中国地质大学（武汉）

学士学位论文

测量坐标系转换方法的探讨与软件实现

学　　号：20161000623

姓　　名：袁凯田

学科专业：测绘工程

指导教师：李玮 讲师

企业导师：胡波 教授级高工

培养单位：地理与信息工程学院

二〇二〇年五月

学士学位论文原创性声明

本人郑重声明：本人所呈交的学士学位论文《测量坐标系转换方法的探讨与软件实现》，是本人在指导老师的指导下，在中国地质大学（武汉）攻读学士学位期间独立进行研究工作所取得的成果。论文中除已注明部分外不包含他人已发表或撰写过的研究成果，对论文的完成提供过帮助的有关人员已在文中说明并致以谢意。

本人所呈交的学士学位论文没有违反学术道德和学术规范，没有侵权行为，并愿意承担由此而产生的法律责任和法律后果。

学位论文作者签名：_____

日　期：　　年　　月　　日

摘 要

随着以 GNSS 为主导的空间定位技术的迅猛发展，当前使用 GNSS 接收机在全球范围内都能获取 WGS-84 或 ITRF 坐标系下的坐标结果，但此坐标成果无法直接用到我国的坐标系统中，因此研究不同坐标系统之间的转换关系，求解各种系数，将所有测量所得的成果，坐标数据转换到同一个基准下，具有重要的现实意义和研究价值。

本文从椭球的基本几何参数和我国常用的几种坐标系（1954 年北京坐标系，1980 西安坐标系，WGS-84 坐标系，CGCS2000 坐标系）的建立方法与基本理论知识出发，探讨不同坐标系之间的转换方法，包括：大地坐标与空间直角坐标相互转化；大地坐标与平面坐标之间的相互转换；不同空间直角坐标系之间的相互转换与参数计算。

根据相关的理论知识和不同坐标的转换方法，利用 C#窗口应用程序设计了一套功能齐全的测量坐标转换软件，能够实现同一参考椭球下和不同参考椭球之间的不同测量坐标之间的相互转换，即大地坐标与空间直角坐标相互转化，不同空间直角坐标系之间的转换；大地坐标与平面坐标之间的相互转换。要求软件操作方便、界面友好,兼顾单点转换与批量转换，并保证精度满足日常测绘工作的要求。并对软件各个功能板块进行测试。

关键词：坐标系，坐标转换，高斯平面投影，七参数模型，测量坐标转换系统

目 录

第一章 绪论

1.1 研究背景

在我国目前的测绘工作中，存在着地心坐标系、参心坐标系和地方独立坐标系这三种坐标系。它们可以通过坐标原点的不同来进行区分。这三类坐标系统服务对象和使用范围都不同，我国目前使用的参心坐标系包括 1954 年北京坐标系和 1980 西安坐标系。随着以全球卫星导航系统为主导的空间定位技术的迅猛发展，和 GPS 及其相关技术的广泛应用，地心坐标系也变得更为普遍，所以在 2008 年 7 月 1 日我国正式启用了一个地心坐标系：2000 国家大地坐标系[1]。而地方独立坐标系则是根据工程测量的需要建立的。

在测绘工作中，为了表示椭球上某一点的位置，就必须建立相应的坐标系，坐标系类型大致可以分为以下几类：大地坐标系、空间直角坐标系和平面直角坐标系等等。椭球上任意一点都可以用不同的坐标系进行表示，并且存在着某种联系和转换规律，这就需要我们对其进一步研究和探讨。

我国主要应用的参考椭球有四个，分别是克拉索夫斯基椭球体、1975 年国际椭球体、WGS-84 椭球体、CGCS2000 椭球体，它们的椭球参数是利用天文大地测量和重力大地测量联合解算获得的。在不同的国家和地区，由于采用了不同的椭球参数，就会形成不同的大地坐标系；而同一国家和地区在不同的历史阶段，采用不同的椭球参数，也会形成不同的大地坐标系。例如：WGS-84 坐标系采用的是 WGS-84 椭球体参数；北京 54 坐标系采用的是克拉索夫斯基椭球体参数；西安 80 坐标系采用 1975 年国际椭球体参数；而 2000 国家大地坐标系采用的是 CGCS2000 椭球体参数。

根据上述内容，可以知道同一基准下坐标和不同基准坐标之间都存在着某种转换关系和规律，所以本文将探讨同一参考椭球下的各个坐标系之间的转换关系和不同参考椭球下的各个坐标系之间的转换关系，并对这些关系进行程序编写加以实现，实现满足测绘工作精度条件下坐标的批量转换。

1.2 国内外的坐标系统发展现状

1.2.1 国内国家大地坐标系统建设概况

我国建国初期，由于技术手段和科技水平不足，但又急需一个大地坐标系作为基准来进行测绘工作的开展，所以先后建立了两个局部大地坐标系。此时建立起的大地坐标系是通过传统大地测量技术获得的，都是参心坐标系。随着空间大地测量的兴起，产生了全球大地坐标系，它的基本特点是地心的、三维的[2]。1960年左右，随着各种卫星的使用和技术手段的创新，我国逐渐进入空间大地测量时代，原来的参心坐标系已经满足不了日常测绘工作的需要，所以我国开始逐步建立地心大地坐标系。

由于历史和技术的原因，建国初期，我国先后使用过 1954 年北京坐标系，1980 西安坐标系，和新 1954 年北京坐标系。这三个坐标系都是参心坐标系。1954 年北京坐标系的原点在前苏联；而 1980 西安坐标系的原点在我国中部，陕西省泾阳县永乐镇。1954 年北京坐标系存在着许多不合理的地方，随着科学技术的发展，逐渐被淘汰，为了适应我国测绘事业的发展，后期我国建立了以陕西省泾阳县永乐镇为原点的 1980 年国家大地坐标。但是因为在此之前用了较长时间的 $BJ54_{旧}$，坐标系统更新换代后，给测绘成果的转换带来了不便，所以又建立了一个过渡的坐标系 $BJ54_{新}$[3]。

但是随着 GNSS 等空间定位技术的不断发展，这几个适用于局部地区的参心坐标系已经不能适应技术的发展和测绘工作的要求。所以，我国有关部门开始进行关于建立地心坐标系方面的研究。先后建立了两个转换参数，分别为地心一号和地心二号。DX-1 和 DX-2 都用于将我国前期建立的参心坐标系换算为我国地心坐标系 DXZ_{78} 或 DXZ_{88} 的坐标。最终在 2008 年 7 月 1 日，正式启用我国的国家大地坐标系，即 CGCS2000，并沿用至今[4]。

这些上述的坐标系为我们的国民经济建设、国防建设、社会发展和科技发展提供了重要的支撑。

1.2.2 国外国家大地坐标系统建设概况

国际地球自转服务局（IERS）于 1988 年 1 月 1 日成立，提供的服务有：国际地球参考系统（ITRS）、国际天球参考系统（ICRS）和地球定向参数（EOP）。采用甚长基线干涉测量(VLBI)、卫星激光测距(SLR)、GPS 和卫星多普勒定轨定位(DORIS)等空间大地测量技术，对地球进行观测，并对其进行联合平差。由此得到

地面观测站的站坐标和速率，以及相应的地球定位定向参数(EOP)，以此结果建立国际地球参考系（ITRS）和国际地球参考框架（ITRF）[5]。目前，ITRF 已成为国际公认的应用最广、精度最高的地心坐标参考框架[6]。

1957 年以来，随着卫星的升空，各国开始着手建立地心坐标系。美国国防部曾先后建立过世界大地坐标系 WGS-60，WGS-66，和 WGS-72[3]。经过长期的精化和调整，建立了 WGS-84 坐标，并沿用至今。

WGS-84 和 ITRF 均属于协议坐标系，其中 ITRF 遵循以下四个原则建立：①原点是包括整个大气和海洋在内的整个地球的质心；②长度是米，是在广义相对论框架下的定义；③定向由 BIH 给出的在历元 1984.0 的地球自转参数确定，④采用的时间基准是满足无整体旋转条件的板块运动模型[2]。而 WGS-84，是一个协议地球参考系。作为一个地心坐标系，它的原点是地球的质心；Z 轴指向协议地球极方向；X 轴指向 BIH1994.0 零度子午面与赤道的交点；Y 轴与 ZOX 平面构成右手坐标系。

在美国的世界大地坐标系之后，前苏联也建立起了自己的全球大地坐标系。CK-90 在 1988 年正式被俄罗斯军方启用，并与 CK-42 并用。之后，俄罗斯又建立了与 GLONASS 卫星导航系统相匹配的地心坐标系 PZ-90。它的精度为 1-2m，控制点的相对精度为 0.2-0.3m[7]。

随着 GNSS 等空间定位技术的不断发展，我国的一些邻近国家都对本国的大地基准进行了更新与换代。例如：马来西亚建立了国家 3 维地心大地坐标系统 NGRF2000，新西兰于 1998 年建立了一个新的大地坐标系统 NZGD2000，日本从 2000 年 4 月开始采用新的大地基准 JGD2000，韩国于 1998 年建立了新的国家 3 维地心大地坐标系统，KGD2000[8]。

各国都在进行坐标系统的更新换代，以满足目前测绘工作的精度需要，来促进国防建设与经济建设。

1.3 研究意义

随着 GPS、VLBI、SLR 等空间测量技术的定位精度不断改进，新的 ITRF(国际地球参考框架)中点的精度已经可以达到毫米级以上。但是，我国目前用于工程设计以及测图的大地控制点大部分采用的仍然是常规的地面测量控制点。但是这些地面测量控制点一般又都是基于 1954 年北京坐标系和 1980 西安坐标系为框架建立的[9]。所以大地坐标系的更新换代,是经济建设、国防建设、社会发展和科技发展的客观需要。

大地坐标系的更新换代就会出现一个问题，大地坐标基准不统一，导致测绘成果难以统一起来，给测绘工作带来不便。随着以 GNSS 为主导的空间定位技术的迅猛发展，当前在全球范围内都能获取 WGS-84 或 ITRF 坐标系下的坐标结果。这些坐标成果与在参心坐标系下获取的坐标成果之间无法统一与直接使用，我们需要把这些不同的大地网统一起来，所以要研究不同坐标系统之间的转换关系，探讨各种转换参数的求解方法，保证测绘成果和资料能统一到目标坐标系下，才能更好地为我国测绘事业的发展服务，这对于我们开展测绘工作和进行国防、经济建设有着重要的现实意义。

1.4 论文目标及主要内容

对我国当前常用大地坐标系统进行介绍，包括椭球的几何参数，坐标系的建立原理，建立过程，介绍并对它们进行比较，分析，研究，找到几种坐标系之间存在着的某种联系和转换规律。探讨同一坐标基准下不同坐标系统之间的转换关系和各个坐标基准之间的转换关系并加以实现。其中包括：空间直角坐标（XYZ）与大地坐标（BLH）之间的相互转化；空间直角坐标系之间的相互转换；大地坐标与平面坐标之间的相互转换等等。

最后根据所阐述的坐标系的转换模型和参数设计相应的坐标系统转换软件，实现大地坐标系和空间直角坐标系的相互转换及其高斯平面坐标之间的相互转换，并对软件进行测试。

本论文的主要内容安排如下：

1、第一章“绪论”。从论文背景出发，阐述进行本课题研究的必要性和意义，并简单介绍了坐标转换的主要研究内容。

2、第二章“常用坐标系介绍”。阐述了地球椭球的几何形状和基本几何参数，对测量坐标系统的类型进行说明，并对我国当前常用的几种坐标系统进行比较，最后介绍了高斯投影的相关内容。

3、第三章“常用大地坐标系统的相互转换”。将坐标转换简单分为两个部分，测量坐标系转换和测量坐标基准转换，并介绍基本的转换方法。

4、第四章“常用大地坐标系相互转换软件的设计与实现”。阐述软件的基本要求，并设计界面和布局；列出系统实现的流程图；最后说明软件的各个功能板块，基于 C#语言实现此软件系统。

5、第五章“应用与测试”。以实例验证软件的可行性和运算精度。

6、第六章“总结”。对论文的内容进行总结，并列出论文和软件系统的不足

之处。

第二章 常用坐标系介绍

2.1 地球椭球

在大地测量中，为了便于计算，把地球表面看作一个椭球体，并称之为地球椭球，如下图 2.1 中所示。具有一定的几何参数、定位以及定向的并且用来表示某一大地面的地球椭球叫做参考椭球[10]。其中：地球椭球的中心为 O 点；NOS 是地球椭球的旋转轴；OA 是椭球的长半轴；ON 是椭球的短半轴；AEE′ 是赤道；SAKN 是地球椭球的子午圈；KQQ′ 是地球椭球的平行圈。

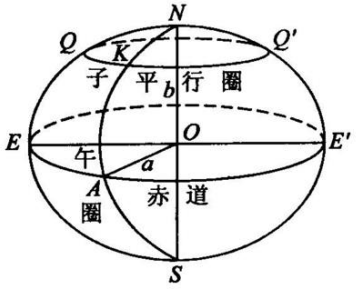

图 2.1 地球椭球

椭球有 5 个基本参数，分别是：长半轴 a、短半轴 b、扁率 α 、第一偏心率 e、第二偏心率 e′ 。其中 a，b 是长度元素，扁率 α 表示椭球的扁平程度。只需要知道其中两个元素（其中一个为长度参数，另一个为任意参数）就可以确定椭球的大小和轮廓。

2.2 地球椭球的基本几何参数

地球椭球的基本几何参数是利用天文测量方法和重力测量方法联合解算获得的数据。建立大地测量参考系统的关键是确定地球椭球的形状和大小。地球椭球由 4 个参数确定：GM(万有引力常数与质量的乘积)与地球实际值相等；U_0(表面的正常重力位)与大地水准面上的重力位相等；J_2(动力形状因子)与实际地球的相等；ω(自转角速率)与实际地球的相等。地球椭球的 4 个参数中，U_0 常由 a(长半轴)代替，J_2 常由 α (扁率)代替[11]。我国主要应用的参考椭球有四个，分别是克拉索夫斯

基椭球体、1975 年国际椭球体、WGS-84 椭球体、CGCS2000 椭球体。我国建立的 BJ54 坐标系采用的是克拉索夫斯基椭球体；西安 80 坐标系采用的是 1975 年国际椭球体；全球定位系统，即 GPS 采用的是 WGS-84 系椭球参数；2000 国家大地坐标系采用的是 CGCS2000 椭球体。这四个椭球的椭球元素在表 2.1 进行展示：

表 2.1 地球椭球的基本几何参数

	克拉索夫斯基椭球	1975 年国际椭球体	WGS-84 椭球体	CGCS2000 椭球体
A(M)	6378345	6378140	6378137	6378137
B(M)	6356863.0187730473	6356755.2881575287	6356752.3142	6356752.3141
A	1/298.3	1/298.257	1/298.257223563	1/298.257222101
E	0.08327194143083	0.08181922145552	0.08181919084255	0.0818191910428
E′	0.08208852182055	0.08209446887259	0.08209443794965	0.0820944381519

2.3 测量坐标系统类型

大地测量学的主要目标之一是测量和描绘地球并监测其变化。为了表示、描绘和分析所获得的测量成果，我们必须建立起大地坐标系。现行的大地坐标系统又分为局部坐标系、地心坐标系和地方独立坐标系这三种。局部大地坐标系的椭球面与本区域大地水准面最为接近，而对原点没有要求；大地坐标系的参考椭球面与全球大地水准面最为接近，且原点要与地球质心相重合[12]；地方独立坐标系的坐标原点与中央子午线由当地测区位置决定。

2.3.1 参心坐标系

参心坐标系是每个国家为了研究本国局部地区的地球信息，在使地面观测数据改算到椭球的各个改正数最小的原则下，选择和局部区域的大地水准面最为密合的椭球作为参考椭球建立的坐标系[13]。“参心”意指参考椭球的中心。由于参心坐标系对椭球中心没有特殊的要求，所以坐标原点一般和地球的质心不重合。参心坐标系有两种表现形式，参心大地坐标系和参心空间直角坐标系[14]。参心坐标系的建立步骤分为以下四步：

①椭球的几何参数的确定（确定椭球大小）；

②确定椭球的中心位置（椭球定位）；

③确定椭球短轴的指向（椭球定向）；

④建立大地原点。

地球参考椭球有两种定位方式，分别是多点定位和一点定位。其中多点定位可以使得在特定区域椭球面与大地水准面最佳符合。根据不同的参考椭球、不同的大地原点和不同的起始观测数据，可以确定不同的参心坐标系。我国曾先后使用过 3 个参心坐标系，分别为：1954 年北京坐标系；1980 西安坐标系；新 1954 年北京坐标系。

2.3.2 地心坐标系

地心坐标系的坐标原点是地球的质量中心，地球椭球的中心与地球的质心相重合，椭球面在全球范围内与大地水准面最为密合，而且椭球的短轴过地球的质心并指向北极[3]。和参心坐标系一致，可以根据坐标表现形式的不同进行划分。分别为：地心大地坐标系；地心空间直角坐标系。

地心空间直角坐标系（X，Y，Z）的确立方法为：X 轴从原点指向格林尼治子午面与赤道的交点；Z 轴从原点指向北极；Y 轴与它们构成右手坐标系。而地心大地坐标系（B，L，H）的确立方法为：大地经度是过地面点的子午面与格林尼治子午面的夹角；大地纬度是过地面点的法线与赤道面的夹角；大地高是地面点沿着法线到椭球面的距离。

地心大地坐标系的建立主要有两种方式。其中，直接通过测量所得的数据，计算求得点位信息的方法称为直接法；通过测量所得的数据，根据一定的数学规律解算获得两个坐标系之间的转换模型和转换参数的方法称为间接法。例如应用全球天文大地水准面差距法以及利用卫星网与地面网重合点的两套坐标建立地心坐标转换参数等方法[15]。我国在前期建立了两套转换参数用于间接法建立地心大地坐标系，分别为地心一号（DX-1）转换参数和地心二号（DX-2）转换参数。我国使用过的地心坐标系包括：DXZ_{78} 坐标、DXZ_{88} 坐标、WGS-84 坐标系和 CGCS2000 坐标系。

2.3.3 地方独立坐标系

在工程测量设计施工阶段布设控制网时，根据一定的精度要求，需要独立设定中央子午线，和平均高程，以此作为基准建立的坐标系称之为地方独立坐标系。测区如果偏离中央子午线太远，或者平均海拔太高，都会引起测量结果的变形太大，使得测量结果不能满足工程需要，所以需要建立独立坐标系[16]。

在工程测量中，施工放样阶段，一般国家所规定的坐标系往往无法满足其精度要求，导致测量所得的坐标数据与实际产生较大偏差，产生此现象的原因大致

可以分为两类。首先国家坐标系中按高斯投影分带，都是固定的带宽间隔，通常是 6 度带或者 3 度带，工程测量施工区域若离该带中央子午线较远，就会产生较大变形；再者国家坐标系的高程归化面是参考椭球面，各地区的地面位置与参考椭球面都有一定的距离，这两项将产生高斯投影变形改正和高程归化改正，经过这两项改正后的长度不可能与实测的长度相等[17]。

高程归化改正：$\Delta D_1 = -S_0 \frac{H_m}{R}$

高斯投影变形改正：$\Delta D_2 = \frac{y_m^2}{2R^2} S$

总变形：$\Delta D = \Delta D_1 + \Delta D_2 = D(\frac{y_m^2}{2R^2} - \frac{H_m}{R})$

上式中：H_m 为平均高程；R 为归算边方向参考椭球法截弧的曲率半径；y_m 为归算边两端点横坐标平均值。

所以为了降低这两个变形改正所带来的误差影响，需要建立一个地方独立坐标系。首先，确定工程测量的测区区域，然后确定其中央子午线和高斯投影面。其中，中央子午线经度为测量区域中心的经度，高斯投影面为该区域的平均高程面。 除此之外，还可以通过总变形等于零的公式推算出一个平均高程面作为高斯投影面。

2.4 我国当前常用坐标系统

2.4.1 1954 年北京坐标系

在我国大地测量初期，采用了一个参心坐标系，即 1954 年北京坐标系，采用的是克拉索夫斯基椭球参数，原点在前苏联的普尔科沃。其采用的基本椭球参数为：长半径 a=6378137m，扁率 α =1/298.3。

由于当时的生产水平不高，1954 年北京坐标系是与前苏联 1942 年坐标系进行联测而得的，因此，可以把它认为是前苏联 1942 年坐标系的延伸。该大地坐标系是按局部平差的方法提供大地点数据的，我国根据此坐标系完成了许多初期的勘测任务并建立了我国的天文大地网。

但是随着科学水平和技术手段的不断发展，我们对测量结果的精度要求越来越高，此坐标系的弊端就逐渐显现出来。例如：椭球参数不够精确，几何大地测量和物理大地测量采用的参考面不统一，参考椭球面与我国的大地水准面有明显

的差距，定向不明确，给坐标的换算带来不便等等。

2.4.2 1980 西安坐标系

1978 年 4 月在西安召开了“全国天文大地网整体平差会议”[15]，在会上有专家提出了 1954 年北京坐标系的一些不足，参考椭球参数需要更加精确，椭球面要与我国大地水准面最为接近等等，为了满足我国测绘工作的需要和精度要求，我国建立了以陕西省泾阳县永乐镇为原点的 1980 年国家大地坐标系。

1980 西安坐标系是在 1954 年北京坐标系的基础上建立的，也是参心坐标系。椭球元素采用 1975 年国际大地测量与地球物理联合会第十六届大会（IUGG）推荐的四个椭球基本参数[3]：

长半径 a=6378140m；

引力常数 $GM=3.986005\times1014\times10^{14}m^3/s^2$；

重力场二阶带球谐系数 $J_2=1.08263\times10^{-3}$；

自转角速度 $\omega=7.292115\times10^{-5}rad/s$。

其定向满足的条件为，椭球短轴平行于地球自转轴，起始大地子午面平行于格林尼治子午面。在我国境内,椭球面和大地水准面最为密合,按多点定位法建立的，大地点高程采用 1956 年黄海高程系作为基准，水准原点高出黄海平均海水面 72.289m[3]。

由于在此之前 1954 年北京坐标系用了很长的时间，很多测绘成果都是采用这个坐标系完成的，但是与 1980 西安坐标系差距较大，给测绘成果的转换带来了不便。所以我国建立了新 1954 年北京坐标系进行过渡，新 1954 年北京坐标系与 1980 西安坐标系有严格的数学转换模型。为了方便两者的相互转化，与 1980 西安坐标系一样，BJ54 $_{新}$也采用了西安原点和 1956 年黄海高程系，定向明确。两者坐标差值在我国大部分地区内小于 5m，部分的地区最大可达 12.9m，所以对于小比例尺地形图的影响可忽略不计[3]。

2.4.3 WGS-84 坐标系

1984 年世界大地坐标系，即 WGS-84，是一个协议地球参考系 CTS[3]。WGS-84 坐标系的原点是地球的质心；Z 轴指向协议地球极方向；X 轴指向 BIH1994.0 零度子午面与赤道的交点；Y 轴与 ZOX 平面构成右手坐标系。其采用的四个基本椭球参数为：

长半径 a=6378137m；

引力常量 GM=3986005×$10^8 m^3/s^2$；

二阶带球谐系数 $\overline{C}_{2.0}$=-484.16685×10^{-6}；

自转角速度 ω=7292115×10^{-11}rad/s。

WGS-84 是由美国国防制图局（DMA）建立的，利用全球定位系统和美国空军的 GPS 卫星跟踪站的观测资料进行了联合解算。WGS-84（G730）与 ITRF92 的符合程度达 10cm 的水平。在这之后又进行了两次精化，提高了精度。分别为 WGS-84(G873)和 WGS-84(G1150)。其中字母“G”表示 GPS 。后面的数字表示的是 GPS 周数[18]。例如：G873 表示第 873 周的第一天。最新的一次精化在 2012 年 2 月 8 日，标号为 WGS-84(G1674) 。此次精化后，WGS-84(G1674)与 ITRF08 在历元 2005.0 处一致[19]。

2.4.4 CGCS2000 坐标系

2000 中国大地坐标系(China Geodetic Coordinate System 2000，CGCS2000)，我们经常称之为 2000 国家大地坐标系。CGCS2000 是全球地心坐标系在我国的具体表现形式，相当于 ITRF1997 下的 2000.0 历元的坐标，误差在 cm 级。作为一个现代地球参考系，它符合国际地球参考系(ITRS)的下列条件[20]：

①原点为地心，是指包括海洋和大气的整个地球的质量中心；

②在广义相对论框架下定义的长度单位；

③Z 轴指向协议地球极方向，X 轴指向 BIH1994.0 零度子午面与赤道的交点，Y 轴与 ZOX 平面构成右手坐标系；

④采取没有整体旋转变化的板块运动模型作为时间基准。

其采用的四个基本椭球参数为：

长半径 a=6378137m；

引力常数 GM=3986004.418×$10^8 m^3/s^2$；

扁率 α=1/298.257222101；

自转角速度 ω=7292115×10^{-11}rad/s。

根据以上述的 4 个常数,利用文献[21]给出的公式,可以算出一系列导出常数。常用的几何参数在上文中的表一中呈现，常用的物理常数：

椭球面正常位 U_0=62636851.7149m^2/s^2；

赤道正常重力 γ_e=9.7803253361m/s^2；

2 阶带谐系数 J_2=0.1082629832258×10^{-2}。

2000 国家大地控制网第一次通过空间测量技术手段将我国各种大地控制网，科学的结合在一起，成为了一个全国统一的国家 3 维大地控制网，不仅提高了精

度，还提高了成果的可靠性。2000 国家 GPS 大地网提供的地心坐标的精度平均优于 3 厘米，为建立我国 3 维地心坐标统提供了高精度的坐标框架[22]。

虽然 CGCS2000 坐标系能够满足目前测绘工作的需要，但是仍然存在着许多不足的地方。第一，CGCS2000 坐标系的控制点的密度仍然需要提高，这一点在我国的西部地区尤为突出；第二，CGCS2000 坐标框架的覆盖范围不够全面，尤其在海洋区域，给海洋上的测绘工作带来不便；第三，CGCS2000 坐标系的精度仍然需要提高，目前的测绘工作需要其提供更为精确的三维变化信息；最后它的点位归算十分困难，CGCS2000 框架采用的是 ITRF97 框架，2000.0 历元，这对于目前广泛采用的 GPS 精密定位，在 ITRF2005 框架和当前历元的点位归算带来不便[23]。

2.5 高斯投影

地球是一个曲面，然而在曲面上研究问题往往不够方便，所以我们需要利用地图数学投影，将椭球面上的要素投影到平面上进行研究。我国采用的是高斯-克吕格投影。高斯-克吕格投影即我们通常所说的“高斯投影”。根据投影性质进行分类，可以把它归为等角横切椭圆柱投影。高斯投影是正形投影，所以可以保证投影前后的角度不变，给测量计算带来方便；又可以保证投影前后图形相似，方便辨认。为了有一个统一的坐标原点，又要保证没有太大的变形，通常需要进行分带投影。我国通常使用 6° 带和 3° 带，在特殊情况下，也可以采用 1.5 度带或任意带。

高斯投影 6° 带，从 0° 子午线起每间隔经差 6° 由西向东进行分带，依次编号，中央子午线的计算公式为 L0=6n-3；高斯投影 3° 带的中央子午线经度 $L_0=3n'$，由此可知，6 度带的中央子午线和分界子午线都是 3 度带的中央子午线；1.5° 带的中央子午线经度为 $L_0=1.5n''$。

在高斯投影平面上，高斯平面坐标系的原点是中央子午线与赤道的交点，x 轴是该带中央子午线的投影，y 轴为赤道的投影，分别以向北和向东为正。因为我国处于北半球，位于高斯投影的一、二区间，x 坐标都是正的，y 坐标有正有负，为了避免出现负的坐标，我们规定在横坐标上加上 500km，并且在前面加上带号[3]。例如：一个 19 度带的自然坐标 y=-3318，其规定坐标为 y=19496682。

2.6 小结

目前我国广泛使用的就是上述四种坐标系，综合这四种我国常用的测量坐标系,在表 2.2 对它们进行了比较：

表 2.2 我国常用四中坐标系比较

坐标系	参考椭球	坐标原点	高程系统
1954 年北京坐标系	克拉索夫斯基椭球体	前苏联的普尔科沃	1956 年黄海高程基准
1980 西安坐标系	1975 年国际椭球体	陕西省泾阳县永乐镇	1956 年黄海高程基准
WGS-84 坐标系	WGS-84 椭球体	地球质心	
CGCS2000 坐标系	2000 参考椭球	地球质心	

在上表 2.2 中可以看到前两个坐标系的原点不在地球质心，采用的参考椭球不一致，但它们都是参心坐标系，都采用 1956 年黄海高程系，其中与我国大地水准面最为接近的坐标系是西安 80 坐标系。我国采用这两种参心坐标系进行测量工作持续了较长时间，也据此获得了许多大地测量成果。但是，随着科学水平的提高和技术手段的发展，对大地原点不作要求的参心坐标系已经不能适应当前测绘工作的精度要求，给测量和测图工作带来不便。所以我国的坐标系统也开始了更新换代，进行地心坐标的相关研究，并在 2008 年 7 月 1 日我国正式启用了一个地心坐标系：2000 国家大地坐标系[1]。

随着以 GNSS 为主导的空间定位技术的迅猛发展，当前使用 GNSS 接收机在全球范围内都能获取 WGS-84 或 ITRF 坐标系下的坐标结果，需要将这些坐标成果统一起来，运用到测绘工作中。因此进行本课题的研究就具有重要的理论意义和实用价值。

第三章 常用大地坐标系统的相互转换

3.1 引言

测量坐标系统的转换由两部分组成，一个是同一基准下坐标系转换，另一个是坐标基准转换。在某一个坐标基准下，把某一点的坐标转换为另一个表现形式的坐标，例如：在 1954 年北京坐标系下，把某一点的空间直角坐标转化为大地坐标，这一类转换称之为同一基准下坐标转换；坐标基准转换则是在不同坐标基准下的同一坐标表现形式的转换，如：1954 年北京坐标系与 1980 西安坐标系下空间直角坐标的转换。坐标基准转换必须要根据已知点的两套坐标求定两个基准之间的转换参数，然后才能进行两个基准之间的坐标转换。此坐标转换软件系统的“基准转换”模块中采用布尔莎七参数模型进行参数的求解。

3.2 同一基准下坐标系转换

3.2.1 大地坐标系转换为空间直角坐标系

椭球面上某一点的坐标信息可以同时用空间直角坐标系和大地坐标系表示，分别用（B，L，H）和（X，Y，Z）表示，所以它们之间一定存在着某种转换方法和规律。如果某一点在椭球面上，则转换公式为：

$$\begin{cases} \mathrm{X} = N\cos B\cos L \\ Y = N\cos B\sin L \\ Z = N(1-e^2)\sin B \end{cases} \tag{3.1}$$

若这一点不在椭球面上，则设大地高为 H，如图 3.1 所示，显然矢量 $\rho = \rho_0 + H\cdot n$，因此有：

$$\rho = \begin{bmatrix} \mathrm{X} \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} (N+H)\cos B\cos L \\ (N+H)\cos B\sin L \\ \left[N(1-e^2)+H\right]\sin B \end{bmatrix} \tag{3.2}$$

其中：N 为卯酉圈的半径，$N = \dfrac{a}{\sqrt{1-e^2\sin^2 B}}$；

a 为地球椭球的长半轴；

b 为地球椭球的短半轴。

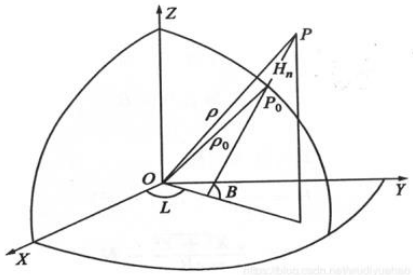

图 3.1 空间直角坐标系与大地坐标系

3.2.2 空间直角坐标系转换为大地坐标系

当某一点的空间直角坐标转换为相应的大地坐标时（XYZ→BLH），基本计算公式有：

$$\begin{cases} \mathrm{L} = \arctan(\dfrac{Y}{X}) \\ \mathrm{tanB} = \dfrac{Z}{\sqrt{X^2+Y^2}} + \dfrac{ce^2 \tan B}{\sqrt{X^2+Y^2}\cdot\sqrt{1+e'^2+\tan^2 B}} \\ H = \dfrac{\sqrt{X^2+Y^2}}{\cos B} - N \end{cases} \tag{3.3}$$

大地纬度的计算通常采用迭代法，$\tan B_1 = \dfrac{Z}{\sqrt{X^2+Y^2}}$，然后用 B 的初值计算 N_1 和 $\sin B_1$，可以简写为：

$$t_{i+1} = t_0 + \frac{Pt_i}{\sqrt{k+t_i^2}} \tag{3.4}$$

其中：$t_0 = \dfrac{Z}{\sqrt{X^2+Y^2}}, p = \dfrac{ce^2}{\sqrt{X^2+Y^2}}, k = 1+e'^2$

所以，$B_i=\arctan t_i$，直到最后两次的 B 值之差小于允许误差为止。实际计算中表明，在保证 H 的计算精度为 0.001m 和 B 的计算精度为 0.00001″ 的情况下，一般需要进行四次迭代左右。

除了迭代算法，还有直接进行公式解算的方法。在熊介编写的《椭球大地测量学》[24]中提到博林曾导出纬度 B 的直接计算公式[25]：

$$\begin{cases} L = \arctan(\dfrac{Y}{X}) \\ B = \arctan(\dfrac{\mathrm{Z} + e'^2 b \sin^3 u}{\sqrt{X^2+Y^2} - e^2 a \cos^3 u}) \\ H = \dfrac{\sqrt{X^2+Y^2}}{\cos B} - N \end{cases} \quad (3.5)$$

式中：e′ 为椭球的第二偏心率，$\mathrm{u} = \arctan(\dfrac{Z \cdot a}{\sqrt{X^2+Y^2} \cdot b})$，$b = \sqrt{(1-e^2)a^2}$

利用（3.5）这个公式比较简单，但是解算精度不高，仅能精确到 0.0001″，但可以基本满足要求，利用博林的直接计算公式进行程序的编写比较简单，可以减少程序运行时间。此坐标转换软件系统中采用迭代法计算纬度 B，这种方法下解算精度较高。

3.2.3 高斯投影正算

高斯投影正算是指由点的大地坐标（B，L）计算获得平面坐标（x，y）的过程。高斯投影是等角投影的一种，必须满足三个条件：中央子午线投影后为直线，中央子午线投影后长度保持不变，具有正形投影条件[3]。

根据以上这三个条件，可以推导出高斯投影的正算公式，为了方便编程，其实用公式可写为：

$$\begin{cases} x = X + \dfrac{l^2}{2} N \sin B \cos B + \dfrac{l^2}{24} N \sin B \cos^3 B (5 - t^2 + 9\eta^2 + 4\eta^4) + \dfrac{l^6}{720} N \sin B \cos^5 B (61 - 58t^2 + t^4) \\ y = lN \cos B + \dfrac{l^3}{6} N \cos^3 B (1 - t^2 + \eta^2) + \dfrac{l^5}{120} N \cos^5 B (5 - 18t^2 + t^4 + 14\eta^2 - 58\eta^2 t^2) \end{cases} \quad (3.6)$$

式中：B 为大地纬度；l=L-L_0，单位是弧度；t=tan B；N 为卯酉圈曲率半径；$\eta = \mathrm{e}' \cdot \cos B$；X 为由赤道起算的子午线弧长。子午线弧长的计算公式如下式 3.7 所示：

$$\mathrm{X} = a_0 B - \sin B \cos B [(a_2 - a_4 + a_6) + (2a_4 - 16a_6/3) \sin^2 B + 16a_6 \sin^4 B / 3] \quad (3.7)$$

式中各参数如下式 3.8 所示：

$$\begin{cases} a_0 = m_0 + \dfrac{1}{2}m_2 + \dfrac{3}{8}m_4 + \dfrac{5}{16}m_6 + \dfrac{35}{128}m_8 \\ a_2 = \dfrac{1}{2}m_2 + \dfrac{1}{2}m_4 + \dfrac{15}{32}m_6 + \dfrac{7}{16}m_8 \\ a_4 = \dfrac{1}{8}m_4 + \dfrac{3}{16}m_6 + \dfrac{7}{32}m_8 \\ a_8 = \dfrac{1}{128}m_8 \end{cases} \tag{3.8}$$

$\mathrm{m}_0 = \mathrm{a}(1-e^2)$，$m_2 = \dfrac{3}{2}e^2\mathrm{m}_0$，$m_4 = \dfrac{5}{4}e^2\mathrm{m}_2$，$m_6 = \dfrac{7}{6}e^2\mathrm{m}_4$，$m_8 = \dfrac{9}{8}e^2\mathrm{m}_6$。

当 l＜3.5°时,公式换算的精度为 0.001m。

3.2.4 高斯投影反算

高斯投影反算是指由一点的高斯平面坐标计算获得对应的大地坐标的过程。其计算公式如下：

$$\begin{cases} \mathrm{B} = B_f - \dfrac{t_f}{2M_f N_f \cos B_f}y^2 + \dfrac{t_f}{24M_f {N_f}^3}(5+3{t_f}^2-9{\eta_f}^2{t_f}^2)y^4 - \dfrac{t_f}{720M_f {N_f}^5}(61+90{t_f}^2+45{t_f}^2)y^6 \\ \mathrm{l} = \dfrac{l}{N_f \cos B_f}y - \dfrac{l}{6{N_f}^3 \cos B_f}(1+2{t_f}^2+{\eta_f}^2)y^3 + \dfrac{l}{120{N_f}^5 \cos B_f}(5+28{t_f}^2+24{t_f}^4+8{\eta_f}^2{t_f}^2)y^5 \end{cases} \tag{3.9}$$

式中，$\mathrm{N_f}$，$\mathrm{t_f}$均为 $\mathrm{B_f}$的函数，可以由 $\mathrm{B_f}$代入求得；$\mathrm{B_f}$是底点纬度，x=X 时子午弧长对应的纬度，可以由子午线弧长反算出来，计算时可以采用迭代算法。

$$\mathrm{F}({B_f}^i) = -\frac{a_2}{2}\sin 2{B_f}^i + \frac{a_2}{4}\sin 4{B_f}^i - \frac{a_2}{6}\sin 6{B_f}^i + \frac{a_2}{8}\sin 8{B_f}^i \tag{3.10}$$

式中：

$$\begin{cases} a_0 = m_0 + \dfrac{1}{2}m_2 + \dfrac{3}{8}m_4 + \dfrac{5}{16}m_6 + \dfrac{35}{128}m_8 \\ a_2 = \dfrac{1}{2}m_2 + \dfrac{1}{2}m_4 + \dfrac{15}{32}m_6 + \dfrac{7}{16}m_8 \\ a_4 = \dfrac{1}{8}m_4 + \dfrac{3}{16}m_6 + \dfrac{7}{32}m_8 \\ a_8 = \dfrac{1}{128}m_8 \end{cases}$$

$\mathrm{m}_0 = \mathrm{a}(1-e^2)$，$m_2 = \dfrac{3}{2}e^2\mathrm{m}_0$，$m_4 = \dfrac{5}{4}e^2\mathrm{m}_2$，$m_6 = \dfrac{7}{6}e^2\mathrm{m}_4$，$m_8 = \dfrac{9}{8}e^2\mathrm{m}_6$。

重复迭代至满足精度，当 l＜3.5°时,公式换算的精度为 0.00001″。

3.3 坐标基准转换

坐标基准转换是指不同坐标系统之间的相互转换，如把某一点的 1954 年北京坐标系坐标转化为 WGS-84 坐标系坐标。由于应用不同的参考椭球体及定位定向技术建立起来的坐标系，都可以转换成空间直角坐标系，因此这几种坐标系之间的基准转换的实质可以理解成为是不同空间直角坐标系之间的相互转换[26]。

如果已知某两个不同空间直角坐标系之间对应的转换参数，即可直接按照相应的计算公式，代入转换参数，进行转换计算。如果两个空间直角坐标系之间的转换参数未知，但是两坐标系间部分公共点的坐标数据已知，就可根据这些已知公共点的坐标信息，利用一定的数学规律进行坐标换算。下面介绍两种转换模型：适用于二维的四参数模型和适用于三维的七参数模型。

3.3.1 四参数转换

四参数坐标转换是指平面二维坐标之间的转换。其中 4 个参数包括：2 个平移参数 ΔX_0 和 ΔY_0，1 个旋转参数 a 和 1 个尺度参数 m。计算这四个参数需要两个以上的公共点，相应的坐标变换公式为[27]：

$$\begin{bmatrix} X_2 \\ Y_2 \end{bmatrix} = (1+m)\begin{bmatrix} \cos a & -\sin a \\ \sin a & \cos a \end{bmatrix}\begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} + \begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix} \tag{3.10}$$

对其进行简化，可以转化为式 3.11 所示：

$$\begin{bmatrix} X_2 \\ Y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & X_1 & -Y_1 \\ 0 & 1 & Y_1 & X_1 \end{bmatrix}\begin{bmatrix} \Delta X_0 \\ \Delta Y_0 \\ C \\ D \end{bmatrix} \tag{3.11}$$

其中：C=mcosa，D=msina。

误差方程个数与公共点个数一致，其误差方程形式可以写成：

$$\begin{bmatrix} V_{X_2} \\ V_{Y_2} \end{bmatrix} = -\begin{bmatrix} 1 & 0 & X_1 & -Y_1 \\ 0 & 1 & Y_1 & X_1 \end{bmatrix}\begin{bmatrix} \Delta X_0 \\ \Delta Y_0 \\ C \\ D \end{bmatrix} - \begin{bmatrix} X_2 \\ Y_2 \end{bmatrix}_{已知值} \tag{3.12}$$

根据最小二乘法的原则，可以列出法方程：

$$V = B\hat{x} - l \tag{3.13}$$

其解为：

$$\hat{x} = (B^T PB)^{-1} B^T PL \qquad (3.14)$$

3.3.2 七参数转换

七参数转换模型有很多种，比较常使用的几种模型有布尔莎模型、武测模型和莫洛琴斯基模型等等。这三种模型均有 7 个转换参数，包括三个平移参数 ΔX_0，ΔY_0 和 ΔZ_0，三个旋转参数 ε_X，ε_Y，ε_Z 和一个尺度变化参数 m。本文重点介绍的是布尔莎七参数模型。

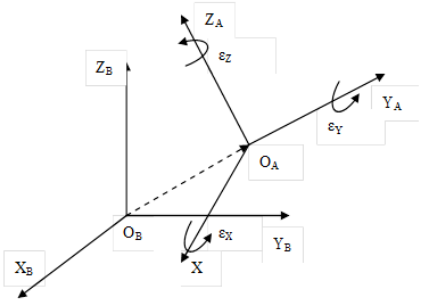

图 3.2 不同空间直角坐标系

如图 3.2 所示，有两个空间直角坐标系 O-XYZ（A）和 O-XYZ（B），这两个坐标系的原点不相一致，所以存在着三个平移参数 ΔX_0，ΔY_0 和 ΔZ_0；又因为这三个坐标轴各不平行，所以存在着三个旋转参数 ε_X，ε_Y，ε_Z；两个坐标系的尺度也不一致，所以存在着一个尺度变化参数 m。相应的坐标变换公式为：

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = (1+m) \begin{bmatrix} 1 & \varepsilon_Z & -\varepsilon_Y \\ -\varepsilon_Z & 1 & \varepsilon_X \\ \varepsilon_Y & -\varepsilon_X & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} + \begin{bmatrix} \Delta X_0 \\ \Delta Y_0 \\ \Delta Z_0 \end{bmatrix} \qquad (3.15)$$

为了求得这七个转换参数，至少需要 3 个公共点，莫洛琴斯基模型和武测模型的旋转参数与布尔莎模型一致，但是三个平移参数各不相同。这三种转换模型各有其特点，但都经过旋转，放缩和平移等步骤，从本质上来说是一致的。因为这类模型都含有七个转换参数，所以通常称之为七参数转换模型。

当公共点多于三个的时候，就需要按照最小二乘法进行平差，误差方程的个数与公共点的个数一致。其误差方程形式可以写成：

$$\begin{bmatrix} \mathrm{V}_{X_2} \\ \mathrm{V}_{Y_2} \\ \mathrm{V}_{Z_2} \end{bmatrix} = -\begin{bmatrix} 1 & 0 & 0 & X_1 & 0 & -Z_1 & Y_1 \\ 0 & 1 & 0 & Y_1 & Z_1 & 0 & -X_1 \\ 0 & 0 & 1 & Z_1 & -Y_1 & X_1 & 0 \end{bmatrix} \begin{bmatrix} \Delta X_0 \\ \Delta Y_0 \\ \Delta Z_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} + \begin{bmatrix} \mathrm{X}_2 \\ Y_2 \\ Z_2 \end{bmatrix}_{\text{已知值}} \tag{3.16}$$

其中：$a_1 = m+1$，$a_2 = a_1\varepsilon_{\mathrm{X}}$，$a_3 = a_1\varepsilon_{\mathrm{Y}}$，$a_4 = a_1\varepsilon_{\mathrm{Z}}$。

根据最小二乘法的原则，可以列出法方程：

$$B^T PB\delta X + B^T PL = 0 \tag{3.17}$$

其解为：

$$\delta \mathrm{X} = -(B^T PB)^{-1} B^T PL \tag{3.18}$$

由于各公共点的坐标存在误差，对于我们求解转换参数有影响，为了提高准确度，我们在选择公共点时应尽量选择一定数量，分布均匀并有较大覆盖区域的公共点[28]。

在测量工作中，我们还需要进行改正数的计算。第一步先计算出公共点的改正数，数值等于已知坐标值减去转换坐标值；第二步计算出非公共点的改正数，式子如下所示：

$$\mathrm{V}' = \frac{\sum_{1}^{n} \mathrm{P_i} V_i}{\sum_{1}^{n} \mathrm{P_i}} \tag{3.19}$$

其中，P 为权，$P_i = \frac{1}{S_i^{\,2}}$。$\mathrm{S_i}$ 指的是非公共点与公共点之间的距离。

第四章 常用坐标系转换软件设计与实现

4.1 转换软件系统算法实现流程

4.1.1 空间直角坐标与大地坐标转换算法实现流程图

其软件实现的流程图如图 4.7 所示：

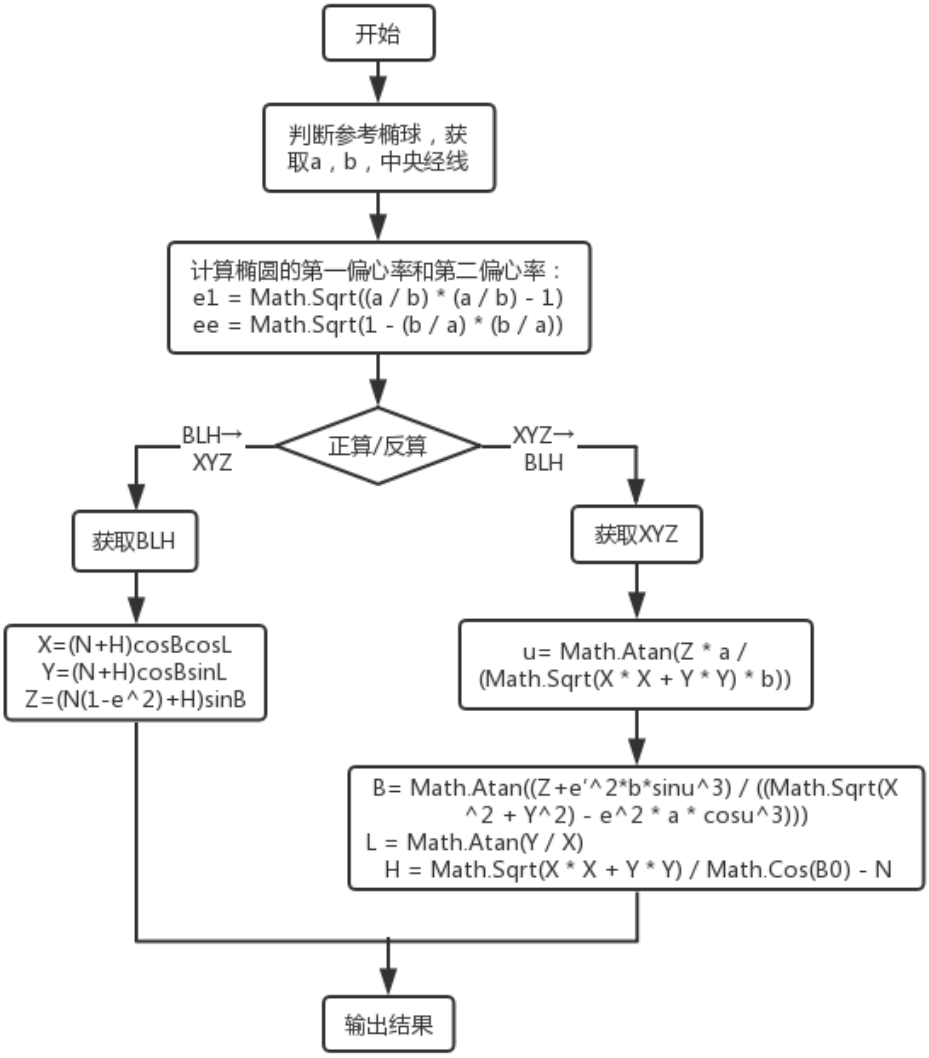

图 4.7 空间直角坐标系与大地坐标系之间转换实现流程图

4.1.2 高斯平面坐标与大地坐标转换算法实现流程图

高斯平面正反算软件实现的流程图如图 4.8 所示：

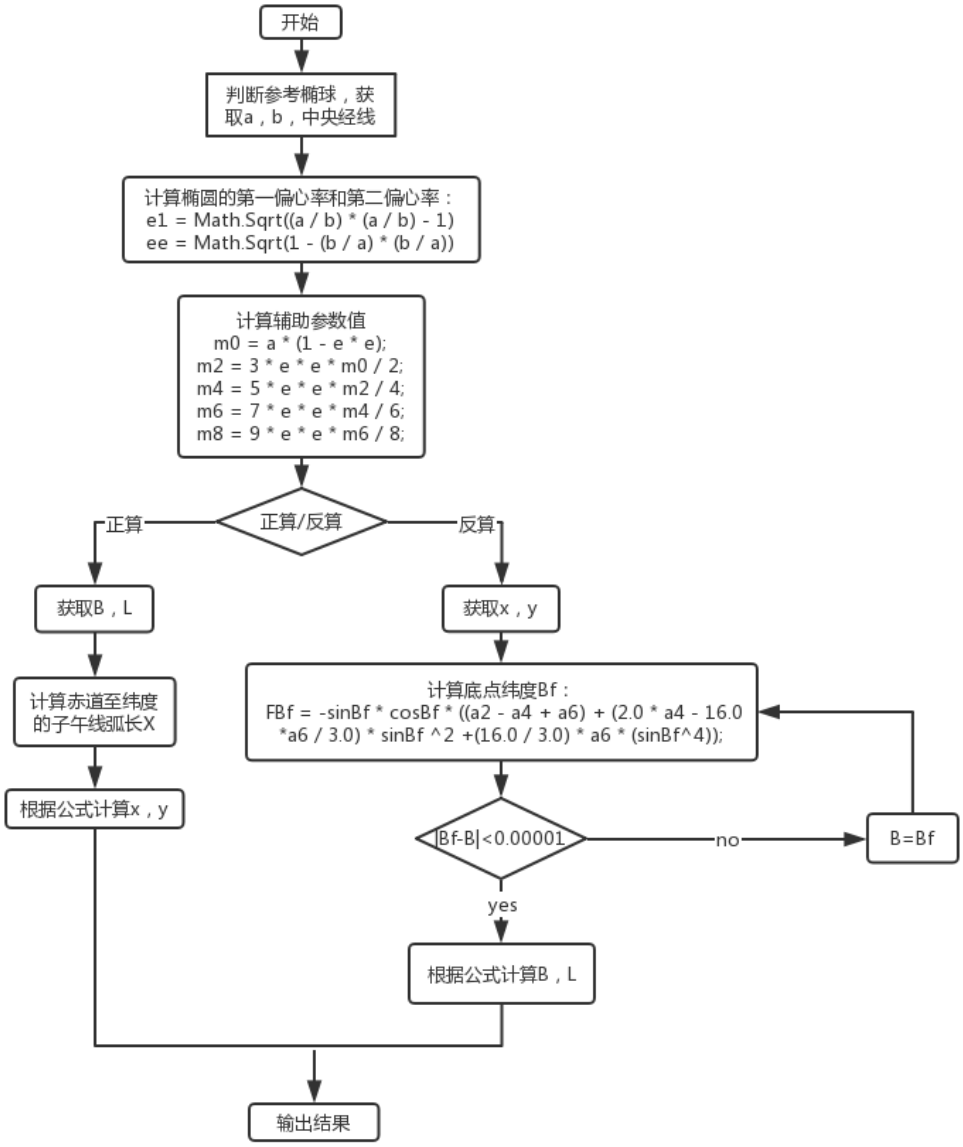

图 4.8 高斯平面坐标与大地坐标转换实现流程图

4.1.3 不同空间直角坐标系转换算法实现流程图

不同空间直角坐标系之间转换的流程图如图 4.9 所示。

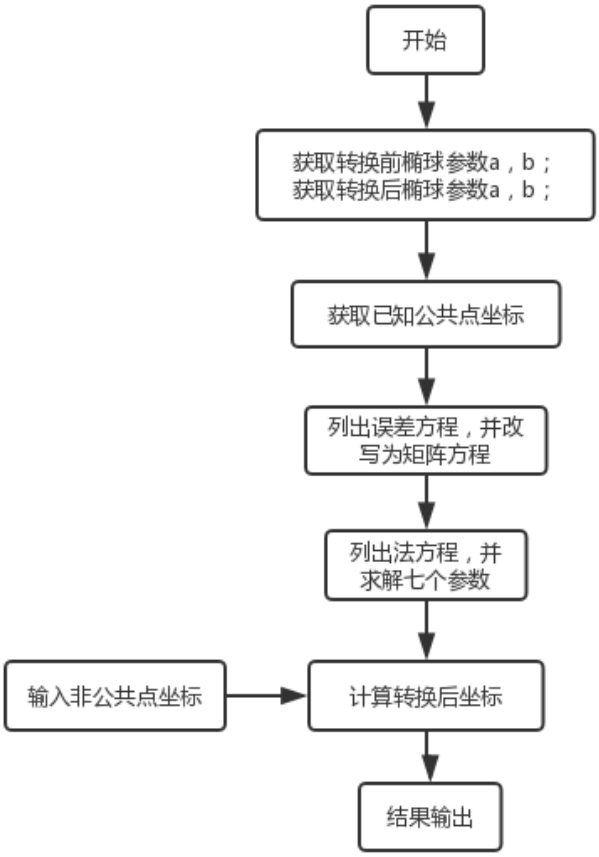

图 4.9 不同空间直角坐标系之间转换实现流程图

4.2 转换软件系统实现

4.2.1 系统开发平台与工具

本软件的开发平台是 Visual Studio 2010。Visual Studio 是一个完整的开发工具，它的功能很齐全，包含许多软件开发工具，把所有语言开发环境都集成在一起，便于用户选择和使用。本软件采用 C#窗口应用台程序进行编写。C#语言简洁明了，运行效率高。在窗口应用台程序中，可以自主选择窗口空间，进行界面设计，方便快捷。

4.2.2 软件基本要求

1、设计软件应界面友好，功能齐全，方便操作。

2、可以实现同一坐标基准下的坐标转换。其中包括：空间直角坐标与大地坐标之间的相互转化；大地坐标与平面直角坐标之间的相互转换（即高斯投影正反算）。

3、可以实现坐标基准转换，即不同空间直角坐标系之间的转换，采用布尔莎七参数模型进行转换参数的求解。

4、用户可以自主选择参考椭球（克拉索夫斯基椭球、1975 年国际椭球体、WGS-84 椭球体、CGCS2000 椭球体），并展示所选参考椭球的基本几何参数（长半轴、扁率……）。

5、在高斯投影正反算模块，用户可以进行带宽和中央子午线的选择。

6、可以实现坐标的单点转换与批量转换。

7、能够方便用户对坐标数据进行添加（包括直接输入和文件输入），并且对计算而得的转换参数进行展示。

8、计算结果可以在系统界面中清晰地展示，也可保存到文本文档中。

4.2.3 界面设计

整体界面要清晰、简洁，方便用户的功能选择，本程序共设计为 3 个窗口，一个主窗口，和两个子窗口，分别为同一基准下坐标转换和坐标基准转换得出窗口。

主界面如图 4.1 所示：

图 4.1 软件主界面

主界面中包含两个功能按钮，分别为同一基准下坐标转换和坐标基准转换。点击这两个按钮分别会跳转到新的窗口界面。

同一基准下坐标转换界面如图 4.2 所示：

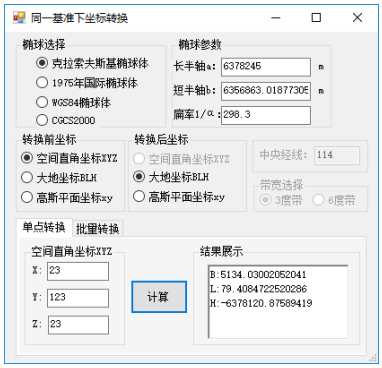

图 4.2 同一基准下坐标转换界面

此界面分为四个板块，分别是椭球选择、椭球参数、坐标系统选择和结果输出。板块一中共有四个椭球体供用户选择，并同时会在板块二中展示出所选参考椭球的部分基本几何参数。用户可以在板块三中对转换前后的坐标系统进行选择，若选择高斯投影正反算，则需要另外输入中央经线并进行带宽选择。板块四包括单点转换和批量转换两部分，输入坐标后，点击计算，可以在结果展示中清晰地看见转换结果，如图 4.3 所示。

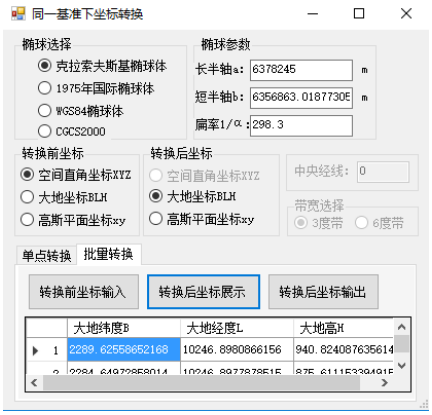

图 4.3 不同基准下坐标转换界面

在板块四中选择批量转换，点击“转换前坐标输入”按钮，会显示出一个坐标加载的对话框，需要用户进行坐标文本文档的选择，如图 4.4 所示：

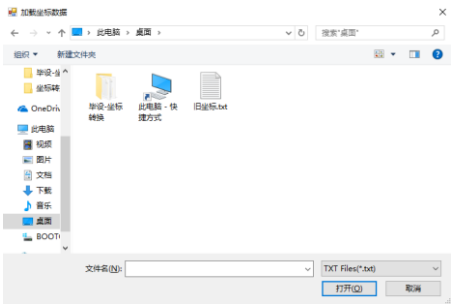

图 4.4 坐标加载界面

其中设置的文本文档输入格式如图 4.5 所示，各个坐标之间用逗号隔开。

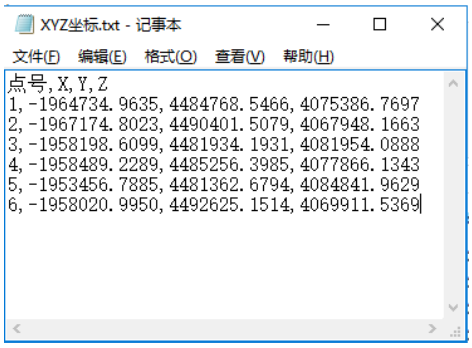

图 4.5 坐标输入文本文档

点击“转换后坐标展示”按钮，可以在结果展示中清楚看到坐标转换结果，如图 4.3 所示。也可以点击“转换后坐标输出”按钮，则会生成一个文本文档，在其中可以清楚地看到坐标转换后的结果。

而在主界面中点击“坐标基准转换按钮”时，则会跳转到另一个界面，如图 4.6 所示：

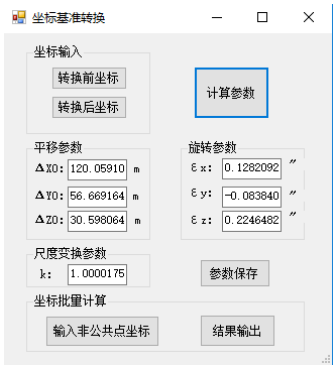

图 4.6 坐标基准转换界面

此界面中分为三个板块，分别为坐标输入，参数展示，和坐标计算输出。在板块一中点击“转换前坐标输入”按钮，会跳出一个坐标载入的对话框，可以选择转换前的已知坐标，文本文档的格式如图 4.5 所示，点击“转换后坐标”按钮，也会跳出一个坐标载入的对话框，选择转换后的已知坐标。点击“计算参数按钮”，就会在板块二中展示计算所得的七个参数，包括三个平移参数 ΔX_0，ΔY_0 和 ΔZ_0，三个旋转参数 ε_X，ε_Y，ε_Z，和一个尺度因子 k（k=m+1）。点击“参数保存”按钮，会生成一个文本文档，将计算而得的七个参数输出，在其中可以清楚地看到参数计算的结果，如图 4.7 所示：

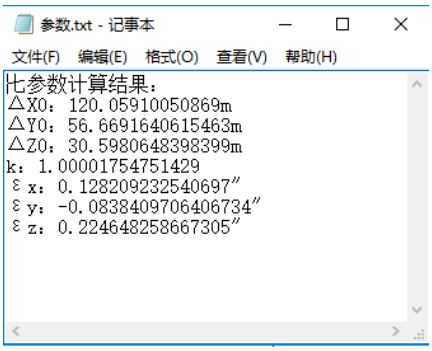

图 4.7 参数计算输出结果

在在板块三中需要输入非公共点坐标，点击“结果输出”按钮，系统会将转换后的坐标结果保存到一个文本文档中，如 4.8 所示。其中还包括采用配置法计算而得的转换改正数。

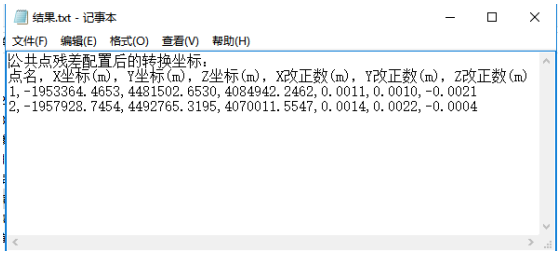

图 4.8 坐标转换输出结果

第五章 应用与测试

5.1 同一基准下坐标转换模块测试

本节将对常用的坐标形式变换模块进行测试，此模块中包括四个常用坐标系（如：北京 54 坐标系，西安 80 坐标系，WGS-84 坐标系和 CGCS2000）下的不同坐标形式变换、输入和输出坐标数据，还分为单点转换和批量转换。对这些功能进行测试，并且比较坐标计算的精度。

5.1.1 大地坐标与空间直角坐标转换模块

此版块测试以 WGS-84 坐标系下已知点坐标数据作为检验对象，现已知 5 个点的 WGS-84 坐标系下的大地坐标和空间直角坐标，如下表所示：

表 5.1 WGS-84 坐标系下已知点大地坐标与空间直角坐标

点号	X	Y	Z	B	L	H
Z2	-196147*.**62	511840*.**28	325063*.**12	3*.**1852539	11*.**0416788	21*.**943
Z4	-196169*.**04	511874*.**37	325008*.**28	3*.**5643222	11*.**0718747	26*.**18
Z7	-196186*.**56	511852*.**10	325022*.**99	3*.**0280682	11*.**1632667	21*.**60
Z8	-196195*.**22	511794*.**62	325101*.**11	3*.**3345925	11*.**2719100	17*.**20
Z9	-196188*.**40	511821*.**64	325060*.**01	3*.**1787111	11*.**2127122	16*.**49

（1）XYZ→BLH

输入上述已知点的空间直角坐标（X，Y，Z），经过此软件解算，获得的大地坐标如下图所示：

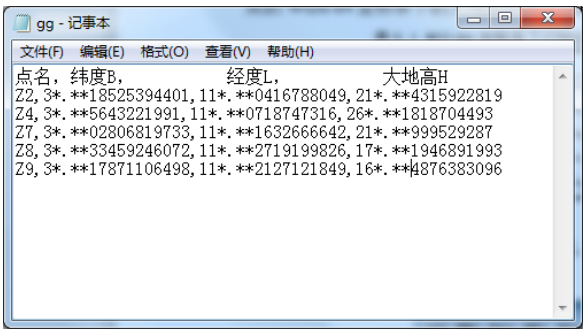

图 5.1 转换后的大地坐标

与已知点坐标相比较，其精度可以达到 10^{-6} 秒，大地高的精度可达 10^{-4} 米，计算结果均在误差允许的范围之内。系统测试结果表明“XYZ→BLH”转换模块所采用的模型正确，程序设计合理。

（2）BLH→XYZ

输入上述已知点的大地坐标（B，L，H），经过此软件的批量转换解算，获得空间直角坐标如下图所示：

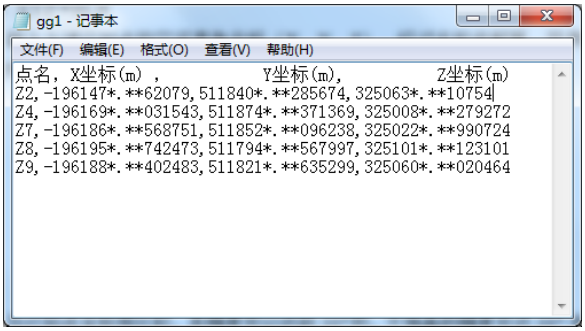

图 5.2 转换后的空间直角坐标

与已知点空间直角坐标相比较，其精度可以达到 10^{-4} 米，计算结果均在误差允许的范围之内。系统测试结果表明“BLH→XYZ”转换模块所采用的模型正确，程序设计合理。

5.1.2 高斯投影正反算模块测试

此版块测试以 WGS-84 坐标系下已知点坐标数据作为检验对象，现已知 3 个点的 WGS-84 坐标系下的大地坐标和高斯投影坐标，如下表所示：

表 5.2 WGS-84 坐标系下已知点大地坐标与高斯投影坐标

点号	B	L	X	Y
1	3*.**0021702	11*.**5514230	341250*.**63	1949668*.**84
2	3*.**1842140	11*.**0410602	341306*.**27	1949692*.**05
3	3*.**3486021	11*.**1410558	341357*.**44	1949718*.**81

（1）高斯投影正算

输入上述已知点的大地坐标，选择 111° 经线作为中央子午线，选择 6 度带，经过此软件批量坐标解算，获得高斯投影坐标如下图所示：

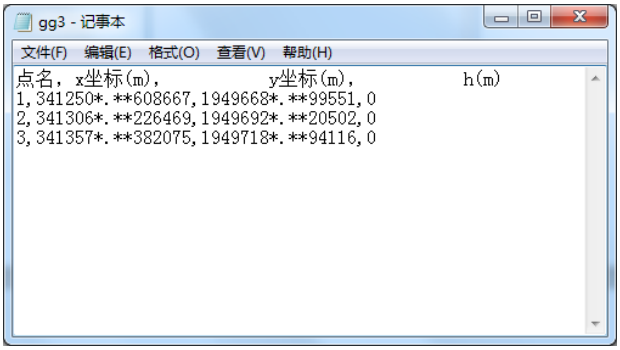

图 5.2 转换后的高斯投影坐标

转换后与已知坐标之差如表 5.3 所示：

表 5.3 WGS-84 坐标系下高斯投影正算坐标误差

点号	ΔX（米）	ΔY（米）
1	0.0003	0.0015
2	0.0005	0.0015
3	0.0005	0.0013

与已知点高斯平面直角坐标相比较，其精度可以达到 10^{-4} 米，计算结果均在误差允许的范围之内。系统测试结果表明“高斯投影正算”转换模块所采用的模型正确，程序设计合理。

（2）高斯投影反算

输入上述已知点的高斯投影坐标，选择 111° 经线作为中央子午线，选择 6 度带，经过此软件批量坐标解算，获得大地坐标如下图所示：

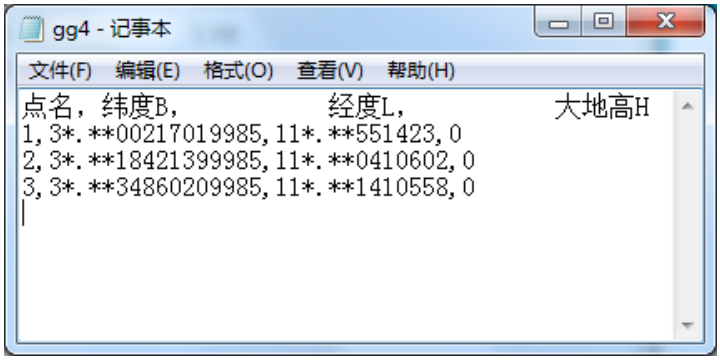

图 5.3 转换后的大地坐标

与已知点大地坐标相比较，其精度可以达到 10^{-5}″，计算结果均在误差允许的范围之内。系统测试结果表明“高斯投影反算”转换模块所采用的模型正确，程序设计合理。

5.2 坐标基准转换模块测试

不同基准坐标转换模块主要包括常用坐标系之间空间直角坐标的相互转换、参数的计算输出、输入转换源数据和输出转换结果，此版块测试以已知点的两套坐标作为检验对象。

现有 4 个已知点的公共坐标，分别编号为 A、B、C、D。还有 2 个已知待转换点的坐标，分别编号为 1、2，如表 5.4 所示：

表 5.4 已知点坐标

点号	X1	Y1	Z1	X2	Y2	Z2
A	-196473*.**35	448476*.**66	407538*.**97	-196464*.**59	448490*.**60	407548*.**81
B	-196717*.**23	449040*.**79	406794*.**63	-196708*.**60	449054*.**60	406804*.**09
C	-195819*.**99	448193*.**31	408195*.**88	-195810*.**01	448207*.**89	408205*.**11
D	-195848*.**89	448525*.**85	407786*.**43	-195839*.**49	448539*.**50	407796*.**71
1	-195345*.**85	448136*.**94	408484*.**29	-195336*.**91	448150*.**50	408494*.**49
2	-195802*.**50	449262*.**14	406991*.**69	-195792*.**51	449276*.**50	407001*.**30

将已知公共点的转换前后坐标分别输入，点击计算参数，软件计算获得的七个参数如下图所示：

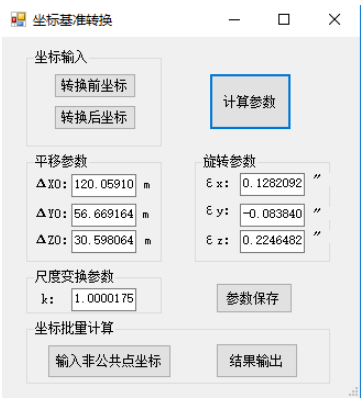

图 5.3 计算的七参数

然后输入需要转换的坐标，经过此软件批量坐标解算，获得转换后的大地直角坐标如下图所示：

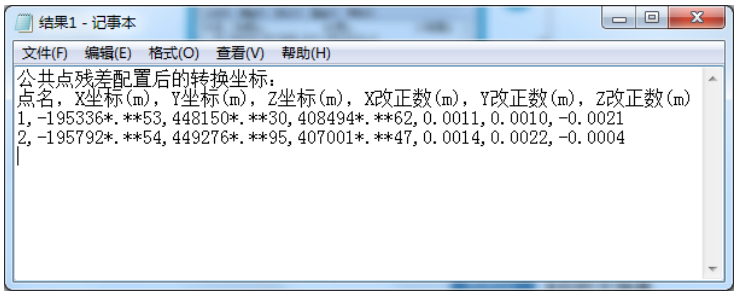

图 5.4 转换后的大地坐标及改正数

其坐标转换计算的改正数如下表所示：

表 5.5 坐标转换的改正数

点号	VX（米）	VY（米）	VZ（米）
1	0.0011	0.0010	-0.0021
2	0.0014	0.0022	-0.0004

然后将计算获得的坐标数据与表 5.4 中的坐标数值进行对比，并计算差值。计算获得的坐标之差如表 5.6 中所示：

表 5.6 转换后坐标与已知坐标之差

点号	ΔX（米）	ΔY（米）	ΔZ（米）
1	-0.0062	-0.002	-0.0187
2	0.0097	0.0145	-0.0083

计算结果均在误差允许的范围之内。系统测试结果表明“不同基准坐标转换”模块所采用的模型正确，程序设计合理。

5.3 小结

本章主要对常用大地坐标系转换软件系统的主要功能界面和功能模块的性能和精度进行了测试，主要包括软件的主要功能界面、四个坐标系统（北京 54 坐标系；西安 80 坐标系；WGS-84 坐标系；CGCS2000 坐标系）下的不同坐标形式变换模块、坐标基准转换模块。并对各个模块中的小功能板块进行检验测试，包括输入和输出坐标数据、单点转换和批量转换、椭球体的选择和椭球参数的展示、中央子午线和带宽的选择等等。

经测试检验，此软件系统性能良好，精度可靠，模型正确，程序设计合理。

第六章 总结

坐标系统是大地测量学的基础，测量和描绘地面点的位置是大地测量学中一项最基础的目标与任务。只有认清并掌握各种坐标系统，才能了解地球椭球，顺利开展测绘工作。随着空间测量技术的定位精度不断改进，新的国际地球参考框架中点的精度已经可以达到毫米级以上，如果仍然采用、参心、低精度的大地坐标系统，必然会带来越来越多的不协调和问题[29]。

因此大地坐标系的更新换代,是经济建设、国防建设、社会发展和科技发展的客观需要。而选择不同的坐标系统，其坐标值也必然不同，我们需要把这些不同的大地网统一起来，所以要研究不同坐标系统之间的转换关系，探讨各种转换参数的求解方法。

本文对地球椭球及测量坐标系统的基本理论做了一个基本介绍。重点研究了同一坐标基准和不同坐标基准坐标转换的方法，设计与实现了本转换软件系统，并通过实际的坐标数据检验其精度和可行性。经检验，各个功能板块模型正确，计算结果均在误差允许的范围之内。在同一基准下进行坐标转换，经纬度误差在10^{-5}秒以内，坐标误差在10^{-4}米以内；进行坐标基准转换时，坐标精度可达毫米级。能够满足实际测绘工作中的精度要求。

鉴于时间和知识水平限制，本文还存在不足：

1、界面还有待进一步改进和简化，功能按钮可以进行完善；

2、添加坐标转换系统的功能，并进行完善,以适应更多的用户需求。

3、可以对更多的坐标系统进行研究，在软件中实现转换。

4、在坐标基准转换中，可以添加多个数学模型，并研究它们之间的精度差别。

参考文献

[1] 陈俊勇.中国现代大地基准——中国大地坐标系统 2000(CGCS 2000)及其框架[J].测绘学报,2008,37(3):269-270.

[2] 魏子卿.2000 中国大地坐标系及其与 WGS84 的比较[J].大地测量与地球动力学,2008,28(5):1-4.

[3] 孔祥元,郭际明,刘宗泉.大地测量学基础[M].武汉:武汉大学出版社,2005.

[4] 龙文星,胡川.浅谈 2000 国家大地坐标系统[J].矿山测量,2009(06):33-35.

[5] CGCS2000 参考框架维持、更新理论与方法研究[D].武汉大学，2018：3-6.

[6] 门葆红,董文亮,孙付平,刘帅.国际地球参考框架建立与维持的研究进展[J].测绘科学,2016,41(02):20-25.

[7] 白鸥,朱筱虹.现代地心坐标系的发展与展望[J].测绘科技信息交流论文集.成都成都地图出版社,2007:494-497.

[8] 陈俊勇. 邻近国家大地基准的现代化[J].测绘通报,2003,10:1-3.

[9] 杨元喜,徐天河.不同坐标系综合变换法[J].武汉大学学报,2001,26(6):509-512.

[10] 孔祥元,梅是义.控制测量学下册[M].武汉:武汉大学出版社,2002:1-3.

[11] 宁津生.现代大地测量参考系统[J].测绘学报,2002(31):7-9.

[12] 杨元喜.中国大地坐标系建设主要进展[J].测绘通报,2005(1):6-9.

[13] 王百胜.传统参心坐标系与国家 2000 大地坐标系转换问题研究[J].中国高新科技,2019(2):22-24.

[14] 范志坚,杜仕龙,付蓉.过渡期内现行四种常用大地坐标系的分析比较[J].测绘科学,2010(35):25-27.

[15] 许家琨.常用大地坐标系的分析比较[J].海洋测绘,2005,25(6):71-73.

[16] 张述清,李永云.地方独立坐标系统的建立及其实现[J].测绘工程.2007,16(4):22-24.

[17] 陈士银.建立地方独立坐标系的方法[J].测绘通报,1997(10):5-7.

[18] 李征航,黄劲松.GPS 测量与数据处理[M].武汉:武汉大学出版社,2016.

[19] 赵忠海,蒋志楠,朱李忠.WGS-84(G1674)与 CGCS2000 坐标转换研究[J].测绘与空间地理信息,2015,38(4):189-191.

[20] 魏子卿.2000 中国大地坐标系[J].大地测量与地球动力学,2008,28(6):1-5.

[21] 魏子卿.关于 2000 中国大地坐标系的建议[J].大地测量与地球动力学,2006,26(2):1-4.

[22] 陈俊勇,杨元喜,王敏,张燕平,唐颖哲,李辉,程鹏飞,孙凤华,张鹏,郭春喜.2000 国家大地控制网的构建和它的技术进步[J].测绘学报,2007,36(01):1-8.

[23] 杨元喜.2000 中国大地坐标系[J].科学通报,2009,54(16):2271-2276.

[24] 雄介.椭球大地测量学[M].北京:解放军出版社,1988,27-30.

[25] Bowring,B.,R.,Transformation from spatial to geographical coordinates,Survey Review No 181,1976,（译文载《测绘科技通讯》,1978,No 4).

[26] 杨蕊.测量坐标系统转换方法研究与实现[D].长安大学,2017：17-18.

[27] 江华,段太生,周适.四参数坐标转换研究及应用分析[J].发展与创新,2019(3):219-220.

[28] 朱小美,张官进,朱楠.基于 MATLAB 的布尔莎模型七参数解算实现[J].北京测绘,2015(5):61-63.

[29] 陈俊勇.关于中国采用维坐标系统的探讨[J].测绘学报,2003,32:283-288.

致 谢

在论文的研究过程中和软件的编写中，我遇到了很多困难，但在老师和同学的帮助下，我都一一克服了。在此我想衷心感谢我的指导老师李玮老师对我耐心深入的指导和帮助，并且感谢来自重庆市勘测院的胡波教授，支持并指导我的毕业设计论文。

论文能够顺利完成，与许多帮助和指导我的老师和同学是分不开的，在此向他们表示衷心的感谢。

感谢论文列出的参考文献的所有作者。

附录

1、椭球参数展示相应的代码为：

```
class
    {
        private static double _a1 = 6378245;
        private static double _b1 = 6356863.0187730473;
        private static double _af1 = 298.3;//BJ54
        ……
        public static void klsfsj(out double a, out double b, out double af)
        {
            a = _a1;
            b = _b1;
            af = _af1;
        }
        ……
    }
```

2、角度转换相应代码为：

（1）角度化弧度：

```
        private static double DMSTORAD(double dmsAngle1)
        {
            double a = Math.Floor(dmsAngle1);
            double b = Math.Floor((dmsAngle1 - a) * 100);
            radAngle = (a+b/60+(dmsAngle1-a-b/100)*10000/3600 ) * Math.PI / 180.0;
            return radAngle;
        }
```

（2）弧度化角度：

```
        private static double RADTODMS(double radAngle)
        {
            dmsAngle = radAngle * 180 / Math.PI;
            double a = Math.Floor(dmsAngle);
            double b = Math.Floor((dmsAngle - a) * 60);
            double c=((dmsAngle -a)*60-b)*60;
            dmsAngle = a + b / 100 + c / 10000;
            return dmsAngle;
        }
```

3、文件读入相应程序代码：

```
    private void button1_Click_1(object sender, EventArgs e)
        {
            OpenFileDialog MyDlg = new OpenFileDialog();
```

```
            MyDlg.Title = "加载坐标数据";
             MyDlg.Filter = "TXT Files(*.txt)|*.txt";
            string pathname = null;
            if (MyDlg.ShowDialog() == DialogResult.OK)
            {
                pathname = MyDlg.FileName;
            }
            else
            {
                MessageBox.Show("加载文件失败", "错误");
                return;
            }
            var reader = new StreamReader(pathname);
            string buf = reader.ReadLine();
            for (int i = 0; i < 20; i++)
            {
                buf = reader.ReadLine();
                length = i;
                if (buf == null) break;
                var arr = buf.Split(',');
                name[i] = arr[0];
                oldX[i, 0] = double.Parse(arr[1]);//X
                oldX[i, 1] = double.Parse(arr[2]);//Y
                oldX[i, 2] = double.Parse(arr[3]);//Z
            }
            reader.Close();
        }
```

4、文件输出相应程序代码：

```
SaveFileDialog dialog = new SaveFileDialog();
            dialog.Filter = "文本文件 t|*.txt";
            if (dialog.ShowDialog() == DialogResult.OK)
            {
                string fileName = dialog.FileName;
                FileStream fs = File.Open(fileName, FileMode.Create, FileAccess.Write);
                StreamWriter wr = new StreamWriter(fs);
                if (radioButton4.Checked)
                {
                    wr.WriteLine("点名，X 坐标 m，          Y 坐标 m,          Z 坐标 m");
                }
                if (radioButton5.Checked)
                {
                    wr.WriteLine("点名，纬度 B，          经度 L，          大地高 H");
```

```
                }
                if (radioButton6.Checked)
                {
                    wr.WriteLine("点名，x 坐标 m，          y 坐标 m，          h(m)");
                }
                for (int i = 0; i < length; i++)
                {
                    wr.WriteLine(name[i] + "," + newX[i, 0] + "," + newX[i, 1] + "," + newX[i, 2]);
                }
                wr.Flush();
                wr.Close();
                fs.Close();
            }
        }
```

5、大地坐标转换为空间直角坐标，在程序中相应的代码为：

```
public static void LBH2XYZ(double _a,double _b,double B, double L, double H, out double X, out double Y, out double Z)
{       ……
        double B1 =  大地_空间.DMSTORAD(B);
        double L1 =  大地_空间.DMSTORAD(L);
        double e2=((a*a-b*b)/(a*a));
        double N = a / Math .Sqrt (1-e2*Math.Sin (B1)*Math.Sin (B1));
        X = (N + H) * Math.Cos(B1) * Math.Cos(L1);
        Y = (N + H) * Math.Cos(B1) * Math.Sin(L1);
        Z = (N * (1 - e2) + H) * Math.Sin(B1);
}
```

6、空间直角坐标转换为大地坐标，在程序中相应的代码为：

```
public static void XYZ2BLH(double _a,double _b,double X, double Y, double Z, out double B, out double L, out double H)
    {
        ……
        double e2 = (a * a - b * b) / (a * a);
        double epie2 = (a * a - b * b) / (b * b);
        double L0 = Math.Atan(Y / X);
        L =  大地_空间.RADTODMS(L0);
        double t0 = Z / Math.Sqrt(X * X + Y * Y);
        double c = (a * a) / b;
        double p = (c * e2) / (Math.Sqrt(X * X + Y * Y));
        double k = 1 + epie2;
        do
        {
            t = Z / (Math.Sqrt(X * X + Y * Y)) + (p * t0) / (Math.Sqrt(k + t0 * t0));
            B0 = Math.Atan(t0);
```

```
                B00 =  大地_空间.RADTODMS(B0);
                B1 = Math.Atan(t);
                B11 =  大地_空间.RADTODMS(B1);
                t0 = t;
            } while (Math.Abs(B11-B00 ) > 0.000000001);
            B = B11;
            double N = a / Math.Sqrt(1 - e2 * Math.Sin(B1) * Math.Sin(B1));
            H = (Math.Sqrt(X * X + Y * Y) / Math.Cos(B1)) - N;
        }
```

7、高斯投影正算，在程序中相应的代码为：

```
 public static void zhengsuan(double daihao,double _a,double _af,double B, double L, double L0, out double x, out double y)
        {
            ……
            X = a0 * radlat - sb * cb * ((a2 - a4 + a6) + (2 * a4 - 16 * a6 / 3) * sb * sb + 16 * a6 * sb * sb * sb * sb / 3.0);
            c = a * a / b;
            v = Math.Sqrt(1 + e1 * e1 * cb * cb);
            N = c / v;
            x = X + N * sb * cb * l * l / 2 + N * sb * cb * cb * cb * (5 - t * t + 9 * ita * ita + 4 * ita * ita * ita * ita) * l * l * l * l / 24 + N *
sb * cb * cb * cb * cb * cb * (61 - 58 * t * t + t * t * t * t * t) * l * l * l * l * l * l * l / 720;
            y = N * cb * l + N * cb * cb * cb * (1 - t * t + ita * ita) * l * l * l / 6 + N * cb * cb * cb * cb * cb * (5 - 18 * t * t + t * t * t * t * t +
14 * ita * ita - 58 * ita * ita * t * t) * l * l * l * l * l * l / 120;
            y =y+_FE+daihao*1000000;//规定坐标
        }
```

8、高斯投影反算，在程序中相应的代码为：

```
 public static void fansuan(double daihao,double _a,double _af,double x, double y, double L0, out double dmsB, out double dmsL)
        {   ……
            Bf0 = X / a0;
            do
            {
                sinBf = Math.Sin(Bf0); cosBf = Math.Cos(Bf0);
                FBf = -sinBf * cosBf * ((a2 - a4 + a6) + (2.0 * a4 - 16.0 * a6 / 3.0) * sinBf * sinBf +(16.0 / 3.0) * a6 * (sinBf * sinBf *
sinBf * sinBf));
                Bf1 = (X - FBf) / a0;
                dB = Bf1 - Bf0;
                Bf0 = Bf1;
            } while (Math.Abs(dB * 180.0 / Math.PI * 3600.0) > 0.00001);
            bf = Bf1;
            c = a * a / b;
            v = Math.Sqrt(1 + e1 * e1 * Math.Cos(bf) * Math.Cos(bf));
            Nf = c / v;
            Mf = c / (v * v * v);
            tf = Math.Sin(bf) / Math.Cos(bf);
```

```
            itaf = e1 * Math.Cos(bf);
            dietaB = tf * y * y / (2 * Mf * Nf) - tf * (5 + 3 * tf * tf + itaf * itaf - 9 * tf * tf * itaf * itaf) * y * y * y * y / (24 * Mf * Nf * Nf
* Nf) + (61 + 90 * tf * tf + 45 * tf * tf * tf * tf) * y * y * y * y * y * y / (720 * Mf * Nf * Nf * Nf * Nf * Nf);
            dietal = y / (Nf * Math.Cos(bf) + (1 + 2 * tf * tf + itaf * itaf) * Math.Cos(bf) * y * y / (6 * Nf)) + (5 + 44 * tf * tf + 32 * tf * tf
* tf * tf - 2 * itaf * itaf - 16 * itaf * itaf * tf * tf) / (360 * Nf * Nf * Nf * Mf * Mf * Math.Cos(bf));
            B = bf - dietaB;
            L = l0 + dietal;
            dmsB =  高斯正反算.RADTODMS(B);
            dmsL =  高斯正反算.RADTODMS(L);
        }
```

9、矩阵乘法的相应代码：

```
public static double[,] MultiplyMatrix(double[,] MatrixEin, double[,] MatrixZwei)//矩阵乘法
        {
            double[,] MatrixResult = new double[MatrixEin.GetLength(0),
MatrixZwei.GetLength(1)];
            for (int i = 0; i < MatrixEin.GetLength(0); i++)
            {
                for (int j = 0; j < MatrixZwei.GetLength(1); j++)
                {
                    for (int k = 0; k < MatrixEin.GetLength(1); k++)
                    {
                        MatrixResult[i, j] += MatrixEin[i, k] * MatrixZwei[k, j];
                    }
                }
            }
            return MatrixResult;
        }
```

10、矩阵转置的相应代码：

```
    public static double[,] Transpose(double[,] iMatrix)//矩阵转置
        {
            int row = iMatrix.GetLength(0);
            int column = iMatrix.GetLength(1);
            double[,] TempMatrix = new double[row, column];
            double[,] iMatrixT = new double[column, row];
            for (int i = 0; i < row; i++)
            {
                for (int j = 0; j < column; j++) {TempMatrix[i, j] = iMatrix[i, j]; }
            }
            for (int i = 0; i < column; i++)
            {
                for (int j = 0; j < row; j++){iMatrixT[i, j] = TempMatrix[j, i];}
            }
```

```
            return iMatrixT;
        }
```

11、矩阵的逆运算的相应代码：

```
    public static double[,] Ni(double[,] iMatrix)
        {
            int i = 0;
            int row = iMatrix.GetLength(0);
            double[,] MatrixZwei = new double[row, row * 2];
            double[,] iMatrixInv = new double[row, row];
            for (i = 0; i < row; i++)
            {
                for (int j = 0; j < row; j++) { MatrixZwei[i, j] = iMatrix[i, j]; }
            }
            for (i = 0; i < row; i++)
            {
                for (int j = row; j < row * 2; j++)
                {
                    MatrixZwei[i, j] = 0;
                    if (i + row == j)
                        MatrixZwei[i, j] = 1;
                }
            }
            for (i = 0; i < row; i++)
            {
                if (MatrixZwei[i, i] != 0)
                {
                    double intTemp = MatrixZwei[i, i];
                    for (int j = 0; j < row * 2; j++){MatrixZwei[i, j] = MatrixZwei[i, j] / intTemp;}
                }
                for (int j = 0; j < row; j++)
                {
                    if (j == i)
                        continue;
                    double intTemp = MatrixZwei[j, i];
                    for (int k = 0; k < row * 2; k++){MatrixZwei[j, k] = MatrixZwei[j, k] - MatrixZwei[i, k] * intTemp;}
                }
            }
            for (i = 0; i < row; i++)
            {
                for (int j = 0; j < row; j++){iMatrixInv[i, j] = MatrixZwei[i, j + row];}
            }
            return iMatrixInv;
```

```
        }
```

12、坐标批量代入：

```
for (int p = 0; p < length; p++)
{
    大地_空间.XYZ2BLH(a, b, oldX[p, 0], oldX[p, 1], oldX[p, 2], out newX[p, 0], out newX[p, 1], out newX[p, 2]);
}
```