**RTKlib关于高精度GPS动态定位处理过程**

[第一章引言 3](#_Toc385755352)

[1.1调用主函数main（rnx2rtkp.c） 3](#_Toc385755353)

[1.2调用后处理函数postpos（postpos.c） 3](#_Toc385755354)

[1.3 处理基站信息execses\_b（postpos.c） 3](#_Toc385755355)

[1.4 处理流动站信息 execses\_r（postpos.c） 4](#_Toc385755356)

[1.5执行处理操作execses（postpos.c） 4](#_Toc385755357)

[1.6函数调用流程图 4](#_Toc385755358)

[第二章文件读取 5](#_Toc385755359)

[2.1观测文件读取readobsnav (postpos.c) 5](#_Toc385755360)

[2.1.1 文件头读取redarnxh (rinex.c) 5](#_Toc385755361)

[2.1.2 文件的记录数据读取readrnxobs (rinex.c) 6](#_Toc385755362)

[2.2导航电文文件读取 7](#_Toc385755363)

[2.2.1 文件头读取 7](#_Toc385755364)

[2.2.2 文件的记录数据读取 7](#_Toc385755365)

[第三章计算基准站位置和速度 8](#_Toc385755366)

[3.1利用导航文件与基准站观测文件求卫星位置、速度和卫星钟钟差 satposs(ephemeris.c) 8](#_Toc385755367)

[3.1.1卫星钟钟差计算 ephclk(ephemeris.c) 8](#_Toc385755368)

[3.1.2 卫星位置计算satpos(ephemeris.c) 9](#_Toc385755369)

[3.2 码伪距单点定位estpos(pntpos.c) 10](#_Toc385755370)

[3.3函数调用流程图 11](#_Toc385755371)

[第四章动态相对定位求流动站位置 12](#_Toc385755372)

[4.1 码伪距单点定位求流动站的近似坐标 pntpos 12](#_Toc385755373)

[4.2 载波相位动态相对定位relpos(rtkpos.c) 12](#_Toc385755374)

[4.2.1 利用导航文件和流动站观测文件求卫星位置和卫星钟钟差satposs(ephemeris.c) 12](#_Toc385755375)

[4.2.2 求基准站对应的非差残差项zdres(rtkpos.c) 12](#_Toc385755376)

[4.2.3 实时状态更新udstate(rtkpos.c) 13](#_Toc385755377)

[4.2.4 求流动站对应的非差残差项zdres(rtkpos.c) 14](#_Toc385755378)

[4.2.5 求双差残差项ddres(rtkpos.c) 14](#_Toc385755379)

[4.2.6 卡尔曼滤波filter(rtkcmn.c) 17](#_Toc385755380)

[4.2.7 模糊度整数估计 resamb\_LAMBDA() 18](#_Toc385755381)

[4.3 函数调用流程图 20](#_Toc385755382)

[第五章总结 21](#_Toc385755383)

[5.1结果输出 21](#_Toc385755384)

[5.2 不足之处 21](#_Toc385755385)

[5.3 下一阶段计划与安排 21](#_Toc385755386)

# 第一章引言

精密GPS动态测量采用载波相位差分技术，其标准测量模式为，一台GPS接收机置于已知点，作为基准站来进行静态测量，另一台GPS接收机置于载体上，作为流动站来进行动态测量。两台接收机同步观测相同的卫星，然后将两台GPS接收机的观测值进行组合处理，就可以获得流动站相对于基准站的坐标和速度。

本文主要介绍用RTKLIB 实现精密动态定位的过程，该过程包括观测文件和导航文件的读取、基准站位置的计算、流动站位置的求解、运行结果的输出，下面将分块讨论。

## 1.1调用主函数main（rnx2rtkp.c）

读一个大型程序首先要找到主函数，rtklib的主函数位于文件rnx2rtkp.c中，主函数中主要定义了一些变量的缺省值。比如定位处理选项、解决方案输出选项等。

## 1.2调用后处理函数postpos（postpos.c）

输入文件包括观测文件、导航文件、精密星历文件等，postpos 在处理输入文件时有两种方法，一种是输入文件可以只包含关键词，然后通过函数处理，将关键词用时间、基准站编号、流动站编号等代替；另一种是直接调用输入文件的文件名，postpos主要是来判断是哪一种输入方式，然后调用相应函数。

## 1.3 处理基站信息execses\_b（postpos.c）

精密GPS动态测量，需要利用基站信息进行差分定位，**execses\_b**

就是用来处理基站信息.

## 1.4 处理流动站信息execses\_r（postpos.c）

execses\_r函数是用来判断调用输入文件时是调用的关键词还是直接使用了文件名，然后分别进行处理。

## 1.5执行处理操作execses（postpos.c）

从execses正式开始执行处理操作，这其中包括文件函数readobsnav，数据的处理函数antpos、procpos等。

## 1.6函数调用流程图

main（rnx2rtkp.c）

postpos（postpos.c）

execses\_b（postpos.c）

execses\_r（postpos.c）

execses（postpos.c）

antpos

(postpos.c)

procpos

(postpos.c)

readobsnav(postpos.c)

注：方框里的字母表示调用的函数，方框里括号内字母表示调用函数所在文件。

# 第二章文件读取

## 2.1观测文件读取readobsnav (postpos.c)

## 2.1.1 文件头读取redarnxh (rinex.c)

RINEX格式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”. 文件头中，每一记录的第61~80列为该行记录的标签，用于说明第1~60列中所表示的内容。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。

RTKLIB中文件头读取的函数调用过程见图2-1。

redarnxh (rinex.c)

redarnxfp (rinex.c)

readrnxfile (rinex.c)

readrnxt (rinex.c)

readobsnav (postpos.c)

图2-1 文件头读取的函数调用过程

### 2.1.2 文件的记录数据读取readrnxobs (rinex.c)

RINEX格式的记录数据紧跟在文件头的后面，在观测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。

在观测值文件中，所记录的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。

RTKLIB中文件头读取的函数调用过程见图2-2。

readobsnav(postpos.c)

readrnxobs (rinex.c)

readrnxfp (rinex.c)

readrnxfile (rinex.c)

readrnxt (rinex.c)

图2-2文件记录数据的函数调用过程

## 2.2导航电文文件读取

### 2.2.1 文件头读取

导航电文文件文件头的格式与观测文件头基本相同，但包含内容略有不同。导航电文文件的文件头存放有文件创建日期、单位名及其他一些相关信息，另外，还有可能会包含电离层模型的参数以及说明GPS时与UTC间关系的参数和跳秒等。

导航文件的文件头读取的函数调用过程与观测文件文件头读取流程相同，见图2-1.

### 2.2.2 文件的记录数据读取

导航电文文件中存放的是所观测卫星的钟差改正模型参数及卫星轨道数据等。由于广播星历每2h更新一次，因此，在导航电文文件中可能会出现某颗卫星具有多个不同参考时刻钟差模型改正参数和轨道数据的情况。

GPS导航电文文件数据记录节中的内容为按卫星和参考时刻存放的各颗卫星的时钟和轨道数据，每颗卫星一个参考时刻的数据占8行，第1行为卫星的PRN号和该卫星钟的参考时刻及其改正模型参数，第2~8行为该卫星的广播轨道数据。

导航文件的文件记录数据读取的函数调用过程与观测文件文件头读取流程基本相同，只是在调用最后一个函数时，导航电文文件调用的是函数readrnxnav。

# 第三章计算基准站位置和速度

GPS提供了三种观测量：伪距、载波相位和多普勒频移。本章主要讨论利用伪距单点定位的方法求取基准站接收机的位置。基准站位置计算调用的函数主要是antpos函数，然后再由antpos函数调用其他函数。

用antpos函数求基站位置有三种方法，第一种是用伪距单点定位的方法求取，调用的是avepos函数；第二种是直接从位置文件中读取，调用的是getstapos函数；第三种是直接从rinex的文件头中读取。本文中采用的是第一种方法。

由于观测文件是以历元的方式存取数据，所以在avepos中要逐历元调用函数pntpos(pntpos.c)处理数据。

Pntpos函数是处理单点定位的函数，该函数包含了卫星位置和钟差计算（satposs）、接收机位置和速度计算（estpos/estvel）等过程。

## 3.1利用导航文件与基准站观测文件求卫星位置、速度和卫星钟钟差satposs(ephemeris.c)

### 3.1.1卫星钟钟差计算ephclk(ephemeris.c)

1、求取卫星信号发射时刻

（3-1）

式中，表示第颗星，表示第个测站，此处只有一个测站即基准站，为P1码的伪距观测值，为光速，表示以接收机钟为时间基准的卫星信号接收时刻，表示以卫星钟为时间基准的卫星信号发射时刻。

2、用广播星历求取卫星钟钟差

（3-2）

（3-3）

（3-4）

式中，为卫星钟的参考时刻，分别为卫星钟偏差、卫星钟的漂移、卫星钟的漂移速度，为卫星钟钟差，卫星钟钟差需经多次迭代求得。

### 3.1.2 卫星位置计算satpos(ephemeris.c)

1、计算观测瞬间卫星的平近点角

（3-5）

（3-6）

（3-7）

式中，为卫星钟的GPS时，为星历的参考时刻，为万有引力常数与地球总质量之积，、为广播星历给出的参数。

2、计算偏近点角

（3-8）

解上述方程需用迭代法（也可用微分迭代法）。

3、计算升交距角、卫星矢径、卫星轨道倾角

（3-9a）

（3-9b）

（3-9c）

（3-10a）

（3-10b）（3-10c）

4、计算卫星在轨道面坐标系中的位置

（3-11a）

（3-11b）

5、计算卫星在ECEF坐标系中的位置

（3-12）

（3-13a）

（3-13b）

（3-13c）

注：卫星速度可由卫星位置差分求得。

## 3.2码伪距单点定位estpos(pntpos.c)

1、求伪距残差（即误差方程自由项）rescode(pntpos.c)

伪距原始观测方程为：

（3-14）

式中，为卫星到基准站的几何距离，为接收机钟差，为卫星钟钟差，为对流层延迟，为电离层延迟，为观测噪声。

将（3-14）线性化后可得：

（3-15）

式中，，为坐标改正项，，，。

将含未知量的项移到等式右边，重组后有

（3-16）

式中含有四个未知量。

令伪距残差，可得误差方程式

（3-17）

式中，



2、最小二乘法求方程解lsq(rtkcmn.c)

（3-18）

式中，为观测值权矩阵，可由对应的每颗星的观测误差求得。

由（3-18）可求得基准站坐标



## 3.3函数调用流程图

antpos(postpos.c)

avepos(postpos.c)

pntpos(pntpos.c)

estvel

estpos

satpos

# 第四章动态相对定位求流动站位置

在动态相对定位中，参考站A的接收机固定在基线向量已知的点上，流动站接收机是移动的，需要确定其在任意历元的位置。本章主要介绍相对定位的处理过程，调用的函数是procpos(postpos.c.)。

在函数procpos中，首先调用函数inputobs（postpos.c）函数来输入每一历元的观测数据和星历数据，然后再调用rtkpos(rtkpos.c)函数来进行精密定位。

下面主要讲述在函数rtkpos中处理数据的具体流程。

## 4.1 码伪距单点定位求流动站的近似坐标pntpos

码伪距单点定位求流动站坐标的过程与第三章所述过程基本相同，只是观测文件改变，并且对于每一个历元需求一个流动站位置。

## 4.2 载波相位动态相对定位relpos(rtkpos.c)

### 4.2.1利用导航文件和流动站观测文件求卫星位置和卫星钟钟差satposs(ephemeris.c)

该过程与3.1中的卫星位置计算过程相同，只是改变了观测文件。

### 4.2.2 求基准站对应的非差残差项zdres(rtkpos.c)

该残差项其实就是基于伪距、载波相位的数学模型，不过是表达形式有点差异。简而言之，残差对于各类误差之和，载波相位需要加上模糊度。（公共部分，一同减去不影响 效果同双差一样）

分析：

接收机用上一秒的定位结果进行泰勒公式展开，此时增量（X, Y, Z）（或者可理解为基线向量）为相对前一定位结果的，最后通过kalman滤波得到残差值， 然后加上上一秒的定位结果，组成当前时刻的位置估计。

（4-1）

（4-2）

（4-3）

（4-4）

式中，、分别表示载波L1、L2对应的观测量，、分别为码观测量，为真实距离，基准站以基准站坐标展开，移动站以前一位置坐标展开，为卫星时钟误差，为对流层误差，、对应L1、L2的天线相位改正。

### 4.2.3 实时状态更新udstate(rtkpos.c)

1、实时位置更新udpos(rtkpos.c)

由4.1每个历元都会利用码伪距单点定位计算出一个流动站的近似位置，再根据近似位置经过载波差分定位求出流动站的高精度坐标。

分析：

这里更新的坐标是用单点定位还是？

利用固定解更新

怎么更新位置的？

根据mode状态

固定解直接更新

否则重新初始化浮点解x, 浮点解误差矩阵P, 观测误差矩阵Q

2、周跳探测

文中主要采用电离层残差法进行周跳探测，过程如下。

组合观测量为：

（4-5）

上式消除了接收机至卫星的几何距离、轨道误差、接收机钟差、卫星钟差和对流层延迟误差，仅与电离层延迟、组合模糊的和观测值噪声有关。它与载体的运动状态无关，可用于静态或动态测量中零差、单差或双差载波相位观测值的周跳探测。在没有周跳时，它随时间变化缓慢。一旦有周跳产生，它就会产生比较显著的变化。因此其相邻历元的差值可以用来检测周跳，其差值为：

（4-6）

式中，。

分析：

周跳产生做什么？

在udbias中有三种不同的周跳检测方式

如果周跳检测 然后重置x, P, slip flag

3、码相位组合求模糊度的浮点解udbias(rtkpos.c)

GPS伪距、载波相位和多普勒观测量表示如下：

（4-7a）

（4-7b）

（4-7c）

（4-7d）

对（4-7a）、（4-7c）分别进行站间差分可得：

（4-8）

（4-9）

将（4-8）、（4-9）进行码相位组合可得：

（4-10）

分析：

代码中/\* estimate approximate phase-bias by phase - code \*/处为上述理论的实现，phase-code = (Pu - Pr) - (Cu - Cr) 为同一颗卫星的移动站与基准站间的单差整周模糊度。存储在浮点解rtk->x中。该模糊度估算存在误差。

### 4.2.4 求流动站对应的非差残差项zdres(rtkpos.c)

（4-11a）

（4-11b）

（4-11c）

（4-11d）

式中，、分别表示载波L1、L2对应的观测量，、分别为码观测量，为上一次的定位结果的坐标到卫星的几何距离，、对应L1、L2的天线相位改正。

### 4.2.5 求双差残差项ddres(rtkpos.c)

双差即不同测站，同步观测同一组卫星，所得的单差之间的差值。双差可以消除接收机钟差。另一方面，双差放大了多路径效应和观测噪声的影响。

1、双差方程（以L1为例）

这里只考虑两颗卫星信号频率相同的情况，则双差方程的最终形式为：

（4-12）

式中，。

2、观测方程线性化

（4-13）

其中，

















式中，对应参考星，对应参考站，参考站的坐标已知，为状态向量，为状态向量的方差矩阵（在程序中用P表示）。

因模糊度的浮点解已在4.2.3中求出，则可求得双差残差项

（4-12a）

（4-12b）

（4-12c）

（4-12d）

式中，对应参考星，对应参考站，、分别表示载波L1、L2对应的双差残差项，、分别表示码观测量P1、P2对应的双差残差项。

实例中，设有7颗星，其中以2号星为参考星，以L1为例，可得出6个双差残差项，即，则的方差协方差矩阵为

（4-13）

式中，，

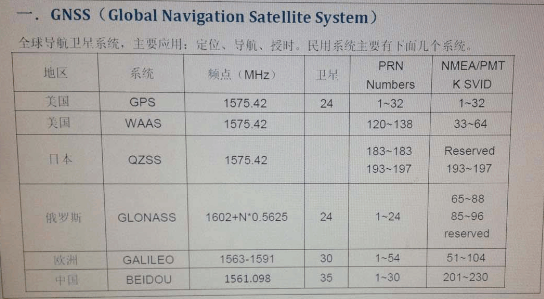
对应参考星，对应卫星的高度角，为载波相位观测误差，为基线向量观测误差，为卫星钟稳定误差。

系数矩阵和状态向量分别为：





式中，=2表示2号星为参考星，为状态向量的方差矩阵（在程序中用表示）。



分析：

双差模型后得到 双差残差v， 双差系数矩阵H， 双差误差矩阵R，这里的未知量只剩增量向量X（x,y,z），双差模糊度已经有单差模糊度近似取代。 如果用LS算法，那么X = (H’RH)-1H‘Rv

如何重构站间单差模糊度 N1 N2…

### 4.2.6 卡尔曼滤波filter(rtkcmn.c)

1、模型方程为：

（4-14）

式中，为系统误差向量，为观测误差向量，，。

实例中，令，量测方程转移矩阵，双差后的残差，，为单位阵。

则有：

（4-15）

其中，表示一步预测误差矩阵

滤波增益（4-16）

（4-17）

（4-18）

式中，对应浮点解，对应浮点解的误差矩阵（当前预测误差矩阵）。

**卡尔曼滤波所用残差v推导**

载波相位数学模型如下：

 （1）

（2）

其中u表示接收机，上标表示卫星1，公式2表示接收机的真实距离在上一秒的位置处进行泰勒展开（相关知识见GPS原理书5.1）。

同理，因为基准站位置已知，不用进行泰勒展开。基准站处的载波相位观测方程如下：

 （3）

在RTKLIB代码中（1）（3）进行了调整，得到对应的残差和为

 （4）

 （5）

同理，对于2号卫星，接收机和基准站对应的残差和为

 （6）

 （7）

对同一颗卫星做站间单差，消除电离层误差，简化有

 （8）

 （9）

其中，和为站间单差整周模糊度。

将残差 做卫星之间的差分，消除接收机和基准站的时钟误差，有

（10）

此时，对于这个来说，我们之前在udstate中利用伪距和载波相位对其已经进行估计，很明显存在误差，那么可以将表示如下：

 （11）

结合10，11我们可以有

 （12）

当有多颗卫星时，上述结果对应代码中的双差残差，对应系数矩阵H，对应未知增量x（此增量是针对前一秒定位的结果以及当前估计的模糊度）。

对于卡尔曼滤波更新公式如下：

 （13）

其中为当前最优估计结果，为当前预测的结果，代码中直接用上一秒的定位结果，为当前的测量值，KF存在量测方程，其中是量测噪声，为当前测量值得到的定位结果，同理，这个值我们可以进行展开为上一秒的定位结果加上当前的增量。则对于公式13有

 （14）

x\_ = x,y,z,有效的站间单差模糊度估计值

-1587089.99560321, 5322413.84545247, 3126227.79664455, -18.3281043171597, -73.7176714346778, -18.2135421726326, -15.7314269142388, -14.4238176986859, -26.3969825490687, -69.7602759123231, -7.49752390453184, -79.3411857922184, -66.7036085872296, -1.30599081027147, -7.82328151486858, 5.59379709840775, -12.366050687374, -56.7115940974637, -22.6879778689011

H\_ 19x28 = 方向向量，L1，…，L2

0.955132532183699, -0.0708452098735328, 0.59549669063765, -0.190293672798365, 0, 0, 0, 0, 0, 0.190293672798365, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.983757575480886, 0.380890643608387, -0.290152665795868, 0, -0.190293672798365, 0, 0, 0, 0, 0.190293672798365, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.22200628771231, -0.384882707728713, 0.574627737504544, 0, 0, -0.190293672798365, 0, 0, 0, 0.190293672798365, 0, 0, 0, 0, 0, 0, 0, 0, 0, -0.0599416908959336, -0.715169148364887, 0.310300831843648, 0, 0, 0, -0.190293672798365, 0, 0, 0.190293672798365, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1.29345215687083, 0.148517910914663, 0.180100196001269, 0, 0, 0, 0, -0.190293672798365, 0, 0.190293672798365, 0, 0, 0, 0, 0, 0, 0, 0, 0, -0.122417026093709, -0.174565690795578, -0.407598458569021, 0, 0, 0, 0, 0, -0.190293672798365, 0.190293672798365, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1.46342237768432, -0.403076244718337, 0.457606520544013, 0, 0, 0, 0, 0, 0, 0.190293672798365, -0.190293672798365, 0, 0, 0, 0, 0, 0, 0, 0, 0.955132532183699, -0.0708452098735328, 0.59549669063765, 0, 0, 0, 0, 0, 0, 0, 0, -0.244210213424568, 0, 0, 0, 0, 0, 0.244210213424568, 0, 0.983757575480886, 0.380890643608387, -0.290152665795868, 0, 0, 0, 0, 0, 0, 0, 0, 0, -0.244210213424568, 0, 0, 0, 0, 0.244210213424568, 0, 0.22200628771231, -0.384882707728713, 0.574627737504544, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -0.244210213424568, 0, 0, 0, 0.244210213424568, 0, -0.0599416908959336, -0.715169148364887, 0.310300831843648, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -0.244210213424568, 0, 0, 0.244210213424568, 0, 1.29345215687083, 0.148517910914663, 0.180100196001269, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -0.244210213424568, 0, 0.244210213424568, 0, -0.122417026093709, -0.174565690795578, -0.407598458569021, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -0.244210213424568, 0.244210213424568, 0, 1.46342237768432, -0.403076244718337, 0.457606520544013, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.244210213424568, -0.244210213424568, 0.955132532183699, -0.0708452098735328, 0.59549669063765, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

P\_ 19x19 浮点解误差矩阵

900, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 900, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 900, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 115.674486557307, 112.471292601211, 114.24612362019, 110.211044870914, 113.73524960261, 109.068656891163, 113.229421079661, 111.363724778943, 2.46995241737116, -0.0219472000886708, 1.36104715575801, -1.78293044462258, 0.962967104436258, -2.67324518017753, 0.568788565958419, -0.884632418635069, 0, 0, 0, 112.471292601211, 116.361642492496, 110.566410608073, 108.884329806632, 114.3468041971, 113.094380011034, 113.640473248813, 110.634667036639, -0.021947148441051, 3.00531310281931, -1.50625396500112, -2.81672346045964, 1.4395188411577, 0.463705459526855, 0.889104235004132, -1.45271706460618, 0, 0, 0, 114.24612362019, 110.566410608073, 115.141552525983, 113.079081900659, 111.543502849874, 111.059408787682, 113.861182918805, 110.502736790734, 1.36104675076443, -1.50625425659143, 2.05455600416012, 0.451931487369516, -0.74487153924181, -1.12198542259784, 1.06109171302698, -1.55551473688996, 0, 0, 0, 110.211044870913, 108.88432980663, 113.07908190066, 117.586382775167, 109.822342277872, 114.477090214077, 110.961532282491, 114.97819587419, -1.78292997485299, -2.81672197574639, 0.451929581042061, 3.95813160096767, -2.08579116175891, 1.54139827131818, -1.19813380269663, 1.93211746172701, 0, 0, 0, 113.73524960261, 114.346804197101, 111.543502849873, 109.822342277872, 114.587321148157, 110.710543604754, 111.823662688644, 113.430573632989, 0.962966617148872, 1.43951834402189, -0.744871623195388, -2.08579214461807, 1.62265914321611, -1.39382477347348, -0.526585591188486, 0.725930028088556, 0, 0, 0, 109.068656891165, 113.094380011035, 111.059408787682, 114.477090214077, 110.710543604752, 117.04330958746, 113.429592326439, 111.117018579391, -2.6732444915387, 0.463704320762454, -1.12198544911712, 1.54140007552057, -1.39382459737721, 3.53579995951369, 0.724894238401597, -1.07674405616528, 0, 0, 0, 113.229421079661, 113.640473248813, 113.86118291880

### 4.2.7 模糊度整数估计resamb\_LAMBDA()

整周模糊度的固定可以提高基线向量解的精度。在载波相位数据处理中，使用双差而非单差非常重要，因为在双差观测方程中，钟差参数已被消除，因此模糊度分离成为可能。整周模糊度成功解算的另一个重要因素是卫星的几何构型，在这里不做讨论。

整周模糊度的解算主要有三个步骤。第一步是从算法角度考虑生成所有可能的整周模糊度组合；第二步是搜索出正确的模糊度组合；第三步是模糊度整数解的检验。

下面主要讲述最小二乘模糊度降相关平差法即lambda算法求解整周模糊度。

1、对4.2.6中所得结果进行变换得

（4-19）

（4-20）

式中，



（4-21）

为状态向量的协方差矩阵，再对做变换得、，对应状态向量中模糊度浮点解，为状态向量中模糊度的协方差矩阵，为状态向量中非模糊度参数与模糊度的协方差矩阵。

2、求模糊度的整数解

对进行Cholosky分解，即

（4-20）

式中，，，是单位下三角矩阵,但非对角元不是整数，对取整为，做变换

（4-20）

再对做Cholosky分解，将分解的单位下三角矩阵取整，用表示，再做变换（4-21）

如此循环，直至分解的单位下三角矩阵取整后为单位阵，得到

（4-22）

取（4-23）

令，则，从而，得出模糊度的固定解。

3、计算整周模糊度条件下的非模糊度参数解

将整周模糊度回代，计算模糊度条件下的非模糊度参数解：

（4-24）

（4-25）

式中，对应状态向量中模糊度浮点解，为状态向量中模糊度间的协方差矩阵，为状态向量中非模糊度参数与模糊度的协方差矩阵，

为非模糊度参数的浮点解。

4、修复单差模糊度

由（4-21）可知，向量表示的是双差模糊度解，将已解算出的整周模糊度回代



若已知参考星所对应的单差模糊度即，便可计算出每颗星对应的单差模糊度解。

浮点解：

55.3895092510936, -0.108399536491934, -2.59801817241568, -3.92005522087558, 8.06384907723277, 51.4363180254271, -10.8705769214131, -12.6373934414555, -78.029368125056, -71.5208237195768, -84.9464109210185, -66.9786954667188, -22.6266737447331, -56.6835873714078, 0, 0, 0, 0, 0

固定解：

55.0000000000002, 4.2632564145606E-14, -1.99999999999875, -3.99999999999995, 8.00000000000182, 50.9999999999999, -10.0000000000002, -12.9999999999998, -78,

-70.9999999999996, -84.9999999999999, -66.9999999999995, -22.9999999999996, -56.0000000000005,

## 4.3 函数调用流程图

procpos(postpos.c)

rtkpos(rtkpos.c)

relpos(rtkpos.c)

zdres(rtkpos.c)

udstate(rtkpos.c)

zdres(rtkpos.c)

ddres(rtkpos.c)

filter(rtkpos.c)

resamb\_LAMBDA

satposs(ephris.c)

# 第五章总结

## 5.1结果输出

结果输出分为两部分，第一部分是输出文件头，在函数execses中调用outhead(postpos.c)函数；第二部分是输出数据的处理结果，在函数procpos(postpos.c)中调用函数outsol(solution.c)。

输出结果依次为历元所对应的时间、坐标、坐标方差与协方差、差分龄期、模糊度测试比率。

## 5.2 不足之处

文中只是讲述了RTKlib用于动态定位的整个流程，许多细节问题未予深入探讨，比如周跳探测的方法、电离层延迟改正、对流层延迟改正均未详细讨论，双频载波观测值的线性组合也未涉及，整周模糊度的解算也只涉及到LAMBDA一种算法，要做的工作还很多，要学的东西还很多。

## 5.3 下一阶段计划与安排

下一阶段，会在前面工作的基础上，改进不足，对于周跳探测、模糊度解算的方法研究上多下功夫，同时在RTKlib与北斗的结合应用上多下功夫。