

**《GNSS原理与应用》**

**课间实习报告**

**学 院: 遥感信息工程学院**

**班 级: 20F10**

**学 号: 2020302131201**

**姓 名: 常耀文**

**指导教师: 付建红**

**2022年11月6日**

**目录**

[1. 周跳检测概述 2](#_Toc118638415)

[2. 问题解决描述 2](#_Toc118638416)

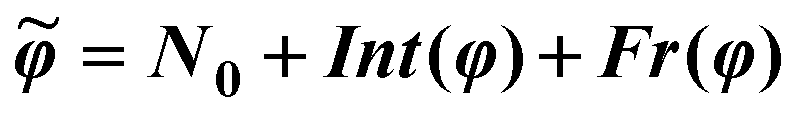
[3. 编程实现思路框图 3](#_Toc118638417)

[4. 具体代码 3](#_Toc118638418)

[5. 实验结果 5](#_Toc118638419)

# 周跳检测概述

在某一特定时刻的载波相位观测值为:

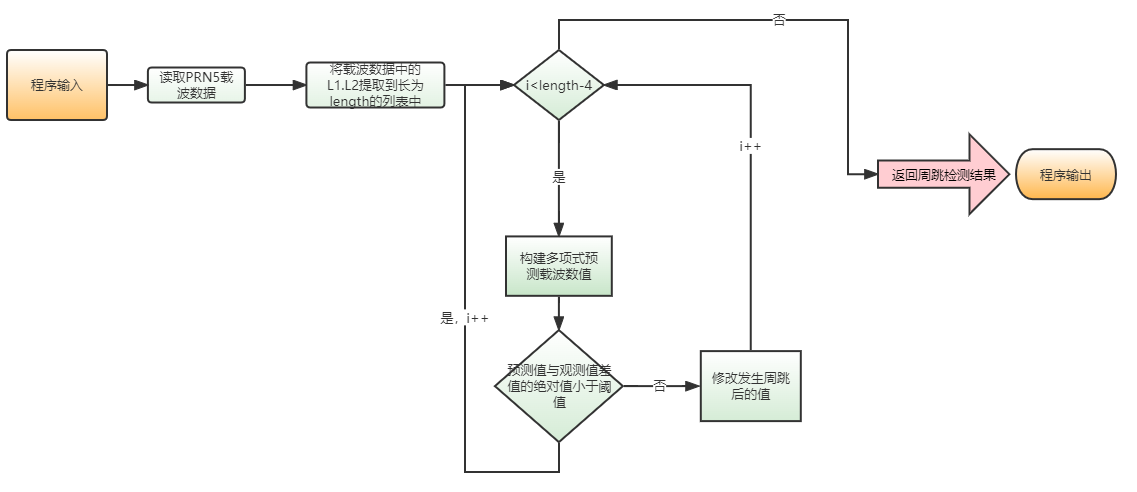


整周计数出现系统偏差，而不足一周的部分仍然保持正确的现象称为整周跳变。

# 问题解决描述

本次实验采用多项式拟合的方法检测，根据 n 个连续相位测量观测值拟合一个 n 阶多项式，据此多项式来预估下一个观测值并与实测值比较，从而来发现周跳并修正整周计数。本次实验采用4阶多项式，利用连续的4个观测值进行拟合多项式。

# 编程实现思路框图



# 具体代码

本次实现代码采用python实现，具体的代码展示如下：

import numpy as np

def solve\_fit(A, X):

   AN = np.array(A)

   TAN = np.linalg.inv(AN)

   XN = np.array(X)

   fit = np.dot(TAN, XN)

   return fit

def cacluate(fit):

    B = np.array([1, 5, 25, 125])

    prediction = np.dot(B, fit)

    return prediction

L1 = []

L2 = []

file = open("PRN5\_HaveCycleSlips.txt", "r")

lines = file.readlines()

for line in lines:

    temp = line.split()

    temp1 = temp[1]

    temp2 = temp[2]

    L1.append(temp1)

    L2.append(temp2)

file.close()

flt\_L1 = [float(x) for x in L1]

flt\_L2 = [float(x) for x in L2]

float\_L1 = np.array(flt\_L1)

float\_L2 = np.array(flt\_L2)

length = len(float\_L1)   # 这里因为L1与L2的长度是相同的所以长度就用L1

i = 0

flag1 = []

flag2 = []

deltaL1=[]

deltaL2=[]

L1cycles\_slips\_value = []

L2cycles\_slips\_value = []

A = [[1,1,1,1],

    [1,2,4,8],

    [1,3,9,27],

    [1,4,16,64]]

while i < (length-4):    #多项式拟合求解的过程

    tempL1 = [[float\_L1[i]],

              [float\_L1[i+1]],

              [float\_L1[i+2]],

              [float\_L1[i+3]]]

    fit1 = solve\_fit(A, tempL1)

    prediction1 = cacluate(fit1)

    deltaL1.append(float\_L1[i + 4] - prediction1)

    if abs(float\_L1[i+4]-prediction1) < 1:

       i += 1

    else:

        print("L1检测到周跳")

        flag1.append(i+5)  #存储周跳的位置

        L1cycles\_slips\_value.append(-(float(float\_L1[i + 4] - prediction1)))

        float\_L1[i+4:length] = (float\_L1[i+4:length]-(float\_L1[i+4]-prediction1)).tolist()

        i += 1

i = 0

while i < (length - 4):  # 多项式拟合求解的过程

    tempL2 = [[float\_L2[i]],

              [float\_L2[i + 1]],

              [float\_L2[i + 2]],

              [float\_L2[i + 3]]]

    fit2 = solve\_fit(A, tempL2)

    prediction2 = cacluate(fit2)

    deltaL2.append(float\_L2[i + 4] - prediction2)

    if abs(float\_L2[i+4] - prediction2) < 1:

        i += 1

    else:

        print("L2检测到周跳")

        flag2.append(i+5)  # 存储周跳的位置

        L2cycles\_slips\_value.append(-(float(float\_L2[i + 4] - prediction2)))

        float\_L2[i+4:length] = (float\_L2[i + 4:length] - (float\_L2[i+4] - prediction2)).tolist()

        i += 1

print("L1发生周跳的次数是： ", len(flag1))

print("L1发生周跳的历元数是：", flag1)

print("周跳大小分别为：", L1cycles\_slips\_value)

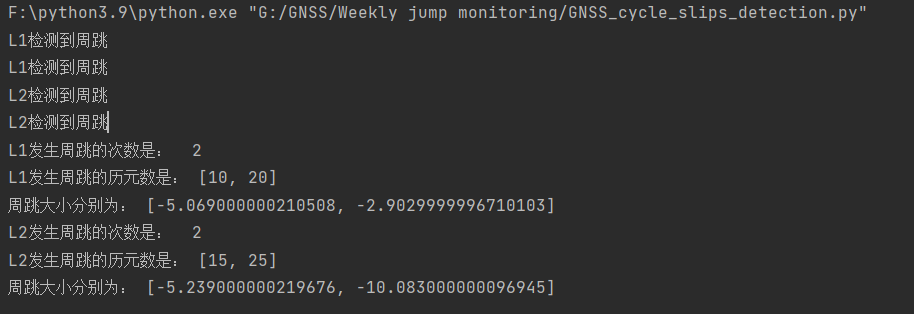
print("L2发生周跳的次数是： ", len(flag2))

print("L2发生周跳的历元数是：", flag2)

print("周跳大小分别为：", L2cycles\_slips\_value)

# 实验结果

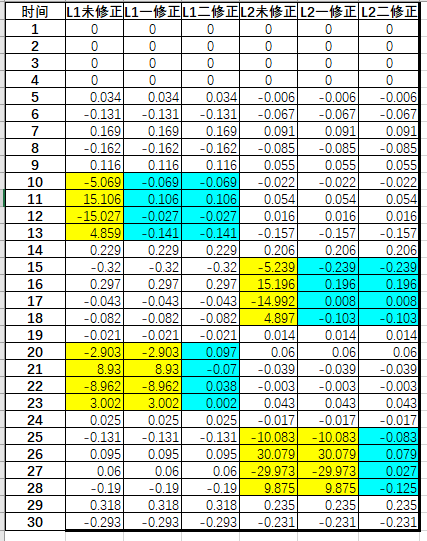
最终周跳检测的结果如下：



可以发现，**L1载波在第10，20个历元处发生周跳，周跳大小可以从代码运行结果中发现，**

**L2载波在第15.，25历元处发生周跳**，大小同样由图可知。

为了具体分析每一处由周跳引起的周跳影响，利用付老师上课讲的excel的方法可以获得相应的改正值，具体展示如下：



黄色表格第一行是首次发生周跳的位置，蓝色是对所有元素改正后的差值，可以发现，excel的结果与代码运行结果对应，验证了结果的正确性。