

地球科学与环境工程学院

《卫星导航定位原理》

课间实验报告书

设计题目： 卫星坐标计算程序

学 号： 2023113911

班 级： 遥感1班

姓 名： 于沛豪

日 期： 2025.4.8

目 录

[一、需求分析 1](#_Toc195117542)

[1.1 设计内容 1](#_Toc195117543)

[1.2 作业需求 1](#_Toc195117544)

[二、实验原理 2](#_Toc195117545)

[2.1坐标计算 2](#_Toc195117546)

[2.2格里高利历至GPS时的换算 4](#_Toc195117547)

[2.2.1 格里高利历至儒勒日的换算 4](#_Toc195117548)

[2.2.2 儒勒日至GPS时的换算 4](#_Toc195117549)

[三、计算程序 5](#_Toc195117550)

[3.1参数读取 5](#_Toc195117551)

[3.2 时间换算 5](#_Toc195117552)

[3.3 常数定义 6](#_Toc195117553)

[3.4 轨道参数计算 6](#_Toc195117554)

[3.5 卫星坐标计算 7](#_Toc195117555)

[3.6 输出与可视化绘图 7](#_Toc195117556)

[四、解算结果 8](#_Toc195117557)

[4.1 解算后坐标 8](#_Toc195117558)

[4.2 可视化绘图 9](#_Toc195117559)

[五、经验总结 9](#_Toc195117560)

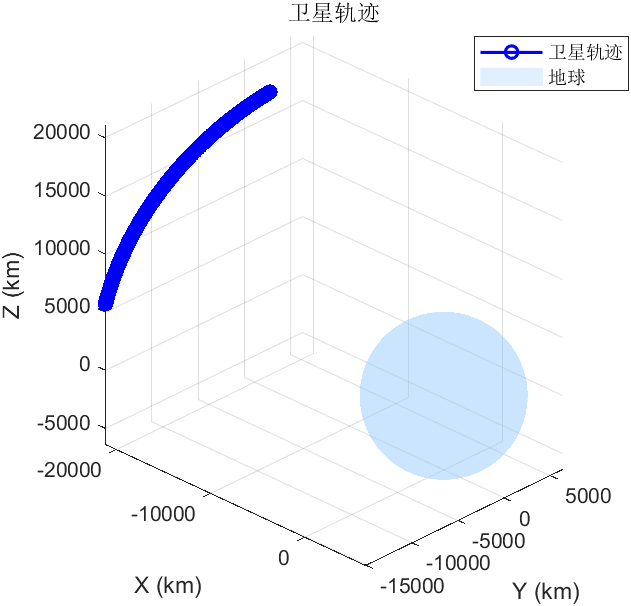
[5.1 小插曲 9](#_Toc195117561)

[5.2 反思与改进 10](#_Toc195117562)

[5.3 总结 10](#_Toc195117563)

# 一、需求分析

1.1 设计内容



**图1 120分钟卫星轨迹**

1.2 作业需求

1. 根据星历文件brdc0010.25p中的数据，编程计算GPS时2025年 1 月1日12点0分00秒 —20分00秒，每隔1分钟的PRN 1（即 G01）的卫星坐标。
2. 撰写卫星坐标计算程序设计报告：包括封面、计算原理、程序 代码、解算结果、经验总结等，以.docx或PDF格式提供
3. 在报告中提供计算结果，列表显示时刻及卫星坐标，也可计算更长的时间段，画图显示

1.2.3环境需求

Windows11 以及MATLAB R2024b。

# 二、实验原理

2.1坐标计算

（1）求轨道长半轴 ：

（2）计算平均角速度 ：

式中，GM 为地球引力常数，其值为：GM＝ 。  
（3）观测时刻 归化到 GPS 时系的计算公式为：

式中： 为相对于参考时刻 的归化时间，但应计及一个星期（共计604800 s）的开始或结束。亦即当 时， 应减去604800 s 。当 时， 应加上 604800 s 。

（4）改正平角速度 ：

（5）计算平近点角 ：

（6）由已知轨道参数按下式计算偏近点角 ：

上述开普勒方程可用迭代法进行解算，即先令 ，代人上式。因为 GPS 卫星轨道的偏心率 约为 0.01 左右，通常进行迭代计算，便可求得偏近点角 。

（7）由下两式计算真近点角 ：

（8）计算升交距角（或称升交角距） ：

式中： 为卫星电文中给出的近地点角距。

（9）计算卫星轨道摄动项改正数：

（10）计算改正后的向径 ：

（11）计算改正后的倾角 ：

上式中， 为轨道倾角变化率， 为相对于参考时刻 的归化时间。

（12）计算观测瞬间升交点的经度 ：



设本周开始时刻（星期日 0 时）格林尼治恒星时为 ，则观测瞬间的格林尼治佰星时为：

式中， 为地球自转角速度，其值为 为一周开始的 GPS 时。有

注意：预报星历中给出的 并不是参考时刻 的升交点赤经 ，而是该值与本周起始时刻的格林尼治恒星时 GAST week 之差。

（13）计算卫星在轨道平面内的坐标 ：

（14）最后，计算卫星在协议地球坐标系中的位置 ：

=(-)()

(-)()=

考虑到地极移动的影响，最后可得到在协议地球坐标系中的空间直角坐标：

=(-)()

(-)()=

设两个点A、B以及坐标分别为A() B()，则 A 和 B 两点之间的距离为:

|  |  |
| --- | --- |
|  | () |

2.2格里高利历至GPS时的换算

2.2.1 格里高利历至儒勒日的换算

格里高利历至儒略日的转换公式可表示如下：

且有如下说明：若 ，则 ；若 ，则 。

式中：JD 为儒略日；为年；为月；为日； 和 s分别为时，分和秒； 为取整函数，有 。

2.2.2 儒勒日至GPS时的换算

GPS 时的起点是 1980年1月6日0时，其对应儒略日为 2444244.5 。故儒略日至 GPS时的转换公式如下：

式中，WN 表示 GPS 时的周数；TOW 表示 GPS 时的周内时间。

# 三、计算程序

3.1参数读取

通过MatLab编写程序，读取G01广播星历参数（参数来自于星历文件brdc0010.25p），代码如下所示：

% 导入广播星历数据（每行含4个参数）

ephemeris = [

  4.500000000000E+01,  1.090625000000E+02,  4.528045754138E-09,  3.131427222067E+00;

  5.554407835007E-06,  2.091019414365E-04,  2.112239599228E-06,  5.153755172729E+03;

  3.024000000000E+05,  2.980232238770E-08, -1.783259133478E+00,  2.980232238770E-08;

  9.596575373827E-01,  3.364062500000E+02, -1.305969830383E+00, -8.338561620276E-09;

  2.750114553310E-10,  1.000000000000E+00,  2.347000000000E+03,  0.000000000000E+00

];

% 解包参数

IODE=ephemeris(1,1);Crs = ephemeris(1,2);Delta\_n =ephemeris(1,3);M0 = ephemeris(1,4);

Cuc =ephemeris(2,1) e   = ephemeris(2,2);Cus     =ephemeris(2,3);sqrt\_a = ephemeris(2,4);

toe =ephemeris(3,1);Cic = ephemeris(3,2);OMEGA   =ephemeris(3,3);Cis    = ephemeris(3,4);

i0  =ephemeris(4,1);Crc = ephemeris(4,2) w       =ephemeris(4,3);OMEGA\_DOT = ephemeris(4,4);

i\_dot = ephemeris(5,1);  % 倾角变化率

3.2 时间换算

通过计算原理部分，先将格里高利历转换为儒勒日，后转换为周内秒

%% 日期转换：

% 格里高利历 -> 儒略日

jd = fix(365.25 \* (year - 1)) + fix(30.6001 \* (month + 13)) + day + 12/24 + 1720981.5;

% 儒略日 -> GPS周及周内秒

gps\_week = fix((jd - 2444244.5) / 7);

tow = mod(jd - 2444244.5, 7) \* 86400;  % TOW (周内秒)

% 时间差（卫星钟参考时间修正）

tk0 = tow - toe;

if tk0 > 302400

    tk0 = tk0 - 604800;

elseif tk0 < -302400

    tk0 = tk0 + 604800;

end

tk = ((0:point\_num) \* 60) + tk0;  % 从tk0开始每分钟1个样本

3.3 常数定义

定义所需常数（书中地球引力常数编写有误，见经验总结部分）

%% 常数

GM = 3.9860047e14;       % 地心引力常数 (m^3/s^2)

We = 7.2921151467e-5;    % 地球自转角速度 (rad/s)

a = sqrt\_a^2;            % 轨道长半轴

n0 = sqrt(GM / a^3);     % 平均角速度

n = n0 + Delta\_n;        % 改正后角速度

3.4 轨道参数计算

通过迭代法解Kepler方程，为保持收敛精度，此处规定1e-9。在真近点角计算时采用atan2函数，不用区分象限。注意摄动改正采用点乘而非乘。

%% 轨道参数计算

Mk = M0 + n \* tk;        % 平近点角

Ek = Mk;                 % 初始化偏近点角

% 解Kepler方程（迭代法）

for iter = 1:10000

    Ek\_next = Mk + e .\* sin(Ek);

    if all(abs(Ek\_next - Ek) < 1e-9)

        disp('偏近点角达到收敛');

        break;

    end

    Ek = Ek\_next;

end

% 真近点角计算

fk = atan2(sqrt(1 - e^2) .\* sin(Ek), cos(Ek) - e);

% 升交角距

phi\_k = fk + w;

% 摄动改正

delta\_u = Cus .\* sin(2 \* phi\_k) + Cuc .\* cos(2 \* phi\_k);

delta\_r = Crs .\* sin(2 \* phi\_k) + Crc .\* cos(2 \* phi\_k);

delta\_i = Cis .\* sin(2 \* phi\_k) + Cic .\* cos(2 \* phi\_k);

% 改正后参数

rk = a \* (1 - e .\* cos(Ek)) + delta\_r;        % 距离

ik = i0 + i\_dot .\* tk + delta\_i;              % 倾角

uk = phi\_k + delta\_u;                         % 升交距角

3.5 卫星坐标计算

使用变量point\_num作为想要计算的点的数量（因为matlab的括号取值同python本身array，均为左闭右开，故此处自加一）

%% 卫星坐标计算

point\_num=point\_num+1;

XYZ = zeros(point\_num, 3);  % 初始化

for i = 1:point\_num

    % 卫星轨道平面坐标

    x\_orb = rk(i) \* cos(uk(i));

    y\_orb = rk(i) \* sin(uk(i));

    % 轨道平面 -> ECEF 转换矩阵

R = [cos(OMEGA\_k(i)),sin(OMEGA\_k(i))\*cos(ik(i)),sin(OMEGA\_k(i))\*sin(ik(i));

     sin(OMEGA\_k(i)),cos(OMEGA\_k(i))\*cos(ik(i)),cos(OMEGA\_k(i)) \* sin(ik(i));

         0          , sin(ik(i))                ,cos(ik(i))];

    % 坐标变换

    XYZ(i,:) = (R \* [x\_orb; y\_orb; 0])';

end

3.6 输出与可视化绘图

通过plot3绘制三维图像，除以1000使坐标轴化为km为单位，控制视窗大小。地球的绘制假想为一个6371km为半径的球体。

disp(XYZ);

%% 可视化轨迹

figure;

plot3(XYZ(:,1)/1e3, XYZ(:,2)/1e3, XYZ(:,3)/1e3, 'b-o', 'LineWidth', 1.5);

grid on;

xlabel('X (km)');

ylabel('Y (km)');

zlabel('Z (km)');

title('卫星轨迹');

axis equal;

view(45, 30);

% 绘制地球

[xe, ye, ze] = sphere(50);

earth\_radius = 6371;

hold on;

surf(earth\_radius\*xe, earth\_radius\*ye, earth\_radius\*ze, ...

    'FaceColor', [0.6 0.8 1], 'EdgeColor', 'none', 'FaceAlpha', 0.3);

legend('卫星轨迹', '地球');

# 四、解算结果

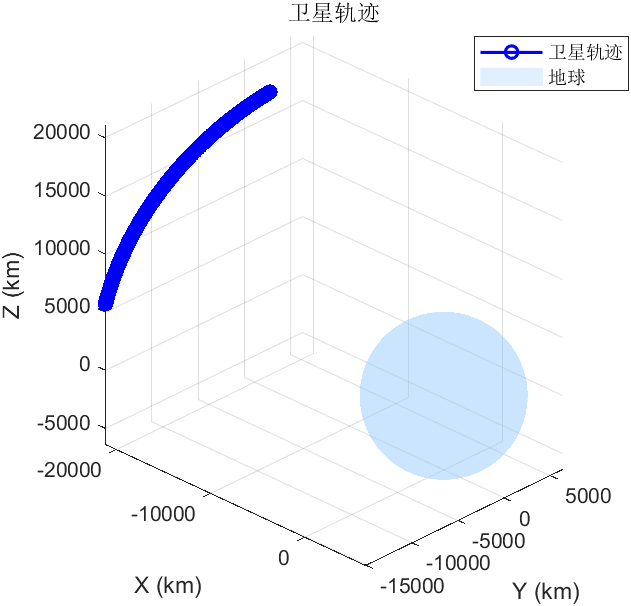
4.1 解算后坐标

**表 1 解算所得XYZ坐标**

|  |  |  |  |
| --- | --- | --- | --- |
| **分钟** | **X** | **Y** | **Z** |
| **0** | -16009488.1292602 | -2472417.97476465 | 21056044.4043865 |
| **1** | -16048742.1192336 | -2627751.77382215 | 21007295.0541760 |
| **2** | -16088432.4469482 | -2782589.40924493 | 20956938.0116368 |
| **3** | -16128552.4365917 | -2936918.20799287 | 20904977.1286531 |
| **4** | -16169095.2770838 | -3090725.58518088 | 20851416.3797379 |
| **5** | -16210054.0232472 | -3243999.04597745 | 20796259.8617306 |
| **6** | -16251421.5970013 | -3396726.18748700 | 20739511.7934857 |
| **7** | -16293190.7885812 | -3548894.70061707 | 20681176.5155521 |
| **8** | -16335354.2577794 | -3700492.37192893 | 20621258.4898429 |
| **9** | -16377904.5352118 | -3851507.08547185 | 20559762.2992966 |
| **10** | -16420834.0236072 | -4001926.82460112 | 20496692.6475285 |
| **11** | -16464134.9991196 | -4151739.67377821 | 20432054.3584731 |
| **12** | -16507799.6126634 | -4300933.82035453 | 20365852.3760175 |
| **13** | -16551819.8912718 | -4449497.55633666 | 20298091.7636256 |
| **14** | -16596187.7394774 | -4597419.28013444 | 20228777.7039530 |
| **15** | -16640894.9407147 | -4744687.49829012 | 20157915.4984532 |
| **16** | -16685933.1587455 | -4891290.82718967 | 20085510.5669747 |
| **17** | -16731293.9391051 | -5037217.99475475 | 20011568.4473486 |
| **18** | -16776968.7105710 | -5182457.84211565 | 19936094.7949677 |
| **19** | -16822948.7866517 | -5326999.32526542 | 19859095.3823570 |
| **20** | -16869225.3670979 | -5470831.51669334 | 19780576.0987341 |

4.2 可视化绘图

因为所给星历文件brdc0010.25p中G01数据为每2小时发布一次广播星历，故可视化绘图采用120min画图显示，若时间段过短，将由于数据过少绘图效果不会太好，若数据过长则由于星历文件参数过时则造成误差过大，故绘图如下所示：

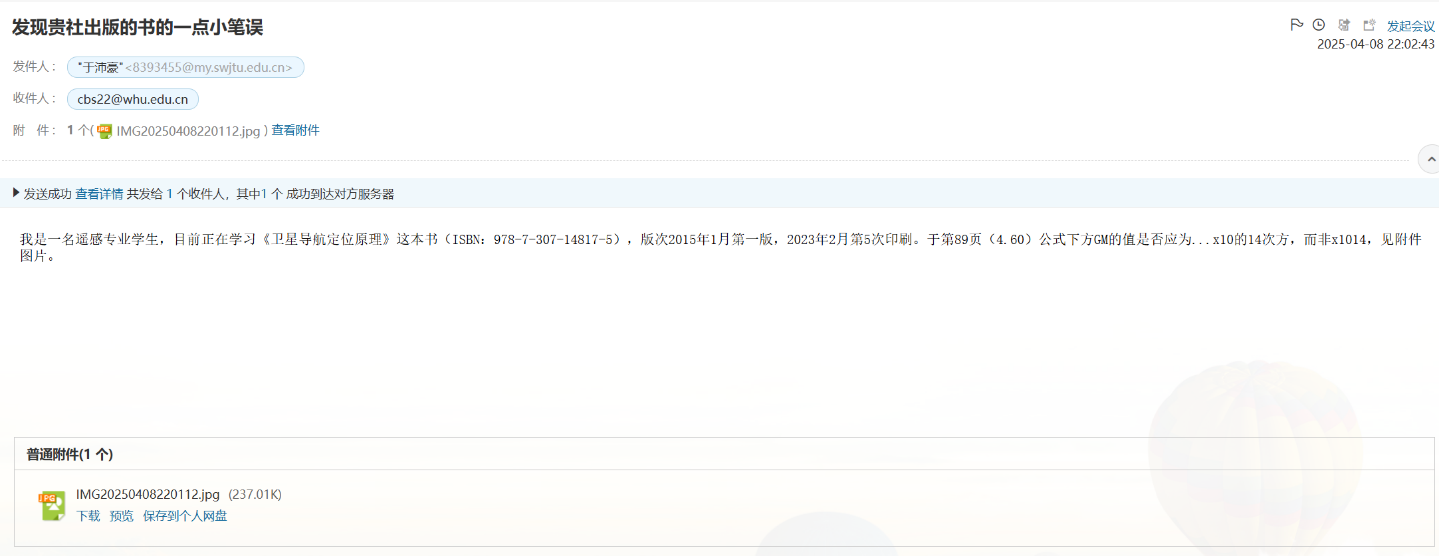


**图2 120分钟卫星轨迹**

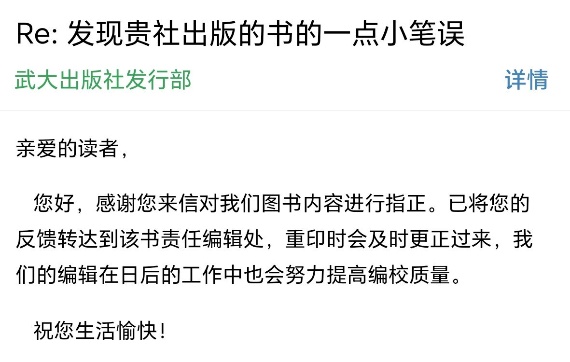
# 五、经验总结

5.1 小插曲

在《卫星导航定位原理》这本书（ISBN：978-7-307-14817-5），版次2015年1月第一版，2023年2月第5次印刷。于第89页（4.60）公式下方GM的值有错。书中数据为GM＝。实际应为GM＝

已发邮件到武汉大学出版社。

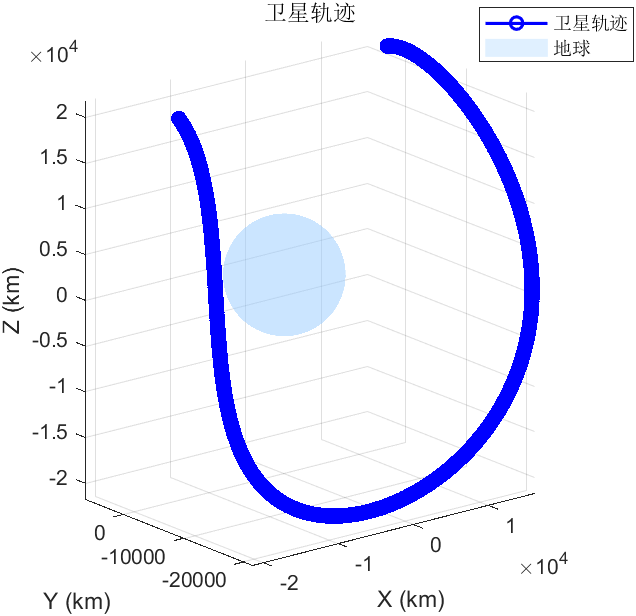
得到回应如下：



5.2 反思与改进

原程序只支持基于一个星历文件的预测，若数据过长则由于星历文件参数过时则造成误差过大。则可以对转换函数整体作为for循环的内容，以星历文件个数（两小时一次）为循环，便可实现将不同星历文件分别预测120min卫星位置后叠加到变量Total\_XYZ中，只需一点小改动便可使预测长时间数据更加精准，因为若预测过长时间段则会星历文件参数过时则造成误差过大，该种办法相当于每个星历文件预测120点后叠加。

通过预测更长时间段（2025-1-1 12:00:00 - 2025-1-2 00:00:00）所生成的图像如下：



因为ECEf是地心地固坐标系，是转动的，所以无法呈现出标准椭圆。至于更好显示出围绕地球转动的效果则可以乘以旋转矩阵改变参数。

5.3 总结

通过这次实验，我加深了对Matlab的理解，其中我明白了规范的代码编写方法，包括如何编写变量名，atan与atan2的细微区别，format的设置，收敛精度的确定，以及3d绘图，即plot3的用法。这些都为我以后的学习帮助了许多。

从查文档了解各个参数的意义，到完成程序编写后，直到最后输出图像的时候，那一刻的自豪感真是无可比拟。

程序已随实验报告打包提交。