

卫星导航定位 算法与程序设计

王 坚

中国矿业大学

二〇一一年十月

目 录

卫星导航定位	1
目 录.....	2
第 1 章 绪论	1
1.1 GPS 算法与程序设计	1
1.2 MATLAB	1
1.3 课程目的及内容	3
第 2 章 程序设计基础.....	4
2.1 程序设计基本思想	4
2.1.1 面向过程程序设计	4
2.1.2 面向对象程序设计	5
2.1.3 面向过程设计与面向对象程序设计对比	6
2.1.4 程序开发过程	7
2.1.5 MATLAB 程序应用基础	7
2.2 矩阵、变量、运算和表达式	8
2.2.1 矩阵的输入	8
2.2.2 矩阵运算	9
2.2.3 变量与表达式	9
2.2.4 矩阵的其他简单运算:	10
2.2.5 数组及其运算	11
2.3 基本数学函数	12
2.4 图形	12
2.4.1 创建二维图形	14
2.4.2 subplot 函数	15
2.4.3 网格和曲面	16
2.5 符号运算	18
2.6 语句结构	20
2.7 MATLAB 函数定义	25
2.8 字符数组和字符串操作	28
2.8.1 MATLAB 的字符数组	28

2.8.2 字符串的比较	29
2.8.3 字符串的归类	29
2.8.4 字符串的查找与替换	30
2.8.5 字符串运算	30
2.8.6 一般通用字符串操作	30
2.9 MATLAB 工具箱简介	31
第 3 章 时空基准	32
3.1 时间基准转换	32
3.1.1 时间系统	32
3.1.2 时间转换	33
3.1.3 不同时间表示法间的转换	35
3.2 坐标转换	37
3.2.1 常用坐标系	38
3.2.2 高斯正反算	40
3.2.3 大地坐标与直角坐标转换	43
3.2.4 ECEF2ENU	45
第 4 章 标准格式数据读取	48
4.1 RINEX 格式	48
4.1.1 概述	48
4.1.2 RINEX 文件的类型及命名规则	49
4.1.3 RINEX 文件的结构	51
4.1.4 RINEX 文件实例	53
4.2 SP3 格式	55
4.2.1 概述	55
4.2.2 SP3 文件命名规则	55
4.2.3 SP3 文件格式及特点	56
4.2.4 SP3 实例	56
4.3 重要 MATLAB 函数的使用及讲解	58
4.3.1 读取 N 导航文件:	58
4.3.2 读取 O 数据文件	58
4.3.3 读取 SP3 数据文件	59
4.4 MATLAB 代码读取标准数据的程序结构	59

第 5 章 卫星坐标计算	61
5.1 开普勒定律	61
5.2 开普勒轨道参数	62
5.2.1 开普勒轨道参数	62
5.2.2 卫星运动力学方程	63
5.3 卫星轨道计算	65
5.3.1 卫星轨道方程	65
5.3.2 开普勒方程	66
5.3.3 卫星坐标计算	66
5.4 卫星坐标计算算法步骤	67
5.4.1 算法	67
5.4.2 重要 MATLAB 函数说明	69
5.4.3 重要代码及标注	71
第 6 章 伪距定位及 DOP 值计算	74
6.1 引言	74
6.2 伪距定位	74
6.3 定位精度因子 DOP 的计算	76
6.4 重要 MATLAB 函数说明	77
第 7 章 电离层延迟和对流层延迟	79
7.1 电离层延迟	79
7.1.1 电离层模型	79
7.1.2 程序用到的有关函数	82
7.1.3 程序说明	85
7.1.4 程序完整过程	86
7.2 对流层延迟	88
7.2.1 延迟模型	88
7.2.2 程序	89
第 8 章 整周模糊度解算	93
8.1 整周模糊度	93
8.2 Z 变换	96
8.3 重排列	97
8.4 搜索	99
8.5 模糊度搜索空间尺寸的确定	100
8.6 程序包使用说明	101

8.6.1 求取浮点解和固定解	101
8.6.2 LAMBDA 可视化程序	103
8.6.3 LAMBDA 程序分解	104
第 9 章 GPS 高程的多项式曲面拟合	105
9.1 拟合模型	105
9.2 X, Y 坐标重心化	105
9.3 结果评价	106
9.4 实例应用	106
9.5 程序代码及说明	108
第 10 章 GPS 通信协议——NMEA-0183	114
10.1 NMEA 通信协议	114
10.2 NMEA 通讯协议硬件接口	114
10.3 GPS 测量中 NMEA 语句解析	115
10.4 NMEA 数据处理中的注意事项	118
第 11 章 实验数据模拟	120
11.1 加载 GPS 轨道参数	120
11.2 卫星位置的计算	120
11.3 卫星天空分布图的绘制	121
11.4 卫星在天空中的运动情况	122
11.5 DOP 值的计算	122
第 12 章 TEQC 软件介绍	124
12.1 TEQC 功能简介	124
12.2 TEQC 命令介绍	124
12.2.1 RINEX 格式转换	125
12.2.2 数据编辑	126
12.2.3 质量检核	129
12.2.4 各历元伪距单点定位计算	131
第 13 章 TEQC 处理实例	132
13.1 格式转换	132
13.2 质量检查	132
13.3 视图文件查看	133
13.4 各历元伪距单点定位计算	144

第1章 绪论

1.1 GPS 算法与程序设计

全球定位系统（Global Positioning System, GPS）是20世纪70年代由美国国防部研制的新一代卫星导航定位系统，向全球范围内的用户全天候提供高精度的导航、定位和授时服务。目前，GPS已在地形测量、交通管理、导航、野外勘探等领域得到了广泛的应用。随着GPS应用领域的迅速扩展，人们急切地需要掌握GPS的相关知识。目前，国内外许多大学已为不同专业学生（电子通信、导航、大地测量等）开设了GPS课程，对GPS测量原理进行学习。

算法（Algorithm）是指解题方案的准确而完整的描述，是一系列解决问题的清晰指令，算法代表着用系统的方法描述解决问题的策略机制。也就是说，能够对一定规范的输入，在有限时间内获得所要求的输出。如果一个算法有缺陷，或不适合于某个问题，执行这个算法将不会解决这个问题。不同的算法可能用不同的时间、空间或效率来完成同样的任务。一个算法的优劣可以用空间复杂度与时间复杂度来衡量。GPS算法就是用系统的方法实现GPS原理，处理GPS原始数据，获取位置、速度等信息。

程序设计（Programming）是给出解决特定问题程序的过程，是软件构造活动中的重要组成部分。程序设计过程应当包括分析、设计、编码、测试、排错等不同阶段。程序设计往往以某种程序设计语言为工具，给出这种语言下的程序。GPS程序设计就是用程序语言实现GPS算法，是GPS原理的最终实现体。

原理——算法——程序设计是一个递进的过程，原理是将事物的抽象化，算法是对抽象的原理作了系统的描述，程序设计则是对这种描述的实现。由此可见，掌握GPS原理只是实现GPS原理的第一步，只有通过GPS算法和程序设计的深入学习才能更好地理解GPS算法，最终实现算法。

1.2 MATLAB

MATLAB是矩阵实验室（Matrix Laboratory）的简称，是美国MathWorks公司出品的商业数学软件，用于算法开发、数据可视化、数据分析以及数值计算的高级技术计算语言和交互式环境。

MATLAB将数值分析、矩阵计算、科学数据可视化以及非线性动态系统的建模和仿真等诸多强大功能集成在一个易于使用的视窗环境中，为科学研究、工程设计以及必须进行有效数值计算的众多科学领域提供了一种全面的解决方案，并在很大程度上摆脱了传统非交互式程序设计语言（如C、Fortran）的编辑模式，代表了当今国际科学计算软件的先进水平。

相较于其他的编程平台，MATLAB有着很多优势：

(1) 良好的工作平台和编程环境

MATLAB由一系列工具构成。这些工具方便用户使用MATLAB的函数和文件，其中许多工具采用的是图形用户界面。包括MATLAB桌面和命令窗口、历史命令窗口、编辑器和调试器、路径搜索和用于用户浏览帮助、工作空间、文件的浏览器。随着MATLAB的商业化以及软件本身的不断升级，MATLAB的用户界面也越来越精致，更加接近Windows的标准界面，人机交互性更强，操作更简单。而且新版本的MATLAB提供了完整的联机查询、帮助系统，极大的方便了用户的使用。简单的编程环境提供了比较完备的调试系统，程序不必经过编译就可以直接运行，而且能够及时地报告出现的错误及进行出错原因分析。

(2) 简单易用的程序语言

MATLAB是一个高级的矩阵/阵列语言，它包含控制语句、函数、数据结构、输入和输出和面向对象编程特点。用户可以在命令窗口中将输入语句与执行命令同步，也可以先编写好一个较大的复杂的应用程序（M文件）后再一起运行。新版本的MATLAB语言是基于最为流行的C++语言基础上的，因此语法特征与C++语言极为相似，而且更加简单，更加符合科技人员对数学表达式的书写格式。使之更利于非计算机专业的科技人员使用。而且这种语言可移植性好、可拓展性极强，这也是MATLAB能够深入到科学研究及工程计算各个领域的重要原因。

(3) 强大的科学计算机数据处理能力

MATLAB是一个包含大量计算算法的集合。其拥有600多个工程中要用到的数学运算函数，可以方便的实现用户所需的各种计算功能。函数中所使用的算法都是科研和工程计算中的最新研究成果，而前经过了各种优化和容错处理。在通常情况下，可以用它来代替底层编程语言，如C和C++。在计算要求相同的情况下，使用MATLAB的编程工作量会大大减少。MATLAB的这些函数集包括从最简单最基本的函数到诸如矩阵，特征向量、快速傅立叶变换的复杂函数。函数所能解决的问题其大致包括矩阵运算和线性方程组的求解、微分方程及偏微分方程的组的求解、符号运算、傅立叶变换和数据的统计分析、工程中的优化问题、稀疏矩阵运算、复数的各种运算、三角函数和其他初等数学运算、多维数组操作以及建模动态仿真等。

(4) 优秀的图形处理功能

MATLAB自产生之日起就具有方便的数据可视化功能，以将向量和矩阵用图形表现出来，并且可以对图形进行标注和打印。高层次的作图包括二维和三维的可视化、图象处理、动画和表达式作图。可用于科学计算和工程绘图。新版本的MATLAB对整个图形处理功能作了很大的改进和完善，使它不仅在一般数据

可视化软件都具有的功能（例如二维曲线和三维曲面的绘制和处理等）方面更加完善，而且对于一些其他软件所没有的功能（例如图形的光照处理、色度处理以及四维数据的表现等），MATLAB同样表现了出色的处理能力。同时对一些特殊的可视化要求，例如图形对话等，MATLAB也有相应的功能函数，保证了用户不同层次的要求。另外新版本的MATLAB还着重在图形用户界面（GUI）的制作上作了很大的改善，对这方面有特殊要求的用户也可以得到满足。

1.3 课程目的及内容

通过MATLAB学习GPS相关算法，更易理解GPS原理，而且不会因为程序语言问题对学习带来太大障碍。即利于对GPS原理的掌握，又利于MATLAB软件和语言的学习。

本书的主要内容如下：

第1章：绪论部分，介绍了GPS原理、算法和程序的关系。

第2章：GPS基本原理、系统结构和发展历史，讲解了MATLAB的基本知识和矩阵的使用。

第3章：GPS中使用的坐标系和时间系统及其相互转换。

第4章：GPS的几种标准数据文件格式。

第5章：利用广播星历文件进行卫星轨道的计算。

第6章：通过伪随机码实现伪距定位及DOP值的计算。

第7章：GPS定位中重要的误差源电离层延迟和对流层延迟原理。

第8章：模糊度的搜索固定方法。

第9章：GPS高程测量与曲面拟合。

第10章：NMEA0183的读写。

第11章：实验数据模拟展示。

第12章：TEQC软件介绍。

第13章：TEQC处理实例。

本书的目的在于为学生提供参与教学、积极思考和动手操作的工具。通过对本书的学习，更易深入地了解GPS系统的原理，掌握关键算法的实现。

第2章 程序设计基础

2.1 程序设计基本思想

程序是以某种语言为工具编制出来的动作序列，它表达了人的思想。计算机程序是用计算机程序设计语言所要求的规范书写出来的一系列动作，它表达了程序员要求计算机执行的操作。

一个程序中应包括以下两个方面的内容。

(1) 对数据的描述。在程序中要制定数据的类型和数据的组织形式，即数据结构。

(2) 对操作的描述。及操作步骤，也就是算法。

数据是操作的对象，操作的目的是对数据进行加工处理，以得到期望的结果。一个程序除了以上两个主要因素之外，还应当采用结构化程序设计方法进行程序设计，并且用某一种计算机语言表示。因此，算法、数据结构、程序设计方法和语言工具4个方面是一个程序设计人员所应具备的知识。四个方面中，算法是灵魂，数据结构是加工对象，语言是工具，编程需要采用合适的方法。

多年来，计算机程序的主要目标是力求编写出短小的代码以使运行速度更快。因为硬件成本和上机运行费很高。当计算机变得更小、更廉价、速度更快时，程序员开发程序、维护程序的费用急剧上升，而计算机硬件和运行的成本快速下降，程序设计的目标也就发生了变化。

在程序正确的前提下，可读性，易维护，可移植是程序设计首要的目标。所谓可读，就是使用良好的书写风格和易懂的语句编写程序。所谓易维护，是指业务需求发生变化时，不需要太多的开销就可以扩展和增强程序的功能。所谓可移植，是指编写的程序在各种计算机和操作系统上都能运行，并且运行结果一样。

程序设计方法正在从面向过程程序设计走向面向对象程序设计。前者考虑问题的方式是面向流程的，一个程序的设计思路就是解决一个问题的流程。而后者是面向问题中的各种独立个体的，程序的分析设计过程就是将程序分解成不同对象（不同概念体）之间的交互的过程。面向对象的设计思想在分析、解决较为复杂的、涉及较多概念的大型现实问题的时候具有更好的效果。

2.1.1 面向过程程序设计

面向对象出现以前，结构化程序设计是程序设计的主流，结构化程序设计又称为面向过程的程序设计。面向过程的程序设计方法是一种以过程为中心的编程思想。就是分析出解决问题所需要的步骤，然后用函数把这些步骤一步一步实现，

使用的时候一个一个依次调用就可以了。在面向过程程序设计中，问题被看作一系列需要完成的任务，函数（在此泛指例程、函数、过程）用于完成这些任务，解决问题的焦点集中于函数。其中函数是面向过程的，即它关注如何根据规定的条件完成指定的任务。

面向过程其实是最为实际的一种思考方式，就算是面向对象的方法也是含有面向过程的思想，可以说面向过程是一种基础的方法，它考虑的是实际的实现，一般的面向过程是从上往下步步求精，所以面向过程最重要的是模块化的思想方法，对比面向过程，面向对象的方法主要是把事物给对象化，对象包括属性与行为，当程序规模不是很大时，面向过程的方法还会体现出一种优势，因为程序的流程很清楚，按着模块与函数的方法可以很好的组织，比如拿学生早上起来的事情来说说这种面向过程吧，粗略的可以将过程拟为：

- (1) 起床
- (2) 穿衣
- (3) 洗脸刷牙
- (4) 去学校

而这4步就是一步一步的完成，它的顺序很重要，你只须一个一个的实现就行了。而如果是用面向对象的方法的话，可能就只抽象出一个学生的类，它包括这四个方法，但是具体的顺序就不能体现出来。

2.1.2 面向对象程序设计

Object Oriented，简称OO，是一种以事物为中心的编程思想。面向对象程序设计的本质是把数据和处理数据的过程当成一个整体——对象。

面向对象设计是一种把面向对象的思想应用于软件开发过程中，指导开发活动的系统方法，是建立在“对象”概念基础上的方法学。对象是由数据和容许的操作组成的封装体，与客观实体有直接对应关系，一个对象类定义了具有相似性质的一组对象。而每继承性是对具有层次关系的类的属性和操作进行共享的一种方式。所谓面向对象就是基于对象概念，以对象为中心，以类和继承为构造机制，来认识、理解、刻画客观世界和设计、构建相应的软件系统。面向对象的编程范式：

- (1) 决定你要的类；
- (2) 给每个类提供完整的一组操作；
- (3) 明确地使用继承来表现共同点。

由这个定义，我们可以看出：面向对象设计就是“根据需求决定所需的类、类的操作以及类之间关联的过程”。

面向对象的程序设计（Object-Oriented Programming，简记为OOP）的一条

基本原则是计算机程序是由单个能够起到子程序作用的单元或对象组合而成。OOP达到了软件工程的三个主要目标：重用性、灵活性和扩展性。为了实现整体运算，每个对象都能够接收信息、处理数据和向其它对象发送信息。面向对象程序设计中的概念主要包括：对象、类、数据抽象、继承、动态绑定、数据封装、多态性、消息传递。通过这些概念面向对象的思想得到了具体的体现。

2.1.3 面向过程程序设计与面向对象程序设计对比

结构化技术的特点：把现实世界描绘为数据在信息系统中的流动，在数据流动过程中数据发生转化。通过自顶向下的程序设计将复杂的程序分解为程序模块的层次图。基本特点概括为：

- (1) 自顶向下；
- (2) 逐步求精；
- (3) 模块化设计；
- (4) 结构化编码。

面向对象的特点：面向对象技术将数据模型和处理模型二者合一，将属性和方法封装在一个对象当中。将信息系统看成是一起工作来完成某项任务的相互作用的对象集合；通过定义系统中所有对象类型并显示对象之间是如何通过相互作用来完成分析任务。面向对象就是既使用对象又使用类和继承等机制，而且对象之间仅能通过传递消息实现彼此通信。

面向对象优点：

- (1) 稳定性好；
- (2) 可重用性好；
- (3) 较易开发大型软件产品；
- (4) 可维护性好。

结构化方法是将系统看成是过程的集合，过程与数据实体之间交互，过程接受输入并产生输出。面向对象方法则不再把程序看成工作在数据上的一系列过程或函数的集合，而是把程序看作是相互协作而又彼此独立的对象的集合。

(1) 从概念方面看，结构化软件是功能的集合，通过模块以及模块和模块之间的分层调用关系实现；面向对象软件是事物对象的集合，通过对象以及对象和对象之间的通讯联系实现；

(2) 从构成方面看，结构化软件是过程和数据集合，以过程为中心；面向对象软件是数据和相应操作的封装，以对象为中心；

(3) 从运行控制方面看，结构化软件采用顺序处理方式，由过程驱动控制；面向对象软件采用交互式、并行处理方式，由消息驱动控制；

(4) 从开发方面看，结构化方法的工作重点是设计；面向对象方法的工作

重点是分析；但是，在结构化方法中，分析阶段和设计阶段采用了不相吻合的表达式，需要把在分析阶段采用的具有网络特征的数据流图转换为设计阶段采用的具有分层特征的软件结构图，在面向对象方法中设计阶段的内容是分析阶段成果的细化，则不存在这一转换问题；

（5）从应用方面看，相对而言，结构化方法更加适合数据类型比较简单的数值计算和数据统计管理软件的开发；面向对象方法更加适合大型复杂的人机交互软件的开发。

2.1.4 程序开发过程

程序员首先在集成开发环境中编辑源程序，或在其他编辑器中输入源程序，然后在集成环境中启动编译程序将源程序转化成目标文件。编译之后，很有可能产生一些编译错误，于是程序员回到编辑状态重新开始编辑程序和编译的过程。同样在紧接着的连接中也会遇到连接错误，此时，又回到编辑状态修改程序。

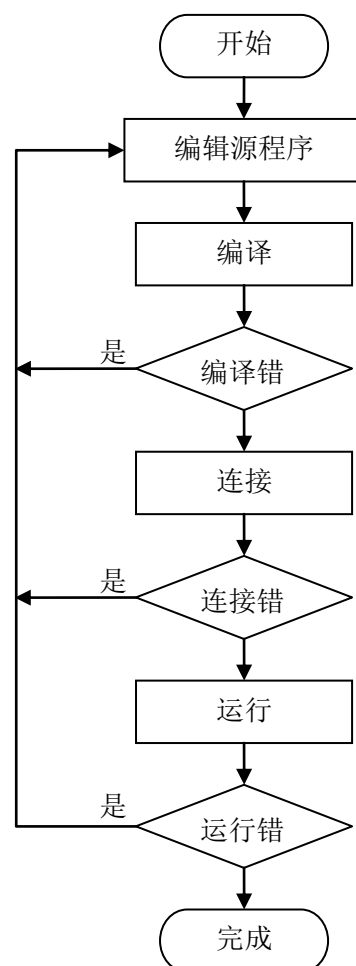


图 2-1 程序开发过程

2.1.5 MATLAB 程序应用基础

MATLAB是矩阵实验室（Matrix Laboratory）的简称，是美国MathWorks公司出品的商业数学软件，用于算法开发、数据可视化、数据分析以及数值计算的高级技术计算语言和交互式环境，主要包括MATLAB和Simulink两大部分。

MATLAB的应用范围非常广，包括数值分析、数值和符号计算、工程与科学绘图、控制系统的设计与仿真、数字图像处理技术、数字信号处理技术、通讯系统设计与仿真、财务与金融工程等。

特点：

（1）高效的数值计算及符号计算功能，能使用户从繁杂的数学运算分析中解脱出来；

（2）具有完备的图形处理功能，实现计算结果和编程的可视化；

（3）友好的用户界面及接近数学表达式的自然化语言，使初学者易于学习和掌握；

（4）功能丰富的应用工具箱，为用户提供了大量方便实用的处理工具。

MATLAB语言称为第四代编程语言，程序简洁、可读性很强而且调试十分容易。

用户如想灵活应用MATLAB去解决实际问题，充分调用MATLAB的科学技术资源，就需要编辑M文件。包含MATLAB语言代码的文件称为M文件，其扩展名为M。

MATLAB的工作模式是驱动模式，即在MATLAB窗口下当用户输入单行指令时，MATLAB立即处理这条指令，并显示结果，这就是MATLAB命令行方式。命令行操作时，MATLAB窗口只允许一次执行一行上的一个或几个语句。

将MATLAB语句构成的程序存储成以M为扩展名的文件，然后再执行该程序文件，这种工作模式称为程序文件模式。程序文件不能在指令窗口下建立，因为指令窗口只允许一次执行一行上的一个或几个语句。

2.2 矩阵、变量、运算和表达式

2.2.1 矩阵的输入

矩阵的输入一般采用直接输入的方式。

注意：

（1）行元素间用空格或逗号（，）隔开；

（2）行与行之间用分号（；）或回车；

（3）整个元素列表用[]括起；

（4）直接输入的矩阵为一全局变量，一直保存在内存中。

例：输入a=[1 2 3;4 5 6]

结果显示：

```
a=
     1     2     3
     4     5     6
```

此外，还可用语句或函数产生，用M文件或外部数据文件产生。

2.2.2 矩阵运算

运算符：+，-，*，/(右除)，\ (左除) 和^(幂)

右除： $C=A/B$ 即C满足 $CB=A$ ，当B可逆时， $A/B=AB^{-1}$

左除： $C=A\backslash B$ 即C满足 $AC=B$ ，当A可逆时， $A\backslash B=A^{-1}B$

幂： $A^n = A * \dots * A$ ； A必须是方阵。

例：矩阵的加减法：

```
a=[1:3;4:6;7:9]
b=a;   c=a+b;   c=a-b
```

注：矩阵相加减必须有相同的维数。

例：矩阵的点乘运算，^运算时矩阵必须为方阵，且只能与数字运算。

$d=a*b$ 必须符合 $m*n$ 与 $n*1$ 的结构。

$d=a.*b$ 矩阵的点乘运算

例：\ (左除)： $A\backslash B = \text{inv}(A)*B$ ，其中 $\text{inv}(A)$ 表示A逆阵。

求解 $AX=B$ ，输入：

```
A=[1 0 0; 0 4 0; 0 0 9];
```

```
B=[1 2 3; 0 1 0; 0 1 1];
```

```
X=A\B
```

/(右除): $A/B = A*\text{inv}(B)$ 。

求解 $XA=B$ ，输入：

```
X=B/A
```

2.2.3 变量与表达式

(1) MATLAB的赋值语句的两种形式：

其一为：<变量>=表达式；

其二为：表达式，将表达式的值赋于一个自动定义的变量 ans。

注：1) 如果以；结尾，则不显示计算结果，否则显示计算结果。

2) 除保留字外，变量可以用字母开头，后跟 19 个字母或数字。变量名

区分大小写，变量使用时不需要先定义，也不必定义变量的类型。

(2) 可以用who或whos来显示已定义的变量：

例如：

输入：who

输出：Your variables are:

A	B	C	a	ans
---	---	---	---	-----

输入：whos

输出	Name	Size	Bytes	Class
	A	2x3	48	double array
	B	3x3	72	double array
	C	2x3	48	double array
	a	3x2	48	double array
	ans	1x1	8	double array

Grand total is 28 elements using 224 bytes

(3) 一些常用的变量：

pi	3.14159265	// π 值
i	sqrt(-1)	//虚数单位
j	same as i	//和 i 相同
eps	floating-point relative precision, 2.2204e-016	//容量变量
realmin	smallest floating-point number, 2.2251e-308	//最小浮点数
realmax	largest floating-point number, 1.7977e+308	//最大浮点数
inf	infinity (任意一个非零数除以 0)	//正无穷大
nan	Not-a-number (0/0 或 inf-inf)	//非数

如：

r=1/0

r=inf

1/r

ans=0

2.2.4 矩阵的其他简单运算：

(1) 基本运算：

1) A'：矩阵转置；

2) inv(A)：A-1；

3) sum(A)：得到一个行向量，其元素为A的每一列的和；

4) diag(A)：得到一个列向量，其元素为A的对角元。

(2) 冒号 (:) 运算符：

1) a:b:c: 生成一个由等差数列构成的行向量X, $X(i+1)-X(i)=b$;

例: 0:pi/4:pi

ans =

0 0.7854 1.5708 2.3562 3.1416

2) 如果省略b, 则等差数列的公差为1;

例: a=0:0.05:1

x=linspace(0,1,75)

a=1:4; b=1:2:7; c=[b,a]

3) 等比数列: logspace(0,2,11) 创建起点为10, 终点为 10^2 , 11个元素, 公比为 $10^{0.2}$;

4) 矩阵的变换: rot90: 矩阵逆时针旋转 $n*90$ 度;

fliplr: 矩阵左右翻转。

flipud: 矩阵上下翻转。

5) 稀疏矩阵的存储;

sparse(A): 用于把完全矩阵压缩为稀疏矩阵。

A=[0, 1, 0, 0; 0, 3, 0, 4; 5, 0, 0, 0; 0, 0, 0, 7]

sparse(A)

ans=

(3,1) 5

(1,2) 1

(2,2) 3

(2,4) 4

(4,4) 7

sparse(i,j,u): 函数直接造成稀疏矩阵, i, j为向量分别对应行号和列号, u也为向量, 存储非元素的值。

i=[1,2,2,3,4]

j=[2,2,4,1,4]

u=[1,3,4,5,7]

A=sparse(i,j,u)

full函数把稀疏矩阵还原为完全矩阵。

2.2.5 数组及其运算

数组可以看作是行向量, 实质为阵列运算。是元素对元素的运算, 用句号(.)来区别。数组和矩阵之间的区别在于运算规则不同, 矩阵运算由线性代数规则来定义。

运算符: +, -和.*, ./, .\, .^

A.*B: A与B对应的元素相乘

A.\B: B的元素除以A的相应元素

A./B: A的元素除以B的相应元素

A.^B: A的元素为底, B的相应元素为幂的数组

例: a=[1:3;4:6;7:9]

b=a; c=a+b; c=a-b

查看下列运算的结果:

a*b a.*b a/b a./b a\b a.\b a^b (指数和底数均为矩阵, 无法求解)
a.^b a' a.'

2.3 基本数学函数

abs (绝对值或复数模)

sqrt (平方根)

real (复数的实部)

imag (复数的虚部)

conj (复数的共轭)

round (舍入为最接近的整数) //round(-0.5)=-1 round(0.4)=0

fix (向0方向舍入为整数) //fix(0.99)=0 fix(1.01)=1

floor (向负无穷大舍入为整数) //floor(-0.5)=-1 floor(0.5)=0

ceil (向正无穷大舍入为整数) //ceil(-0.5)=0 ceil(0.6)=1

sign (符号函数)

rem (x,y) (取余数函数) //得到x/y的余数, rem(11,4)=3

sin cos tan asin atan //三角函数都是面向阵列中的元素操作, 角度单位均为弧度。

atan2(y,x) //-pi <= atan2(y,x) <= pi

sinh (双曲正弦) cosh tanh

exp (以e为底的指数)

log (自然对数)

log10 (常用对数)

bessel (贝塞尔函数)

gamma (伽码函数)

2.4 图形

在分析问题的结果的时候, 图形往往是一种很有用的工具, 它可以帮助我们

直观地了解结果的某些性态。**MATLAB**提供了强大的图形功能，能够在根据向量或矩阵给定数据来生成图形。

MATLAB图形有数据可视化和图形处理两大功能，在数据的可视化部分，**MATLAB**可使用户计算所得到的数据根据不同情况转化成相应的图形。用户可以选择直角坐标、极坐标等不同的坐标系；它可以表现出平面曲线、空间曲线，绘制直方图、向量图、柱状图及空间网状面图、空间表面图等。

图形有4种：

通用图形函数、二维图形函数、三维图形函数、特殊图形函数。

表2-1 图形绘制步骤

步 骤	典型代码
1) 准备数据	<pre>x=0:0.2:12 y1=bessel(1,x); y2=bessel(2,x); y3=bessel(3,x);</pre>
2) 选择窗口，决定绘图位置	<pre>figure(1) subplot(2,2,1)</pre>
3) 调用基本绘图函数	<pre>h=plot(x,y1,x,y2,x,y3);</pre>
4) 设置绘图线条样式和标记	<pre>set(h,'LineWidth',2,{ 'LineStyle'},... {'--',':','-.'}) set(h,{ 'Color'},{ 'r';'g';'b'})</pre>
5) 设置坐标轴范围、刻度和栅格线	<pre>axis[0 12 -0.5 1]) grid on</pre>
6) 标记图形坐标轴、图形图例以及其它文字	<pre>xlable('Time') ylable('Amplitude') legend(h,'First','Second','Third') title('Bessel Functions') [y,ix]=min(y1); text(x(ix),y,'First Min\rightarrow'... 'HorizontalAlignment','right')</pre>
7) 打印图形	<pre>print -dps2</pre>

图形窗口（**Figure Window**）是所有**MATLAB**的图形输出的专用窗口。**figure**: 用来打开一个绘图窗口以供后续绘图命令输出图形。

创建图形窗口的命令：**figure**

两种格式：**figure**

figure(n)

2.4.1 创建二维图形

Plot是最简单而且使用最广泛的一个线型绘图函数，是MATLAB的内部函数。作用：可以生成线段、曲线和参数方程曲线图形。几种基本命令形式：

(1) 向量式：plot(v)

参数v可以是向量、实数阵和复数阵。v是长度为n的数值向量，生成(i,v(i))的一条折线，坐标轴的范围由MATLAB系统根据向量的长度元素的大小自动生成，当向量的元素充分多时，即可以得到一条外观光滑的曲线。

例：x=0:pi/20:2*pi;plot(x)

x=peaks;plot(x)

(2) 参数式：plot(x,y)

x,y都是长度为n的向量，它的作用是在坐标系中生成顺序连接顶点{x(i),y(i)}的折线，绘得的连线图以x为横坐标，y为纵坐标。这种调用可以用来生成参数方程的图形。

例：y=sin(x);plot(x,y)

例：x1=0:1:10*pi;

x2=0:0.01:10*pi;

y1=sin(x1).*x1;

y2=sin(x2).*x2;

plot(x1,y1)

hold on

plot(x2,y2)

(3) 矩阵式：plot(y)

y是一个(m*n)的矩阵，为矩阵的每一列画出一条线。同时以矩阵的行向量为基准对x轴进行分度和标注，标准时，采用向量1:m，这里m为矩阵的行数。

例：z=peaks; %%产生一个49*49的矩阵

plot(z)

(4) 混合式：plot(x,y)

如果X,Y均为向量，则长度必须相等，亦即参数式；如果X是向量，而Y是一个矩阵，X的长度与矩阵Y的行数或列数相等，则它的作用是将向量X与矩阵Y的每列或每行的向量相对应作折（曲）线，当Y是方阵时，则将向量X与矩阵Y的列向量相对应作图；如果X是矩阵，Y是向量，Y的长度等于X的行数或列数，则将X的每列或每行的向量与Y相对应作图。同样，当X是方阵时，则将X的各列与Y相对应作图；如果X和Y都是矩阵，且维数相同，那么按列与列的对应方式

来作图。

例： `y=1:length(peaks);plot(peaks,y)`

(5) 综合调用： `plot(x1,y1,x2,y2,...)`

矩阵中有多个矩阵调用对，其中的每一对按前四种方式之一进行调用，不同的矩阵对之间，其维数可以不同。

例： `t=0:pi/10:2*pi;`

`y1=sin(t);`

`plot(t,y1);`

可以使用`plot`同时显示由若干个x-y对表示的图形：

`y2=sin(t-0.5);`

`y3=sin(t-0.25);`

`plot(t,y1,t,y2,t,y3);`

从图上我们可以看出，为了体现不同x-y对表示的图形，MATLAB自动地使用了不同的颜色；此外我们还可以使用如下格式来设置图形的性态。

2.4.2 subplot 函数

在一个已存在图形的绘图窗口中增加一个图形（在这之前讲的都是覆盖方式）`hold on`，加到图形上，`hold off` 则取代当前窗口。

允许在同一绘图窗口中显示或在同一纸上打印多个图形。

`subplot(m,n,p)`：将一个绘图窗口分割成`m*n`个子窗口，每一小块有自己的坐标轴，`p`为活动窗口，并且在`p`个窗口绘制当前图形，如果坐标系已存在，则`subplot(p)`设置当前坐标系。

例：

`t=0:pi/10:2*pi;`

`[x,y,z]=cylinder(4*cos(t));`

%% （生成一个单位圆柱体，采用[0, 1]之间的等距节点，每一个节点的旋转半径是 $4\cos(t)$ ，相当于 $4\cos(t)$ 母线 `CYLINDER(R,N)`, `CYLINDER(R)`（此时，`n=20`））

`subplot(2,2,1); mesh(x) %% (x=1:m, y=1:n, z=x, c=x)`

`subplot(2,2,2); mesh(y);`

`subplot(2,2,3); mesh(z);`

`subplot(2,2,4); mesh(x,y,z)`

`subplot(111)`是个特殊的情况，与 `subplot(1,1,1)`不同，它使下一条绘图指令在窗口中执行 `elf` 和 `reset` 指令（即删除当前图形的所有子对象），然后在默认位置创建一个坐标系。

绘制复数图形: $\text{plot}(z) \Leftrightarrow \text{plot}(\text{real}(z), \text{imag}(z))$

例:

```
t=0:pi/10:2*pi;
plot(exp(i*t),'-o')
```

2.4.3 网格和曲面

MATLAB具有强大的三维图形处理功能, 包括三维数据显示、空间、曲面、分块、填充以及曲面光顺着色、视点变换、旋转、隐藏等功能和操作。

(1) $\text{plot3}(x,y,z,\text{Linespec})$

plot3 是一个可用来画单变量的三维函数, 函数格式除了包括第三维的信息之外, 其它与二维函数 plot 相同。

例: 绘制一个三维螺旋线。

```
t=0:pi/50:8*pi;
figure(1)
plot3(sin(t),cos(t),t)
grid off,axis square
title('三维螺旋线')
```

例: 增加维数的 plot3 命令可以使多个二维图形沿一个轴排列起来, 而不是直接将二维图形叠加到另一个的上面。下例就是二维图形在三维空间的排列。

```
x=linspace(0,3*pi);
z1=sin(x);
z2=sin(2*x);
z3=sin(3*x);
y1=zeros(size(x));
y3=zeros(size(x));
y2=y3/2;
plot3(x,y1,z1,x,y2,z2,x,y3,z3);
grid,xlabel('x-axis'),ylabel('y-axis'),zlabel('z-axis')
title('sin(x),sin(2x),sin(3x)')
```

(2) 平面网格点的生成

在数学上, 函数 $z=f(x,y)$ 的图形是三维空间的曲面, 在MATLAB中, 总是假设函数 $z=f(x,y)$ 是定义在一个矩形的区域 $D=[x_0,x_m]*[y_0,y_n]$ 上的。为了绘制在区域 D 上的三维曲面, MATLAB的方法是首先将 $[x_0,x_m]$ 在 x 方向上分成 m 份, 将 $[y_0,y_n]$ 在 y 方向分成 n 份, 由各分划点分别作平行于坐标轴的直线, 将区域 D 分成 $m*n$ 个小矩阵块, 计算出在网格点的函数值。对于每个小矩形, 在空间中决定出

4个顶点($x_i, y_i, f(x_i, y_i)$), 连接4个顶点得到一个空间中的四边形片。所有这些四边形片一起构成函数 $z=f(x, y)$ 定义在区域D上的空间网格曲面。

为方便起见, MATLAB中用meshgrid函数产生x,y轴向的网格数据, meshgrid函数的一般格式为:

$[X, Y] = \text{meshgrid}(x, y)$

其中, x, y 分别指向x轴向和y轴向的数据点, 当 x 为n维向量, y 为m维向量时, X, Y 均为 $m \times n$ 矩阵, $X(i, j)$ 和 $Y(i, j)$ 共同指向平面上的一点, 当 x, y 轴取同一向量时, 可简写成 $[X, Y] = \text{meshgrid}(x)$

例: 数学函数 $z = \sin(\sqrt{x^2 + y^2}) / \sqrt{x^2 + y^2}$, 定义在区域 $[-8, 8] \times [-8, 8]$ 上。在生成网格点后, 计算网格点上的函数值。

```
x=-8:0.5:8;
y=x;
[X,Y]=meshgrid(x,y);
R=sqrt(X.^2+Y.^2)+eps;
Z=sin(R)./R
```

(3) 网格曲面 (mesh 函数)

mesh 函数绘制三维空间上的网格曲面, 如上例最后加上 mesh(Z), 生成一曲面。

mesh 函数的其它格式为:

mesh(X,Y,Z): X, Y, Z 为同维数的矩阵。

mesh(X,Y,Z,C): C 称为颜色矩阵。网格曲面的顶点对应于空间的顶点($X(i, j), Y(i, j), Z(i, j)$), 而网格曲面的网格线的颜色由 C 值根据当前的色谱来着色。这种调用形式还可以用来生成参数曲面片。

mesh(x,y,Z,C): 其中, x 和 y 是向量, Z 和 C 是同维数的矩阵, 且向量 x 的长度等于矩阵 Z 的列数, 而向量 y 的长度等于矩阵 Z 的行数。在这种情况下, 网格曲面的曲格顶点是($x(j), y(i), Z(i, j)$), 网格线的长度由矩阵 C 决定。

mesh(Z,C): Z 和 C 都是 $m \times n$ 矩阵, 该形式与 mesh(x,y,Z,C)等价。

与 mesh 相关的另外两个函数是 meshc 和 meshz, 它们的调用形式与 mesh 相同。meshc 函数可同时绘制出轮廓图 (即等高线图)。meshz 的作用除了生成与 mesh 相同的网和曲面之外, 还在曲面下面加上一个长方体的台柱, 使图形更加美观。

例: 曲面图

```
[x,y]=meshgrid(-3:.125:3);
z=peaks(x,y);c=ones(size(z));
```

```

figure(1)
mesh(x,y,z,c),grid on
title('多峰函数的网格曲线')
例: [x,y]=meshgrid(-8:0.5:8);
r=sqrt(x.^2 + y.^2) + eps; /*加 eps 是为了防止出现 0/0 的情形*/
z=sin(r)./r;
mesh(x,y,z);
例: 曲面与等高线
x=-8:0.5:8;
y=x;
[X,Y]=meshgrid(x,y);
R=sqrt(X.^2+Y.^2)+eps;
Z=sin(R)./R
subplot(2,2,1);meshc(Z)
subplot(2,2,2);meshz(Z)

```

2.5 符号运算

(1) 关系运算:

MATLAB有六种关系运算符: < <= > >= == ~= 结果1表示true, 0表示false。其中< <= > >=只用于操作数的实部比较, == ~=用于比较实部和虚部。

1) 当两个变量是标量时, a和b的关系成立, 结果为1, 否则结果为0。

2) 当比较a和b是两个维数相同的数组时, 按相同位置比较, 结果是一个维数和a相同的数组, 其元素由1和0组成。

3) 当比较的一个是数组a, 一个是标量b时, 则把标量b和数组a的每一个元素按标量关系逐个比较, 结果是一个维数和a相同的数组, 其元素由1和0组成。

```

例: a=rand(5);b=a
c=a==b;
c=a>b
c=a>1;c=a>0;c=a>0.5
x=(1:10);t=x>5

```

(2) 逻辑运算: | (或), & (与), ~ (非) 和逻辑函数or、and、not、xor及any、all、find

MATLAB提供三个逻辑操作符&、|、~, 同时又存在三个相应的M文件: all、

or、not，这二组的作用是相同的，只是使用格式稍有差异。xor是第四个逻辑运算函数，完成异或操作。

在逻辑操作中，所有输入元素的非零值都当作1处理。逻辑运算的优先级最低。

例：x=[25 -5; 0 0.001]

~x

ans=

0 0

1 0

y=[1 0;1 0]

z1=x&y; z2=and(x,y); z3=xor(x,y);

z1=

1 0

0 0

z2=

1 0

0 0

z3=

0 1

1 1

MATLAB还提供了许多测试用的逻辑函数，巧妙地使用这些函数，可得到意想不到的结果。

any：若作用于一向量x，则当x中至少有一个非零元素时返回1，否则返回0，若x是矩阵，则对每一列执行any，返回一个元素为0或1的向量。主要作用测试矩阵中是否有非零元素。

如：a=rand(15)

any(a)=(1 1 1 1 1)

all：测试向量，若向量的所有元素非零，返回 1，否则返回 0，若对矩阵，则对每一列执行 all 操作。主要作用测定矩阵中是否有零元素。

all(a)

例：(ex24.m)

x=linspace(0,5,100) //产生 0~5 之间均匀分布的 100 个数据

y=cos(x)

z=(y>0).*y //将 cos 函数的负数置为 0


```

z=z+0.3*(y<0)           //将 cos 函数的正数值增加 30%
z=(x<=4).*z             //将超过 4 时的 z 值置为 0。
plot(x,z)

```

find: 找出矩阵中非零元素及其下标。

```

例: a=zeros(5,20);
a(3,7)=0.5;a(4,15)=-0.4;
[i,j,v]=find(a)
i=
     3
     4
j=
     7
    15
v=
    0.5000
   -0.4000

```

运算符的优先级:

在一个表达式内可能有几个运算符。**MATLAB**为确定运算的次序设定了运算符的优先级。在同一优先级时,程序先左后右执行,在优先级不同时,先高级后低级执行。

最高	()
↓	~ (取反)
	. ^ + -
	. * ./ .\ */ \+ -
	: < <= > >= ++ ~=-
最低	&

2.6 语句结构

(1) if, else, elseif, end语句

if语句的格式:

if <逻辑表达式>

语句集

elseif <逻辑表达式>

语句集

else

语句集

end

仅由if和end组成的语句，可根据逻辑表达式的值选择是否执行。

例：if rem(a,2)=0

disp('a is even')

b=a/2

end

例：if x

disp('矩阵x全为0')

else

disp('矩阵x不全为0')

end

例：if and(a==1,b>5)

.....

end

if是MATLAB中最常用的条件执行语句，它与end语句一起构成各种格式，

(2) for-end循环

for 循环，指定次数的重复循环执行语句。

格式：for <循环参数> = <初态>:<步长>:<终态>

循 环 体

end

当步长=1 时，步长可以省略。

例：n=4;

for i=1:n

for j=1:n

if i==j

a(i,j)=2;

elseif min([i,j])==1

a(i,j)=1;

else

a(i,j)=0;

end

end

```
end
disp('The matrix A is');disp(a)
```

执行时：

The matrix A is

```
2  1  1  1
1  2  0  0
1  0  2  0
1  0  0  2
```

例：(test1.m)

```
for i=1:m
    for j=1:n
        a(i,j)=1/(i+j-1);
    end
end
```

例：for i=9.8:3:-9

i

end

执行时：i=9.800 i=6.800 3.800 i=0.800 i=-2.200 i=-5.200 i=-8.200

例：s='abcdefghijk'

for i=s

i

end

%i 分别等于 s 中和每一个字符。

例：c={'aaa','bbb','ccc'}

for i=c

i

end

i='aaa' i='bbb' i='ccc'

例：可利用数组（阵列）任意指定循环变量的值。(ex25.m)

```
varx=[7 3 10 5];
vary=zeros(size(v));
k=0;
for x=varx
    k=k+1
```

```

        vary(k)=x.^2;
    end
    disp([v,w])

```

例：（ex31.m）求 100-200 之间的素数。

```

for m=101:2:200
    k=fix(sqrt(m));
    for i=2:k+1
        if rem(m,i)==0
            break;
        end
    end
    if i>=k+1
        disp(m)
    end
end

```

（3）while-end循环

不定次数重复的循环执行语句。

格式：while <逻辑表达式>
 循环体

```

    end

```

例：a=150;
 while a>0.1
 a=a/2;
 end

（4）break语句：

一个跳出循环的命令,导致最内层的 while,for,if 语句终止。

例：（test3.m）

```

a = 0; fa = -inf;
b = 3; fb = inf;
while b-a > eps*b
    x = (a+b)/2;
    fx = x^3-2*x-5;
    if fx == 0
        break
    end
end

```

```

elseif sign(fx) == sign(fa)
    a = x; fa = fx;
else
    b = x; fb = fx;
end
end
x

```

(5) return语句

return 语句终止当前的命令序列，把控制返回到调用函数或键盘。

(6) 输入/输出语句

1) 获得用户输入 **input**: 显示一段信息，并读取用户输入。

语法: `x=input(提示信息,"输入类型")`，缺省为double型。

例: `a=input('Please input a member')`

`a=input('Please input a char','s')`

2) 暂停执行，等待用户按键 **pause**: 程序暂停执行。

语法: `pause(n)`

`pause`

其中: `n`为暂停的秒数。

3) 建立完整的图形界面: 菜单输入函数 **menu**。

例: (ex27.m)

```

s=menu('color selection','red','green','blue','yellow','black')
switch(s)
case 1
    scolor='red';
case 2
    scolor='green';
case 3
    scolor='blue';
case 4
    scolor='yellow';
case 5
    scolor='blue';
otherwise
    disp('error');

```

```
end  
scolor
```

运行结果：



4) 输出函数 print, printopt

2.7 MATLAB 函数定义

(1) 自定义函数

用M函数自定义函数，函数M文件必须以函数的名称来作为文件名，M文件的格式如下：

<因变量>=<函数名>(<自变量>)

自变量和因变量都可以是矩阵或几个矩阵。

M文件的第一行包括function，该文件就是函数文件，函数和命令文件的区别是：命令文件的变量用完后保存在内存，而函数文件文件内定义的量仅在函数文件内部起作用。函数文件有多个输出变量时，用[]括起，有多个输出变量时，用（）括起。

例：(quroot.m)

```
function x=quroot(a)  
if abs(a(1)) > eps  
    d=a(2)^2-4*a(1)*a(3);  
    if d >= 0  
        e=sqrt(d);  
        f=2*a(1);  
        x(1)=(-a(2)+e)/f;
```

```

        x(2)=(-a(2)-e)/f;
    end
elseif abs(a(2)) > eps
    x=-a(3)/a(2);
else
    error('coefficient error!')
end

```

函数调用为:

```

a=[1 2 1]
x=quroot(a)
x=
    -1    -1

```

```

/*子函数调用*/
function [mean, stdev] = stat(x)
n=length(x);
mean = avg(x,n);
stdev = sqrt(sum((x-avg(x,n)).^2)/n);

function mean = avg(x,n)
mean = sum(x)/n;

```

在函数文件中可以包含多个函数，其中第一个函数称为主函数，其函数名与文件名相同，它可以由其它文件引用，其它函数称为子函数，它只能由函数中的主函数和其它子函数引用。

函数中有两个永久变量 `nargin`, `nargout` 表示引用函数时给出的输入变量数和输出变量数。

```

function c=testrg1(a,b)
if(nargin==1)
    c=a.^2;
else (nargin==2)
    c=a*b;
end

```

输入一个变量时，计算这一变量的平方数

输入二个变量时，计算这两个变量的积。

函数的递归调用，在调用一个函数的过程中又出现直接或间接地调用该函数本身。

例：求 $n!$ (ex33.m)

```
function y=ex33(n) %% 求 n!  
if n<0  
    error('n is smaller than 0')  
    return  
end  
if n==0|n==1  
    y=1  
else  
    y=n*ex33(n-1);  
end
```

(2) 全局变量

局部变量和全局变量：

函数工作空间中，变量有三类：

- 1) 由调用函数传递输入和输出数据的变量；
- 2) 在函数内部临时产生的变量（局部变量）；
- 3) 由调用函数空间，基本工作空间或其它函数工作空间提供的全局变量。

global： 让不同的函数访问同一个变量

只要在函数中：**global 变量名;%% 该变量为全局变量**

```
function h = falling(t)  
global GRAVITY  
h = 1/2*GRAVITY*t.^2;
```

例：(ex28.m) 对于函数 $z=\alpha(x-1)^2+\beta(y+1)^2$ 。编写相应的函数文件，其中 α 和 β 采用全局变量进行参数传递。

```
global alpha beta  
x=[0:0.02:2];  
y=[-2:0.02:0];  
[X,Y]=meshgrid(x,y);  
subplot(2,2,1);alpha=1;beta=1;  
z=fun1(X,Y);mesh(z);  
subplot(2,2,2);alpha=2;beta=1;  
z=fun1(X,Y);mesh(z)
```



```

subplot(2,2,3);alpha=1;beta=2;
z=fun1(X,Y);mesh(z)
subplot(2,2,4);alpha=0.8;beta=0.5;
z=fun1(X,Y);mesh(z)
function z=fun1(x,y)
global alpha beta
z=alpha*(X-1).^2+beta*(Y+1).^2

```

(3) 向量化

MATLAB的实际使用中，有些循环可直接转换成向量操作，这样可大大提高程序执行速度，这种技术称为循环的向量化，在编写程序时，应尽量避免采用循环，将它转变为向量进行处理。以便最大限度地提高用MATLAB编写的程序的效率。

如何将循环变量变换成向量操作是一个比较复杂的问题。简单问题很容易转换为向量操作，但复杂的问题不易直观转换，视实际问题而定。

例：比较如下两个算法：

(t1.m)

```

tstart=clock;
x=0.01;
for k=1:1000
    y(k) = log10(x);
    x = x + 0.01;
end
tend=clock;
interval = tend - tstart

```

(t2.m)

```

tstart = clock;
x=0.01:0.01:10;
y=log10(x);
tend = clock;
interval = tend - tstart

```

t2.m的执行速度远远低于t1.m的执行速度。

2.8 字符数组和字符串操作

2.8.1 MATLAB 的字符数组

在MATLAB中，字符串都被视作字符数组。

字符串的建立：`s='字符串'`，即用`'`将输入的字符串括起来。注意不是`"`，这点与其它高级语言不同。而要建立一个字符串矩阵，则可以这样建立：

```
sa=['string1','string2'.....]
```

与数组不同，字符串矩阵的每一行字符元素的个数可以不同，但是每一行的所有字符串中的字符的总个数必须相同，如果不满足这个条件，即使每行中字符串的个数相同，也会出错。事实上，MATLAB将一个行内的所有字符串都合并起来，构成一个字符串，单个字符串之间不加空格，这正是每行中输入的字符串的个数可以不同的原因。

`char()`函数自动将最短的字符串添加空格，使各行长度相等。

```
例: a=['aaa' 'bbb';'cc' 'dd']
```

系统报错，必须为

```
a=['aaa' 'bbb';'cc' 'dddd']
```

利用这个特点，可以用`[]`将任意字符串连接起来。

```
例: sa=['aa' 'bb' 'cc';'dd' 'ee' 'f' 'k']
```

```
sa=
```

```
    aabbcc
```

```
    ddeefk
```

```
例: [sa(1,:) sa(2,:)]
```

```
sa=
```

```
    aabbccddeefk
```

注：字符串的每个字符（包括空格）都是矩阵的一个元素。

处理字符矩阵时当作数据矩阵来处理

2.8.2 字符串的比较

函数`strcmp(str1,str2)`：比较两个字符串是否相同，若相等则返回1值，若不相等则返回0值。

函数`strncmp(str1,str2,n)`：比较两个字符串的前n个字符是否相同，若相等则返回1值，若不相等则返回0值。

2.8.3 字符串的归类

函数`isletter(S)`：判断串中元素是否为字母。

函数`isspace(S)`：判断串中元素是否为空格。

这二个函数的返回结果为与字符串长度相等的矢量，1表示结果为真，0表示结果为假。

2.8.4 字符串的查找与替换

函数findstr('str1','str2')：在字符串str1中查找子串str2，返回str2在str1中的起始位置。

函数strrep('str1','str2','str3')：字符串替换

2.8.5 字符串运算

字符串及字符串矩阵可以进行加、减、乘、除四则运算和其它数字运算（利用ASCII码）。

例：'e'+'f'=203

2.8.6 一般通用字符串操作

通用字符串操作包括字符串与ASCII间的转换、字符与数据间的相互转换，字符串大小写间的转换、字符串中空格或删除等。

表2-2 通用字符串操作

函数名	函数功能
string(A)	将一个整数数组转换成字符串矩阵
abs(S)	将字符串转换成ASCII码
double(S)	将字符串转换成相应的ASCII码（双精度）
isstr(S)	确认是否为字符串
deblank(C)	删除字符串结尾处的空格
disp(n)	输入n个空格
str2mat(s1,s2,s3,...)	将字符串变为字符串矩阵
upper(S)	将字符串进行大写转换
lower(S)	将字符串进行小写转换
a=eval('')	将字符串作为命令执行
strmatch('substr',S)	匹配字符串操作
strtok('string',d)	得到指定的子串
int2str(A)	将整数转换为字符串
num2str(A)	将浮点数转换为字符串
str2num(S)	将字符串转换为浮点数
dec2hex(A)	把十进制整数转换为十六进制字符串
hex2dec(S)	把十六进制字符串转换为十进制整数
hex2num(S)	把十六进制字符串转换为浮点数
dec2bin(A)	把十进制数转换为二进制字符串

例：打印出ASCII码从32-126之间所有可显示的字符。

```
char(32:126)
```

```
char(48:126)
```

2.9 MATLAB 工具箱简介

MATLAB包括拥有数百个内部函数的**主包**和三十几种**工具包**。工具包又可以分为**功能性工具包**和**学科工具包**。功能工具包用来扩充MATLAB的符号计算，可视化建模仿真，文字处理及时实时控制等功能。学科工具包是专业性比较强的工具包，控制包，信号处理工具包，通信工具包等都属于此类。

开放性使MATLAB广受用户欢迎。除内部函数外，所有MATLAB主包文件和各种工具包都是可读可修改的文件，用户通过对源程序的修改或加入自己编写程序构造新的专用工具包。

MATLAB Main Toolbox——MATLAB主工具箱

Control System Toolbox——控制系统工具箱

Communication Toolbox——通讯工具箱

Financial Toolbox——财政金融工具箱

System Identification Toolbox——系统辨识工具箱

Fuzzy Logic Toolbox——模糊逻辑工具箱

Higher-Order Spectral Analysis Toolbox——高阶谱分析工具箱

Image Processing Toolbox——图象处理工具箱

等

第3章 时空基准

3.1 时间基准转换

3.1.1 时间系统

时间：物质存在和运动的基本形式之一。判别事件发生的先后顺序和运动的快慢程度。

时间间隔：是指客观物质运动两个不同状态之间所经历过的时间历程。

时刻：是指客观物质在某种运动状态的瞬间与时间坐标轴的原点之间的时间间隔。

目前人类认识时间的水平为 $10^{-24} \sim 10^{18}$ 秒。

时间是物质存在和运动的客观形式，建立时间单位必须以物质的运动为依据。选取的物质运动形式不同，就会有不同的时间系统。这种运动必须满足下列要求：

- (1) 必须是连续的周期性运动。
- (2) 运动的周期必须有足够的稳定性。
- (3) 这种周期运动必须可复现，即可观测的。

目前有几种时间系统在使用，它们基于不同的周期性运动来作为时间基准。为了保证时间具有一定的精确性，要求这种周期性运动必须是均匀及连续的。基于不同的周期性运动，时间系统主要有以下几种：

(1) 世界时时间系统

世界时时间系统以地球自转这种周期性运动作为时间基准。地球自转运动与人类生活关系及其密切，且在一定范围内非常稳定，所以很自然的把地球自转作为时间基准。但是，由于观察地球自转运动的空间参考点不同，世界时时间系统又分为恒星时和平太阳时。

1) 恒星时

由春分点（地球公转轨道面对赤道面的升交点）作为观察地球自转的空间参考点，由它的周日视运动所确定的时间系统称为恒星时。恒星时在数值上等于春分点相对于本地子午圈的时角。同一时间不同子午圈上的恒星时不同，所以恒星时具有地方性，有时也称地方恒星时。春分点是观测不到的，所以只能通过观测恒星来间接的推算春分点的位置。

2) 平太阳时

地球相对于太阳自转一周的时间称为太阳日。不过地球围绕太阳公转的轨道

面为椭圆，使得太阳日不是很均匀，这样得到的时间系统计时就不会很准确。所以，将平太阳的周日视运动（即真太阳视运动的全年平均值）作为时间基准，地球相对于平太阳自转一周的时间叫做平太阳日。在格林尼治的平太阳时称为世界时（Universal Time），我们日常生活中采用的计时单位即平太阳时。

根据近代天文观测发现地球自转并非均匀不变，地球自转的不均匀性包括长期减慢，各种周期性起伏以及不规则变化，部分原因在于潮汐变形和其他质量转移导致的极惯性矩的变化，部分原因是地球自转轴本身的摆动，因此也导致了恒星时与世界时不再是均匀的时间计量系统。

（2）原子时

物质内部原子跃迁时，辐射和吸收的电磁波频率具有很高的稳定性，所以原子时便成为最理想的时间系统。定义为：位于海平面上的铯原子基态的两个超精细能级，在零磁场中辐射振荡9192631770周所持续的时间，为1原子时秒。原子时不是天文意义上的时间系统，而是用物理方法建立的时间系统，它是通过原子钟来守时和授时。

（3）协调世界时

自原子时诞生后，世界时作为时间计量基准的作用不断减弱。但世界时仍旧被用于地球的空间姿态描述，用于研究地球自转的变化规律。由于地球自转的速度不断减慢，世界时与原子时之间的差距不断增大，原子时将昼夜变化不同步，而使得人们日常生活的起居受到影响。例如：若采用原子时作为日常时间系统，太阳东升西落的时间会不断变化，太阳升起的时间会不断延后。为此，引入协调世界时的概念，记为UTC，取原子时的秒长作为时间单位以度量时间间隔，但通过调整时刻，使得UTC及UT的时刻差不要太大。即通过控制两者之差，使得两者时刻差的绝对值不超过0.9秒，若超过则实行跳秒。跳秒时间安排在每年民用时间1月1日0时或7月1日0时，人为的在00:00:00之前加上23:59:60（称为正跳秒）；否则在23:59:58之后紧接着是00:00:00（负跳秒）。若发生正跳秒，则UTC时间将调慢1秒，否则调快1秒。至今发生的跳秒均为正跳秒。由此可见，UTC已不再是一种天文的时间系统，而是民用的时间系统，其时间调整方案不是因为天文概念上的要求，而是民用习惯上的要求。

3.1.2 时间转换

时间标示法指的是表示时间的方法。有别于定义时间尺度的时间系统，是建立在时间系统之上的时间表示方法。在GPS应用和数据处理中，会涉及到以下几种时间标示法：

- 日常生活中普遍采用的历法标示法；
- 科学领域中普遍采用的儒略时法；

- GPS 系统内部所采用的 GPS 时标示法；
- GPS 测量中采用的年积日标示法。

(1) 历法

历法是日常生活中最常用的时间标示法。在历法中，规定了“年”的起始时间、长度和分划，制定了“日”以及更长的时间分划单位（如“星期”、“月”和年）的编排规则，用年、月、日来标示时间。

从古至今，世界上出现了很多历法：

- 1) 罗马历 (Roman Calendar)
- 2) 儒略历 (Julian Calendar)
- 3) 格里高利历 (Gregorian Calendar)
- 4) 我国的农历 (Chinese Traditional Calendar)

目前，世界上广泛采用的历法是格里高利历。

表 3-1 格里高利历中每月的天数

月份	一	二	三	四	五	六	七	八	九	十	十一	十二
天数	31	28/29	31	30	31	30	31	31	30	31	30	31

上表中，闰年的二月为29天，否则为28天。闰年的规定：在年号前能被4整除的年份中，除了那些能被100整除但不能被400整除的年份外，其余的均为闰年。

历法时间标示法符合人们的日常生活习惯，能够容易地反映出季节、节假日等日常生产、生活所需的重要信息。但是，由于这种方法不是采用连续的数值来标示时间，因而不适合用于科学计算。

(2) 儒略时

儒略时是一种采用连续数值标示时间的方法，于1583年在法国出生的Joseph Justus Scaliger 提出。根据定义，儒略日是指从-4712年1月1日12时（即公元4713年1月1日12时）开始计算的天数。

优点：

- 采用连续的数值来标示时间，适合科学计算。
- 可以很方便地将采用不同方法所标示的时间联系起来。

缺点：

无法直接反映季节等信息，故日常生活中不常用。

(3) 约化儒略时

由于儒略时的计时起点距今较为久远，若将现今时间用儒略时表示，数值非常大。1973年国际天文学联合会（IAU-International Astronomical Union）提出了约化儒略日（MJD-Modified Julian Day）的时间标示法。

转换公式：

$$MJD=JD-2400000.5$$

这样，用约化儒略日所标示的时间就小得多。

约化儒略日实际上给出的是从1858年11月17日子夜（儒略日为2400000.5）开始计算的天数。另外，约化儒略日与儒略日在时间参考上也有所不同，约化儒略日以子夜为参考，而儒略日则以正午为参考。

（4）GPS时

GPS系统内部所采用的时间系统是GPS时，其时间零点定义为1980年1月5日夜与1980年1月6日晨之间的子夜。GPS时系统在标示时间时所采用的最大时间单位为周（week，604800秒），其标示时间的方法是从1980年1月6日0时开始起算的周数（WN-Week Number）加上被称为周内时间（TOW-Time of Week）的从每周周六/周日子夜开始起算的秒数。

例如：“1980年1月6日0时0分0秒”用GPS时标示法则为“第0周0秒”；而“2004年5月1日10时5分15秒”用GPS时标示法则为“第1268周第554715秒”。在GPS卫星所发送的导航电文中，时间信息的标示就是采用这种形式。

（5）年积日

年积日（DOY-Day of Year）指的是从每年的1月1日起开始累计的天数，计数从1开始（即每年1月1日的年积日为1），如2004年5月1日的年积日为122。

在GPS中的用途：年积日通常用来区分观测时段，常用于GPS观测文件的命名。例如，在RINEX格式中就规定：在数据文件的8字符主文件名中，第5-7个字符为观测起始时刻的年积日。

当然，要想确定实际日期，只有年积日是不够的，还必须给出年份信息。

3.1.3 不同时间表示法间的转换

（1）格里高利历与儒略日间的相互转换

1) 格里高利历至儒略日的转换

采用下面的算法，可将用格里高利历所标示的时间转换为儒略日：

$$JD=floor(365.25*(y+4716))+floor(30.6001*(m+1))+D+h/24-1537.5$$

且：

若 $m \leq 2$, 则 $y=Y-1$, $m=M+12$;

若 $m > 2$, 则 $y=Y$, $m=M$ 。

式中：JD为儒略日；Y为年；M为月；D为日； $h=H+Min/60+s/3600$, H, Min和S分别为时，分和秒；floor()为取整函数，有 $floor(a) \leq a$ ，例如 $floor(-2.3)=-3$ 。

2) 儒略日至格里高利历的转换

转换公式如下：

$$a=floor(julday+0.5);$$


```

b=a+1537;
c = floor((b-122.1)/365.25);
d = floor(365.25*c);
e = floor((b-d)/30.6001);
D = floor(b-d- floor(30.6001*e)+ rem(julday+.5,1)); %天
H=(rem(julday+.5,1))*24; %时
M=e-1-12*floor(e/14); %月
Y=c-4715-floor((7+M)/10); %年

```

Rem 函数是取余函数 $\text{rem}(x,y)=x-y*\text{fix}(x/y)$, 其中 $\text{fix}()$ 是向0取整, 例如 $\text{fix}(-1.5)=-1$, $\text{fix}(1.5)=1$ 。

上面的转换算法仅在1900年3月1日至2100年2月28日期间有效。

(2) GPS时间与儒略日间的相互转换

1) 儒略日至GPS时的转换

GPS时的起点是1980年1月6日0时（儒略日为2444244.5），用相对于该起点的周数（WN）和在一周内的秒数（TOW）表示，因而，可以采用下面的算法将儒略日转换为GPS时：

```

day_of_week = rem(floor(julday+.5),7)
WN = floor((JD-2444244.5)/7)
TOW= (rem(D,1)+day_of_week+1)*86400

```

2) GPS时至儒略日的转换

转换公式为： $JD=WN*7+TOW/86400+2444244.5$

(3) 年积日与格里高利历间的相互转换

1) 格里高利历至年积日的转换

采用下面算法，可由格里高利历所标示的日期（Y年M月D日）计算出年积日（DOY）：

- ①计算出Y年1月1日的儒略日 JD_0 ；
- ②计算出Y年M月D日的儒略日JD；
- ③儒略日JD对应的年积日DOY可采用下式： $DOY=JD-JD_0+1$ 。

2) 年积日至格里高利历间的转换

采用以下步骤，可由Y年的年积日（DOY）计算出用格里高利历所标示的日期（Y年M月D日）：

- ①计算出Y年1月1日的儒略日 JD_0 ；
- ②儒略日可采用下式计算： $JD=JD_0+DOY-1$
- ③根据JD计算出所对应的格里高利历的日期。

注：MATLAB的rem()和mod()函数

rem(x,y): 求整除x/y的余数

mod(x,y): 求模

rem(x,y)=x-y.*fix(x./y); (fix()向靠近0方向取整)

mod(x,y)=x-y.*floor(x./y); (floor()向左取整)

如果x和y的符号相同（同为‘+’，同为‘-’），那么rem(x,y)=mod(x,y)（正数与正数，负数与负数，取整结果两个函数效果一样）

如果x和y的符号相反，那么mod(x,y)=rem(x,y)+y(正数与负数的取整，看你希望得到什么结果就选择用什么函数)

当正数与负数取余时，当得到的余数结果的符号希望跟除数(x)一样，用rem()函数；当得到的余数结果的符号希望跟被除数(y)一样，用mod()函数

例如：

rem(3,2)=1;mod(3,2)=1;

rem(-3,-2)=-1;mod(-3,-2)=-1;

rem(3,-2)=1;mod(3,-2)=-1;

rem(-3,2)=-1;mod(-3,2)=1;

mod(3,2)=rem(3,2)=1;

mod(3,-2)=rem(3,-2)-2=1-2=-1;

mod(-3,2)=rem(-3,2)+2=-1+2=1;

floor(x): 得到不大于x的整数值;

round(x): 得到最接近x 的整数值;

ceil(x): 得到不小于x的整数值;

fix(x): 得到绝对值不大于x绝对值的整数值。

用原数减去整数部分不就是小数部分了吗

format long %设置为小数点后9位

format short %设置为小数点后4位

vpa(pi,200) %设置pi为小数点后200位

3.2 坐标转换

全球定位系统的最基本任务是确定用户在空间的位置。而所谓用户的位置，实际上是指该用户在特定坐标系的位置坐标，位置是相对于参考坐标系而言的，为此，首先要设立适当的坐标系。坐标系统是由原点位置、3个坐标轴的指向和尺度所定义，根据坐标轴指向的不同，可划分为两大类坐标系：天球坐标系和地球坐标系。

由于坐标系相对于时间的依赖性，每一类坐标系又可划分为若干种不同定义的坐标系。不管采用什么形式，坐标系之间通过坐标平移、旋转和尺度转换，可以将一个坐标系变换到另一个坐标系去。

3.2.1 常用坐标系

(1) WGS-84

WGS-84的定义：WGS-84是修正NSWC9Z-2参考系的原点和尺度变化，并旋转其参考子午面与BIH定义的零度子午面一致而得到的一个新参考系，WGS-84坐标系的原点在地球质心，Z轴指向BIH1984.0定义的协定地球极（CTP）方向，X轴指向BIH1984.0的零度子午面和CTP赤道的交点，Y轴和Z、X轴构成右手坐标系。它是一个地固坐标系。

WGS-84椭球及其有关常数：WGS-84采用的椭球是国际大地测量与地球物理联合会第17届大会大地测量常数推荐值，其四个基本参数

长半径： $a=6378137\pm 2$ （m）；

地球引力常数： $GM=3986005\times 10^8\text{m}^3\text{s}^{-2}\pm 0.6\times 10^8\text{m}^3\text{s}^{-2}$ ；

正常化二阶带谐系数： $C_{20}=-484.16685\times 10^{-6}\pm 1.3\times 10^{-9}$ ；

地球自转角速度： $\omega=7292115\times 10^{-11}\text{rads}^{-1}\pm 0.150\times 10^{-11}\text{rads}^{-1}$ 。

(2) BJ54

1) 1954年北京坐标系（BJ54旧）

①坐标原点：前苏联的普尔科沃。

②参考椭球：克拉索夫斯基椭球。

③平差方法：分区分期局部平差。

2) 存在的问题：

①椭球参数有较大误差。

②参考椭球面与我国大地水准面存在着自西向东明显的系统性倾斜。

③几何大地测量和物理大地测量应用的参考面不统一。

④定向不明确。

(3) GDZ80

1) 1980年国家大地坐标系（GDZ80）

①坐标原点：陕西省泾阳县永乐镇。

②参考椭球：1975年国际椭球。

③平差方法：天文大地网整体平差。

2) 特点：

①采用1975年国际椭球。

②参心大地坐标系是在1954年北京坐标系基础上建立起来的。

③椭球面同似大地水准面在我国境内最为密合，是多点定位。

④定向明确。

⑤大地原点地处我国中部。

⑥大地高程基准采用1956年黄海高程。

(3) 高斯坐标系

高斯投影又称横轴椭圆柱等角投影，属于正形投影。设想有一椭圆柱横套在地球椭球体外面，并与地球椭球的某一条子午线相切，椭圆柱中心轴通过椭球体中心并与椭球的长轴重合，然后用一定的投影方法将中央子午线两侧各一定经差范围内的地区投影到椭圆圆柱面上。如图2-1所示。

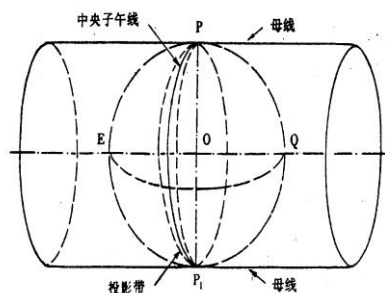


图 3-1 横切椭圆柱投影

高斯投影的性质：1) 投影后角度不变；

2) 长度比与点位有关，与方向无关；

3) 离中央子午线越远变形越大。

为了控制投影后的长度变形，采用分带投影的方法。常用3度带或6度带分带，城市或工程控制网坐标采用任意带分带。

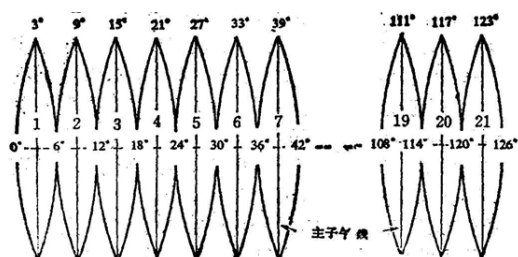


图 3-2 高斯分带 (6°)

下式为6°带和3°带中央子午线与带号关系：

$$L_0 = 6n - 3 \quad \text{或为} \quad n = \frac{1}{6}(L_0 + 3)$$

$$L_0' = 3n' \quad \text{或为} \quad n' = \frac{L_0'}{3}$$

高斯投影坐标正算：由大地坐标系 (B, L) 推求高斯平面坐标系 (x, y) ；

高斯投影坐标反算：由高斯平面坐标系 (x, y) 推求大地坐标系 (B, L) ；

3.2.2 高斯正反算

(1) 高斯正算

设椭球面上一点投影到平面后为 d 点，大地坐标为 (B, L) ，投影后中央子午线经度为 L_0 。

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

其中：

X 为赤道至纬度 B 的子午线长度；

$$l = \frac{L - L_0}{\rho}, t = \tan B;$$

$$N \text{ 卯酉圈曲率半径, } N = \frac{a}{\sqrt{1 - e^2 \sin^2 B}};$$

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

a 为椭球长半轴， b 为椭球短半轴， $e = \frac{\sqrt{a^2 - b^2}}{a}$ 为地球第一偏心率，

$e' = \frac{\sqrt{a^2 - b^2}}{b}$ 地球第二偏心率。

(2) 高斯反算

设 N, t, η, a, e, e' 同上

$$\begin{cases} a_1 = N \cos B \\ a_2 = \frac{1}{2} N \sin B \cos B \\ a_3 = \frac{1}{6} N \cos^3 B (1 - t^2 + \eta^2) \\ a_4 = \frac{1}{24} N \sin B \cos^3 B (5 - t^2 + 9\eta^2 + 4\eta^4) \\ a_5 = \frac{1}{120} N \cos^5 B (5 - 18t^2 + t^4 + 14\eta^2 - 58\eta^2 t^2) \\ a_6 = \frac{1}{720} N \sin B \cos^5 B (61 - 58t^2 + \eta^4) \end{cases}$$

$$\begin{cases} \beta_0 = 1 - \frac{3}{4}e'^2 + \frac{45}{64}e'^4 - \frac{175}{256}e'^6 + \frac{11025}{16384}e'^8 \\ \beta_2 = \beta_0 - 1 \\ \beta_4 = \frac{15}{32}e'^4 - \frac{175}{384}e'^6 + \frac{3675}{8192}e'^8 \\ \beta_6 = -\frac{35}{96}e'^6 + \frac{735}{2048}e'^8 \\ \beta_8 = \frac{315}{1024}e'^8 \end{cases}$$

Step1:

$$\begin{cases} B^0 = \frac{x}{c\beta_0} \\ a_1^0 = \frac{a \cos B^0}{\sqrt{1 - e^2 \sin^2 B^0}} \\ l^0 = \frac{y}{a_1^0} \end{cases}$$

Step2:

$$\begin{cases} F_x(B) = \left\{ c\beta_2 + \left[c\beta_4 + (c\beta_6 + c\beta_8 \cos^2 B) \cos^2 B \right] \cos^2 B \right\} \sin B \cos B \\ F_x(B, l) = a_2 l^2 + a_4 l^4 + a_6 l^6 \\ F_y(B, l) = a_3 l^3 + a_3 l^3 \end{cases}$$

Step3:

$$\begin{cases} B^i = \frac{x - F_x(B^{i-1}) - F_x(B^{i-1}, l^{i-1})}{c\beta_0} \\ a_1^i = \frac{a \cos B^i}{\sqrt{1 - e^2 \sin^2 B^i}} \\ l^i = \frac{y - F_y(B^i, l^{i-1})}{a_1^i} \end{cases}$$

Step4:

判断 $\begin{cases} B^i - B^{i-1} \leq \varepsilon \\ l^i - l^{i-1} \leq \varepsilon \end{cases}$, 若否回到 step2, 若是输出 B^i, l^i , $B = B^i$, $L = l^i + L_0$ 。

(3) 转换实例

1) 正算

代码:

```

function [x,y]=gauss_zhengsuan(B,L,L0)
l=L-L0;
l=deg2rad(l); %角度转换为弧度
B=deg2rad(B);

a=6378137; %椭球参数
f=1/298.257223563;
b=a-a*f;
c=a^2/b;
e=sqrt(a^2-b^2)/a;
e1=sqrt(a^2-b^2)/b;
Beta0=1-(3/4)*e1^2+(45/64)*e1^4-(175/256)*e1^6+(11025/16384)*e1^8;
Beta2=Beta0-1;
Beta4=(15/32)*e1^4-(175/384)*e1^6+(3675/8192)*e1^8;
Beta6=-(35/96)*e1^6+(735/2048)*e1^8;
Beta8=(315/1024)*e1^8;
Xb0=c*(Beta0*(B)+sin(B)*(Beta2*cos(B)+Beta4*cos(B)^3+Beta6*cos(B)^5+Beta8*cos(B)^7));% 计算赤道至纬度 B 的子午线长度

Np=a/sqrt(1-(e*sin(B))^2); % 卯酉圈曲率半径
m0=l*cos(B);
t=tan(B);
in2=(e1)^2*cos(B)^2;

x=Xb0+(1/2)*Np*t*m0^2+(1/24)*(5-t^2+9*in2+4*in2^2)*Np*t*m0^4+(1/720)*(61-58*t^2+t^4+270*in2-330*in2*t^2)*Np*t*m0^6;
y=Np*m0+(1/6)*(1-t^2+in2)*Np*m0^3+(1/120)*(5-18*t^2+t^4+14*in2-58*t^2*in2)*Np*m0^5;

```

数据： B = 32.04934783333333, L = 118.90422794444444, L0=117

调用： [x,y]=gauss_zhengsuan(B,L,L0)

结果： x = 3548910.811290287, y = 179854.6172135982

说明： WGS84 椭球内， Y 没有加 500KM。

2) 反算

代码：

```

function [B,L]=gauss_fansuan(x,y,L0)
%x,y 高斯坐标， L0 中央子午线经度
a=6378137; %椭球参数
f=1/298.257223563;
b=a-a*f;
c=a^2/b;
e=sqrt(a^2-b^2)/a;
e1=sqrt(a^2-b^2)/b;

Beta0=1-(3/4)*e1^2+(45/64)*e1^4-(175/256)*e1^6+(11025/16384)*e1^8;
Beta2=Beta0-1;
Beta4=(15/32)*e1^4-(175/384)*e1^6+(3675/8192)*e1^8;
Beta6=-(35/96)*e1^6+(735/2048)*e1^8;

```

```

Beta8=(315/1024)*e1^8;
B0=x/(c*Beta0); % 初值
aa0=(a*cos(B0))/sqrt(1-e^2*sin(B0)^2);
% 初值
l0=y/aa0; % 初值

N=a*sqrt(1-e^2*sin(B0)^2);
t=tan(B0);
in=e1*cos(B0);
a1=N*cos(B0);
a2=(1/2)*N*sin(B0)*cos(B0);
a3=(1/6)*N*cos(B0)^3*(1-t^2+in^2);
a4=(1/24)*N*sin(B0)*cos(B0)^3*(5-t^2
+9*in^2+4*in^4);
a5=(1/120)*N*cos(B0)^5*(5-18*t^2+t^
4+14*in^2-58*in^2*t^2);
a6=(1/720)*N*sin(B0)*cos(B0)^5*(61-
58*t^2+t^4);

F_xB=(c*Beta2+(c*Beta4+(c*Beta6+c*
Beta8*cos(B0)^2)*cos(B0)^2)*cos(B0)^
2)*sin(B0)*cos(B0);
F_xBl=a2*10^2+a4*10^4+a6*10^6;
F_yBl=a3*10^3+a5*10^5;
B1=(x-F_xB-F_xBl)/(c*Beta0);
aa1=(a*cos(B1))/sqrt(1-e^2*(sin(B1))^2
);
l1=(y-F_yBl)/aa1;
while abs(B1-B0)>=0.0001 &&
abs(l1-l0)>=0.0001
    B0=B1;
    aa0=aa1;
    l0=l1;
end
L=rad2deg(l1)+L0;
B=rad2deg(B1);

```

数据: $x = 3548910.811290287$, $y = 179854.6172135982$, $L_0 = 117$

调用: `[B,L]=gauss_fansuan(x,y,L0)`

结果: $B = 32.04918010686117$, $L = 118.9042238375928$

说明: WGS84 椭球内, 输入 BL 单位为度。

3.2.3 大地坐标与直角坐标转换

(1) 转换公式

对同一空间点, 直角坐标系与大地坐标系参数间有如下转换关系:

$$X = (N + H) \cos B \cos L$$

$$Y = (N + H) \cos B \sin L$$

$$Z = (N(1 - e^2) + H) \sin B$$

$$L = \arctan(Y / X)$$

$$B = \arctan\{Z(N + H) / [\sqrt{X^2 + Y^2} N(1 - e^2) + H]\}$$

$$H = Z / \sin B - N(1 - e^2)$$

式中： $N = a / \sqrt{1 - e^2 \sin^2 B}$ 为卯酉圈半径； $a, b, e = \sqrt{(a^2 - b^2) / a^2}$ 分布为椭球的长半轴、短半轴和第一偏心率。

(2) 转换实例

1) 直角坐标系转大地坐标系

程序(xyz2llh):

function llh = xyz2llh(xyz)

```

    x = xyz(1);
    y = xyz(2);
    z = xyz(3);
    x2 = x^2;
    y2 = y^2;
    z2 = z^2;

    a = 6378137.0000; % 椭球长半轴
    b = 6356752.3142; % 椭球短半轴
    e = sqrt(1 - (b/a)^2);
    b2 = b*b;
    e2 = e^2;
    ep = e*(a/b);
    r = sqrt(x2+y2);
    r2 = r*r;
    E2 = a^2 - b^2;
    F = 54*b2*z2;
    G = r2 + (1-e2)*z2 - e2*E2;
    c = (e2*e2*F*r2)/(G*G*G);
    s = (1 + c + sqrt(c*c +
2*c))^(1/3);
    P = F / (3 * (s+1/s+1)^2 * G*G);

    Q = sqrt(1+2*e2*e2*P);
    ro = -(P*e2*r)/(1+Q) +
sqrt((a*a/2)*(1+1/Q) ...
- (P*(1-e2)*z2)/(Q*(1+Q)) -
P*r2/2);
    tmp = (r - e2*ro)^2;
    U = sqrt( tmp + z2 );
    V = sqrt( tmp + (1-e2)*z2 );
    zo = (b2*z)/(a*V);
    height = U*( 1 - b2/(a*V) );
    lat = atan( (z + ep*ep*zo)/r );

    temp = atan(y/x);
    if x >= 0
        long = temp;
    elseif (x < 0) && (y >= 0)
        long = pi + temp;
    else
        long = temp - pi;
    end

```

```

llh(1) = lat;
llh(2) = long;
llh(3) = height;

```

数据: xyz = [-2144846.77399203, 4397711.29219396, 4078090.74292948]

调用: llh = xyz2llh(xyz)

结果: llh = [0.698127383628, 2.024570848501, 196.349307953178]

说明: WGS84 椭球内, 大地坐标经纬度为弧度。

2) 大地坐标系转直角坐标系

程序(llh2xyz):

```

function xyz = llh2xyz(llh)
    phi = llh(1);
    lambda = llh(2);
    h = llh(3);

    a = 6378137.0000; % 椭球长半轴
    b = 6356752.3142; % 椭球短半轴
    e = sqrt(1 - (b/a).^2);

    x = (a*coslam)/tmpden + h*coslam*cosphi;
    y = (a*sinlam)/tmpden + h*sinlam*cosphi;
    tmp2 = sqrt(1 - e*e*sinphi*sinphi);
    z = (a*tmp*sinphi)/tmp2 + h*sinphi;
    coslam = cos(lambda);
    sinlam = sin(lambda);
    tan2phi = (tan(phi))^2;
    tmp = 1 - e*e;
    tmpden = sqrt(1 + tmp*tan2phi);

    xyz(1) = x;
    xyz(2) = y;
    xyz(3) = z;

```

数据: llh = [0.698127383628, 2.024570848501, 196.349307953178]

调用: xyz = llh2xyz(llh)

结果: xyz = [-2.144846.77399408, 4.397711.29219308, 4.078090.74292935]

说明: WGS84 椭球内, 大地坐标经纬度为弧度。

3.2.4 ECEF2ENU

(1) 转换模型

WGS84坐标系下空间直角坐标转换到站心空间直角坐标相当于空间中两个空间直角坐标系下的变换。主要包括坐标原点的平移和三个坐标轴的旋转。

如图3-3所， L 和 B 为 P 点的经纬度。 $XYZ-O$ 为WGS84下空间直角坐标系 P 和 Q 坐标分别为 $X_Q Y_Q Z_Q$ 和 $X_P Y_P Z_P$ 。 $x^* y^* z^*-P$ 为站心直角坐标系， P 为原点， $x^* y^* z^*$ 为 Q 点的坐标。

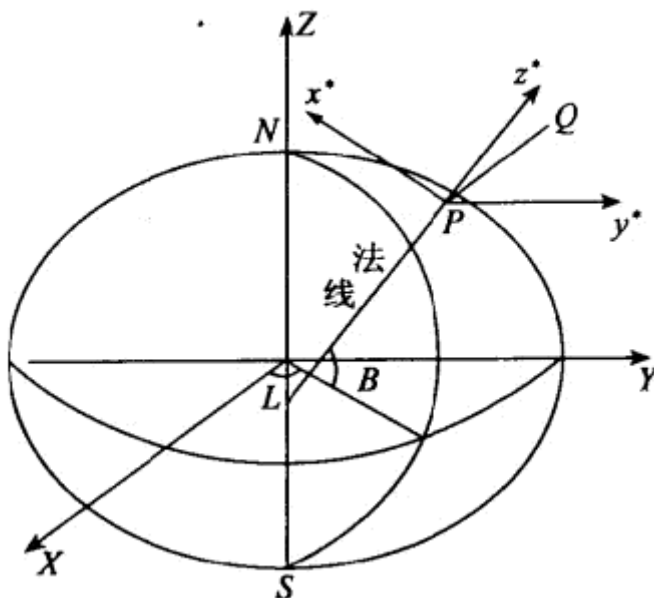


图 3-3 空间直角坐标系与站心直角坐标系的几何关系

转换关系为:

$$\begin{bmatrix} X_Q \\ Y_Q \\ Z_Q \end{bmatrix} = \begin{bmatrix} X_P \\ Y_P \\ Z_P \end{bmatrix} + \begin{bmatrix} -\sin B \cos L & -\sin L & \cos B \cos L \\ -\sin B \sin L & \cos L & \cos B \sin L \\ \cos B & 0 & \sin B \end{bmatrix} \begin{bmatrix} x^* \\ y^* \\ z^* \end{bmatrix}$$

$$\begin{bmatrix} x^* \\ y^* \\ z^* \end{bmatrix} = \begin{bmatrix} -\sin B \cos L & -\sin L & \cos B \cos L \\ -\sin B \sin L & \cos L & \cos B \sin L \\ \cos B & 0 & \sin B \end{bmatrix} \begin{bmatrix} X_Q - X_P \\ Y_Q - Y_P \\ Z_Q - Z_P \end{bmatrix}$$

(2) 转换实例

1) XYZ转NEU

程序

```
function enu = xyz2enu(xyz,orgxyz)
```

```
    difxyz = xyz - orgxyz;
```

```
    orgllh = xyz2llh(orgxyz); %将站心原点空间直角坐标坐标转换大地坐标
```

```
    phi = orgllh(1); %纬度
```

```
    lam = orgllh(2); %经度
```

```
    sinphi = sin(phi);
```

```
    cosphi = cos(phi);
```

```

sinlam = sin(lam);
coslam = cos(lam);
R = [ -sinlam      coslam      0      ; ... % 旋转矩阵
      -sinphi*coslam -sinphi*sinlam cosphi; ...
      cosphi*coslam  cosphi*sinlam sinphi];
enu = R*difxyz;
数据: xyz=[-2408881.4329;4698174.3713;3566517.8462]
orgxyz=[-2408000;4698000;3566000]
enu = xyz2enu(xyz,orgxyz)
调用: enu = xyz2enu(xyz,orgxyz)
结果: enu=[704.8615;114.8683;751.9771]
说明: WGS84 椭球,调用程序 xyz2llh。

```

2) NEU转XYZ

程序

```

function xyz = enu2xyz(enu,orgxyz)
    orgllh = xyz2llh(orgxyz);      %将站心原点空间直角坐标转换大地坐标
    phi = orgllh(1);              %纬度
    lam = orgllh(2);              %经度
    sinphi = sin(phi);
    cosphi = cos(phi);
    sinlam = sin(lam);
    coslam = cos(lam);
    R = [ -sinlam      coslam      0      ; ...% 旋转矩阵
          -sinphi*coslam -sinphi*sinlam cosphi; ...
          cosphi*coslam  cosphi*sinlam sinphi];
    difxyz = inv(R)*enu;
    xyz = orgxyz + difxyz;
数据: enu=[704.8615;114.8683;751.9771]
      orgxyz=[-2408000;4698000;3566000]
      xyz = enu2xyz(enu,orgxyz)
调用: xyz = enu2xyz(enu,orgxyz)
结果: xyz=[-2408881.4329;4698174.3713;3566517.8462]
说明: WGS84 椭球,调用程序 xyz2llh。

```

第 4 章 标准格式数据读取

4.1 RINEX 格式

4.1.1 概述

GPS 数据处理时所采用的观测数据来自野外观测的 GPS 接收机。接收机在野外进行观测时,通常将采集的数据记录在接收机内部存储器或可移动的储存介质中。在完成观测后,需要将数据传输到计算机中,以便进行分析。这一过程通常是利用 GPS 接收机厂商所提供的数据传输软件来进行。传输到计算机中的数据一般采用 GPS 接收机厂商所定义的专有格式以二进制文件的形式储存。一般说来,不同 GPS 接收机厂商所定义的专有格式各不相同,有时甚至同一厂商不同型号仪器的专有格式也不相同。

国际上为了能统一使用来自不同接收机的数据,设计了一种与接收机无关的 RINEX (The Receiver Independent Exchange Format) 格式,该格式采用文本文件形式储存数据,目前已使用 2.11 以后的版本。为了将不同型号接收机 (OEM 板) 采集的数据进行统一处理,必须将原始数据向目前 GPS 通用的 RINEX 数据交换格式进行转换。

(1) 本机格式 (Native Format)

定义: 接收机内部存储数据的格式

存储方式: 二进制

内容: 观测值、广播星历、电离层信息、气象元素等

特点:

- 1) 不同厂家接收机的本机格式各不相同;
- 2) 与接收机配套的数据处理软件一般可以直接读取本机格式的数据,但是一般不能读取其他厂家本机格式的数据;
- 3) 数据存储紧凑,含有一些专有信息;
- 4) 不利于多种型号接收机联合作业。

(2) RINEX 格式

定义: 与接收机无关的数据交换格式/Receiver Independent Exchange

历史: RINEX 格式由瑞士尼泊尔尼大学天文学院 (Astronomical Institute, University of Berne) 的 Werner Gurtner 于 1989 年提出。现在 RINEX 格式已经成为 GPS 测量应用等的标准数据格式,几乎所有测量型 GPS 接收机厂商都提供将其专有格式文件转换为 RINEX 格式文件的工具,经过多年不断修订完善,

目前最为普遍的是 RINEX 格式的第 2 版, 该版本能够用于静态和动态 GPS 测量在内的不同观测模式数据。

存储方式: ASCII/文本文件

内容: 观测值、星历(导航信息)、钟数据、气象数据等

特点:

- 1) 通用性强, 已成为领域内的标准;
- 2) 大多数软件均可处理;
- 3) 利于多种型号接收机联合作业。

4.1.2 RINEX 文件的类型及命名规则

(1) RINEX 文件的类型

在 RINEX 格式的第 2 版中定义了 6 种不同类型的数据文件, 分别用于存放不同类型的数据, 它们分别是:

观测值文件——用于存放 GPS 观测值;

导航电文文件——用于存放 GPS 卫星导航电文;

气象数据文件——用于存放在测站处测定的气象数据;

GLONASS 导航电文文件——用于存放 GLONASS 卫星导航电文;

GEO 导航电文文件及卫星——用于存放在增强系统中搭载有类 GPS 信号发生器的地球同步卫星(GEO)的导航电文;

卫星和接收机钟文件——用于存放卫星和接收机时钟信息。

对于大多数 GPS 测量应用的用户来说, RINEX 格式的观测数据、导航电文和气象数据文件最为常见。前两类数据在进行数据处理分析时通常是必需的, 而其他类型的数据则是可选的, 特别是 GLONASS 导航电文文件和 GEO 导航电文文件平时并不多见。

(2) RINEX 文件命名规则

RINEX 格式对数据文件的命名有特殊规定, 以使用户能够仅通过文件名就能区分数据文件的归属、类型和所记录的数据时间。完整的文件名由用于表示文件归属的 8 字符长度主文件名和用于表示文件类型的 3 位字符长度扩展名两部分组成, 其具体形式如下:

ssssdddf.yyt

其中,

ssss: 4 字符长度的测站号。

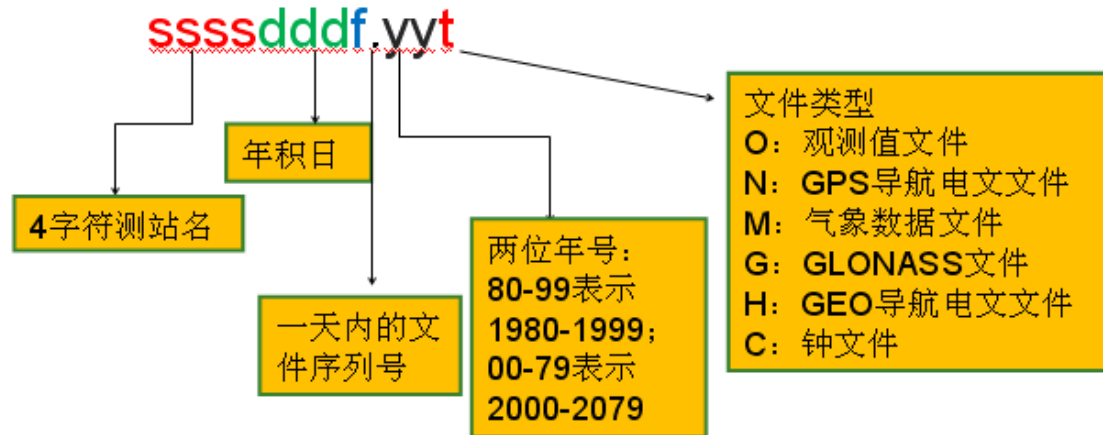
ddd: 文件中第一个记录所对应的年积日。

f: 一天内的文件序号, 有时也称为时段号, 取值为 0~9, A~Z; 当为 0 时表示文件包含了当天所有的数据。

yy: 年份。例如, 80-99:1980-1999; 00-79:2000-2079。

t: 文件类型, 为下列字母中的一个: O—观测值文件; N—GPS导航电文文件; M—气象数据文件; G—GLONASS导航电文文件; H—GEO导航电文文件; C—钟文件。

命名规则: 8+3文件名:



例如: 文件名为WHN11410.04O的RINEX格式数据文件, 为点WHN1在2004年5月20日(年积日为141)整天的观测数据文件; 而文件名为WHN11410.04N的RINEX格式数据文件, 则相应为在改点上进行观测的接收机所记录的导航电文文件。

(3) RINEX 文件特点

1) RINEX 格式的数据文件采用文本形式进行存储, 可以使用任何标准文本编辑器进行查阅编辑。RINEX 格式文件的结构是以节、记录、字段和列为单位逐级组织的。所有类型的 RINEX 格式文件, 都由文件头和数据记录两节所组成。每一节中含有若干记录, 每一记录通常为一行, 由若干字段组成, 每行最大字符数为 80。当一个记录的内容超过 80 个字符时, 可以续行, 字段在行中所处位置及宽度有严格规定, 不能错位。

2) RINEX 格式文件的文件头用于存放与整个文件有关的全局性信息, 位于每个文件的最前部, 其最后一个记录为 “END OF HEADER”。在文件头中, 每一记录的第 61-80 列为该行记录的标签, 用于说明相应行上第 1-60 列中所表示的内容。观测值文件的文件头存放有文件的创建日期、单位名、测站名、天线信息、测站近似坐标、观测值数量及类型、观测历元间隔等信息。

3) 导航电文的文件头存放有创建日期、单位名及其它一些相关信息, 另外还有可能会存放包含电离层模型参数以及说明 GPS 时与 UTC 间关系的参数和条秒等。

4) 气象数据文件的文件头则存放有文件创建日期、观测值类型、传感器信息和气象传感器近似位置及其它一些相关信息。

5) RINEX 格式文件的记录数据紧跟在文件头的后面, 随文件类型的不同, 所存放数据的内容和具体格式也不相同。

6) 观测值文件中存放的是观测过程中每一观测历元所观测到的卫星及载波相位、伪距和多普勒等类型的观测值数据等, 所包含的实际观测值类型与接收机所记录类型及格式转换时的参数设置有关。

7) 导航电文文件中存放的是所观测卫星的钟差改正模型及卫星轨道数据等。

8) 气象数据文件中存放的是观测过程中每隔一段时间在测站天线附近测定的干温、相对湿度和气压等数据。

9) 每一个观测值文件或气象数据文件, 通常包含一个测站在一个观测时段中所获得的数据, 不过在快速静态或动态测量应用中, 流动接收机通过一次设站所采集的多个测站的数据可以包含在一个数据文件中。

10) 在观测值文件中, 所记录载波相位数据的单位为周, 伪距数据的单位为 m。观测值所对应的时标 (即观测时刻) 是依据接收机钟的读数所生成的, 而不是标准的 GPS 时, 因而在该时标中含有接收机的钟差。

4.1.3 RINEX 文件的结构

现采用表格的形式详细介绍 RINEX2.11 的内容, 表格分三部分组成:

文件头标签: 这一栏将直接给出出现在文件头中某行上“标签部分” (第 61-80 列) 的内容, 在 RINEX 文件中, 它们通常用简明的英文全称或缩略语表示。若在这一行中存放有多种内容, 则在标签中用“/”分隔。

说明: 在这一栏中, 将对与前面文件头标签同处一行的第 1-60 列存放数据的内容进行说明, 若在某一行上存放有多个内容, 则将在多个列表项目中进行说明。

格式: 在这一栏中, 将对与前面文件头标签同处一行上的第 1-60 列所存放数据的格式进行说明。由于在 RINEX 格式中, 对文件格式的定义非常严格, 数据必须根据定义存放在相应的列上, 不允许有任何的错位, 因而在使用时必须特别注意。

在这里, 格式说明采用 Fortran 程序设计语言中的格式说明方式, 一个格式说明项通常具有如下形式:

[r]fw.[m]

其中:

r: 重复因子, 表示后面的内容将重复的次数, 给部分是可选的;

f: 数据类型符, 在 RINEX 格式说明中, 用到了如下数据类型: X—空格; A—字符型; I—整型; F—单精度浮点型; D—双精度浮点型; w—字段宽度;

m: 在字段中最少的数字或字符数, 当数据类型为单精度浮点型或双精度浮

点型时，表示小数位数，该部分是可选的。

例如：格式说明符“F9.2, 11X, A1,19X”表示这一行内容从第 1 列开始一次是宽度为 9 位、小数点后有 2 位的单精度浮点数，11 个空格、宽度为 1 的字符串和 19 个空格；

格式说明符“3F14.4”表示这一行内容从第 1 列宽度为 14、小数点后有 4 位的单精度浮点数；

格式说明符“7 (3X, A1,I12)”则表示这一行内容从第 1 列开始，将 3 个空格、宽度为 1 的字符串和宽度为 2 的整型这些内容重复 7 次。

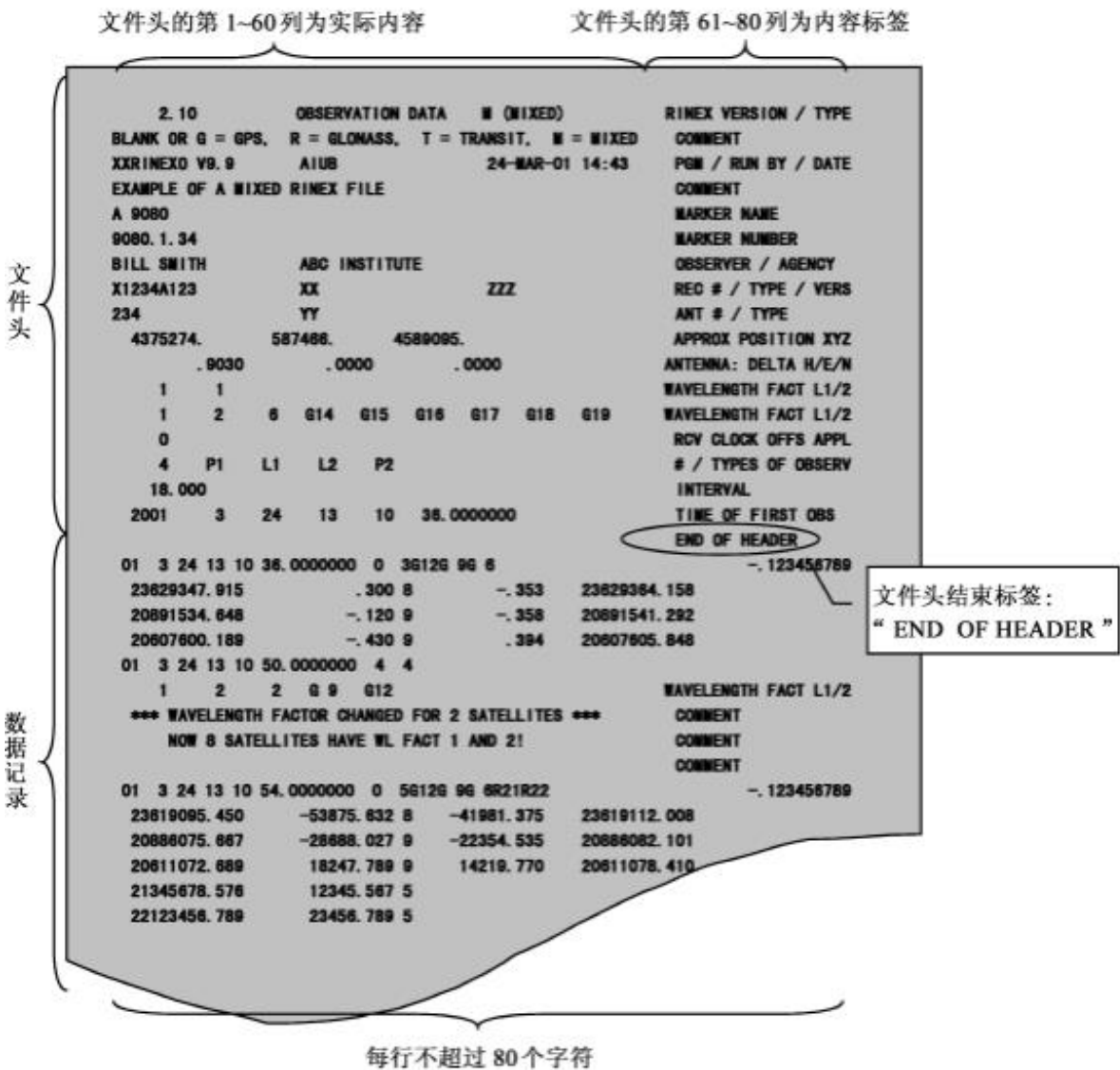


图 4-1 RINEX 格式观测数据文件的结构

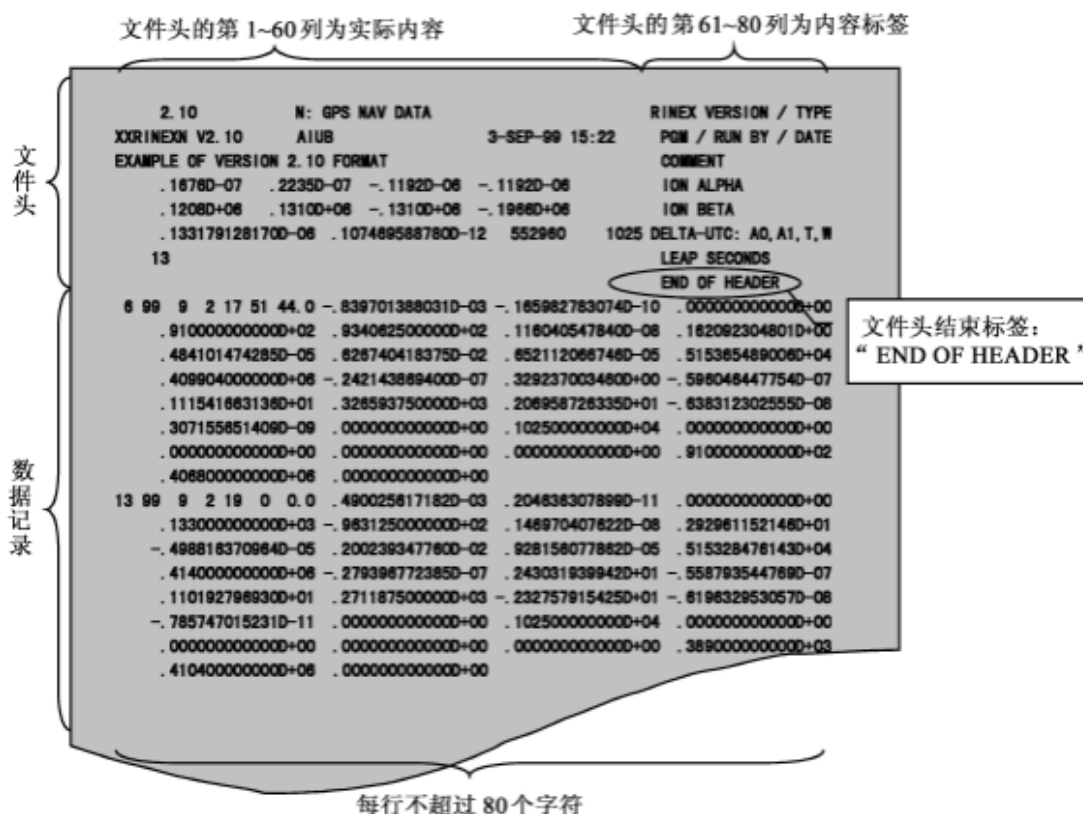


图 4-2 RINEX 格式导航电文文件的结构

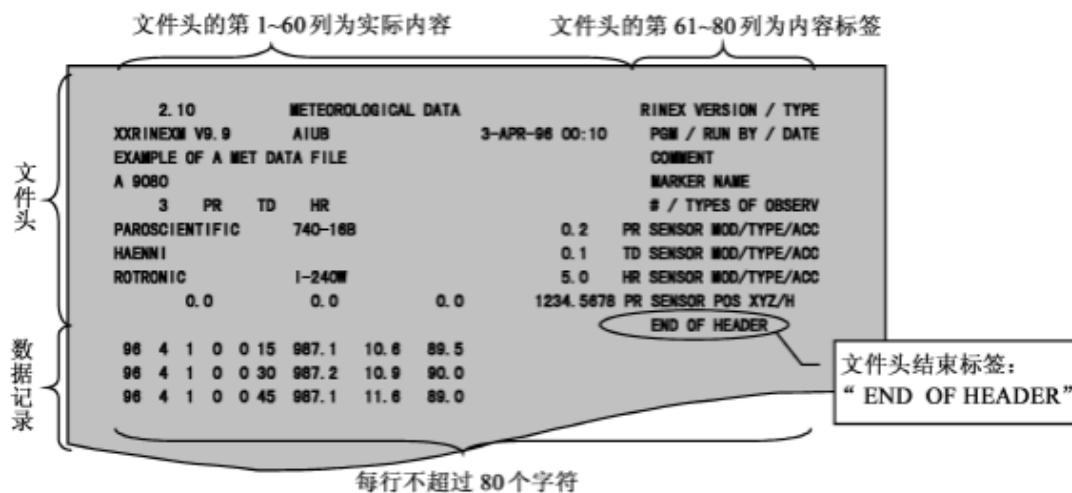


图 4-3 RINEX 格式气象数据文件的结构

4.1.4 RINEX 文件实例

实例 1 是一个 RINEX 格式的 GPS 观测数据文件。为了方便较为全面的了解，这里所给出的数据结构较为复杂，实际后总大多数数据文件结构要比它简单。

2.10 OBSERVATION DATA N (MIXED) RINEX VERSION / TYPE
BLANK OR G = GPS, R = GLOASS, T = TRANSIT, N = MIXED COMMENT
XXRINEX V0.9 AIUB 24-MAR-01 14:43 PGM / RUN BY / DATE

EXAMPLE OF A MIXED RINEX FILE

A 9080
9080.1.34
BILL SMITH
X1234A123
234

ABC INSTITUTE
XX
YY

4375274. 587488. 4589095.
.0030 .0000 .0000
1 1
1 2 8 G14 G15 G16 G17 G18 G19
0
4 P1 L1 L2 P2
18.000
2001 3 24 13 10 36.0000000

01 3 24 13 10 36.0000000 0 36120 98 8
23629347.915 .380 8 -.353 23629384.158
20891534.848 -.120 9 -.358 20891541.292
20807600.189 -.430 9 .334 20807605.848

01 3 24 13 10 50.0000000 4 4
1 2 2 8 9 G12
*** WAVELENGTH FACTOR CHANGED FOR 2 SATELLITES ***
NOW 8 SATELLITES HAVE WL FACT 1 AND 2!

01 3 24 13 10 54.0000000 0 56120 98 8R1R22
23619085.450 -53835.632 8 -41981.375 23619112.008
20888075.867 -28688.027 9 -22354.535 20888082.101
20811072.889 18247.789 9 14219.770 20811078.410
21345678.576 12345.567 5
22123456.789 23456.789 5

01 3 24 13 11 0.0000000 2 1
*** FROM NOW ON KINEMATIC DATA! ***
01 3 24 13 11 48.0000000 0 46180120 98 8
21110991.758 18119.980 7 12560.510 21110998.441
23588424.398 -215050.557 8 -167571.734 23588438.570
20889878.790 -113803.187 8 -88677.926 20889884.938
20821843.727 73787.482 7 57505.177 20821848.278

A 9080
9080.1.34
.0030 .0000 .0000
-> THIS IS THE START OF A NEW SITE <-

01 3 24 13 12 8.0000000 0 46180120 98 9
21112588.384 24515.877 8 19102.783 21112596.187
23578228.338 -288624.234 7 -209317.284 23578244.398
20825218.088 82581.207 7 72141.846 20825223.795
20864539.893 -141858.838 8 -110539.435 20864545.943

01 3 24 13 13 1.2345678 5 0
4 1
(AN EVENT FLAG WITH SIGNIFICANT EPOCH)
01 3 24 13 14 12.0000000 0 46180120 98 8
21124905.133 88551.30216 08778.62854 21124972.2754

测站近似坐标
在缺省情况下, L₁ 和 L₂ 相位为全波长。
天线高及偏心, 这里的天线高为 0.9030m。
6 颗卫星的观测值有例外, 其中 L₁ 相位为全波长, 而 L₂ 相位为半波长。
这里给出了文件中观测值类型的数量, 并指定了不同类型观测值的存储顺序。
这是一个历元的数据记录, 所占行数与卫星数和观测值类型数有关。在这里, 由于只有 4 类观测值, 故每颗卫星的记录只占一行。
根据卫星列表, 此行是 PRN 号为 9 的 GPS 卫星的记录。
本历元的卫星列表。卫星号前的“G”表示其为 GPS 卫星。
本历元的卫星数
历元标志。0 表示正常。
历元时刻
接收机钟差
在此记录后, 还有一行数据属于本历元。
历元标志。2 表示天线开始移动, 后面为动态数据。
根据文件头中的观测值类型列表, 这一块是此历元中各颗卫星 L₁ 上 P 码的观测值 (P₁)。
根据文件头中的观测值类型列表, 这一块是此历元各颗卫星的 L₂ 载波相位观测值 (L₂) 信息。注意, 观测值本身仅精确到小数点后 3 位, 后面的数字为 L₁ 和信号强度。

实例 2 是一个 RINEX 格式的 GPS 导航文件

2.10 N: GPS NAV DATA RINEX VERSION / TYPE
XXRINEX V2.10 AIUB 3-SEP-99 15:22 PGM / RUN BY / DATE
EXAMPLE OF VERSION 2.10 FORMAT COMMENT

.18760-07 .22350-07 -.11920-06 -.11920-06
.12080+06 .13100+06 -.13100+06 -.19880+06
.1331791281700-06 .1074695887800-12 552980 1025 DELTA-UTC: A0, A1, T, W
LEAP SECONDS
END OF HEADER

13
99 9 2 17 51 44.0 -.8397013880310-03 -.1859827830740-10 .000000000000+00
.010000000000+02 .834082500000+02 .1180405478400-08 .1820823048010+00
.4841014742850-05 .6267404183750-02 .8521120867480-05 .5153854890080+04
.408604000000+08 -.2421438864000-07 .3292370034600+00 -.5980464477540-07
.1115416631360+01 .326583750000+03 .2089567263350+01 -.6383123025550-08
.3071558514080-08 .000000000000+00 .102500000000+04 .000000000000+00
.000000000000+00 .000000000000+00 .000000000000+00 .910000000000-02
.408800000000+08 .000000000000+00

13 99 9 2 18 0 0.0 .4900256171820-03 .2048363078990-11 .000000000000+00
.133000000000+03 .883125000000+02 .1489704078220-08 .2829811521480+01
-.4988163708840-05 .2002383477800-02 .8281560778620-05 .5153284781430+04
.414000000000+08 -.2783987723850-07 .2430319399420+01 -.5587935447890-07
.1101827989300+01 .271187500000+03 -.2327579154250+01 -.8198329530570-08
-.7857470152310-11 .000000000000+00 .102500000000+04 .000000000000+00
.000000000000+00 .000000000000+00 .000000000000+00 .389000000000-03
.410400000000+08 .000000000000+00

卫星的 PRN 号
13 号卫星的导航电文数据, 通常两个小时更新一次。

实例 3 是一个 RINEX 格式的气象数据文件

2.10 METEOROLOGICAL DATA				RINEX VERSION / TYPE	
XXRINEX V2.0	AIUB	3-APR-98 00:10		PGM / RUN BY / DATE	
EXAMPLE OF A MET DATA FILE				COMMENT	
A 9080				MARKER NAME	
				# / TYPES OF OBSERV	
3	PR	TD	HR		
PAROSIDENTIFIC	740-188			0.2	PR SENSOR MOD/TY/ACC
HAENNI				0.1	TD SENSOR MOD/TY/ACC
ROTRONIC	1-240W			5.0	HR SENSOR MOD/TY/ACC
0.0	0.0	0.0		1234.5678	PR SENSOR POS XYZ/H
				END OF HEADER	
96	4	1	0	0 15	987.1 10.8 88.0
96	4	1	0	0 30	987.2 10.9 90.0
96	4	1	0	0 45	987.1 11.8 88.0

这里给出了文件中气象观测值类的数量及各类观测值的排列方式。

一个历元的气象观测值记录，其中不同类型的观测值按文件头中所给出的顺序依次列出。

4.2 SP3 格式

4.2.1 概述

SP3精密星历数据格式的全称是标准产品第3号（Standard Product# 3），它是一种在卫星大地测量中广泛采用的数据格式，由美国国家大地测量委员会（NGS）提出，专门用于存储 GPS卫星的精密轨道数据。

1985年，NGS公布了第一代轨道格式（SP1、ECF1、SP2、ECF2），之后又加入了第5种格式（EF13），这5种格式构成了NGS的第一代卫星轨道数据格式。1989年，NGS又提出了3种新轨道数据格式，分别为SP3、ECF3、EF18。

SP3格式文件是文本文件，其基本内容是卫星位置和卫星钟记录，另外，还可以包含卫星的运行速度和钟的变率。若在SP3格式文件第一行中有位置记录标记“P”，则表示文件中未包含卫星速度信息；若第一行中有速度记录标记“V”，则表示在文件中，对每一历元、每一卫星均已计算出了卫星的速度和钟的变率。

在SP3格式文件的第一行中，有一个专门表示轨道数据所属坐标参照系的字段。通常情况下，SP3轨道的计算和分发都是在一个国际地球参考框架下进行的。

SPS格式所涉及的时间均为GPS时，即使对采用格里高利或约化儒略日所标示的时间也是如此。因此，在SP3格式中，未包含用于将GPS转换为UTC的信息。

4.2.2 SP3 文件命名规则

SP3 格式对数据文件的命名与 RINEX 格式类似，由用于表示文件归属的 8 字符长度主文件名和用于表示文件类型的 3 位字符长度扩展名两部分组成，其具体形式如：

tttwwwd.sp3

其中，

ttt: 精密星历的类型；

www: GPS 周；

d: 星期，0 表示星期日，1~6 分别表示星期一~星期六。

例如: igs12901.sp3

igs (事后精密星历)、igr (快速精密星历)、igu (预报精密星历)

4.2.3 SP3 文件格式及特点

(1) 与 RINEX 格式文件一样, SP3 格式文件也是以节、记录、字段和列为单位逐级组织的, 并且同样分为文件头和数据记录两节, 每一节由若干记录所组成。所不同的是, SP3 格式文件的记录长度被严格限定为 60 列, 而不是 RINEX 格式文件的 80 列。

(2) SP3 格式文件文件头的第 1 行第 2 个字符为版本标识符。第 1 行由采用格里高利历所标示的首个历元的轨道数据时间、星历文件中的历元数、进行数据处理时所采用数据类型的描述符、轨道类型描述符和轨道发布机构描述符所组成。第 2 行所包含的内容

(3) 有轨道数据首个历元的 GPS 周、GPS 周以内的秒数、以秒为单位的历元间隔、约化儒略日的整数部分及小数部分。第 3~7 行为卫星的 PRN 号。在列出了所有的 PRN 号后, 剩下的位置用零值填充。第 8~12 行为卫星轨道精度指数, 若为 0, 表示精度未知。由精度指数 n 计算实际精度 σ 的方法为:

$$\sigma = 2^n$$

mm。例如, 如果精度指数为 5, 则实际精度为 $\sigma = 2^5$ mm。

第 13~18 行用来对 SP3 格式的文件进行一些扩充, 在文件中添加一些附加信息。第 19~22 行可为任意内容的注释。第 23 行为历元的日期和时间。第 24 行为卫星的位置 (或速度) 和钟差 (或钟漂)。

4.2.4 SP3 实例

实例 1 为精密星历文件:

卫星	卫星列表 (PRN列表)	精度	卫星坐标	卫星钟差
1	1137.662333	24450.419935	10434.025839	168.957503
2	22554.164846	6701.404774	-12156.548088	-354.131603
3	-12835.803249	19823.526060	12033.738279	12.655252
4	26036.203647	-5147.508988	-2778.935922	693.981565
5	-8981.105708	-20915.047563	-13623.748269	291.406843
6	-7243.996679	-19828.275194	16147.726225	-0.315725
7	15634.362803	-542.356344	-21103.968489	578.771698

实例2为P型文件:

```
#aP1994 12 17 0 0 0.00000000 96 d ITR92 FIT NGS
## 779 518400.00000000 900.00000000 49703 0.00000000000000
+ 25 1 2 4 5 6 7 9 12 14 15 16 17 18 19 20 21 22
+ 23 24 25 26 27 28 29 31 0 0 0 0 0 0 0 0 0 0
+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
++ 7 6 5 5 5 5 5 5 5 5 6 5 5 5 5 6 5 5
++ 5 5 6 5 5 5 5 5 0 0 0 0 0 0 0 0 0 0
++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
%e cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc
%e cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc
%f 0.0000000 0.000000000 0.00000000000 0.000000000000000
%f 0.0000000 0.000000000 0.00000000000 0.000000000000000
%i 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
%i 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
/* CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
/* CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
/* CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
/* CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
* 1994 12 17 0 0 0.00000000
P 1 16258.524750 -3529.015750 -20611.427050 -62.540600
P 2 -21998.652100 -8922.083550 -12229.824050 -131.326200
P 4 -26019.547600 4809.810900 -2508.578200 3.544600
P 5 7014.950200 21130.960300 -14387.334650 79.692800
*
*
*
P 28 13204.937750 -20485.533400 10794.787000 55.200800
P 29 -1638.431050 -24391.479200 10455.312650 3.690300
P 31 6265.255800 -25687.986950 -753.359000 70.830800
* 1994 12 17 0 15 0.00000000
P 1 15716.820135 -1169.850480 -21281.578766 -62.542746
P 2 -22813.261065 -9927.616864 -9816.490189 -131.328686
```

实例3为V型文件:

```
#AV1994 12 17 0 0 0.00000000 96 D ITR92 FIT NGS
## 779 518400.00000000 900.00000000 49703 0.00000000000000
+ 25 1 2 4 5 6 7 9 12 14 15 16 17 18 19 20 21 22
+ 23 24 25 26 27 28 29 31 0 0 0 0 0 0 0 0 0 0
+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
+ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
++ 7 6 5 5 5 5 5 5 5 5 6 5 5 5 5 6 5 5
++ 5 5 6 5 5 5 5 5 0 0 0 0 0 0 0 0 0 0
++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
++ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
%e cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc
%e cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc cc
%f 0.0000000 0.000000000 0.00000000000 0.000000000000000
%f 0.0000000 0.000000000 0.00000000000 0.000000000000000
%i 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
%i 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
/* CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
/* CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
/* CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
/* CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
* 1994 12 17 0 0 0.00000000
P 1 16258.524750 -3529.015750 -20611.427050 -62.540600
V 1 -8560.373522 25605.954994 -9460.427179 -0.024236
P 2 -21998.652100 -8922.083550 -12229.824050 -131.326200
V 2 -9852.750736 -12435.176313 25738.634180 -0.029422
P 4 -26019.547600 4809.810900 -2508.578200 3.544600
V 4 2559.038002 -3340.527442 -31621.490838 0.016744
*
*
*
P 29 -1638.431050 -24391.479200 10455.312650 3.690300
V 29 5754.005457 -12065.761570 -27707.056273 0.003537
P 31 6265.255800 -25687.986950 -753.359000 70.830800
V 31 3053.344058 -63.091750 31910.454757 0.033749
```

4.3 重要 MATLAB 函数的使用及讲解

4.3.1 读取 N 导航文件：

RINEXE.m: 读取RINEX导航信息文件，将信息以矩阵的格式写入文件。矩阵中有21行，每一列储存一颗卫星的信息。

调用方式： `rinexe('SITE247J.01N','eph.dat')`

读取星历文件'SITE247J.01N'中的星历信息，并以矩阵形式写入到文件'eph.dat'。

get_eph.m: 星历文件中包含的星历信息被存入矩阵。矩阵行数为21，列数为有星历信息的卫星个数

调用方式： `Eph = get_eph('eph.dat');`

读取'eph.dat'文件中的星历信息存入矩阵Eph中。

4.3.2 读取 O 数据文件

anheader.m: 分析Rinex文件的文件头，输出观测类型和天线设置类型。

调用方式： `[Obs_types, ant_delta, ifound_types, eof] = anheader(file)`

Obs_types: 观测值类型

ant_delta: 天线高及偏心

ifound_types: 找到相对应观测类型的标识码

eof: 判断文件是否结束的标识码

读取rinex观测文件的头文件信息，返回值为四个变量。

fepoch_0.m: 在打开的RINEX文件中利用识别标志fid寻找下一历元。

从历元序列中生成历元时间、卫星编号、文件结束标识符。

只有以O结束的观测文件才会被处理。

调用方式： `[time, dt, sats, eof] = fepoch_0(fid)`

time: 一周内的秒数（第几秒）

sats: 卫星PRN

eof: 文件结束标识符

读取fid对应rinex观测文件，返回某一历元数据的历元时间、卫星编号、文件结束标识符等。

Grabdata.m: 在选定的历元读取编号卫星对应的观测值，

调用方式： `Obs = grabdata(fid, NoSv, NoObs)`

读取fid对应rinex观测文件，根据卫星数NoSv和观测类型数NoObs返回大小为[NoSv, NoObs]一个历元的观测数据矩阵，当NoObs>5时，令NoObs=5。

fobs_typ: 返回观测值矩阵中的列i。其中包含观测类型‘type’。

调用方式: `i = fobs_typ(Obs_types, type)`

返回type (可以使L1、L2、C1、P1、P2等) 在rinex文件观测类型序列中的位置。

Julday.m: 将给出的年(四位数)、月、日、时转换成儒略时间。

只有在1900年3月1日至2100年2月28日, 转换才有效。

调用方式: `jd = julday(2011,10,17,12)`

输入时间2011年10月17日12时, 返回儒略日2455852。

gps_time.m: 将儒略日时间转换成GPS时间

调用方式: `[week,sec_of_week] = gps_time(2455852)`

输入儒略日时间, 返回GPS周和周内秒`week = 1658, sec_of_week = 129600`

4.3.3 读取 SP3 数据文件

ReadSP3.m: 读取SP3精密卫星轨道数据文件

调用方式: `[SP3data,numsat,header] = ReadSP3(filename)`

SP3data: 轨道信息

Numsat: 卫星数目

Header: 文件头

读取SP3文件filename, 并以矩阵形式返回数据信息。

FormSP3.m: 读取SP3精密卫星轨道数据文件, 并按卫星PRN存为矩阵形式

调用方式: `[Time,SP3X,SP3Y,SP3Z,Clock,remark,new_prn] =`

`FormSP3(filename,PRN,sep,lep)`

根据指定的卫星编号以及起止时间读取SP3文件filename, 并返回相关信息。

4.4 MATLAB 代码读取标准数据的程序结构

(1) 读取N导航文件:

1) 打开文件, 获取星历数据开始位置。利用findstr 及循环语句逐行查找字符串“END OF HEADER”, 获取星历数据开始位置, 并记录文件头行数。

2) 获取星历数目。利用循环将整个文件读取一遍, 得到数据部分的行数, 又每颗卫星的星历数据固定是8行, 从而得到历元数目。

3) 读取星历数据。重头读取星历文件, 跳过文件头, 利用循环读取每颗卫星对应的星历数据, 提取出计算卫星位置必要的21个参数, 按卫星存入矩阵eph。

4) 保存文件。将得到的星历数据矩阵写到文件中, 便于以后处理数据时使用。

(2) 读取O数据文件

1) 打开文件，获取文件头信息。利用`anheader`获取观测值类型、天线高及偏心信息。

2) 获取观测历元信息。利用`fepoch_0`获得某一历元的观测时间和卫星的PRN号序列。

3) 获取观测数据。根据卫星数目和观测类型数目利用`grabdata`获得观测数据，并以矩阵形式输出。

(3) 读取SP3数据文件

1) 打开文件，读取文件头。文件头共22行，使用`fgetl`函数循环读取。

2) 读取精密轨道信息。根据文件结构，分别读取卫星PRN号及其对应的轨道位置及钟差数据。

3) 加入时间信息。将时间信息和精密轨道信息存入同一个矩阵中输出。

第5章 卫星坐标计算

在利用GPS信号进行导航定位时，为解算用户在地心坐标系中的位置，GPS接收机需要测定测站到卫星的距离并且要知道同一卫星在同一时刻的地心坐标。卫星的地心坐标是从卫星的导航电文中提供的开普勒轨道参数和轨道摄动修正量按一定公式计算的。下面就详细叙述GPS卫星计算的步骤，首先介绍开普勒定律。

5.1 开普勒定律

第一定律：

行星在通过太阳的平面内沿椭圆轨道运行，太阳位于椭圆的一个焦点上。这就是行星运动第一定律，又叫“轨道定律”。

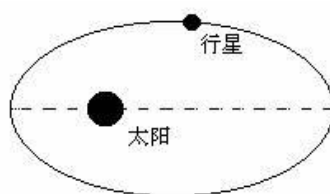


图 5-1 开普勒第一定律

第二定律：

在椭圆轨道上运行的行星速度不是常数，而是在相等时间内，行星与太阳的连线所扫过的面积相等。这就是行星运动第二定律，又叫“面积定律”。

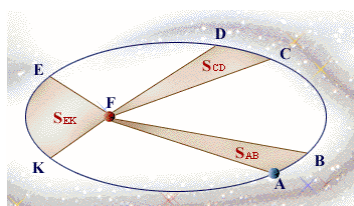


图 5-2 $S_{AB} = S_{CD} = S_{EK}$

第三定律：

太阳系内所有行星公转周期的平方同行星轨道半长径的立方之比为一常数，这一定律也叫“调和定律”。

$$n = \sqrt{\frac{GM}{a^3}}$$

5.2 开普勒轨道参数

5.2.1 开普勒轨道参数

卫星的空间位置主要由六个开普勒轨道参数来描述，为了简化研究，我们可以近似的认为卫星只受地心引力作用，在天体力学中将其称为二体问题。卫星运动简化为二体问题后，忽略了太阳、月亮对卫星的引力，大气阻力，太阳光压，地球潮汐等作用力，此时，卫星的运动轨迹简化为用数学公式表达的椭圆轨道，地球位于椭圆的一个焦点。但是，将卫星绕地球的复杂运动看作为二体问题需要满足以下四个条件：

- (1) 将地球看作一个质点，或者密度均匀分布的球体；
- (2) 与地球质量相比，卫星的质量可以忽略不计；
- (3) 认为卫星运动的环境属于真空状态，不考虑大气和光压等阻力；
- (4) 忽略了太阳，月亮，其他星体对卫星的引力。

卫星的运动轨道见图5-3，采用真天球坐标系。卫星的运动轨迹为一椭圆，地球质心 F 位于椭圆的一个焦点， X 轴指向春分点， Z 轴指向地球自转轴， Y 轴按构成右手坐标系取向。卫星运动轨道平面与赤道平面的交线称为节线，卫星作上升运动时与赤道的交点称为升交点。升交点的赤径用 Ω 表示，近地点角距用 ω 表示，真近点角用 f 表示（ ω 和 f 位于卫星轨道平面上），轨道面的倾角用 i 表示，开普勒椭圆长半轴用 a 表示，扁率用 e 表示。由图5-3可以看出，（ Ω ， i ）唯一地确定了卫星轨道平面在真天球坐标系中的位置；（ Ω ， ω ， i ）确定了开普勒椭圆在轨道平面上的定向；（ a ， e ， f ）唯一的确定了卫星轨道的形状、大小和卫星在轨道上的瞬时位置。

（ a ， e ， f ， Ω ， i ， ω ）称为开普勒六参数。其中，真近点角 f 是与时间有关的函数，其它参数都是常数，是由卫星的发射条件决定的。

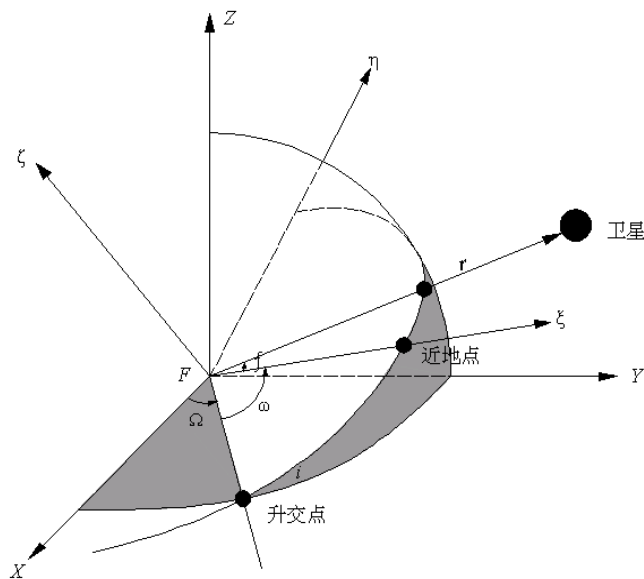


图 5-3 开普勒轨道六参数

5.2.2 卫星运动力学方程

二体问题在研究卫星空间运动中起了很大的作用，它将运动方程简单地表达出来。因为它以地球质心为椭圆轨道的一个焦点，除了地球对卫星的引力外，忽略了其他阻力。它可以计算短时间内、低精度的卫星位置，还可以构建卫星的可视图，二体问题的描述是在惯性坐标系里进行。

根据牛顿定律，可以方便地得到二体问题的卫星运动方程。卫星和地球间的引力大小为

$$F = \frac{k^2 m M}{r^2} \quad (5-1)$$

其中 k^2 是常数， M 和 m 分别表示地球和卫星的质量， r 是地球和卫星间的距离。

而地球对卫星的引力可表示为

$$F = ma \quad (5-2)$$

其中 a 是卫星在引力作用下所产生的加速度。

根据以上两式，可得加速度表达式

$$a = \frac{k^2 M}{r^2} \quad (5-3)$$

表达为向量形式

$$\ddot{\vec{r}}_s = -k^2 M \frac{\vec{r}}{r^3} = -\mu \frac{\vec{r}}{r^3} \quad (5-4)$$

其中 μ 为常量

$$\mu = k^2 M \quad (5-5)$$

若包括地球质量包括大气层， $\mu = 3986005 \times 10^8 m^3 s^{-2}$ 。

由于力的作用是相互的，地球同样受到万有引力，地球在万有引力作用下同样产生加速度，同理于卫星加速度表达式（5-3、5-4），地球在卫星引力作用下产生的加速度表达式如下：

$$\ddot{\vec{r}}_e = k^2 m \frac{\vec{r}}{r^3} = \mu \frac{\vec{r}}{r^3} \quad (5-6)$$

其中

$$\mu = k^2 m \quad (5-7)$$

因为牛顿第二定律只适用于惯性坐标系，故（5-4、5-6）式为卫星和地球在某一惯性坐标系内的运动方程。若要讨论卫星相对于地球质心的运动，必须讲坐标系原点移至地球质心，可得卫星相对于地球质心的加速度 $\ddot{\vec{r}}$ 如下：

$$\ddot{\vec{r}} = \ddot{\vec{r}}_e - \ddot{\vec{r}}_s = -k^2 (M + m) \frac{\vec{r}}{r^3} \quad (5-8)$$

在讨论卫星与地球的二体问题时，由于地球质量远远大于卫星质量，通常略去卫星质量 m 项，式（5-8）可以写为：

$$\ddot{\vec{r}} = \ddot{\vec{r}}_e - \ddot{\vec{r}}_s = -k^2 M \frac{\vec{r}}{r^3} \quad (5-9)$$

设以O为原点的直角坐标系为O-XYZ，卫星的坐标为(X, Y, Z)，则卫星的地心向量 $\vec{r} = (X, Y, Z)$ ，加速度 $\ddot{\vec{r}} = (\ddot{X}, \ddot{Y}, \ddot{Z})$ ，代入（5-9）式得：

$$\left. \begin{aligned} \ddot{X} &= -\mu \frac{X}{r^3} \\ \ddot{Y} &= -\mu \frac{Y}{r^3} \\ \ddot{Z} &= -\mu \frac{Z}{r^3} \end{aligned} \right\} \quad (5-10)$$

其中， $r = \sqrt{X^2 + Y^2 + Z^2}$ 。

（5-10）式就是卫星大地测量中常用的在地心直角坐标系中二体问题分量形式的微分方程。它是三个二阶非线性微分方程组。

解（5-10）式，必须找出包含有六个相互独立的积分常数，这六个积分常数可以用上述六个轨道参数代替。其解的一般形式为：

$$\left. \begin{aligned} r &= g(a, e, i, \Omega, \omega, \tau, t) \\ \frac{dr}{dt} &= g'(a, e, i, \Omega, \omega, \tau) \end{aligned} \right\} \quad (5-11)$$

从（5-10）式可以看出，在二体问题情况下，给定六个轨道参数，即可确定任意时刻 t 的卫星位置及其运动速度。

5.3 卫星轨道计算

5.3.1 卫星轨道方程

图5-4中定义了以卫星轨道平面为基准面的三维坐标系（ $o-\xi, \eta, \varsigma$ ），原点在地球质心C， ξ 指向近地点， η 指向降交点， ς 垂直轨道平面，偏近点角E，真近点角f 如图所示。同时定义坐标系 $p-o-q$ ，原点在椭圆轨道的球心，p，q轴与 ξ ， η 分别平行。

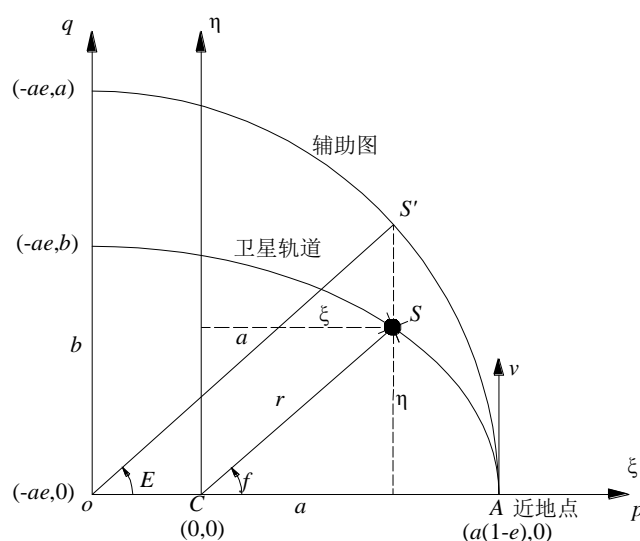


图 5-4 偏近点角 E 与真近点角 V

由开普勒第一定律，卫星的运行轨迹为椭圆，在 $p-o-q$ 坐标内有：

$$\frac{p^2}{a^2} + \frac{q^2}{b^2} = 1 \quad (5-12)$$

其中a，b为卫星运行轨迹的长、短轴。

卫星S的位置在 $\xi-C-\eta$ 坐标系中（ ξ ， η ）为：

$$\xi = r \cos f = a \cos E - ae = a(\cos E - e) \quad (5-13)$$

再由公式（5-12）：

$$q = \sqrt{\left(1 - \frac{p^2}{a^2}\right)b^2} \quad (5-14)$$

$$p = a \cos E \quad (5-15)$$

将（5-15）带入（5-14）可得：

$$\eta = q = b \sin E = a \sqrt{1-e^2} \sin E \quad (5-16)$$

则在 $\xi-\eta-\zeta$ 坐标系下，卫星相对地球质心的距离向量

$$r = \begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} = \begin{bmatrix} a(\cos E - e) \\ a \sqrt{1-e^2} \sin E \\ 0 \end{bmatrix} \quad (5-17)$$

卫星与地球质心的几何距离为：

$$\|r\| = a(1 - e \cos E) \quad (5-18)$$

其中， E 随着时间的变化而变化，以上是以开普勒第一定律为基础推导卫星相对地球的位置表达式结果。

5.3.2 开普勒方程

定义 T 为卫星沿椭圆轨道运动周期， n 为平均运动角速度，则平均角速度为：

$$n = \frac{2\pi}{T} \quad (5-19)$$

则开普勒第三定律可表示为：

$$n = \sqrt{\frac{GM}{a^3}} \quad (5-20)$$

其中， GM 为地球重力常量，数值为 $3986005 \times 10^8 m^3 s^{-2}$ 。

假定 t_0 为卫星通过近地点 A 的时刻，对应 t 时刻的平近点角：

$$\mu(t) = n(t - t_0)$$

对应任意 t 时刻的开普勒方程可描述为：

$$E = \mu + e \sin E \quad (5-21)$$

此方程建立了偏近点角 E 与平近点角 μ 之间的关系，通过迭代可以求解 t 时刻偏近点角 E 的数值。带入公式（5-17）即可求得卫星在 $\xi-\eta-\zeta$ 坐标系中的坐标，通过坐标转换可转换为地固坐标系的坐标。

再结合方程（5-17）可得到真近点角 f 可表示为：

$$f = \arctan \frac{\eta}{\xi} = \arctan \frac{\sqrt{1-e^2} \sin E}{\cos E - e} \quad (5-22)$$

公式（5-22）给出了任意时刻 t ，对应的偏近点角 E 与真近点角 f 之间的关系，轨道扁率 e 不随时间变化而变化。

5.3.3 卫星坐标计算

卫星轨道平面相对于地心坐标系（ $O-X,Y,Z$ ）的位置基本不变，即从空间中

看，卫星轨道平面相对地球赤道平面的位置不变。地球格林尼治子午面绕地球自转轴旋转，约24小时/天。GPS卫星在椭圆轨道上，以3.87km/s的速度每天旋转两圈， k 卫星在 t_j 时刻的地心坐标可表示为：

$$\begin{bmatrix} X^K(t_j) \\ Y^K(t_j) \\ Z^K(t_j) \end{bmatrix} = R_3(-\Omega_j^k) R_3(-i_j^k) R_3(-\omega_j^k) \begin{bmatrix} r_j^k \cos f_j^k \\ r_j^k \sin f_j^k \\ 0 \end{bmatrix} \quad (5-23)$$

其中， Ω_j^k ， i_j^k ， ω_j^k 分别为 k 卫星 t_j 时刻的升交点赤经，卫星轨道面倾角，近地点角距。

$$R_3(-\Omega_j^k) = \begin{bmatrix} \cos \Omega & -\sin \Omega & 0 \\ \sin \Omega & \cos \Omega & 0 \\ 0 & 0 & 1 \end{bmatrix} R_3(-i_j^k) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i & -\sin i \\ 0 & \sin i & \cos i \end{bmatrix}$$

$$R_3(-\omega_j^k) = \begin{bmatrix} \cos \omega & -\sin \omega & 0 \\ \sin \omega & \cos \omega & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

卫星 k 与地球质心距离 $r_j^k = \|r(t_j)\|$ ，由（5-18）式知， a ， e 为定值， E 随时间 t_j 变化而变化。在计算卫星位置时，可以从广播星历中得到这些参数，将其带入上面的公式计算即可。

实际的卫星绕地球运动，还会受摄动力的影响，摄动力参数和开普勒六参数相似，也由广播星历提供。广播星历的精度为10m左右，显然不能满足高精度的大地测量，可以选择使用精密星历解算卫星位置，精度可达到分米级。

计算卫星位置需要广播星历中以下参数：

μ ， Δn ， μ ， \sqrt{a} ， Ω ， I ， ω ， $\dot{\Omega}$ ， \dot{I} ， C_{uc} ， C_{us} ， C_{rc} ， C_{rs} ， C_{ic} ， C_{is} ， t_{oe} 。其中， $\dot{\Omega} = \partial \Omega / \partial t$ ， $\dot{I} = \partial I / \partial t$ ， C_{uc} ， C_{us} ， C_{rc} ， C_{rs} ， C_{ic} ， C_{is} 为纬度幅角，轨道半径，轨道倾角的正余弦调和项改正，单位为弧度，是由于摄动力而引起的改正项。

5.4 卫星坐标计算算法步骤

5.4.1 算法

(1) 计算卫星运行的平均角速度 n

根据开普勒第三定律，卫星运行的平均角速度 n_0 可以用下式计算：

$$n_0 = \sqrt{k^2 M / a^3} = \sqrt{\mu} / (\sqrt{a})^3 \quad (5-24)$$

式中， μ 为WGS-84坐标系中的地球引力常数，且 $\mu = 3.986005 \times 10^{14} m^3 / s^2$ 。平

均角速度 n_0 加上卫星电文给出的摄动改正数 Δn ，便得到卫星运行的平均角速度 n

$$n = n_0 + \Delta n \quad (5-25)$$

(2) 计算归化时间 t_k

首先对观测时刻 t' （以子夜零时为开始时刻）作卫星钟差改正

$$t = t' - \Delta t$$

$$\Delta t = a_0 + a_1(t' - t_{oc}) + a_2(t' - t_{oc})^2$$

然后将观测时刻 t 归化到GPS时系

$$t_k = t - t_{oe} \quad (5-26)$$

式中， t_k 称作相对于参考时刻 t_{oe} 的归化时间（注意： $t_{oc} \neq t_{oe}$ ）

(3) 观测时刻卫星平近点角 M_k 的计算

$$M_k = M_0 + nt_k \quad (5-27)$$

式中， M_0 是卫星电文给出的参考时刻 t_{oe} 的平近点角。

(4) 计算偏近点角 E_k

$$E_k = M_k + e \sin E_k \quad (E_k, M_k \text{ 以弧度计}) \quad (5-28)$$

上述方程可用迭代法进行解算，即先令 $E_k = M_k$ ，代入上式，求出 E_k 再代入上式计算，因为GPS卫星轨道的偏心率 e 很小，因此收敛快，只须迭代计算两次便可求得偏近点角 E_k 。

(5) 真近点角 f_k 的计算

$$f_k = \arctan \frac{\eta}{\xi} = \arctan \frac{\sqrt{1-e^2} \sin E}{\cos E - e} \quad (5-29)$$

(6) 升交距角 Φ_k 的计算

$$\Phi_k = V_k + \omega \quad (5-30)$$

式中， ω 为导航电文给出的近地点角距。

(7) 摄动改正项 δ_u ， δ_r ， δ_i 的计算

$$\begin{cases} \delta_u = C_{uc} \bullet \cos(2\Phi_k) + C_{us} \bullet \sin(2\Phi_k) \\ \delta_r = C_{rc} \bullet \cos(2\Phi_k) + C_{rs} \bullet \sin(2\Phi_k) \\ \delta_i = C_{ic} \bullet \cos(2\Phi_k) + C_{is} \bullet \sin(2\Phi_k) \end{cases} \quad (5-31)$$

δ_u ， δ_r ， δ_i 分别为升交距角 u 的摄动量，卫星矢径 r 的摄动量和轨道倾角 i 的摄动量。

(8) 计算经过摄动改正的升交距角 u_k 、卫星矢径 r_k 和轨道倾角 i_k

$$\begin{cases} u_k = \Phi_k + \delta_u \\ r_k = a(1 - e \cos E_k) + \delta r \\ i_k = i_0 + \delta_i + \dot{I}t_k \end{cases} \quad (5-32)$$

(9) 计算卫星在轨道平面坐标系的坐标

卫星在轨道平面直角坐标系（X轴指向升交点）中的坐标为

$$\begin{cases} x_k = r_k \cos u_k \\ y_k = r_k \sin u_k \end{cases} \quad (5-33)$$

(10) 观测时刻升交点经度 Ω_k 等于观测时刻升交点赤径 Ω （春分点和升交点之间的角距）与格林尼治视恒星时GAST（春分点和格林尼治起始子午线之间的角距）之差，

$$\Omega_k = \Omega - \text{GAST} \quad (5-34)$$

又因为

$$\Omega = \Omega_{oe} + \dot{\Omega}t_k \quad (5-35)$$

式中， Ω_{oe} 为参考时刻 t_{oe} 的升交点的赤径；

$\dot{\Omega}$ 是升交点赤径的变化率，卫星电文每小时更新一次 $\dot{\Omega}_{oe}$ 和 t_{oe} 。

此外，卫星电文中提供了一周的开始时刻 t_w 的格林尼治视恒星时GAST_w。由于地球自转作用，GAST不断增加，所以

$$\text{GAST} = \text{GAST}_w + \omega_e t \quad (5-36)$$

式中， $\omega_e = 7.29211567 \times 10^{-5} \text{ rad/s}$ 为地球自转的速率； t 为观测时刻。

由式（5-35）和式（5-36），得

$$\Omega_k = \Omega_{oe} + \dot{\Omega}t_k - \text{GAST}_w - \omega_e t \quad (5-37)$$

由式（5-34）得

$$\Omega_k = \Omega_0 + (\dot{\Omega} - \omega_e)t_k - \omega_e t_{oe} \quad (5-38)$$

式中， $\Omega_0 = \Omega_{oe} - \text{GAST}_w$ ， Ω_0 、 Ω 、 t_{oe} 的值都可以从卫星电文中获取。

(11) 计算卫星在地心固定坐标系中的直角坐标

把卫星在轨道平面直角坐标系中的坐标进行旋转变换，可得出卫星在地心固定坐标系中的三维坐标：

$$\begin{bmatrix} X_k \\ Y_k \\ Z_k \end{bmatrix} = \begin{bmatrix} x_k \cos \Omega_k - y_k \cos i_k \sin \Omega_k \\ x_k \sin \Omega_k + y_k \cos i_k \cos \Omega_k \\ y_k \sin i_k \end{bmatrix} \quad (5-39)$$

5.4.2 重要 MATLAB 函数说明

在编写计算每颗卫星位置程序时，每颗卫星都需要存储导航电文（*.n文件）中21个参数（存放在eph星历矩阵中），分别为svprn, af2, M0, roota, deltan, ecc, omega, cuc, cus, crc, crs, i0, idot, cic, cis, Omega0, Omegadot, toe, af0, af1, toc。详细情况对照代码学习。

程序实现主要分为三个模块实现：时间转换模块；读广播星历文件（n文件）模块；计算卫星位置模块。

下面简单回忆时间转换和读星历文件模块所需要的函数和实现的功能，在此基础上学习计算卫星位置模块。

（1）时间转换模块

主要用到两个函数，分别如下：

1) `jd=julday(year,month,day,hour)`

year—年，四位，如2011；

month—月，1-12；

day—日；

hour—时间，小时+不足一小时部分（用小数表示），如：3点45分4秒，

`hour=3+45/60+4/3600`；

返回值—儒略日。

2) `[week,sow]=gps_time(jd)`

jd—儒略日；

返回值—week, GPS周；sow, GPS秒。

（2）读星历文件模块

主要用到读n文件函数和读星历文件函数，分别如下：

1) `rinexe(ephemerisfile, outputfile)`

ephemerisfile—n文件名（注意：n文件要放在rinexe函数所在目录下）；

outputfile—文本文件名，将读到n文件的数据存放在此文本文件中；

返回值—无。

2) `eph = get_eph(ephemeridesfile)`

ephemeridesfile—文件名，将存放在此文本中的导航电文数据提取出来，为计算卫星位置做准备（注意：ephemeridesfile文件要存放在get_eph函数所在目录下）。

返回值—eph星历矩阵，将读到的数据存入内存（注意：eph星历矩阵，每颗卫星存放21个参数）

（3）计算卫星位置模块

主要用到四个函数，分别为读o文件函数、eph星历矩阵的选择函数、修复函

数（修复GPS时间超限或溢出）、计算卫星位置函数。

1) 读观测值文件 (*.o文件) 函数

在求解卫星位置时，第一需要利用o文件中每个历元的历元时刻 t 。在计算某时刻卫星位置时，这里的某时刻便是o文件历元时刻 t 。第二需要利用读取的每个历元不同的卫星PRN号。根据PRN号和历元时刻 t 在广播星历n文件中搜索相同的卫星PRN号、合适的历元时刻，利用其对应的数据，计算卫星位置。

2) $\text{col_Eph}(t) = \text{find_eph}(\text{Eph}, \text{sats}(t), \text{time})$

Eph—星历矩阵，存放计算卫星位置所需的n文件数据

sats(t)—sats中存放o文件中某个历元观测到的卫星PRN号， t 为循环控制，sats(t)为sats中某颗卫星。

time—o文件中某个历元的历元时刻

返回值—col_Eph矩阵存放从Eph星历矩阵中选出的用于计算的列数，如col_Eph=[1, 2]，代表Eph中第1，第2列用于计算卫星位置。

3) $\text{dt} = \text{check_t}(\text{time})$

time—儒略日；

返回值—修复后的儒略日。

4) $\text{X} = \text{satpos}(\text{tx_GPS}, \text{Eph}(:,k))$

tx_GPS—上节所述的归化时间，用儒略日表示的；

Eph(:,k)—Eph星历矩阵中的某一系列数据；

返回值—卫星在地心地固坐标系中坐标。

5.4.3 重要代码及标注

下面为计算卫星位置模块主要代码，附有标注。

(1) %主循环部分

```
for i = 1:m
    k = col_Eph(i);
    tx_RAW = time - obs(i)/v_light;
    t0c = Eph(21,k);
    dt = check_t(tx_RAW-t0c); % 得到GPS归化时间
    tcorr = (Eph(2,k)*dt + Eph(20,k))*dt + Eph(19,k);
    tx_GPS = tx_RAW-tcorr;
    X = satpos(tx_GPS, Eph(:,k));% satpos函数实现卫星位置的计算
end
```

(2) %satpos 函数部分

```
function satp = satpos(t,eph);
```

```
GM = 3.986008e14; % GM为地球重力常量
Omegae_dot = 7.2921151467e-5; % Omegae_dot为地球自转的速率
```

```
svprn = eph(1);
af2 = eph(2);
M0 = eph(3);
roota = eph(4);
deltan = eph(5);
ecc = eph(6);
omega = eph(7);
cuc = eph(8);
cus = eph(9);
crc = eph(10);
crs = eph(11);
i0 = eph(12);
idot = eph(13);
cic = eph(14);
cis = eph(15);
Omega0 = eph(16);
Omegadot = eph(17);
toe = eph(18);
af0 = eph(19);
af1 = eph(20);
toc = eph(21);
```

```
% 坐标计算过程
```

```
A = roota*roota;
tk = check_t(t-toe); % 返回修复后的儒略日
n0 = sqrt(GM/A^3); % 计算平均角速度
n = n0+deltan;
M = M0+n*tk; % 计算观测时刻卫星平近点角
M = rem(M+2*pi,2*pi); % 计算观测时刻卫星偏近点角
E = M;
for i = 1:10 % 循环迭代10次
```

```

E_old = E;
E = M+ecc*sin(E);
dE = rem(E-E_old,2*pi);
if abs(dE) < 1.e-12
    break;
end
end
E = rem(E+2*pi,2*pi);
v = atan2(sqrt(1-ecc^2)*sin(E), cos(E)-ecc);      %计算真近点角
phi = v+omega;                                     %计算升交距角
phi = rem(phi,2*pi);
u = phi+ cuc*cos(2*phi)+cus*sin(2*phi);           %计算摄动改正项
r = A*(1-ecc*cos(E)) + crc*cos(2*phi)+crs*sin(2*phi);
i = i0+idot*tk      + cic*cos(2*phi)+cis*sin(2*phi);
Omega = Omega0+(Omegadot-Omegae_dot)*tk-Omegae_dot*toe;      %计
算升交点的经度
Omega = rem(Omega+2*pi,2*pi);                      %变化为弧度0-2pi
x1 = cos(u)*r;
y1 = sin(u)*r;      %计算卫星在轨道平面坐标系中的位置
satp(1,1) = x1*cos(Omega)-y1*cos(i)*sin(Omega);
satp(2,1) = x1*sin(Omega)+y1*cos(i)*cos(Omega);      %计算卫星地心
固定坐标系中的直角坐标
satp(3,1) = y1*sin(i);

```

第6章 伪距定位及DOP值计算

精度因子（dilution of precision, DOP）因计算简单表达直观成为评价卫星导航定位性能的重要指标之一。在GPS定位技术的应用中，《工程测量规范》及《煤矿测量规程》等均对其作出要求以满足城市和矿区地面形变测量的精度要求。在区域导航定位中，常使用增设伪卫星等手段以改变卫星分布的几何结构，降低DOP值从而提高导航定位精度。在同等用户等效距离误差下，其DOP值愈小，星座分布结构越好，定位精度愈高。

6.1 引言

（1）DOP的数学意义

DOP是基于最小二乘平差解的权逆矩阵对角线的重要组成部分，它反映着未知参数协方差对角线元素与畏惧等效方差的比值，体现着观测信息对结算的未知参数的贡献程度。

（2）DOP的测量意义

由于测量卫星及接收机空间几何分布的影响带来的用户等效距离误差（UERE）到最终定位误差或定时误差的放大系数称之为DOP，也称之为精度衰减因子。据用户所使用的不同的定位模式，DOP可细分为绝对定位因子及相对定位因子。其中，依用户所关注的不同性能指标，DOP又可细分为GDOP、PDOP、HDOP、VDOP和TDOP。

6.2 伪距定位

在某一时刻 t ，卫星在卫星钟的控制下发出某一结构的测距码，与此同时，接收机在接收机钟的控制下产生或复制出相同的测距码（复制码）。卫星发出的测距码经 Δt 时间传播后为接收机接受。由接收机所产生的复制码经一个时间延迟器延迟时间 Γ 之后与接收机接收到的卫星信号比对。若两个信号未对齐，则调整延迟器延迟时间 Γ ，直至对齐为止。此时，复制码的延迟时间就等于卫星信号的传播时间，则卫地间的伪距 ρ ：

$$\rho = \Delta t \times c \quad (6-1)$$

因卫星钟和接收机钟均不可避免的存在误差，因此上述公式将不可避免地受到钟不同步的影响，此外卫星信号还需经过电离层和对流层才能到达地面测站，在电离层和对流层中信号的传播速度不等于光速，所以上述公式所得距离并不是真正的卫星至地面测站的真正距离，故称之为伪距。

考虑到接收机钟差、卫星钟差以及信号经过电离层和对流层的延迟，式(6-1)可以改写成：

$$\rho' = \rho_i' - V_{ion} - V_{trop} + cV_{t_i^s} - cV_{tr} \quad (6-2)$$

设第*i*颗卫星观测瞬间在空间的位置为 $(X^i, Y^i, Z^i)^T$ ，接收机在空间的位置为 $(X, Y, Z)^T$ 。则从接收机至卫星的几何距离可写为：

$$\rho_i' = \sqrt{(X^i - X)^2 + (Y^i - Y)^2 + (Z^i - Z)^2} \quad (6-3)$$

考虑到卫星星历误差在传播路径上的投影、多路径误差以及测量噪声可以通过选择高精度星历、选择合适观测站、通过数理统计概述等方法进行减弱，将式(6-3)带入式(6-2)得伪距测量的观测方程：

$$\rho' = \sqrt{(X^i - X)^2 + (Y^i - Y)^2 + (Z^i - Z)^2} - V_{ion} - V_{trop} + cV_{t_i^s} - cV_{tr} \quad (6-4)$$

GPS卫星发射测距信号和导航电文，导航电文含有卫星的位置信息。用户用GPS接收机在某一时刻同时接收三颗以上的GPS卫星信号，测量出测站点（接收机天线中心）P至三颗以上GPS卫星的距离并结算出该时刻GPS卫星的空间坐标，据此利用距离交会法结算处测站P的位置。

设时刻*t_i*在测站点P用GPS接收机同时测得P点至三颗GPS卫星S1、S2、S3的距离 ρ^1, ρ^2, ρ^3 。则有：

$$\begin{cases} \rho^1 = \sqrt{(X - X^1)^2 + (Y - Y^1)^2 + (Z - Z^1)^2} \\ \rho^2 = \sqrt{(X - X^2)^2 + (Y - Y^2)^2 + (Z - Z^2)^2} \\ \rho^3 = \sqrt{(X - X^3)^2 + (Y - Y^3)^2 + (Z - Z^3)^2} \end{cases} \quad (6-5)$$

非线性方程线性化方法主要通过泰勒级数展开或非线性最小二乘方法解决。对于GPS伪距方法主要通过泰勒展开对伪距观测方程进行线性化。

对不同历元对不同卫星同步观测的伪距观测方程中含有观测站坐标及接收机钟差四个参数，设 $(X_0, Y_0, Z_0)^T, (\delta_x, \delta_y, \delta_z)$ 分别为观测站坐标的近似值与改正数，将伪距观测方程进行泰勒展开并令：

$$\begin{cases} (d_\rho / d_x)_{x_0} = (X_s^j - X_0) / \rho_0^j = l^j \\ (d_\rho / d_y)_{y_0} = (Y_s^j - Y_0) / \rho_0^j = m^j \\ (d_\rho / d_z)_{z_0} = (Z_s^j - Z_0) / \rho_0^j = n^j \end{cases} \quad (6-6)$$

若可见卫星数 $n \geq 4$ ，则依预报卫星的位置及初始测站坐标并令 $c\delta t_k = \delta\rho$ ，伪距观测线性化方程组形式可写为：

$$\begin{bmatrix} \rho_0^1 \\ \rho_0^2 \\ \vdots \\ \rho_0^n \end{bmatrix} - \begin{bmatrix} l^1 m^1 n^1 - 1 \\ l^2 m^2 n^2 - 1 \\ \vdots \\ l^n m^n n^n - 1 \end{bmatrix} \begin{bmatrix} \delta_x \\ \delta_y \\ \delta_z \\ \delta_\rho \end{bmatrix} = \begin{bmatrix} \rho^1 + \delta\rho_1^1 + \delta\rho_2^1 - c\delta t^1 \\ \rho^2 + \delta\rho_1^2 + \delta\rho_2^2 - c\delta t^2 \\ \vdots \\ \rho^n + \delta\rho_1^n + \delta\rho_2^n - c\delta t^n \end{bmatrix} \quad (6-7)$$

6.3 定位精度因子 DOP 的计算

伪距观测方程线性化之后，令

$$A_i = \begin{bmatrix} l^1 m^1 n^1 - 1 \\ l^2 m^2 n^2 - 1 \\ \vdots \\ l^n m^n n^n - 1 \end{bmatrix}, \quad \begin{aligned} \delta_X &= (\delta x, \delta y, \delta z, \delta \rho)^T \\ L^j &= \rho^j + \delta\rho_1^j + \delta\rho_2^j + c\delta t^j - \rho_0^j \\ L_i &= (L^1, L^2, \dots, L^n)^T \end{aligned} \quad (6-8)$$

依最小二乘知：

$$Q_x = (A_i^T A_i)^{-1} \quad (6-9)$$

Q_x 即 t 时刻未知参数的协因数阵在空间直角坐标中的一般形式为：

$$Q_x = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ q_{41} & q_{42} & q_{43} & q_{44} \end{bmatrix} \quad (6-10)$$

实际生产中因根据需要需提高某方向上的定位精度故常采用其在大地坐标系中的表达形式来估算测站点的位置精度，依据方差与协方差传播定律可得：

$$Q_B = R \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix} R^T = \begin{bmatrix} \dot{q}_{11} & \dot{q}_{12} & \dot{q}_{13} \\ \dot{q}_{21} & \dot{q}_{22} & \dot{q}_{23} \\ \dot{q}_{31} & \dot{q}_{32} & \dot{q}_{33} \end{bmatrix} \quad (6-11)$$

其中： $R = \begin{bmatrix} -\sin\alpha\cos\beta & -\sin\alpha\sin\beta & \cos\alpha \\ -\sin\beta & \cos\beta & 0 \\ \cos\alpha\cos\beta & \cos\alpha\sin\beta & \sin\alpha \end{bmatrix}$ 为旋转矩阵， α 、 β 分别为测站的经、纬度。

则 t 时刻，绝对定位精度因子为：

3、各类 DOP 计算

1) 平面位置精度因子 HDOP: $HDOP = \sqrt{\dot{q}_{11} + \dot{q}_{22}}$

2) 高程精度因子 VDOP: $VDOP = \sqrt{\dot{q}_{33}}$

3) 空间位置精度因子 PDOP: $PDOP = \sqrt{\dot{q}_{11} + \dot{q}_{22} + \dot{q}_{33}}$

4) 接收机钟差精度因子 TDOP: $TDOP = \sqrt{\dot{q}_{44}}$

5) 几何精度因子GDOP: $GDOP = \sqrt{q_{11} + q_{22} + q_{33} + q_{44}}$

6.4 重要 MATLAB 函数说明

模拟星座生成函数gensv

```
function [svxyzmat,svid] = gensv(usrxyz,time,maskang,orgxyz)
global SVIDV MV OMGV RV INCLV TOEV
%判断用户初始位置
if nargin<4,orgxyz=[];end
if nargin<3,maskang=5;end
if nargin<2,error('insufficient number of input arguments');end
if isempty(orgxyz),usrece=usrxyz;else,usrece=enu2xyz(usrxyz,orgxyz);end
if isempty(orgxyz),enuflg=0;else,enuflg=1;end
i = 0;
numsv = max(size(SVIDV));
svid = 0;
for N = 1:numsv,
    svxyz = svpos(RV(N),TOEV(N),MV(N),OMGV(N),INCLV(N),time);
    svenu = xyz2enu(svxyz,usrece); % User is origin
    el = (180/pi)*atan2(svenu(3),norm(svenu(1:2)));
    if el > maskang,
        i = i + 1;
        if enuflg==0,
            svxyzmat(i,:) = svxyz'; % ECEF position
        else
            svxyzmat(i,:) = (xyz2enu(svxyz,orgxyz))'; % ENU position
        end
        svid(i) = SVIDV(N);
    end,
end,
Comment:
    INPUTS:
        usrxyz(1:3) = user position in cartesian ECEF coordinates if orgxyz is not
        provided.
        time = time of week in seconds.
```

maskang = satellite visibility mask angle (in degrees).

orgxyz(1:3) = optional argument. Cartesian ECEF coordinates of locally-level ENU coordinate origin.

OUTPUTS:

svxyzmat = matrix of visible satellite positions in cartesian coordinates.

svxyzmat(i,1:3) = x,y,z coordinates for satellite i.

svid = vector of identification numbers for the visible satellites corresponding to svxyzmat.

第7章 电离层延迟和对流层延迟

7.1 电离层延迟

电离层是高度在60-1000千米间的大气层。在太阳紫外线、X射线、γ射线和高能粒子等的作用下，电离层中的中性气体分子部分被电离，产生了大量的电子和正离子，从而形成了一个电离区域。电磁波信号（此处是卫星信号）在穿过电离层时，其传播速度会发生变化，变化程度主要取决于电离层中的电子密度和信号频率；其传播路径也会略微弯曲（但对测距结果产生的影响不大，一般情况下不予考虑），从而使得信号的传播时间 $\Delta t'$ 乘以真空中的光速后所得到的距离不等于从信号源至接收机的几何距离。对GPS测量来讲，这种误差在天顶方向可达十几米，在高度角为5°时可超过50米，因而必须仔细加以改正。

当用单频接收机时，在精度要求不高的情况下，常用的模型有本特模型，国际参考电离层模型和克罗布歇模型；当要求较高的精度时，可以利用GNSS双频资料建立VETC模型。当用双频接收机时，可以用双频改正模型。在GPS现代化实施后，及随后的各类卫星中将增设第三个民用频率信号，用户将能同时用三个频率的信号来进行测距。

7.1.1 电离层模型

（1）Klobucharm模型（克罗布歇模型）

此模型是针对单频接收机而言的，其精度不高。

1) 改正公式

这是一个被单频GPS接收机用户所广为采用的电离层延迟改正模型。该模型将晚间的电离层时延视为常数，取值为5ns，把白天的时延看成是余弦函数中正的部分。于是天顶方向调制在L1载波上的测距码的电离层时延 T_g 可表示为

$$T_g = 5 \times 10^{-9} + A \cos \frac{2\pi}{P} (t - 14^h) \quad (7-1)$$

振幅 A 和周期 P 分别为：

$$\begin{aligned} A &= \sum_{i=0}^3 \alpha_i (\varphi_m)^i \\ P &= \sum_{i=0}^3 \beta_i (\varphi_m)^i \end{aligned} \quad (7-2)$$

全球定位系统向单频接收机用户提供的电离层延迟改正时就采用上述模型。其中 α_i 和 β_i 是地面控制系统根据该天为一年中的第几天（将一年分为37个区间）以及前5天太阳的平均辐射流量（共分10档）从370组常数中选取的，然后编入

GPS卫星导航电文播发给用户。

2) 计算方法

下面介绍一下式(7-1)中的参数 t 和式(7-2)中参数 φ_m 的准确含义及计算方法。在介绍之前有必要引入中心电离层的概念。我们知道,电离层分布在离地面60-1000km的区域内。当卫星不在测站的天顶时,信号传播路径上每点的经度和纬度均不相同,于是我们就需要对每个微分线段 ds 分别进行计算,然后积分求得总的电离层延迟量。但这么做会使计算变得十分复杂,所以在计算时我们通常采用下列方法:将整个电离层压缩为一个单层,将整个电离层中的自由电子都集中在该单层上,用它来代替整个电离层。这个电离层就称为中心电离层。中心电离层离地面的高度通常取350km。式中的参数 t 和式中的参数 φ_m 分别为卫星信号传播路径与中心电离层的交点 P' 的时角和地磁纬度,因为只有 P' 才能反映卫星信号所受到的电离层延迟的总的情况,某点的地磁纬度为过该点的法线与地磁场的赤道平面之间的交角。 t 和 φ_m 的计算方法如下:

①计算测站 P 和交点 P' 在地心的夹角 EA :

$$EA = \left(\frac{445^\circ}{el + 20^\circ} \right) - 4^\circ \quad (7-3)$$

这是一个近似公式,精度已能满足要求。式中 el 为卫星在测站 P 处的高度角。

②计算交点 P' 的地心经度 $\varphi_{p'}$ 和纬度 $\lambda_{p'}$:

$$\begin{cases} \varphi_{p'} = \varphi_p + EA \cdot \cos \alpha \\ \lambda_{p'} = \lambda_p + EA \cdot \frac{\sin \alpha}{\cos \varphi_p} \end{cases} \quad (7-4)$$

式中: φ_p , λ_p 为测站的地心经、纬度; α 为卫星方位角。 $\lambda_{p'}$ 的单位为度。

③计算观测瞬间交点 P' 处的地方时 t 。

若观测时刻的世界时为UT,则有:

$$t = UT + \frac{\lambda_{p'}}{15} \quad (7-5)$$

t 的单位为小时。

④计算 P' 的地磁纬度 φ_m 。

由于地球的磁北极位于

$$\begin{aligned} \varphi &= 78.4^\circ N \\ \lambda &= 291.0^\circ E \end{aligned}$$

因此 P' 处的地磁纬度可用下式计算:

$$\varphi_m = \varphi_{p'} + 11.6^\circ \cos(\varphi_{p'} - 291.0^\circ) \quad (7-6)$$

利用式(7-3)至式(7-6)以及卫星导航电文给出的 α_i 和 β_i 就能求出观测时刻天顶方向的电离层延迟 T_g 。根据每颗卫星的天顶距 Z 就能计算出该时刻每颗卫星信号电离层时延 T_g' ：

$$T_g' = T_g \cdot \sec Z \quad (7-7)$$

注意， Z 不是卫星在测站 P 处的天顶距，而是交点 P' 处的天顶距。计算 $\sec Z$ 的近似公式为：

$$\sec Z = 1 + 2 \left(\frac{90^\circ - el}{90^\circ} \right)^3 \quad (7-8)$$

除了此电离层模型外，还有本特模型、国际参考电离层模型等。这几种电离层延迟模型都是反映长时期内全球平均状况的经验模型。利用这些模型来估计某一时刻某一地点电离层延迟模型的精度均不够理想，其误差为实际延迟量的20%-40%。使用上述模型的优点是单频用户无需其他支持系统即可获得近似的电离层延迟改正数。

3) 算法过程

已知大地经度、大地纬度、卫星的高度角和卫星测站的方位角。

①计算测站 P 和 P' 在地心的夹角

$$\Psi = \frac{0.137}{E + 0.11} \theta \quad (7-9)$$

②计算交点 P' 的地心纬度

$$\Phi_l = \Phi_u + \Psi \cos A \quad (7-10)$$

$$\text{其中} \begin{cases} \Phi_l > +0.416 & \Phi_l = +0.416 \\ \Phi_l < -0.416 & \Phi_l = -0.416 \end{cases}$$

③计算交点 P' 的地心经度

$$\lambda_l = \lambda_u + \frac{\Psi \sin A}{\cos \Phi_l} \quad (7-11)$$

④计算地磁纬度

$$\Phi_m = \Phi_l - 0.64 \cos \Phi_l \quad (7-12)$$

⑤计算观测瞬间交点 P' 的地方时

$$t = 4.32 \times 10^4 + GPST \text{ (秒)} \quad (7-13)$$

$$\text{其中} \begin{cases} t > 86400 & t = t - 86400 \\ t < 0 & t = t + 86400 \end{cases}$$

⑥计算倾斜因子

$$F = 1.0 - 1.6 \times 10^{-5} f^2 \quad (7-14)$$

⑦计算电离层时间延迟

$$T_{iono} = F \times \left[5 \times 10^{-9} + \sum_{n=0}^3 \alpha_n \phi_m^n \times \left(1 - \frac{x^2}{2} + \frac{x^4}{24} \right) \right] \quad (7-15)$$

其中

$$x = \frac{2\pi(t - 50400)}{\sum_{n=0}^3 \beta_n \phi_m^n}$$

(2) 双频改正模型

由GPS测量的知识可知

$$\begin{aligned} P_1 &= \rho^* + I - e_1 \\ \Phi_1 &= \rho^* - I + \lambda_1 N_1 - \varepsilon_1 \\ P_2 &= \rho^* + (f_1/f_2)^2 I - e_2 \\ \Phi_2 &= \rho^* - (f_1/f_2)^2 I + \lambda_2 N_2 - \varepsilon_2 \end{aligned} \quad (7-16)$$

由上式的第4式减第2式，可得电离层延迟

$$I = \frac{(\Phi_2 - \lambda_2 N_2) - (\Phi_1 - \lambda_1 N_1)}{1 - (f_1/f_2)^2} \quad (7-17)$$

在不考虑模糊度的情况下，可得

$$I = \frac{\Phi_2 - \Phi_1}{1 - (f_1/f_2)^2} \quad (7-18)$$

7.1.2 程序用到的有关函数

(1) strmatch 函数

功能：搜索指定串

两种形式：

```
x = strmatch('str', STRS)
x = strmatch('str', STRS, 'exact')
```

描述：

`x = strmatch('str', STRS)`搜索STRS中以'str'开头的字符串，返回值为以'str'开头的字符串所在的行号。

`x = strmatch('str', STRS, 'exact')`搜索STRS中与'str'完全匹配的字符串，返

回值为字符串'str'在STRS中的行号。

例：

```
x = strmatch('max', strvcat('max', 'minimax', 'maximum'))
x=1
3
x = strmatch('max', strvcat('max', 'max', 'maximum'),'exact')
x=1
2
```

(2) find 函数

功能：用于查询非零元素的行和列标志

`ind = find(X)` 查询非零元素的位置，如果X是一个行向量，则返回一个行向量，否则，返回一个列向量。如果X全是零元素或者是空数组，则返回一个空数组。

`[row,col V] = find(X, ...)` 查询满足一定条件的元素的行和列

X一般为逻辑表达式，如 $X > 5$ 等。

其中 row 返回满足条件的元素行的位置；

col 返回满足条件的元素的列的位置；

V 如果X是一个逻辑表达式，则返回一个逻辑数组，一般如果存在满足要求的元素，则返回一个单位列向量。如果不存在满足要求的元素，则返回一个空向量。

例

```
X = [1 0 4 -3 0 0 0 8 6];
indices = find(X)
indices =
    1     3     4     8     9
```

```
X = [3 2 0; -5 0 7; 0 0 1];
[i,j,v] = find(X)
i =
    1
    2
    1
    2
    3
```



```

j =
     1
     1
     2
     3
     3

v =
     3
    -5
     2
     7
     1

```

```

x = [11  0  33  0  55]';
find(x)
ans =
     1
     3
     5

find(x == 0)
ans =
     2
     4

find(0 < x & x < 10*pi)
ans =
     1

```

find 函数的其他用法可以参考 MATLAB 中的帮助文件。

(3) 非数 NaN

按 IEEE 规定, $\frac{0}{0}$, $\frac{\infty}{\infty}$, $0 \times \infty$ 等运算都会产生非数 (Not a Number)。该数在 MATLAB 中用 NaN 或 nan 描述。

根据 IEEE 数学规范, NaN 具有以下性质:

- 1) NaN 参与运算所得的结果也是 NaN, 具有传递性;
- 2) 非数没有“大小”概念, 因此不能比较两个非数的大小。

非数的功用:

- 1) 真实记述 $\frac{0}{0}$, $\frac{\infty}{\infty}$, $0 \times \infty$ 运算的后果
- 2) 避免可能因 $\frac{0}{0}$, $\frac{\infty}{\infty}$, $0 \times \infty$ 运算而造成程序执行的中断;
- 3) 在数据可视化中, 用来裁剪图形。

(4) isnan(A) 函数

功能: 返回一个与A相同维数的数组, 若A的元素为NaN (非数值), 在对应位置上返回逻辑1 (真), 否则返回逻辑0 (假)。对虚数z, 如果z的实部或虚部是NaN, 那么isnan(z)函数返回逻辑1, 如果实部和虚部都是inf或者有限值, 则返回逻辑0。

例:

```
a=[-2 -1 0 1 2]
```

```
isnan(1./a)
```

```
warning:Divide by zero
```

```
ans=
```

```
0 0 0 0 0
```

```
isnan(0./a)
```

```
warning: Divide by zero
```

```
ans=
```

```
0 0 1 0 0
```

以上函数的详细用法可参考MATLAB中的帮助文件。

7.1.3 程序说明

在根据载波相位观测的原理进行设计算法时, 需要5个已知值, 光速C, f1和f2的频率已经与之对应的波长。下面就要对文件进行操作, 打开文件后, 首先要读取o文件的头文件, 我们对此感兴趣的只是其中的观测类型, 一般观测文件中有7种类型:

L1: L1载波上的相位观测值

L2: L2载波上的相位观测值

C1: 采用L1上C/A码所测的伪距;

P1: 采用L1上的P码所测的伪距;

P2: 采用L2上的P码所测的伪距;

D1: L1上的多普勒频率;

D2: L2上的多普勒频率;

在此调用了anheader函数 (此函数在前面已经应用过), 将上述七种观测值付给变量Obs_types。根据以上的原理, 我们提取头文件中的P1、P2和L1、L2 (具

体提取过程见第一个循环)。程序中的match就是L1、L2和P1、P2所在文件中的具体位置。

第二个for循环是针对每个历元的，在此首先调用了fepoch函数（前面也用到过），在这里需要每一个历元所观测到的卫星数目以及卫星的Prn号，程序中的NoSv代表每个利用观测到的卫星总数，然后grabdata函数提取四种类型的观测值，将四种观测值赋给变量obs，这个循环中的嵌套循环就是针对各个观测历元的卫星计算其相位差，将其结果保存与变量phase_diff中。

经过上面的过程，其相位差已计算出来。我们假定第一个观测历元无延迟，后面的观测历元都以第一个历元的相位差为基础，最后利用上面的式（7-18）计算每颗卫星的延迟量。

7.1.4 程序完整过程

```
clc;
clear;
v_light = 299792458;           % 光速
f1 = 154*10.23E6;             % L1的频率
f2 = 120*10.23E6;             % L2 的频率
lambda1 = v_light/f1;          % L1的载波波长
lambda2 = v_light/f2;          % L2的载波波长
ofile = 'kofi1.01o';           % 观测文件
fid = fopen(ofile,'rt');        % 打开观测文件
[Obs_types, ant_delta, ifound_types, eof1] = anheader(ofile);           %
                                     读取o文件中的观测类型等（本程序主要用到）
NoObs_types = size(Obs_types,2)/2;
obsstr(1,1:2) = 'P1'; % P1
obsstr(2,1:2) = 'P2'; % P2
obsstr(3,1:2) = 'L1'; % Phi1
obsstr(4,1:2) = 'L2'; % Phi2
match = zeros(1,4);
****下面的for循环式提取需要的观测类型*****
for t = 1:4
    for ii = 1:NoObs_types
        mat = strmatch(obsstr(t,1:2),Obs_types(1,2*ii-1:2*ii),'exact');
        if isempty(mat) == 0, match(1,t) = ii; end
    end
end
```

```

end
Oc = match;%观测类型所在的列。即P1为第四个数，P2为第5个。L1为第1个，L2
    为第2个
qend = 300;%总的观测历元数
phase_diff = zeros(32,qend)*inf;%32代表有32颗卫星
*****核心部分*****
for q = 1:qend
    [time, dt, sats, eof] = fepoch_0(fid);%本程序中用的fepoch_0函数用于提取每
    个历元所观测的卫星个数
    NoSv = size(sats,1);%每个历元所观测到的卫星总数
    obsm = grabdata(fid, NoSv, NoObs_types);
    obs = obsm(:,Oc); % 观测类型P1 P2 Phi1 Phi2
    for t = 1:NoSv
        Phi1 = obs(t,3)*lambda1; %L1载波的相位值
        Phi2 = obs(t,4)*lambda2; %L2载波的相位值
        sat = sats(t); %观测到的卫星的Prn号
        phase_diff(sat,q) = Phi1-Phi2;
    end
end;
pd = phase_diff-phase_diff(:,1)*ones(1,qend);%假定首次观测无电离层延迟，后面
299个历元的电离层延迟是相对于首次观测的延迟
*****删掉有NaN的行，用不同颜色区分不同卫星的对流层延迟*****
prn = find(~isnan(pd(:,1)));
pd(isnan(pd(:,1:qend))) = [];
pd = reshape(pd,length(prn),qend);
pd = pd/(1-(f1/f2)^2);
*****画出有关卫星的电离层延迟*****
figure(1);
plot(1000*pd,'linewidth',2)
legend(eval('num2str(prn)'),2)
title('Ionospheric Delay From {\itL}_1 and {\itL}_2 Phase','fontsize',16)
ylabel('Ionospheric Delay [mm]','fontsize',16)
xlabel('Epochs [1 s interval]','fontsize',16)
set(gca,'fontsize',16)

```

7.2 对流层延迟

卫星导航定位中的对流层延迟通常是泛指电磁波信号在通过高度在50千米以下的未被电离的中性大气层时所产生的信号延迟。在研究信号延迟的过程中，我们不再将该大气层细分为对应层和平流层，也不再顾及两者之间性质上的差别。由于80%的延迟发生在对流层，所以我们将发生在该中性大气层中的信号延迟统称为对流层延迟。对流层中的大气成分比较复杂，主要有氮和氧组成，此外还含有少量的水蒸气及氩、二氧化碳、氢气等其气体。大气中还含有某些不定量的混合物，如硫化物、煤烟和粉尘等。

普通GPS测量中常用的对流层延迟模型有：Hopfield模型、Modified Hopfield模型、Saastamoinen模型、simplified Saastamoinen模型和Black模型。为了提高这几中模型的精度，专家和学者们提出了不少对流层延迟改正中的投影函数模型：如Marini模型、Chao模型、NMF模型、VMF1模型等。下面详细介绍Modified Hopfield折射改正模型

7.2.1 延迟模型

Modified hopfield折射改正模式，对单程距离的改正式如下：

$$\Delta d_r = 10^3 [N_1 \sum_{j=1}^9 \alpha_{j1} \frac{R_1^j}{j} + N_2 \sum_{j=1}^9 \alpha_{j2} \frac{R_2^j}{j}]$$

式中： Δd_r 为距离改正量 （m）

$$R_i = \sqrt{(a_e + h_i)^2 - (a_e + h_k)2 \cos 2(E) - (a_e + h_k) \sin(E)}, \quad i = 1, 2$$

$$i = \begin{cases} 1 & \text{干分量} \\ 2 & \text{湿分量} \end{cases}$$

a_e ：地球椭球长半径=6378.137 km

E ：卫星高度角

$$\alpha_{1i} = 1 \quad \alpha_{2i} = 4a_i \quad \alpha_{3i} = 6a_i^2 + 4b_i \quad \alpha_{4i} = 4a_i(a_i^2 + 3b_i) \quad \alpha_{5i} = a_i^4 + 12a_i^2 b_i + 6b_i^2$$

$$\alpha_{6i} = 4a_i b_i(a_i^2 + 3b_i) \quad \alpha_{7i} = b_i(6a_i^2 + 4b_i) \quad \alpha_{8i} = 4a_i b_i^3 \quad \alpha_{9i} = b_i^4$$

$$a_i = -\frac{\sin(E)}{h_i - h_k} \quad b_i = -\frac{\cos^2(E)}{2a_e(h_i - h_k)}$$

$$N_1 = r_{sea} \left[\frac{h_d - h_k}{h_d} \right]^4 \quad N1: \text{对流层干分量折射系数}$$

$$N_2 = \frac{\left[\frac{0.371900}{T_{sea}} - 12.92 \times 10^{-6} \right]}{T_{sea}} e_s \left[\frac{h_w - h_k}{h_w} \right]^4 \quad \text{N2: 对流层湿分量折射系数}$$

干对流层高度: $h_1 = (1.1385 \times 10^{-5}) T_{ea}$

湿对流层高度: $h_2 = 1.1385 \times 10^{-5} [(1255/T_{sea} + 0.05)/r_{sea}]$

海水面上的干分量折射系数: $r_{sea} = (77.624 \times 10^{-6}/T_{sea}) P_{sea}$

海水面上的大气压力: $P_{sea} = P(T_{sea}/T_p)^{e_m}$

海水面上的水蒸气压力: $e_s = e_0(T_{sea}/T_e)^{4e_m}$

在高度 h_p 上量测气压时的温度: $T_p = T - 6.5(h_p - h_t)$

在高度 h_h 上量测湿度时的温度: $T_e = T - 6.5(h_h - h_t)$

T : 在高度 h_t 的绝对温度 (K)

h_p : 量测大气压力的大地高 (km)

h_t : 量测大气温度的大地高 (km)

h_h : 量测大气湿度的大地高 (km)

在高度 h_h 上的水蒸气压力可有相对湿度 R_h (%) 计算而得

$$e_0 = \frac{R_h}{100} \times 6.11 \times 10^{\frac{7.5(T_0 - 273.15)}{237.3 + T_0 - 273.15}}$$

R_h : 在高度 h_h 的相对湿度 (%)

e_0 : 测站的水蒸气压力 (mbar)

由大气平均重力值 $g_m = 978.77(\text{cm}/\text{s}^2)$, 干分量大气常数 $C_d = 2.8704 \times 10^6(\text{cm}^2/\text{s}^2 \times \text{K}^{-1})$, 大气温度梯度 $b = 6.5 \times 10^{-5} \text{K}/\text{m}$, 求得 $e_m = -978.77 / (2.8704 \times 10^6 \times (-6.5 \times 10^{-5}))$

对流层延迟的计算比较容易, 但牵涉到的公式比较多, 需要谨慎, 以防出错。

7.2.2 程序

%sinel 卫星高度角的正弦值

%hsta 测站高 (单位: km)

%p 在高度hp处的大气压 (单位: mbar)

%tkel 高度htkel的地表温度 (单位: 开尔文)

%hum 高度为hhum处的湿度 (以百分数表示)

%hp 测量气压时的高程值 (单位: km)

%htkel 测量温度时的高程值 (单位: km)

%hhum 测量湿度时的高程值 (单位: km)

```

clc
clear

hsta=0;
p=1013.25;
hp=0;
tkel=293.16;
hum=50;
htkel=0;
hhum=0;
el=linspace(10,90,81);
    dddr=[];
for j=1:81
    sinel(j)=sin(deg2rad(el(j)));

a_e = 6378.137;      % 地球的长半轴长度
b0 = 7.839257e-5;% 扁率
tlapse = -6.5; % 温度递减率
tkhum = tkel+tlapse*(hhum-htkel); % 测量湿度时的温度
atkel = 7.5*(tkhum-273.15)/(237.3+tkhum-273.15);%湿度计算
e0 = 0.0611*hum*10^atkel; %湿度计算
tksea = tkel-tlapse*htkel; %海水面温度
em = -978.77/(2.8704e6*tlapse*1.0e-5);
tkelh = tksea+tlapse*hhum; %从海水面算起的温度
e0sea = e0*(tksea/tkelh)^(4*em); %海水面上的水蒸气压力
tkelp = tksea+tlapse*hp;%测量气压时是温度
psea = p*(tksea/tkelp)^em; %海水面上的大气压力
if sinel(j) < 0
    sinel(j) = 0;
end;
tropo = 0;
done = 'FALSE';
refsea = 77.624e-6/tksea; %干折射分量的前半部分
htop = 1.1385e-5/refsea;%干对流层高度

```

```

refsea = refsea*psea; %%海面上干分量折射系数
ref = refsea*((htop-hsta)/htop)^4; %干分量折射率
while 1
    rtop = (a_e+htop)^2-(a_e+hsta)^2*(1-sinel(j)^2);%公式中Ri的前半部分
    if rtop < 0, rtop = 0; end; % check to see if geometry is crazy
    rtop = sqrt(rtop)-(a_e+hsta)*sinel(j);%公式中的Ri
    a = -sinel/(htop-hsta); %公式中的a
    b = -b0*(1-sinel^2)/(htop-hsta); %公式中的b
    rn = zeros(8,1);
    for i = 1:8
        rn(i) = rtop^(i+1);
    end;
    alpha = [2*a, 2*a^2+4*b/3, a*(a^2+3*b),...
             a^4/5+2.4*a^2*b+1.2*b^2, 2*a*b*(a^2+3*b)/3,...
             b^2*(6*a^2+4*b)*1.428571e-1, 0, 0];
    if b^2 > 1.0e-35, alpha(7) = a*b^3/2; alpha(8) = b^4/9; end;%
    dr = rtop;
    dr = dr+alpha*rn;
    tropo = tropo+dr*ref*1000;%干延迟与湿延迟的和
    if done == 'TRUE ', ddr = tropo; break; end;
    done = 'TRUE ';
    refsea = (371900.0e-6/tksea-12.92e-6)/tksea;
    htop = 1.1385e-5*(1255/tksea+0.05)/refsea;%湿对流层高度
    ref = refsea*e0sea*((htop-hsta)/htop)^4;%对流层湿分量折射系数
end;
dddr=[dddr ddr];
end
x=10:1:90;
y=dddr;
figure(1)
plot(x,y)
xlabel('高度角(°)');
ylabel('延迟量(m)');
补充说明：上面程序中的hsta、P、tkel、hum、hp、htkel、hhum的值分别为

```


0、1013.25、293.16、50、0、0、0。在用对流层公式计算延迟时，用户都需要携带气象仪器来实际测定气象元素，这给在一个局部区域中从事精度要求不高的一般的GPS用户造成诸多不便。采用标准气象元素法时减少了其工作量。在标准气象元素中，通常规定，温度 $t=20^{\circ}$ （即293.16K）， $P=1013.25\text{mbar}$ ， $RH=50\%$ 。

第8章 整周模糊度解算

8.1 整周模糊度

利用载波相位测量值进行精密相对定位的根本问题是求解测量值，尤其是双差测量值的整周模糊度。

不同的整周模糊度求解算法会有着不同的求解思路，利用伪距帮助估算整周模糊度的取整方法是最简单、最直接的一种，而其他大量的算法则集中在求解一个整数型最小二乘问题。基于求解整数型最小二乘问题的整周模糊度求解算法一般是将某个目标函数最小化，比如目前应用最多的是使模糊度残余平方和最小。由于整数型最小二乘问题不存在解析解，因而一类整周模糊度求解算法是依据某种准则通过搜索的方式来确定模糊度的，而在那些基于整周模糊度残余平方和最小原则的整数型最小二乘估算法中，较为著名的有最小二乘模糊度搜索算法（LSAST）、优化Cholesky分解算法、LAMBDA算法、快速模糊度搜索滤波法（FASF）、快速模糊度解算法（FARA）、OMEGA算法、零空间方法等。

整合模糊度的求解一般分两步完成：首先是通过一定算法求解出整周模糊度，然后再验证所解得的整周模糊度的正确性。模糊度残余值的大小通常被用来验证整周模糊度求解值的正误，而有的验证方法需要首先求解出最优和次优两个整周模糊度解。

将基于双差载波相位的相对定位方程统一写成如下的一个线性矩阵形式：

$$y = A(\Delta b_{ur}) + Ba \quad (8-1)$$

其中， y 为接收机给出的双差载波相位测量值向量， Δb_{ur} 可以理解成未知的基线向量或者基线向量校正量， a 为需要被求解的双差整周模糊度向量， A 和 B 为常系数矩阵，而其他一些未知的测量误差和噪声均被忽略。一大类整周模糊度求解算法是基于最小二乘原理，它们的最优解 $(\Delta b_{ur}, a)$ 能使测量残余的平方和最小，即

$$\min_{\Delta b_{ur}, a} \|y - A(\Delta b_{ur}) - Ba\|^2 = \min_{\Delta b_{ur}, a} (y - A(\Delta b_{ur}) - Ba)^T (y - A(\Delta b_{ur}) - Ba) \quad (8-2)$$

或者更为普遍地说，使测量残余的加权平方和最小，即

$$\min_{\Delta b_{ur}, a} \|y - A(\Delta b_{ur}) - Ba\|_C^2 = \min_{\Delta b_{ur}, a} (y - A(\Delta b_{ur}) - Ba)^T C (y - A(\Delta b_{ur}) - Ba) \quad (8-3)$$

其中， C 通常取值为测量值 y 的误差协方差矩阵 Q_y 之逆，其中 Q_y 对称、正定。以最小二乘作为求解准则所获得的整周模糊度结果具有最高的正确率，这使得进行这种无偏估算的LAMBDA算法具有很大的吸引力。

若 a 的值不必为一列整数，则最小二乘法可直接用来求解式（8-1），所得的

浮点数值解 $(\Delta b_{ur}, a)$ 能满足式(8-3)。尽管浮点解不是我们最后需要的整数解，但它经取整后的值通常可作为整数解的初始估计值。因为这里整周模糊度 a 要求为整数，所以对式(8-1)的最小二乘解属于整数型最小二乘问题，而它的解一般来说没有解析形式。求解整数型二乘问题的一类方法是搜索，即首先确定一个整周模糊度向量的搜索空间并设置一个目标函数，然后计算在搜索空间内各个整数值格网点上的目标函数值，最后将那个使目标函数值达到最小的整数值格网点作为整周模糊度的最优解。可见，搜索空间需要足够大，使它必定包含整周模糊度的正确解，而它又必须尽可能地小，以减少搜索计算量。假设搜索空间为一个 $M-1$ 维的正方体，并在每一维上搜索浮点解附近左右的 L 个整数，那么我们总共需要搜索 $(M-1)^L$ 个点，而这在 M 和 L 较大时意味着很大的计算量。

LAMBDA设定了如式(8-3)所示的目标函数，并通过以下的浮点解、整周模糊度估算和整数解三步来完成对该目标函数的最小化。

第一步：

不考虑整周模糊度 a 的整数要求而直接求解出满足(8-3)的浮点型加权最小二乘解 $\hat{\Delta b}_{ur}$ 和 \hat{a} 。这一步的浮点解已由最小二乘方法给出，且又给出了这一解的协方差矩阵 $Q_{[\hat{\Delta b}_{ur}, \hat{a}]}$ ，其中 $[\hat{\Delta b}_{ur}, \hat{a}]$ 代表由竖向量 $\hat{\Delta b}_{ur}$ 和 \hat{a} 先后排在一起组成的竖向量。我们将协方差矩阵 $Q_{[\hat{\Delta b}_{ur}, \hat{a}]}$ 分解成如下形式：

$$Q_{[\hat{\Delta b}_{ur}, \hat{a}]} = Cov \begin{bmatrix} \hat{\Delta b}_{ur} \\ \hat{a} \end{bmatrix} = \begin{bmatrix} Q_{\hat{\Delta b}_{ur}} & Q_{\hat{\Delta b}_{ur}, \hat{a}} \\ Q_{\hat{\Delta b}_{ur}, \hat{a}}^T & Q_{\hat{a}} \end{bmatrix} \quad (8-4)$$

其中， $Q_{\hat{\Delta b}_{ur}}$ 为 $\hat{\Delta b}_{ur}$ 的协方差矩阵， $Q_{\hat{a}}$ 为 \hat{a} 的协方差矩阵，而 $Q_{\hat{\Delta b}_{ur}, \hat{a}}$ 为 $Q_{[\hat{\Delta b}_{ur}, \hat{a}]}$ 的右上角部分，它代表 $\hat{\Delta b}_{ur}$ 和 \hat{a} 之间相关性。

第二步：

以整数向量 a 与浮点数 \hat{a} 之间的距离平方为目标函数，搜索整周模糊度 a ，使这一目标函数达到最小值，即

$$\min_a \|a - \hat{a}\|_{Q_{\hat{a}}^{-1}}^2 \quad (8-5)$$

而我们将满足上式的整数解记为最优解 \hat{a} 。第一步与这一步实际上也将式(8-3)分成两部分进行最小化。

对于如式(8-5)所示的最小二乘问题，若 $Q_{\hat{a}}^{-1}$ 为一对角阵，则最优整数解 \hat{a} 相当明显，它直接等于 \hat{a} 的四舍五入取整值。然而，我们在前面已经指出， $Q_{\hat{a}}^{-1}$ 通常并不是一个对角阵，这种不同整周模糊度值直之间的相关性不再使浮点解 \hat{a} 的取整值为最优解 \hat{a} ，于是最优解 \hat{a} 需要搜索才被找到。LAMBDA算法规定了如下

一个关于整周模糊度 \mathbf{a} 的整数解的搜索空间：

$$\|\mathbf{a} - \hat{\mathbf{a}}\|_{\mathbf{Q}_a^{-1}}^2 < T \quad (8-6)$$

其中 T 是一个取整适当的门限。由上式所限定的搜索空间是一个多维椭球体，球体内部的（8-5）。

然而，具有实际意义的权系数矩阵 \mathbf{Q}_a^{-1} 对不同测量值有着不等的权重。当 \mathbf{Q}_a^{-1} 对不同测量值之间的权重相差太大时，以上椭球搜索空间就会变得相当狭长，这会使最优整数解 $\hat{\mathbf{a}}$ 看上去并不一定在浮点解 $\hat{\mathbf{a}}$ 附近，有可能离 $\hat{\mathbf{a}}$ 很远，而面对这种困境，遍历搜索或许成了唯一的手段，为了让最优整数解 $\hat{\mathbf{a}}$ 出现在浮点解 $\hat{\mathbf{a}}$ 附近，从而相应地限制搜索在 $\hat{\mathbf{a}}$ 附近进行，以提高搜索效率，LAMBDA算法通过以下的 \mathbf{Z} 变换将原先在一个狭长椭球体内对 \mathbf{a} 的搜索变成一个近似球体空间内对 \mathbf{z} 的搜索

$$\mathbf{z} - \hat{\mathbf{z}} = \mathbf{Z}(\mathbf{a} - \hat{\mathbf{a}}) \quad (8-7)$$

相应的，式（8-5）就等价地变换成

$$\min_{\mathbf{a}} \|\mathbf{a} - \hat{\mathbf{a}}\|_{\mathbf{Q}_a^{-1}}^2 = \min_{\mathbf{z}} \|\mathbf{z} - \hat{\mathbf{z}}\|_{\mathbf{Z}^T \mathbf{Q}_a^{-1} \mathbf{Z}}^2 \quad (8-8)$$

其中，权系数阵从原先的 \mathbf{Q}_a^{-1} 变为对角阵 $\mathbf{Z}^T \mathbf{Q}_a^{-1} \mathbf{Z}$ 。在完成了用来降低整周模糊度之间相关性的 \mathbf{Z} 变换后，LAMBDA算法才进行实质性的整周模糊度搜索求解。若 $\mathbf{Z}^T \mathbf{Q}_a^{-1} \mathbf{Z}$ 的确为一个对角阵，则对式（8-8）的求解变得相当容易，它的最优整数解 $\hat{\mathbf{z}}$ 直接等于向量 $\hat{\mathbf{z}}$ （即 $\mathbf{Z}\hat{\mathbf{a}}$ ）的四舍五入取整值，而我们接着可将最优解 $\hat{\mathbf{z}}$ 反变换成最优整数解 $\hat{\mathbf{a}}$ ，即

$$\hat{\mathbf{a}} = \mathbf{Z}^{-1} \hat{\mathbf{z}} \quad (8-9)$$

我们从以上的求解过程中可以看出，变换矩阵 \mathbf{Z} 及其逆变换矩阵 \mathbf{Z}^{-1} 的元素均应该为整数，而这不但保证了该变化是一种一一映射关系，而且还说明矩阵 \mathbf{Z} 和 \mathbf{Z}^{-1} 的行列式值均为1，即变化前后搜索空间的体积保持不变。可是，具有这些特性的变换 \mathbf{Z} 很难将 \mathbf{Q}_a^{-1} 对角化，而事实上 $\mathbf{Z}^T \mathbf{Q}_a^{-1} \mathbf{Z}$ 只是一个近似的对角阵。LAMBDA算法需要通过对 \mathbf{Q}_a^{-1} 进行一系列的整数变换才能完成近似的对角化，然后采用基于 \mathbf{LDL}^T 分解的序贯条件最小二乘的方法进行搜索，整个计算甚为复杂。简而言之，第二步是LAMBDA算法的关键，它的输入量基本上仅是 $\hat{\mathbf{a}}$ 和 \mathbf{Q}_a ，而输出量是 $\hat{\mathbf{a}}$ 。

第三步：

将整周模糊度最优整数解 $\hat{\mathbf{a}}$ 代入式（8-1），从而求解出基线向量 $\Delta \mathbf{b}_{ur}$ 的最优“整数”解 $\hat{\Delta \mathbf{b}}_{ur}$

8.2 Z 变换

Z变换的方法很多，下面介绍一种算法。

对整周模糊度的方差协方差阵进行 $L^T D L$ 分解，下三角矩阵L和对角阵D写成分块矩阵

$$L = \begin{bmatrix} L_{11} & & \\ L_{21} & L_{22} & \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \text{ 和 } D = \begin{bmatrix} D_{11} & & \\ & D_{22} & \\ & & D_{33} \end{bmatrix}$$

子矩阵 L_{22} 是二阶的， L_{11} 是 $i-1$ 阶的， L_{33} 是 $n-i-1$ 阶的。

我们对第 i 个和第 $(i+1)$ 个整周模糊度进行二维的整数变换，利用如下的分块矩阵

$$Z = \begin{bmatrix} I_{i-1} & & \\ & Z_{22} & \\ & & I_{n-i-1} \end{bmatrix}$$

Z_{22} 是一个2阶满秩矩阵，我们可以得到新的三角分解 $L'' D' L'$ ，其中

$$L = \begin{bmatrix} L_{11} & & \\ \bar{L}_{21} & \bar{L}_{22} & \\ L_{31} & \bar{L}_{32} & L_{33} \end{bmatrix} \text{ 和 } D = \begin{bmatrix} D_{11} & & \\ & \bar{D}_{22} & \\ & & D_{33} \end{bmatrix}$$

只有 L_{21} 、 L_{22} 、 L_{32} 和 D_{22} 发生了改变，对 $Q_{\hat{N}}$ 进行整数变换，可以得到

$$Z^T Q_{\hat{N}} Z = \begin{bmatrix} Q_{11} & & \\ Z_{22}^T Q_{21} & Z_{22}^T Q_{22} Z_{22} & \\ Q_{31} & Q_{32} Z_{22} & Q_{33} \end{bmatrix} = \begin{bmatrix} Q_{11} & & \\ \bar{Q}_{21} & \bar{Q}_{22} & \\ Q_{31} & \bar{Q}_{32} & Q_{33} \end{bmatrix}$$

$$L'' D' L' = Z^T L^T D L Z$$

整数高斯变换

$$Z = \begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix}$$

$$\bar{L}_{32} = \begin{bmatrix} l_{i+2,i} + \alpha l_{i+2,i+1} & l_{i+2,i+1} \\ l_{i+3,i} + \alpha l_{i+3,i+1} & l_{i+3,i+1} \\ \vdots & \vdots \\ l_{n,i} + \alpha l_{n,i+1} & l_{n,i+1} \end{bmatrix}$$

$$\bar{L}_{22} = \begin{bmatrix} 1 & 0 \\ l_{i+1,i+\alpha} & 1 \end{bmatrix}$$

$$\bar{D}_{22} = \begin{bmatrix} d_i & 0 \\ 0 & d_{i+1} \end{bmatrix}$$

$$\bar{L}_{21} = L_{21}$$

我们选择 $\alpha = -l_{i+1,i}$ ， $l_{i+1,i}$ 一般情况下不是整数，为满足要求， α 取 $l_{i+1,i}$ 的最接近整数的负值。我们在 (i, j) 位置进行Z变换，元素的改变如下：

$$l'_{i,j} = l_{i,j} + \alpha$$

$$l'_{k,j} = l_{k,j} + \alpha l_{k,i} \quad (k = i+1, \dots, n)$$

整周模糊度浮点解 \hat{N} 经过转换 $\hat{z} = Z^T \hat{N}$ 。因子L经过变换 $L' = LZ$ ，可以得出 $L^T D L' = Q_{\hat{z}}$ 。

Z矩阵的初始值设为单位阵，经过变换之后为下三角矩阵。

伪代码：

```
for i=n:-1:1
    for j=i+1:n
        u=n*int(L(j,i))
        if u≠0
            L(j:n,i)=L(j:n,i)-u*L(j:n,j)
            Z(1:n,i)=Z(1:n,i)-u*Z(1:n,j)
            a(i)=a(i)-u*a(j)
        end
    end
end
end
```

8.3 重排列

为使变换之后的Q阵尽量不相关，对D进行重排列得到 \bar{D} ，重排列之后的 \bar{D} 满足：

$$\bar{d}_1 \geq \bar{d}_2 \geq \dots \geq \bar{d}_n$$

对 $Q_{\hat{a}}$ 的因式分解 $L^T D L$ 写成分块矩阵形式：

$$Q_{\hat{a}} = L^T D L = \begin{bmatrix} L_{11}^T & L_{21}^T & L_{31}^T \\ & L_{22}^T & L_{32}^T \\ & & L_{33}^T \end{bmatrix} \begin{bmatrix} D_1 & & \\ & D_2 & \\ & & D_3 \end{bmatrix} \begin{bmatrix} L_{11} & & \\ L_{21} & L_{22} & \\ L_{31} & L_{32} & L_{33} \end{bmatrix}$$

做如下定义：

$$P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, P_{k,k+1} = \begin{bmatrix} I_{k-1} & & \\ & P & \\ & & I_{n-k-1} \end{bmatrix}$$

式 $P_{k,k-1}^T Q_{\hat{a}} P_{k,k-1}$ 的 $L^T D L$ 分解式：

$$P_{k,k-1}^T Q_{\hat{a}} P_{k,k-1} = \begin{bmatrix} L_{11}^T & \bar{L}_{21}^T & L_{31}^T \\ & \bar{L}_{22}^T & \bar{L}_{32}^T \\ & & L_{33}^T \end{bmatrix} \begin{bmatrix} D_1 & & \\ & \bar{D}_2 & \\ & & D_3 \end{bmatrix} \begin{bmatrix} L_{11} \\ \bar{L}_{21} & \bar{L}_{22} \\ L_{31} & \bar{L}_{32} & L_{33} \end{bmatrix}$$

式中:

$$\bar{D}_2 = \begin{bmatrix} \bar{d}_k & \\ & \bar{d}_{k+1} \end{bmatrix}, \bar{d}_{k+1} = d_k + l_{k+1,k}^2 d_{k+1}, \bar{d}_k = \frac{d_k}{d_{k+1}} d_{k+1},$$

$$\bar{L}_{22} = \begin{bmatrix} 1 & \\ \bar{l}_{k+1,k} & 1 \end{bmatrix}, \bar{l}_{k+1,k} = \frac{d_{k+1} l_{k+1,k}}{\bar{d}_{k+1}},$$

$$\bar{L}_{21} = \begin{bmatrix} -l_{k+1,k} & 1 \\ \frac{d_k}{d_{k+1}} & \bar{l}_{k+1,k} \end{bmatrix} L_{21} = \begin{bmatrix} -l_{k+1,k} & 1 \\ \frac{d_k}{d_{k+1}} & \bar{l}_{k+1,k} \end{bmatrix} L(k:k+1, 1:k-1),$$

$$\bar{L}_{32} = L_{32} P = [L(k+2:n, k+1) \quad L(k+2:n, 1:k)]$$

伪代码:

i1=n-1;sw=true

while sw=true

 i=n;sw=false

 while sw=false and i>1

 i=i-1

 if i≤i1

 apply algorithm ZTRAN to column i %ZTRAN Z 变换

 end

$\delta = D(i, i) + L(i+1, i)^2 * D(i+1, i+1)$

 if $\delta < D(i+1, i+1)$

$\lambda_3 = D(i+1, i+1) * L(i+1, i) / \delta; \eta = D(i, i) / \delta$

$D(i, i) = \eta * D(i+1, i+1)$

$D(i+1, i+1) = \delta$

 for j=1:i-1

$\lambda_1 = L(i, j); \lambda_2 = L(i+1, j)$

$L(i, j) = -L(i+1, i) * \lambda_1 + \lambda_2$

$L(i+1, j) = \eta * \lambda_1 + \lambda_3 * \lambda_2$

 end

$L(i+1, i) = \lambda_3$

```

swap columns L(i+2:n,i) and L(i+2:n,i+1)
swap columns Z(1:n,i) and L(1:n,i+1)
swap celements a(i) and a(i+1)
i1=i;sw=true
end
end
end

```

8.4 搜索

搜索的目的在满足式 $(\hat{a} - a)^T Q_a^{-1} (\hat{a} - a) \leq \chi^2$ 的基础上获得最小值。

对 Q_a^{-1} 进行 LDL^T 分解后，上式可以写成

$$\sum_{i=1}^n d_i \left[(a_i - \hat{a}_i) + \sum_{j=i+1}^n l_{ji} (a_j - \hat{a}_j) \right]^2 \leq \chi^2$$

通过移项可以得到：

$$\underbrace{\left[(a_{i+1} - \hat{a}_{i+1}) + \sum_{j=i+2}^n l_{j,i+1} (a_j - \hat{a}_j) \right]^2}_{left_{i+1}} \leq \frac{\chi^2}{d_{i+1}} - \frac{1}{d_{i+1}} \sum_{l=i+2}^n d_l \underbrace{\left[(a_l - \hat{a}_l) + \sum_{j=l+1}^n l_{j,l} (a_j - \hat{a}_j) \right]^2}_{right_{l+1}}$$

当第 $i+1$ 个整周模糊度被固定为 α_{i+1} ，我们计算第 i 个整周模糊度的边界：

$$\underbrace{\left[(a_i - \hat{a}_i) + \sum_{j=i+1}^n l_{j,i} (a_j - \hat{a}_j) \right]^2}_{left_i} \leq \frac{\chi^2}{d_i} - \frac{1}{d_i} \sum_{l=i+1}^n d_l \underbrace{\left[(a_l - \hat{a}_l) + \sum_{j=l+1}^n l_{j,l} (a_j - \hat{a}_j) \right]^2}_{right_l}$$

上式可以简化为

$$\underbrace{\left[(a_i - \hat{a}_i) + \sum_{j=i+1}^n l_{j,i} (a_j - \hat{a}_j) \right]^2}_{left_i} \leq \frac{d_{i+1}}{d_i} \underbrace{(right_{i+1} - left_{i+1})}_{right_i}$$

迭代计算的初始值为：

$$\underbrace{[(a_n - \hat{a}_n)]^2}_{left_n} \leq \frac{\chi^2}{\underbrace{d_n}_{right_n}}$$

上面式子可以写成：

$$\left| (a_i - \hat{a}_i) + \sum_{j=i+1}^n l_{j,i} (a_j - \hat{a}_j) \right| \leq \sqrt{right_i}$$

$$\begin{aligned}
-\sqrt{right_i} &\leq (a_i - \hat{a}_i) + \sum_{j=i+1}^n l_{j,i}(a_j - \hat{a}_j) \leq \sqrt{right_i} \\
-\sqrt{right_i} - \sum_{j=i+1}^n l_{j,i}(a_j - \hat{a}_j) &\leq a_i - \hat{a}_i \leq \sqrt{right_i} - \sum_{j=i+1}^n l_{j,i}(a_j - \hat{a}_j) \\
\hat{a}_i - \sqrt{right_i} - \sum_{j=i+1}^n l_{j,i}(a_j - \hat{a}_j) &\leq a_i \leq \hat{a}_i + \sqrt{right_i} - \sum_{j=i+1}^n l_{j,i}(a_j - \hat{a}_j)
\end{aligned}$$

8.5 模糊度搜索空间尺寸的确定

对于模糊度搜索的多维椭球区域，其搜索空间的大小完全取决于 χ^2 的选择，为了获得合理的 χ^2 值，存在多种选择的办法和手段，这里本文仅讨论在实际应用中经常考虑的两种方式。第一种方式是确保搜索空间内至少存在一个整数矢量，第二种方式是确保搜索空间内至少存在两个整数矢量。后一种方式对于实际应用非常重要，采用这种方式可以提高了模糊度搜索的可靠性，这是因为，在搜索空间中不仅包含了到实数模糊度估计矢量距离最短整数矢量，并且一定还包含了到实数模糊度估计矢量距离次最短整数矢量。

第一种方式处理方法如下：对于去相关处理后得到的新的实数模糊度估计矢量 \hat{z} ，进行简单的取整处理，得到最接近 \hat{z} 的整数矢量，然后利用得到的整数矢量替代不等式中的 z ，取等式成立计算出相应 χ^2 值。

这种方法确保了搜索空间至少包含一个整数矢量，也就是 \hat{z} 的取整值。由于转换后模糊度具有较低的相关性，使得搜索空间不会包含太多的整数矢量，实际上这种情况是经常出现的，因为，在多数情况下对 \hat{z} 取整得到的整数矢量，就是整数最小二乘估计的解。但是，也应当注意，由于转换后模糊度的相关性即使很小，但仍然不等于零，这就不能确保 \hat{z} 取整得到的整数矢量就是整数最小二乘估计的解，因此，还应当在模糊度搜索空间中进行搜索。

第二种方式处理方法如下：与第一种方式类似，首先对全部 \hat{z} 取整，将获得最接近它的整数矢量，这时就得到了一个整数矢量 z_1 。然后将 z_1 中的一个元素保持原值，其它元素取次接近 \hat{z} 的整数，对于 n 维整数矢量 z_1 而言，共可以构造 n 个这样的整数矢量，这样本文就得到了 $n+1$ 个整数矢量。利用得到的整数矢量集，分别替代不等式中的 z ，取等式成立计算出相应 χ^2 值，这时可以得到 $n+1$ 个不同的 χ^2 值，取其中次最小的 χ^2 值，构造模糊度搜索空间，这就确保了搜索空间至少包含两个整数矢量。实际上采用这种方法构造的搜索空间，所包含的整数矢量个数也不会太多，因此，便于快速搜索。

8.6 程序包使用说明

本程序包主要实现三个功能：

(1) 组成最小二乘方程，求得浮点解，并利用浮点解和协方差矩阵输入求得固定解；

(2) LAMBDA可视化程序；

(3) LAMBDA算法的详细分解，分解的程序分别实现了Z变换和重排列、搜索尺寸确定和搜索四个功能

8.6.1 求取浮点解和固定解

(1) 数据说明：

1) 基线长为3220.336M，接收机型号为双频JAVAD LEGACY，观测时间2003年2月18日上午9:00~11:00，共2个小时，

2) 采样率为2秒，截止高度角为15度。用GPS软件对这段数据进行处理,得到的基线向量为 $\text{REAL VALUE OF BASELINE}=[2.000555546637404\text{E}-001$
 $1.998734104105978\text{E}-001 \quad 9.991743662107913\text{E}-002]$,

3) 取其中762个历元(原始数据中的1-762历元)的L1单频数据,观测卫星是2号、31号、3号、27号、11号、13号和1号。

4) 用GPS软件对这段数据进行处理，其中组成双差时的6个卫星对为：2-31、31-3、3-27、27-11、11-13、13-1。

5) $\text{REAL VALUE OF AMBIGUITY N}=[11 \ 9 \ 23 \ 18 \ 5 \ 0]$

6) 数据存入到DATA.mat文件中，1-2列是卫星PRN号，3-5列是 $\Delta X \Delta Y \Delta Z$ 系数，6-11列是模糊度系数，12列是观测值，3-11列构成最小二乘解算过程的A矩阵。

(2) 程序说明

本程序是在已知GPS载波双差方程系数和观测值的基础上利用最小二乘求解浮点解和协方差，将浮点解和协方差输入的LAMBDA程序求得整周模糊度固定解。

MAIN.m 主程序，程序21行“M=90”，为解算所需要的历元数，当历元数M较小时，整周模糊度不能有效固定：

```
%REAL VALUE OF BASELINE=[2.000555546637404E-001  
1.998734104105978E-001
```

```
%9.991743662107913E-002] 基线向量的真值
```

```
%REAL VALUE OF AMBIGUITY N=[11 9 23 18 5 0] 整周模糊度固定解真值
```

```
clc
```

```

clear
load DATA;    %加载数据
A0=DATA(:,3:11); % READ OBSERVATION EQUATION COEFFICIENT 读取
观测方程系数
L0=DATA(:,12);    % READ OBSERVATION 读取观测值
P0=1/7*[6    5    4    3    2    1
        5   10   8    6    4    2
        4    8   12    9    6    3
        3    6    9   12    8    4
        2    4    6    8   10    5
        1    2    3    4    5    6]; %生成权矩阵
M=90;    % EPOCH 历元数
P=kron(eye(M),P0); % WEIGHT MATRIX 权矩阵
%RESULT OF LS
A=A0(1:6*M,:); % OBSERVATION EQUATION COEFFICIENT OF LS 提取前M
个历元的观测方程系数矩阵
L=L0(1:6*M,:); % OBSERVATION OF LS 提取前M个历元的观测值
ATA= A'*P*A;
X=inv(ATA)*A'*P*L; % LS 最小二乘计算双差值和浮点解
a=X(4:9,:);    % AMBIGUITY FLOAT SOLUTION 提取整周模糊度浮点解
Qx=inv(ATA); % 计算协方差
Qa=Qx(4:9,4:9); % COVARIANCE OF AMBIGUITY FLOAT SOLUTION 整周
模糊度浮点解协方差
[afixed,norm1]=lambda1(a,Qa) % AMBIGUITY FIXED SOLUTION 整周模糊
度解算

```

运行程序屏幕输出：

M=30 时，

afixed =

16	9
11	8
37	14
21	18
18	-2
8	-3

M=90 时，

afixed =

11	16
9	11
23	37
18	21
5	18
0	8

afixed为求得的固定解矩阵，第一列为最优解，第二列为次优解，当M=90时，解算的最优解和真值相同，整周模糊度可以有效固定，当M=30时，整周模糊度不能固定。

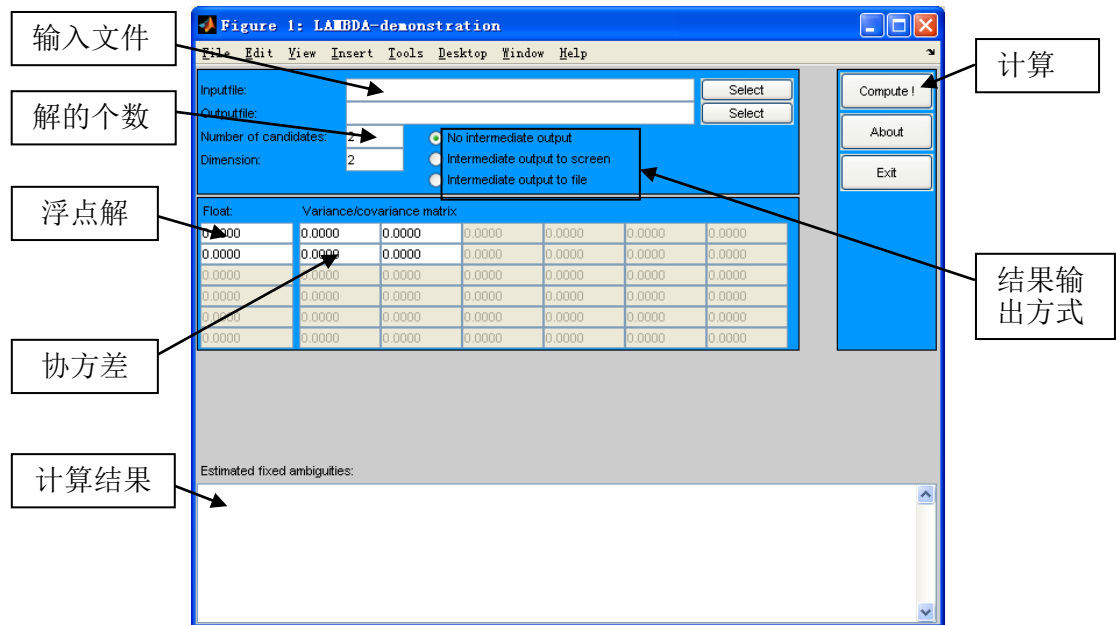
8.6.2 LAMBDA 可视化程序

(1) 数据说明

本部分程序数据包括GEOFREE.MAT、SMALL.MAT、SIXDIM.MAT、LARGE.MAT、AMB18.MAT对应模糊度个数为2、3、6、12、18，本可视化程序只能处理模糊度个数小于6的数据。

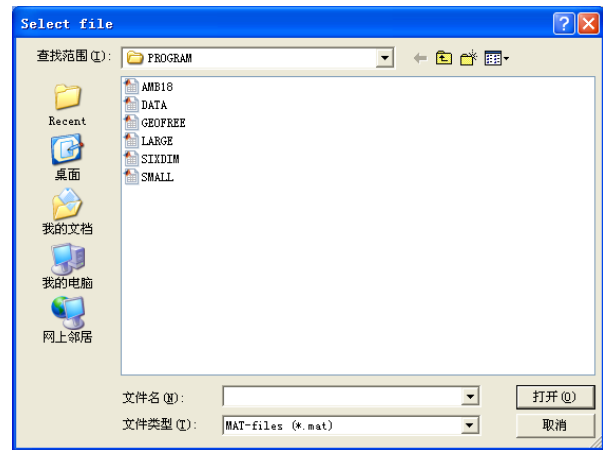
(2) 程序说明：

启动MATLAB，在“Command Window”输入LDEMO，显示可视化界面：



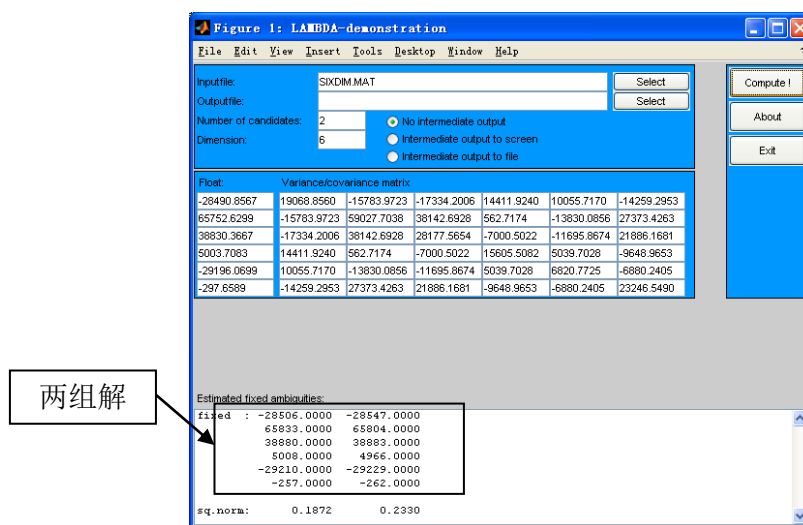
8-1 LAMBDA 可视化程序

点击“Select”按钮，显示：



8-2 选择输入文件

选择合适的数据后，点击“Compute”按钮，得到计算结果：



8-3 计算结果

当Number of candidates输入2时，调用LAMBDA1程序计算固定解；当Number of candidates输入4时，调用LAMBDA2程序计算固定解。LAMBDA1和LAMBDA2区别在于计算输出的参数不同：

function [afixed,sqnorm,Qahat,Z] = lambda1 (afloat,Qahat,ncands,fidlog)

function [afixed,sqnorm] = lambda2 (afloat,Qahat)

文件中的“LAMBDA.PDF”文件对此部分可视化程序有详细说明。

8.6.3 LAMBDA 程序分解

为更加直观理解LAMBDA，将LAMBDA算法分解为decorrel0.m、chistart0.m和lsearch0.m。

decorrel0.m（Z变换和重排列）：输入浮点解a和协方差Q，输出变换之后的L、D矩阵；

chistart0.m（搜索尺寸的确定）：输入L、D矩阵和浮点解a，输出整周模糊度搜索尺寸Chi2；

lsearch0.m（整周模糊度搜索）：输入L、D矩阵、浮点解a和搜索尺寸Chi2，输出固定解afixed；

三个程序可以分别独立运行。

第9章 GPS 高程的多项式曲面拟合

9.1 拟合模型

多项式曲面拟合法是近年来使用的主要拟合方法, 设点的高程异常 ζ 与其平面坐标 x 、 y 有以下关系:

$$\zeta = f(x, y) + \varepsilon \quad (9-1)$$

式中, ζ 为高程异常, $f(x, y)$ 为 ζ 的趋势值, ε 为误差。二次多项式曲面 $f(x, y)$ 可表示为:

$$f(x, y) = a_0 + a_1x + a_2y + a_3x^2 + a_4xy + a_5y^2 + \dots \quad (9-2)$$

(9-1) 式写成矩阵形式有:

$$\zeta = AX + \varepsilon \quad (9-3)$$

式中

$$\zeta = \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_n \end{bmatrix} \quad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix} \quad X = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \quad A = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & \dots \\ 1 & x_2 & y_2 & x_2^2 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & y_n & x_n^2 & \dots \end{bmatrix}$$

(9-3) 式的误差方程为:

$$V = AX - \zeta \quad (9-4)$$

对于每个已知点, 列出上述方程。当已知点个数大于参数个数时, 由最小二乘法拟合出系数参数 a_i , 进而可求出测区内任意点的高程异常值。其中 $f(x, y)$ 常选用三参数、四参数和六参数多项式, 如果取未知数一次项, 则称为平面拟合, 即“最小二乘平面拟合”; 取二次项, 称为二次多项式曲面拟合; 取三次项, 称为三次多项式曲面拟合。

多项式拟合是按削高补低的原则拟合出一个多项式曲面, 来代表拟合区域的似大地水准面, 适于平坦地区。高程异常变化复杂的区域, 所得结果误差较大。对于二次曲面多项式表示的是一个“抛物单曲面”, 仅有一个凹面或凸面, 如果高程逼近场存在多个凹凸面时, 会出现较大误差。

9.2 X, Y 坐标重心化

鉴于计算过程中坐标数值较大, 不方便计算, 在不影响计算结果的前提下, 我们进行坐标重心化, 重心化后对多项式模型中系数的求取无影响, 不会影响到最后高程异常的结果和精度评定。

此处重心化只在二维方向上进行，重心坐标为 $\left(\frac{\sum_{i=1}^n X_i}{n}, \frac{\sum_{i=1}^n Y_i}{n} \right)$ ，然后

用各自方向的 X 和 Y 坐标与其求差，可提高运算效率。

9.3 结果评价

我们通常采用内符合精度和外符合精度来评定拟合结果，内符合精度表明的是拟合点与所选模型的符合程度，而外符合精度表示的是检核点与所选模型的符合程度。

(1) 内符合精度为：根据 n 个参与拟合计算的已知点的值 ξ_i 与拟合值 ξ'_i ，用

$$V_i = \xi_i - \xi'_i \quad (9-5)$$

求拟合残差 V_i ，按下式计算拟合法确定正常高的内符合精度 $m_{\text{内}}$

$$m_{\text{内}} = \pm \sqrt{[V_i V_i] / (n-1)} \quad (9-6)$$

(2) 外符合精度为：根据 t 个检核点的 ξ_i 值与拟合值 ξ'_i 之差 V_i 。按下式计算拟合法确定正常高的外符合精度 $m_{\text{外}}$ 。

$$m_{\text{外}} = \pm \sqrt{[V_i V_i] / (t-1)} \quad (9-7)$$

实验表明，各高程拟合模型的内符合精度都较高，而外符合精度相对较低，所以实验中一般取外符合精度作为评价各拟合方法好坏的标准，称之为拟合中误差，记作 M 。

(3) GPS水准的精度评定

$$\text{三等几何水准: } \varepsilon_{\text{限}} = \pm 12\sqrt{L} \text{ (mm)} \quad (9-8)$$

$$\text{四等几何水准: } \varepsilon_{\text{限}} = \pm 20\sqrt{L} \text{ (mm)} \quad (9-9)$$

式中 L (单位: km) 为测线长度。

9.4 实例应用

试验中采用的数据来自于内蒙古酸刺沟矿区GPS静态观测成果。测区控制点20个，全部进行了水准联测，图9-1为测区地形图，各GPS控制点分布及GPS网形图如下图9-2

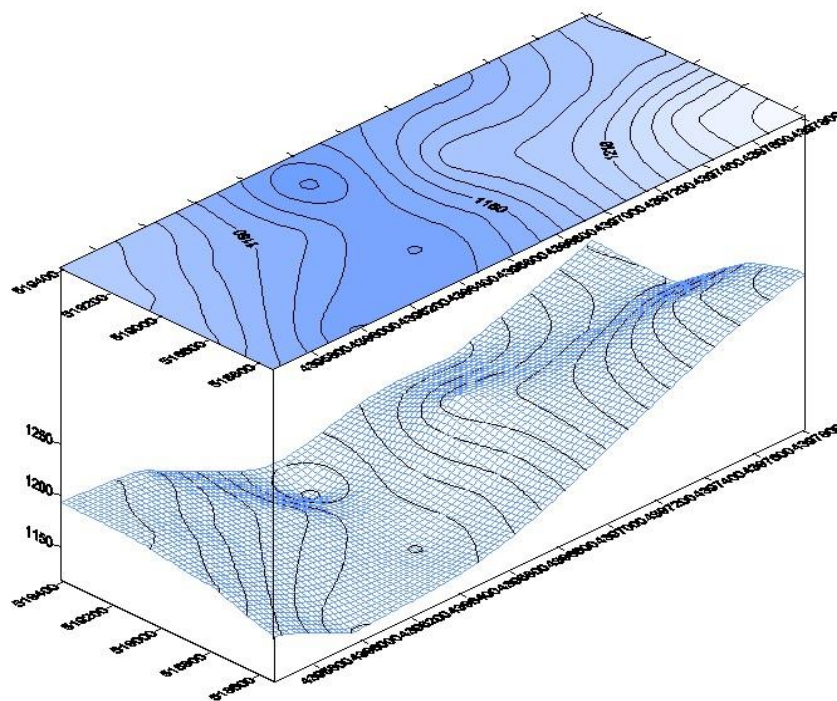


图 9-1 测区地形图

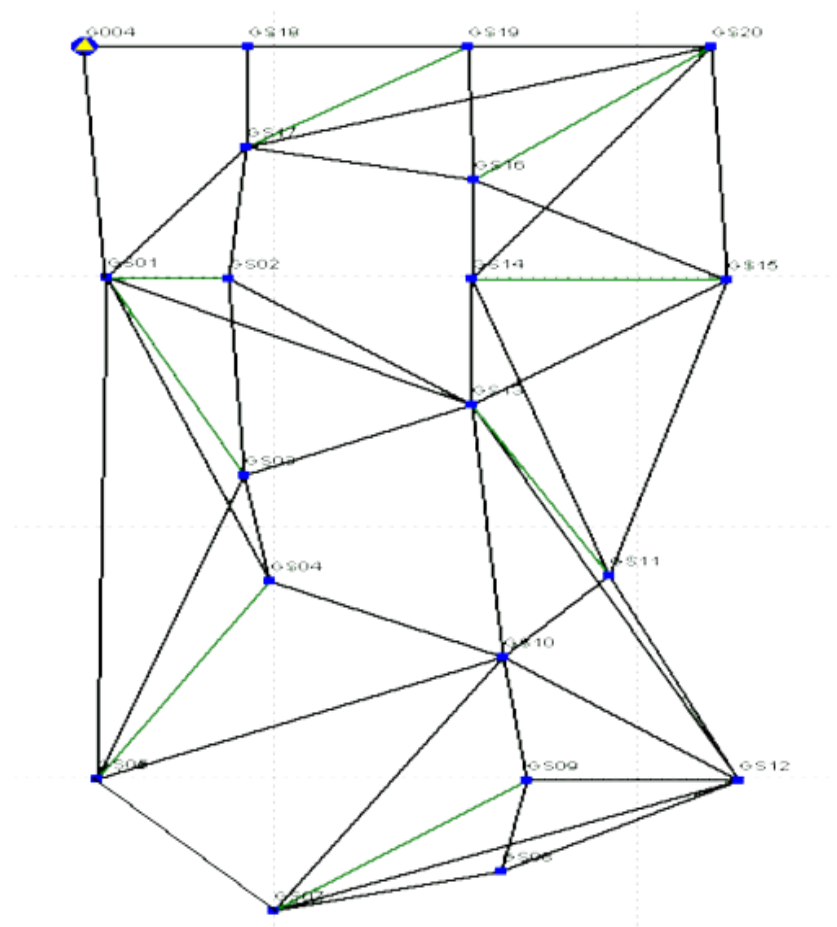


图 9-2 矿区 GPS 网形图

本矿区所选20个控制点中，拟合点14个，如表8-1，检核点6个如表8-2，分别作为基础计算数据。

表 9-1 矿区 GPS 水准拟合点/m

点号	X	Y	大地高	水准高
G004	4397812.422	518567.676	1135.761	1268.8
GS01	4397242.348	518598.796	1084.243	1217.284
GS03	4396755.658	518781.018	1019.463	1152.515
GS04	4396458.155	518770.372	996.003	1129.054
GS05	4395967.756	518543.732	995.584	1128.631
GS07	4395647.298	518776.384	1048.102	1181.139
GS10	4396270.707	519077.595	1020.377	1153.41
GS12	4395967.614	519387.158	1055.402	1188.402
GS14	4397240.453	519080.256	1051.179	1184.212
GS15	4397239.117	519415.653	1020.103	1153.134
GS16	4397483.975	519081.101	1058.219	1191.249
GS18	4397812.373	518782.033	1119.181	1252.216
GS19	4397812.451	519072.061	1067.319	1200.348
GS20	4397811.895	519392.048	1070.013	1203.035

表 9-2 矿区 GPS 水准检核点/m

点号	X	Y	大地高	水准高
GS02	4397241.313	518759.791	1077.026	1210.065
GS08	4395743.952	519077.408	1055.851	1188.869
GS09	4395967.745	519107.966	1036.757	1169.779
GS11	4396510.181	519261.591	983.278	1116.317
GS13	4396931.061	519079.137	1057.159	1190.183
GS17	4397565.166	518782.106	1114.692	1247.724

9.5 程序代码及说明

由于多项式拟合原理相同，而且二次多项式拟合工程中应用较多，故此处针对二次多项式拟合代码加以说明：

nihemain.m %nihemain-主函数

```

*****

% % 二次曲面多项式拟合
clc;clear;
ofile1='sznhd.txt';
% % 读取拟合点数据（Pointnv1-点号，Obs1-X,Y,Z,H，n-点个数）
[Pointnv1,Obs1,n]=readnhobs(ofile1);
% % 坐标重心化
[pointX1,pointY1,anomaly1]=gravity(Obs1,n);
% % 初始化
[aa]=initialization(pointX1,pointY1,anomaly1,6);
% % 求取多项式系数
[a]=correct(pointX1,pointY1,anomaly1,n,aa);
% % 精度及残差输出
[nfhpre,res1]=precision(pointX1,pointY1,anomaly1,n,a)

% % 对检核点进行拟合
ofile2='jhd.txt';
% % 读取拟合点数据
[Pointnv2,Obs2,m]=readjhobs(ofile2);
% % 坐标重心化
[pointX2,pointY2,anomaly2]=gravity(Obs2,m);
% % 精度及残差输出
[wfhpre,res2]=precision(pointX2,pointY2,anomaly2,m,a)
*****

readjhobs .m          % readjhobs-读取 txt 文档“检核点”的函数
*****

function [Pointnv,Obs,m]=readjhobs(file)
fid=fopen(file,'rt');
Obs=[];
Pointnv=[];
m=0;
while ~feof(fid)
    line=fgetl(fid);
    m=m+1;
end

```

```

    [pointnv,x,y,z,h]=strread(line,'%s %f %f %f %f', 'delimiter', ',');
    Pointnv=[Pointnv;pointnv];
    Obs=[Obs;x y z,h];
end
fclose(fid);

% readjhobs-读取 txt 文档“检核点”的函数
.....省略，原理相同
*****

gravity.m                                % 重心化坐标
*****

% point——写入点的坐标（包括正常高）（X,Y,Z,H）
% n——写入点的个数
% pointX——X 重心化后的值
% pointY——Y 重心化后的值
% anomaly——写入点的高程异常
function [pointX,pointY,anomaly]=gravity(point,n)
sumx=0;
sumy=0;
x=0;
y=0;
for i=1:n
    x=x+point(i,1);
    y=y+point(i,2);
end
xz=x/n;
yz=y/n;
% % X,Y 坐标重心化
for i=1:n
    pointX(i,1)=point(i,1)-xz;
    pointY(i,1)=point(i,2)-yz;
    anomaly(i,1)=point(i,3)-point(i,4);
end
*****

```

```

item .m                                % 项数矩阵
*****

% pointX——X 重心化后的值
% pointY——Y 重心化后的值
% m——初始化用到点的个数
% A——项数矩阵
function A=item(pointX,pointY,m)
A0=ones(m,1);
A1=pointX(1:m,1);
A2=pointY(1:m,1);
for i=1:m
    A3(i,1)=pointX(i,1)^2;
    A4(i,1)=pointY(i,1)^2;
    A5(i,1)=pointX(i,1)*pointY(i,1);
end
A=[A0 A1 A2 A3 A4 A5];
*****

initialization .m                      % 初始化
*****

% pointX——X 重心化后的值
% pointY——Y 重心化后的值
% anomaly——写入点的高程异常
% m——初始化用到点的个数
% aa——初始化所得系数
function [aa]=initialization(pointX,pointY,anomaly,m)
A=item(pointX,pointY,m);
b=anomaly(1:m);
% 行列式-看是否奇异
ans=det(A);
if (ans~=0)
    aa=A\b;
end
*****

correct .m                             % 求取改正后多项式系数

```

```

*****

% pointX——X 重心化后的值
% pointY——Y 重心化后的值
% anomaly——写入点的高程异常
% n——写入点的个数
% aa——初始化所得系数
% a——改正后多项式系数
function [a]=correct(pointX,pointY,anomaly,n,aa)
% % 最小二乘求多项式系数
[B]=item (pointX,pointY,n);
for i=1:n

l(i,:)=anomaly(i)-(aa(1)*B(i,1)+aa(2)*B(i,2)+aa(3)*B(i,3)+aa(4)*B(i,4)+aa(5)*B(i,5)
+aa(6)*B(i,6));
end
da=inv(B'*B)*B'*l;
% % 改正 a
a=aa+da;
*****

precision .m % 残差及精度求取
*****

% pointX——X 重心化后的值
% pointY——Y 重心化后的值
% anomaly——写入点的高程异常
% n——写入点的个数
% precision——精度
% res——残差
function [precision,res]=precision(pointX,pointY,anomaly,n,a)
%res1-残差
[B]=item(pointX,pointY,n);
sum=0;
for i=1:n

anomaly0(i,1)=a(1)*B(i,1)+a(2)*B(i,2)+a(3)*B(i,3)+a(4)*B(i,4)+a(5)*B(i,5)+a(6)*B(

```

```
i,6);  
    %放大 100 倍（单位 cm）  
    res(i,1)=(anomaly(i,1)-anomaly0(i,1))*100;  
    vv(i,1)=(res(i,1))^2;  
    sum=sum+vv(i,1);  
end  
%内符合精度  
precision=sqrt(sum/(n-1));
```

第 10 章 GPS 通信协议——NMEA-0183

10.1 NMEA 通信协议

GPS接收机与手持机之间的数据交换格式一般都由生产厂商缺省定制，其定义内容普通用户很难知晓，且不同品牌、不同型号的GPS接收机所配置的控制应用程序也因生产厂家的不同而不同。所以，对于通用GPS应用软件，需要一个统一格式的数据标准，以解决与任意一台GPS的接口问题。NMEA-0183数据标准就是解决这类问题的方案之一。NMEA协议是为了在不同的GPS导航设备中建立统一的RTCM（海事无线电技术委员会）标准，它最初是由美国国家海洋电子协会（NMEA—The National Marine Electronics Association）制定的。NMEA协议有0180、0182和0183这3种，0183可以认为是前两种的升级，这一标准在兼容NMEA-0180和NMEA-0182标准的基础上，增加了GPS、测深仪、罗经方位系统等多种设备的接口和通信协议定义，同时，标准还允许一些特定设备制造商对其设备通信自行定义协议。由于NMEA-0183标准的通用性和灵活性，因而成为目前使用最为广泛的一种。GPS接收机根据NMEA-0183协议的标准规范，将位置、速度等信息通过串口传送到PC机、PDA等设备。

10.2 NMEA 通讯协议硬件接口

符合NMEA0183标准的GPS接收机的硬件接口能够兼容计算机的RS-232C协议串口，然而，严格来说NMEA标准不是RS-232C，规范推荐发送设备的电平符合EIA422（也称为RS-422），是一个与RS-232C不同的系统。标准RS-232C采用负逻辑，即逻辑“1”表示-5V~-15V，逻辑“0”表示+5V~+15V，利用传输信号线和信号地线之间的电压差进行传输。而EIA-422是利用导线之间的信号电压差来传输信号的，其每个通道要用两条信号线，一条是逻辑“1”，一条是逻辑“0”，通过传输线驱动器和传输线接收器实现逻辑电平和电位差之间的转换，一般允许驱动器输出为 $\pm 2V \sim \pm 6V$ 。

虽然存在区别，但在实际使用中，如果只是接收GPS的输出，则只需两根信号线 GPS数据输出线和信号地线，可以直接将EIA-422输出通道两条信号线中的一条同计算机的RS-232C输入线相连。

NMEA通讯协议所定义的标准通讯接口参数为：

波特率：4800bit/s；

数据位：8位（第7位为0）；

停止位：1位（或者更多）；

奇偶校验：无；

10.3 GPS 测量中 NMEA 语句解析

NMEA通讯协议所规定的通讯语句都已是以ASCII码为基础的，NMEA-0183协议语句的数据格式如下：“\$”为语句起始标志；“,”为域分隔符；“*”为校验和识别符，其后面的两位数为校验和，代表了“\$”和“*”之间所有字符的按位异或值（不包括这两个字符）；“/”为终止符，所有的语句必须以来结束，也就是ASCII 字符的“回车”（十六进制的0D）和“换行”（十六进制的0A）。

NMEA-0183标准应用于GPS方面时，数据串以“\$GP”开头，主要有GGA、GLL、ZDA、GSV、GST、GSA、ALM等格式，这些格式的作用分别是：

\$GPGGA：输出GPS的定位信息；\$GPGLL：输出大地坐标信息；

\$GPGSV：输出可见的卫星信息；\$GPZDA：输出UTC时间信息；

\$GPGST：输出定位标准差信息；\$GPGSA：输出卫星DOP值信息；

\$GPALM：输出卫星星历信息。

下面给出一些常用NMEA-0183语句的字段定义解释：

\$GPGGA（定位信息）

例：\$GPGGA,092204.999,4250.5589,S,14718.5084,E,1,04,24.4,19.7,M,,,0000*1F

字段 0	\$GPGGA，语句 ID，表明该语句为 Global Positioning System Fix Data（GGA）GPS 定位信息
字段 1	UTC 时间，hhmmss.sss，时分秒格式
字段 2	纬度 ddmm.mmmm，度分格式（前导位数不足则补 0）
字段 3	纬度 N（北纬）或 S（南纬）
字段 4	经度 dddmm.mmmm，度分格式（前导位数不足则补 0）
字段 5	经度 E（东经）或 W（西经）
字段 6	GPS 状态，0=未定位，1=非差分定位，2=差分定位，3=无效 PPS，6=正在估算
字段 7	正在使用的卫星数量（00 - 12）（前导位数不足则补 0）
字段 8	HDOP 水平精度因子（0.5 - 99.9）
字段 9	海拔高度（-9999.9 - 99999.9）
字段 10	地球椭球面相对大地水准面的高度
字段 11	差分时间（从最近一次接收到差分信号开始的秒数，如果不是差分定位将为空）
字段 12	差分站 ID 号 0000 - 1023（前导位数不足则补 0，如果不是差分定位将为空）
字段 13	校验值

\$GPGSA（当前卫星信息）

例：\$GPGSA,A,3,01,20,19,13,,,,,,,,,40.4,24.4,32.2*0A

字段 0:	\$GPGSA, 语句 ID, 表明该语句为 GPS DOP and Active Satellites (GSA) 当前卫星信息
字段 1:	定位模式, A=自动手动 2D/3D, M=手动 2D/3D
字段 2	定位类型, 1=未定位, 2=2D 定位, 3=3D 定位
字段 3	PRN 码 (伪随机噪声码), 第 1 信道正在使用的卫星 PRN 码编号 (00) (前导位数不足则补 0)
字段 4	PRN 码 (伪随机噪声码), 第 2 信道正在使用的卫星 PRN 码编号 (00) (前导位数不足则补 0)
字段 5	PRN 码 (伪随机噪声码), 第 3 信道正在使用的卫星 PRN 码编号 (00) (前导位数不足则补 0)
字段 6	PRN 码 (伪随机噪声码), 第 4 信道正在使用的卫星 PRN 码编号 (00) (前导位数不足则补 0)
字段 7	PRN 码 (伪随机噪声码), 第 5 信道正在使用的卫星 PRN 码编号 (00) (前导位数不足则补 0)
字段 8	PRN 码 (伪随机噪声码), 第 6 信道正在使用的卫星 PRN 码编号 (00) (前导位数不足则补 0)
字段 9	PRN 码 (伪随机噪声码), 第 7 信道正在使用的卫星 PRN 码编号 (00) (前导位数不足则补 0)
字段 10	PRN 码 (伪随机噪声码), 第 8 信道正在使用的卫星 PRN 码编号 (00) (前导位数不足则补 0)
字段 11	PRN 码 (伪随机噪声码), 第 9 信道正在使用的卫星 PRN 码编号 (00) (前导位数不足则补 0)
字段 12	PRN 码 (伪随机噪声码), 第 10 信道正在使用的卫星 PRN 码编号 (00) (前导位数不足则补 0)
字段 13	PRN 码 (伪随机噪声码), 第 11 信道正在使用的卫星 PRN 码编号 (00) (前导位数不足则补 0)
字段 14	PRN 码 (伪随机噪声码), 第 12 信道正在使用的卫星 PRN 码编号 (00) (前导位数不足则补 0)
字段 15	PDOP 综合位置精度因子 (0.5 - 99.9)
字段 16	HDOP 水平精度因子 (0.5 - 99.9)
字段 17	VDOP 垂直精度因子 (0.5 - 99.9)
字段 18	校验值

\$GPGSV (可见卫星信息)

例: \$GPGSV,3,1,10,20,78,331,45,01,59,235,47,22,41,069,,13,32,252,45*70

字段 0	\$GPGSV, 语句 ID, 表明该语句为 GPS Satellites in View (GSV) 可见卫星信息
字段 1	本次 GSV 语句的总数目 (1 - 3)
字段 2	本条 GSV 语句是本次 GSV 语句的第几条 (1 - 3)
字段 3	当前可见卫星总数 (00 - 12) (前导位数不足则补 0)
字段 4	PRN 码 (伪随机噪声码) (01 - 32) (前导位数不足则补 0)
字段 5	卫星仰角 (00 - 90) 度 (前导位数不足则补 0)
字段 6	卫星方位角 (00 - 359) 度 (前导位数不足则补 0)
字段 7	信噪比 (00 - 99) dbHz
字段 8	PRN 码 (伪随机噪声码) (01 - 32) (前导位数不足则补 0)
字段 9	卫星仰角 (00 - 90) 度 (前导位数不足则补 0)
字段 10	卫星方位角 (00 - 359) 度 (前导位数不足则补 0)
字段 11	信噪比 (00 - 99) dbHz
字段 12	PRN 码 (伪随机噪声码) (01 - 32) (前导位数不足则补 0)
字段 13	卫星仰角 (00 - 90) 度 (前导位数不足则补 0)
字段 14	卫星方位角 (00 - 359) 度 (前导位数不足则补 0)
字段 15	信噪比 (00 - 99) dbHz
字段 16	校验值

\$GPRMC (推荐定位信息数据格式)

例: \$GPRMC,024813.640,A,3158.4608,N,11848.3737,E,10.05,324.27,150706,,,A*50

字段 0	\$GPRMC, 语句 ID, 表明该语句为 Recommended Minimum Specific GPS/TRANSIT Data (RMC) 推荐最小定位信息
字段 1	UTC 时间, hhmmss.sss 格式
字段 2	状态, A=定位, V=未定位
字段 3	纬度 ddmm.mmmm, 度分格式 (前导位数不足则补 0)
字段 4	纬度 N (北纬) 或 S (南纬)
字段 5	经度 dddmm.mmmm, 度分格式 (前导位数不足则补 0)
字段 6	经度 E (东经) 或 W (西经)
字段 7	速度, 节, Knots
字段 8	方位角, 度
字段 9	UTC 日期, DDMMYY 格式
字段 10	磁偏角, (000 - 180) 度 (前导位数不足则补 0)
字段 11	磁偏角方向, E=东 W=西
字段 16	校验值

\$GPVTG（地面速度信息）

例：\$GPVTG,89.68,T,,M,0.00,N,0.0,K*5F

字段 0	\$GPVTG，语句 ID，表明该语句为 Track Made Good and Ground Speed（VTG）地面速度信息
字段 1	运动角度，000 - 359，（前导位数不足则补 0）
字段 2	T=真北参照系
字段 3	运动角度，000 - 359，（前导位数不足则补 0）
字段 4	M=磁北参照系
字段 5	水平运动速度（0.00）（前导位数不足则补 0）
字段 6	N=节，Knots
字段 7	水平运动速度（0.00）（前导位数不足则补 0）
字段 8	K=公里/时，km/h
字段 9	校验值

\$GPGLL（地理定位信息）

例：\$GPGLL,4250.5589,S,14718.5084,E,092204.999,A*2D

字段 0	\$GPGLL，语句 ID，表明该语句为 Geographic Position（GLL）地理定位信息
字段 1	纬度 ddm m.mmm m，度分格式（前导位数不足则补 0）
字段 2	纬度 N（北纬）或 S（南纬）
字段 3	经度 ddd m m.mmm m，度分格式（前导位数不足则补 0）
字段 4	经度 E（东经）或 W（西经）
字段 5	UTC 时间，hhmmss.sss 格式
字段 6	状态，A=定位，V=未定位
字段 7	校验值

10.4 NMEA 数据处理中的注意事项

如果开发基于GPS的应用系统，就需要将GPS作为信息源，正确接收和解析GPS发送的NMEA—0183数据。此时，在编程实现时需要注意几个问题：

（1）通讯端口的设置

虽然NMEA规范推荐的串行通讯参数为“波特率：4800；奇偶校验：无；数据位：8；停止位：1，但也有厂商的产品允许用户将波特率设置的更高，此时需要注意设置计算机的接口参数与GPS设备一致。

（2）所需信息的正确提取

NMEA—0183是以语句形式发送数据的，接收机可能发送很多类型的语句，而我们需要的可能只是某些语句中的几个字段。因此就需要对接收到的数据进行

解析，取得所需的信息。另外，可能会由于小数点位数不同等原因，语句的长度是可变的，因而分离感兴趣的信息时，不能按照该信息在语句中所处的字符位置来查找，只能依据逗号分隔符，这一点在数据提取的过程中非常重要。

第 11 章 实验数据模拟

11.1 加载 GPS 轨道参数

根据gps轨道参数的.mat文件，利用load（）函数加载卫星轨道参数。

程序：

```
load gpskep
SVIDV = [SVIDV; gpskep(:,1)]; % 卫星ID号
MV = [MV; gpskep(:,2)];      % 平近点角（单位“度”）
OMGV = [OMGV; gpskep(:,3)];  % 卫星升交点赤经（单位“度”）
RV = [RV; gpskep(:,4)];      % 卫星轨道半径（单位“米”）
INCLV = [INCLV; gpskep(:,5)]; % 卫星轨道倾角（单位“度”）
TOEV = [TOEV; gpskep(:,6)];  % 卫星参考时刻（单位“秒”）
```

例子：已知GPS轨道参数文件（gpskep.mat），编写一个加载GPS轨道参数的文件（调试程序见loadgps01.m）。

11.2 卫星位置的计算

输入卫星轨道参数，计算卫星的位置。

程序：

```
mu = 3986005e8; % mu 即万有引力常数mu=GM=3986005e8
OMGedot = 7.2921151467e-5; % 地球自转角速度（单位“弧度/秒”）
inclr = incl*pi/180; % 角度也弧度之间的转换
n = sqrt(mu/(r*r*r)); % 计算参考时刻(toe)的平均角速度
tk = t - toe;
Mk = Mo*pi/180 + n*tk;
OMGk = OMGo*pi/180 - OMGedot*tk - OMGedot*toe;
ip(1,1) = r*cos(Mk);
ip(2,1) = r*sin(Mk);
cosomg = cos(OMGk);
sinomg = sin(OMGk);
cosi = cos(inclr);
ROT = [cosomg  -cosi*sinomg; ...
       sinomg  cosi*cosomg; ...
```

```

0      sin(inclr)]; %旋转矩阵
svxyz = ROT*ip;%卫星位置
例子：已知以下卫星轨道参数，计算卫星位置（调试程序见svpos01.m）
r = 2.6560623690000000e+07;
toe = 0;
Mo = 1.9096000000000000e+02;
OMGo = 3.2573028400000000e+02;
incl = 55;
t = 39600;

```

11.3 卫星天空分布图的绘制

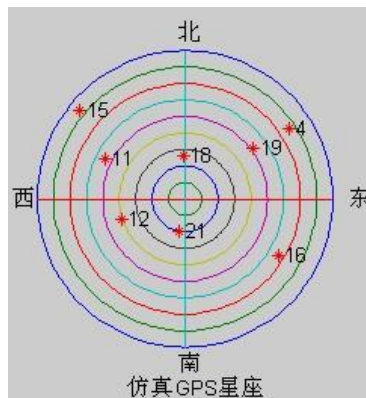
根据GPS时间、卫星轨道参数和高度截止角可以画出该时刻可见卫星的天空分布图。

主要程序如下：

```

t = 39600; %给定的GPS时间
usrllh = [34.18*pi/180 117.17*pi/180 100]; %某点的坐标北纬34.18°，东经117.17°
usrxyz = llh2xyz(usrllh);
loadgps %加载GPS卫星开普勒参数
[svxyzmat,svid,EL] = gensv(usrxyz,t,5); %（测站位置，t时刻，高度角）计算t
时刻可见卫星的位置
skyplot(svxyzmat,svid,usrxyz) %生成卫星方位角和仰角图text(0.4,0.9,'Simulated
GPS Constellation')
text(0.4,0.8,'User at Lat 34.18 deg, Lon 117.17 deg')
运行结果：

```



例子：已知某点的坐标为（34.18°N，117.17°E）、GPS卫星轨道参数和gps时刻

t=39600, 模拟该点上空的可见卫星分布图。（调试程序见 satellite_plotting.m）

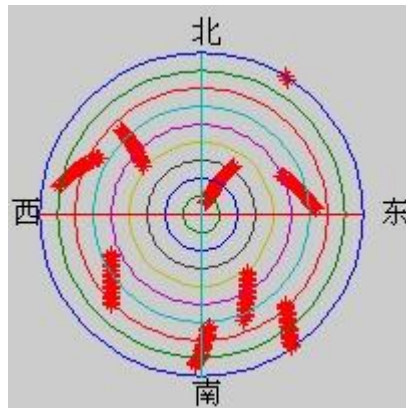
11.4 卫星在天空中的运动情况

我们知道，在已知卫星地面某点坐标和时间后可计算出卫星在天空中的位置，现在以某一点为坐标的中心位置，通过一个for循环来实现该点上空某一时间点可见卫星的运动轨迹图。

程序：

```
usrllh = [34.18*pi/180 117.17*pi/180 0];    %地面某一点的经纬度
usrxyz = llh2xyz(usrllh);    %坐标转换
loadgps                                     %加载GPS轨道参数
for t = 200:200:3400, %卫星运动时间
    [svxyzmat,svid] = gensv(usrxyz,t,0); %计算卫星位置
    pause(0.1)
    skyplot(svxyzmat,svid,usrxyz,0,0)
    hold on
end
```

运行结果：



例子：已知某点的大地经纬度（34.18°N，117.17°E），请绘制出在t=[200:3400]时刻该点上空可见卫星的运动轨迹图。（具体程序见satellite_motion.m）

11.5 DOP 值的计算

根据方向余弦矩阵H，可以求解DOP值。

程序：

```
dm = inv(H'*H);
```

```
dop (1) = sqrt(trace(dm(1:2,1:2))); % HDOP
dop (2) = sqrt(trace(dm(1:3,1:3))); % PDOP
dop (3) = sqrt(dm(4,4));           % TDOP
dop (4) = sqrt(trace(dm));          % GDOP
```

例子：已知方向余弦矩阵H，求解DOP值（调试程序见dop01.m）。其中

H=

-0.575267692562410	-0.282194358018692	-0.767745678070765	1
0.395436332553577	-0.686310858408309	-0.610415852126587	1
0.663049311240294	-0.582573049299453	0.470078986015870	1
-0.347623062776392	-0.935975796553668	-0.0557450849105693	1
-0.453641271749212	-0.485386400990706	-0.747401925538761	1
0.298494814650756	-0.519823120533455	-0.800427866197493	1
0.883527222476291	-0.235645855032620	-0.404784730627621	1
0.504229536981618	-0.847836725492826	0.164089795358841	1
0.779215619809086	-0.468193193468515	-0.416675114970040	1
0.655330469943731	0.187140957560124	-0.731792482310941	1
0.475920977048175	-0.334128391902954	-0.813546213395316	1
0.242898490859542	-0.965443902807577	-0.0944372472588566	1
0.966132752688850	-0.251506482499427	0.0577234219587820	1
0.499655045945132	-0.466599621641382	-0.729814790303463	1
-0.412198539439776	-0.219489528107253	-0.884260544825433	1

第 12 章 TEQC 软件介绍

本章首先介绍了TEQC软件的基本功能，包括格式转换、数据编辑、质量检查、单点定位。接着列出了一些TEQC软件的常用命令，并作了详细的注释。

12.1 TEQC 功能简介

TEQC (Translate/Edit/QualityCheck/ Coordinate) 是由UNAVCO Facility研制的为地学研究GPS监测站数据管理服务的公开免费软件，可用于检查双频GPS接收机的动态和静态数据质量。它利用伪距观测值和载波相位观测值的线形组合来进行GPS数据中的误差估计，在快速评定GPS数据质量方面有非常大的优势，一方面速度快，没有繁琐的操作步骤，只用几条简单的命令即可；另一方面能对GPS观测数据进行多角度全方位的质量分析，能分别从卫星高度角、方位角、多路径效应、电离层延迟误差、电离层延迟变化率、信噪比等方面在Qcview配合用图形的形式直观的反映GPS观测数据的质量。

TEQC主要有格式转换(Translate)、数据编辑(Edit)、质量检查(QualityCheck)单点定位(Coordinate)四方面的功能。其中，格式转换可将许多不同厂家的GPS接收机观测文件转换成标准格式RINEX文件；编辑功能可用于对RINEX文件的字头块部分，也可以进行数据文件的任意切割与合并、卫星系统的选择和删减、接收机通道的选择和卫星高度角的设置等；质量检查可以反映出GPS数据的电离层延迟、多路径效应、周跳、信噪比等信息；单点定位则可以粗略计算出点位在WGS-84坐标系中的坐标和在大地坐标系中的坐标。它的四个模块是相互独立、互不影响的，既可以单独使用其中的一个模块，也可以组合使用。它不仅可以处理GPS系统，也可以处理GLONASS系统和GNSS系统。但它也有明显的不足之处：基于DOS界面，对它的操作是建立在命令的基础上，因此要想用好它必须熟悉许多繁杂的命令；能很好的反映观测数据的质量，但在对观测数据的修复改正功能还不是很完善。

12.2 TEQC 命令介绍

下面简单介绍TEQC软件的操作格式。TEQC软件的命令格式非常有规律，其基本格式为：`teqc{options}[File1 File2...]>File`，其中teqc是可执行程序的名字（如teqc.exe），options控制参数，teqc软件包含了三百种左右的参数，可以控制完成各种功能，如格式转换、数据编辑、质量检查、点位坐标计算和帮助等。在参数前总会包含有“+”或“-”符号，“+”表示打开某一参数功能，也可以表示输出数据到屏幕（或文件）。“-”表示关闭某一参数功能或输入数据到文件。File1、File2为待处

理的数据文件。>File表示把处理结果保存到File中，若File所在目录已有名为File的文件，则覆盖之，否则自动创建名为File的文件。

12.2.1 RINEX 格式转换

不同的GPS接收机采用不同的数据格式，这给GPS数据处理软件造成了一定的困难，为了解决后处理软件数据的交换和输入格式问题，伯尔尼大学天文研究所提出了独立于接收机类型的数据交换格式RINEX，这一数据格式在科学研究领域得到了广泛的应用。大部分GPS接收机生产厂家都提供了相应的转换软件，但并非完全标准，在实际使用中，经常遇到格式不兼容的问题。TEQC软件的格式转换具有很好的通用性和较强的功能，适用于目前较常用的多数GPS接收机，而且还在不断的扩展。它主要是通过读入数据文件的开头部分来自动识别接收机的类型。

目前，TEQC可以转换的接收机数据文件大多是国外比较有名的GPS接收机，比如：美国的AOA、ASHTECH、MOTOROLA、TRIMBLE、TECOM INDUSRTIES、ROGUE,瑞士的LEICA，加拿大的CMC，瑞士的WILD等，但是却不能识别中国的HD系列接收机和南方的北极星系列接收机。

以TRIMBLE 4700为例，介绍把TRIMBLE 4700接收机所接收到的数据文件转换成RINEX观测数据文件、导航数据文件和气象文件，其他类型的接收机转换与之类似。命令格式如下：（注意TEQC区分大小写，所以我们在写命令时一定要注意。）

```
teqc -tr do -week 1323 +nav trimbel.05n trimbel.dat > trimbel. 05o
```

式中，-tr 指明接收机的类型为trimble；do指明输入文件为Dat文件、输出为RINEX 观测数据文件（o文件）；-week 1323（可选）指明观测日期对应的GPS周数，或以年/月/日方式表示（即上述命令也可以表示成teqc -tr do -week 2005 / 05 / 18 +nav trimbel.05n trimbel.dat > trimbel. 05o）；+nav 指明同时输出RINEX导航数据文件；转换结果文件为观测数据文件trimbel.05o 和导航文件trimbel.05n。

其中，上述命令的控制参数设置项根据GPS种类的不同可自行设置，常用的GPS类型对应的控制参数如下：

表 12-1 常用 GPS 对应的设置项

控制参数	对应的 GPS 类型	控制参数	对应的 GPS 类型
-aoa	AOA(JPL)	-rock	Rockwell
-ash	Ashtech	-tr	Trimble
-cmc	Canadian Marconi	-ti	Tecom Indusrties
-lei	Leica	-rogue	Rogue
-mot	Motorola	-wild	Wild

12.2.2 数据编辑

TEQC具有强大的数据编辑功能。下面将把其主要的编辑功能命令逐一介绍:

(1) 头文件的编辑

RINEX的观测数据文件 (*.??o)、导航数据文件(*.??n)和气象数据文件(*.??m)的文件头部分均可利用TEQC软件进行设置和更改,还可添加新的注释行且原有的注释行保持不变。利用TEQC软件对RINEX观测数据文件 (*.??o) 的字头块部分进行设置和更改的格式为:

```
teqc -O.* “ ” input file > output file
```

```
teqc -O.mo good trimble.05o > temprimble.05o
```

式中, -O.mo good 指明将文件的“测站名”记录更改为“good”, temprimble.05o 为更改后输出的RINEX观测数据文件。在这里必须重新设置输出文件名, 即不能与被处理文件trimble.05o同名, 否则输出文件为空文件。

除了可更改测站名外, 还可以修改其他信息。其他常用的设置项如表12-2所示。

表 12-2 o5 文件常用设置项

指令	含义	指令	含义
-O.mn	设置测站点编号	-O.at	设置天线类型
-O.rn	设置接收机编号	-O.pe	设置天线高偏心改正
-O.rt	设置接收机类型	-O.int	设置采样间隔 (s)
-O.an	设置天线编号	-O.c	追加注释

RINEX导航数据文件和气象数据文件的编辑工作与观测数据文件类似, 只不过导航数据文件的设置项以-N.开头, 气象数据文件以-M.开头, 而观测文件是以-O. 开头。

(2) RINEX文件的切割

在我们进行GPS数据观测时, 一般情况下刚开始一段时间的观测数据精度较差, 所以对观测数据文件中时间的选取就势在必行。在TEQC中, 利用时间窗可对RINEX文件进行任意的切割, 使得对RINEX文件的提取相当容易。时间窗的设置常采用以下几种格式:

1) [start] +d[Y,M ,d ,h ,m ,s] 指定以RINEX文件开始观测时间为上限的时间间隔+d[Y,M ,d ,h ,m ,s]。若提取的时间段以年(月/日/小时/分钟/秒)为单位则控制参数应为dY (dM/dd/dh/dm/ds),例如提取trimble中前10个小时的数据命令格式如下:

```
teqc +dh 10 timble.05o>trimble10.05o
```

2)-d[Y,M,d,h,m,s][end] 指定以RINEX 文件结束观测时间为下限的时间间隔-d[Y,M,d,h,m,s]的数据。

3) -st [YYMMddhhmmss[. sss ...]] [end] 指定时间窗上限-st [YYMMddhhmmss[. sss ...]],默认以RINEX文件结束观测时间为下限。

4) [start]-e[YYMMddhhmmss[. sss ...]] 指定时间窗的下限- e[],默认以RINEX文件开始观测时间为上限。

5) [YYMMddhhmmss[. sss ...]] - e [YYMMddhhmmss[. sss ...]], 提取时间段 st [YYMMddhhmmss[. sss ...]]到e[YYMMddhhmmss[. sss ...]]的观测数据。如：要想提取trimble中10~13点的数据，命令如下：

```
teqc -st 20040420100000 -e 20040420120000 trimble.05o>trimble1012.05o
```

6) -st[YYMMddhhmmss[. sss ...]] + d[Y,M,d,h,m,s],提取从 st[YYMMddhhmmss[. sss ...]]开始顺延d[Y,M,d,h,m,s]时间的数据信息。

(3) RINEX文件的合并

TEQC既然可以对RINEX进行切割，就可以对其进行合并。但要注意的是要合并的文件除了要求是RINEX格式外，还必须实在时间上连续的文件。合并的命令格式如下：

```
teqc>file1 file2...>myfile
```

表示把file1, file2...合并为myfile并输出。

(4) 卫星系统的选择和特定卫星的禁用

对于GPS/GLONASS双星接收机，可以使用TEQC 进行卫星的选用，如去掉GLONASS 卫星数据的指令是：teqc - R 输入文件> 输出文件；禁用prn # 的GPS卫星的观测数据指令是：teqc - G # 输入文件> 输出文件。其中prn #为卫星的编号。

(5) 设置卫星高度角

电离层延迟、多路径效应、接收机噪声是影响GPS数据质量的主要因素，在进行对流层和电离层延迟分析时，需要考虑低高度角卫星，用如下指令可以设置卫星高度角限值。命令格式为：teqc -set-mask x 输入文件> 输出文件。

如要把trimble.05o中卫星的高度角限值设为15度，命令为：teqc -set-mask 15 trimble.05o>trimble15.05o。

(6) 设置观测值类型

GPS/GLONASS的观测值类型一般用L1（L1载波相位）、L2（L2载波相位）、C1（L1的C/A码伪距）、P1（L1的P 码伪距）、P2（L2的P 码伪距）、D1（L1的多普勒观测值）、D2（L2的多普勒观测值）表示，选取并按指定顺序形成RINEX

文件的指令是: `teqc -O. obs“观测值类型”输入文件> 输出文件`, 应用格式如下:

`teqc -O. obs“L1L2P1P2”trumble. 05o > trumblell. new`

(7) 卫星范围的设定

在用GPS接收机的采集数据的时候, 用户可以自行设定接收卫星的最大编号。命令格式为: `teqc -n_GPS # file1>file2`。若接收数据中有大于设定的卫星编号, 则此组数据将被删除。# 的范围是 $0<\#<256$, 默认的卫星编号为32。

`teqc -n_GLONASS # file1>file2` 设定期望的GLONASS卫星的最大编号。若接收数据中有大于设定的卫星编号, 则此组数据将被删除。其中: # 的范围是 $0<\#<256$, 默认的编号为24。

`teqc -n_gssp # file1>file2` 设定期望的gssp卫星的最大编号, 若接收数据中有大于设定的卫星的编号, 此组数据同样也会被删除。# 的范围是 $0<\#<256$, 默认的期望卫星最大编号为51。

(8) 接收机通道的设定

命令解析: `teqc +ch file1>file2` 用GPS接收机的所有的通道 (默认)

`teqc -ch # file1>file2` 禁用gps接收机的# 频道

(9) 多路径误差的有关设置

在TEQC中, 可以对多路径效应进行相应的设置。主要有:

1) 多路径效应的开启和关闭

开启多路径效应的命令格式为: `teqc +ma file1>file2`

关闭多路径效应的命令格式为: `teqc -ma file1>file2`

2) 多路径误差的均值设定

L1的期望的多路径误差均值设定命令: `teqc -mp1_rms # file1>file2`

L2的期望的多路径误差均值设定命令: `teqc -mp2_rms # file1>file2`

其中, # 是以厘米为单位的, L1的默认值为50cm, L2的默认值为65cm

3) 平均移动点数

命令: `teqc -mp_win # file1>file2`. 其中 $0<\#<65536$, 默认为50

(10) 信噪比的设定

载波L1上允许的最小信噪比设定: `teqc -min_L1 # FILE1 >FILE2`

载波L2上允许的最小信噪比设定: `teqc -min_L2 # FILE1 >FILE2`

其中: $0<\#<9$, 默认为0

(11) 电离层的有关设定

电离层设定有电离层延迟变化率和电离层延迟误差两个方面。命令分别如下:

1) 电离层延迟。命令: `teqc -ion_jump # file1>file2` 设置电离层延迟的最大值为# cm, 默认为 $1.00e+035$ 。

2) 电离层变化率。命令: `teqc -iod_jump # file1>file2` 设置电离层延迟变化率为# cm, 默认为400cm/m。

(12) GPS采样间隔的设定

在GPS测量中, 一般观测间隔设定为5s或30s, 设定格式如下:

`teqc -O.int # file1>file2` 其中#>0

`teqc -O.st YYYY MM DD HH MM SS -O.dec # file1>file2` (用这个)

(13) 观测文件的信息查询

1) `teqc +config file` 通过此命令可以看到观测文件file 的各项设置。包括: GPS的频道设置, GPS、GLONASS、GLNASS卫星的编号设定, 卫星周等信息。

2) `teqc +meta file` 通过此命令可显示观测文件的名称, 大小, 始末历元, 采样率, 测站名, 站点编号, 天线编号, 天线类型, 天线在大地坐标系中的坐标, 天线高, 接收机编号, 接收机类型, 坐标系等信息。

(14) 各种GPS接收机、天线类型的查询

我们可以通过命令 `teqc +igs`得到目前全世界范围内最流行的GPS接收机类型及天线类型。

(15) TEQC 软件输出中的符号说明

在teqc程序中, 我们常可以看到各种各样的符号。而通过命令`teqc ++sym` 可以查看各种符号代表的意义。

(16) 帮助信息

我们要想熟练使用任何软件, 帮助信息必不可少, 它可以帮助我们快速掌握我们所需的知识。在TEQC中弹出帮助信息的命令如下:

`teqc +help`

通过此命令, 我们不但可以看到teqc软件中的全部命令, 而且他有英文注释, 可以知道各具体命令的功能。

12.2.3 质量检核

TEQC应用软件的数据质量检查功能可以处理静态或动态双频GPS和GLONASS导航定位系统的接收数据。只有单点数据并且包含广播星历信息才能进行数据质量检查, 主要利用了伪距和载波相位观测值的线性组合方法。

根据是否利用导航文件信息, TEQC分为qc2lite和qc2full两种检核方式。

(1) qc2lite 方式

如果输入文件只有RINEX 观测数据文件而没有导航数据文件,那么TEQC将会在qc2lite方式下运行。运行teqc + qc trimble.05o, TEQC 则对文件trimble. 05o在qc2lite 方式下进行质量检核。通常在缺省状态下,质量检核的结果会生成报告文件trimble. 05s 和数据文件trimble.ion (电离层延迟误差)、trimble. iod (电离层延迟变化率)、trimble.mp1 (L1载波C/A码或P码伪距的多路径影响)、trimble.mp2 (L2载波P码伪距的多路径影响)、trimble.sn1 (L1载波的信噪比)、trimble.sn2 (L2载波的信噪比)。

(2) qc2full 方式。

如果输入文件为RINEX观测数据文件和导航数据文件, 运行teqc +qc -nav trimble.05n trimble.05o; 或者导航数据文件和观测数据文件在同一目录下,则TEQC会自动搜索导航数据文件, 而无需用-nav 指定, 即运行teqc + qc trimble. 05o, 此时TEQC则对文件trimble.05o 在qc2full 方式下进行质量检核。检核的结果除qc2lite 方式下的报告文件和数据文件外, 还增添了卫星和接收机天线的位置信息以及两个数据文件trimble. azi (方位角) 和 trimble. ele (高度角)。

报告文件trimble.05s称之为质量汇总文件, 包括有各个接受卫星的数据质量状态。其中的参数信息有:

1) 信息总结。有各颗卫星的接收数据情况, 观测时间, 文件名, 首历元、末历元, GPS天线在WGS-84中的坐标, 在大地坐标系中的坐标, 采样率, 没捕捉到的卫星数目及编号, 在RINEX中的期望观测值数, 采集百分比, 多路径平均误差, 平均移动点数, 始终漂移, 探测到的周跳数, 卫星两个高度角之间的观测值数目信息。

2) 观测统计量。如平均多路径误差, 总平均高度角, 周跳数目等。

3) QC设置参数。如接收机的通道最大值, 电离层延迟变化率的最大允许值, 期望的平均多路径误差等。

数据文件又成为视图文件, 在GPS数据质量检核中, 除汇总文件外的所有文件, 通称为视图文件。生成的视图文件如下表所示:

表 12-3 GPS 质量检查中生成的视图文件后缀及其含义

文件类型	含义	文件类型	含义
.mp1	L ₁ 载波 C/A 码或 P 码伪距的多路径影响	.iod	电离层延迟变化率
.mp2	L ₂ 载波 P 码伪距的多路径影响	.ion	电离层延迟误差
.sn1	L ₁ 载波的信噪比	.azi	卫星方位角
.sn2	L ₂ 载波的信噪比	.ele	卫星高度角

我们可以利用软件QCVIEW对数据质量检查结果进行分析，使得观测数据的质量在图像上得到直观反映。

12.2.4 各历元伪距单点定位计算

虽然TEQC是一专业的GPS数据预处理软件，但是也可进行各历元伪距的单点定位。计算单点在空间坐标系中的坐标命令格式如下：

```
teqc +qc +eepx trimble.05o>trimblexyz.xyz
```

计算单点在大地坐标系中的坐标命令格式如下：

```
teqc +qc +eepg trimble.05o>trimblebl.xyz
```

注意：虽然在上述命令中没有用到导航文件（.N），但是在所进行操作的目录下必须有导航文件，否则会显示：The instruction at 0x0043fe01 referenced memory at 0x0000000c. The memory could not be read。错误信息，不能运行上述命令。

第 13 章 TEQC 处理实例

在本例中，将对2004年4月14日使用TRIMBLE 4700类型的3台接收机观测1小时的数据cann105a.dat, iryml05a.dat, melb105a.dat进行处理（在此我们仅对cann105a.dat的处理给出详细说明，其他两个观测文件的处理过程与此相同），过程如下：

13.1 格式转换

首先用命令：`teqc -tr do -week 1266 +nav cann105a.04n cann105a.dat >cann105a.04o` 将cann105a.dat文件转换成标准文件RINEX的格式。通过本命令后，将会生成观测文件cann105a.04o和导航文件cann105a.04n。

13.2 质量检查

在转换生成RINEX文件的同时就可以编辑RINEX文件的数据，当然也可以在生成RINEX文件后在进行编辑。在此，由于TEQC的各个模块就独立的，我们就暂时跳过这一步，直接查看观测数据的质量，然后对数据进行适当的编辑。

使用命令`teqc +qc cann105a.04o`进行质量检核。运行此命令后，将会生成cann150a.mp1,cann150a.mp2,cann150a.sn1,cann150a.sn2,cann150a.ele,cann150a.azi,cann150a.iod,cann150a.ion,cann150a.04s等文件。

汇总文件cann150a.04s的有关信息。用word程序打开cann150a.04s，我们可以看到观测文件的名称为cann105a.04o，观测时间段为2004年4月14日00: 00: 00到00: 59: 55.001，接收机接收到的卫星编号及质量，天线在空间坐标系中的坐标x: -4346271.2673 y: 2595234.2754 z:3867368.9447 (m)以及在大地坐标系中的坐标-37.564855 deg 149.157839 deg 158.5777 m，采样的时间间隔为5秒，接收到11颗卫星的信号，没有接收到的卫星编号分别为2 5 6 7 9 10 11 12 14 15 17 18 21 22 23 25 26 28 29 30 32，卫星高度角的限值为10o，观测值总数为7842个，其中高度角大于10o的有5338个，高度角大于10o但由于其他误差超限而被删除的观测值有194个，载波在L1上的多路径误差均值为0.071614 m，在L2上的均值为 0.113825 m，平均移动点数为50个，最大的电离层延迟率为400.00 cm/min，L1的期望多路径平均值为50.00cm，L2的期望多路径平均值为65.00cm，以5o为单位各个高度角范围内的观测值数目分别为：

85 - 90	0	0	0.000000
80 - 85	0	0	0.000000
75 - 80	0	0	0.000000

70 - 75	115	0	0.013890
65 - 70	396	0	0.025700
60 - 65	369	0	0.018057
55 - 60	336	0	0.030205
50 - 55	294	0	0.034133
45 - 50	219	0	0.046327
40 - 45	441	0	0.041733
35 - 40	482	0	0.044777
30 - 35	387	0	0.064248
25 - 30	289	0	0.071027
20 - 25	617	0	0.085342
15 - 20	643	0	0.124734
10 - 15	742	1	0.144828
5 - 10	281	0	0.114023
0 - 5	0	0	0.000000

13.3 视图文件查看

(1) 查看卫星高度角

命令: `qcview cann105a.ele`。图13-1是第19颗卫星的高度角曲线图,从图中我们可以看出它的高度角变化范围为 $0^{\circ}\sim 19^{\circ}$ 之间,图13-2是第13颗卫星的高度角曲线图,从图中得知它的变化范围为 $50^{\circ}\sim 74^{\circ}$ 之间。

(2) 查看卫星方位角

命令: `qcview cann105a.azi`。从图13-3中我们可以看出第13颗卫星的方位角变化范围为 $-140^{\circ}\sim 170^{\circ}$ 之间,而从图13-4中看出第19颗卫星的方位角变化范围为 $40^{\circ}\sim 60^{\circ}$ 之间。

(3) 查看卫星的多路径误差

命令: `qcview cann105a.mp1` (查看载波L1上的多路径误差) 命令: `qcview cann105a.mp2` (查看载波L2上的多路径误差)。从图13-5我们可以看出第19颗卫星在L1上的多路径误差MP1的变化范围为 $-30\sim 50\text{cm}$,从图13-7可以看出第19颗卫星在L2上的多路径误差变化范围为 $-2.5\sim 1.0\text{m}$,说明此颗卫星多路径误差影响较大。从图13-6可以得知第13颗卫星在L1上的多路径误差范围为 $-10\sim 10\text{cm}$,而从图13-8中可得知第13颗卫星在L2上的多路径误差变化范围为 $-0.1\sim 0.1\text{m}$,说明此颗卫星的多路径误差比较小。

(4) 查看电离层延迟误差

命令: `qcview cann105a.ion`。从图13-9可以看出第19颗卫星的电离层延迟误差, 其值在-0.3~3m之间, 并且在第560历元处探测到一个“I”周跳。从图13-10中可以看到第13颗卫星的电离层延迟误差为-0.1~0.3。从中我们不难看出卫星19的电离层延迟误差大于卫星13的电离层误差。

(5) 查看电离层延迟变化率

命令: `qcview cann105a.iod`。从图13-11我们很容易看出第19颗卫星的电离层延迟变化率为-0.9~1.0之间, 在第560个历元处同样也有一个“I”标记。从图13-12可以看出第13颗卫星的电离层延迟变化率为-0.1~0.1之间, 变化幅度相当小。

(6) 查看信噪比

命令: `qcview cann105a.sn1 (sn2)`。第19颗卫星的信噪比在1-6之间, 第13颗卫星的信噪比在8-9之间。

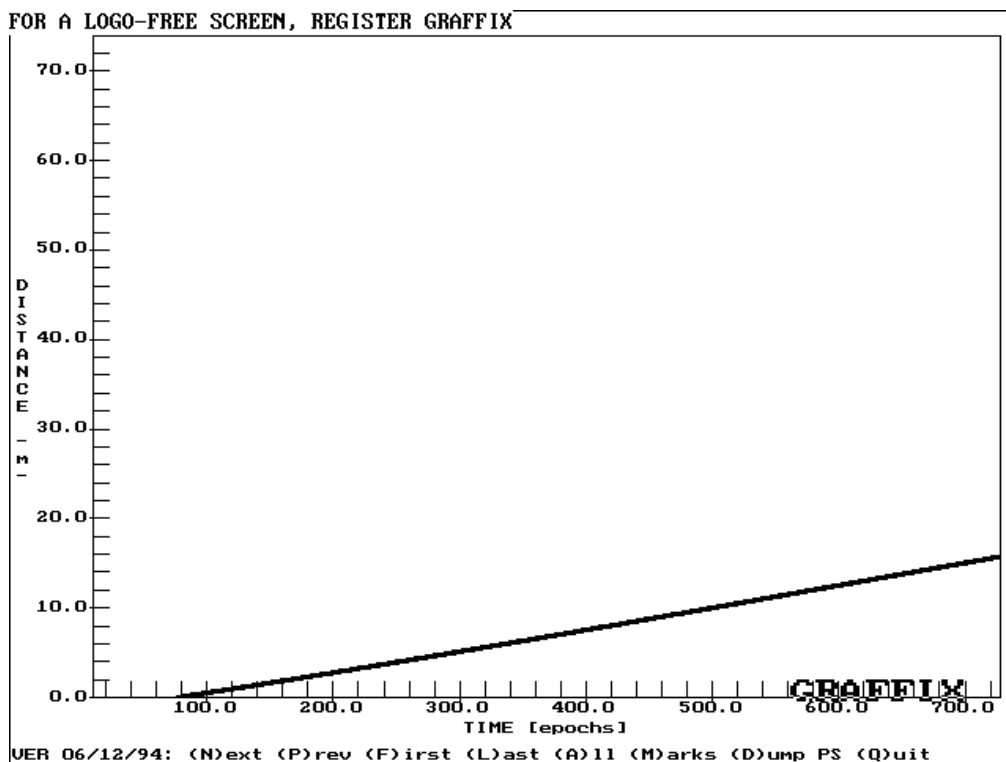


图 13-1 第 19 颗卫星的高度角 (ele)

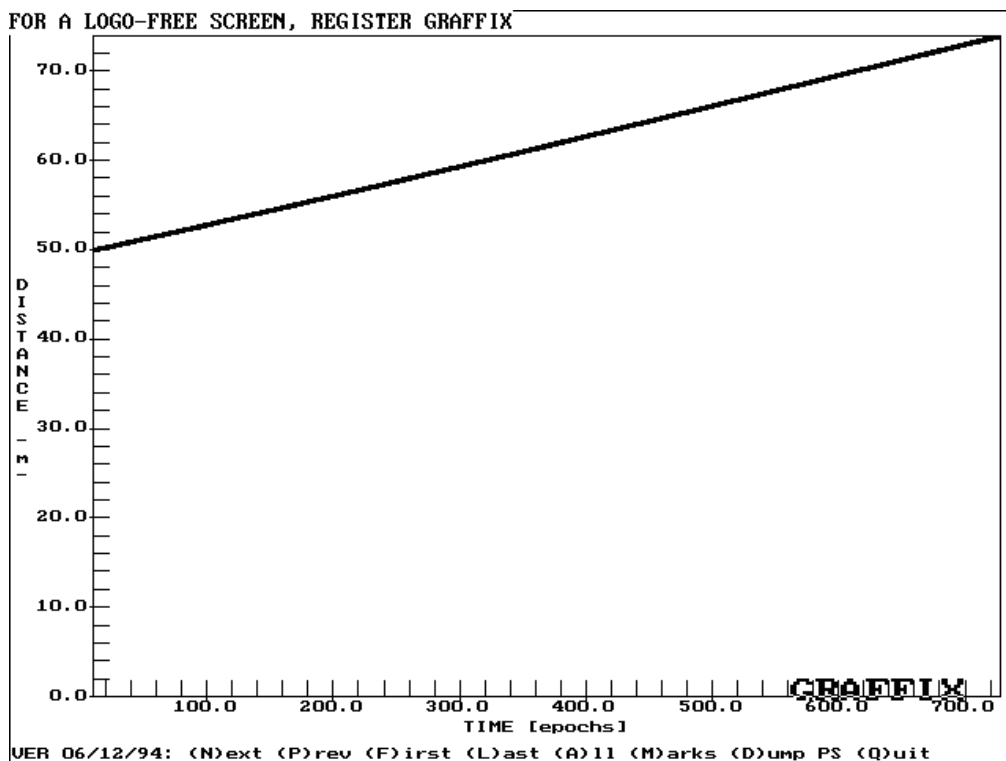


图 13-2 第 13 颗高度角 (ele)

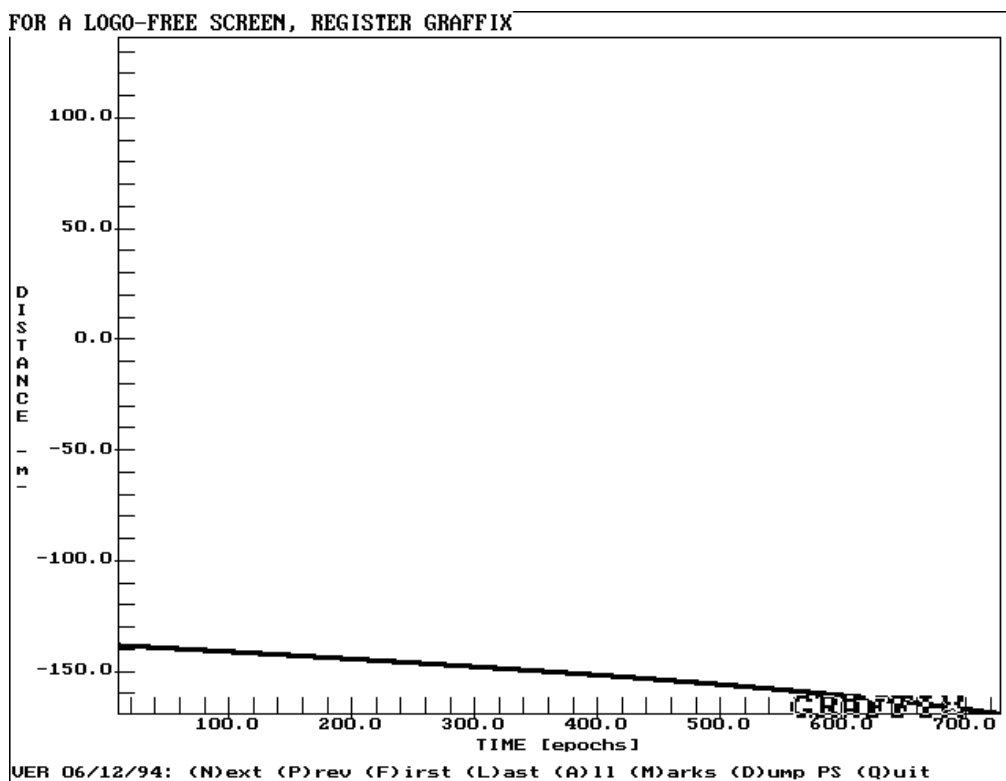


图 13-3 第 19 颗卫星的方位角 (azi)

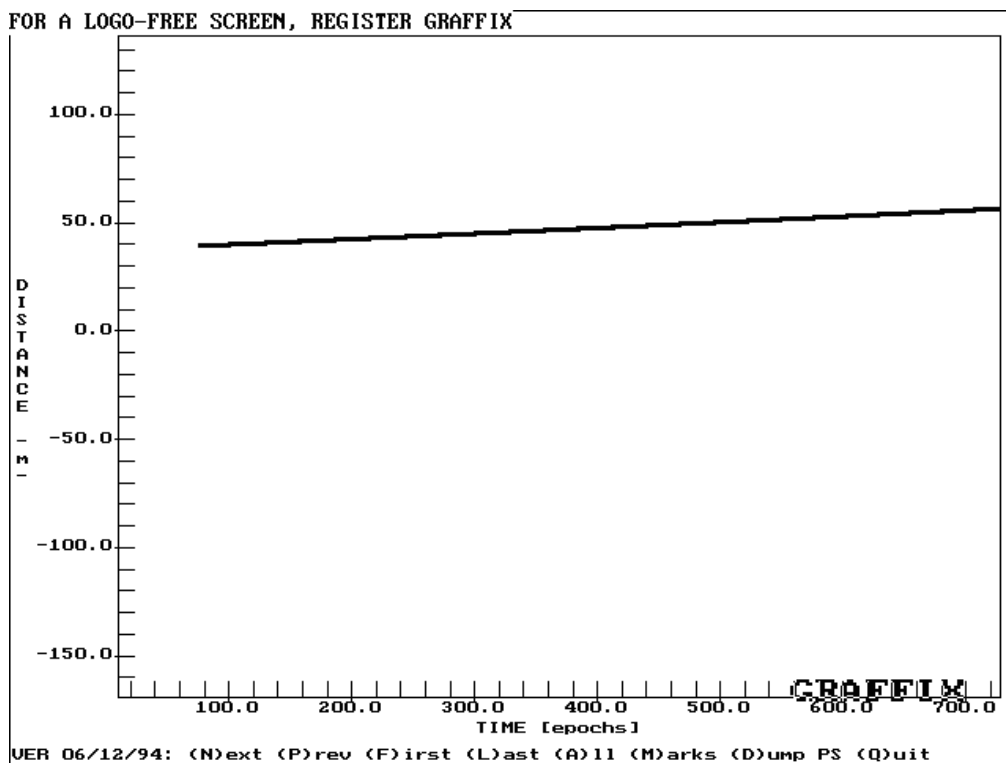


图 13-4 第 13 颗卫星的方位角 (azi)

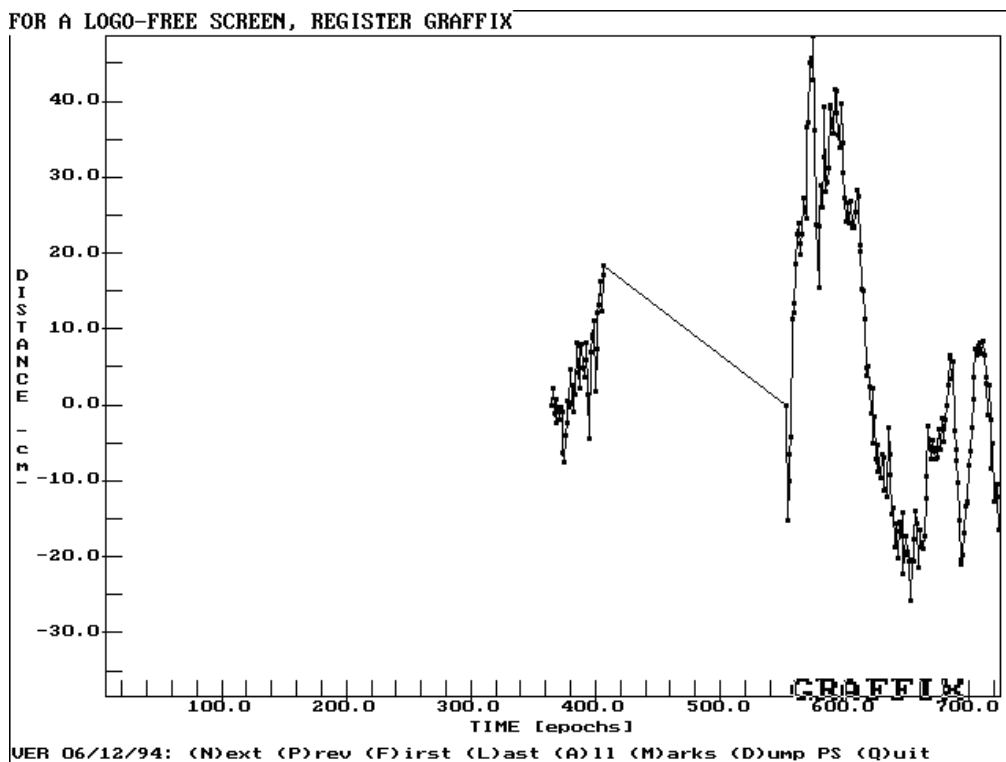


图 13-5 第 19 颗卫星的多路径误差 (mp1)

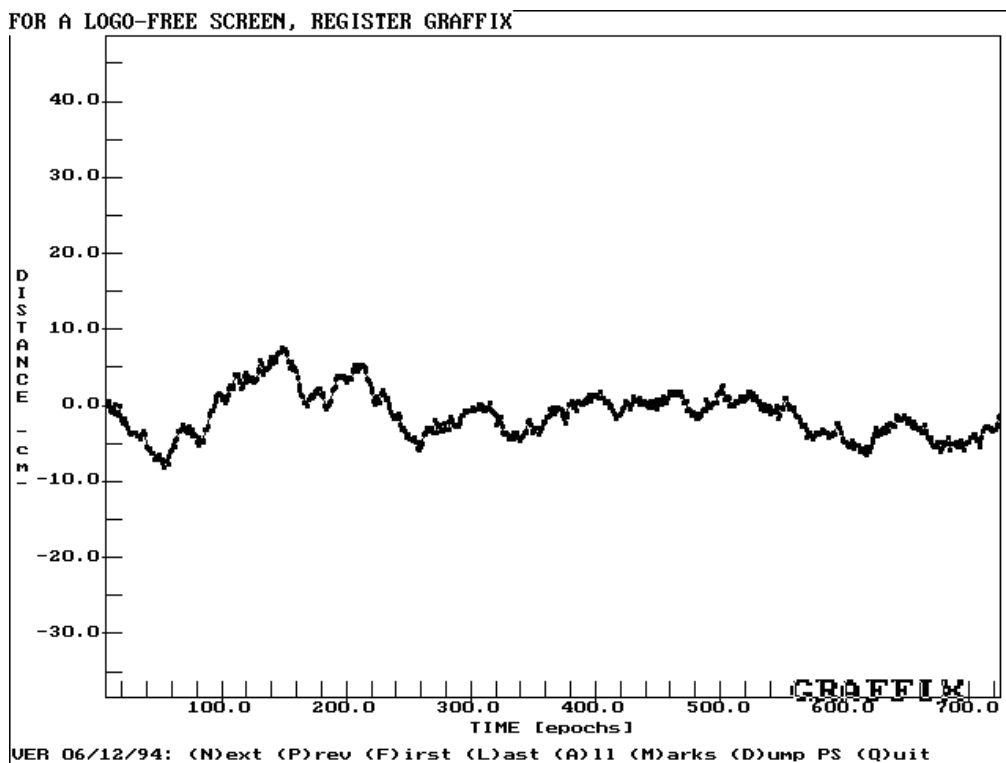


图 13-6 第 13 颗卫星的多路径误差 (mp1)

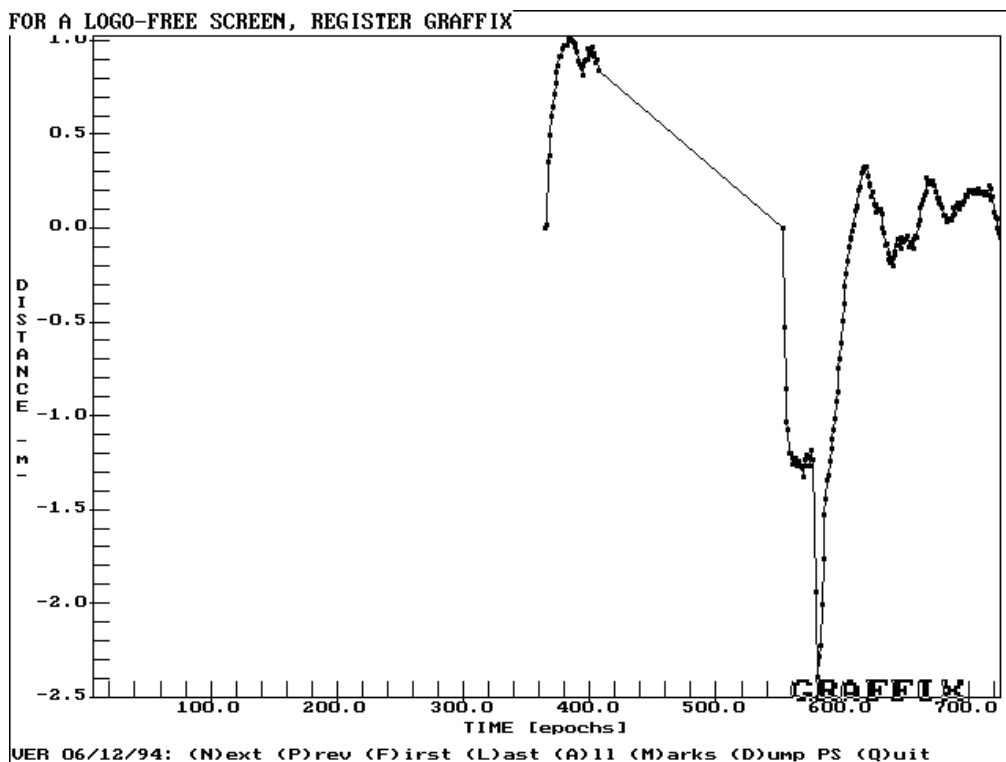


图 13-7 第 19 颗卫星的多路径误差

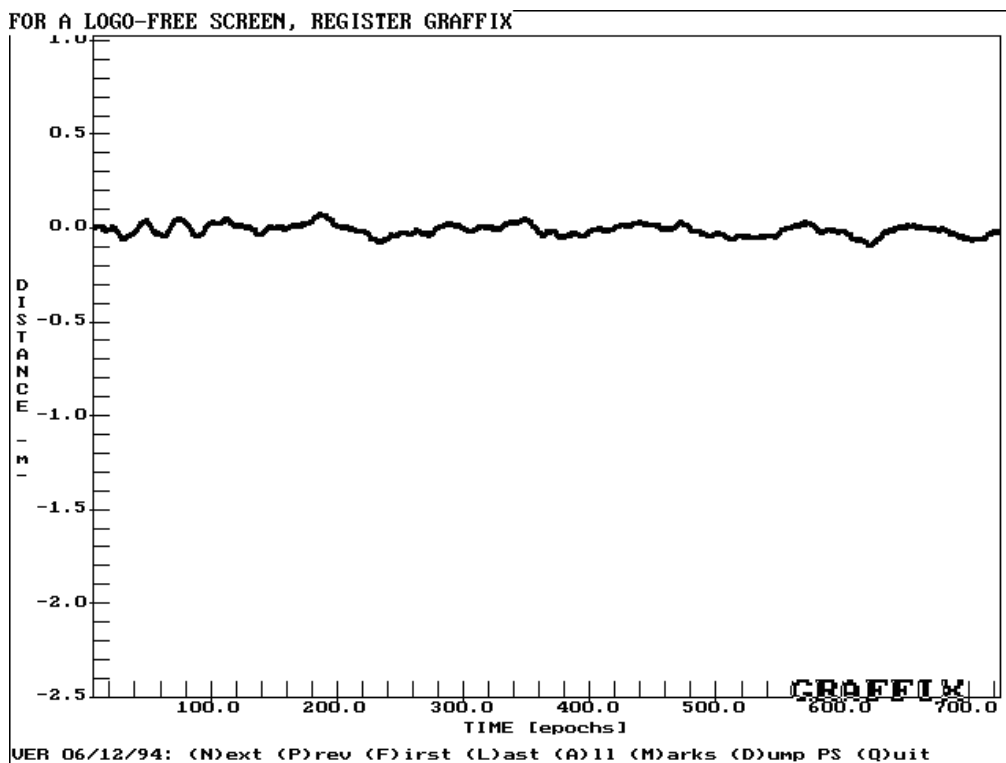


图 13-8 第 13 颗卫星的多路径误差

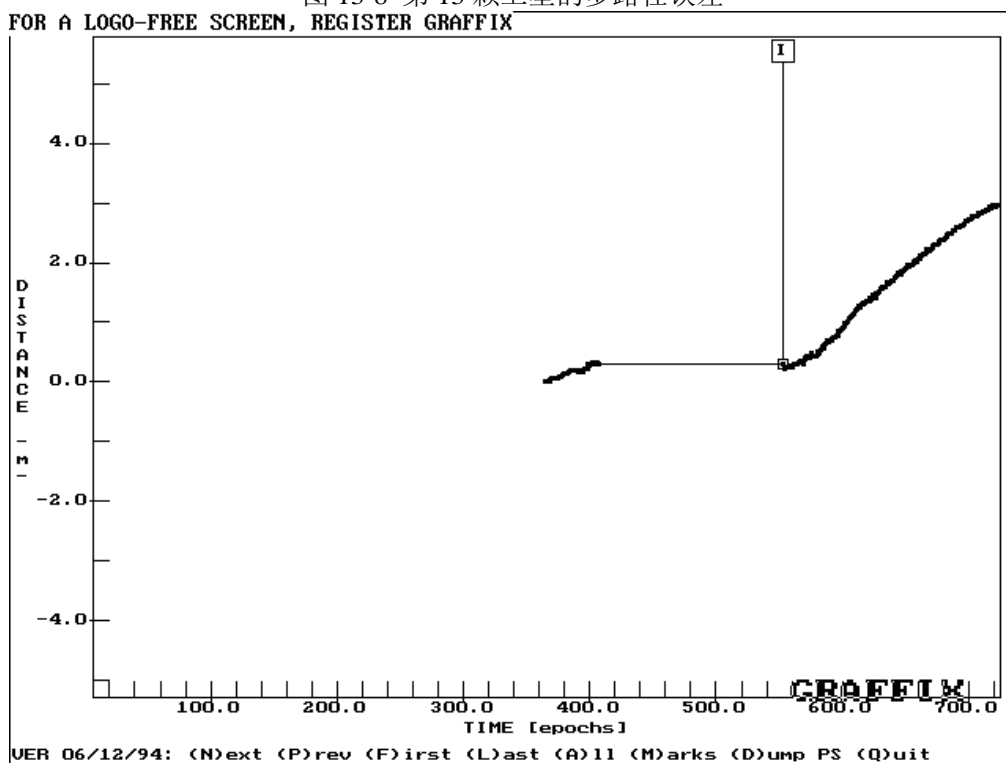


图 13-9 第 19 颗卫星的电离层延迟误差

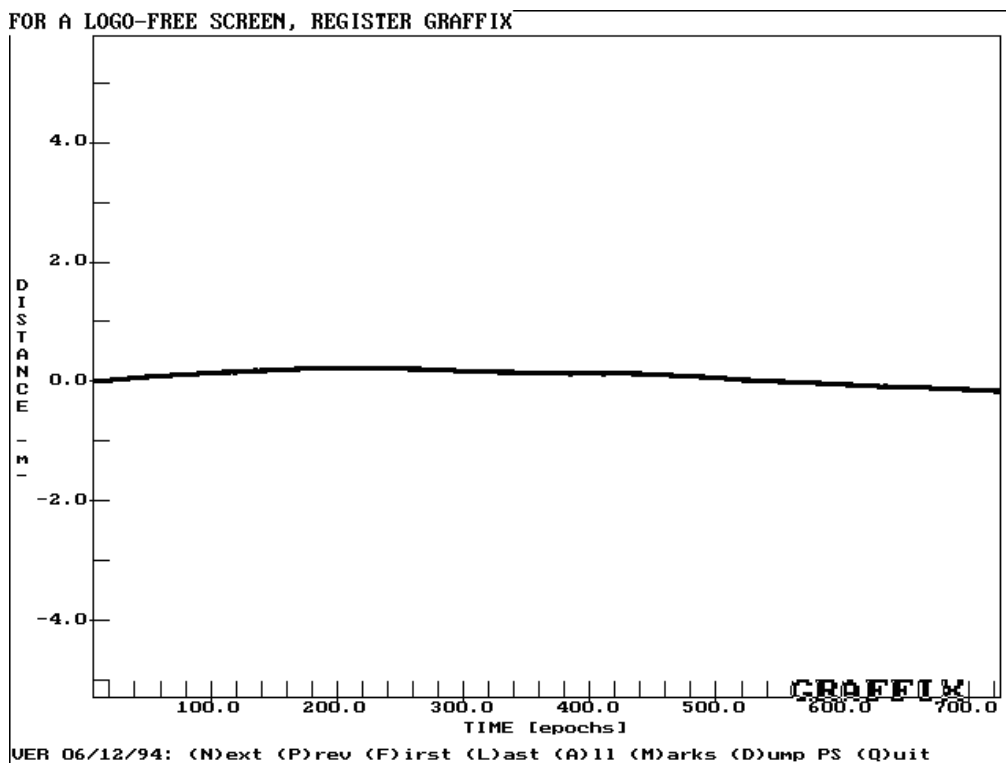


图 13-10 第 20 颗卫星的电离层延迟误差

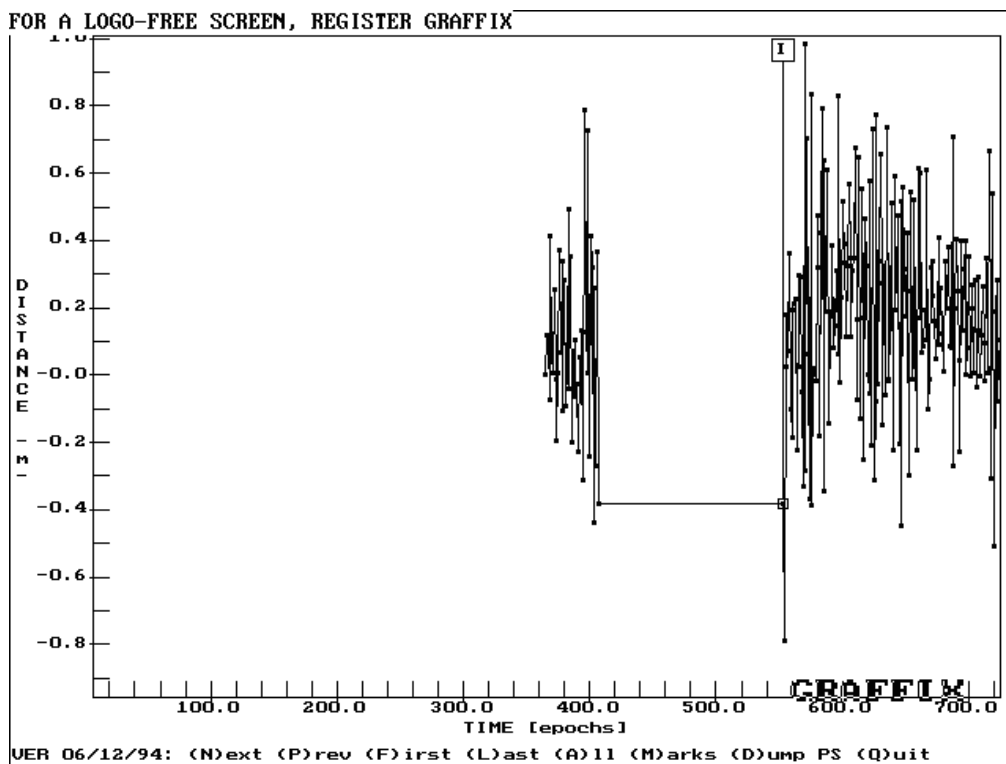


图 13-11 第 19 颗卫星的电离层延迟变化率

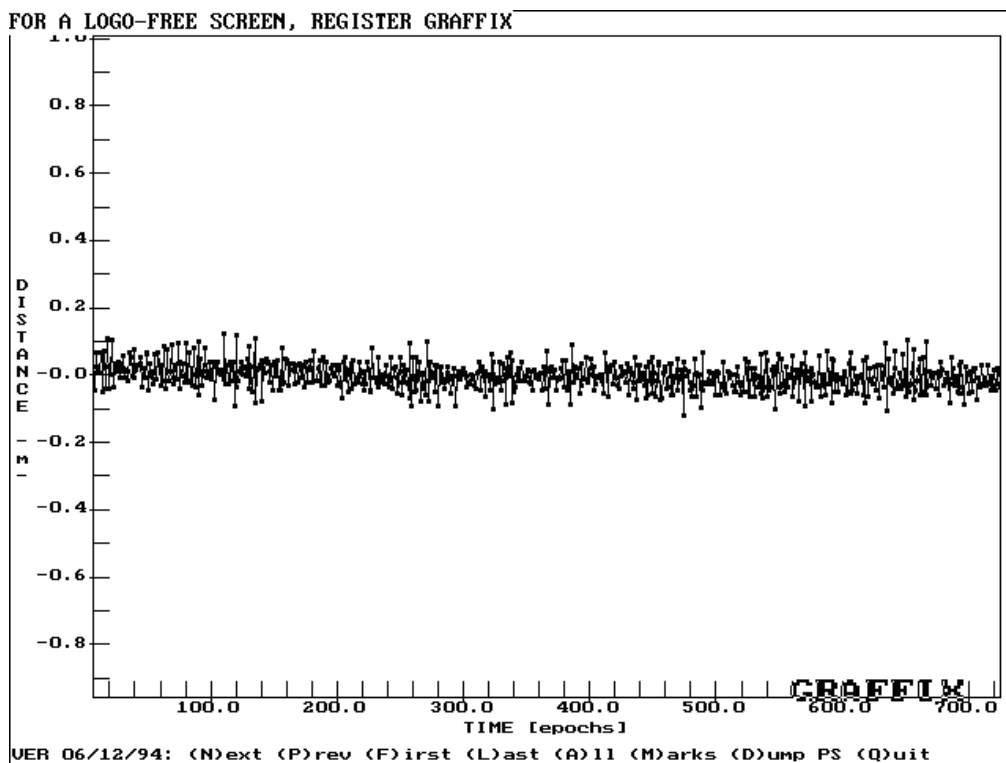


图 13-12 第 13 颗卫星的电离层延迟变化率

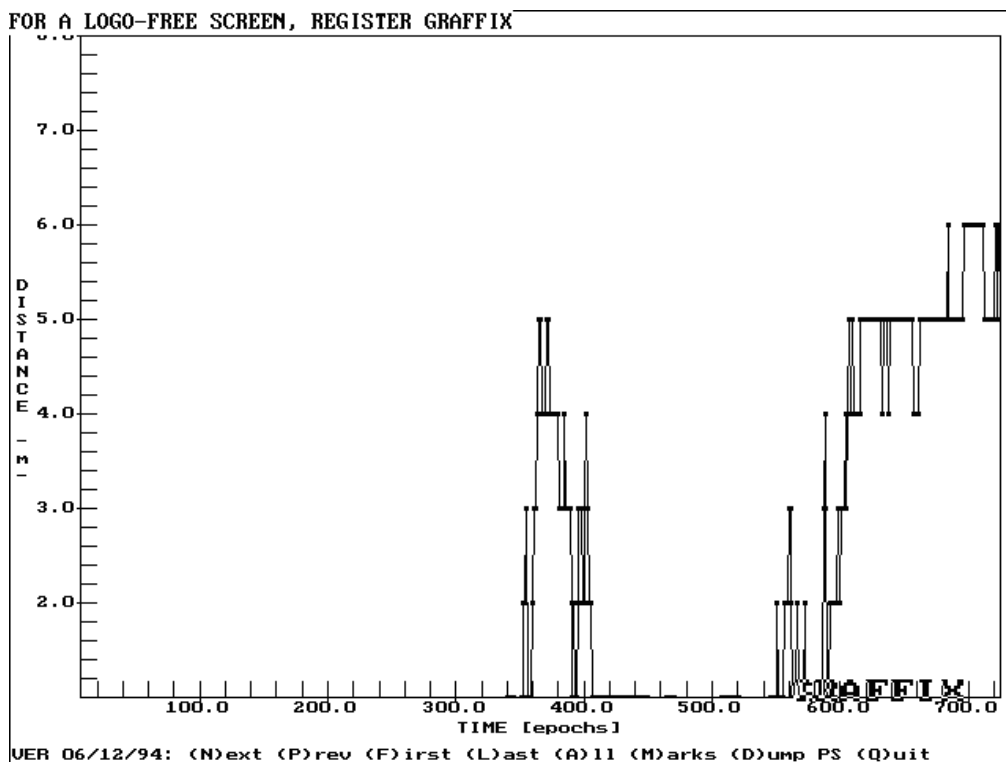


图 13-13 第 19 颗卫星的信噪比

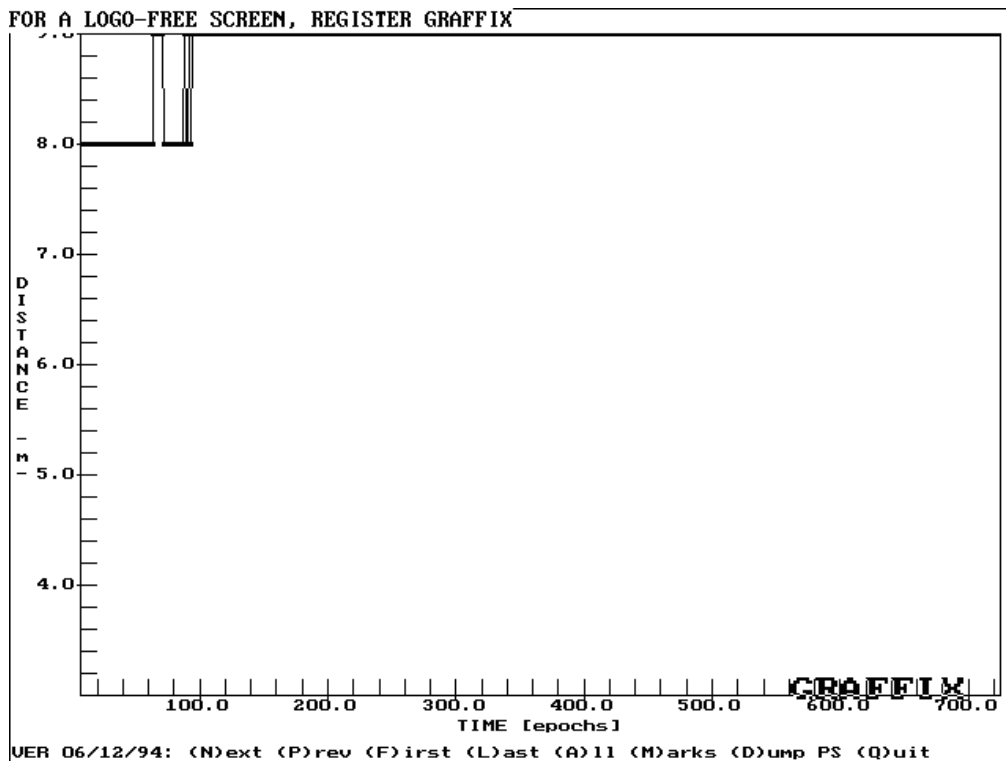


图 13-14 第 13 颗卫星的信噪比

通过对视图文件的分析，我发现下面几个问题：

(1) 卫星的高度角默认设置为 10° ，多路径误差比较大，电离层的影响也比较大。

(2) 接收机刚开始跟踪卫星信号，即刚开机时各种误差影响严重。

(3) 第10颗卫星和第24颗卫星的信号很差，并且极不稳定。

针对以上情况，对数据进行了适当的编辑。

(1) 将卫星的高度角设为 20° 。(teqc -set-mask20 cann105a.04o>cann105.04o)

(2) 删除刚开始5分钟的接收数据。(teqc -stm 5 cann105.04o>cann.04o)

(3) 删除prn10和prn24。(teqc -G10 cann.04o>can.04o,teqc -G24 can.04o>ca.04o) 经过修改后，我们使用命令 fc ca.04s cann105a.04s两个比较汇总文件发现主要有以下不同之处：

Comparing files ca.04S and CANN105A.04S

***** ca.04S

Time of start of window : 2004 Apr 14 00:05:0.000

Time of end of window : 2004 Apr 14 00:59:55.001

Time line window length : 59.83 minute(s), ticked every 10.0 minute(s)

antenna WGS 84 (xyz) : -4346286.3208 2595242.0474 -3867379.2493 (m)
 antenna WGS 84 (geo) : S 37 deg 33' 53.41" E 149 deg 09' 28.26"
 antenna WGS 84 (geo) : -37.564836 deg 149.157851 deg
 WGS 84 height : 178.2633 m
 |qc - header| position : 53 m
 ***** CANN105A.04S
 Time of start of window : 2004 Apr 14 00:00: 0.000
 Time of end of window : 2004 Apr 14 00:59:55.001
 Time line window length : 59.92 minute(s), ticked every 10.0 minute(s)
 antenna WGS 84 (xyz) : -4346271.2673 2595234.2754 -3867368.9447 (m)
 antenna WGS 84 (geo) : S 37 deg 33' 53.48" E 149 deg 09' 28.22"
 antenna WGS 84 (geo) : -37.564855 deg 149.157839 deg
 WGS 84 height : 158.5777 m
 |qc - header| position : 39 m

 ***** ca.04S
 Possible obs > 0.0 deg: 7832
 Possible obs > 10.0 deg: 5630
 Complete obs > 10.0 deg: 5331
 Deleted obs > 10.0 deg: 194
 Moving average MP1 : 0.071605 m
 Moving average MP2 : 0.113879 m
 ***** CANN105A.04S
 Possible obs > 0.0 deg: 7842
 Possible obs > 10.0 deg: 5637
 Complete obs > 10.0 deg: 5338
 Deleted obs > 10.0 deg: 194
 Moving average MP1 : 0.071614 m
 Moving average MP2 : 0.113825 m
 Points in MP moving avg : 50

***** ca.04S

Obs reported w/ code | phase : 6308

Obs deleted (any reason) : 977

Obs complete : 5331

***** CANN105A.04S

Obs reported w/ code | phase : 6315

Obs deleted (any reason) : 977

Obs complete : 5338

***** ca.04S

mean MP1 rms : 0.071612 m

total mean elevation : 36.62 degrees

MP1 obs > 10 : 5331

qc MP1 slips < 25 : 1

***** CANN105A.04S

mean MP1 rms : 0.071622 m

total mean elevation : 36.62 degrees

MP1 obs > 10 : 5338

qc MP1 slips < 25 : 1

***** ca.04S

mean MP2 rms : 0.113879 m

total mean elevation : 36.62 degrees

MP2 obs > 10 : 5331

qc MP2 slips < 25 : 1

***** CANN105A.04S

mean MP2 rms : 0.113825 m

total mean elevation : 36.62 degrees

MP2 obs > 10 : 5338

qc MP2 slips < 25 : 0

13.4 各历元伪距单点定位计算

我们通过命令 `teqc +qc +eepx cann105a.04o>cann105xyz.xyz` 计算出未进行编辑时测站cann点在空间直角坐标系中的坐标，并在EXCEL中求出它的平均值为（-4346277.783，2595238.651，-3867373.914）。利用`teqc +qc +eepg cann105 .04o>cann105abl.xyz` 计算出测站cann点在大地坐标系中的坐标（S 037°33'53.451099"，E 149°09'28.203263"，167.820）。再计算编辑后文件ca.04o的坐标，在空间直角坐标系中的坐标为（-4346270.789，2595248.651，-3867373.414，），在大地坐标系中的坐标为（S 037°33'53.451059"，E 149°09'28.203413 "，167.823）。把处理后的结果列入表13-1中。

表13-1 对cann105a.04o、irym105a.04o和melb105a.04o用不同软件处理的结果比较

处理软件	测站点	X(m)	Y(m)	Z(m)
TEQC	Cann	2595238.650	-4346277.783	-3867373.914
	Irym	3236564.599	-4171282.264	-3566683.873
	Melb	2896656.465	-4131288.167	-3888703.177
HDS2003	Cann	2595265.165	-4346254.633	-3867371.714
	irym	3236588.625	-4171260.776	-3566685.849
	melb	2896679.742	-4131265.001	-3888699.608
差值	Cann	-26.515	-23.150	-2.2
	Irym	-24.026	-21.884	2.024
	Melb	-23.277	-23.166	-3.569