

# matlab开发文档

## 一. 开发环境及工具

硬件运行环境为本试验所用的计算机设备，CPU为Inter(R)Xeno(R) Gold 5218 CPU @ 2.3GHz，GPU为NVIDIA GeForce RTX 4060。

开发环境基于Windows 11操作系统，matlab R2022a。

## 二. 程序的功能作简要介绍

### 1、程序的功能作简要介绍

本次《测绘程序设计》期末考核要求同学们进行“附和导线近似平差”和“高斯正算”两个程序的编写，以检验其对测绘知识和程序设计的掌握程度。

首先，要完成这个任务，需要理解并掌握附和导线近似平差和高斯正算的基本原理，以及相关的数学模型。然后，根据给定的数据和条件，进行程序设计和编码。

在程序设计中，本程序努力做到以下几点：

1. 正确性：程序应能正确完成计算任务，得出正确的结果。
2. 完整性：程序应结构完整，包括合理的函数设计、语句调用等。
3. 规范性：程序命名、注释、变量和函数命名等应符合规范。
4. 优化性：程序应简洁易读，有良好的可读性和可维护性。

### 附和导线

附和导线是指连接两个已知控制点的一种导线，通常用于测量和计算两点之间的距离、角度和坐标。在附和导线程序中，需要输入导线的起点和终点的坐标以及观测的边长和角度，程序会根据这些数据计算出导线的长度、方位角、高程等信息，并输出相应的结果。附和导线程序的功能还包括对导线进行平差处理，消除误差，提高测量精度。

### 高斯正算

高斯正算是一种用于计算坐标正反算的方法，即根据已知点的坐标和方位角计算出待测点的坐标。在高斯正算程序中，用户需要输入已知点和待测点的坐标以及方位角等信息，程序会根据高斯投影的原理和公式，计算出待测点的坐标，并输出结果。高斯正算程序还可以进行坐标转换和地图投影等操作，是测量学中常用的工具之一。

## 2、流程图

- 顶层数据流图

## 附和导线

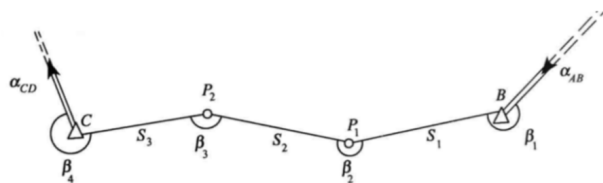
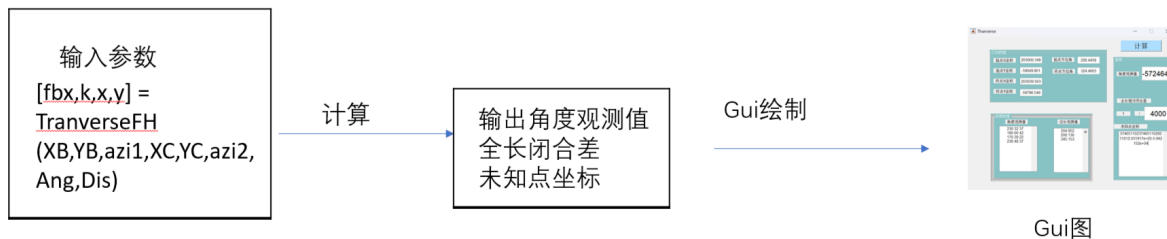


图 1 单一附和导线略图

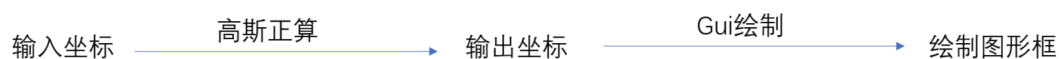
### 附和导线计算

## 高斯正算

B=37.3915  
L=111.5654  
中央子午线 经度为 114 度

x = 5391779.90793839  
y = -181055.800288334

高斯		高斯	
大地经度	37.3915	计算	
大地纬度	111.5654		
中央经线	114		
椭球选择	3		
		横坐标	5391779.90793839
		纵坐标	-181055.80028833



## 三. 编写目的

- 期末考核共包含两道题目，即“附和导线近似平差”和“高斯正算”程序设计
- 提高程序编写水平
- 练习开发文档编写

## 四. 算法设计思路（函数模块）

### 附和导线

function附和计算代码块

```

1 function [fbx,k,x,y] = TranverseFH(XB,YB,azi1,XC,YC,azi2,Ang,Dis)
2 %% 解算附和导线
3 % x,y 返回待定点坐标, fwj 平差后各边的方位角
4 % fbx 返回角度闭合差,k全长相对闭合差
5 % XB,YB,XC,YC 已知点坐标x
6 % azi1和azi2已知起始边和终边方位角
7 % Ang,Dis是观测的角度(弧度)和边长
8 %数据输入和处理
9 n=length(Ang); %获得观测角个数
10 [radangle]=dms_rad(Ang);% 角度转换为弧度
11 zb=sum(radangle); %观测角之和
12 azi1=dms_rad(azi1);
13 azi2=dms_rad(azi2);
14 fb=zb-n*pi-azi2+azi1; %计算角度闭合差
15 fb=rem(fb,2*pi); %取余数
16 fbx=round(fb*206264.80); %将角度闭合差换成秒
17 radangle=radangle-fb./n;
18 for i=1:n
19     if i==1
20         fwj(i) = azi1-pi+radangle(i);
21     else
22         fwj(i)=fwj(i-1)-pi+radangle(i);
23     end
24 end
25 dx=cos(fwj(1:n-1)).*Dis;
26 dy=sin(fwj(1:n-1)).*Dis;
27 fx=sum(dx)+XB-XC;
28 fy=sum(dy)+YB-YC;
29 fs=sqrt(fx^2+fy^2);
30 Zd=sum(Dis);
31 k=fix(Zd/fs);
32 dx=dx-(fx/Zd).*Dis;%坐标增量闭合差的分配
33 dy=dy-(fy/Zd).*Dis;
34 for i=1:n-1
35     if i==1
36         x(i)=XB+dx(i);
37         y(i)=YB+dy(i);
38     else
39         x(i)=x(i-1)+dx(i);

```

```
40         y(i)=y(i-1)+dy(i);
41     end
42 end
```

## gui代码块

```
1 function pushbutton1_Callback(hObject, eventdata, handles)
2 %pushbutton1代表是"计算", 有关计算的代码都在此按钮下面
3 str=get(handles.edit1,'string');
4 X1=str2num(str);
5 str=get(handles.edit2,'string')
6 Y1=str2num(str);
7 str=get(handles.edit3,'string')
8 X2=str2num(str);
9 str=get(handles.edit4,'string')
10 Y2=str2num(str);
11 str=get(handles.edit5,'string')
12 azi1=str2num(str);
13 str=get(handles.edit6,'string')
14 azi2=str2num(str);
15 %输入角度和距离
16 str=get(handles.edit9,'string')
17 Ang=str2num(str);
18 str=get(handles.edit13,'string')
19 Dis=str2num(str);
20 [fbx,k,x,y]=TranverseFH(X1,Y1,azi1,X2,Y2,azi2,Ang',Dis')
21 str1=sprintf('%8.8f',fbx)
22 set(handles.edit14,'string',str1);
23 str1=sprintf('%8.8f',k)
24 set(handles.edit15,'string',str1);
25 strx=[];
26 for i=1:size(x,2)
27     str1=sprintf('%i', '%.3f', '%.3f\n', i, x(i), y(i));
28     strx=[strx str1];
29 end
30 set(handles.edit17,'string',strx)
```

# 高斯正算

## get\_X模块

这个模块用B,a,b求解X

```
1 function[X]=get_X(B,a,b)
2 e2=sqrt(a^2-b^2)/b;
3 beita0=1-3/4*(e2^2)+45/64*e2^4-175/256*e2^6+11025/16384*e2^8
4 beita2=beita0-1
5 beita4=15/32*e2^4+175/384*e2^6+3675/8164*e2^8
6 beita6=-35/96*e2^6+735/2048*e2^8
7 beita8=315/1024*e2^8
8 c=a^2/b;
9 X=c*(beita0*B+(beita2*cos(B)+beita4*cos(B)^3+beita6*cos(B)^7)*sin(B));
10 end
```

## function 坐标转化模块

```
1 function [x,y]=BL2xy(B,L,L0,n)
2 %l0是中央经线，B是大地纬度，L是大地经度
3 switch n
4     case(1)%WGS84
5         a=6378137.0;
6         b=6356752.3142;
7     case(2)%IUGG75
8         a=6378140.0;
9         b=6356755.2881;
10    case(3)%CGCS2000椭球坐标系,在作业中采用
11        a=6378137.0;
12        b=6356752.3141;
13 end
14 B=dms_rad(B);
15 L=dms_rad(L);
16 L0=dms_rad(L0);
17 X=get_X(B,a,b);
18 e1=sqrt(a^2-b^2)/a; %第一偏心率
```

```

19 N=a/sqrt(1-e1^2*sin(B)^2);
20 l=L-L0;%计算出距离中央经线的距离
21 t=tan(B);
22 e2=sqrt(a^2-b^2)/b;%计算第二偏心率
23 ita=e2*cos(B);
24 x=X+N/2*sin(B)*cos(B)^2+...
25     N/24*sin(B)*cos(B)^3*(5-t^2+9*ita^2+ita^4)*l^4 ...%注意等式和...中有空格
26 +N/720*sin(B)*cos(B)^5*(61-58*t^2+t^4)*l^6;
27 y=N*cos(B)*l+N/6*cos(B)^3*(1-t^2+ita^2)*l^3 ...
28     +N/120*cos(B)^5*(5-18*t^2+t^4+14*ita^2-58*t^2*ita^2)*l^5;
29 end

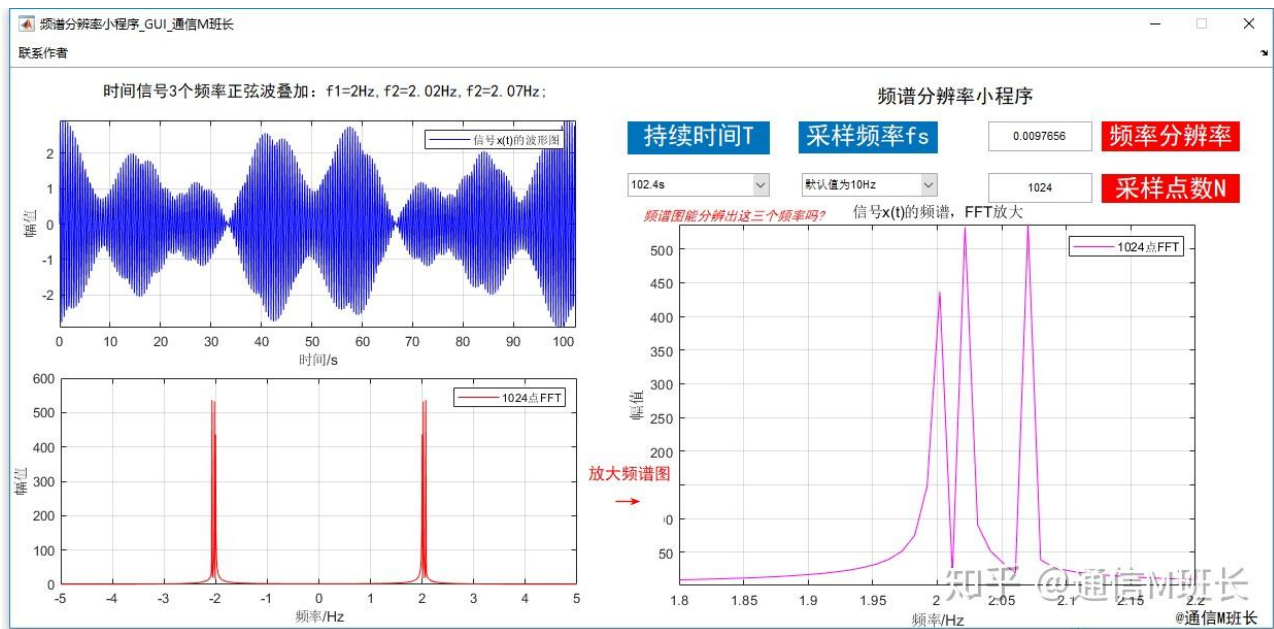
```

高斯正算输入数据

$[x,y]=BL2xy(37.3915,111.5654,114,3)$

## 五. 用户界面与交互

- 将要采用如下图形用户界面标准或产品系列的风格;



- 屏幕布局

大地经度

37.3915

大地纬度

111.5654

中央经线

114

椭球选择

2

计算

横坐标

5391782.42835264

纵坐标

-181055.88561797

椭球选择: (1) WGS84 (2) IUGG75 (3)CGCS2000

## 六. 运行结果

### 1附和导线

Thanverse

已知数据

起点X坐标

203000.348

起点方位角

226.4459

起点Y坐标

-59049.801

终点方位角

324.4603

终点X坐标

203039.503

终点Y坐标

-59796.549

观测数据

角度观测值

230 32 37  
180 00 42  
170 39 22  
236 48 37

边长观测值

204.952  
200.130  
345.153

计算

面板

角度观测值

-572464-

全长相对闭合差

1 / 4000

未知点坐标

37465110237465110292  
11012.031917e+05-5.942  
152e+04

## 附和导线

起点X 203000.348  
-59049.801  
203039.503  
-59796.549

$$\alpha_{AB} = 226^\circ 44' 59'', \alpha_{CD} = 324^\circ 46' 03'';$$

226.4459  
324.4603

230 32 37 204.952  
180 00 42 200.130  
170 39 22 345.153  
236 48 37

输入坐标 → 计算 → 输出坐标 → Gui绘制 → Gui图

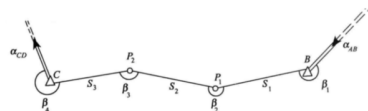


图 1 单一附和导线略图

$$[fbx, k, x, y] = \text{TraverseFH}(XB, YB, azi1, XC, YC, azi2, Ang, Dis)$$

## 2高斯正算

## 高斯正算

B=37.3915  
L=111.5654  
中央子午线 经度为 114 度

$$x = 5391779.90793839$$

$$y = -181055.800288334$$

输入坐标 → 高斯正算 → 输出坐标 → Gui绘制 → 绘制图形框

## 七. 主要变量说明表

### 附和导线近似平差程序【函数 TraverseFH】主要变量说明表

变量名	输入/输出	类型	说明
fbx	输出	数值	角度闭合差，单位换算为秒
k	输出	数值	全长相对闭合差（可能是近似值）的整数部分
x	输出	数组	待定点的 X 坐标数组
y			



	输出	数组	待定点的 Y 坐标数组
<b>XB, YB</b>	输入	数值	已知起始点的 X 和 Y 坐标
<b>XC, YC</b>	输入	数值	已知终点的 X 和 Y 坐标
<b>azi1</b>	输入	数值	已知起始边的方位角（度或弧度），函数内部转换为弧度
<b>azi2</b>	输入	数值	已知终边的方位角（度或弧度），函数内部转换为弧度
<b>Ang</b>	输入	数组	观测的角度数组（假设输入为度，函数内部转换为弧度）
<b>Dis</b>	输入	数组	观测的边长数组
<b>n</b>	内部变量	数值	观测角的个数，通过length(Ang)计算得出
<b>radangle</b>	内部变量	数组	将观测的角度数组从度转换为弧度后的数组
<b>zb</b>	内部变量	数值	观测角之和（弧度）
<b>fb</b>	内部变量	数值	计算出的角度闭合差（弧度），并取 $2\pi$ 的余数标准化
<b>fwj</b>	内部变量	数组	平差后各边的方位角数组（弧度）
<b>dx, dy</b>	内部变量	数组	根据平差后的方位角和边长计算出的坐标增量数组
<b>fx, fy</b>	内部变量	数值	坐标增量的闭合差在 X 和 Y 方向上的分量
<b>fs</b>	内部变量	数值	坐标增量闭合差的长度（直线距离）
<b>Zd</b>	内部变量	数值	所有观测边长的总和

高斯正算程序【函数BL2xy】主要变量说明表

变量名	输入/输出	类型	说明
<b>x, y</b>	输出	数值	投影后的平面直角坐标

			系的 X 和 Y 坐标
B	输入	数值	大地纬度，函数内部将其从度转换为弧度
L	输入	数值	大地经度，函数内部将其从度转换为弧度
L0	输入	数值	中央经线，函数内部将其从度转换为弧度
n	输入	数值	椭球模型选择参数，用于选择不同的椭球体参数 a 和 b
a	内部变量	数值	椭球体的长半轴长度，根据 n 的值选择不同的数值
b	内部变量	数值	椭球体的短半轴长度，根据 n 的值选择不同的数值
e1	内部变量	数值	第一偏心率，根据 a 和 b 计算得出
N	内部变量	数值	卯酉圈曲率半径，根据 B、a 和 e1 计算得出
l	内部变量	数值	经度差，即大地经度 L 与中央经线 L0 的差值（弧度）
t	内部变量	数值	正切值 $\tan(B)$ 的简写，用于后续计算
e2	内部变量	数值	第二偏心率，根据 a 和 b 计算得出
ita	内部变量	数值	由 e2 和 B 计算得出的中间变量，用于后续计算
X	内部变量	数值	由纬度 B、a 和 b 通过 <code>get_X</code> 函数计算得出的中间值

注意：`dms_rad` 函数是将角度从度分秒格式转换为弧度的辅助函数

## 八. 难点预估

1字体不居中



解决方案：将fontunits修改为normalization，然后将fontsize修改为0.7

## 2坐标位数过少的问题



```
1 str1=sprintf('%8.8f',x)
2 set(handles.edit5,'string',str1);
3 str1=sprintf('%8.8f',y) %把位数修改到8位
4 set(handles.edit6,'string',str1);
```

大地经度

37.3915

大地纬度

111.5654

中央经线

114

椭球选择

2

计算

横坐标

5391782.42835264

纵坐标

-181055.88561797

椭球选择:

(1) WGS84 (2) IUGG75 (3)CGCS2000

九、主要程序运行截图

```
编辑器 - C:\Users\sand\Documents\MATLAB\sy4\TranverseFH.m
get_X.m x BL2xy.m x TranverseFH.m x untitled.m x Thanv
1 function [fbx,k,x,y] = TranverseFH(XB,YB,azi1,XC,YC,azi2,Ang,Dis)
2 %% 解算附和导线
3 % x,y 返回待定点坐标, fwj 平差后各边的方位角
4 % fbx 返回角度闭合差,k全长相对闭合差
5 % XB,YB,XC,YC 已知点坐标x
6 % azi1和azi2已知起始边和终边方位角
7 % Ang,Dis是观测的角度(弧度)和边长
8 %数据输入和处理
9 n=length(Ang); %获得观测角个数
10 [radangle]=dms_rad(Ang);% 角度转换为弧度
11 zb=sum(radangle); %观测角之和
12 azi1=dms_rad(azi1);
13 azi2=dms_rad(azi2);
14 fb=zb-n*pi-azi2+azi1; %计算角度闭合差
15 fb=rem(fb,2*pi); %取余数
16 fbx=round(fb*206264.80); %将角度闭合差换成秒
17
18 radangle=radangle-fb./n;
19 for i=1:n
20     if i==1
21         fwj(i) = azi1-pi+radangle(i);
22     else
23         fwj(i)=fwj(i-1)-pi+radangle(i);
24     end
25 end
26 dx=cos(fwj(1:n-1)).*Dis;
27 dy=sin(fwj(1:n-1)).*Dis;
28 fx=sum(dx)+XB-XC;
29 fy=sum(dy)+YB-YC;
30 fs=sqrt(fx^2+fy^2);
31 Zd=sum(Dis);
32 k=fix(Zd/fs);
33 dx=dx-(fx/Zd).*Dis;%坐标增量闭合差的分配
34 dy=dy-(fy/Zd).*Dis;
35 for i=1:n-1
36     if i==1
37         x(i)=XB+dx(i);
38         y(i)=YB+dy(i);
39     else
40         x(i)=x(i-1)+dx(i);
41         y(i)=y(i-1)+dy(i);
42     end
43 end
```

```

get_X.m  BL2xy.m  TranverseFH.m  untitled.m  Thanverse.m  main.m
1  function [x,y]=BL2xy(B,L,L0,n)
2  %L0是中央经线, B是大地纬度, L是大地经度
3
4  switch n
5      case(1)%WGS84
6          a=6378137.0;
7          b=6356752.3142;
8      case(2)%IUGG75
9          a=6378140.0;
10         b=6356755.2881;
11     case(3)%CGCS2000椭球坐标系,在作业中采用
12         a=6378137.0;
13         b=6356752.3141;
14     end
15
16     B=dms_rad(B);
17     L=dms_rad(L);
18     L0=dms_rad(L0);
19     X=get_X(B,a,b);
20     e1=sqrt(a^2-b^2)/a; %计算第一偏心率
21     N=a/sqrt(1-e1^2*sin(B)^2);
22     l=L-L0;%计算出距离中央经线的距离
23     t=tan(B);
24     e2=sqrt(a^2-b^2)/b;%计算第二偏心率
25     ita=e2*cos(B);
26     x=X+N/2*sin(B)*cos(B)^2+...
27         N/24*sin(B)*cos(B)^3*(5-t^2+9*ita^2+ita^4)*1^4 ...%注意等式和...中有空格
28     +N/720*sin(B)*cos(B)^5*(61-58*t^2+t^4)*1^6;
29
30     y=N*cos(B)*l+N/6*cos(B)^3*(1-t^2+ita^2)*1^3 ...
31         +N/120*cos(B)^5*(5-18*t^2+t^4+14*ita^2-58*t^2*ita^2)*1^5;
32
33     end

```

```

get_X.m x BL2xy.m x TranverseFH.m x untitled.m x Thanverse.n
255 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'default
256     set(hObject,'BackgroundColor','white');
257 end
258
259
260 % --- Executes on button press in pushbutton1.
261 function pushbutton1_Callback(hObject, eventdata, handles)
262 %pushbutton1代表是"计算", 有关计算的代码都在此按钮下面
263 str=get(handles.edit1,'string');
264 X1=str2num(str);
265 str=get(handles.edit2,'string')
266 Y1=str2num(str);
267
268 str=get(handles.edit3,'string')
269 X2=str2num(str);
270 str=get(handles.edit4,'string')
271 Y2=str2num(str);
272
273 str=get(handles.edit5,'string')
274 azi1=str2num(str);
275 str=get(handles.edit6,'string')
276 azi2=str2num(str);
277
278 %输入角度和距离
279 str=get(handles.edit9,'string')
280 Ang=str2num(str);
281 str=get(handles.edit13,'string')
282 Dis=str2num(str);
283 [fbx,k,x,y]=TranverseFH(X1,Y1,azi1,X2,Y2,azi2,Ang',Dis')
284
285 str1=sprintf('%8.8f',fbx)
286 set(handles.edit14,'string',str1);
287
288 str1=sprintf('%8.8f',k)
289 set(handles.edit15,'string',str1);
290 strx=[];
291 for i=1:size(x,2)
292     str1=sprintf('%i', '%.3f', '%.3f\n', i, x(i), y(i));
293     strx=[strx str1];

```

```
get_X.m x BL2xy.m x TranverseFH.m x untitled.m x Thanverse.m x
1 function[X]=get_X(B,a,b)
2
3 e2=sqrt(a^2-b^2)/b;
4 beita0=1-3/4*(e2^2)+45/64*e2^4-175/256*e2^6+11025/16384*e2^8
5 beita2=beita0-1
6 beita4=15/32*e2^4+175/384*e2^6+3675/8164*e2^8
7 beita6=-35/96*e2^6+735/2048*e2^8
8 beita8=315/1024*e2^8
9 c=a^2/b;
10 X=c*(beita0*B+(beita2*cos(B)+beita4*cos(B)^3+beita6*cos(B)^7)*sin(B));
11 end
12 %[X]=get_X(0.6571928776,6378137,6356752.3142)
13 %format long g 切换输出
```