

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«МЭИ»**

**ИНСТИТУТ РАДИОТЕХНИКИ И ЭЛЕКТРОНИКИ**

**КАФЕДРА РАДИОТЕХНИЧЕСКИХ СИСТЕМ**

**КУРСОВАЯ РАБОТА**

**по дисциплине**

**«АППАРАТУРА ПОТРЕБИТЕЛЕЙ СПУТНИКОВЫХ  
РАДИОНАВИГАЦИОННЫХ СИСТЕМ»**

**по теме**

**Разработка модуля расчёта координат спутника Beidou**

**ФИО СТУДЕНТА: ЖЕРЕБЦОВ И.С.**

**ГРУППА: ЭР-15-16**

**ВАРИАНТ №: 2**

**ДАТА: \_\_\_\_\_**

**ПОДПИСЬ: \_\_\_\_\_**

**ФИО ПРЕПОДАВАТЕЛЯ: КОРОГОДИН И.В.**

**ОЦЕНКА: \_\_\_\_\_**

**МОСКВА  
2021**

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ .....</b>	<b>3</b>
<b>ЗАДАНИЕ .....</b>	<b>4</b>
<b>ПЕРВЫЙ ЭТАП: Использование сторонних средств .....</b>	<b>6</b>
1. Определение формы орбиты и положения спутника на ней.....	6
2. Расчет графика угла места собственного спутника от времени .....	7
3. Расчет диаграммы угла места и азимута спутника .....	9
4. Формирование таблицы эфемерид собственного спутника.....	11
<b>ВТОРОЙ ЭТАП: Моделирование.....</b>	<b>12</b>
1. Реализация в Matlab.....	12
2. Траектории спутника.....	12
3. Расчет диаграммы угла места и азимута спутника .....	14
<b>ТРЕТИЙ ЭТАП: Моделирование .....</b>	<b>15</b>
1. Реализация в Matlab.....	15
2. Траектории спутника.....	15
3. Расчет диаграммы угла места и азимута спутника .....	16
<b>ЗАКЛЮЧЕНИЕ.....</b>	<b>18</b>
<b>Список литературы .....</b>	<b>19</b>

## ВВЕДЕНИЕ

**Цель проекта** - добавление в программное обеспечение приемника функции расчета положения спутника Beidou на заданное время по данным его эфемерид.

Требования к разрабатываемому программному модулю:

- требования назначения;
- отсутствие утечек памяти;
- малое время выполнения;
- низкий расход памяти;
- корректное выполнение при аномальных входных данных.

Для достижения цели выполняется ряд задач, соответствующих этапам проекта и контрольным мероприятиям:

- обработка данных от приемника, работа со сторонними сервисами для подготовки входных и проверочных данных для разрабатываемого модуля;
- моделирование модуля в Matlab/Python;
- реализация программного модуля на C/C++, включая юнит-тестирование в Check.

Конечная цель всего курсового проекта - получить библиотеку функций на «C++», позволяющую рассчитывать положение спутника Beidou по его эфемеридам.

## ЗАДАНИЕ

### Этап 1. Использование сторонних средств

Конечная цель всего курсового проекта - получить библиотеку функций на Си++, позволяющую рассчитывать положение спутника Beidou по его эфемеридам. На первом этапе подготовим вспомогательные данные для разработки: эфемериды и оценки положения спутника от сторонних сервисов (чтобы было с чем сравниваться на след. этапах)

На крыше корпуса Е МЭИ установлена трехдиапазонная антенна Narxon NX-CSX601A. Она через 50-метровый кабель, сплиттер, bias-tee и усилитель подключена к трем навигационным приемникам:

Javad Lexon LGDD,

SwiftNavigation Piksi Multi,

Clonicus разработки ЛНС МЭИ.

Эти приемники осуществляют первичную обработку сигналов Beidou B1I, выдавая по интерфейсам соответствующие потоки данных - наблюдения псевдодальностей и эфемериды спутников. Данные от приемника Clonicus, записанные вечером 16 февраля 2021 года, доступны в рабочем репозитории (директория logs) в нескольких форматах.

### Этап 2. Моделирование

Эфемериды - параметры некоторой модели движения спутника. В разных ГНСС эти модели разные, а значит отличается и формат эфемерид, и алгоритмы расчета положения спутника.

Одна из самых простых и удобных моделей - в системе GPS. Beidou наследует данную модель.

Требуется реализовать на языке Matlab или Python функцию расчета положения спутника Beidou на заданный момент по шкале времени UTC. В качестве эфемерид использовать данные, полученные на предыдущем этапе.

Построить трехмерные графики множества положений спутника Beidou с системным номером, соответствующим номеру студента по списку. Графики в двух вариантах: в СК ECEF WGS84 и соответствующей ей инерциальной СК. Положения должны соответствовать временному интервалу с 8:00 МСК 16 февраля до 06:00 МСК 17 февраля 2021 года. Допускается использовать одни и те же эфемериды на весь рассматриваемый интервал.

Построить SkyView за указанный временной интервал (напоминаю, антенна на крыше корпуса Е) и сравнить результат с Trimble GNSS Planning Online, полученный на прошлом этапе.

### Этап 3. Реализация

Требуется разработать на языке C/C++ функцию расчета положения спутника Beidou на заданное время по шкале UTC, минимизируя время её исполнения и количество затрачиваемой оперативной памяти. Вызов функции не должен приводить к выбросу исключений или утечкам памяти при любом наборе входных данных.

Функция расчета положения спутника в Matlab/Python относительно проста, т.к. доступны библиотеки линейной алгебры и решения уравнений. Но при разработке встраиваемого ПО приходится сохранять лицензионную частоту, минимизировать вычислительную нагрузку и затраты памяти. Поэтому отобразить модель из Matlab/Python в прошивку приемника дословно, как правило, не получается. В рассматриваемом примере потребуется, как минимум, выполнить свою реализацию решения трансцендентного уравнения.

Программный модуль должен сопровождаться unit-тестами под check:

Тесты функции решения уравнения Кеплера

Тест расчетного положения спутника в сравнении с Matlab/Python с шагом 0.1 секунды.

Во время второго теста должно вычисляться и выводиться средняя длительность исполнения функции. Допускается использовать одни и те же эфемериды на весь рассматриваемый интервал (как на предыдущем этапе).

Требуется провести проверку на утечки памяти с помощью утилиты valgrind.

## ПЕРВЫЙ ЭТАП: Использование сторонних средств

### 1. Определение формы орбиты и положения спутника.

Используя сервис «Википедия» определим ID и SCN спутника 7:

№	Спутник	PRN	Дата (UTC)	Ракета	NSSDC ID	SCN	Орбита	Статус	Система
—	Бэйдоу-1 А	N/A	30.10.2000, 16:30	CZ-3A	2000-069A	26599	ГСО, 140° в. д.	выведен с декабря 2011	Бэйдоу-1
—	Бэйдоу-1 В	N/A	20.12.2000, 16:20	CZ-3A	2000-082A	26643	ГСО, 80° в. д.	выведен с декабря 2011	
—	Бэйдоу-1 С	N/A	24.05.2003, 16:34	CZ-3A	2003-021A	27813	ГСО, 110,5° в. д.	выведен с декабря 2012	
—	Бэйдоу-1 D	N/A	02.02.2007, 16:28	CZ-3A	2007-003A	30323	сведён с орбиты <sup>[23]</sup>	выведен с февраля 2009	
1	Компас М1	N/A	13.04.2007, 20:11	CZ-3A	2007-011A	31115	ССО, ~21 500 км	выведен	
2	Компас G2	N/A	14.04.2009, 16:16	CZ-3C	2009-018A	34779	неконтролируемая <sup>[24]</sup>	выведен	
3	Компас G1	N/A	16.01.2010, 16:12	CZ-3C	2010-001A	36287	ГСО, 140° в. д. <sup>[15]</sup>	в резерве	
4	Компас G3	N/A	02.06.2010, 15:53	CZ-3C	2010-024A	36590	ГСО, 110,5° в. д.	в резерве	
5	Компас IGSO-1	C06	31.07.2010, 20:50	CZ-3A	2010-036A	36828	Геосинхронная, накл. 55°, 118° в. д.	действующий	
6	Компас G4	C04	31.10.2010, 16:26	CZ-3C	2010-057A	37210	ГСО, 160° в. д.	действующий	
7	Компас IGSO-2	C07	17.12.2010, 20:20	CZ-3A	2010-068A	37256	Геосинхронная, накл. 55°, 118° в. д.	действующий	

Рисунок 1 - Состояние системы BeiDou с сайта Википедия

Из таблицы рисунка 1, с сервис «Википедия», видно, что спутник 7 имеет ID 2010-068A и SCN 37256. Введем в сервисе CelesTrak, SCN спутника и проведем моделирование на момент времени 15:00, 16 февраля 2021, так как на данном сервисе отсчет времени происходит по UTC(0).

18:00 МСК 16 февраля до 06:00 МСК 17 февраля 2021 года.

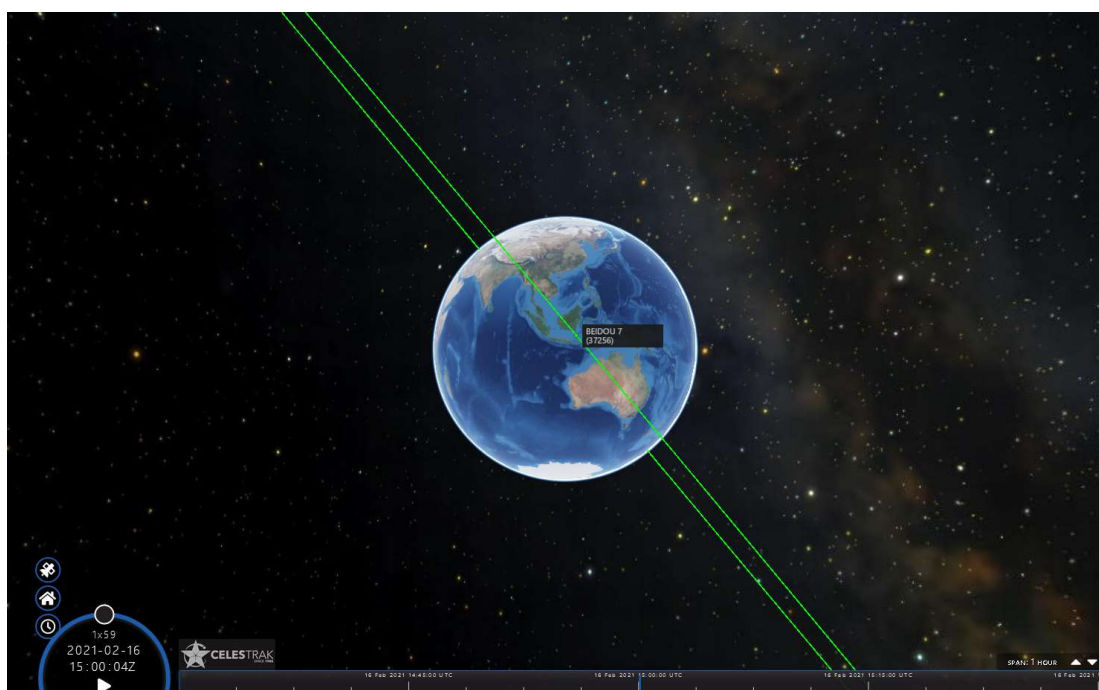


Рисунок 2 -Моделирование с помощью сервиса CelesTrak

## 2. Расчет графика угла места собственного спутника от времени

Настроим для моделирования GNSS Planning Online, координаты установим в соответствии с расположением антенны – и они будут соответствовать значению корпуса Е МЭИ «55.756555, 37.702868». Начальное время будет соответствовать 18:00, временной пояс будет равен +3 (UTC +3) на всем этапе моделирования в сервисе GNSS Planning Online.

Settings

Latitude: N 55° 45' 23.598"

Longitude: E 37° 42' 10.3248"

Height: 500 m

Elevation cutoff: 10

Day: 16.02.2021 Today

Start time: 18:00 UTC +03:00

Period [hours]: 24

Time zone: (UTC+03:00) Moscow, St. Petersburg

Apply

Рисунок 3 – Заданные параметры моделирования в сервисе Trimble GNSS Planning

Так же ограничим количество отображаемых спутников и оставим в моделирование только нужны нам спутник – C07.

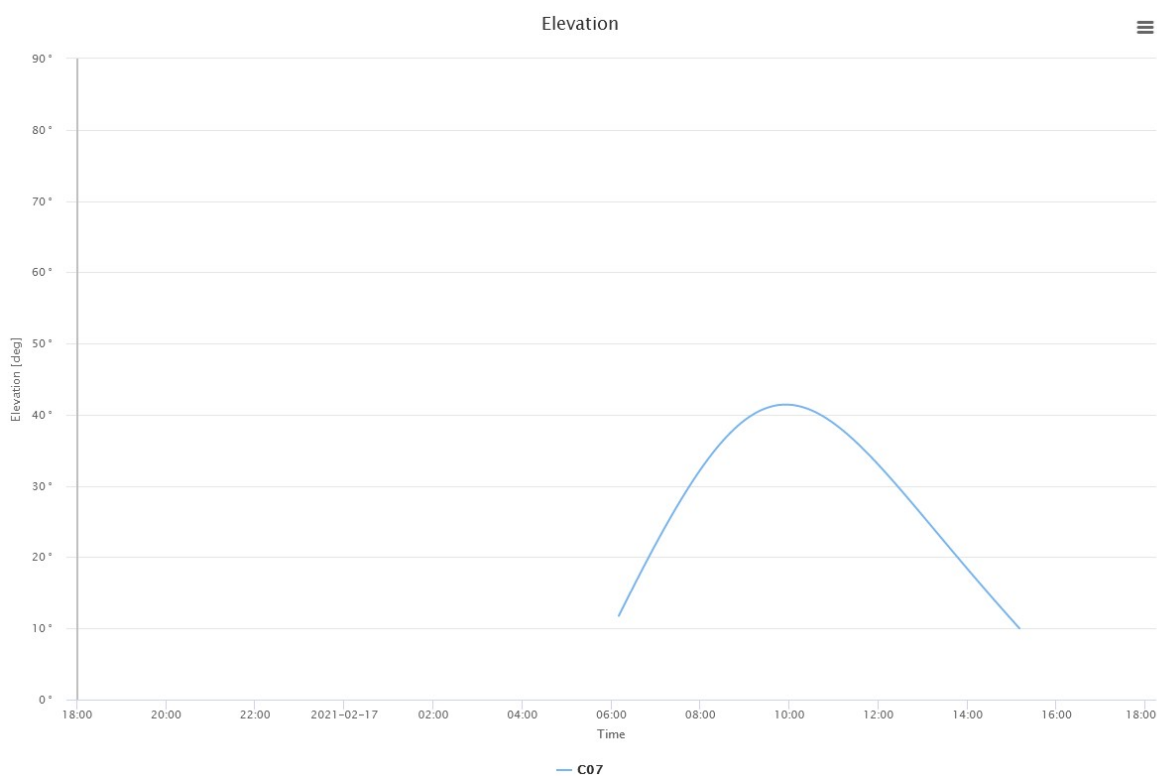


Рисунок 4 – График угла места собственного спутника от времени



По графику видно, что на указанном в задании интервале с 18:00 – 06:00, спутник не был виден, в область видимости же он попадает только после конца заданного интервала и виден 17 февраля с 6:10 до 15:10.

### 3. Расчет диаграммы угла места и азимута спутника

Так как сервис для определения Sky Plot используется тот же - Trimble GNSS Planning Online, то настройки оставим прежние, и проведем моделирование Sky Plot во временном интервале с 6:10 до 15:10 и зафиксируем положение спутника на небосводе в критических точках.

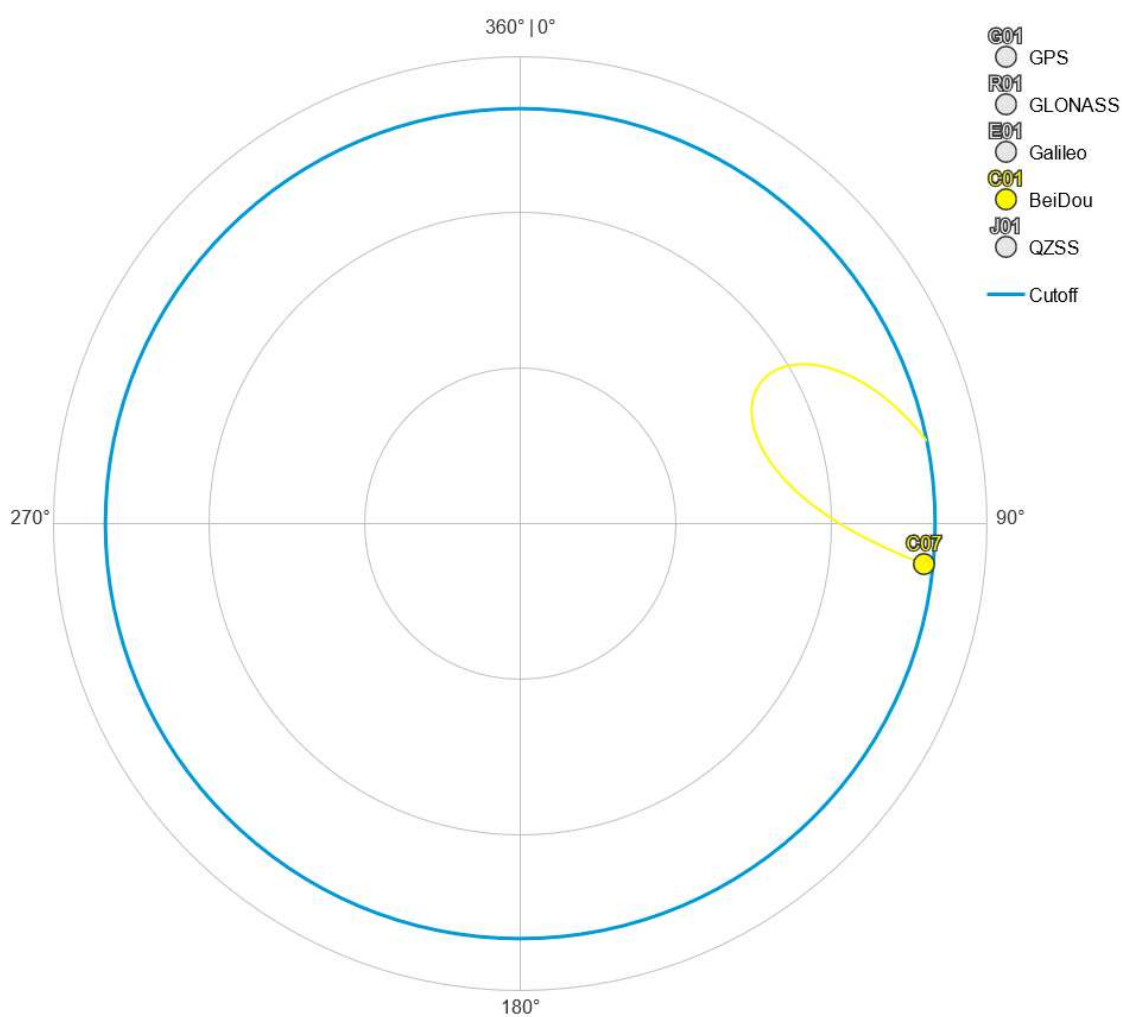


Рисунок 5 – Моделирование с помощью сервиса Trimble GNSS Planning в 17 февраля 2021 в 06:10

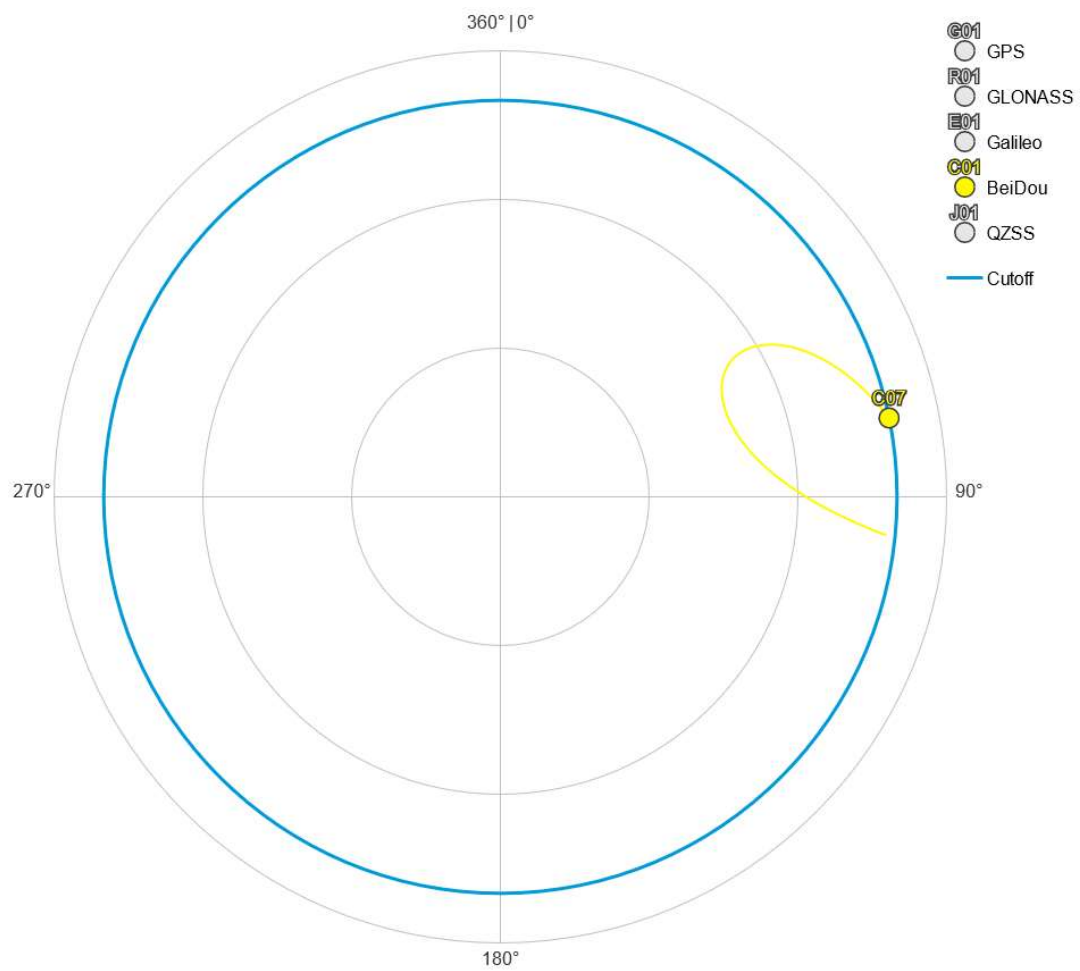


Рисунок 6 – Моделирование с помощью сервиса Trimble GNSS  
Planning в 17 февраля 2021 в 15:10

#### 4. Формирование таблицы эфемерид собственного спутника

Таблица 1. Значения эфемерид спутника

Параметры	Значения	Размерность
Sat	07	-
Toe	284400000.000	мс
Crs	1.16406250000000000e+01	м
Dn	1.98079681996976564e-12	рад/с
M0	-2.45617118216572505e+00	рад
Cuc	2.35158950090408325e-07	рад
e	8.14774842001497746e-03	-
Cus	-2.74321064352989197e-06	рад
sqrtA	6.49292568778991699e+03	м <sup>1/2</sup>
Cic	-2.70549207925796509e-07	рад
omega0	2.63970155955976082e+00	рад
Cis	-1.01979821920394897e-07	рад
i0	8.91248838651520714e-01	рад
Crs	3.05328125000000000e+02	м
omega	-2.52291283308052350e+00	рад
OmegaDot	-2.82868925483299065e-12	рад/мс
iDot	-2.30009580822278564e-13	рад/мс
Tgd	2.43000000000000000e+05	мс
toc	2.84400000000000000e+08	мс
af2	0.00000000000000000e+00	мс <sup>-1</sup>
af1	-1.36628486302470264e-11	-
af0	-9.36393141746520996e-01	мс
URA	0	-
IODE	257	-
IODC	1	-
codeL2	0	-
L2P	0	-
WN	789	-

## ВТОРОЙ ЭТАП: Моделирование

### 2.1 Реализация в Matlab

Для расчета были использованы данные из таблицы 1, полученные на прошлом этапе. Начальная точка во времени была выбрана 16-ое число. Так как 16-ое число является началом 2-ой недели, то можем отнять от 16-ти 14 дней. Поскольку время у нас московское, то вычтем 3 часа (UTC+3) и переведем в секунды и получим:  $((16-2\cdot 7)+18-3) \cdot 60 \cdot 60 = 226800$  [с].

### 2.2 Траектории спутника

Алгоритм расчета был взят из [ИКД Beidou, сигнал В1 стр.36]. Скорость вращения земли ( $\Omega_e$ ) была взята из [ИКД Beidou, сигнал В1 стр.3]. Построенные траектории в различных СК можно увидеть на рис. 7 и 8. Алгоритм в приложение А.

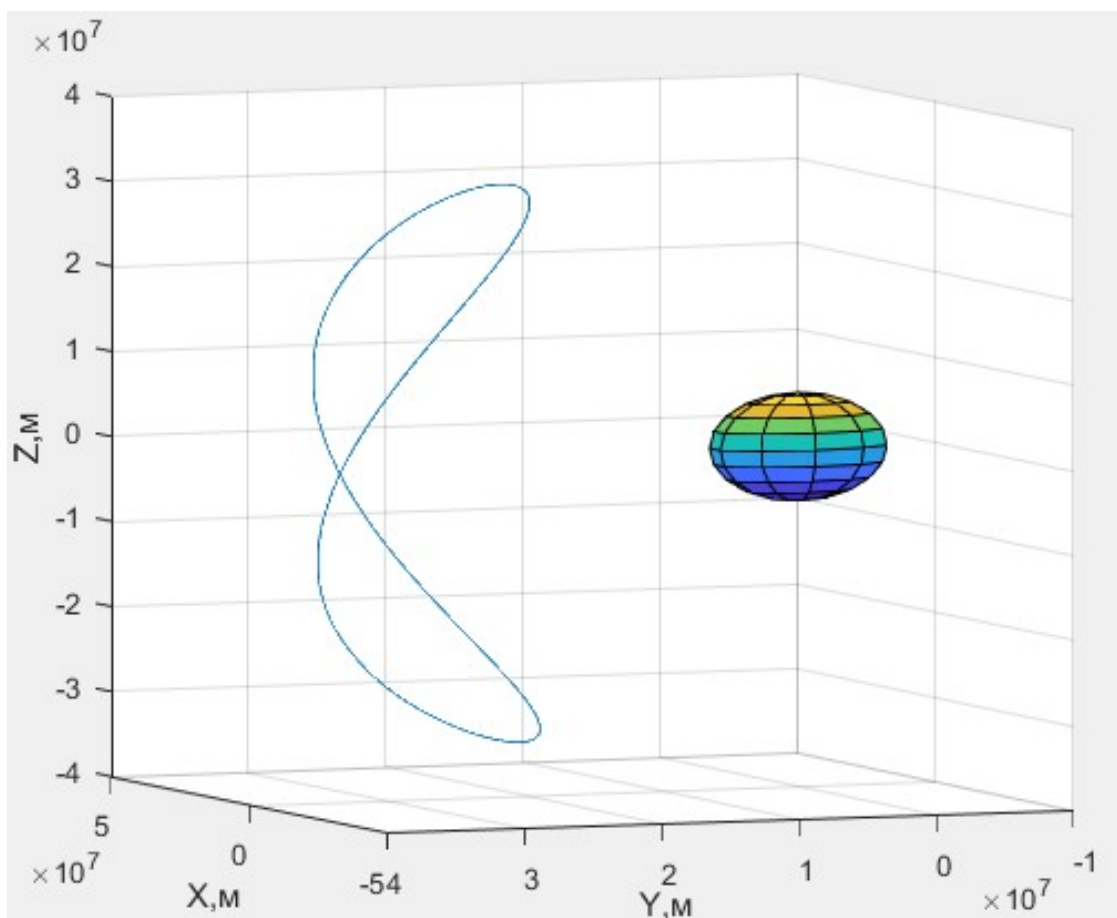


Рисунок 7 – Траектория спутника в СК ECEF WGS84

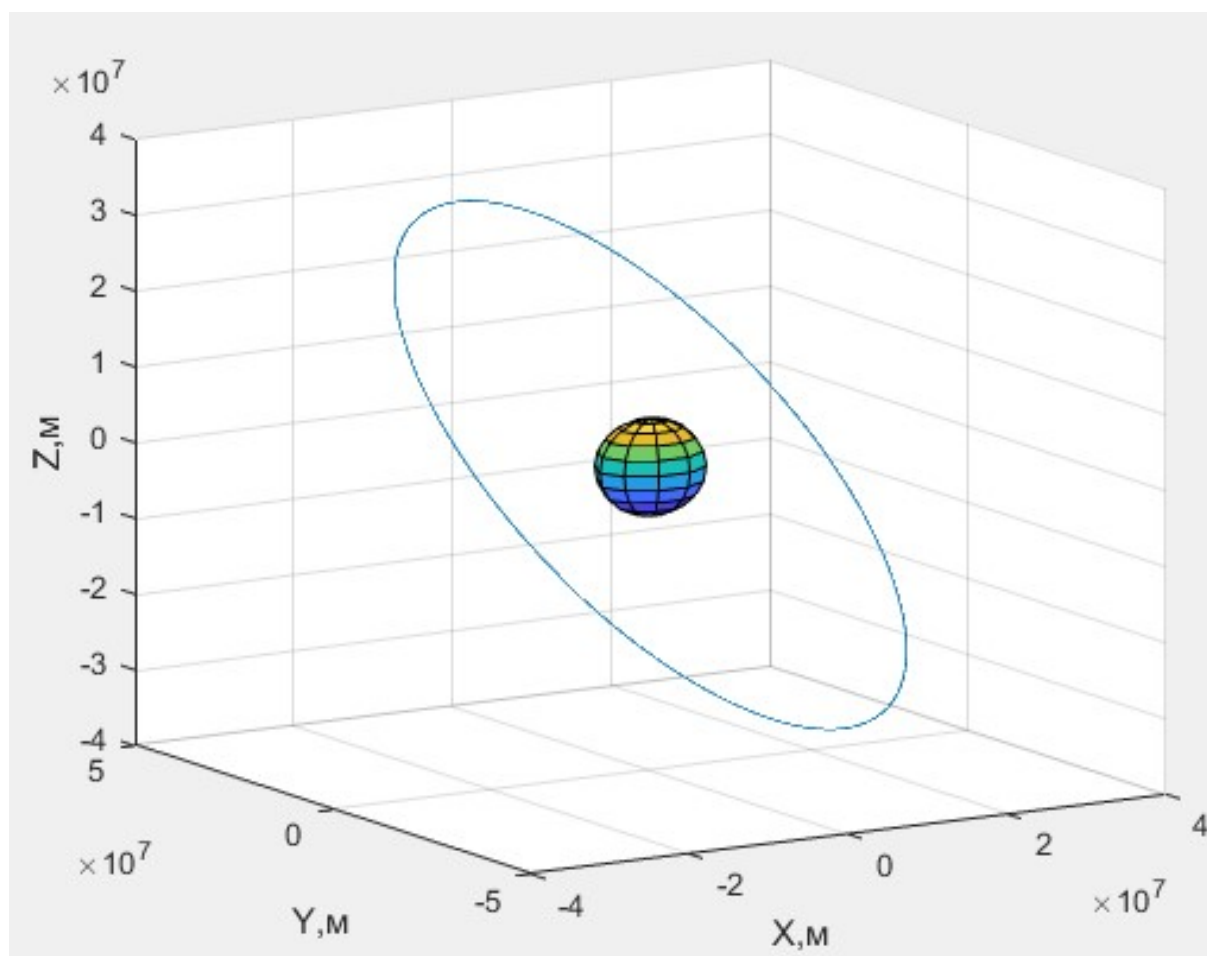


Рисунок 8 – Траектория спутника в инерциальной СК

### 2.3 Построение SkyView.

Для сравнения с моделью полученный с помощью сервиса Trimble GNSS Planning в 1-ом этапе с рассчитанной SkyView. Полученный результат продемонстрирован на рисунке 9.

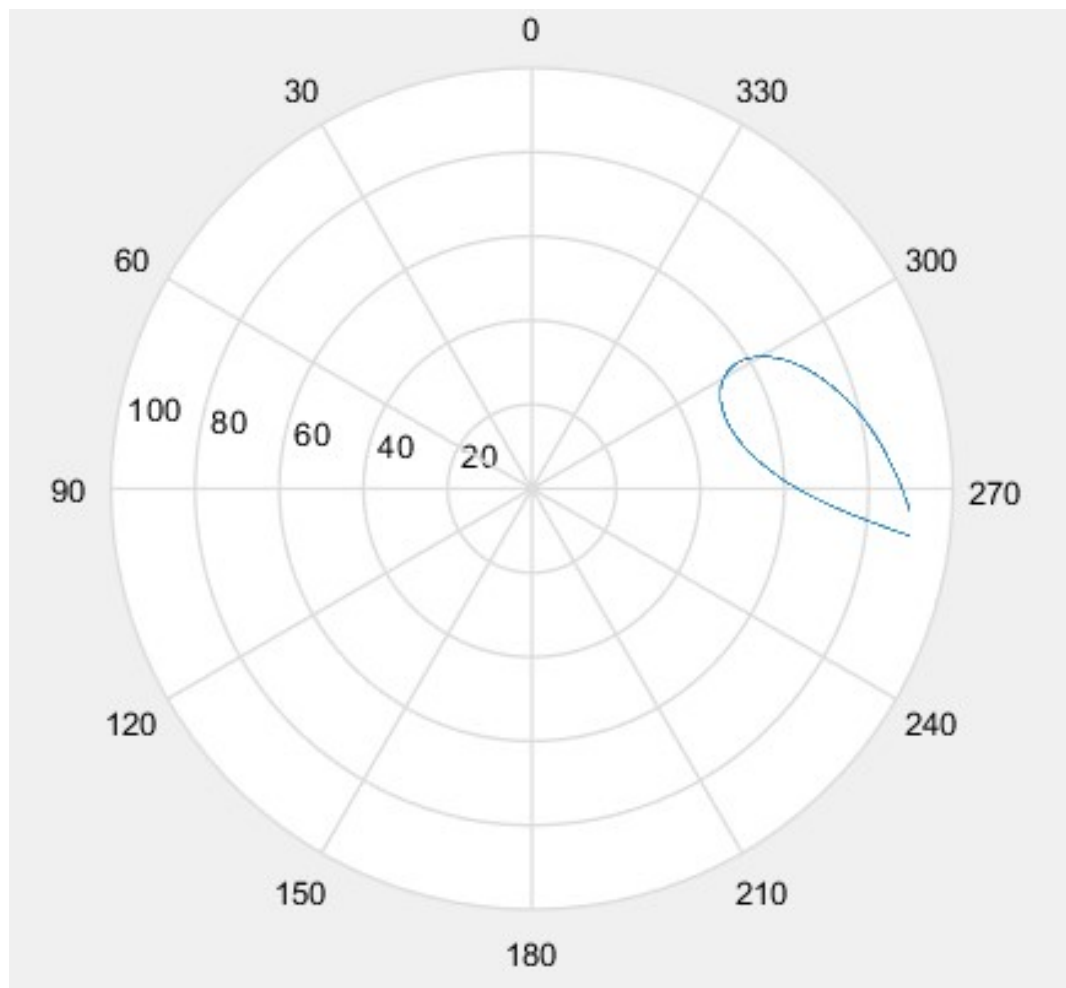


Рисунок 9 – Рассчитанный SkyView

## ТРЕТИЙ ЭТАП: Реализация

### 3.1 Описание кода реализации

Задача была решена с помощью компилятора Code::Blocks. Программа представляет собой сборку из трех главных и двух заголовочных файлов. Далее будут рассмотрены данные файлы.

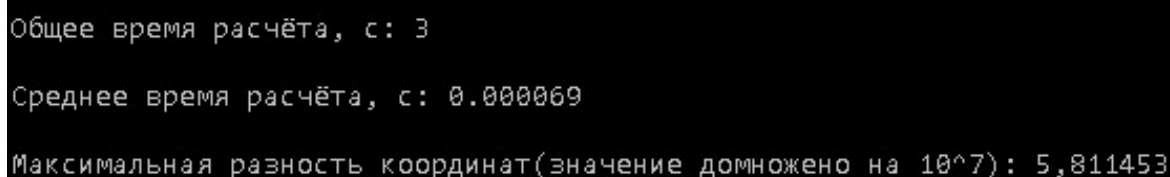
Файл `main.cpp` это основа проекта. В нем происходит запись координат и сравнение со значениями, полученными в Matlab на прошлом этапе. Его можно подробнее увидеть в приложении Б. Для сравнения значений, Matlab записывает координаты в файл «`matlab.txt`».

В файле «`Soordinate.cpp`» программный код происходит расчёт координат по эфемеридам, находящимся в `main.cpp`. С кодом можно ознакомиться в приложении В.

Решение уравнения Кеплера реализуется в файле «`Kepler.cpp`». Код можно увидеть в приложении Г.

А также два заголовочных файла «`Kepler.h`» и «`Soordinate.h`». Содержание данных файлов можно увидеть в приложениях Д и Е.

### 3.2 Результат



```
Общее время расчёта, с: 3
Среднее время расчёта, с: 0.000069
Максимальная разность координат(значение домножено на 10^7): 5,811453
```

Рисунок 10 – Результат работы программы

Следовательно, максимальная разность координат значения 581 нм. Общее время расчета колеблется в районе 3 секунд, время одной итерации примерно 69 мкс.

### 3.3 Анализ работы программы

Дальнейший анализ производился с помощью Visual Studio где есть вкладка отладки. С помощью неё мы и будем проводить анализ программы.

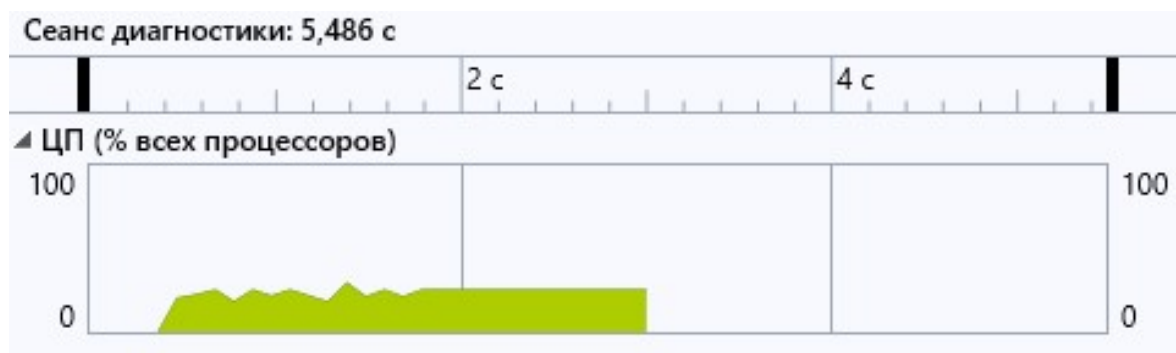


Рисунок 11 – График ЦП

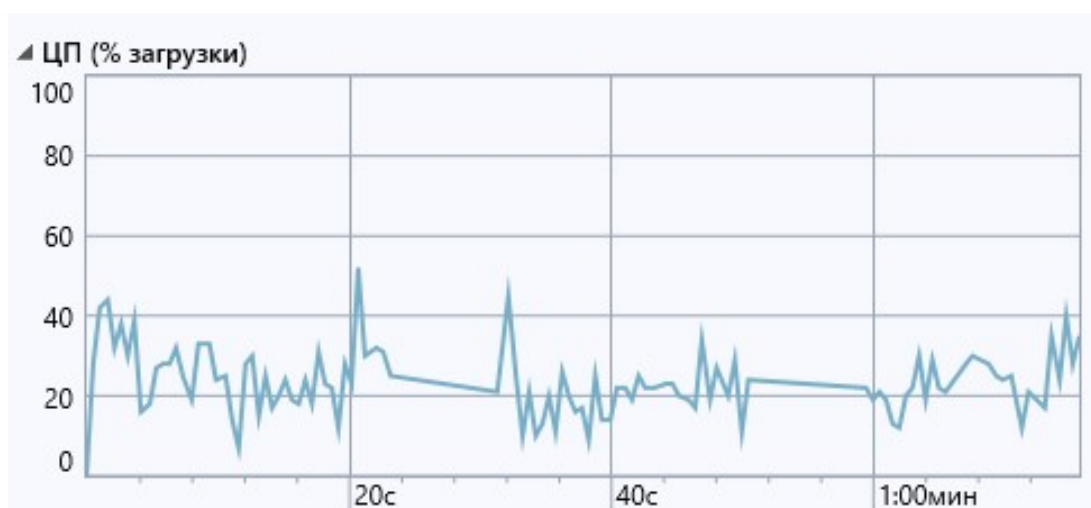


Рисунок 12 – График загрузки ЦП



Критический путь		
Имя функции	% затраченного инклюзивного времени	% затраченного эксклюзивного времени
 main	97.27	0.03
 std::to_string	58.69	1.18
 std::basic_string<char,std::char_traits<char>,std...	27.15	0.04
 sprintf_s	12.75	11.89
 std::basic_string<char,std::char_traits<char>,std...	12.24	0.04

Рисунок 13 – Распределение вызовов по времени

Функции, выполняющие максимальную индивидуальную работу	
Имя	Эксклюзивное время %
_RTC_CheckEsp	19.93
std::_Container_proxy::_Container_proxy	12.20
sprintf_s	11.89
std::allocator<std::_Container_proxy>::allocate	9.06
std::basic_ostream<char,std::char_traits<char> >::...	8.54

Рисунок 13 – Распределение функций по времени

На рисунке 15 показано использование памяти процесса по времени. Утечки не были обнаружены.

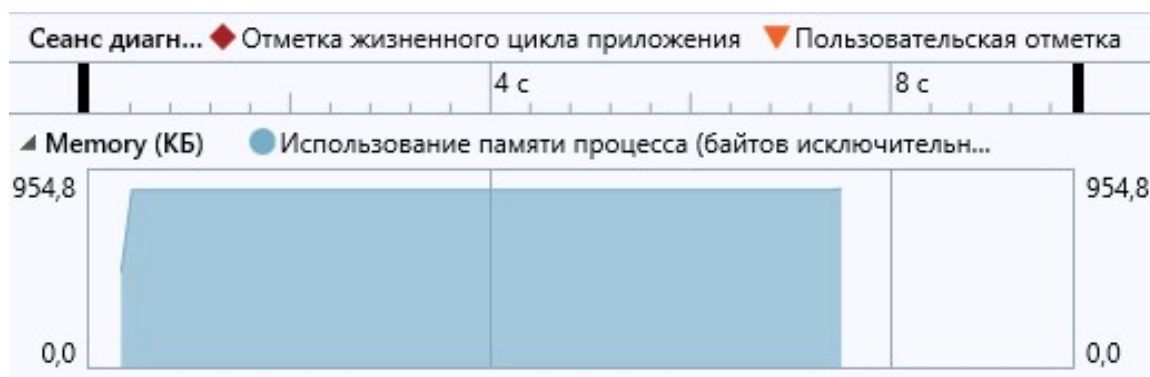


Рисунок 14 – Результат работы программы

При расчете в разных средах программирования или Matlab и C++ имеют некоторые отличия, которые, как и во времени выполнения (что можно заметить невооружённым взглядом), так и в точности.

## **ЗАКЛЮЧЕНИЕ**

В данном курсовом проекте были проведены:

- ознакомительные работы с помощью таких сторонних средств как «Trimble GNSS Planning» и CelesTrak. Данные сервисы (средства) показывали то, что мы должны будем сделать в последующих этапах.
- разработки в среде Matlab, был произведён расчет координат для заданного спутника, так же были получены сравнимые значения и графические отображения траекторий и основных характеристик.
- была проанализирована разработанная программа, но и полученные ранее результаты программы как написанные на C++ так и в Matlab.

## **Список использованной литературы**

- 1) [https://www.srns.ru/wiki/Аппаратура\\_потребителей\\_спутниковых\\_радионавигационных\\_систем\\_\(дисциплина\)](https://www.srns.ru/wiki/Аппаратура_потребителей_спутниковых_радионавигационных_систем_(дисциплина))
- 2) BeiDou Navigation Satellite System Signal In Space, Interface Control Document, Open Service Signal B1I (Version 3.0), China Satellite Navigation Office, February 2019.
- 3) <https://ru.wikipedia.org/wiki/Бэйдоу>
- 4) [https://gssc.esa.int/navipedia/index.php/GPS\\_and\\_Galileo\\_Satellite\\_Coordinates\\_Computation](https://gssc.esa.int/navipedia/index.php/GPS_and_Galileo_Satellite_Coordinates_Computation)
- 5) [http://www2.unb.ca/gge/Resources/beidou\\_icd\\_english\\_ver2.0.pdf](http://www2.unb.ca/gge/Resources/beidou_icd_english_ver2.0.pdf)

## • ПРИЛОЖЕНИЕ А

```

close all;
clear all;
clc;

Toe = 284400;
Crs = 1.1640625000000000e+01;
mu = 3.986004418e+14;
dn = 1.98079681996976564e-12;
Cuc = 2.35158950090408325e-07;
e = 8.14774842001497746e-03;
Cus = -2.74321064352989197e-06;
A = 6.49292568778991699e+03^2;
Cic = -2.70549207925796509e-07;
Wo = 2.63970155955976082e+00;
Cis = -1.01979821920394897e-07;
Io = 8.91248838651520714e-01;
Crc = 3.05328125000000000e+02;
Mo = -2.45617118216572505e+00;
We = 7.2921150e-5;
W = -2.52291283308052350e+00;
Wdot = -2.82868925483299065e-12;
idot = -2.30009580822278564e-13;
T = 226800;
wur = 55.45235679;
dl = 37.42120145;
H = 200;

for j=1:86400;
no = sqrt(mu/(A^3));
Tk=T-Toe;
n=no+dn;
M = Mo+n*Tk;
E=0;

for l=1:100;
E=M+e*sin(E);
end

nu = atan2(sqrt(1-e^2)*sin(E),cos(E)-e);
F1 = nu+W;
du = Cus*sin(2*F1)+Cuc*cos(2*F1);
dr = Crs*sin(2*F1)+Crc*cos(2*F1);
di = Cis*sin(2*F1)+Cic*cos(2*F1);
F2 = F1+du;
r = A*(1-e*cos(E))+dr;
i = Io+di+idot*Tk;
poX = r*cos(F2);
poY = r*sin(F2);
Omega = Wo+(Wdot-We)*(Tk)-We*Toe;
x = poX*cos(Omega)-poY*cos(i)*sin(Omega);
y = poX*sin(Omega)+poY*cos(i)*cos(Omega);
z = poY*sin(i);
Resfix(j,:) = [x y z];
phi = We*Tk;
xc = x*cos(phi)-y*sin(phi);
yc = x*sin(phi)+y*cos(phi);
zc = z;
ResECI(j,:)=[xc yc zc];
[East, North, Up] = ecef2enu(x, y, z, wur, dl,H, wgs84Ellipsoid);
R = sqrt(East^2 + North^2 + Up^2);

```

```

el(j) = rad2deg(-asin(Up/R))+90;
az(j) = atan2(East, North);
T=T+1;
end

[X, Y, Z] = sphere(10);
figure;plot3(Resfix(:,1),Resfix(:,2),Resfix(:,3));
hold on;
surf(X*6.371*10^6, Y*6.371*10^6, Z*6.371*10^6);
grid on;
xlabel('X,m');
ylabel('Y,m');
zlabel('Z,m');
figure; plot3(ResECI(:,1),ResECI(:,2),ResECI(:,3));
hold on;
surf(X*6.371*10^6, Y*6.371*10^6, Z*6.371*10^6);
grid on;
xlabel('X,m');
ylabel('Y,m');
zlabel('Z,m');
s = 1;

for y = 1:length(el);
    if el(y) <= 90;
        Cel(s) = el(y);
        Caz(s) = az(y);
        s = s+1;
    end
end

figure;
polar(2*pi-Caz, Cel);
camroll(90);

```

```

#include <iostream>
#include <fstream>
#include <string>
#include "Soordinate.h"
#include "Kepler.h"
#include <ctime>
#include <cmath>

using namespace std;

int main()
{
    Efemeridi Efemer;
    Efemer.mu = 3.986004418E+14;
    Efemer.we = 7.2921150e-5;
    Efemer.toe = 284400;
    Efemer.A = 4.215808398716217e+07;
    Efemer.e = 8.14774842001497746e-03;
    Efemer.M0 = -2.45617118216572505;
    Efemer.omega = -2.52291283308052350;
    Efemer.i0 = 8.91248838651520714e-01;
    Efemer.omega0 = 2.63970155955976082;
    Efemer.del_n = 1.98079681996976564e-12;
    Efemer.i_dot = -2.30009580822278564e-13;
    Efemer.omega_dot = -2.82868925483299065e-12;
    Efemer.c_uc = 2.35158950090408325e-07;
    Efemer.c_us = -2.74321064352989197e-06;
    Efemer.c_rc = 3.05328125000000000e+02;
    Efemer.c_rs = 1.16406250000000000e+01;
    Efemer.c_ic = -2.70549207925796509e-07;
    Efemer.c_is = -1.37835741043090820E-07;

    time_t start, end;
    double t = 226800;
    double delta_t = 1;
    double* coord = new double[3];
    double* coord_matlab = new double[3];
    double max_del = 0;
    ofstream out;
    out.open("cpp.txt");
    ifstream in("matlab.txt");
    time(&start);
    for (int i = 0; i < (43200 / delta_t); i++)
    {
        Ocoord(t, coord, Efemer);
        t += delta_t;
        string coord_str1 = to_string(coord[0]);
        string coord_str2 = to_string(coord[1]);
        string coord_str3 = to_string(coord[2]);
        out << coord_str1 << " " << coord_str2 << " " << coord_str3 <<
endl;

        in >> coord_matlab[0] >> coord_matlab[1] >> coord_matlab[2];
        for (int j = 0; j < 3; j++)
        {
            if (abs(coord[j] - coord_matlab[j]) > max_del)
            {
                max_del = abs(coord[j] - coord_matlab[j]);
            }
        }
    }
    max_del = max_del * 10000000;

```

```

time(&end);
in.close();
delete[] coord;
delete[] coord_matlab;
coord = nullptr;
coord_matlab = nullptr;
double seconds = difftime(end, start);
string seconds1 = to_string(seconds / 43200 / delta_t);
setlocale(LC_ALL, "rus");
cout << "\n\t\tОбщее время расчёта, с: " << seconds << endl;
cout << "\n\t\tСреднее время расчёта, с: " << seconds1 << endl;
string max_dell = to_string(max_del);
cout << "\n\t\tМаксимальная разность координат (значение домножено на
10^7): " << max_dell << endl;
out.close();
in.close();
}

```

## ПРИЛОЖЕНИЕ В

```
#include "Soordinate.h"
#include <iostream>
#include <math.h>
#include <ostream>
#include "Kepler.h"

using namespace std;

void Ocoord(double t, double* coord, Efemeridi Ef)
{
    double tk = t - Ef.toe;
    double Mk = Ef.M0 + (sqrt(Ef.mu) / pow(sqrt(Ef.A), 3) + Ef.del_n) * tk;
    double Ek = kepler(Mk, Ef.e);
    double Vk = atan2(sqrt(1 - pow(Ef.e, 2)) * sin(Ek), cos(Ek) - Ef.e);
    double Uk = Ef.omega + Vk + Ef.c_uc * cos(2 * (Ef.omega + Vk)) + Ef.c_us * sin(2 *
(Ef.omega + Vk));
    double rk = Ef.A * (1 - Ef.e * cos(Ek)) + Ef.c_rc * cos(2 * (Ef.omega + Vk)) +
Ef.c_rs * sin(2 * (Ef.omega + Vk));
    double ik = Ef.i0 + Ef.i_dot * tk + Ef.c_ic * cos(2 * (Ef.omega + Vk)) + Ef.c_is *
sin(2 * (Ef.omega + Vk));
    double lambk = Ef.omega0 + (Ef.omega_dot - Ef.we) * tk - Ef.we * Ef.toe;
    coord[0] = (cos(-lambk) * cos(-Uk) - sin(-lambk) * cos(-ik) * sin(-Uk)) * rk;
    coord[1] = (-sin(-lambk) * cos(-Uk) - cos(-lambk) * cos(-ik) * sin(-Uk)) * rk;
    coord[2] = (sin(-ik) * sin(-Uk)) * rk;
}
```

## ПРИЛОЖЕНИЕ Г

```
#include "Kepler.h"
#include <math.h>

double kepler(double Mk, double e) {
    double Ek = Mk;
    double Ek1;
    do {
        Ek1 = Ek;
        Ek = Mk + e * sin(Ek);
    } while (fabs(Ek1 - Ek) > 0.0000001);
    return Ek;
}
```

## ПРИЛОЖЕНИЕ Д

```
#ifndef KEPLER_H
#define KEPLER_H

double kepler(double Mk, double e);

#endif /* #ifndef KEPLER_H */#pragma once
```



```

#ifndef COORDINATE_H
#define COORDINATE_H
typedef struct {
    double mu;
    double we;
    double toe;
    double A;
    double e;
    double M0;
    double omega;
    double i0;
    double omega0;
    double del_n;
    double i_dot;
    double omega_dot;
    double c_uc;
    double c_us;
    double c_rc;
    double c_rs;
    double c_ic;
    double c_is;
} Efemeridi;
void Ocoord(double t, double* coord, Efemeridi Ef);

#endif /* #ifndef GPSSVPOS_H */#pragma once

```