



导航常用坐标系转换程序实验报告

院系： 测绘学院

专业： 导航工程（智能导航实验班）

班级： 智能导航一班

学号： 2023302143029

姓名： 秦旗峰

教师： 辜声峰

完成时间： 2024 年 4 月 19 日

目录

一、引言

二、数学原理

1、大地坐标系和空间直角坐标系

2、大地坐标系和空间直角坐标系的转换

三、程序设计

1、头文件的引入

2、命名空间声明和常量定义

3、函数的定义

4、输入与输出参数

5、函数返回值

6、单位

四、结果分析与小结

1、计算结果

2、数据分析

3、小结

一、引言

大地坐标系是大地测量学与导航学中常用的一种坐标系，亦称地理坐标系或椭球坐标系。它是以经过椭球定位后的地球椭球作为基本参考面的一种坐标系。空间直角坐标系是现代大地测量与导航中常用的又一种坐标系。大地坐标系就是我们熟知的经纬度坐标系，在生活行为中起着极大的作用，但大地坐标系在数学运算中往往不便，而空间直角坐标系恰恰在数学上极为便捷，但其远离生活，无法用在日常生活中。因此实现大地坐标系和空间直角坐标系的相互转换是导航学和大地测量学的一个重要课题。

二、数学原理

1、大地坐标系和空间直角坐标系

(1) 大地坐标系 (B, L, H)

地面上任意一点 P'在大地坐标系中的位置是用三个坐标分量:大地纬度 B、大地经度 L 和大地高 H 来表示的。它们统称为 P'点的大地坐标。其中，大地纬度 B 是过 P'点的椭球面法线与椭球赤道面之间的夹角，取值为 $0^{\circ}\sim 90^{\circ}$ 。北半球的大地纬度为正，自赤道向北量取，称为北纬；南半球的大地纬度为负，自赤道向南量取，称为南纬。大地经度 L 是过 P'点的大地子午面与大地起始子午面间的夹角，取值 $0^{\circ}\sim 180^{\circ}$ 。从起始子午面向东量取为正，称为东经，从起始子午面向西量取为负，称为西经。大地高 H 是 P'点沿法线至椭球面间的垂直距离 P'P。(B, L) 称为大地坐标系中的水平分量，H 称为大地坐标系的垂直分量（高程）

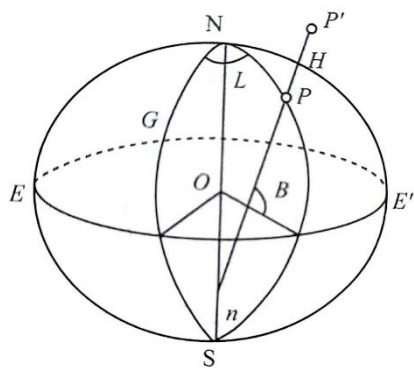


图 4-2 大地坐标系

(2) 空间直角坐标 (X, Y, Z)

空间直角坐标系是现代大地测量与导航中常用的又一种坐标系。该坐标系的坐标原点位于地球椭球的中心;z 轴与地球椭球短轴(旋转轴)一致，指向北极(N);以起始大地子午面与椭球赤道面的交线为 x 轴，指向起始子午线;y 轴垂直于 x 轴和 z 轴组成右手坐标系。空间某点的位置用三维直角坐标(X, Y, Z)来表示。在上述情况下大地坐标系与空间直角坐标系之间存在严格的数学转换关系。

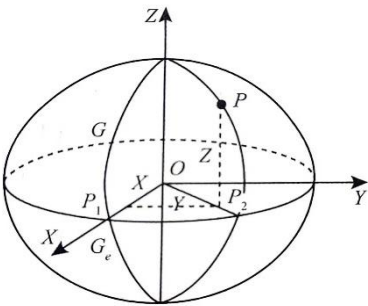


图 4-3 空间直角坐标系与子午面直角坐标系

2、大地坐标系和空间直角坐标系的转换

(1) 子午面直角坐标系

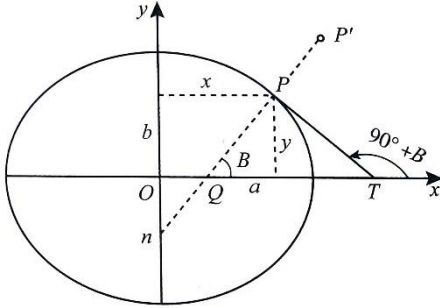


图 4-4 子午面直角坐标系

为了建立大地坐标系与空间直角坐标系之间的转换关系，有必要引入一个中间坐标系——子午面直角坐标系。在图中 P 点为空间某点 P' 在地球椭球上的投影点，即为过 P' 点的法线与地球椭球面的交点。过 P 点和椭球的短轴作一平面(子午面)，该平面与椭球的交线为一椭圆，称为过 P 点的子午圈。以椭圆的中心 O 为坐标原点在子午面上建立一个平面直角坐标系，其 x 轴与子午椭圆的长轴重合，y 轴与椭圆的短轴重合。在子午面直角坐标系中 P 点的位置用平面直角坐标(x, y)表示。

过 P 点作子午椭圆的切线 TP，TP 与 x 轴之间的夹角为 $90^\circ + B$ 。该夹角的正切值为 $\tan(90^\circ + B)$ 称为切线的斜率。子午椭圆的曲线方程为：

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

根据导数的性质，曲线方程一阶导数 $\frac{dy}{dx}$ 就等于该切线的斜率 $\tan(90^\circ + B)$ 。将上式对 x 求导数后有 $\frac{2x}{a^2} + \frac{2y}{b^2} \frac{dy}{dx} = 0$, 即 $\frac{dy}{dx} = -\frac{b^2 y}{a^2 x} = \tan(B + 90^\circ) = -\cot B$,

$$\tan B = \frac{a^2 y}{b^2 x} = \frac{1}{1 - e^2} \frac{y}{x}$$

或写成

$$y = x(1 - e^2) \tan B$$

将上式代入椭圆方程式后可得 $\frac{x^2}{a^2} + \frac{x^2(1-e^2)^2 \tan^2 B}{a^2(1-e^2)} = 1$, 整理后可得 $\frac{x^2}{a^2} \cdot \frac{1-e^2 \sin^2 B}{\cos^2 B} = 1$ 。

引入辅助参数 $W = \sqrt{1 - e^2 \sin^2 B}$ 后有：

$$x = \frac{a \cos B}{W}$$

代入并引入辅助参数 V 可得：

$$y = \frac{a(1 - e^2)}{W} \sin B = \frac{b \sin B}{V}$$

在这里我们就得到了子午面直角坐标系 x, y 与大地纬度 B 之间的函数关系式。

由子午面直角坐标系图可知，x 坐标实际上就是通过 P 点的平行圈（纬度）的半径 r，如果令图中 Pn 为 N，N 就是过 P 点的卯酉圈的曲率半径，则有：

$$\begin{cases} r = x = \frac{N \cos B}{a} \\ N = \frac{a}{W} \end{cases}$$

于是，可以进一步得到：

$$\begin{cases} x = N \cos B \\ y = N(1 - e^2) \sin B = \frac{b}{V} \sin B \end{cases}$$

(2) 坐标转化

从图 4-3 可以看出空间直角坐标 XYZ 与子午直角坐标 x、y 之间有如下关系式：

$$\begin{cases} X = x \cos L \\ Y = x \sin L \\ Z = y \end{cases}$$

L 为 P 点的大地经度。代入可得：

$$\begin{cases} X = N \cos B \cos L \\ Y = N \cos B \sin L \\ Z = N(1 - e^2) \sin B \end{cases}$$

即求得椭球面上 P 点的空间直角坐标与大地坐标之间的转换关系式。

空间内 P' 点的大地高为 H，即法线到椭球面的距离，所以由上式，容易得到已知空间任意一点的大地坐标 (B, L, H) 后计算其空间直角坐标的公式：

$$\begin{cases} X = (H + N) \cos B \cdot \cos L \\ Y = (H + N) \cos B \cdot \sin L \\ Z = [N(1 - e^2) + H] \sin B \end{cases}$$

通过上式的逆运算，就可以从某点的空间直角坐标系 (X, Y, Z) 计算该点的大地坐标 (B, L, H) 的公式如下：

$$\begin{cases} \tan B = \frac{Z + Ne^2 \sin B}{\sqrt{X^2 + Y^2}} = \frac{1}{\sqrt{X^2 + Y^2}} \left(Z + \frac{ae^2 \tan B}{\sqrt{1 + (1 - e^2) \tan^2 B}} \right) \\ L = \arctan \frac{Y}{X} = \arcsin \frac{Y}{\sqrt{X^2 + Y^2}} = \arccos \frac{X}{\sqrt{X^2 + Y^2}} \\ H = \frac{\sqrt{X^2 + Y^2}}{\cos B} - N \end{cases}$$

经度 L 的求解较为方便，纬度 B 和大地高 H 为超越方程，很难求得解析解。H 与 B 相关，故我们先讨论 B 的解法。

$\tan B = \frac{Z + Ne^2 \sin B}{\sqrt{X^2 + Y^2}} = \frac{1}{\sqrt{X^2 + Y^2}} \left(Z + \frac{ae^2 \tan B}{\sqrt{1 + (1 - e^2) \tan^2 B}} \right)$ 为三角函数形式的超越方程，一般使用迭代

法求解其根。 $B = \arctan \left(\frac{Z + Ne^2 \sin B}{\sqrt{X^2 + Y^2}} \right)$

在这里我们令 $\Delta Z = Ne^2 \sin B$ 那么 $B = \arctan \left(\frac{Z + \Delta Z}{\sqrt{X^2 + Y^2}} \right)$ 并且给予初值 $\Delta Z_0 = e^2 Z$

将 ΔZ_0 带入 $B = \arctan \left(\frac{Z + \Delta Z}{\sqrt{X^2 + Y^2}} \right)$ ，我们得到一个值，将其命名为 B_0

由这个 B_0 反代入 $\Delta Z = Ne^2 \sin B$ 可以得到另外一个 ΔZ ，记作 ΔZ_1

将 ΔZ_1 又代入 $B = \arctan \left(\frac{Z + \Delta Z}{\sqrt{X^2 + Y^2}} \right)$ 就可求解处第二个 B 值，记作 B_1

.....

如此循环代入，当出现 $|B_{n+1} - B_n| < \varepsilon$ 时 (ε 为一个极小的值)，即 B_n 收敛时，以 B_n 极限为此方程的解

当求出 B 时，将 B 反代回 $H = \frac{\sqrt{X^2 + Y^2}}{\cos B} - N$ ，即可将 H 求解得出。

利用迭代法我们可以通过给定已知的 (X, Y, Z) 求解出 (B, L, H)，从而实现从大地坐标系到空间直角坐标系的转化。

三、程序设计

编程要求：

- 题目：已知如下点的空间直角坐标，请将其转为大地坐标，对转换结果的经度、纬度各增加1度，再转为空间直角坐标

X	Y	Z
-2148745.464	4426640.856	4044655.710
-2148744.866	4426643.990	4044656.003
-2148746.822	4426645.265	4044653.164
-2148743.658	4426641.009	4044652.936
-2148749.679	4426642.122	4044654.862
-2148740.708	4426644.350	4044658.894

1、头文件的引入

```
1  #include<iostream>
2  #include<iomanip>
3  #include<cmath>
```

本次编程使用了 `<iostream>` `<iomanip>` `<cmath>` 等官方库头文件。`<iostream>` 用于控制输入输出流，`<iomanip>` 控制数据的输出格式，`<cmath>` 用于数学计算。

2、命名空间声明和常量定义

```
4  using namespace std;
5  #define a 6378137.00
6  #define b 6356752.31424517
7  #define f 1 / 298.257223563
8  #define e 0.0818191908426
9  #define e2 0.0066943799013
10 #define pi acos(-1)
```

常量已经给出，在此定义常量

参数	值	注释
<i>a</i>	6378137.00	地球半长轴(m)
<i>b</i>	6356752.31424517	地球短半轴(m)
<i>f</i>	1/298.257223563	地球扁率
<i>e</i>	0.0818191908426	第一偏心率
<i>e</i> ²	0.0066943799013	第一偏心率平方

3、函数的定义

需要定义两个函数，分别是 void BLH2XYZ () 和 void XYZ2BLH ()

(1) BLH2XYZ(double B, double L, double H) 大地坐标系向空间直角坐标系转化函数

```
20 void BLH2XYZ(double B, double L, double H)
21 {
22     double N = a / sqrt(1 - e2 * sin(B*pi / 180) * sin(B*pi / 180));
23     double X = (N + H) * cos(B*pi / 180) * cos(L*pi / 180);
24     double Y = (N + H) * cos(B*pi / 180) * sin(L*pi / 180);
25     double Z = (N * (1 - e2) + H) * sin(B*pi / 180);
26     cout << "经纬度分别加1° 后坐标转化为XYZ分别为" << setprecision(12) << "X=" << X << ", Y=" << Y << ", Z=" << Z << endl;
27 }
```

$$\begin{cases} X = (H + N) \cos B \cdot \cos L \\ Y = (H + N) \cos B \cdot \sin L \\ Z = [N(1 - e^2) + H] \sin B \end{cases}$$

需要将数学推导的公式转化为编程语言。应该注意 C++ 中所有三角函数的运算都是以弧度制为基础，在程序编写时要注意弧度与角度之间的相互转换。

(2) XYZ2BLH(double X, double Y, double Z) 空间直角坐标系向大地坐标系转化函数

```
31 void XYZ2BLH(double X, double Y, double Z)
32 {
33     double B = 0, H = 0; //利用X, Y, Z求B, L, H
34     double L = atan(Y / X) * 180 / pi;
35     if (L < 0) L += 180;
36     double Z0 = e * e * Z; //迭代法求B
37     double B0 = atan((Z + Z0) / sqrt(X*X + Y * Y));
38     for (int i = 0; i <= 15; i++)
39     {
40         Z0 = e * e * sin(B0) * a / sqrt(1 - e * e * sin(B0) * sin(B0));
41         B0 = atan((Z + Z0) / sqrt(X*X + Y * Y));
42         B = B0;
43     }
44     H = sqrt(X*X + Y * Y + (Z + Z0)*(Z + Z0)) - a / sqrt(1 - e * e * sin(B0) * sin(B0));
45     B = B * 180 / pi;
46     cout << "此坐标转化为BLH分别为" << setprecision(12) << "B=" << B << ", L=" << L << ", H=" << H << endl;
}
```

在此处，使用 $L = \arctan \frac{Y}{X}$ 作为 L 的计算方式，需要注意的是， $y = \arctan(x)$ 的值域为 $(-\frac{\pi}{2}, \frac{\pi}{2})$ 而 L (大地经度) 的取值范围是 $0^\circ \sim 180^\circ$ ，因此如果算得的 L 为负数则加上 180° ，使得 L 的值落在 $90^\circ \sim 180^\circ$ ，L 为正数（位于 $0^\circ \sim 90^\circ$ ）则无需改变。（注：东西经通过 Y 的正负判断）

为了方便表示，本函数的编写中 $B_0 = B_n$ ， $Z_0 = \Delta Z_n$ ，迭代采用了循环次数限制。

在数据验证过程中，我发现使用阈值限制 ($|B_{n+1} - B_n| < \varepsilon$) 所得到的 B 值误差相较于次数限制更大，此处原因不明。

在函数计算过程中，我使用 $e \times e$ 代替了 e^2 ，因为直接使用 e^2 的误差更大。

```
44     BLH2XYZ(B + 1, L + 1, H); //加1° 处理
45 }
```

在 XYZ2BLH 函数体中对经纬度加 1° 处理。完成目标任务。

4、输入与输出参数

```
14  int main()
15  {
16      double X, Y, Z;
17      cout << "请依次输入X, Y, Z: " << endl;
18      cin >> X >> Y >> Z;
19      void XYZ2BLH(double X, double Y, double Z);           //声明函数
20      XYZ2BLH(X, Y, Z);
21      return 0;
22  }
```

在 main 函数中先定义变量 X, Y, Z, 输入依此 (X, Y, Z), 后声明 XYZ2BLH () 函数, 在 XYZ2BLH () 函数中输出 (B, L, H), 并且将 B, L 分别加 1°处理后调用 BLH2XYZ () 函数, 输出处理后的 (X, Y, Z)

5、函数返回值

```
23  void BLH2XYZ(double B, double L, double H)
31  void XYZ2BLH(double X, double Y, double Z)
14  int main()
```

定义的两个函数均为 void 类型, 无返回值, 仅用作数值操作和计算

main 函数返回值为: 0

6、单位

B (大地纬度)、L (大地经度) 的单位均为角度 (°), H (大地高), (X, Y, Z) 的单位是米 (米), 注意角度和弧度之间的相互转换

四、结果分析与小结

1、计算结果

输入第一组数据: X=-2148745.464 Y=4426640.856 Z=4044655.710

请依次输入 X, Y, Z:

-2148745.464 4426640.856 4044655.710

此坐标转化为BLH分别为B=39.608598523,L=115.892501853,H=87.4920821032

经纬度分别加1°后坐标转化为XYZ分别为X=-2193317.51921,Y=4324667.60117,Z=4129579.41542

D:\VS\Project_VS\Project4\Debug\Project4.exe (进程 3136)已退出, 返回代码为: 0。

若要在调试停止时自动关闭控制台, 请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口...

网站验证:

输入坐标

输入:

-2148745.464 ,4426640.856 ,4044655.710

转换结果

输出:

115.8925018533 , 39.608598523 , 87.4920821078

验证结果中 B (大地纬度) 和 L (大地经度) 的精确度较高, 而 H (大地高) 的误差在小数点后 9 位较为明显, 总体精度较高。

将第一组数据输出结果带回验证加 1°:

X=-2193317.51921,Y=4324667.60117,Z=4129579.41542

请依次输入 X, Y, Z:

-2193317.51921 4324667.60117 4129579.41542

此坐标转化为BLH分别为B=40.6085985255,L=116.892501853,H=87.4921971094

经纬度分别加1°后坐标转化为XYZ分别为X=-2234300.40833,Y=4221194.88014,Z=4213256.27355

D:\VS\Project_VS\Project4\Debug\Project4.exe (进程 5264)已退出, 返回代码为: 0。

若要在调试停止时自动关闭控制台, 请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口...

误差存在, 在小数点第 8 位体现, 并且 B 和 L 的误差具有增大的趋势。

之后, 分别代入后几组数据进行验证。

X=-2148744.866 Y=4426643.990 Z=4044656.003

请依次输入 X, Y, Z:

-2148744.866 4426643.990 4044656.003

此坐标转化为BLH分别为B=39.6085858663,L=115.892479653,H=89.6498071365

经纬度分别加1°后坐标转化为XYZ分别为X=-2193316.9983,Y=4324670.72783,Z=4129579.75284

转换结果

115.8924796535 , 39.6085858663 , 89.6498071412

X=-2148746.822 Y=4426645.265 Z=4044653.164

请依次输入X, Y, Z:
-2148746.822 4426645.265 4044653.164
此坐标转化为BLH分别为B=39.6085546758,L=115.89249366,H=89.3815648248
经纬度分别加1°后坐标转化为XYZ分别为X=-2193318.98311,Y=4324672.02067,Z=4129576.94874

转换结果

115.8924936599 , 39.6085546758 , 89.3815648286

X=-2148743.658 Y=4426641.009 Z=4044652.936

请依次输入X, Y, Z:
-2148743.658 4426641.009 4044652.936
此坐标转化为BLH分别为B=39.6085830126,L=115.892482157,H=85.2219978664
经纬度分别加1°后坐标转化为XYZ分别为X=-2193315.76012,Y=4324667.81799,Z=4129576.63026

转换结果

115.8924821572 , 39.6085830126 , 85.2219978711

X=-2148749.679 Y=4426642.122 Z=4044654.862

请依次输入X, Y, Z:
-2148749.679 4426642.122 4044654.862
此坐标转化为BLH分别为B=39.6085755299,L=115.892539569,H=89.2469366081
经纬度分别加1°后坐标转化为XYZ分别为X=-2193321.72029,Y=4324668.82777,Z=4129578.61921

转换结果

115.8925395689 , 39.6085755299 , 89.2469366128

X=-2148740.708 Y=4426644.350 Z=4044658.894

请依次输入X, Y, Z:
-2148740.708 4426644.350 4044658.894
此坐标转化为BLH分别为B=39.6086144931,L=115.892434267,H=90.3435730077
经纬度分别加1°后坐标转化为XYZ分别为X=-2193312.87488,Y=4324671.0896,Z=4129582.61778

转换结果

115.8924342672 , 39.6086144931 , 90.3435730133

与网站验证结果对照，发现误差主要在H（大地高）的数据上体现。

2、数据分析

(1) 数据精度

计算得到的大地纬度和大地精度误差较小，大地高的误差小于 10^{-8} 。

(2) 误差来源

本次实验所用迭代法求大地纬度是误差的主要来源，由于没有解析解，迭代出来的为近似解，导致误差必然出现。

同时，误差出现可能还与计算机工作机理有关，例如使用次数循环误差较小，而阈值限制误差较大，这一点我还不是特别明白其中的原因。

误差的另一个来源是常量的定义，例如，定义 π 时使用 $\arccos(-1)$ 的误差相较于直接定义数值小很多，使用 $e \times e$ 的误差比直接使用 e^2 给定数值更小。

(3) 减小误差

在编程过程中减小误差的最好方式是找到更好的算法以及充分了解计算机的工作原理。

3、小结

本次上机编程完成了大地坐标系和空间直角坐标系两大常用坐标系之间的转换。大地坐标系实用而不利于计算，空间直角坐标系利于数学计算而不够直观，二者各有利弊。用编程实现二者的转换，并加以验证，较为精确的实现了坐标变换。为今后的专业学习打下基础。