



# 《大地测量学基础》

## 实 习 报 告

学    院： 地理空间信息学院  
专    业： 测量工程  
班    级： 大本五十八班  
姓    名： 郑培智  
学    号： 3392010051  
辅导教员： 吕志平  
时    间： 2012年12月1日

## 编程实习（三）：高斯投影正反算及邻带换算

### 一、计算步骤

#### （1）高斯投影正解计算

1、计算子午线弧长：

$$X = 111134.8611B - (32005.7799 \sin B + 133.9238 \sin^3 B + 0.6973 \sin^5 B + 0.0039 \sin^7 B) \cos B$$

2、计算方程参数：

$$N = \frac{C}{\sqrt{1 + e'^2 \cos^2 B}}$$

$$t = \tan B$$

$$\eta = e' \cos B$$

$$m = \cos B \cdot l$$

3、计算最终坐标：

$$x = X + N \cdot t \left[ \frac{1}{2} m^2 + \frac{1}{24} (5 - t^2 + 9\eta^2 + 4\eta^4) + \frac{1}{720} (61 - 58t^2 + t^4) m^6 \right]$$

$$y = N \left[ m + \frac{1}{6} (1 - t^2 + \eta^2) m^3 + \frac{1}{120} (5 - 18t^2 + t^4 + 14\eta^2 - 58\eta^2 t^2) m^5 \right]$$

#### （2）高斯投影反解计算

$$X = 111134.8611B - (32005.7799 \sin B + 133.9238 \sin^3 B + 0.6973 \sin^5 B + 0.0039 \sin^7 B) \cos B$$

利用迭代法求 $\mathbf{B}_f$

$$\text{初始值: } B_f^{(1)} = \frac{X}{11134.8611}$$

$$\text{迭代式: } B_f^{(i+1)} = \frac{X - F(B_f^{(i)})}{11134.8611}$$

$$\text{其中: } F(B_f^{(i)}) = -(32005.7799 \sin B_f^{(i)} + 133.9238 \sin^3 B_f^{(i)} + 0.6973 \sin^5 B_f^{(i)} + 0.0039 \sin^7 B_f^{(i)}) \cos B_f^{(i)}$$

$$\left. \begin{aligned} B^\circ &= B_f^\circ - \frac{1}{2} V_f^2 t_f \left[ \left( \frac{y}{N_f} \right)^2 - \frac{1}{12} (5 + 3t_f^2 + \eta_f^2 - 9\eta_f^2 t_f^2) \left( \frac{y}{N_f} \right)^4 \right] \\ &\quad + \frac{1}{360} (61 + 90t_f^2 + 45t_f^4) \left( \frac{y}{N_f} \right)^6 \left] \frac{180}{\pi} \right. \\ l^\circ &= \frac{1}{\cos B_f} \left[ \left( \frac{y}{N_f} \right) - \frac{1}{6} (1 + 2t_f^2 + \eta_f^2) \left( \frac{y}{N_f} \right)^3 \right. \\ &\quad \left. + \frac{1}{120} (5 + 28t_f^2 + 24t_f^4 + 6\eta_f^2 + 8\eta_f^2 t_f^2) \left( \frac{y}{N_f} \right)^5 \right] \frac{180}{\pi} \end{aligned} \right\}$$

最后可得:  $\mathbf{B} = \mathbf{B}^0$

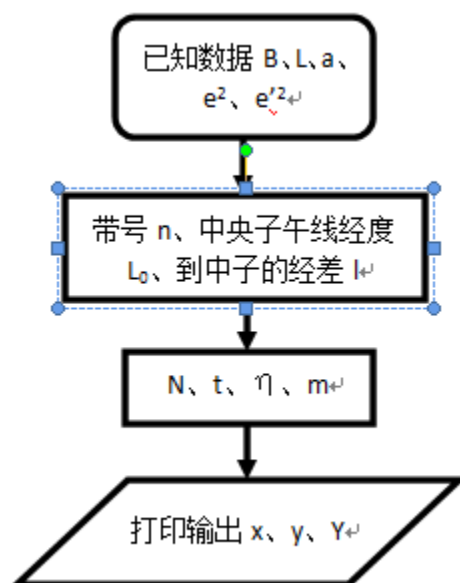
$$\mathbf{L} = \mathbf{L}_0 + \mathbf{l}^\circ$$

### (3) 高斯坐标的邻带换算

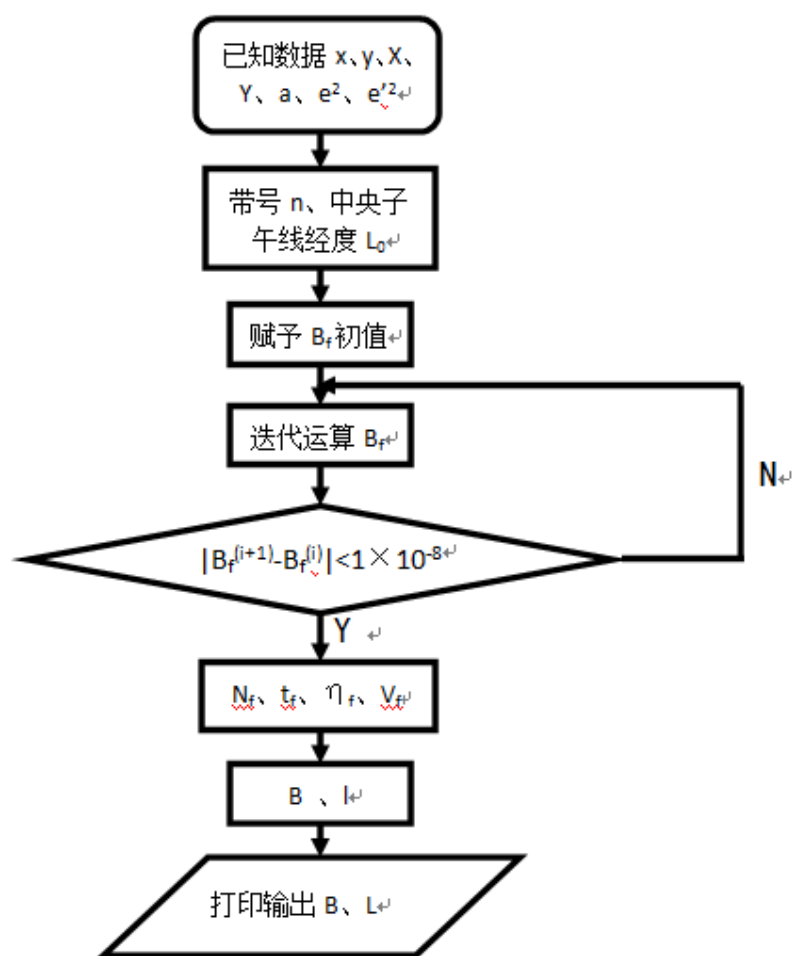
运用到高斯投影的正反算，公式与之前相同。

## 2、 程序框图

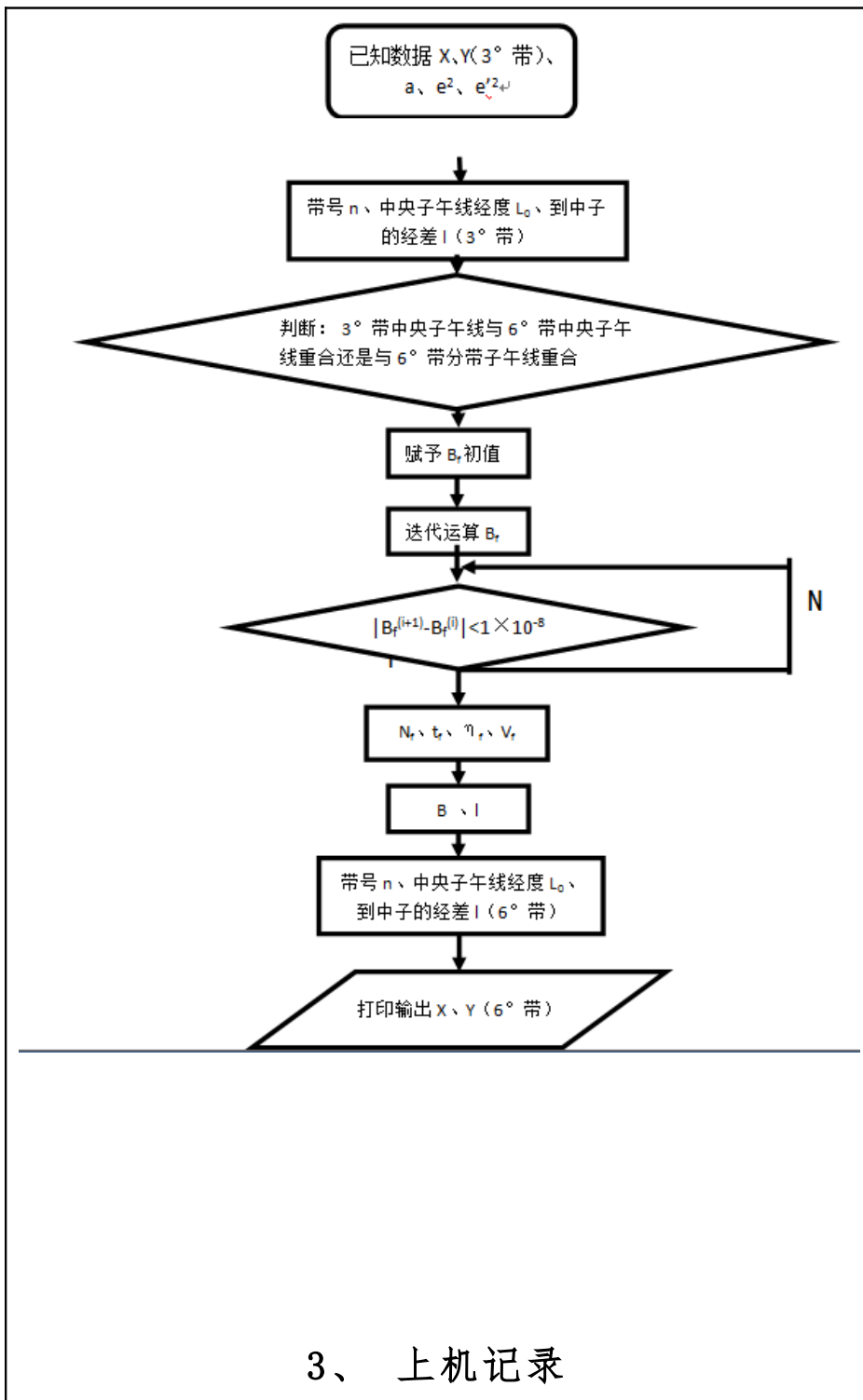
### （一）高斯投影正解计算



### （二）高斯投影反解计算



### (三) 高斯坐标的邻带换算



| 上机次数 | 时间及内容                                                                                                                                                                        |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 第一次  | <p>2012年11月21日14:30——16:10</p> <p>仔细翻阅课本，熟悉了高斯投影正反算的公式以及如何进行邻带换算，进行了简单的正算编程，把文件里的数据用数组的形式输入，并检查了它的正确性。</p>                                                                   |
| 第二次  | <p>2012年11月26日14:30——16:10</p> <p>继续进行高斯投影的正算，发现其中的变量太多，于是简化了公式，三个椭球参考系采用了一个通用公式，大大减少了程序的复杂性。最后结果正确，但是无法按照书上的格式输出。</p>                                                       |
| 第三次  | <p>2012年11月26日19:30——21:30</p> <p>进行了高斯投影反算的编程，由于有了正算的基础，反解的输入很快，只是在求B0的时候，循环迭代出了点问题，结果差别很大，于是向会的同学请教，顺利解决了问题。</p>                                                           |
| 第四次  | <p>2012年11月28日14:30——16:10</p> <p>初步开始邻带换算的编程，发现这个程序需要高斯投影正反算的基础，于是将所有的变量进行了整理，方便在邻带换算的时候使用。但是本次编程并没有太多进展。</p>                                                               |
| 第五次  | <p>2012年11月28日20:30——21:30</p> <p>开始进行邻带换算的编程，但是发现其中牵扯到许多椭球参数，于是我先将其统一为克拉索夫斯基椭球参数进行运算，结果正确。</p>                                                                              |
| 第六次  | <p>2012年12月1日18:30——21:15</p> <p>接着上次的计算，虽然我看到自己在同一个椭球参数下运算正确，但是我运用到高斯投影正反算的程序时，发现结果不对，不知道哪个数据时那种椭球参数的答案，最后和班里的同学对了答案，才渐渐地清楚怎样输出正确的答案。最后整理了程序，删去了不必要的程序段，简化了一些程序段，输出了文件。</p> |

|  |  |
|--|--|
|  |  |
|--|--|

## 4、源程序及结果

### （一）高斯投影正解计算

```
// 高斯投影正算.cpp : Defines the entry point for the console application.
//

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
//*****
//*****子函数*****
//*****
//将度分秒连写的角度（double 型）化为弧度值
#include <math.h>
double dms_rad(double a)
{
    //提取角度值的符号
    double sign=(a<0.0)?-1.0:1.0;
    a=fabs(a);

    //提取角度值的整数
    int d=(int)((a+0.00001)/10000.0);
    a=a-d*10000.0;
    if(a<0.0){d=d-1;a=a+10000;}

    //提取角度值的整分及秒值
    int m=(int)((a+0.00001)/100.0);
    a=a-m*100;
    if(a<0.0){m=m-1;a=a+100.0;}
```



```

a=sign*(d*3600.0+m*60.0+a)/206264.806247096363;

return (a);
}

//将角度的弧度值化为度分秒连写的角度(double型)
#include <stdio.h>
#include<stdlib.h>
#include <math.h>

double rad_dms(double a)
{
    a=a*206264.806247096363;

    double sign=(a<0.0)?-1.0:1.0;
    a=fabs(a);

    int d=(int)(a/3600.0+0.0000001);
    a=a-d*3600.0;
    if(a<0.0){d=d-1;a=a+3600.0;}

    int m=(int)(a/60.0+0.0001);
    a=a-m*60.0;
    if(a<0.0){m=m-1;a=a+60.0;}

    a=d*10000.0+m*100.0+a;

    return a*sign;
}

//*****
//*****主函数*****
//*****

int main(int argc, char* argv[])
{FILE*fp;
fp=fopen("已知数据.txt","r");    //导入所要转换的L,B文件
int n;
if(fp==NULL)

```

```

{
    printf("cannot open the file\n");
    exit(0);
}

fscanf(fp, "%d", &n);
double *b=new double [n];
double *L=new double [n];
int *l=new int [n];
int *ll=new int [n];
int *lll=new int [n];
int *lll=new int [n];
for(int j=0; j<n; j++)
{
    fscanf(fp, "%lf%lf", b+j, L+j);
    //printf("%lf\n", L[j]);
}

fclose(fp);

fp=fopen("椭圆参数.txt", "r");    //导入坐标系参数文件
if(fp==NULL)
{
    printf("cannot open the file\n");
    exit(0);
}

double a[3], e2[3], e12[3];
for (int i=0; i<3; i++)
{
    fscanf(fp, "%lf%lf%lf", &a[i], &e2[i], &e12[i]);
}

fclose(fp);
//printf("%lf\n", e2[2]);

double *B=new double [n];
for(i=0; i<n; i++)

```

```

{
    B[i]=dms_rad(b[i]);
    l[i]=floor(L[i]/10000);
    ll[i]=floor(l[i]/6)+1;//6 度带号
    ll[i]=floor((l[i]-1.5)/3+1);//3 度带号
    ll1[i]=ll[i]*3;
    //printf("%d\n",ll1[i]);
}

double A1[3],B1[3],C1[3],D1[3];
double *X=new double [3*n];
double *yita=new double [3*n];
double *yita2=new double [3*n];
double *yita4=new double [3*n];
double *t=new double [3*n];
double *t2=new double [3*n];
double *t4=new double [3*n];
double *m=new double [3*n];
double *m2=new double [3*n];
double *m4=new double [3*n];
double *N=new double [3*n];
double *x=new double [3*n];
double *y=new double [3*n];
for(i=0;i<3;i++)
{
    A1[i]=1+3*e2[i]/4+45*e2[i]*e2[i]/64+175*e2[i]*e2[i]*e2[i]/256;
    B1[i]= 3*e2[i]/4+15*e2[i]*e2[i]/16+525*e2[i]*e2[i]*e2[i]/512;
    C1[i]= 15*e2[i]*e2[i]/64+105*e2[i]*e2[i]*e2[i]/256;
    D1[i]= 35*e2[i]*e2[i]*e2[i]/521;
    //printf("%lf\n",B1[i]);
    for(int j=0;j<n;j++)
    {
        X[3*j+i]=a[i]*(1-e2[i])*(A1[i]*B[j]-B1[i]*sin(2*B[j])/2+C1[i]*sin(4*B[j])/4-
        D1[i]*sin(6*B[j])/6);
        yita[3*j+i]=sqrt(e12[i])*cos(B[j]);
        t[3*j+i]=tan(B[j]);
        m[3*j+i]=cos(B[j])*(dms_rad(L[j])-dms_rad((ll[j]*6-3)*10000));
    }
}

```

```

        //printf("%d\n",l1[j]);

        N[3*j+i]=a[i]/sqrt(1-e2[i]*sin(B[j])*sin(B[j]));

        yita2[3*j+i]=yita[3*j+i]*yita[3*j+i];

        yita4[3*j+i]=yita2[3*j+i]*yita2[3*j+i];

        t2[3*j+i]=t[3*j+i]*t[3*j+i];

        t4[3*j+i]=t2[3*j+i]*t2[3*j+i];

        m2[3*j+i]=m[3*j+i]*m[3*j+i];

        m4[3*j+i]=m2[3*j+i]*m2[3*j+i];

        x[3*j+i]=X[3*j+i]+N[3*j+i]*t[3*j+i]*(m2[3*j+i]/2+(5-t2[3*j+i]+9*yita2[3*j+i]
+4*yita4[3*j+i])*m4[3*j+i]/24+

                (61-58*t2[3*j+i]+t4[3*j+i])*m2[3*j+i]*m4[3*j+i]/720);

        y[3*j+i]=N[3*j+i]*(m[3*j+i]+(1-t2[3*j+i]
+yita2[3*j+i])*m2[3*j+i]*m[3*j+i]/6+(5-18*t2[3*j+i]+t4[3*j+i]+14*

                yita2[3*j+i]-58*yita2[3*j+i]*t2[3*j+i])*m4[3*j+i]*m[3*j+i]/120);

        printf("%1f\n",y[3*j+i]+21500000);

    }

}

double *m1=new double [3*n];
double *m12=new double [3*n];
double *m14=new double [3*n];
double *x1=new double [3*n];
double *y1=new double [3*n];

for(i=0;i<3;i++)
{
    A1[i]=1+3*e2[i]/4+45*e2[i]*e2[i]/64+175*e2[i]*e2[i]*e2[i]/256;
    B1[i]= 3*e2[i]/4+15*e2[i]*e2[i]/16+525*e2[i]*e2[i]*e2[i]/512;
    C1[i]=
                15*e2[i]*e2[i]/64+105*e2[i]*e2[i]*e2[i]/256;
    D1[i]=
                35*e2[i]*e2[i]*e2[i]/521;

    //printf("%1f\n",B1[i]);

    for(int j=0;j<n;j++)
    {
        X[3*j+i]=a[i]*(1-e2[i])*(A1[i]*B[j]-B1[i]*sin(2*B[j]))/2+C1[i]*sin(4*B[j])/4-
D1[i]*sin(6*B[j])/6);

        yita[3*j+i]=sqrt(e12[i])*cos(B[j]);

        t[3*j+i]=tan(B[j]);

        m1[3*j+i]=cos(B[j])*(dms_rad(L[j])-dms_rad(l11[j]*10000));

```

```

        //printf("%d\n",l1[j]);
        N[3*j+i]=a[i]/sqrt(1-e2[i]*sin(B[j])*sin(B[j]));
        yita2[3*j+i]=yita[3*j+i]*yita[3*j+i];
        yita4[3*j+i]=yita2[3*j+i]*yita2[3*j+i];
        t2[3*j+i]=t[3*j+i]*t[3*j+i];
        t4[3*j+i]=t2[3*j+i]*t2[3*j+i];
        m12[3*j+i]=m1[3*j+i]*m1[3*j+i];
        m14[3*j+i]=m12[3*j+i]*m12[3*j+i];
        x1[3*j+i]=X[3*j+i]+N[3*j+i]*t[3*j+i]*(m12[3*j+i]/2+(5-t2[3*j+i]
+9*yita2[3*j+i]+4*yita4[3*j+i])*m14[3*j+i]/24+
        (61-58*t2[3*j+i]+t4[3*j+i])*m12[3*j+i]*m14[3*j+i]/720);
        y1[3*j+i]=N[3*j+i]*(m1[3*j+i]+(1-t2[3*j+i]
+yita2[3*j+i])*m12[3*j+i]*m1[3*j+i]/6+(5-18*t2[3*j+i]+t4[3*j+i]+14*
        yita2[3*j+i]-58*yita2[3*j+i]*t2[3*j+i])*m14[3*j+i]*m1[3*j+i]/120);
        //printf("%lf\n",x1[3*j+i]);
    }
}
//输出未知量
fp=fopen("运算结果.txt","w");
if(fp==NULL)
{ printf("can not open the file\n");
  exit(0);
}
fprintf(fp,"*****高斯投影正算*****\n");
fprintf(fp,"6° 带\n");
fprintf(fp,"已知数据: \n" );
        fprintf(fp,"B==%lf\tB==%lf\nL==%lf\tL==%lf\n",b[0],b[1],L[0],L[1]);
        fprintf(fp,"在克拉索夫斯基椭球中: \n" );
                fprintf(fp,"x==%15.8lf\tx==%15.8lf\ny==%15.8lf\ty==%15.8lf\nY==%d
%15.8lf\ty==%d%15.8lf\n",x[0],x[3],y[0],y[3],l1[0],y[0]+500000,l1[1],y[3]+500000);

        fprintf(fp,"在 IUGG1975 椭球中: \n" );
                fprintf(fp,"x==%15.8lf\tx==%15.8lf\ny==%15.8lf\ty==%15.8lf\nY==%d
%15.8lf\ty==%d%15.8lf\n",x[1],x[4],y[1],y[4],l1[0],y[1]+500000,l1[1],y[4]+500000);

        fprintf(fp,"在 CGCS2000 椭球中: \n" );

```

```

        fprintf(fp, "x==%15.81f\tx==%15.81f\nty==%15.81f\ty==%15.81f\nY==%d%15.81f\ty==%d%15.81f\n", x[2], x[5], y[2], y[5], l1[0], y[2]+500000, l1[1], y[5]+500000);

        fprintf(fp, "3° 带\n");

        fprintf(fp, "已知数据: \n" );

        fprintf(fp, "B=%1f\t\tB=%1f\nL=%1f\tL=%1f\n", b[0], b[1], L[0], L[1]);

        fprintf(fp, "在克拉索夫斯基椭球中: \n" );

        fprintf(fp, "x==%15.81f\tx==%15.81f\nty==%15.81f\ty==%15.81f\nY==%d%15.81f\ty==%d%15.81f\n", x1[0], x1[3], y1[0], y1[3], l1[0], y1[0]+500000, l1[1], y1[3]+500000);

        fprintf(fp, "在 IUGG1975 椭球中: \n" );

        fprintf(fp, "x==%15.81f\tx==%15.81f\nty==%15.81f\ty==%15.81f\nY==%d%15.81f\ty==%d%15.81f\n", x1[1], x1[4], y1[1], y1[4], l1[0], y1[1]+500000, l1[1], y1[4]+500000);

        fprintf(fp, "在 CGCS2000 椭球中: \n" );

        fprintf(fp, "x==%15.81f\tx==%15.81f\nty==%15.81f\ty==%15.81f\nY==%d%15.81f\ty==%d%15.81f\n", x1[2], x1[5], y1[2], y1[5], l1[0], y1[2]+500000, l1[1], y1[5]+500000);

delete

m, m1, m2, m12, m4, m14, x, x1, y, y1, t, t2, t4, N, X, yita, yita2, yita4, b, B, l, L, l1, l11, l111;

        return 0;
}

```

用文件输出的结果（度分秒为连写形式）：

\*\*\*\*\*高斯投影正算\*\*\*\*\*

6°带

已知数据:

B==405832.330000

B==352640.380000

L==1001020.110000

L==1150851.220000

在克拉索夫斯基椭球中:

x==4538610.95075792

x==3925560.03445743

y== 98666.62571306

y== -168198.57799644

Y==17598666.62571306

Y==20331801.42200356

在IUGG1975椭球中:

x==4538532.84615559

x==3925492.27711263

y== 98665.02193939

y== -168195.83639686

Y==17598665.02193939

Y==20331804.16360314

在CGCS2000椭球中:

x==4538530.72893427

x==3925490.44692123

y== 98664.97542576

y== -168195.75714371

Y==17598664.97542576

Y==20331804.24285629

3°带

已知数据:

B=405832.330000

B=352640.380000

L=1001020.110000

L=1150851.220000

在克拉索夫斯基椭球中:

x==4538610.95075792

x==3924588.05386890

y== 98666.62571306

y==104193.07502019

Y==33598666.62571306

Y==38604193.07502019

在IUGG1975椭球中:

x==4538532.84615559

x==3924520.31236680

y== 98665.02193939

y==104191.37669133

Y==33598665.02193939

Y==38604191.37669133

在CGCS2000椭球中:

x==4538530.72893427

x==3924518.48263338

y== 98664.97542576

y==104191.32759683

Y==33598664.97542576

Y==38604191.32759683

## (二) 高斯投影反解计算

```
// 高斯投影反算.cpp : Defines the entry point for the console application.
```

```
//
```

```
#include<stdio.h>
```

```
#include<math.h>
```

```
#include<stdlib.h>
```

```
*****
```

```
*****子函数*****
```

```
*****
```

```
//将度分秒连写的角度(double型)化为弧度值
```

```

#include <math.h>

double dms_rad(double a)
{
    //提取角度值的符号
    double sign=(a<0.0)?-1.0:1.0;
    a=fabs(a);

    //提取角度值的整数
    int d=(int)((a+0.00001)/10000.0);
    a=a-d*10000.0;
    if(a<0.0){d=d-1;a=a+10000;}

    //提取角度值的整分及秒值
    int m=(int)((a+0.00001)/100.0);
    a=a-m*100;
    if(a<0.0){m=m-1;a=a+100.0;}
    a=sign*(d*3600.0+m*60.0+a)/206264.806247096363;

    return (a);
}

//将角度的弧度值化为度分秒连写的角度(double型)
#include <stdio.h>
#include<stdlib.h>
#include <math.h>

double rad_dms(double a)
{
    a=a*206264.806247096363;

    double sign=(a<0.0)?-1.0:1.0;
    a=fabs(a);

    int d=(int)(a/3600.0+0.0000001);
    a=a-d*3600.0;
    if(a<0.0){d=d-1;a=a+3600.0;}

```



```

        int m=(int) (a/60.0+0.0001);

        a=a-m*60.0;

        if(a<0.0) {m=m-1;a=a+60.0;}


        a=d*10000.0+m*100.0+a;


        return a*sign;
}

//*****
//*****主函数*****
//*****

int main(int argc, char* argv[])
{FILE*fp;

fp=fopen("已知数据.txt","r");    //导入所要转换的x,y,X,Y文件
int n;
if(fp==NULL)
{
    printf("cannot open the file\n");
    exit(0);
}

fscanf(fp,"%d",&n);
double *x=new double [n];
double *y=new double [n];
double *X=new double [n];
double *Y=new double [n];
for(int j=0;j<n;j++)
{
    fscanf(fp,"%lf%lf%lf%lf",x+j,y+j,X+j,Y+j);
    //printf("%lf\n",y[j]);
}

fclose(fp);


fp=fopen("椭圆参数.txt","r");    //导入坐标系参数文件
if(fp==NULL)
{

```

```

        printf("cannot open the file\n");
        exit(0);
    }

    double a[3],e2[3],e12[3];
    for (int i=0;i<3;i++)
    {
        fscanf(fp,"%lf%lf%lf",&a[i],&e2[i],&e12[i]);
    }
    fclose(fp);
    double A1[3],B1[3],C1[3],D1[3];
    double *B=new double [3*n];
    double *L6=new double [3*n];
    double *L3=new double [3*n];
    double *b=new double [3*n];
    double *b1=new double [3*n];
    double *B0=new double [3*n];
    double *yita=new double [3*n];
    double *yita2=new double [3*n];
    double *t=new double [3*n];
    double *t2=new double [3*n];
    double *t4=new double [3*n];
    double *V=new double [3*n];
    double *N=new double [3*n];

    for(i=0;i<3;i++)
    {
        A1[i]=1+3*e2[i]/4+45*e2[i]*e2[i]/64+175*e2[i]*e2[i]*e2[i]/256;
        B1[i]= 3*e2[i]/4+15*e2[i]*e2[i]/16+525*e2[i]*e2[i]*e2[i]/512;
        C1[i]=
            15*e2[i]*e2[i]/64+105*e2[i]*e2[i]*e2[i]/256;
        D1[i]=
            35*e2[i]*e2[i]*e2[i]/512;

        //printf("%lf\n",B1[i]);
        for(int j=0;j<n;j++)
        {
            b[3*j+i]=x[j]/(a[i]*(1-e2[i])*A1[i]);

```

```

do
{
    b1[3*j+i]=b[3*j+i];

    B0[3*j+i]=x[j]/(a[i]*(1-e2[i])*A1[i])+(B1[i]*sin(2*b[3*j+i])/2-
C1[i]*sin(4*b[3*j+i])/4+D1[i]*sin(6*b[3*j+i])/6)/A1[i];

    b[3*j+i]=B0[3*j+i];

    } while(fabs(B0[3*j+i]-b1[3*j+i])>dms_rad(0.00000000001));

    yita[3*j+i]=sqrt(e12[i])*cos(B0[3*j+i]);
    yita2[3*j+i]=yita[3*j+i]*yita[3*j+i];
    t[3*j+i]=tan(B0[3*j+i]);
    t2[3*j+i]=t[3*j+i]*t[3*j+i];
    t4[3*j+i]=t2[3*j+i]*t2[3*j+i];
    //printf("%lf\n",B0[3*j+i]);
    N[3*j+i]=a[i]/sqrt(1-e2[i]*sin(B0[3*j+i])*sin(B0[3*j+i]));
    V[3*j+i]=sqrt(1+e12[i]*cos(B0[3*j+i])*cos(B0[3*j+i]));

    B[3*j+i]=rad_dms(B0[3*j+i]-
V[3*j+i]*V[3*j+i]*t[3*j+i]*((y[j]*y[j]/N[3*j+i]/N[3*j+i])-(5+3*t2[3*j+i]+yita2[3*j+i]-
9*yita2[3*j+i]*t2[3*j+i])/12
        *(y[j]*y[j]/N[3*j+i]/N[3*j+i])*(y[j]*y[j]/N[3*j+i]/N[3*j+i])
+ (61+90*t2[3*j+i]+45*t2[3*j+i])/360*(y[j]*y[j]/N[3*j+i]/N[3*j+i])
        *(y[j]*y[j]/N[3*j+i]/N[3*j+i])*(y[j]*y[j]/N[3*j+i]/N[3*j+i]))/2);
    L6[3*j+i]=rad_dms(dms_rad(1170000)+((y[j]/N[3*j+i])-(1+2*t2[3*j+i]
+yita2[3*j+i])*(y[j]*y[j]*y[j]/N[3*j+i]/N[3*j+i]/N[3*j+i])/6+(5+28*t2[3*j+i]
+24*t4[3*j+i]
        +6*yita2[3*j+i]
+8*yita2[3*j+i]*t2[3*j+i])*(y[j]*y[j]*y[j]*y[j]*y[j]/N[3*j+i]/N[3*j+i]/N[3*j+i]/N[3*j+i]
/N[3*j+i])/120)/cos(B0[3*j+i]));
    L3[3*j+i]=rad_dms(dms_rad(600000)+((y[j]/N[3*j+i])-(1+2*t2[3*j+i]
+yita2[3*j+i])*(y[j]*y[j]*y[j]/N[3*j+i]/N[3*j+i]/N[3*j+i])/6+(5+28*t2[3*j+i]
+24*t4[3*j+i]
        +6*yita2[3*j+i]
+8*yita2[3*j+i]*t2[3*j+i])*(y[j]*y[j]*y[j]*y[j]*y[j]/N[3*j+i]/N[3*j+i]/N[3*j+i]/N[3*j+i]
/N[3*j+i])/120)/cos(B0[3*j+i]));
    printf("%lf\n",L3[3*j+i]);
}

```

```

}

//输出未知量
fp=fopen("运算结果.txt","w");

if(fp==NULL)
{
printf("can not open the file\n");
exit(0);
}

fprintf(fp, "*****高斯投影反算*****\n");
fprintf(fp, "6° 带\n");
fprintf(fp, "已知数据: \n" );

fprintf(fp, "x==%lf\tx==%lf\ty==%lf\t\tty==%lf\nX==%lf\tx==%lf\ty==%lf\ty==%lf\n", x[0], x[1], y[0], y[1], X[0], X[1], Y[0], Y[1]);

fprintf(fp, "在克拉索夫斯基椭球中: \n" );

fprintf(fp, "B==%15.81f\txB==%15.81f\txL==%15.81f\txL==%15.81f\n", B[0], B[3], L6[0], L6[3]);

fprintf(fp, "在 IUGG1975 椭球中: \n" );

fprintf(fp, "B==%15.81f\txB==%15.81f\txL==%15.81f\txL==%15.81f\n", B[1], B[4], L6[1], L6[4]);

fprintf(fp, "在 CGCS2000 椭球中: \n" );

fprintf(fp, "B==%15.81f\txB==%15.81f\txL==%15.81f\txL==%15.81f\n", B[2], B[5], L6[2], L6[5]);

fprintf(fp, "3° 带\n");
fprintf(fp, "已知数据: \n" );

fprintf(fp, "x==%lf\tx==%lf\ty==%lf\t\tty==%lf\nX==%lf\tx==%lf\ty==%lf\ty==%lf\n", x[0], x[1], y[0], y[1], X[0], X[1], Y[0], Y[1]);

fprintf(fp, "在克拉索夫斯基椭球中: \n" );

fprintf(fp, "B==%15.81f\txB==%15.81f\txL==%15.81f\txL==%15.81f\n", B[0], B[3], L3[0], L3[3]);

fprintf(fp, "在 IUGG1975 椭球中: \n" );

fprintf(fp, "B==%15.81f\txB==%15.81f\txL==%15.81f\txL==%15.81f\n", B[1], B[4], L3[1], L3[4]);

```

```

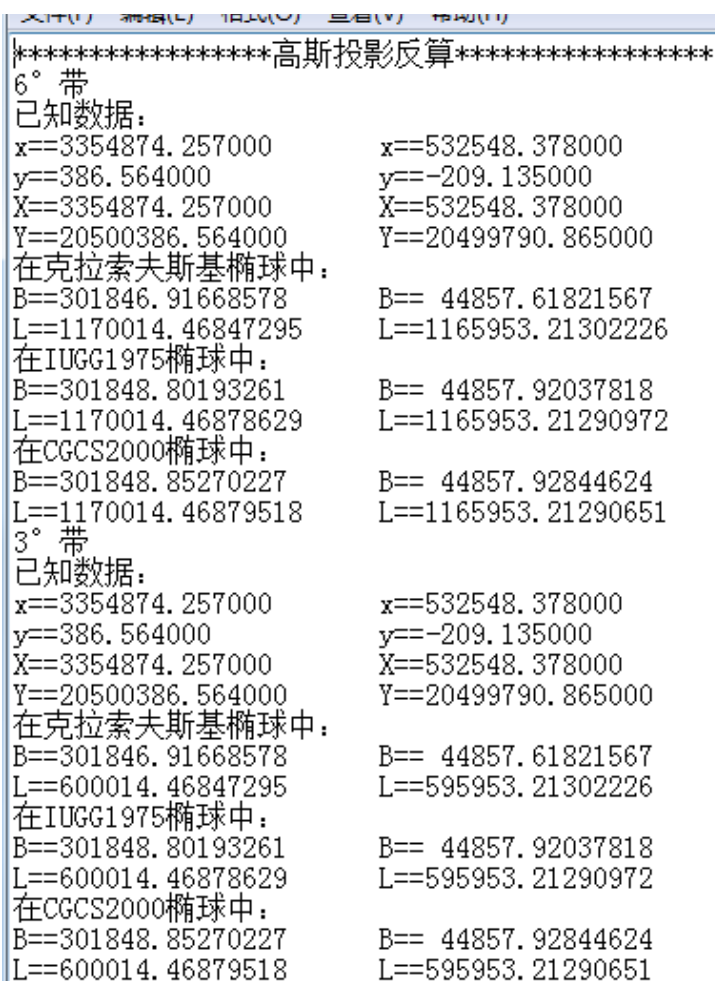
    fprintf(fp, "在 CGCS2000 椭球中: \n" );

    fprintf(fp, "B==%15.81f\tB==%15.81f\nL==%15.81f\tL==%15.81f\n", B[2], B[5], L3[2], L3[5]);

    return 0;
}

```

用文件输出的结果：



```

*****高斯投影反算*****
6° 带
已知数据:
x==3354874.257000      x==532548.378000
y==386.564000          y== -209.135000
X==3354874.257000      X==532548.378000
Y==20500386.564000     Y==20499790.865000
在克拉索夫斯基椭球中:
B==301846.91668578     B== 44857.61821567
L==1170014.46847295    L==1165953.21302226
在IUGG1975椭球中:
B==301848.80193261     B== 44857.92037818
L==1170014.46878629    L==1165953.21290972
在CGCS2000椭球中:
B==301848.85270227     B== 44857.92844624
L==1170014.46879518    L==1165953.21290651
3° 带
已知数据:
x==3354874.257000      x==532548.378000
y==386.564000          y== -209.135000
X==3354874.257000      X==532548.378000
Y==20500386.564000     Y==20499790.865000
在克拉索夫斯基椭球中:
B==301846.91668578     B== 44857.61821567
L==600014.46847295     L==595953.21302226
在IUGG1975椭球中:
B==301848.80193261     B== 44857.92037818
L==600014.46878629     L==595953.21290972
在CGCS2000椭球中:
B==301848.85270227     B== 44857.92844624
L==600014.46879518     L==595953.21290651

```

### (三) 高斯坐标的邻带换算

```

#include "stdafx.h"

#include<math.h>

```

```

#include <stdio.h>

#include <stdlib.h>

double dms_rad(double a)           //角度化弧度
{
    double sign=(a<0.0) ? -1.0 : 1.0;
    a=fabs(a);
    int d=(int)((a+0.00001)/10000.0);
    a=a-d*10000.0;
    if(a<0.0){d=d-1;a=a+10000;}
    int m=(int)((a+0.00001)/100.0);
    a=a-m*100;
    if(a<0.0){m=m-1;a=a+100.0;}
    a=sign*(d*3600.0+m*60.0+a)/206264.806247096363;
    return a;
}

double rad_msd(double a)           //弧度化角度
{
    a=a*206264.806247096363;
    double sign=(a<0.0)?-1.0:1.0;
    a=fabs(a);int d=(int)(a/3600.0+0.0000001);
    a=a-d*3600.0;
    if(a<0.0){d=d-1;a=a+3600.0;}
    int m=(int)(a/60.0+0.0001);
    a=a-m*60.0;
    if(a<0.0){m=m-1;a=a+60.0;}
    a=d*10000.0+m*100.0+a;
    return a*sign;
}

int main(int argc, char* argv[])
{
    FILE *f1;
    f1=fopen("known.txt","r");
    if(f1==NULL)
    {
        printf("can not open the file\n");
        exit(0);
    }

    double *e1=new double [3];           //读入三种参考椭球的参数

```

```

double *e2=new double [3];

double *a=new double [3];

for(int i=0;i<3;i++)

{

    fscanf(f1,"%lf%lf%lf",&e1[i],&e2[i],&a[i]);

    printf("%16.14lf %16.14lf %14.4lf\n",e1[i],e2[i],a[i]);

}

int n;

fscanf(f1,"%d",&n);

double *X3=new double[2*n];

double *Y3=new double[2*n];

for(i=0;i<n;i++)

{   fscanf(f1,"%lf%lf",&X3[i],&Y3[i]);

    printf("%14.4lf %14.4lf\n",X3[i],Y3[i]);

}

fclose(f1);

double *L=new double[3*n];

double *B=new double[3*n];

double *L1=new double[3*n];

double *B1=new double[3*n];

double *l=new double[3*n];

double *X6=new double[3*n];

double *Y6=new double[3*n];

double *y6=new double[3*n];

for(i=0;i<n;i++)                                //高斯投影反算得大地坐标

{

    for(int k=0;k<3;k++)

    {   double    A2,B2,C2,D2,b,b1,B0,N,ton,t,V,c,c0;

        int    q;

        A2=1+3*e1[k]/4+45*e1[k]*e1[k]/64+175*e1[k]*e1[k]*e1[k]/256;

        B2=3*e1[k]/4+15*e1[k]*e1[k]/16+525*e1[k]*e1[k]*e1[k]/512;

        C2=15*e1[k]*e1[k]/64+105*e1[k]*e1[k]*e1[k]/256;

        D2=35*e1[k]*e1[k]*e1[k]/512;

        b=X3[i]/(a[k]*(1-e1[k])*A2);

        do

```

```

    {   b1=b;

        B0=X3[i]/(a[k]*(1-e1[k])*A2)+(B2*sin(2*b)/2-C2*sin(4*b)/4+D2*sin(6*b)/6)/A2;

        b=B0;

    } while(fabs(B0-b1)>dms_rad(0.0000000001));

    ton=sqrt(e2[k])*cos(B0);

    t=tan(B0);

    N=a[k]/sqrt(1-e1[k]*sin(B0)*sin(B0));

    V=sqrt(1+e2[k]*cos(B0)*cos(B0));

    q=(int)(Y3[i]/1000000);

    c0=Y3[i]-q*1000000-500000;

                                c=c0/N;          B1[3*i+k]=B0-V*t*(c*c-(5+3*t*t+ton*ton-
9*t*t*ton*ton)*pow(c,4)/12+(61+90*t*t+45*pow(t,4))*pow(c,6)/360)/2;

                                B[3*i+k]=rad_msd(B1[3*i+k]);          l[3*i+k]=(c-
(1+2*t*t+ton*ton)*pow(c,3)/6+(5+28*t*t+24*t*t*t*t+6*ton*ton+8*ton*ton*t*t)*pow(c,5)/120)/cos(B0);

    L1[3*i+k]=l[3*i+k]+dms_rad((q*3)*10000);

    L[3*i+k]=rad_msd(L1[3*i+k]);          //高斯投影反算得大地坐标

    //printf("%15.4lf%15.4lf\n",B[3*i+k],L[3*i+k]);

double ton1,t1,X1,N1,m,l1;

int q1;

q1=(int)(L[3*i+k]/60000)+1;

l1=L1[3*i+k]-dms_rad(6*q1*10000-30000);

X1=a[k]*(1-e1[k])*(A2*B1[3*i+k]-B2*sin(2*B1[3*i+k])/2+C2*sin(4*B1[3*i+k])/4-
D2*sin(6*B1[3*i+k])/6);

    ton1=sqrt(e2[k])*cos(B1[3*i+k]);

    t1=tan(B1[3*i+k]);

    m=cos(B1[3*i+k])*l1;

                                N1=a[k]/sqrt(1-e1[k]*sin(B1[3*i+k])*sin(B1[3*i+k]));

X6[3*i+k]=X1+N1*t1*(m*m/2+(5-t1*t1+9*ton1*ton1+4*pow(ton1,4))*pow(m,4)/24+(61-
58*t1*t1+pow(t1,4))*pow(m,6)/720);          y6[3*i+k]=N1*(m+(1-t1*t1+ton1*ton1)*pow(m,3)/6+(5-
18*t1*t1+pow(t1,4)+14*ton1*ton1-58*ton1*ton1*t1*t1)*pow(m,5)/120);

    Y6[3*i+k]=q1*1000000+500000+y6[3*i+k]

    printf("X=%12.5lf    Y=%13.5lf\n",X6[3*i+k],Y6[3*i+k])

}

    printf("\n");

}

```



```

    fl=fopen("result.txt","w");

    if(fl==NULL)

    { printf("can not open the file\n");

        exit(0);

    }

    fprintf(fl,"=====3度带与6度带换算=====\\n");

    for(i=0;i<2;i++)

    { fprintf(fl,"第%d组数据: \\n",i+1);

        fprintf(fl,"          3度带下坐标:  X=%14.4lf  Y=%14.4lf\\n",X3[i],Y3[i]);

        for(int k=0;k<3;k++)

            { if(k==0) fprintf(fl,"          克拉索夫斯基椭球下6度带坐标为:  ");

                if(k==1) fprintf(fl,"          IUGG1975椭球下6度带坐标为:  ");

                if(k==2) fprintf(fl,"          CGCS2000椭球下6度带坐标为:  ");

                fprintf(fl,"X=%12.4lf  Y=%12.4lf\\n",X6[3*i+k],Y6[3*i+k]);

            }

        }

    return 0;

}

```

用文件输出的结果：

```

=====3度带与6度带换算=====
第1组数据:
    3度带下坐标:  X= 3858520.6946  Y= 41512354.9834
    克拉索夫斯基椭球下6度带坐标为:  X=3858520.6946  Y=21512354.9834
    IUGG1975椭球下6度带坐标为:      X=3858520.6946  Y=21512354.9834
    CGCS2000椭球下6度带坐标为:      X=3858520.6946  Y=21512354.9834
第2组数据:
    3度带下坐标:  X= 3858853.5671  Y= 42420902.8543
    克拉索夫斯基椭球下6度带坐标为:  X=3860592.2478  Y=21695272.9325
    IUGG1975椭球下6度带坐标为:      X=3860592.1772  Y=21695266.4644
    CGCS2000椭球下6度带坐标为:      X=3860592.1751  Y=21695266.2813

```

## 5、 实习体会

这次是我们大地测量学基础的第三次编程，也是最后一次。这次我的编程总得来说还算顺利，但是同时也遇到了一些问题，耗时也比较长，主要问题如下：

1、在写读入数据的文件时，总是图省事，想复制以前写过的程序中的读入方法，导致自己对其有点遗忘。

2、有时输出的数据会是乱码，最可能的原因就是没有给变量赋初值。

3、有时在输出时会出现关闭程序的现象，原因就是在循环的时候语句出错，导致了死循环。

4、在最后的邻带换算问题上，乍一看觉得问题很简单，觉得能够很快完成。但是真正动手起来才发现存在很多问题，在以后的实习中，不能再有眼高手低的现象。

在此次编程结束后，觉得自己也有了很大的进步，在程序的算法上也有了多种方法，主要的体会如下：

1、在编程的时候要利用大块的时间，不然一旦思路被打断，再次进入状态还要花时间。

2、如果调用了动态数组，在用完之后要释放空间，避免最后出现内存被占满的情况，这样也不容易出现死循环。

3、自己动手查阅资料。在遇到不会的语法时，比如如何做子程序等，可以上网查阅相关的资料，也可以请教他人。

4、我感觉编程的时候还是要在遇到问题的时候大家一起讨论，不然自己一个人用自己的思路想问题，就算出错了，也未必能查出来。

5、把程序中的错误记录下来，最好是有截屏，这样在以后就能避免相同的错误再次出现。

这次编程结果还是没有用 MFC 的格式输出，是因为自己对 C++ 还没有掌握。希望自己能够在以后利用点滴的时间多多进行语言的学习，尽快地提升自己编程能力。