# 程序功能简介

本软件为GNSS接收机坐标估算软件，通过已给的导航定位源文件，可借助该程序进行导入数据并计算得到经过最小二乘法、卡尔曼滤波法、卡尔曼滤波CV模型与卡尔曼滤波CA模型四种算法改正后的GNSS坐标，并将结果进行绘图。软件的主要功能包括数据的导入与存储、解算算法的选择、解算历元的范围选择，解算状态的显示和解算结果的绘图。

## 数据载入

文件类型：Rinex格式；

数据输入格式为：

1. O文件：头文件+数据文件 数据文件包括单次历元内的基本信息与历元内每颗卫星的卫星编号+伪距观测值+卫星钟差+卫星对流层改正数+卫星的坐标信息+卫星的高度角；
2. P文件：头文件+数据文件 数据文件包括每颗卫星的记录时间和与卫星有关的各个参数。

首先读取O文件的头文件，提取测站的近似坐标，再读取个每个历元的基本信息，再通过基本信息循环读入每颗卫星的信息，P文件同理；

文件内容如图1和图2所示：

图 1卫星O文件数据

图 2卫星P文件数据

## 坐标值的计算

程序先通过头文件确定接收机的近似坐标值，再通过卫星伪距单点定位方程列出伪距观测方程，并对伪距进行泰勒级数展开，之后通过相应的算法计算坐标的改正数，并将改正数与近似坐标相加，从而得到接收机坐标较为准确的估计值。

## 数据信息的显示与绘图

在导入数据后，程序会自动提取头文件内容中的近似坐标，并显示在UI上，点击解算后，UI会实时显示正在解算第多少个历元，并将当前历元计算得到的改正数和改正后的坐标实时显示在UI上，同时将这些信息存储在矩阵变量内，在点击绘图后，会自动以历元为横坐标，解算结果为纵坐标进行绘图。

## 数据的保存

点击解算后，UI会实时显示正在解算第多少个历元，并将当前历元计算得到的改正数和改正后的坐标实时显示在UI上，同时将这些信息存储在矩阵变量内，在点击保存后，选择要保存的路径和要保存的文件名，程序会自动以历元数+X坐标+Y坐标+Z坐标+X改正数+Y改正数+Z改正数的格式将矩阵中的内容输出到文件中。

# 主要算法设计与流程图

## 算法设计

此程序是通过对卫星伪距单点定位方程进行线性化后，分别通过最小二乘算法、卡尔曼滤波算法、卡尔曼滤波CV算法和卡尔曼滤波CA算法计算坐标改正数后对接收机近似坐标进行改正，得到估计后的接收机坐标。

## 算法流程图

# 主要函数和变量说明

## 主要函数

包括了读取数据、数据计算、数据绘图与数据输出的主要使用函数。

1. void **txt\_reader**(std::string filename, std::vector<Satellite>& vec)

功能：将TXT的数据读取到结构体数组中；

输入：数据文件的绝对路径，结构体变量名；

输出：接收机头文件中的近似坐标与所有历元的所有卫星信息

1. void MainWindow::**on\_LSM\_Button\_clicked**()

功能：运用最小二乘法计算接收机坐标的改正数；

输入：存储卫星信息的结构体数组与接收机近似坐标的数组

输出：通过最小二乘法解算的接收机坐标改正数与经过改正后的接收机坐标

1. void MainWindow::**on\_KALMAN\_Button\_clicked**()

功能：运用卡尔曼滤波法计算接收机坐标的改正数；

输入：存储卫星信息的结构体数组与接收机近似坐标的数组

输出：通过卡尔曼滤波法解算的接收机坐标改正数与经过改正后的接收机坐标

1. void MainWindow::**on\_KALMANCVButton\_clicked**()

功能：运用卡尔曼滤波CV模型计算接收机坐标的改正数；

输入：存储卫星信息的结构体数组与接收机近似坐标的数组

输出：通过卡尔曼滤波CV模型解算的接收机坐标改正数与改正后的接收机坐标

1. void MainWindow::**on\_KALMANCAButton\_clicked**()

功能：运用卡尔曼滤波CA模型计算接收机坐标的改正数；

输入：存储卫星信息的结构体数组与接收机近似坐标的数组

输出：通过卡尔曼滤波CA模型解算的接收机坐标改正数与改正后的接收机坐标

1. drawxx::**drawxx**(Eigen::MatrixXd POS1, int geshu1, double \*APPROX1, QWidget \*parent)

功能：对解算改正后的X坐标进行绘图

输入：经过改正后的坐标矩阵，解算的历元个数，接收机头文件中近似坐标

输出：X坐标的曲线图

1. drawyy::**drawyy**(Eigen::MatrixXd POS1, int geshu1, double \*APPROX1, QWidget \*parent)

功能：对解算改正后的Y坐标进行绘图

输入：经过改正后的坐标矩阵，解算的历元个数，接收机头文件中近似坐标

输出：Y坐标的曲线图

1. drawzz::**drawzz**(Eigen::MatrixXd POS1, int geshu1, double \*APPROX1, QWidget \*parent)

功能：对解算改正后的Z坐标进行绘图

输入：经过改正后的坐标矩阵，解算的历元个数，接收机头文件中近似坐标

输出：Z坐标的曲线图

1. void MainWindow::**baocun**()

功能：对解算结果进行保存

输入：改正数矩阵，经过改正后的坐标矩阵

输出：历元数+X坐标+Y坐标+Z坐标+X改正数+Y改正数+Z改正数的结果文件

## 主要变量

1. typedef struct Data
2. { std::string PRN;double satnum, sats, satPos[3], sat\_Clock, Elevation, Azimuth, Prange[2], L[2], Trop\_Delay, Trop\_Map, Relativity, Sagnac, Tide\_Effect, Antenna\_Height, Sat\_Antenna, Tide\_Offset[2], Windup; }Data;//小结构体，用于存放单颗卫星的数据
3. typedef struct Satellite {int Satellite\_Num;double gpst, ztd, gps\_clock, glonass\_clock;std::vector<Data> data;
4. }Satellite;//大结构体，用于存放整个历元的数据
5. double APPROX\_POSITION[3];//全局变量，接收机近似坐标
6. double X0=0, Y0=0, Z0=0;//全局变量存储接收机位置
7. MatrixXd POS(2, 2);//坐标结果矩阵
8. MatrixXd ANS(4, 1); //坐标改正数矩阵
9. int geshu; //解算历元数
10. MatrixXd H0(1, 4);//创建系数矩阵
11. MatrixXd L(2, 2);//创建观测值矩阵
12. MatrixXd lsm\_ans(4, 1);//创建改正数计算矩阵
13. MatrixXd P0(2, 2);//创建权矩阵
14. MatrixXd fuzhu(2, 2);//创建辅助计算矩阵
15. double dx; //创建X差值
16. double dy; //创建Y差值
17. double dz; //创建Z差值
18. double R0; //创建距离
19. double l; //创建系数l
20. double m; //创建系数m
21. double n; //创建系数n

# 程序主界面与绘图