" + Math.Round(fangweijiao53.Radians, 3) + "\n");
204.
205.                richTextBox1.AppendText("\n");
206.                richTextBox1.AppendText("*****\n");
207.                richTextBox1.AppendText("\n");
208.                #endregion
209.
210.                // 计算未知边长
211.                #region 计算未知边长
212.                Edge edge12 = new Edge(KnownPoints[0], KnownPoints[1]);

```

```

213.            richTextBox1.AppendText("1-2 长度
为:" + Math.Round(edge12.Length, 3) + "(m)\n");
214.            Edge edge14 = Edge.CosineLaw(edge12.Length, Edges[1].Length,
Angles[1].Radians);
215.            richTextBox1.AppendText("1-4 长度
为:" + Math.Round(edge14.Length, 3) + "(m)\n");
216.            Edge edge45 = Edge.CosineLaw(Edges[0].Length, Edges[2].Length,
Angles[7].Radians);
217.            richTextBox1.AppendText("4-5 长度
为:" + Math.Round(edge45.Length, 3) + "(m)\n");
218.            Angle angle6you = new Angle();
219.            angle6you.Radians = Math.Asin(edge12.Length * Math.Sin(Angles[1].Radians) / edge14.Length);
220.            Angle angle6zuo = new Angle();
221.            angle6zuo.Radians = Angles[5].Radians - angle6you.Radians;
222.            Edge edge13 = Edge.SineLaw(edge14.Length, Angles[2].Radians,
angle6zuo.Radians);
223.            richTextBox1.AppendText("1-3 长度
为:" + Math.Round(edge13.Length, 3) + "(m)\n");
224.            Angle angle1you = new Angle();
225.            angle1you.Radians = Math.Asin(Edges[1].Length * Math.Sin(Angles[1].Radians) / edge14.Length);
226.            Angle angle1zuo = new Angle();
227.            angle1zuo.Radians = Angles[0].Radians - angle1you.Radians;
228.            Edge edge34 = Edge.SineLaw(edge14.Length, Angles[2].Radians,
angle1zuo.Radians);
229.            richTextBox1.AppendText("3-4 长度
为:" + Math.Round(edge34.Length, 3) + "(m)\n");
230.            Angle angle5you = new Angle();
231.            angle5you.Radians = Math.Asin(Edges[2].Length * Math.Sin(Angles[7].Radians) / edge45.Length);
232.            Angle angle5zuo = new Angle();
233.            angle5zuo.Radians = Angles[4].Radians - angle5you.Radians;
234.            Edge edge35 = Edge.SineLaw(edge45.Length, Angles[3].Radians,
angle5zuo.Radians);
235.            richTextBox1.AppendText("3-5 长度
为:" + Math.Round(edge35.Length, 3) + "(m)\n");
236.            #endregion
237.
238.            // 计算坐标初值
239.            #region 计算未知点坐标
240.            UnknownPoints[1] = Algo.CoordinateForwardCalculation(KnownPoints[1], Edges[1].Length, fangweijiao24.Radians);

```



```

241.         UnknownPoints[0] = Algo.CoordinateForwardCalculation(Unknown
           Points[1], edge34.Length, fangweijiao43.Radians);
242.         UnknownPoints[3] = Algo.CoordinateForwardCalculation(Unknown
           Points[1], Edges[0].Length, fangweijiao46.Radians);
243.         UnknownPoints[2] = Algo.CoordinateForwardCalculation(Unknown
           Points[3], Edges[2].Length, fangweijiao65.Radians);
244.
245.         richTextBox1.AppendText("\n");
246.         richTextBox1.AppendText("*****\n
           ");
247.         richTextBox1.AppendText("\n");
248.
249.         richTextBox1.AppendText("点 3 坐标初始
           值:" + '(' + Math.Round(UnknownPoints[0].x, 3) + ',' + Math.Round(Unknown
           Points[0].y, 3) + ')' + '\n');
250.         richTextBox1.AppendText("点 4 坐标初始
           值:" + '(' + Math.Round(UnknownPoints[1].x, 3) + ',' + Math.Round(Unknown
           Points[1].y, 3) + ')' + '\n');
251.         richTextBox1.AppendText("点 5 坐标初始
           值:" + '(' + Math.Round(UnknownPoints[2].x, 3) + ',' + Math.Round(Unknown
           Points[2].y, 3) + ')' + '\n');
252.         richTextBox1.AppendText("点 6 坐标初始
           值:" + '(' + Math.Round(UnknownPoints[3].x, 3) + ',' + Math.Round(Unknown
           Points[3].y, 3) + ')' + '\n');
253.         #endregion
254.
255.         #region 计算角度方程
256.         AngleEquation aequation1 = Algo.CreateAdjustmentEquation(
257.             KnownPoints[0], // 点j
258.             UnknownPoints[0], // 点k
259.             KnownPoints[1], // 点h
260.             Angles[0],
261.             fangweijiao31,
262.             chushifangweijiao,
263.             Rho
264.         );
265.         aequation1.Constant += 2 * Math.PI;
266.         equations.Add(aequation1);
267.
268.         AngleEquation aequation2 = Algo.CreateAdjustmentEquation(
269.             KnownPoints[1], // 点j
270.             KnownPoints[0], // 点k
271.             UnknownPoints[1], // 点h
272.             Angles[1],

```

```

273.         chushifangweijiao,
274.         fangweijiao24,
275.         Rho
276.     );
277.     equations.Add(aequation2);
278.
279.     AngleEquation aequation3 = Algo.CreateAdjustmentEquation(
280.         UnknownPoints[0], // 点j
281.         UnknownPoints[1], // 点k
282.         KnownPoints[0], // 点h
283.         Angles[2],
284.         fangweijiao43,
285.         fangweijiao31,
286.         Rho
287.     );
288.     equations.Add(aequation3);
289.
290.     AngleEquation aequation4 = Algo.CreateAdjustmentEquation(
291.         UnknownPoints[0], // 点j
292.         UnknownPoints[1], // 点k
293.         KnownPoints[0], // 点h
294.         Angles[2],
295.         fangweijiao43,
296.         fangweijiao31,
297.         Rho
298.     );
299.
300.     aequation4.Constant += 2 * Math.PI;
301.     equations.Add(aequation4);
302.
303.     AngleEquation aequation5 = Algo.CreateAdjustmentEquation(
304.         UnknownPoints[1], // 点j
305.         UnknownPoints[0], // 点k
306.         UnknownPoints[3], // 点h
307.         Angles[4],
308.         fangweijiao46,
309.         fangweijiao43,
310.         Rho
311.     );
312.     equations.Add(aequation5);
313.
314.     AngleEquation aequation6 = Algo.CreateAdjustmentEquation(
315.         UnknownPoints[1], // 点j
316.         KnownPoints[1], // 点k

```

```

317.             UnknownPoints[0], // 点h
318.             Angles[5],
319.             fangweijiao24,
320.             fangweijiao43,
321.             Rho
322.         );
323.         equations.Add(aequation6);
324.
325.         AngleEquation aequation7 = Algo.CreateAdjustmentEquation(
326.             UnknownPoints[1], // 点j
327.             UnknownPoints[0], // 点k
328.             UnknownPoints[3], // 点h
329.             Angles[4],
330.             fangweijiao53,
331.             fangweijiao65,
332.             Rho
333.         );
334.         equations.Add(aequation7);
335.
336.         AngleEquation aequation8 = Algo.CreateAdjustmentEquation(
337.             UnknownPoints[3], // 点j
338.             UnknownPoints[2], // 点k
339.             UnknownPoints[1], // 点h
340.             Angles[7],
341.             fangweijiao65,
342.             fangweijiao46,
343.             Rho
344.         );
345.         equations.Add(aequation8);
346.         EquationFormatter equationFormatter = new EquationFormatter(
347.             );
348.
349.         tabControl1.SelectedTab = tabPage2;
350.         string jiaoequations = equationFormatter.GetEquationString()
351.         ;
352.         richTextBox1.Text += jiaoequations;
353.     }
354.
355.     private void 结果
356.         toolStripButton4_Click(object sender, EventArgs e)
357.         {
358.             // 创建矩阵 B (11x8)

```

```

358.         var B = Matrix<Complex>.Build.DenseOfArray(new Complex[,]
359.         {
360.             { -0.276, 0, 0, 0.048, 0, 0, 0, 0 }, // v1
361.             { 0, 0.219, -0.563, 0, 0, 0, 0, 0 }, // v2
362.             { 0.284, 0, -0.008, 0.515, 0, -0.563, 0, 0 }, // v3
363.             { 0.391, 0, 0, 1.274, 0, 0.008, -0.399, 0.712 }, // v4
364.             { 0.008, -0.453, 0, 0.563, -0.434, 0, 0.445, -0.129 }, // v5
365.             { -0.008, -0.211, 0, -0.563, 0.862, 0, 0, 0 }, // v6
366.             { -0.399, 0, 0.604, 0.712, 0, -0.206, -0.206, -0.308 }, // v
367.             { 0, 0.445, -0.206, -0.129, -0.239, 0, 0.437, 0 }, // v8
368.             { 0, -0.872, 0, 0, -0.591, 0, 0, 0 }, // v9
369.             { 0, 0.278, 0, 0, 0.961, -0.278, 0, -0.961 }, // v10
370.             { 0, 0, -0.832, 0, 0, 0.555, 0.832, -0.555 } // v11
371.         });
372.
373.         var l = MathNet.Numerics.LinearAlgebra.Vector<Complex>.Build
374.         .Dense(new Complex[]
375.         {
376.             -1.47, 0.07, 3.4, -3.4, 0, 0, 0, 0, 0, 0, 0
377.         });
378.         // 创建对角矩阵 P (11x11)
379.         var P = Matrix<Complex>.Build.DiagonalOfDiagonalVector(MathN
380.         et.Numerics.LinearAlgebra.Vector<Complex>.Build.Dense(new Complex[]
381.         {
382.             1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2
383.         }));
384.         // 计算 NBB = B^T * P * B
385.         var NBB = B.ConjugateTranspose() * P * B;
386.         var w = B.ConjugateTranspose() * P * l;
387.         var x = NBB.ConjugateTranspose() * w;
388.         var Qxx = NBB.ConjugateTranspose();
389.         Qxxquan = Qxx;
390.         var v = B * x - l;
391.         var vmatrix = Matrix<Complex>.Build.Dense(v.Count, 1);
392.
393.         for (int i = 0; i < v.Count; i++)
394.         {
395.             vmatrix[i, 0] = v[i];
396.         }
397.         var resultMatrix = vmatrix.ConjugateTranspose() * P * vmatri
x;

```

```

398.         var result = resultMatrix[0, 0].Real; // 只取实部
399.         var seigema = Math.Round(Math.Sqrt(result) / 3, 1);
400.         seigemaquan = seigema;
401.         var Dxx = seigema * seigema * Qxx;
402.
403.         string NBBOutput = GetFormattedMatrixString(NBB, "NBB");
404.         string wOutput = GetFormattedVectorString(w, "w");
405.         string xOutput = GetFormattedVectorString(x, "x");
406.         string QxxOutput = GetFormattedMatrixString(Qxx, "Qxx");
407.         string vOutput = GetFormattedVectorString(v, "v");
408.         string seigemaOutput = GetFormattedScalarString(seigema, "se
igema");
409.         string DxxOutput = GetFormattedMatrixString(Dxx, "Dxx");
410.
411.         richTextBox1.Clear();
412.         richTextBox1.AppendText(NBBOutput);
413.         richTextBox1.AppendText(wOutput);
414.         richTextBox1.AppendText(xOutput);
415.         richTextBox1.AppendText(QxxOutput);
416.         richTextBox1.AppendText(vOutput);
417.         richTextBox1.AppendText(seigemaOutput);
418.         richTextBox1.AppendText(DxxOutput);
419.         StringBuilder sb = new StringBuilder();
420.         sb.AppendLine("更新后的点坐标: ");
421.
422.         for (int i = 0; i < UnknownPoints.Count; i++)
423.         {
424.             double originalX = UnknownPoints[i].x;
425.             double originalY = UnknownPoints[i].y;
426.
427.             // 获取 v 中对应位置的数值, 并乘以 0.001
428.             double vValue = v[i].Real * 0.01;
429.             double vValue2 = v[i+1].Real * 0.01;
430.
431.             // 更新点的坐标
432.             double newX = Math.Round(originalX + vValue, 3);
433.             double newY = Math.Round(originalY + vValue2, 3);
434.
435.             // 创建新的点, 并将其添加到 updatedPoints 列表中
436.             updatedPoints.Add(new Point(newX, newY));
437.             sb.AppendFormat("
点 {0}: ({1}, {2})\n", i + 3, updatedPoints[i].x, updatedPoints[i].y);
438.         }
439.

```

```

440.
441.
442.         richTextBox1.AppendText(sb.ToString());
443.         tabControl1.SelectedTab = tabPage2;
444.     }
445.
446.     private string GetFormattedMatrixString(Matrix<Complex> matrix,
        string matrixName)
447.     {
448.         StringBuilder sb = new StringBuilder();
449.         sb.AppendLine($"矩阵 {matrixName}:");
450.
451.         for (int i = 0; i < matrix.RowCount; i++)
452.         {
453.             for (int j = 0; j < matrix.ColumnCount; j++)
454.             {
455.                 double roundedValue = Math.Round(matrix[i, j].Real,
                    3);
456.                 sb.Append(roundedValue.ToString("0.000") + " ");
457.             }
458.             sb.AppendLine();
459.         }
460.
461.         sb.AppendLine(); // 添加空行作为分隔
462.         return sb.ToString();
463.     }
464.
465.     private string GetFormattedVectorString(MathNet.Numerics.LinearAlgebra.Vector<Complex> vector, string vectorName)
466.     {
467.         StringBuilder sb = new StringBuilder();
468.         sb.AppendLine($"向量 {vectorName}:");
469.
470.         for (int i = 0; i < vector.Count; i++)
471.         {
472.             double roundedValue = Math.Round(vector[i].Real, 3);
473.             sb.Append(roundedValue.ToString("0.000") + " ");
474.         }
475.
476.         sb.AppendLine(); // 添加换行符
477.         sb.AppendLine(); // 添加空行作为分隔
478.         return sb.ToString();
479.     }
480.

```

```

481.         private string GetFormattedScalarString(double scalar, string scalarName)
482.         {
483.             return $"标量 {scalarName}: {Math.Round(scalar, 3)}\n\n";
484.         }
485.
486.         private void 清除
            ToolStripMenuItem_Click(object sender, EventArgs e)
487.         {
488.             richTextBox1.Clear();
489.         }
490.
491.         private void 新建文件
            ToolStripMenuItem_Click(object sender, EventArgs e)
492.         {
493.             // 创建一个 SaveFileDialog 对象
494.             SaveFileDialog sfd = new SaveFileDialog();
495.
496.             // 设置对话框的标题和文件扩展名过滤器
497.             sfd.Title = "选择文件路径";
498.             sfd.Filter = "文本文件 (*.txt)|*.txt";
499.
500.             // 显示对话框并获取用户的选择
501.             if (sfd.ShowDialog() == DialogResult.OK)
502.             {
503.                 // 获取用户选择的文件路径和文件名
504.                 string filePath = sfd.FileName;
505.
506.                 // 创建文件
507.                 using (StreamWriter sw = new StreamWriter(filePath))
508.                 {
509.                     // 写入指定格式的内容
510.                     sw.WriteLine("格式:");
511.                     sw.WriteLine("点号,x,y");
512.                     sw.WriteLine("端点 1,端点 2,角度");
513.                     sw.WriteLine("端点 1,端点 2,长度");
514.                 }
515.
516.                 // 提示用户文件已创建
517.                 MessageBox.Show("文件已创建。");
518.             }
519.         }
520.

```

```

521.         private void 帮助
           ToolStripMenuItem_Click(object sender, EventArgs e)
522.         {
523.             // 假设 richTextBox1 是您的富文本框控件
524.             StringBuilder sb = new StringBuilder();
525.
526.             // 添加帮助文档内容
527.             sb.AppendLine("欢迎使用我们的导线网平差及精度评定程序!");
528.             sb.AppendLine();
529.             sb.AppendLine("***打开文件***");
530.             sb.AppendLine("点击菜单栏上的 '文件' 选项, 然后选择 '打开'。");
531.             sb.AppendLine("您可以选择一个包含导线网数据的文件, 例如 '导线网数
           据.txt'。");
532.             sb.AppendLine();
533.             sb.AppendLine("***初始化***");
534.             sb.AppendLine("在打开文件后, 您需要点击 '初始化' 按钮来准备数据。
           ");
535.             sb.AppendLine("初始化步骤将读取文件中的数据并将其加载到程序中。
           ");
536.             sb.AppendLine();
537.             sb.AppendLine("***平差计算***");
538.             sb.AppendLine("初始化完成后, 您需要点击 '平差计算' 按钮来开始计算。
           ");
539.             sb.AppendLine("平差计算将使用您提供的数据来计算导线网的精度。");
540.             sb.AppendLine();
541.             sb.AppendLine("***结果***");
542.             sb.AppendLine("计算完成后, 您可以在界面上直接查看结果。");
543.             sb.AppendLine("您还可以点击 '保存报告' 或 '保存图片' 按钮来保存
           结果。");
544.             sb.AppendLine();
545.             sb.AppendLine("***新建文件***");
546.             sb.AppendLine("点击菜单栏上的 '文件' 选项, 然后选择 '新建文件'。
           ");
547.             sb.AppendLine("这将创建一个新的文件, 并显示数据格式。");
548.             sb.AppendLine();
549.             sb.AppendLine("***图像操作***");
550.             sb.AppendLine("点击菜单栏上的 '图像' 选项, 然后选择 '生成误差椭
           圆'。");
551.             sb.AppendLine("这将使用计算结果生成误差椭圆的图像。");
552.
553.             // 使用 MessageBox 显示帮助文档内容, 并设置字体大小
554.             MessageBox.Show(sb.ToString(), "帮助文档
           ", MessageBoxButtons.OK, MessageBoxIcon.Information, MessageBoxDefaultBut

```



```

        ton.Button1, MessageBoxOptions.DefaultDesktopOnly | MessageBoxOptions.Rig
        htAlign);
555.    }
556.
557.    private void 数据
        ToolStripMenuItem_Click(object sender, EventArgs e)
558.    {
559.        tabControl1.SelectedTab = tabPage1;
560.    }
561.
562.    private void 退出
        ToolStripMenuItem_Click(object sender, EventArgs e)
563.    {
564.        Application.Exit();
565.    }
566.
567.    private void 保存当前报告
        ToolStripMenuItem_Click(object sender, EventArgs e)
568.    {
569.        // 创建一个 SaveFileDialog 对象
570.        SaveFileDialog sfd = new SaveFileDialog();
571.
572.        // 设置对话框的标题和文件扩展名过滤器
573.        sfd.Title = "选择文件路径";
574.        sfd.Filter = "文本文件 (*.txt)|*.txt";
575.
576.        // 显示对话框并获取用户的选择
577.        if (sfd.ShowDialog() == DialogResult.OK)
578.        {
579.            // 获取用户选择的文件路径和文件名
580.            string filePath = sfd.FileName;
581.
582.            // 创建文件并写入富文本框内容
583.            using (StreamWriter sw = new StreamWriter(filePath))
584.            {
585.                sw.Write(richTextBox1.Text);
586.            }
587.
588.            // 提示用户文件已保存
589.            MessageBox.Show("文件已保存。");
590.        }
591.    }
592.

```

```

593.         private void 保存图像
           ToolStripMenuItem_Click(object sender, EventArgs e)
594.         {
595.             SaveFileDialog saveDialog = new SaveFileDialog();
596.             saveDialog.Filter = "Bitmap Image (*.bmp)|*.bmp|JPEG Image (
               *.jpg)|*.jpg|PNG Image (*.png)|*.png";
597.             saveDialog.Title = "保存误差椭圆";
598.             saveDialog.ShowDialog();
599.
600.             if (saveDialog.FileName != "")
601.             {
602.                 Bitmap bitmap = new Bitmap(pictureBox1.Width, pictureBox
                   1.Height);
603.                 pictureBox1.DrawToBitmap(bitmap, pictureBox1.ClientRecta
                   ngle);
604.
605.                 string fileName = saveDialog.FileName;
606.                 ImageFormat format = ImageFormat.Jpeg;
607.                 if (fileName.EndsWith(".bmp", StringComparison.OrdinalIg
                   noreCase))
608.                 {
609.                     format = ImageFormat.Bmp;
610.                 }
611.                 else if (fileName.EndsWith(".png", StringComparison.Ordi
                   nalIgnoreCase))
612.                 {
613.                     format = ImageFormat.Png;
614.                 }
615.
616.                 bitmap.Save(saveDialog.FileName, format);
617.             }
618.         }
619.
620.         private void 误差椭圆
           ToolStripMenuItem_Click(object sender, EventArgs e)
621.         {
622.             pictureBox2.Image = Properties.Resources.yuan;
623.             tabControl1.SelectedTab = tabPage3;
624.         }
625.
626.         private void DrawEllipsesOnPictureBox(Ellipse ellipse1, Ellipse
           ellipse2, PictureBox pictureBox)
627.         {
628.             // 创建Bitmap 对象, 用于缓存绘制内容

```

```

629.         Bitmap bmp = new Bitmap(pictureBox.Width, pictureBox.Height)
        ;
630.
631.         // 获取 Graphics 对象, 用于在 Bitmap 上绘制
632.         Graphics g = Graphics.FromImage(bmp);
633.
634.         // 清除 pictureBox 背景
635.         g.Clear(pictureBox.BackColor);
636.
637.         // 定义画笔
638.         Pen pen1 = new Pen(Color.Red, 2); // 第一个椭圆的颜色
639.         Pen pen2 = new Pen(Color.Blue, 2); // 第二个椭圆的颜色
640.
641.         // 转换椭圆参数以适应 pictureBox 坐标系
642.         float centerX1 = (float)ellipse1.center.x;
643.         float centerY1 = (float)ellipse1.center.y;
644.         float semiMajorAxis1 = (float)ellipse1.E;
645.         float semiMinorAxis1 = (float)ellipse1.F;
646.
647.         float centerX2 = (float)ellipse2.center.x;
648.         float centerY2 = (float)ellipse2.center.y;
649.         float semiMajorAxis2 = (float)ellipse2.E;
650.         float semiMinorAxis2 = (float)ellipse2.F;
651.
652.         // 绘制第一个椭圆
653.         g.DrawEllipse(pen1, centerX1 - semiMajorAxis1, centerY1 - se
            miMinorAxis1, 2 * semiMajorAxis1, 2 * semiMinorAxis1);
654.
655.         // 绘制第二个椭圆
656.         g.DrawEllipse(pen2, centerX2 - semiMajorAxis2, centerY2 - se
            miMinorAxis2, 2 * semiMajorAxis2, 2 * semiMinorAxis2);
657.
658.         // 将绘制的内容显示在 pictureBox 上
659.         pictureBox.Image = bmp;
660.
661.         // 释放资源
662.         pen1.Dispose();
663.         pen2.Dispose();
664.         g.Dispose();
665.     }
666.
667.     private void 放大
        ToolStripMenuItem_Click(object sender, EventArgs e)
668.     {

```

```

669.         if (pictureBox2.Image != null)
670.         {
671.             pictureBox2.SizeMode = PictureBoxSizeMode.Zoom;
672.         }
673.     }
674.
675.     private void 缩小
        ToolStripMenuItem_Click(object sender, EventArgs e)
676.     {
677.         if (pictureBox2.Image != null)
678.         {
679.             pictureBox2.SizeMode = PictureBoxSizeMode.Normal;
680.         }
681.     }
682.
683.     private void 路径
        ToolStripMenuItem_Click(object sender, EventArgs e)
684.     {
685.         tabControl1.SelectedTab = tabPage4;
686.     }
687. }
688. }

```

## Algo.cs

```

1. using System;
2. using System;
3.
4. namespace 导线网平差及精度评定程序
5. {
6.     public class Algo
7.     {
8.
9.         //坐标正算
10.        public static Point CoordinateForwardCalculation(Point pointA, double distance, double azimuthRadians)
11.        {
12.            double x2 = pointA.x + distance * Math.Sin(azimuthRadians);
13.            double y2 = pointA.y + distance * Math.Cos(azimuthRadians);
14.            return new Point(0, x2, y2); // 这里假设点B的序号为0,可以根据实际情况调整
15.        }
16.
17.        //坐标反算计算初始方位角
18.        public Angle coordinateBackwardCalculation(Point p1, Point p2)

```

```

19.      {
20.          // 计算两点之间的差值
21.          double deltaX = p2.x - p1.x;
22.          double deltaY = p2.y - p1.y;
23.
24.          // 计算弧度
25.          double radian = Math.Atan2(deltaY, deltaX);
26.          Angle angle = new Angle();
27.          // 将弧度转换为度
28.          double degree = Angle.DegreesToRadians(radian);
29.
30.          // 根据象限调整角度
31.          if (deltaX >= 0 && deltaY >= 0)
32.          {
33.          }
34.          else if (deltaX < 0 && deltaY >= 0)
35.          {
36.              // 第二象限, 角度加上 180°
37.              degree = Math.PI - degree;
38.          }
39.          else if (deltaX < 0 && deltaY < 0)
40.          {
41.              // 第三象限, 角度加上 180°
42.              degree += Math.PI;
43.          }
44.          else if (deltaX >= 0 && deltaY < 0)
45.          {
46.              // 第四象限, 角度加上 360°
47.              degree = 2 * Math.PI - degree;
48.          }
49.          degree %= 2 * Math.PI;
50.          // 将度数存储到Angle 实例中
51.          angle.Radians = degree;
52.
53.
54.          // 返回Angle 实例
55.          return angle;
56.      }
57.
58.      // 计算其他方位角
59.      public static Angle CalculateAzimuth(Angle angleRadians, Angle angleDifferenceRadians)
60.      {
61.          Angle azimuthRadians = new Angle();

```

```

62.         azimuthRadians.Radians = angleRadians.Radians - Math.PI + angleDifferenceRadians.Radians;
63.         while (azimuthRadians.Radians < 0)
64.         {
65.             azimuthRadians.Radians += 2 * Math.PI;
66.         }
67.         while (azimuthRadians.Radians > 2 * Math.PI)
68.         {
69.             azimuthRadians.Radians -= 2 * Math.PI;
70.         }
71.         return azimuthRadians;
72.     }
73.
74.     // 计算角度 l
75.     public double CalculateLi(Angle alpha, Angle beta, Angle gamma)
76.     {
77.         double li = alpha.Radians -( beta.Radians - gamma.Radians);
78.         return li;
79.     }
80.
81.
82.     // 计算角度方程系数创建方程类
83.     public AngleEquation CreateAdjustmentEquation(Point j, Point k, Point h, Angle alpha, Angle beta, Angle gamma, double rhoDoublePrime)
84.     {
85.         AngleEquation equation = new AngleEquation();
86.
87.         // 计算坐标差
88.         double deltaXjk0 = k.x - j.x;
89.         double deltaYjk0 = k.y - j.y;
90.         double deltaXjh0 = h.x - j.x;
91.         double deltaYjh0 = h.y - j.y;
92.
93.         // 计算边长
94.         double sjk0 = Math.Sqrt(deltaXjk0 * deltaXjk0 + deltaYjk0 * deltaYjk0);
95.         double sjh0 = Math.Sqrt(deltaXjh0 * deltaXjh0 + deltaYjh0 * deltaYjh0);
96.
97.         // 计算系数
98.         equation.CoefficientXj = rhoDoublePrime * (deltaYjk0 / (sjk0 * sjk0) - deltaYjh0 / (sjh0 * sjh0));
99.         equation.CoefficientYj = -rhoDoublePrime * (deltaXjk0 / (sjk0 * sjk0) - deltaXjh0 / (sjh0 * sjh0));

```

```

100.         equation.CoefficientXk = -rhoDoublePrime * deltaYjk0 / (sjk0
        * sjk0);
101.         equation.CoefficientYk = rhoDoublePrime * deltaXjk0 / (sjk0
        * sjk0);
102.         equation.CoefficientXh = rhoDoublePrime * deltaYjh0 / (sjh0
        * sjh0);
103.         equation.CoefficientYh = -rhoDoublePrime * deltaXjh0 / (sjh0
        * sjh0);
104.
105.         // 计算常数项 li
106.         double li = CalculateLi(alpha, beta, gamma);
107.         equation.Constant = li;
108.
109.         return equation;
110.     }
111.
112.
113.     public Ellipse CalculateEllipse(double Qxx, double Qyy, double Q
        xy, double seigema, Point point)
114.     {
115.         double E, F, K, QEE, QFF;
116.         double PhiE;
117.
118.         // 计算 K
119.         K = Math.Sqrt((Qxx - Qyy) * (Qxx - Qyy) + 4 * Qxy * Qxy);
120.
121.         // 计算 QEE 和 QFF
122.         QEE = 0.5 * (Qxx + Qyy + K);
123.         QFF = 0.5 * (Qxx + Qyy - K);
124.
125.         // 计算 tan(φE)
126.         PhiE = Math.Atan((QEE - QFF) / Qxy);
127.
128.         // 创建 Angle 对象
129.         Angle angle = new Angle();
130.         angle.Radians = PhiE;
131.         // 计算 E 和 F
132.         E = seigema * Math.Abs(QEE);
133.         F = seigema * Math.Abs(QFF);
134.
135.         // 创建 Ellipse 对象
136.         Ellipse ellipse = new Ellipse(E, F, angle, point);
137.
138.         return ellipse;

```

```

139.         }
140.
141.     }
142. }

```

## Angle.cs

```

1. using System;
2. using System;
3.
4. namespace 导线网平差及精度评定程序
5. {
6.     public class Angle
7.     {
8.         public int AngleNumber { get; set; }
9.
10.        public int Degrees { get; set; }
11.
12.        public int Minutes { get; set; }
13.
14.        public double Seconds { get; set; }
15.
16.        public double Radians { get; set; }
17.
18.        // 构造函数, 接受度分秒格式
19.        public Angle(int angleNumber, int degrees, int minutes, double seconds)
20.        {
21.            AngleNumber = angleNumber;
22.            Degrees = degrees;
23.            Minutes = minutes;
24.            Seconds = seconds;
25.            Radians = ToRadians();
26.        }
27.
28.        // 构造函数, 传入弧度
29.        public Angle(int angleNumber, double radians)
30.        {
31.            AngleNumber = angleNumber;
32.            Radians = radians;
33.            RadiansToDegrees(radians);
34.        }
35.
36.        public Angle()
37.        {

```



```

38.     }
39.
40.     // 将以弧度为单位的浮点数转换为度分秒
41.     private void RadiansToDegrees(double radians)
42.     {
43.         double degrees = radians * (180.0 / Math.PI);
44.         FromDecimalDegrees(degrees);
45.     }
46.     public static double DegreesToRadians(double degrees)
47.     {
48.         return degrees * Math.PI / 180.0;
49.     }
50.
51.     // 将以度为单位的浮点数转换为度分秒
52.     private void FromDecimalDegrees(double decimalDegrees)
53.     {
54.         Degrees = (int)decimalDegrees;
55.         double minutesPart = (decimalDegrees - Degrees) * 60;
56.         Minutes = (int)minutesPart;
57.         Seconds = (minutesPart - Minutes) * 60;
58.     }
59.
60.     public static Angle Parse(string angleString, int angleNumber)
61.     {
62.         double decimalDegrees = double.Parse(angleString);
63.         int degrees = (int)decimalDegrees;
64.         double minutesPart = (decimalDegrees - degrees) * 100;
65.         int minutes = (int)minutesPart;
66.         double seconds = (minutesPart - minutes) * 100;
67.
68.         return new Angle(angleNumber, degrees, minutes, seconds);
69.     }
70.
71.
72.     // 实现弧度加法
73.     public static Angle operator +(Angle a1, Angle a2)
74.     {
75.         return new Angle(0, a1.Radians + a2.Radians);
76.     }
77.
78.     // 实现弧度减法
79.     public static Angle operator -(Angle a1, Angle a2)
80.     {
81.         return new Angle(0, a1.Radians - a2.Radians);

```

```

82.     }
83.
84.     // 将度分秒转换为弧度
85.     private double ToRadians()
86.     {
87.         return (Degrees + Minutes / 60.0 + Seconds / 3600.0) * Math.P
            I / 180.0;
88.     }
89. }
90.}

```

## AngleEquation.cs

```

1. using System;
2. using System;
3.
4. namespace 导线网平差及精度评定程序
5. {
6.     public class AngleEquation
7.     {
8.         public double CoefficientXj { get; set; }
9.         public double CoefficientYj { get; set; }
10.        public double CoefficientXk { get; set; }
11.        public double CoefficientYk { get; set; }
12.        public double CoefficientXh { get; set; }
13.        public double CoefficientYh { get; set; }
14.
15.        // 常数项
16.        public double Constant { get; set; }
17.
18.        // 构造函数
19.        public AngleEquation()
20.        {
21.            // 初始化系数和常数项
22.        }
23.
24.        // 计算方程的值
25.        public string CalculateValue()
26.        {
27.            string expression = string.Format("{0}{1} * j.x {2}{3} * j.y
                {4}{5} * k.x {6}{7} * k.y {8}{9} * h.x {10}{11} * h.y {12}{13}",
28.                CoefficientXj >= 0 ? "+" : "", Math.Round(Math.Abs(Coeffi
                cientXj), 3), // j.x 的符号和数值
29.                CoefficientYj >= 0 ? "+" : "-", Math.Round(Math.Abs(Coeffi
                cientYj), 3), // j.y 的符号和数值

```

```

30.         CoefficientXk >= 0 ? "+" : "-", Math.Round(Math.Abs(CoefficientXk), 3), // k.x 的符号和数值
31.         CoefficientYk >= 0 ? "+" : "-", Math.Round(Math.Abs(CoefficientYk), 3), // k.y 的符号和数值
32.         CoefficientXh >= 0 ? "+" : "-", Math.Round(Math.Abs(CoefficientXh), 3), // h.x 的符号和数值
33.         CoefficientYh >= 0 ? "+" : "-", Math.Round(Math.Abs(CoefficientYh), 3), // h.y 的符号和数值
34.         Constant >= 0 ? "+" : "-", Math.Round(Math.Abs(Constant), 3)); // 常数项的符号和数值
35.
36.
37.
38.
39.         return expression;
40.     }
41. }
42. }

```

## Edge.cs

```

1. using System;
2. // Edge.cs
3. using System;
4.
5. namespace 导线网平差及精度评定程序
6. {
7.     public class Edge
8.     {
9.         // 边的端点
10.        public int Start { get; set; }
11.        public int End { get; set; }
12.
13.        // 边长
14.        public double Length { get; set; }
15.
16.        public Edge()
17.        {
18.        }
19.
20.        // 构造函数
21.        public Edge(int start, int end, double length)
22.        {
23.            Start = start;
24.            End = end;

```

```

25.         Length = length;
26.     }
27.
28.
29.
30.     public Edge(Point p1, Point p2)
31.     {
32.         Start = p1.PointNumber;
33.         End = p2.PointNumber;
34.         Length = CalculateLength(p1, p2);
35.     }
36.
37.     // 计算边长
38.     private double CalculateLength(Point p1, Point p2)
39.     {
40.         return Math.Sqrt(Math.Pow(p2.x - p1.x, 2) + Math.Pow(p2.y - p
41.         1.y, 2));
42.     }
43.     public static Edge CosineLaw(double a, double b, double gammaRadi
44.     ans)
45.     {
46.         Edge edge = new Edge();
47.         edge.Length = Math.Sqrt(a * a + b * b - 2 * a * b * Math.Cos(
48.         gammaRadians));
49.         return edge;
50.     }
51.     public static Edge SineLaw(double a, double b, double angleRadian
52.     s)
53.     {
54.         Edge edge = new Edge();
55.         double c = a * Math.Sin(angleRadians) / Math.Sin(b);
56.         edge.Length = c;
57.         return edge;
58.     }
59. }

```

## Ellipse.cs

```

1. using System;
2. using System;
3. using System.Collections.Generic;
4. using System.Linq;
5. using System.Text;
6. using System.Threading.Tasks;

```

```

7.
8. namespace 导线网平差及精度评定程序
9. {
10.     public class Ellipse
11.     {
12.         public double E; // 长轴长度
13.         public double F; // 短轴长度
14.         public Angle orientation; // 长轴与X轴的夹角
15.         public Point center; // 椭圆中心点
16.
17.         public Ellipse(double E, double F, Angle orientation, Point center)
18.         {
19.             this.E = E;
20.             this.F = F;
21.             this.orientation = orientation;
22.             this.center = center;
23.         }
24.
25.     }
26. }
27.

```

## EquationFormatter.cs

```

1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. using System.Threading.Tasks;
6.
7. namespace 导线网平差及精度评定程序
8. {
9.     class EquationFormatter
10.    {
11.        public string GetEquationString()
12.        {
13.            string equationString = "v1 = -0.276x1 + 0.048y1 - 1.47\n";
14.            equationString += "v2 = 0.219x2 - 0.563y2 + 0.07\n";
15.            equationString += "v3 = 0.284x1 + 0.515y1 - 0.008x2 - 0.563y2\n";
16.            equationString += "v4 = 0.391x1 + 1.274y2 + 0.008x2 + 0.563y2\n";
17.            equationString += "v5 = 0.008x1 + 0.563y1 - 0.453x2 - 0.434y2\n";
18.            equationString += "v6 = 0.445x4 - 0.129y4\n";
19.        }
20.    }
21. }

```

```

18.         equationString += "v6 = -0.008x1 - 0.563y1 - 0.211x2 + 0.862y
    2\n";
19.         equationString += "v7 = -0.399x1 + 0.712y1 + 0.604x3 - 0.403y
    3 - 0.206x4 - 0.308y4\n";
20.         equationString += "v8 = 0.445x2 - 0.129y2 - 0.206x3 - 0.308y3
    - 0.239x4 + 0.437y4\n";
21.         equationString += "v9 = -0.872x2 - 0.591y2\n";
22.         equationString += "v10 = 0.278x2 + 0.961y2 - 0.278x4 - 0.961y
    4\n";
23.         equationString += "v11 = -0.832x3 + 0.555y3 + 0.832x4 - 0.555
    y4\n";
24.         return equationString;
25.     }
26. }
27. }
28.

```

## Point.cs

```

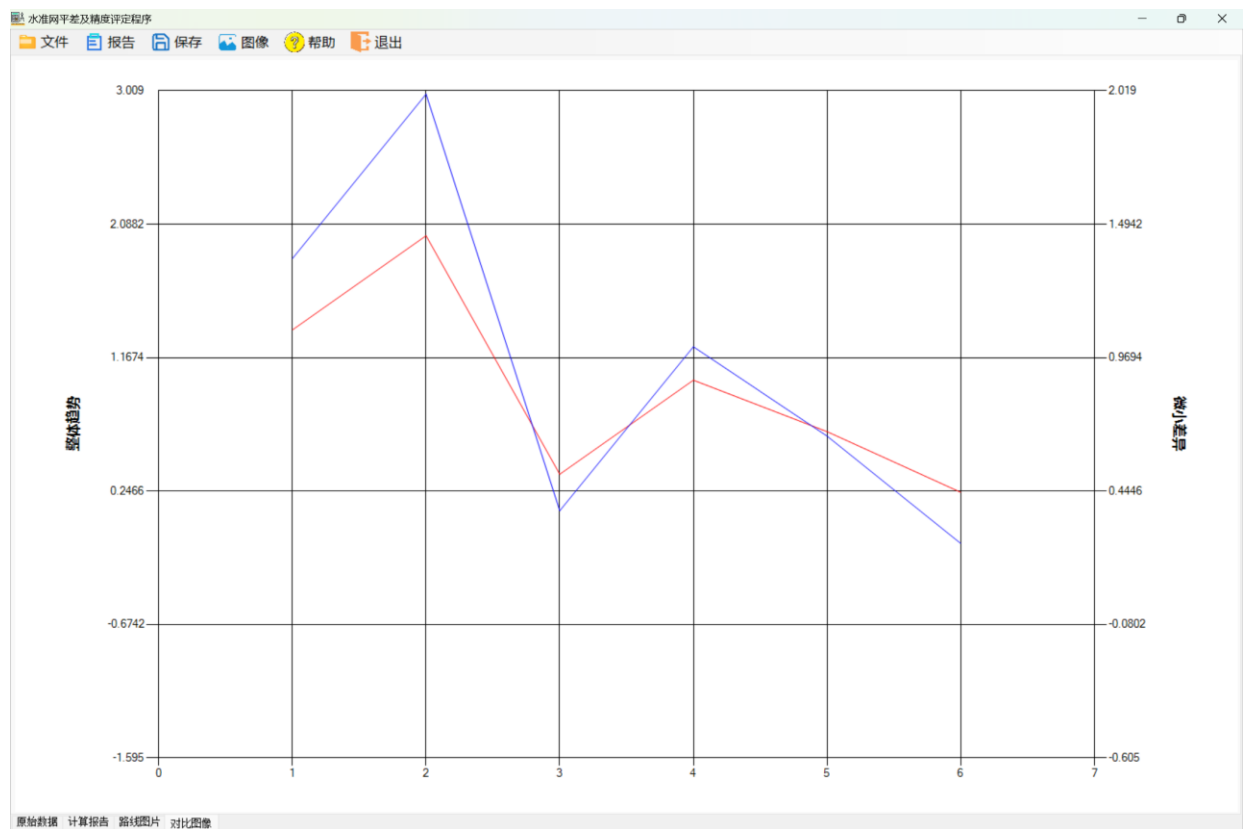
1. using System;
2.
3. namespace 导线网平差及精度评定程序
4. {
5.     public class Point
6.     {
7.
8.         public int PointNumber { get; set; }
9.
10.        public double x { get; set; }
11.
12.        public double y { get; set; }
13.
14.        // 构造函数
15.        public Point(int pointNumber, double x, double y)
16.        {
17.            PointNumber = pointNumber;
18.            this.x = x;
19.            this.y = y;
20.        }
21.
22.        public Point(double v1, double v2)
23.        {
24.            this.x = v1;
25.            this.y = v2;
26.        }

```

```
27.  }  
28. }
```

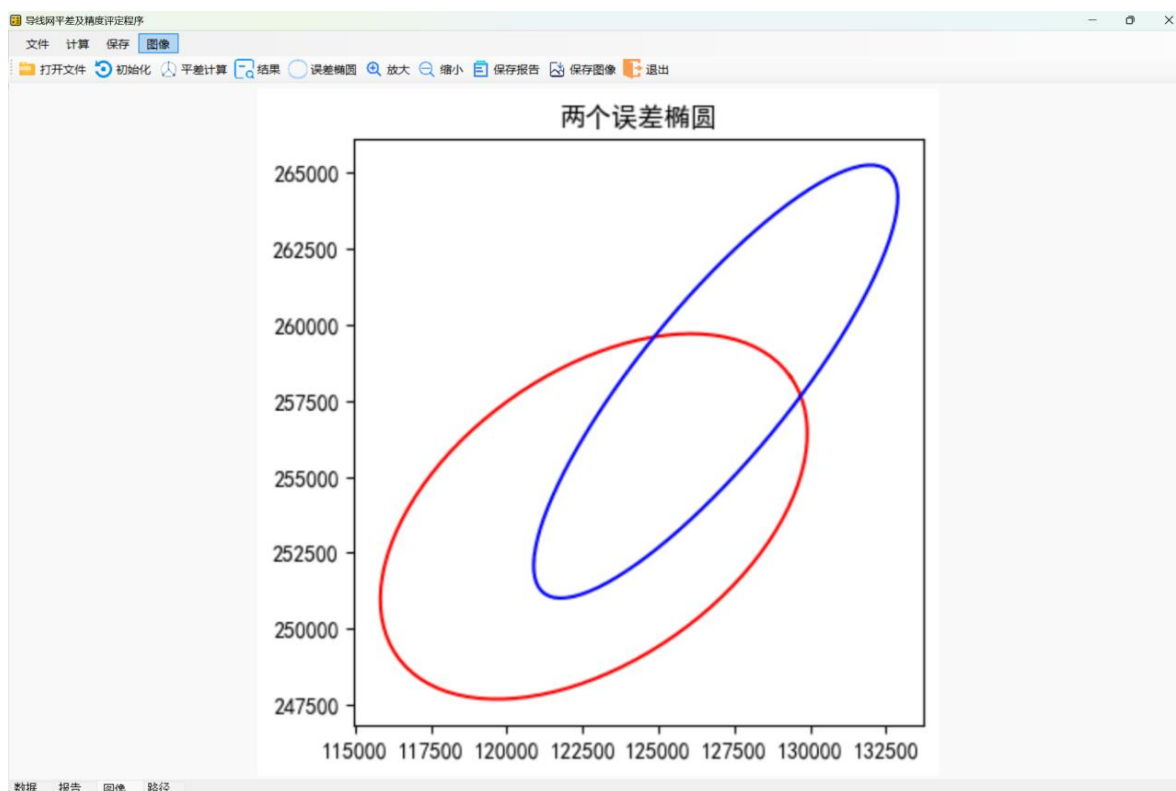
## 六、实例结果分析

### 6.1 水准网平差点对比



### 6.2 导线网误差椭圆





## 七、总结与体会

在参与测量平差课程设计的过程中，我深刻体会到了理论与实践相结合的重要性，以及科学研究中的严谨态度和勤奋精神对于个人成长的价值。本次课程设计不仅让我对测量平差的理论知识有了更深入的理解，还锻炼了我的实际操作能力和计算机编程技能。

在水准网平差及精度评定程序设计部分，我首先系统地复习了水准网平差的理论知识，包括观测值条件方程、误差方程的建立，以及法方程的组成和解算。通过实际操作，我将理论知识与实际问题相结合，学会了如何正确应用公式，处理观测数据，得出了各待定点的高程平差值，并对这些平差值的精度进行了评定。这个过程不仅加深了我对水准网平差理论的理解，还提高了我的数据分析能力和解决实际问题的能力。

在导线网平差及精度评定程序设计部分，我面临着新的挑战。我需要选择合适的平差方法，求出各待定点的坐标值，并评定其中 1-2 个点的精度。在这个过程中，我学会了如何综合考虑各种因素，选择最合适的平差方法，并通过计算机编程实现了这一过程。我还学会了如何绘制误差椭圆，这对我理解测量结果的精度和可靠性具有重要意义。这一部分的设计不仅提高了我的空间想象能力和计算机绘图能力，还增强了我面对复杂问题的解决能力。

在整个课程设计过程中，我深刻体会到了严谨、求实的学风对于科学研究的重要性。每一个步骤都需要严格遵循平差原理，确保数据的准确性和可靠性。同时，勤奋、进取的学风也使我在遇到困难时能够不断探索、积极求解，最终完成了课程设计任务。

此外，本次实验使我认识到计算机编程能力在科学研究中的重要性。通过编写程序，我能够高效地处理大量数据，提高实验的准确性和效率。同时，编程过程中的逻辑思维和问题解决能力也得到了锻炼。我学会了如何将抽象的数学公式转化为具体的计算机代码，这对我未来的科研工作具有极大的帮助。

在总结我的实验报告时，我感到非常荣幸能够参与这次测量平差课程设计。这次设计不仅让我在专业知识和技能上有了很大的提升，还让我在个人素质和科学研究态度上有了更深的认识。我将继续保持这种严谨、求实、勤奋、进取的学风，不断提升自己的专业能力和科学素养，为将来的科研工作和实际应用打下坚实的基础。

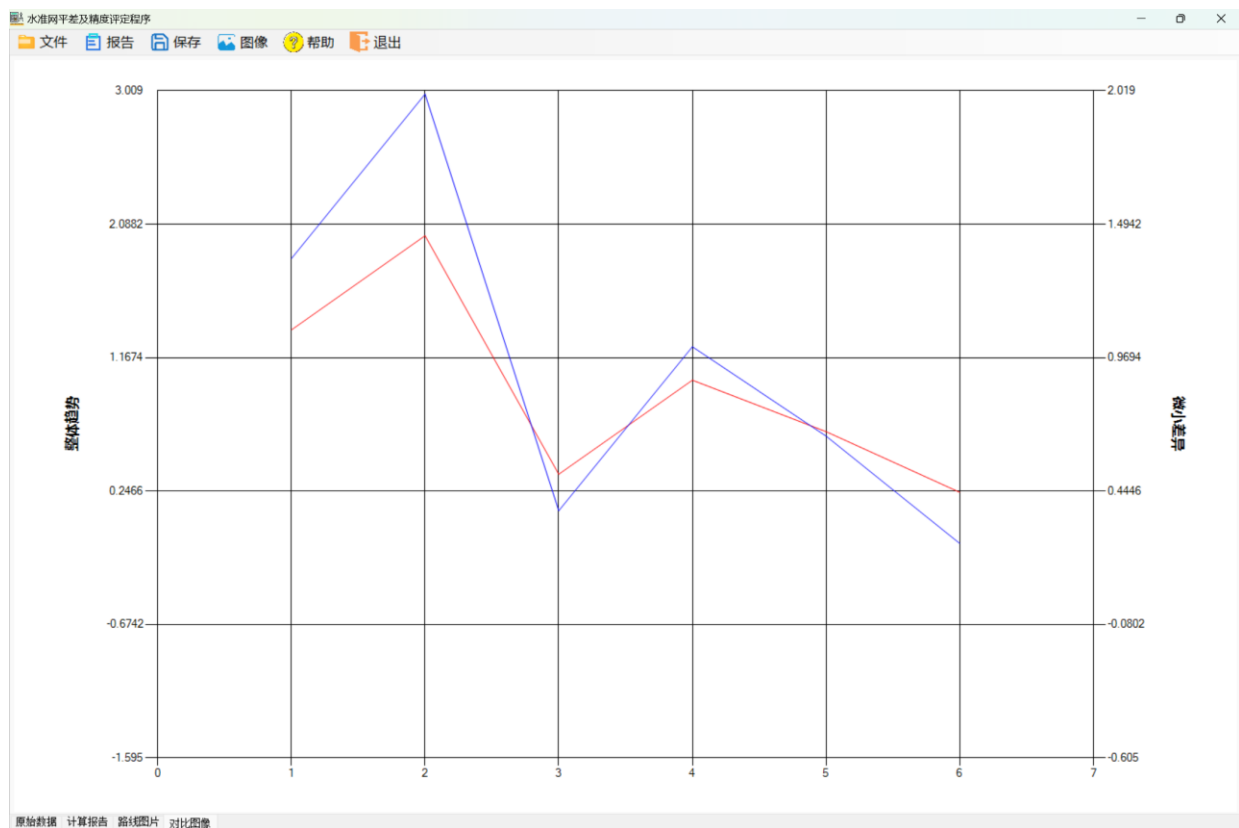
## 八、附件

### 8.1 水准网程序界面



水准网平差及精度评定程序				
文件 报告 保存 图像 帮助 退出				
起始端点	终止端点	高差观测值(m)	测段距离(km)	序号
1	3	1.359	1.1	1
1	4	2.009	1.7	2
2	3	0.363	2.3	3
2	4	1.012	2.7	4
3	4	0.657	2.4	5
3	5	0.238	1.4	6
5	2	-0.595	2.6	7
H1:5.016 H2:6.016				
原始数据 计算报告 路线图片 对比图像				

水准网平差及精度评定程序				
文件 报告 保存 图像 帮助 退出				
<p>得正点5不是二条及二条以上水准路线交叉点 把h6和h7合并为一条水准路线 设3, 4点高程依次为x1, x2</p> <p>Matrix B:</p> <pre> 1 0 0 1 1 0 0 1 -1 1 -1 0 </pre> <p>Matrix P:</p> <pre> 0.91 0 0 0 0 0 0.59 0 0 0 0 0 0.44 0 0 0 0 0 0.37 0 0 0 0 0 0.42 0 0 0 0 0.25 </pre> <p>Matrix NBB:</p> <pre> 2.01 -0.42 -0.42 1.38 </pre> <p>Vector w:</p> <pre> 1.680 -4.030 </pre> <p>Vector x:</p> <pre> 0.240 -2.860 </pre> <p>改正数: 0.243, -2.857, 4.243, 0.143, 3.901, 0.615, 1.142 改正后高程: 1.359, 2.006, 0.367, 1.012, 0.661, 0.239, -0.594</p> <p>*****</p> <p>问题一:</p> <pre> 3号点高程:6.3752m 4号点高程:7.0221m 5号点高程:6.6099m </pre> <p>*****</p> <p>问题二:</p> <pre> sigma:2.2mm CD点高程中误差: C:1.6mm D:1.9mm </pre> <p>*****</p> <p>问题三:</p> <pre> h5中误差:2.2mm </pre>				
原始数据 计算报告 路线图片 对比图像				



## 8.2 导线网程序界面

导线网平差及精度评定程序

文件 计算 保存 图像

打开文件 初始化 平差计算 结果 误差椭圆 放大 缩小 保存报告 保存图像 退出

点号	x(m)	y(m)	起始点号	终止点号	角度	端点号1	端点号2	边长
1	121088.500	259894.000	1	2	0.0000	4	6	4451.417
2	127990.100	255874.600	1	3	72.10284	2	4	5504.592
			2	4	0.0000	6	5	5509.209
			2	1	66.27289			
			3	1	0.0000			
			3	4	88.58295			
			3	5	212.10036			
			4	6	0.0000			
			4	3	85.13374			
			4	2	217.37126			
			5	6	0.0000			
			5	3	79.09487			
			6	4	0.0000			
			6	5	72.24564			

已知信息

数据 报告 图像 路径

