**Московский Энергетический Институт (Технический Университет)**

**Институт радиотехники и электроники**

**Кафедра радиотехнических систем**

**Курсовой проект по курсу “Аппаратура потребителей СРНС”**

**Разработка модуля расчета координат спутника GPS**

**Руководитель проекта:**

**Корогодин И. В.**

**Автор проекта:**

**Студент гр. ЭР-15-14 Цырульникова Л. А.**

**Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Члены комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**“\_\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г.**

**Москва 2019**

**Оглавление**

**1 ЭТАП I. Использование сторонних средств** .............................................. 3

1.1 Описание процесса использования RTKLIB............................................... 3

1.2 Эфемериды собственного спутника............................................................. 3

1.3 График угла места собственного спутника от времени по данным Trimble GNSS Planning Online на заданный интервал времени................................................ 5

1.4 SkyView по данным Trimble GNSS Planning Online.................................... 6

**2 ЭТАП II. Моделирование** .............................................................................. 7

2.1 Постановка задачи........................................................................................... 7

2.2 Результаты моделирования ............................................................................ 7

2.3 Листинг программы в Matlab.........................................................................12

**3 ЭТАП III. Реализация** .................................................................................... 15

3.1 Выод............................................................. ....................................................16

3.2 Листинг программы ....................................................................................... 17

**ЭТАП I. Использование сторонних средств**

1. **Описание процесса использования RTKLIB**

В пакете RTKLIB для первой части курсового расчета использовали такие программы, как RTKLIB и RTKCONV.

Программа RTKNAVI позволяет вывести таблицу текущих и предыдущих эфемерид. После запуска программы загружаем бинарный файл с данными со спутника. Для этого в верхней панели, справа от GPST, нажимаем на кнопку I, после чего открывается окно Input Streams. Отмечаем галочкой Rover (приемник), выбираем Type–> File, Format –> NVS BINR, в строке Input File Paths задаем место нахождения бинарного файла (BINR\_evening или BINR\_morning). Нажимаем «ОК». В окне RTKNAVI ver.2.4.2 запускаем программу (нажимаем Start). Далее нажимаем на маленький квадратик над кнопкой Start. В верхней панели в выпадающем меню выбираем Nav GPS. По окончанию работы нажать ‘Stop’ и ‘Exit’.

Программа RTKCONV позволяет конвертировать бинарный файл в текстовый формат RINEX, в частности получить текстовый nav-файл с эфемеридами GPS. Запускаем программу. В строке “RTSM, RCV RAW or RINEX OBS ?” выбираем бинарный файл, который хотим конвертировать. В строке Output Directory выбираем место, куда нужно сохранить полученные, конвертированные файлы. В панели “RINEX OBS/NAV/GNAV/HNAV/QNAV/LNAV and SBS” указываем нужные нам форматы, которые хотим получить, в нашем случае это nav формат. По окончанию работы нажать “Exit”.

1. **Эфемериды собственного спутника по данным RTKNAVI из состава RTKLIB (спутник GPS №15)**

Утро:

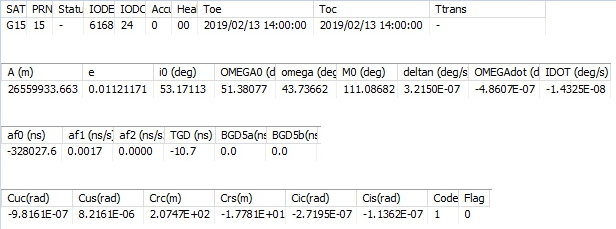


Рисунок 1 - Эфемериды спутника (утро)

Вечер:

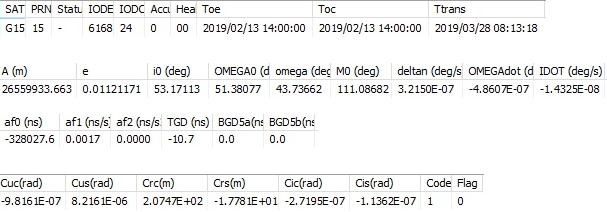


Рисунок 2 - Эфемериды спутника (вечер)

Эфемериды в вечернее и утреннее время совпадают.

1. **Эфемериды собственного спутника в nav-файле RINEX**

BINR\_evening

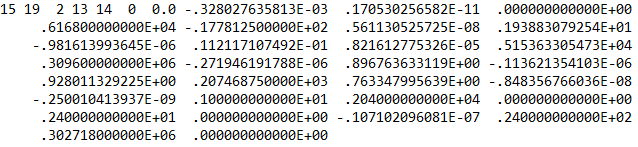


Рисунок 3 - Эфемериды в nav-файле

**4. График угла места собственного спутника от времени по данным Trimble GNSS Planning Online на заданный интервал времени: с 12:00 13.02.19 до 00:00 14.02.19.**

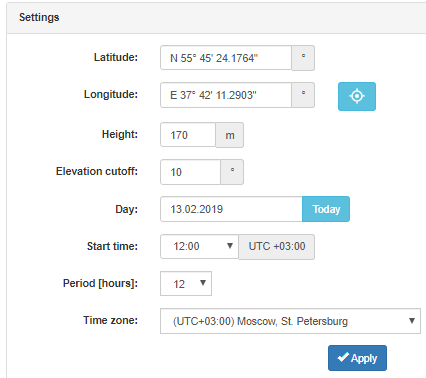


Рисунок 4 - Настройки на заданное время, место и продолжительность

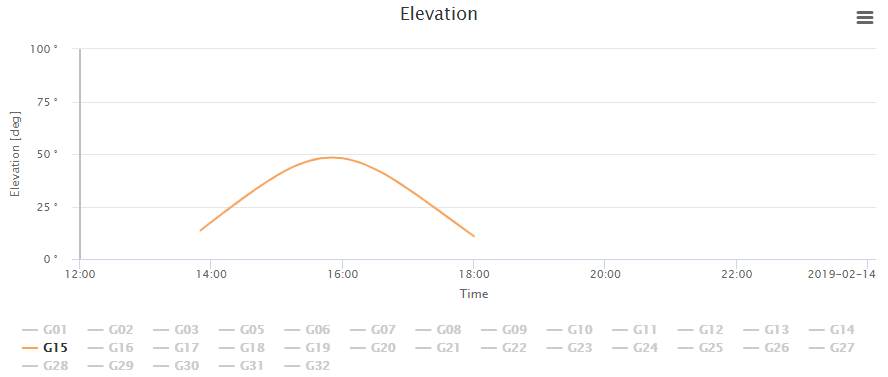


Рисунок 5 - График угла места

**5. SkyView по данным Trimble GNSS Planning Online на заданный интервал времени: с 12:00 13.02.19 до 00:00 14.02.19.**

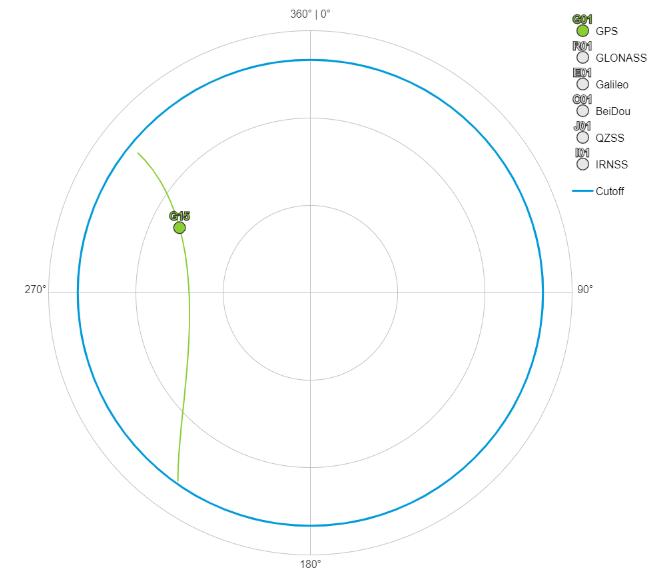


Рисунок 6 - SkyView

**ЭТАП II. Моделирование**

**Постановка задачи:**

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

Построить трехмерные графики множества положений спутника GPS 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 12:00 13.02.19 до 00:00 14.02.19. 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](http://www.trimble.com/gnssplanningonline/), полученный на прошлом этапе.

**2. Результаты моделирования в Matlab**

Для расчетов координат использовалось ИКД системы GALILEO, т.к. алгоритм идентичен алгоритму GPS, но при этом он более наглядный.

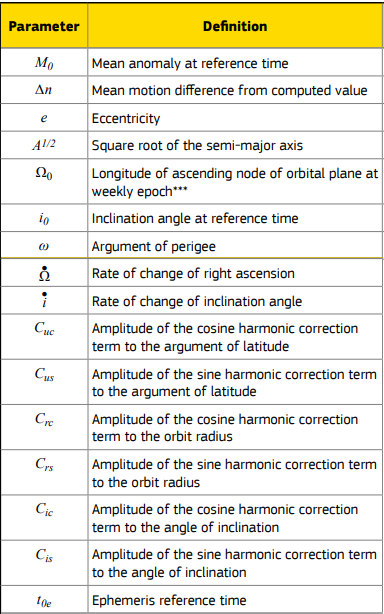


Рисунок 7 – Эфемериды, обозначения

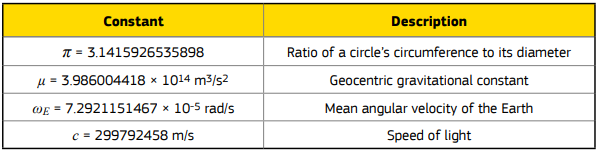
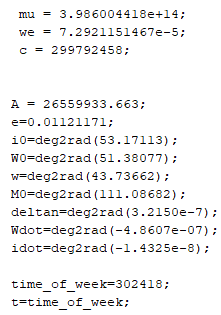
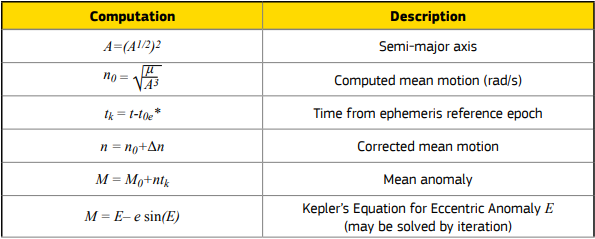


Рисунок 8 - константы

**Таким образом, эфемериды на заданный НКА (15):**



**Для расчета координат НКА воспользуемся алгоритмом:**



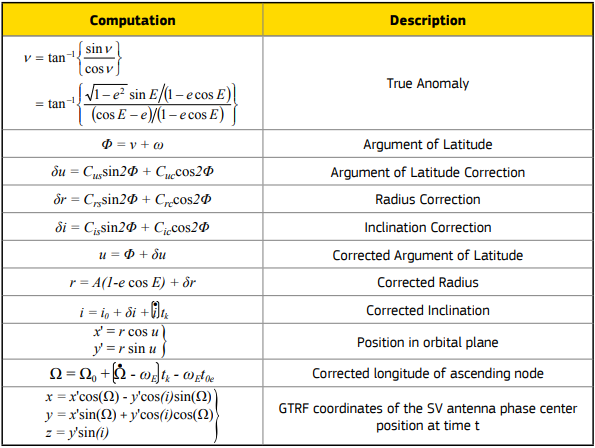


Рисунок 9 - Алгоритм расчета

**Результат:**

1.Трехмерные графики траектории движения спутника GPS(15) в инерциальной и неинерциальной системах координат:

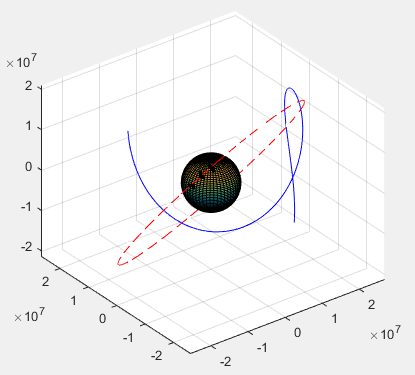


Рисунок 10 - Траектория НКА. Сплошная линия – траектория НКА в неинерциальной СК, пунктирная – траектория НКА в инерциальной СК

2.SkyView

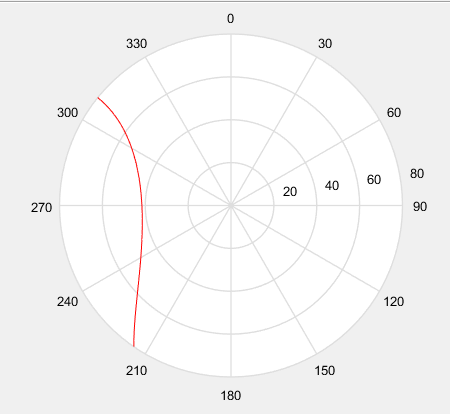
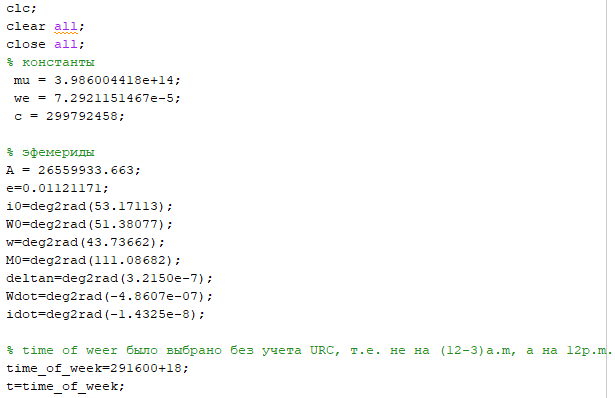
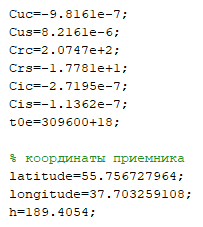


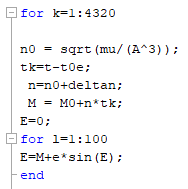
Рисунок 11 - SkyView

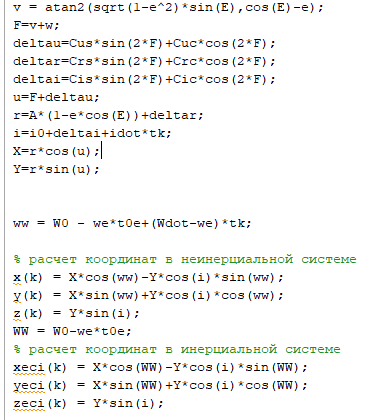
**Вывод:** SkyView рассчитанный в Matlab и построенный через специалзированную программу [Trimble GNSS Planning Online](http://www.trimble.com/gnssplanningonline/) совпадают.

**Листинг программы в Matlab:**









%пересчет из ECEF в ENU

[x0 y0 z0] = ecef2enu(x(k),y(k),z(k),latitude,longitude,h, wgs84Ellipsoid);

%заменила atan на atan2 для того,чтобы упростить условия,т.к. возможно там была ошибка

teta(k) = rad2deg(atan2(z0,sqrt(x0^2+y0^2)));

phi(k) = atan2(x0,y0);

if teta(k)<10 % elevation cutoff брался равным 10, для построения SkyView Trimble GNSS Planning Online

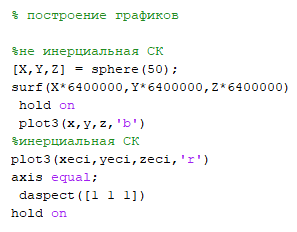
teta(k) = NaN;

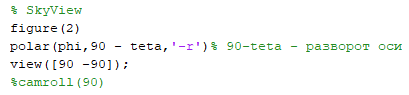
phi(k) = NaN;

end

t=t+10;

end





**ЭТАП III. Реализация**

Проект собирается и запускается, ошибок и исключений не выбрасывает. Лог-файлы при запуске проекта:

"Kursovoy3.exe" (Win32). Загружено "C:\Users\andre\Desktop\MVS tests\Kursovoy3\Debug\Kursovoy3.exe". Символы загружены.

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\ntdll.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\mscoree.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\kernel32.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\KernelBase.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Program Files\AVAST Software\Avast\x86\aswhook.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\vcruntime140d.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\ucrtbased.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\msvcp140d.dll".

Поток 0x2d70 завершился с кодом 0 (0x0).

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\advapi32.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\msvcrt.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\sechost.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\rpcrt4.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\sspicli.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\cryptbase.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\bcryptprimitives.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\Microsoft.NET\Framework\v4.0.30319\mscoreei.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\shlwapi.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\combase.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\ucrtbase.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\gdi32.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\gdi32full.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\msvcp\_win.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\user32.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\win32u.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\imm32.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\kernel.appcore.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\version.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\Microsoft.NET\Framework\v4.0.30319\clr.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\msvcr120\_clr0400.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\psapi.dll".

"Kursovoy3.exe" (Win32). Выгружено "C:\Windows\SysWOW64\psapi.dll"

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\assembly\NativeImages\_v4.0.30319\_32\mscorlib\a5a47e8e5e2880adecca43eb928673f1\mscorlib.ni.dll".

"Kursovoy3.exe" (CLR v4.0.30319: DefaultDomain). Загружено "C:\Windows\Microsoft.Net\assembly\GAC\_32\mscorlib\v4.0\_4.0.0.0\_\_b77a5c561934e089\mscorlib.dll". Загрузка символов пропущена. Модуль оптимизирован, включен параметр отладчика "Только мой код".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\ole32.dll".

"Kursovoy3.exe" (CLR v4.0.30319: DefaultDomain). Загружено "C:\Users\andre\Desktop\MVS tests\Kursovoy3\Debug\Kursovoy3.exe". Символы загружены.

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\Microsoft.NET\Framework\v4.0.30319\clrjit.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\SysWOW64\oleaut32.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\Microsoft.NET\Framework\v4.0.30319\diasymreader.dll".

"Kursovoy3.exe" (Win32). Загружено "C:\Windows\assembly\NativeImages\_v4.0.30319\_32\System\2fb412bd4cb1d503da1dc52104e9b1a4\System.ni.dll".

"Kursovoy3.exe" (CLR v4.0.30319: DefaultDomain). Загружено "C:\Windows\Microsoft.Net\assembly\GAC\_MSIL\System\v4.0\_4.0.0.0\_\_b77a5c561934e089\System.dll". Загрузка символов пропущена. Модуль оптимизирован, включен параметр отладчика "Только мой код".

Поток 0x1384 завершился с кодом 0 (0x0).

Поток 0x197c завершился с кодом 0 (0x0).

Поток 0x9c8 завершился с кодом 0 (0x0).

Программа "[4656] Kursovoy3.exe" завершилась с кодом 0 (0x0).

На рисунке показано, что все тесты пройдены успешно

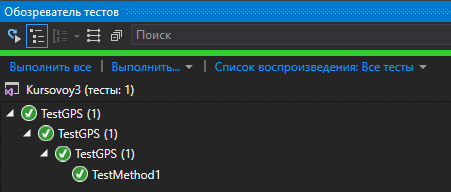


Рисунок 12 - Обозреватель тестов

Время выполнения программы в среде Matlab – 0.214852 сек.

Время выполнения программы в IDE Visual Studio – 0.032503 сек.

Проверка на утечки памяти осуществлялась в библиотеке Microsoft – библиотеки CRT. Необходимо подключить два заголовочных файла (stdlib.h, crtdbg.h) и вызвать метод \_CrtSetDbgLeaks() после выполнения основной программы. Программа не выявила никаких утечек памяти (видно из консоль логов). Вероятно это связано с тем, что динамическая память не выделялась “вручную”, а вместо динамических массивов использовались вектора и ассоциативные массивы из STL, где эти операции скрыты “под капотом”.

**Вывод:**

В результате время выполнения программы в IDE Visual Studio примерно в 2 раза меньше, чем время выполнения в Matlab. Хотя в идеале разница должна быть более значительной (порядка 50 раз). Несоответствие вызвано недостаточной оптимизацией.

**Листинг программы:**

Файл testGPS.cpp:

#include "pch.h"

#include "CppUnitTest.h"

#include "time.h"

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

#include "C:\\Users\\Lyuda\\Desktop\\MVS tests\Kursovoy3\Kursovoy3\Kursovoy3.cpp"

#include <stdlib.h>

#include <crtdbg.h>

#define \_CRTDBG\_MAP\_ALLOC

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

using std::ifstream;

using std::string;

using std::vector;

namespace TestGPS

{

TEST\_CLASS(TestGPS)

{

public:

TEST\_METHOD(TestMethod1)

{

// Массивы для записи результата

float xCoordsOfSat[43200];

float yCoordsOfSat[43200];

float zCoordsOfSat[43200];

// Структура с параметрами спутника

params gpsParams;

// Векторы для координат из матлаба

vector<float> coordsXMatlab;

vector<float> coordsYMatlab;

vector<float> coordsZMatlab;

// Считываем координаты из файлов

ifstream file1("C:\\Users\\Lyuda\\Desktop\\MVS tests\\x\_coords.txt");

ifstream file2("C:\\Users\\Lyuda\\Desktop\\MVS tests\\y\_coords.txt");

ifstream file3("C:\\Users\\Lyuda\\Desktop\\MVS tests\\z\_coords.txt");

if (file1 && file2 && file3) {

vector<string> vecStr1, vecStr2, vecStr3;

string str1, str2, str3;

while (true) {

str1.clear();

str2.clear();

str3.clear();

getline(file1, str1);

getline(file2, str2);

getline(file3, str3);

if (!str1.empty() && !str2.empty() && !str3.empty()) {

vecStr1.push\_back(str1);

vecStr2.push\_back(str2);

vecStr3.push\_back(str3);

} else {

break;

}

}

for (int i = 0; i < vecStr1.size(); ++i) {

for (int j = 0; j < vecStr1[i].size(); ++j) {

coordsXMatlab.push\_back(vecStr1[i][j]);

}

}

for (int i = 0; i < vecStr2.size(); ++i) {

for (int j = 0; j < vecStr2[i].size(); ++j) {

coordsYMatlab.push\_back(vecStr2[i][j]);

}

}

for (int i = 0; i < vecStr3.size(); ++i) {

for (int j = 0; j < vecStr3[i].size(); ++j) {

coordsZMatlab.push\_back(vecStr3[i][j]);

}

}

file1.close();

file2.close();

file3.close();

}

// Вычисляем время начала программы

unsigned int startTime = clock();

positionOfSatellite(gpsParams, xCoordsOfSat, yCoordsOfSat, zCoordsOfSat);

for (int i = 0; i < 43199; i++) {

Assert::AreEqual(xCoordsOfSat[i], coordsXMatlab[i]);

Assert::AreEqual(yCoordsOfSat[i], coordsYMatlab[i]);

Assert::AreEqual(zCoordsOfSat[i], coordsZMatlab[i]);

}

// Анализ утечки памяти

\_CrtDumpMemoryLeaks();

std::cout << (startTime / CLOCKS\_PER\_SEC);

}

};

}

Файл Kursovoy3.cpp:

#include "pch.h"

#include <iostream>

#include "time.h"

#define pi 3,1415926535

// Структура с входными параметрами

struct params {

float A,

M0,

Toe,

omegaZero,

omega,

omegaDot,

omegaDotE,

eccentricity,

inclination,

motionDiff,

M,

IDOT,

Cus,

Cuc,

Crs,

Crc,

Cis,

Cic;

};

// Функция, реализующая решение уравнения Кеплера

float keplersEquation(float anomaly, float eccentricity, float acc) {

float previouslyEk = 0;

float solution = 0;

while (true) {

solution = anomaly + eccentricity \* sin(previouslyEk);

if (abs(previouslyEk - solution) <= acc) {

break;

}

previouslyEk = solution;

}

return solution;

}

// Перевод из градусов в радианы

float degToRad(float degree) {

return degree \* (pi / 180);

}

// Вычисление положения спутника на заданный момент времени

void positionOfSatellite(struct params gpsParams, float (&coordsX)[43200], float (&coordsY)[43200], float (&coordsZ)[43200]) {

gpsParams.A = 26559933.663;

gpsParams.M0 = degToRad(111.08682);

gpsParams.Toe = 302418 + 18;

gpsParams.omegaZero = degToRad(51.38077);

gpsParams.omega = degToRad(43.73662);

gpsParams.omegaDot = degToRad(-4.8607e-07);

gpsParams.omegaDotE = 7.2921151467e-5;

gpsParams.eccentricity = 0.01121171;

gpsParams.inclination = degToRad(53.17113);

gpsParams.motionDiff = degToRad(3.2150e-7);

gpsParams.M = 3.986004418e+14;

gpsParams.IDOT = degToRad(-1.4325e-8);

gpsParams.Cus = 8.2161e-6;

gpsParams.Cuc = -9.8161e-7;

gpsParams.Crs = -1.7781e+1;

gpsParams.Crc = 2.0747e+2;

gpsParams.Cis = -1.1362e-7;

gpsParams.Cic = -2.7195e-7;

for (int i = 0; i < 43199; i++) {

int momentOfTime = 302418 + i;

int Tk = momentOfTime - gpsParams.Toe;

// Определяем точность параметров

float accuracy = pow(10, -7);

if (Tk > 302400) {

Tk = Tk - 604800;

}

else if (Tk < -302400) {

Tk = Tk + 604800;

}

float n0 = pow(gpsParams.M / (pow(gpsParams.A, 3)), 0.5);

float n = n0 + gpsParams.motionDiff;

float anomaly = gpsParams.M0 + n \* Tk;

// Решаем уравнение Кеплера

float Ek = keplersEquation(anomaly, gpsParams.eccentricity, accuracy);

float Vk = atan2((pow((1 - gpsParams.eccentricity \* gpsParams.eccentricity), 0.5) \* sin(Ek) / (1 - gpsParams.eccentricity \* cos(Ek))), ((cos(Ek) - gpsParams.eccentricity) / (1 - gpsParams.eccentricity \* cos(Ek))));

float Fk = Vk + gpsParams.omega;

float deltaUk = gpsParams.Cus \* sin(2 \* Fk) + gpsParams.Cuc \* cos(2 \* Fk);

float deltaRk = gpsParams.Crs \* sin(2 \* Fk) + gpsParams.Crc \* cos(2 \* Fk);

float deltaIk = gpsParams.Cis \* sin(2 \* Fk) + gpsParams.Cic \* cos(2 \* Fk);

float Uk = Fk + deltaUk;

float Rk = gpsParams.A \* (1 - gpsParams.eccentricity \* cos(Ek)) + deltaRk;

float Ik = gpsParams.inclination + deltaIk + gpsParams.IDOT \* Tk;

float Wk = gpsParams.omegaZero + (gpsParams.omegaDot - gpsParams.omegaDotE) \* Tk - gpsParams.omegaDotE \* gpsParams.Toe;

float x = Rk \* cos(Uk);

float y = Rk \* sin(Uk);

// Координаты в системе ECEF

float ecefX = x \* cos(Wk) - y \* cos(Ik) \* sin(Wk);

float ecefY = x \* sin(Wk) + y \* cos(Uk) \* cos(Wk);

float ecefZ = y \* sin(Ik);

// Координаты в системе ECI

float theta = gpsParams.omegaDot \* Tk;

float eciX = ecefX \* cos(theta) - ecefY \* sin(theta);

float eciY = ecefX \* sin(theta) + ecefY \* cos(theta);

float eciZ = ecefZ;

coordsX[i] = eciX;

coordsY[i] = eciY;

coordsZ[i] = eciZ;

}

}

int main()

{

return 0;

}