**Этап 1. Использование.**

1. **Описание процесса использования RTKLIB**

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

Javad Lexon LGDD,

SwiftNavigation Piksi Multi,

FPGA-based приемник на основе нашего ядра CoreZh.

Приемники осуществляют первичную обработку сигналов, выдавая по интерфейсам соответствующие потоки данных - наблюдения псевдодальностей и эфемериды спутников. Необходимо обработать данные от приемника CoreZh, представленные в бинарном виде в формате NVS BINR.

Для проведения вторичной обрабтки имеющихся наблюдения используется подпрограмма RTKNAVIпрограммы RTKlib

В процессе обработки происходит извлечение информации о эффемеридах спутников, необходимых для следующего этапа. Так же, построена зависимость угла места от времени и его проекция на полусферу наблюдаемого неба.

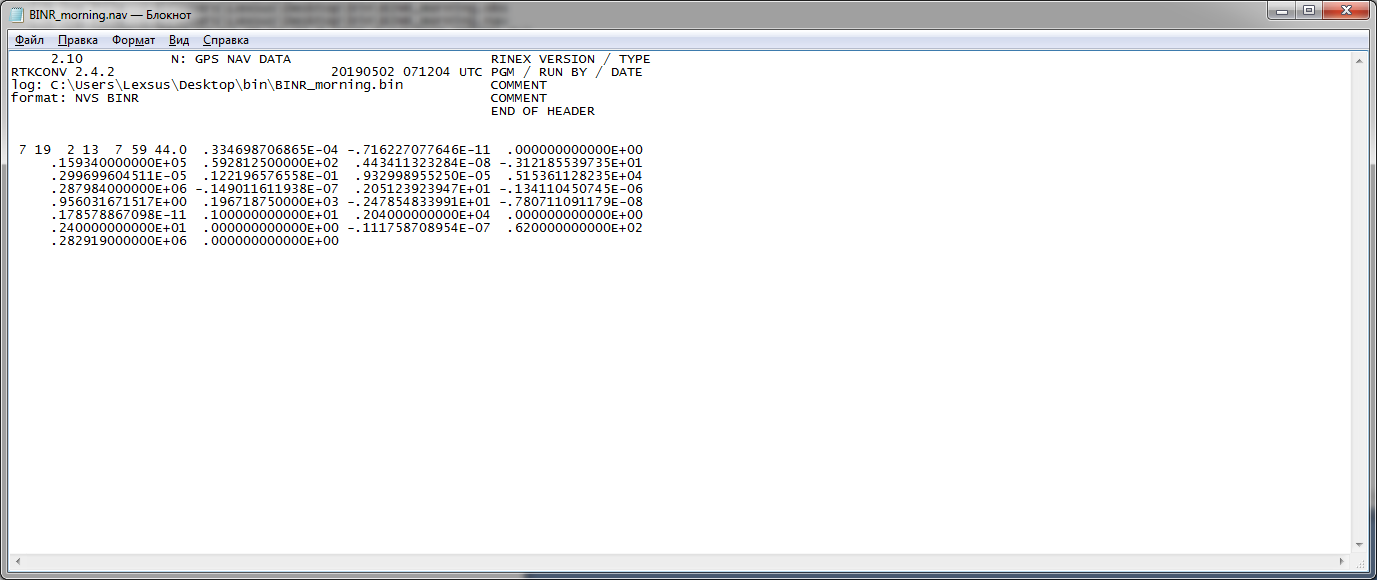


Рис.1 Эфемериды спутника №7 получены из .nav файла утренних наблюдений.

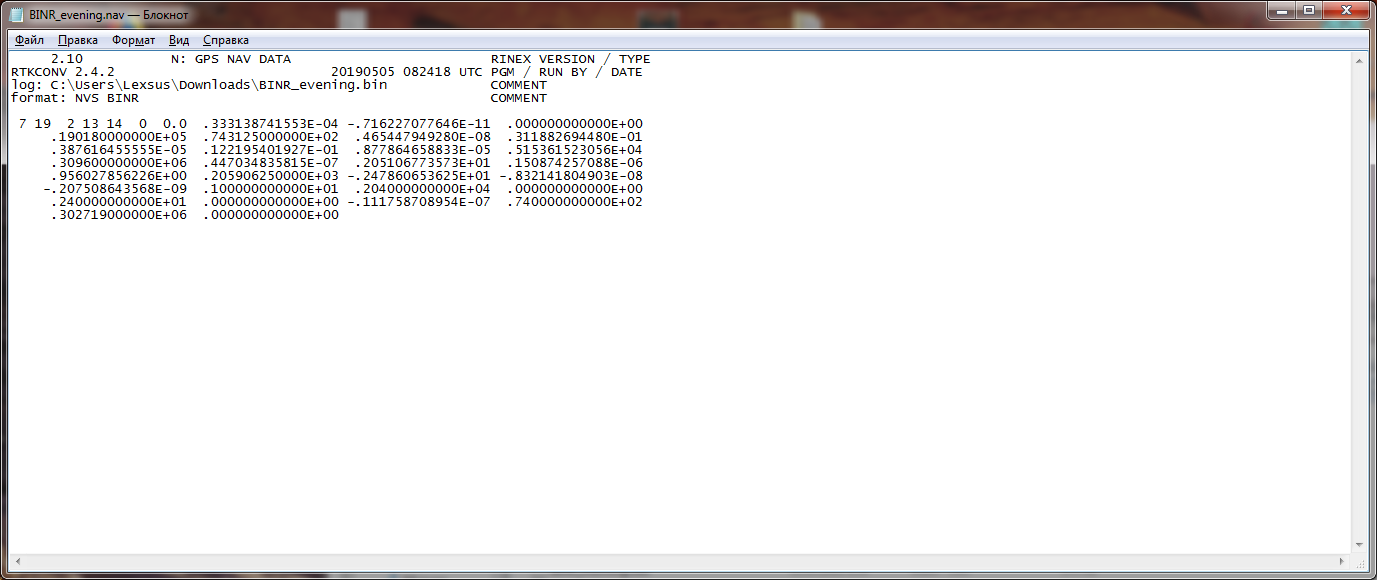


Рис.2 Эфемериды спутника №7 получены из .nav файла вечерних наблюдений.

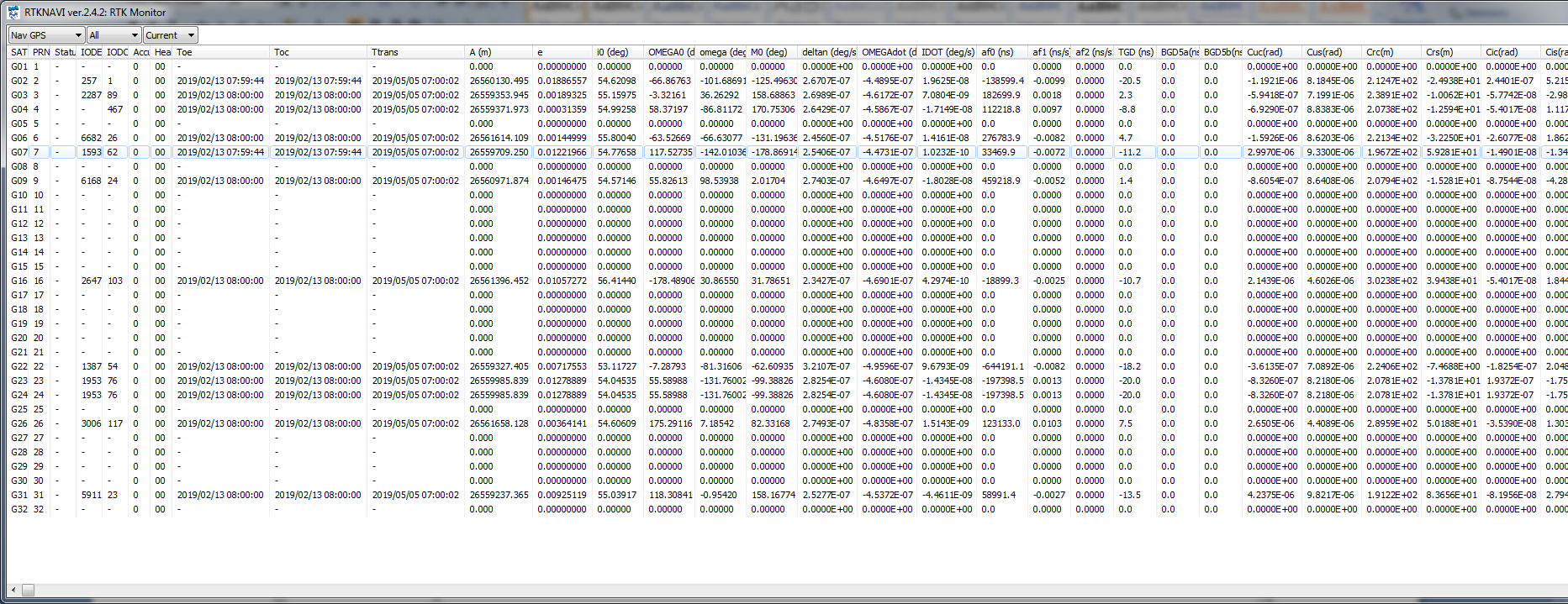


рис.3 Эфемериды, полученные с помощью подпрограммы RTKNAVI для утреннего наблюдения

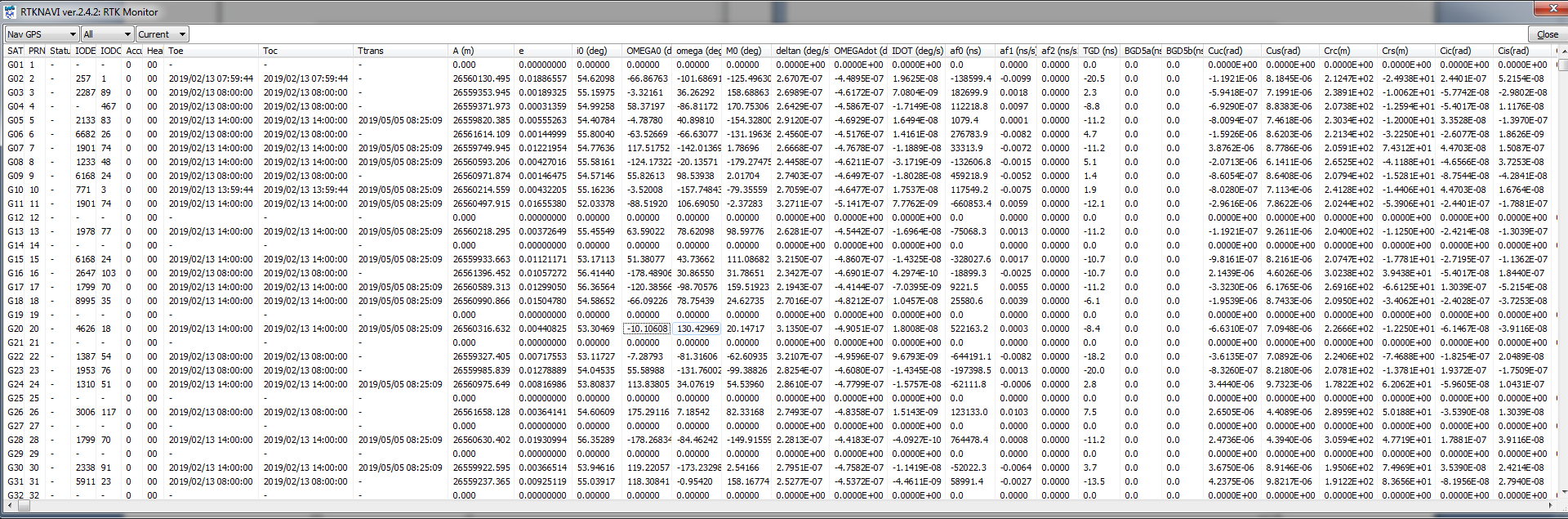


рис.4 Эфемериды, полученные с помощью подпрограммы RTKNAVI для вечернего наблюдения

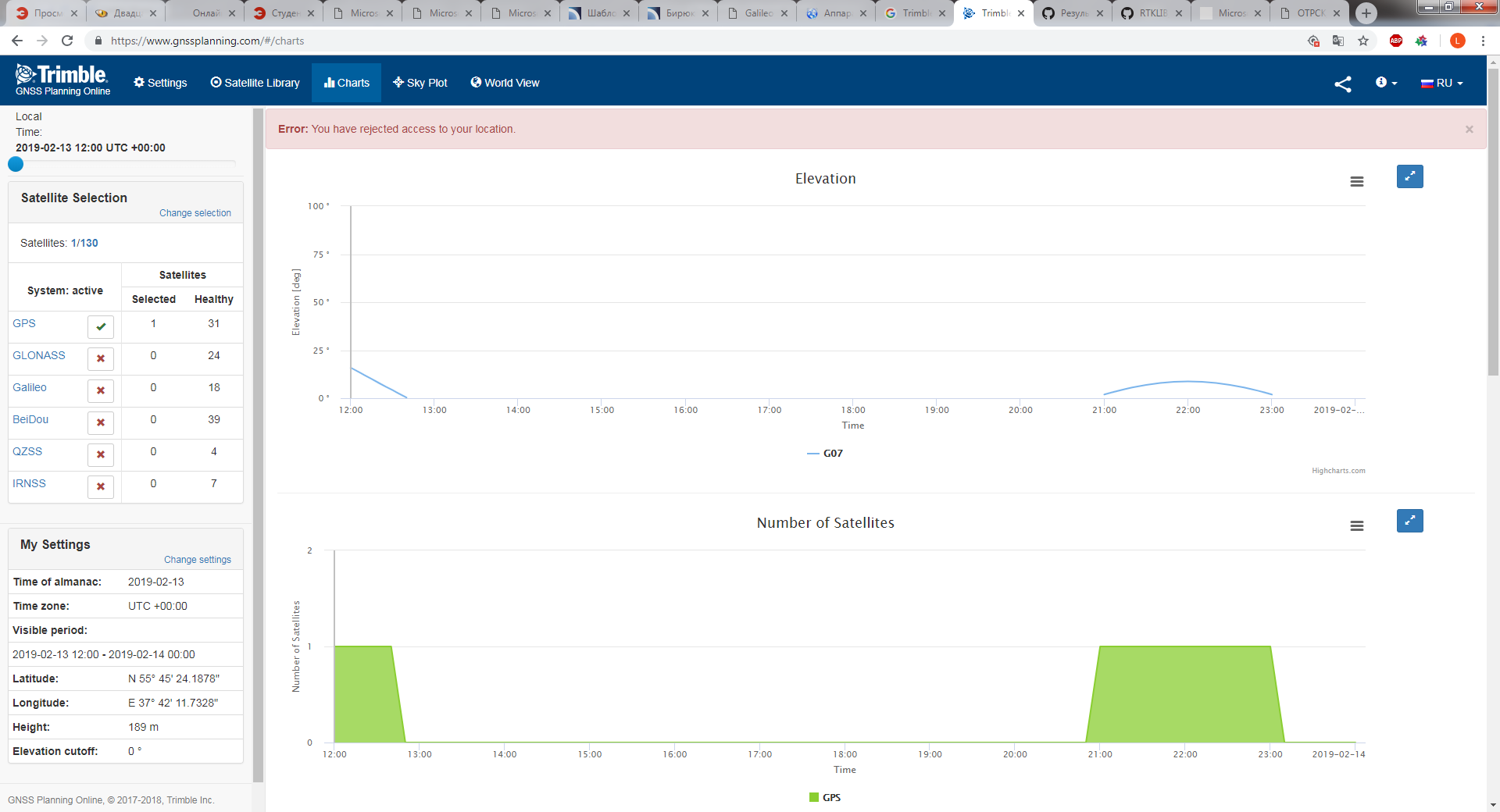


рис.5 зависимость угла места от времени на промежутке в 12 часов.

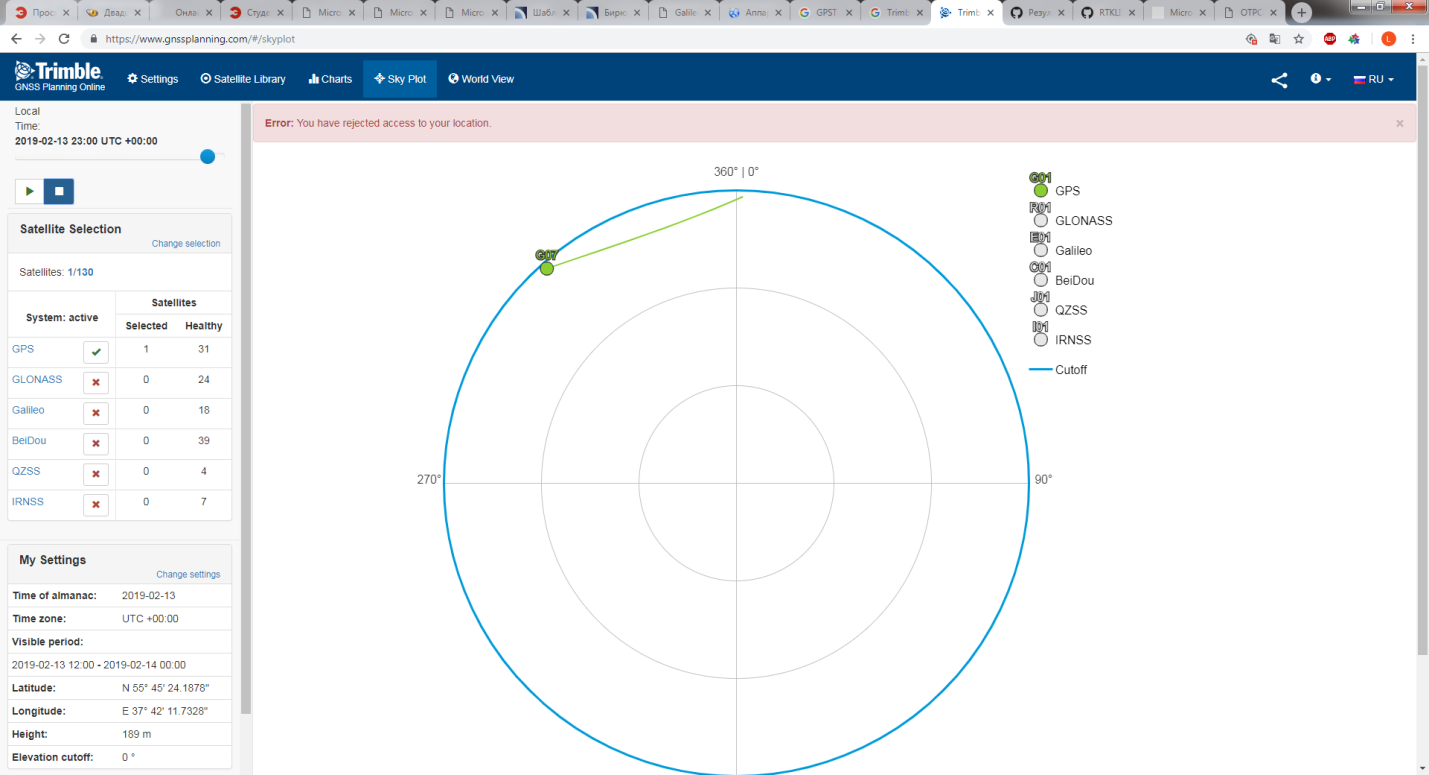
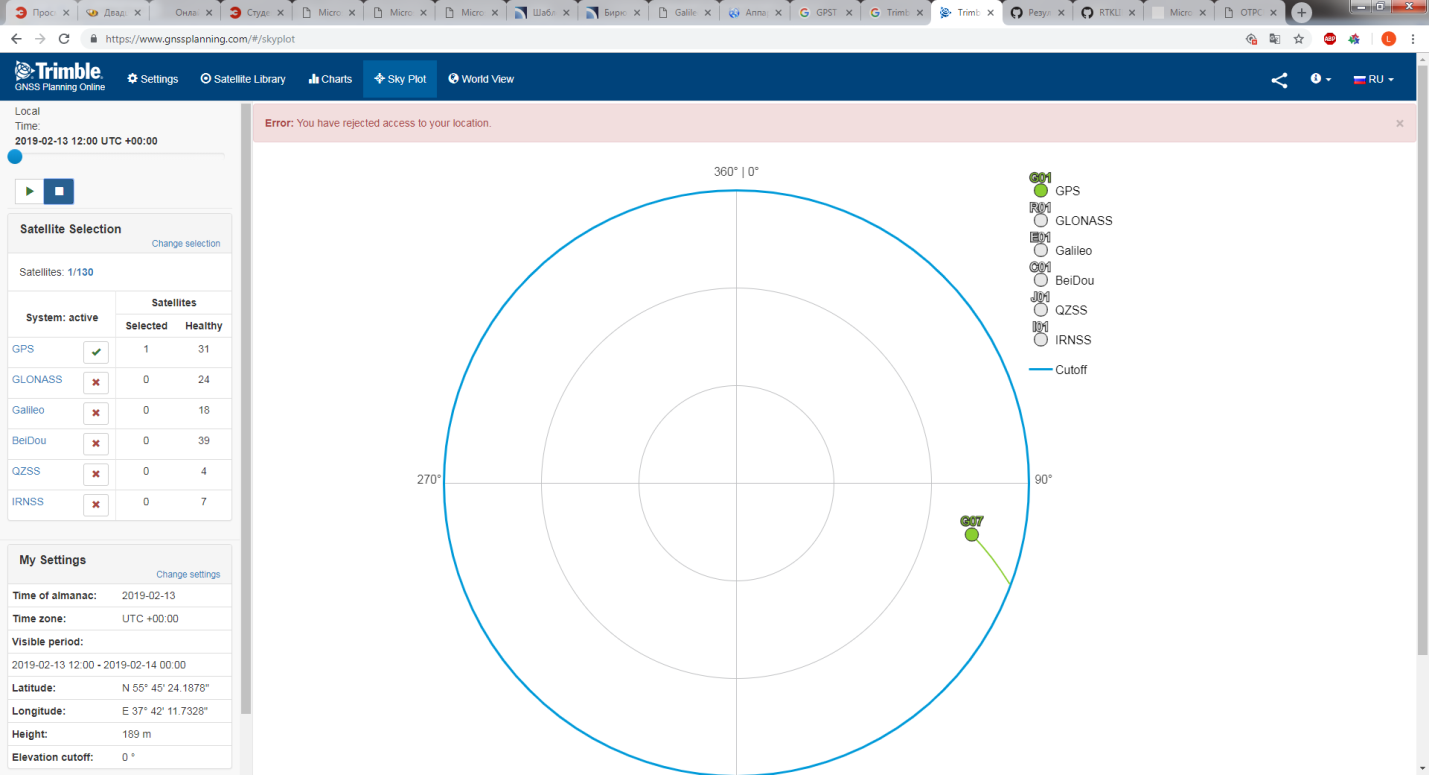


рис.6.Проекция на небо траектории спутника для разных промежутков времени(спутник находится за пределами области видимости большую часть моделируемого промежутка времени)

**Этап 2**

На предыдущем этапе получено решение навигационной задачи с помощью программы вторичной обработки измерений, например, RTKLIB. В процессе работы она рассчитывает положение спутников на соответствующий момент сигнального времени.

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.

Код программы выполняющей расчет и построение траектории на промежутке времени с 12:00-0:00 13.02.19 с шагом в 100 секунд:

clc;

clear all;

close all;

nu = 3.986004418e+14;

We = 7.2921151467e-5;

c = 299792458;

TOW=302418;

t=TOW

i0=deg2rad(54.77658);

A = 26559709.250;

e=0.01221966;

W0=deg2rad(117.52735);

Wdot=deg2rad(-4.4731e-07);

t0e=288000+18;

dn=deg2rad(2.5406e-7);

M0=deg2rad(-178.86914);

W=deg2rad(-142.01036);

idot=deg2rad(1.0232e-10);

af0=33469.9;

af1=-0.0072;

af2=0;

Cuc=2.997e-6;

Cus=9.33e-6;

Crc=1.9762e+2;

Crs=5.9281e+1;

Cic=-1.4901e-8;

Cis=0.13411e-6;

latitude=55.756727964;

longitude=37.703259108;

h=189.4054;

for j=1:432

n0 = sqrt(nu/(A^3));

tk=t-t0e;

n=n0+dn;

M = M0+n\*tk;

E=0;

for l=1:100

E=M+e\*sin(E);

end

v = atan2(sqrt(1-e^2)\*sin(E),cos(E)-e);

F=v+W;

du=Cus\*sin(2\*F)+Cuc\*cos(2\*F);

dr=Crs\*sin(2\*F)+Crc\*cos(2\*F);

di=Cis\*sin(2\*F)+Cic\*cos(2\*F);

u=F+du;

r=A\*(1-e\*cos(E))+dr;

i=i0+di+idot\*tk;

Xplan=r\*cos(u);

Yplan=r\*sin(u);

Omega=W0+(Wdot-We)\*(tk)-We\*t0e;

x=Xplan\*cos(Omega)-Yplan\*cos(i)\*sin(Omega);

y=Xplan\*sin(Omega)+Yplan\*cos(i)\*cos(Omega);

z=Yplan\*sin(i);

Resfix(j,:)=[x y z];

phi=We\*tk;

xeci=x\*cos(phi)-y\*sin(phi);

yeci=x\*sin(phi)+y\*cos(phi);

zeci=z;

ResECI(j,:)=[xeci yeci zeci];

[East, North, Up] = ecef2enu(x, y, z, latitude, longitude,h, wgs84Ellipsoid);

RtoS = sqrt(East^2 + North^2 + Up^2);

elevation(j) = rad2deg(-asin(Up/RtoS))+90;

azimuth(j) = atan2(East, North);

t=t+100;

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);

hold off;

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);

hold off;

figure;

polar (2\*pi-azimuth, elevation); %trimble дает график с осью по часовой стрелке, матлаб против

grid on; %

camroll(90);

V = sqrt((Resfix(1,1)-Resfix(2,1))^2+(Resfix(1,2)-Resfix(2,2))^2+(Resfix(1,3)-Resfix(2,3))^2)

R=sqrt((Resfix(1,1))^2+(Resfix(1,2)^2+(Resfix(1,3))^2))-6371000

R=sqrt((Resfix(2,1))^2+(Resfix(2,2)^2+(Resfix(2,3))^2))-6371000

Vx=(Resfix(1,1)-Resfix(2,1));

Vy=(Resfix(1,2)-Resfix(2,2));

Vz=(Resfix(1,3)-Resfix(2,3));

Результат работы программы:

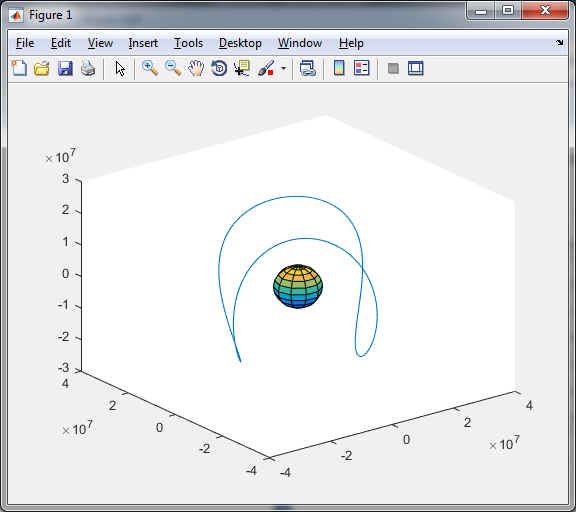


рис.7.траектория в СК ECEF WGS84

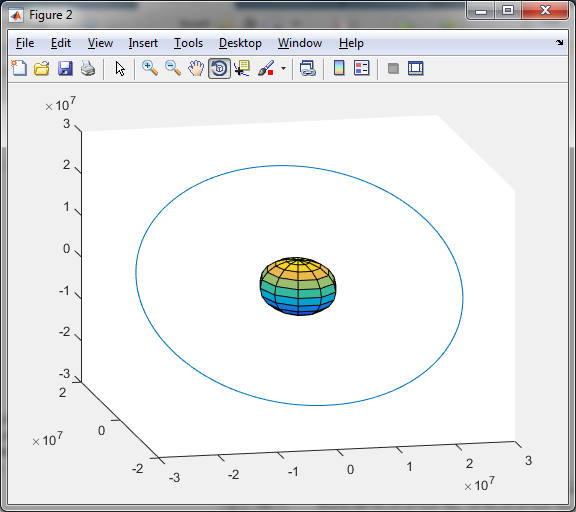


рис.8 траектория в ECI

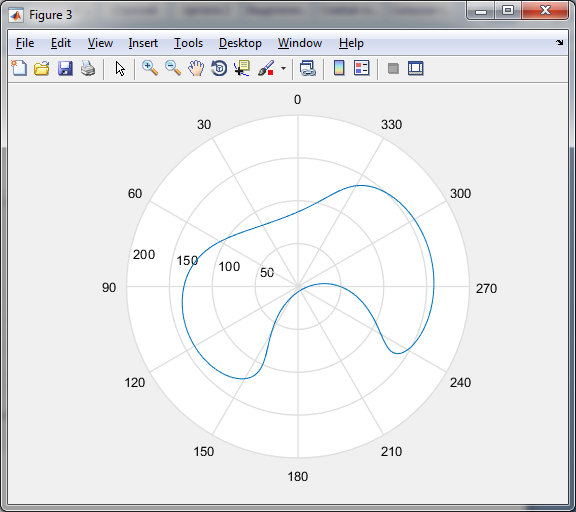


рис.9. skyview за 24 часа

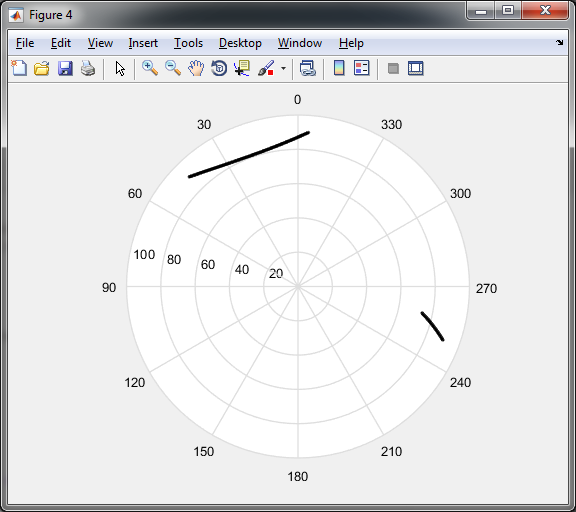


рис.9. skyview на заданный 12 часовой промежуток времени.(спутник на углах склонения больше 90 невозможно наблюдать с поверхности земли)

Сравнивая расчитанный skyview и полученный из "Trimble" отмечаем их полное сходство на построеном участке (для углов склонения меньше 90).

**Этап 3**

Требуется разработать на языке С/С++ функцию расчета положения спутника GPS на заданное время по шкале GPST, 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.

Разработанная программа на языке С++:

// ConsoleApplication12.cpp: определяет точку входа для консольного приложения.

//

#include<iostream>

#include<conio.h>

#include<math.h>

void main()

{

float nu = 3.986004418e+14;

float We = 7.2921151467e-5;

int c = 299792458;

float pi = 3.1415926535;

float TOW=302418;

float t = TOW;

float i0 = 54.77658 \* (pi / 180);

float A = 26559709.250;

float e = 0.01221966;

float W0 = 117.52735 \* (pi / 180);

float Wdot = -4.4731e-07 \* (pi / 180);

float t0e = 288000 + 18;

float dn = 2.5406e-7 \* (pi / 180);

float M0 = -178.86914 \* (pi / 180);

float W = -142.01036 \* (pi / 180);

float idot = 1.0232e-10 \* (pi / 180);

float af0 = 33469.9;

float af1 = -0.0072;

float af2 = 0;

float Cuc = 2.997e-6;

float Cus = 9.33e-6;

float Crc = 1.9762e+2;

float Crs = 5.9281e+1;

float Cic = -1.4901e-8;

float Cis = 0.13411e-6;

double n0, tk, n, M, E, v, F, du, dr, di, u, r, i, Xplan, Yplan, Omega, x, y, z,E2;

for (int j = 0; j < 100; j++) {

n0 = sqrt(nu / (pow(A, 3)));

tk = t - t0e;

n = n0 + dn;

M = M0 + n \* tk;

E2 = 00;

do {

E = E2;

E2 = M + e \* sin(E);

} while (abs(E2 - E) > 0.000001);

v = atan2(sqrt(1 - pow(e, 2)) \* sin(E), cos(E) - e);

F = v + W;

du = Cus \* sin(2 \* F) + Cuc \* cos(2 \* F);

dr = Crs \* sin(2 \* F) + Crc \* cos(2 \* F);

di = Cis \* sin(2 \* F) + Cic \* cos(2 \* F);

u = F + du;

r = A \* (1 - e \* cos(E)) + dr;

i = i0 + di + idot \* tk;

Xplan = r \* cos(u);

Yplan = r \* sin(u);

Omega = W0 + (Wdot - We) \* (tk)-We \* t0e;

x = Xplan \* cos(Omega) - Yplan \* cos(i) \* sin(Omega);

y = Xplan \* sin(Omega) + Yplan \* cos(i) \* cos(Omega);

z = Yplan \* sin(i);

t = t + 0.1;

std::cout.precision(5);

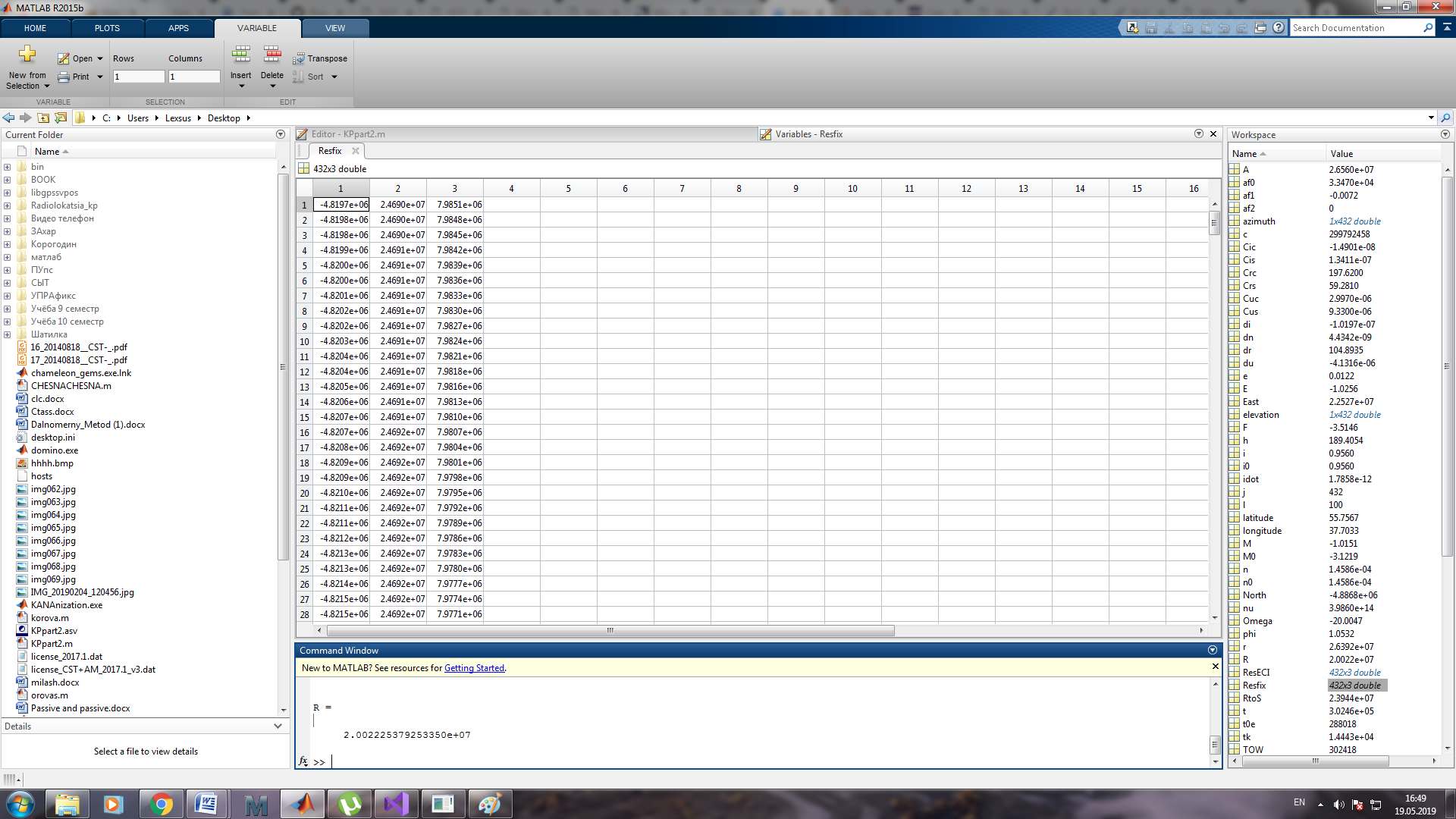
std::cout << x << " " << y << " " << z<< std::endl;

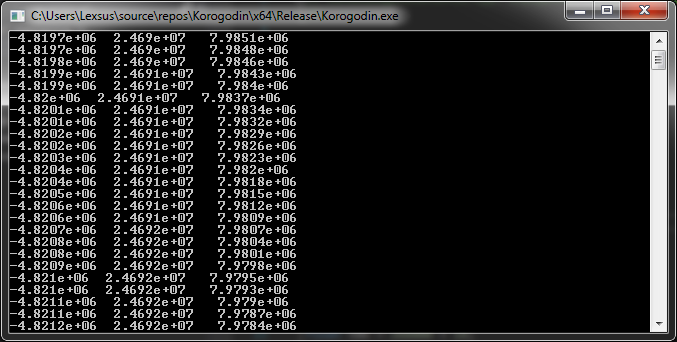
}

\_getch();

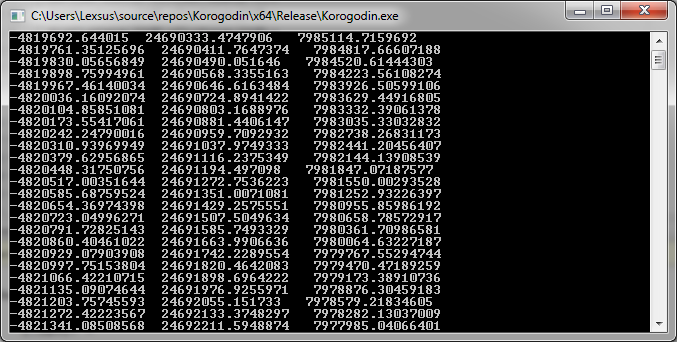
}

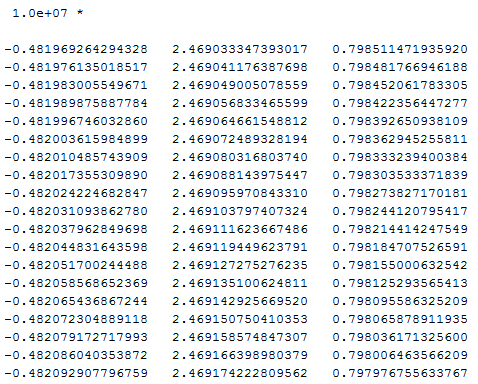
Тест и сравние решений полученных в MatLab и С++ с шагом 0.1 секунда

 Матлаб

 С++

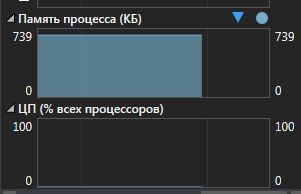
Значения совпадают, максимальное отклонение 0.0001е+6 выявлено в координате х. Связано с различным форматом отображения в разных средах

 С++

Матлаб

Максимальное расхождение составляет 0.005 метров среднее 0.0005 метров по всем координатам.

Запуск в цикличном режима. Утечки памяти не выявлено



Длительность выполнения функции(одиночный расчет, без вывода в консоль)=255мкс

