程序开发文档

参赛学校：西南交通大学

参赛人员：许少鑫，刘义豪

参赛时间：2022/07/12

目录

[一、 程序功能简介 3](#_Toc108599918)

[二、 算法设计与流程图 4](#_Toc108599919)

[2.1 算法设计 4](#_Toc108599920)

[2.1.1白塞尔大地反算 4](#_Toc108599921)

[2.1.2 白塞尔大地正算 4](#_Toc108599922)

[2.2 流程图设计 4](#_Toc108599923)

[三、 主要函数与变量说明 6](#_Toc108599924)

[3.1计算类函数与变量说明 6](#_Toc108599925)

[3.1.1计算类函数说明 6](#_Toc108599926)

[3.1.2 计算类变量说明 7](#_Toc108599927)

[3.2 绘图类函数与变量说明 7](#_Toc108599928)

[3.2.1绘图类函数说明 7](#_Toc108599929)

[3.2.2绘图类变量说明 8](#_Toc108599930)

[四、 主要程序运行界面 9](#_Toc108599931)

[五、 使用说明 14](#_Toc108599932)

[5.1数据说明 14](#_Toc108599933)

[5.2程序操作说明 14](#_Toc108599934)

[5.2.1 数据读取写入操作 14](#_Toc108599935)

[5.2.2白塞尔大地主题解算 15](#_Toc108599936)

[5.2.3 成果报告输出以及保存 16](#_Toc108599937)

[5.2.4 示意图输出以及保存 17](#_Toc108599938)

大地主题解算计算程序开发文档

# 程序功能简介

本程序功能主要包括：数据文件的读取（文本文档数据），利用白塞尔法进行大地主题反算，利用白塞尔法进行大地主题正算，成果报告、成果示意图的输出以及保存，如图1.1。



图1.1 程序功能简要介绍

# 算法设计与流程图

## 2.1 算法设计

### 2.1.1白塞尔大地反算

利用起算数据采取逐次趋近法计算起点大地方位角和经差，当两次迭代后产生的δ相同或小于允许值时，则该次迭代结果便作为计算结果。根据大地方位角计算大地线长度，最后计算大地反方位角。

### 2.1.2 白塞尔大地正算

首先计算起点的归化纬度，利用椭球数据和起算数据计算白塞尔大地主题正算中的赋值系数，通过迭代运算计算球面长度，得到球面长度后计算精度差改正数，最后利用球面长度和经差计算终点的大地坐标和坐标方位角。

## 2.2 流程图设计

本程序主要流程为：程序开始——文本文档数据的写入——判定是否符合数据规范——白塞尔大地主题反算——白塞尔大地主题正算——成果报告和示意图的输出和保存——程序结束，如图2.1



图2.1 程序流程图

# 主要函数与变量说明

## 3.1计算类函数与变量说明

### 3.1.1计算类函数说明

观测值变量用于保存每一对大地坐标的数据，其主要属性包括两个大地点和大地线长。其中点为点类对象，包含经纬度，大地方位角，和点名属性，其关系类图如图3.1所示。

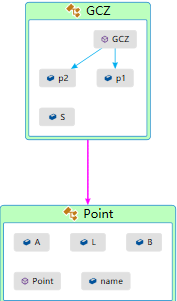


图3.1 观察值类与点类的关系及其属性

另外还需设计一个椭球类来放置椭球参数，主要属性如图3.2所示。

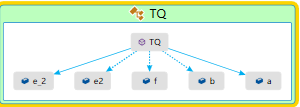


图3.2 椭球类属性

### 3.1.2 计算类变量说明

|  |  |  |
| --- | --- | --- |
| 变量名 | 数据类型 | 数据说明 |
| Aaa | nnn | nnn |
| ；‘l | nn | NN |
| nnnn | nnn | NN |
| nnn | n | NN |

主要含有两个计算函数，分别是正算和反算函数，其输入值为上述的GCZ类变量和椭球类变量。

static public void ZS(GCZ gcz, TQ tq)//正算

static public void FS(GCZ gcz,TQ tq)//反算

## 3.2 绘图类函数与变量说明

主要含有两个计算函数，分别是正算和反算函数，其输入值为上述的GCZ类变量和椭球类变量。

static public void ZS(GCZ gcz, TQ tq)//正算

static public void FS(GCZ gcz,TQ tq)//反算

## 3.2.1绘图类函数说明

foreach (GCZ gcz in gczs)

{

if (gcz.p1.B > maxB) maxB = gcz.p1.B;

if (gcz.p1.L > maxL) maxL = gcz.p1.L;

if (gcz.p1.B < minB) minB = gcz.p1.B;

if (gcz.p1.L < minL) minL = gcz.p1.L;

if (gcz.p2.B > maxB) maxB = gcz.p2.B;

if (gcz.p2.L > maxL) maxL = gcz.p2.L;

if (gcz.p2.B < minB) minB = gcz.p2.B;

if (gcz.p2.L < minL) minL = gcz.p2.L;

}

//计算图像上绘图时最大点和最小点

dsb = Math.Abs(maxB - minB) / (rect.Height - 2 \* margin);

dsl = Math.Abs(maxL - minL) / (rect.Width - 2 \* margin);

//得到将实际距离转换成为屏幕距离的比例尺

foreach (GCZ gcz in gczs)

{

double x0 = margin + (gcz.p1.B - minB) / dsb;

double y0 = (rect.Height - margin) - (gcz.p1.L - minL) / dsl;

double x1 = margin + (gcz.p2.B) / dsb;

double y1 = (rect.Height - margin) - (gcz.p2.L - minL) / dsl;

g.DrawLine(p1, (float)x0, (float)y0, (float)x1, (float)y1);

//绘制大地线

double x = margin + (gcz.p1.B - minB) / dsb;

double y = (rect.Height - margin) - (gcz.p1.L - minL) / dsl;

string name0 = gcz.p1.name;

double xx = margin + (gcz.p2.B) / dsb;

double yy = (rect.Height - margin) - (gcz.p2.L - minL) / dsl;

string name1 = gcz.p2.name;

g.DrawEllipse(p2, (float)x, (float)y, (float)2, (float)2);

g.FillEllipse(b2, (float)x, (float)y, (float)2, (float)2);

g.DrawEllipse(p2, (float)xx, (float)yy, (float)2, (float)2);

g.FillEllipse(b2, (float)xx, (float)yy, (float)2, (float)2);

g.DrawString(name0, f1, b1, (float)x, (float)y);

g.DrawString(name1, f1, b1, (float)xx, (float)yy);

//绘制数据点和结果点

}

if (mousedown)//判断鼠标是否按下

{

endPoint = e.Location;

dx = (endPoint.X - startPoint.X)\*dsb;

dy = (endPoint.Y - startPoint.Y)\*dsl;

//鼠标移动距离

minB -= dx;

minL += dy;

//图像移动距离

startPoint = e.Location;

}

### 3.2.2绘图类变量说明

public static double minB; //存储画布横轴最小值

public static double maxB; //存储画布横轴最大值

public static double minL; //存储画布纵轴最小值

public static double maxL; //存储画布纵轴最大值

public static double margin; //画布边界规定

public static double dsb; //绘图比例尺，将横轴上的实

//距离换为屏幕距离

public static double dsl; //绘图比例尺，将纵轴上的实际

//距离转换为屏幕距离

public static bool mousedown=false; //图像移动，记录鼠标结束

public static System.Drawing.Point endPoint; //图像移动，记录鼠标结束位置

public static System.Drawing.Point startPoint;//图像移动，记录鼠标开 //始位置

# 主要程序运行界面

程序运行开始读取数据界面，如图4.1，图4.2

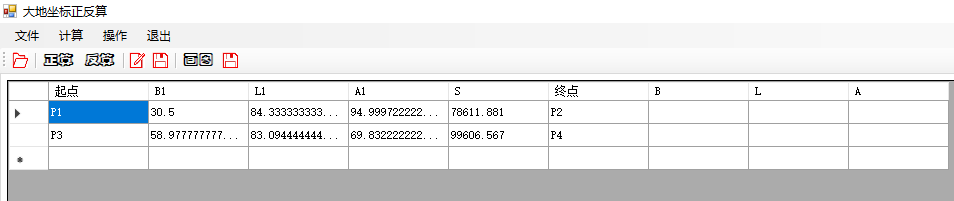


图4.1 程序运行开始读取数据界面（正算）

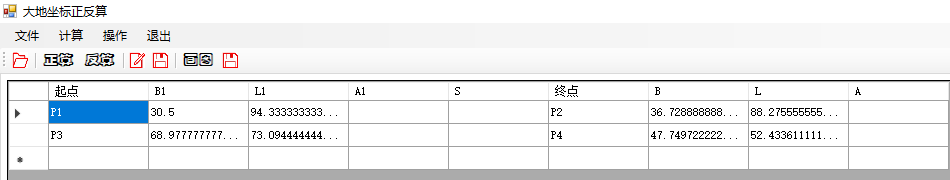


图4.2 程序运行开始读取数据界面（反算）

白塞尔大地主题反算，如图4.3

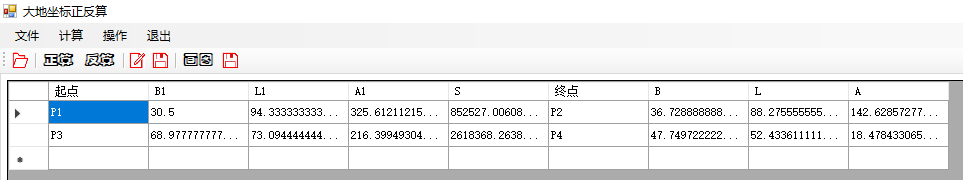


图4.3 白塞尔大地主题反算

白塞尔大地主题正算，如图4.4

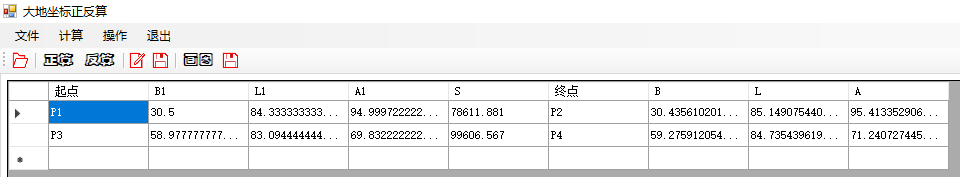


图4.4 白塞尔大地主题反算

白塞尔大地主题解算成果报告输出，如图4.5，图4.6

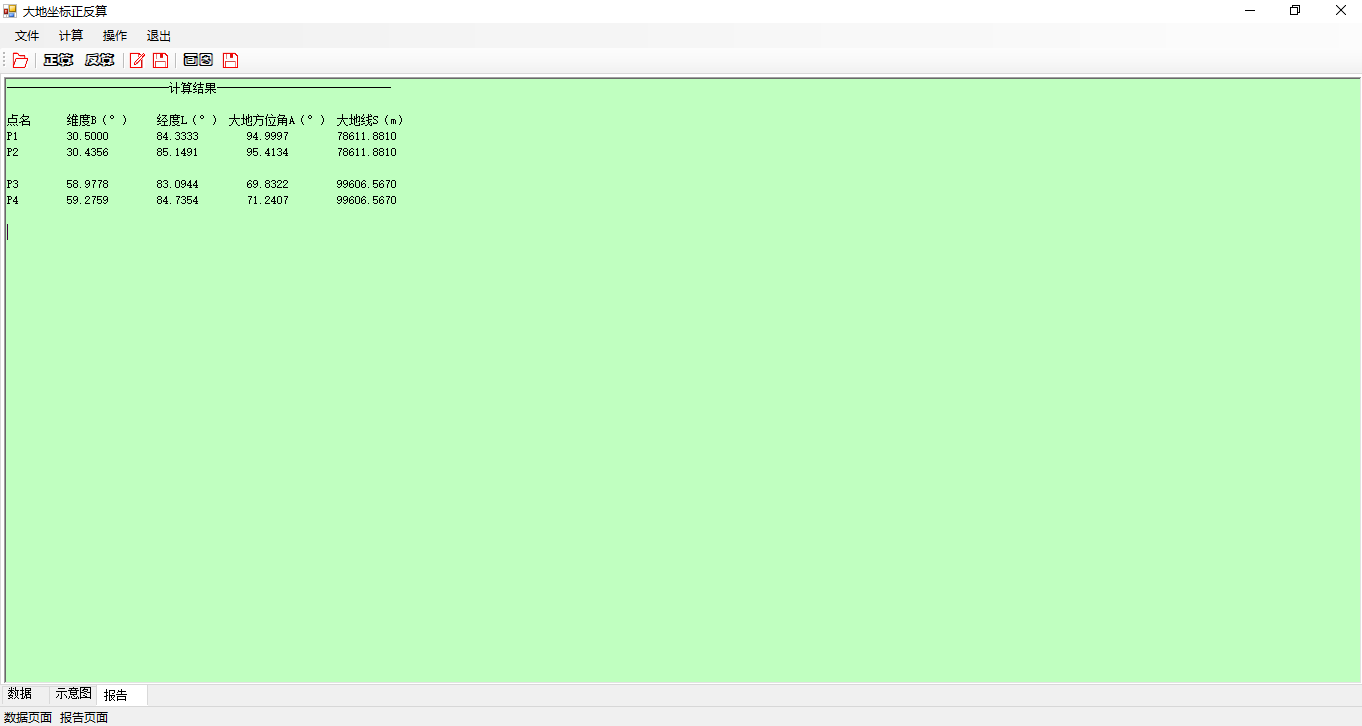


图4.5 白塞尔大地主题解算成果报告（正算）

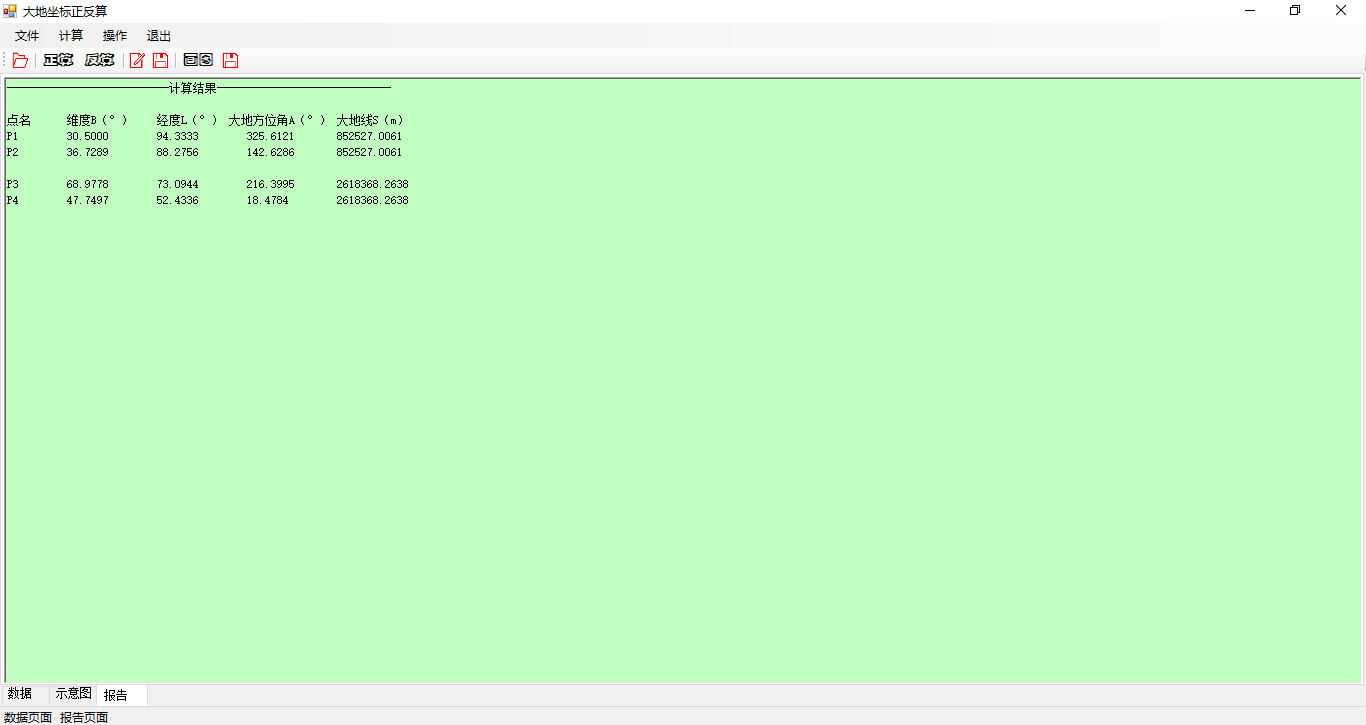


图4.6 白塞尔大地主题解算示意图输出（反算）

白塞尔大地主题解算示意图输出，如图4.7，图4.8

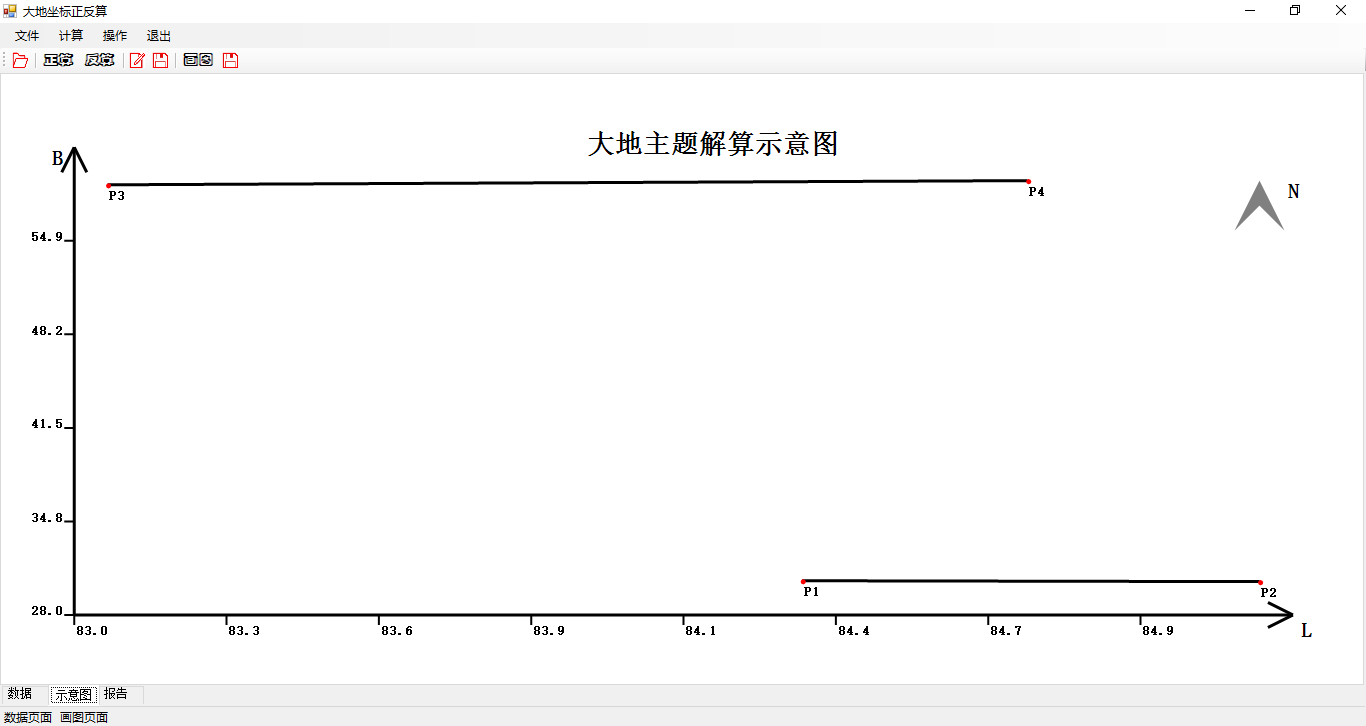


图4.7 白塞尔大地主题解算示意图输出（正算）

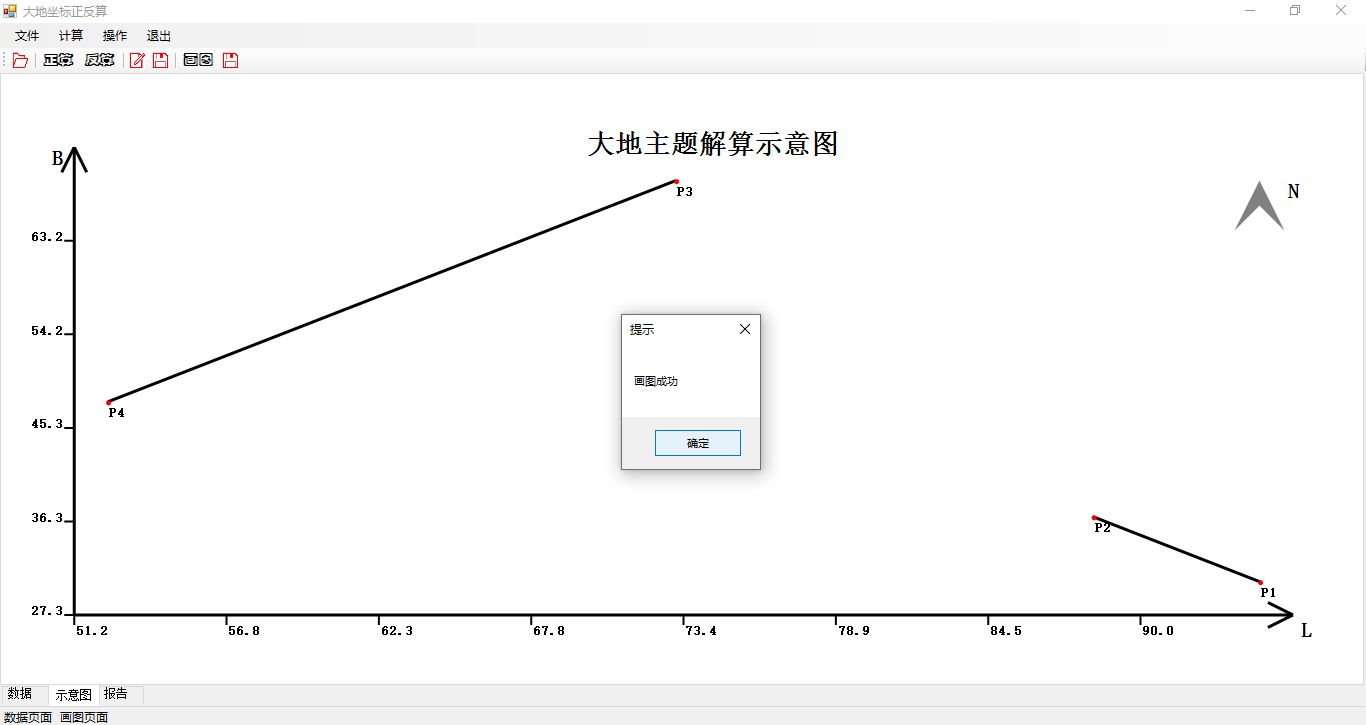


图4.8 白塞尔大地主题解算示意图输出（反算）

白塞尔大地主题解算成果报告保存，如图4.9，图4.10

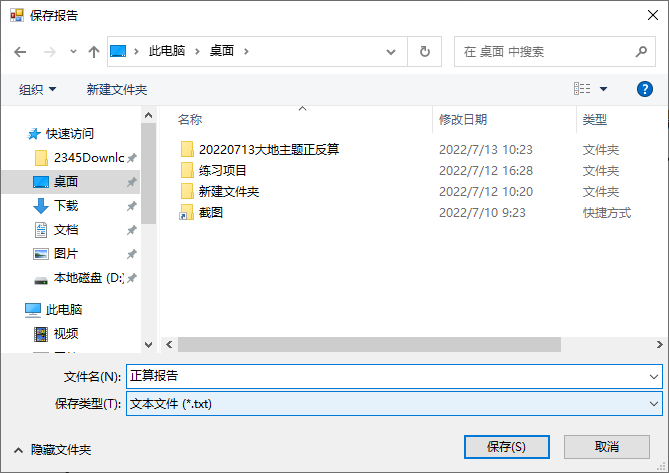


图4.9 白塞尔大地主题反解算成果报告保存（正算）

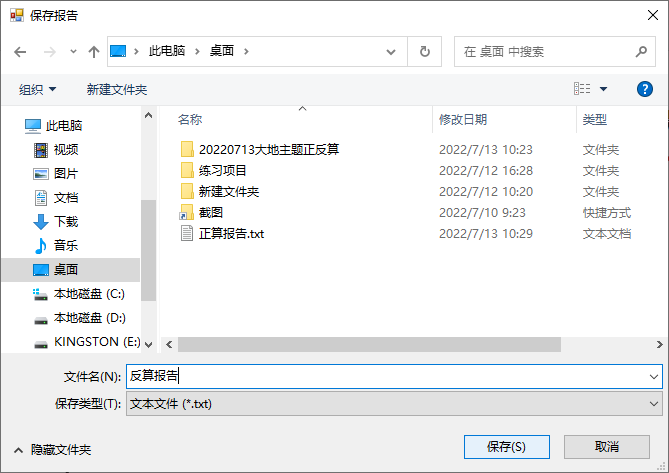


图4.10 白塞尔大地主题反解算成果报告保存（反算）

白塞尔大地主题解算示意图保存，如图4.11，图4.12

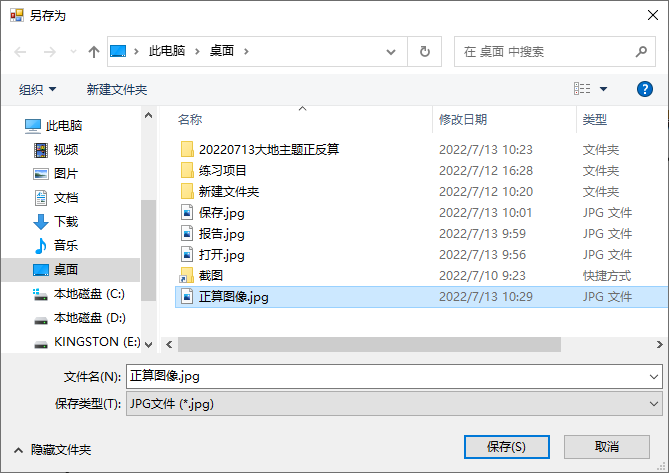


图4.11 白塞尔大地主题解算示意图保存（正算）

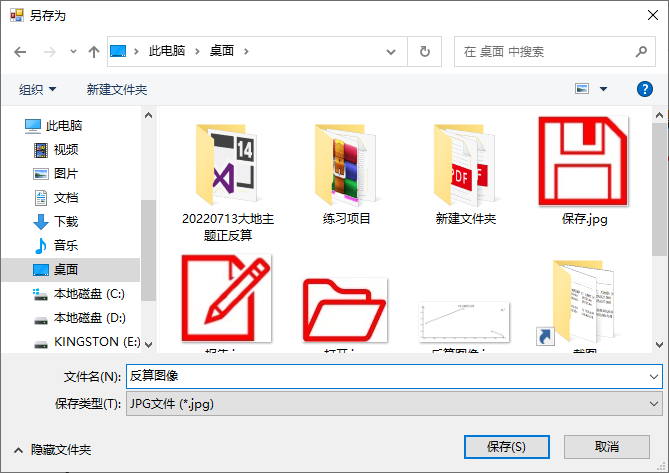


图4.12 白塞尔大地主题解算示意图保存（正算）

# 使用说明

## 5.1数据说明

采用文本文档作为输入数据格式，正算数据为：第一行为椭球长半轴，扁率倒数；第二行及以后行为起点，纬度（dd.mmss），经度（dd.mmss），大地方位角（dd.mmss），大地线长度（m）,终点。反算数据为：第一行为椭球长半轴，扁率倒数；第二行及后续行为，起点，纬度（dd.mmss），经度（dd.mmss）,终点，纬度（dd.mmss），经度（dd.mmss）。如图5.1（反算数据），图5.2（正算数据）.

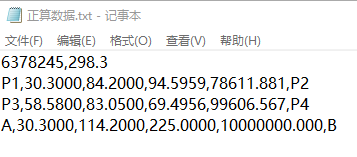


图5.1 白塞尔大地主题反算数据

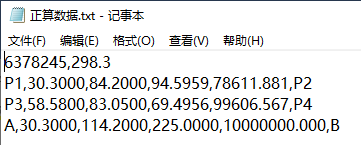


图5.2 白塞尔大地主题正算数据

## 5.2程序操作说明

### 5.2.1 数据读取写入操作

点击菜单栏中“打开”选项卡，点击“打开数据”，或直接点击工具栏中的文件图片按钮，如图5.3和图5.4

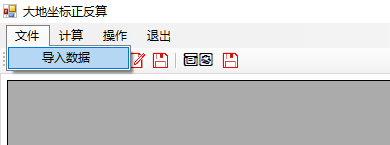


图5.3 打开数据操作1

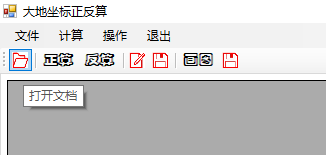


图5.4 打开数据操作2

### 5.2.2白塞尔大地主题解算

点击菜单栏中“计算”选项卡，点击“正算”进行白塞尔大地正算或点击“反算”进行大地主题反算，如图5.5

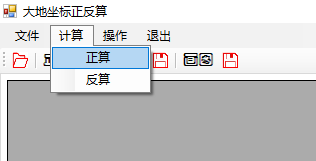
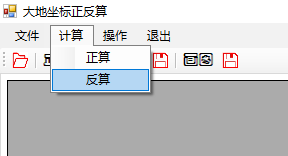


图5.5 数据计算操作1

或点击工具栏中“正算”按钮进行白塞尔大地反算，点击“反算”按钮进行白塞尔大地正算，如图5.6

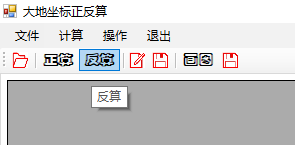
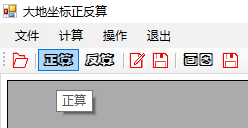


图5.6 数据计算操作2

### 5.2.3 成果报告输出以及保存

得到计算结果后，点击生成报告，即可生成报告，如图5.7，图5.8

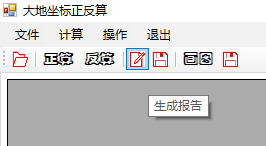
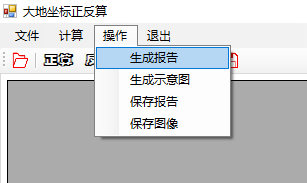


图5.7 成果报告输出

得到报告以后，点击保存报告，弹出对话框选择保存路径，即可实现报告保存，如图5.9和图5.10

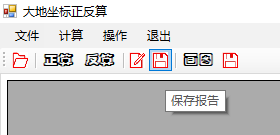
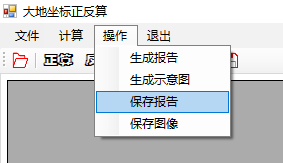


图5.8 成果报告保存

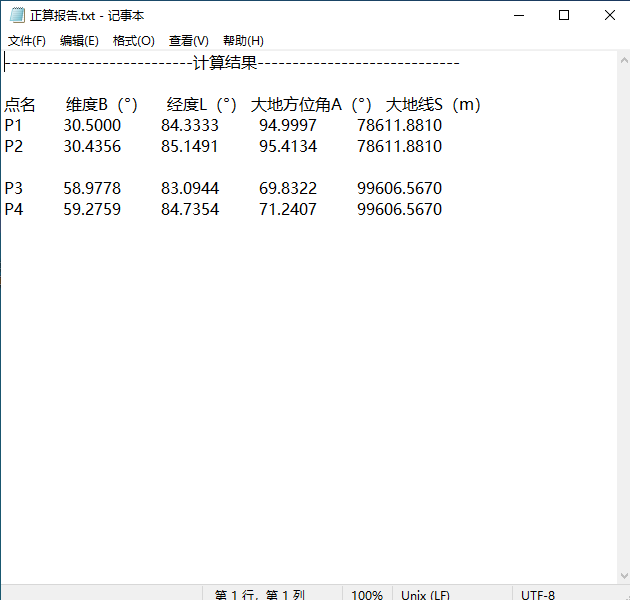


图5.9 成果报告保存样例（正算）

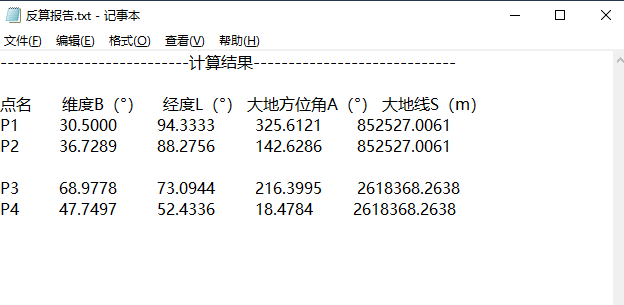


图5.10 成果报告保存样例（反算）

### 5.2.4 示意图输出以及保存

得到计算结果后，点击生成示意图，即可生成示意图，如图5.11， 图5.12

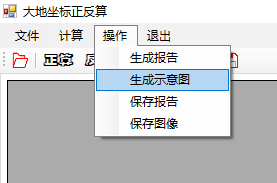
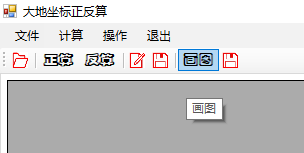


图5.11 示意图报告输出

得到示意图以后，点击保存示意图，弹出对话框选择保存路径，即可实现示意图保存，如图5.13和图5.14

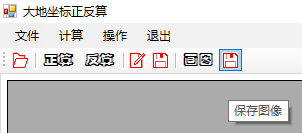
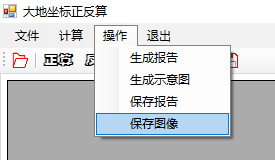


图5.12 示意图保存

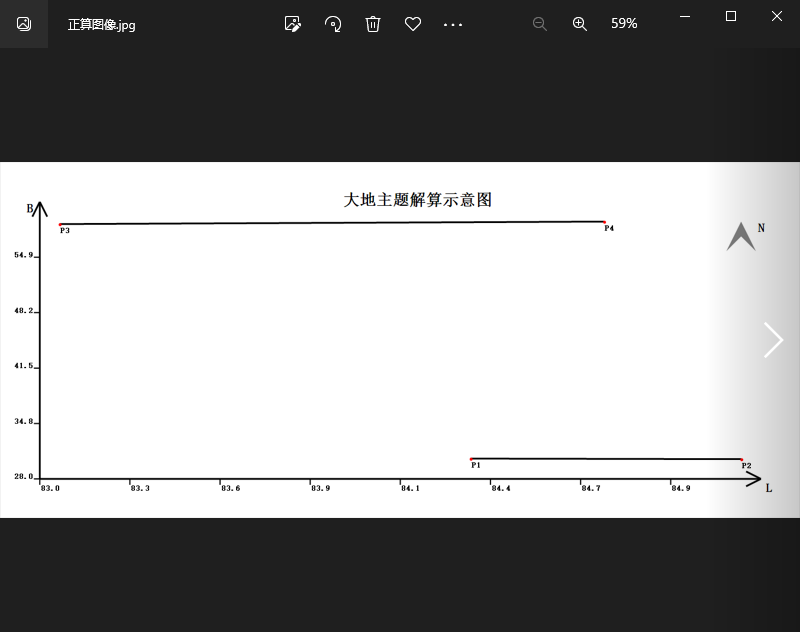


图5.13 示意图保存样例（正算）

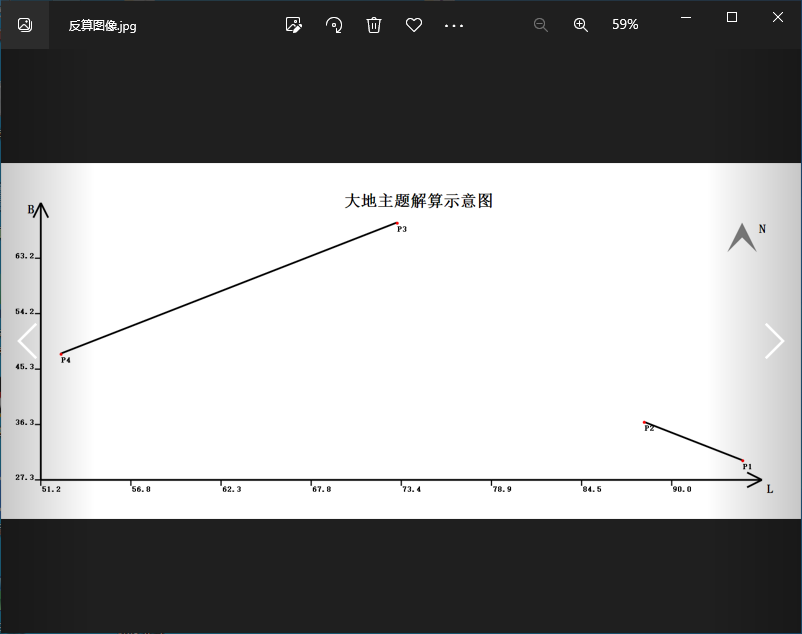


图5.14 示意图保存样例（反算）