Pr\_coord1

llh.lat=0.2443;

llh.lon=0.872664626;

llh.h=215;

ab.a=6378137.0;

ab.b=6356752.314;

[XYZ] = ECEFLLH\_N(llh,ab);

disp('ECEF координаты:');

disp(XYZ);

ECEFLLH\_N

function [XYZ] = ECEFLLH\_N(llh,ab)

%Имя функции: ECEFLLH\_N

%Назначение функции: преобразование координат из географической системы в прямоугольную

%Входные данные:

%llh.lon-долгота;

%llh.lat-широта;

%llh.h-высота;

%ab.a-большая полуось эллипсоида;

%ab.b- малая полуось эллипсоида в WGS-84;

%Выходные данные:

%XYZ.x,XYZ.y,XYZ.z- координаты X, Y, Z соответственно в ECEF

% Справочные данные:

%ECEF- прямоугольная геоцентрическая система координат

%a=6378137.0 (м)- большая полуось эллипсоида для WGS-84;

%b=6356752.314 (м)- малая полуось эллипсоида для WGS-84;

%A\_PZ90\_M =6 378 136 (м)- большая полуось эллипсоида для ПЗ 90;

%B\_PZ90\_M = 6356751.36174 (м)- малая полуось эллипсоида для ПЗ 90;

a2=ab.a\*ab.a;

b2=ab.b\*ab.b;

r=a2/sqrt(a2\*cos(llh.lat)\*cos(llh.lat)+b2\*sin(llh.lat)\*sin(llh.lat));

XYZ.x=(r+llh.h)\*cos(llh.lat)\*cos(llh.lon);

XYZ.y=(r+llh.h)\*cos(llh.lat)\*sin(llh.lon);

XYZ.z=(b2/a2\*r+llh.h)\*sin(llh.lat);

LLHECEF\_N

function [llh] = LLHECEF\_N(XYZ,ab)

%Имя функции: LLHECEF\_N

%Назначение функции: преобразование координат из прямоугольной системы в

%Входные данные:

%XYZ.x,XYZ.y,XYZ.z- координаты X, Y, Z соответственно в ECEF

%ab.a-большая полуось эллипсоида;

%ab.b- малая полуось эллипсоида в WGS-84;

%Выходные данные:

%llh.lon-долгота;

%llh.lat-широта;

%llh.h-высота;

% Справочные данные:

%ECEF- прямоугольная геоцентрическая система координат

%a=6378137.0 (м)- большая полуось эллипсоида для WGS-84;

%b=6356752.314 (м)- малая полуось эллипсоида для WGS-84;

%A\_PZ90\_M =6 378 136 (м)- большая полуось эллипсоида для ПЗ 90;

%B\_PZ90\_M = 6356751.36174 (м)- малая полуось эллипсоида для ПЗ 90;

a=6378137.0;

b=6356752.314;

a2=ab.a\*ab.a;

b2=ab.b\*ab.b;

xy = sqrt(XYZ.x\*XYZ.x + XYZ.y\*XYZ.y);

thet = atan(XYZ.z\*ab.a/(xy\*ab.b));

esq = 1.0-b2/a2;

epsq = a2/b2-1.0;

llh.lat = atan((XYZ.z+epsq\*ab.b\*(sin(thet)^3))/(xy-esq\*ab.a\*(cos(thet)^3)));

llh.lon = atan2(XYZ.y,XYZ.x);%!

if llh.lon < 0

llh.lon = 2\*pi + llh.lon;

end ;

r = a2/sqrt(a2\*cos(llh.lat)\*cos(llh.lat) + b2\*sin(llh.lat)\*sin(llh.lat));

llh.h = xy/cos(llh.lat)-r;

end

PR\_COORD2

%Имя m- файла:Pr\_Coord2.m

ab.a=6378137.0;

ab.b=6356752.314;

XYZ.x=3.9790e+06;XYZ.y=4.7419e+06;

XYZ.z=1.5328e+06;

[llh] = LLHECEF\_N(XYZ,ab)

TOP\_COORD

function [top] = top\_coord(rec\_llh, rec\_xyz, nlo\_xyz)

% Имя функции: top\_coord

%Назначение функции: расчет топоцентрических координат объекта по заданным

%географическим (долгота, широта, высота) и геоцентрическим (x, y, z)

%координатам приемника и геоцентрическим координатам объекта (x, y, z)

% Входные данные:

% rec\_llh.lat - широта (рад) приемника;

%rec\_llh.lon -- долгота (рад) приемника;

%rec\_llh.h- высота (м) приемника;

%прямоугольные геоцентрические координаты приемника (м):

% rec\_xyz.x

% rec\_xyz.y

%,, rec\_xyz.z

%прямоугольные геоцентрические координаты объекта (м):

% ns.x - координата x;

% ns.y -координата y;

% ns.z- координата z ;

% Выходные данные:

% top.s - проекция вектора дальности на ось (м) , направленную на Юг (South)

% top.e - проекция вектора дальности на ось (м) , направленную на Восток (East)

% top.z - проекция вектора дальности на ось (м) , направленную в Зенит

% top.daln - дальность до объекта (м)

% top.az - угол азимута объекта (градус)

% top.el - угол видимости объекта (градус)

rx = nlo\_xyz.x - rec\_xyz.x;

ry = nlo\_xyz.y - rec\_xyz.y;

rz = nlo\_xyz.z - rec\_xyz.z;

r\_sat = sqrt(rx\*rx + ry\*ry + rz\*rz);

r\_rec = sqrt((rec\_xyz.x)^2 + (rec\_xyz.y)^2+ (rec\_xyz.z)^2);

top.r = r\_sat;

rx1 = rx; ry1 = ry; rz1 = rz;

sin\_lat = sin(rec\_llh.lat);

cos\_lat = cos(rec\_llh.lat);

sin\_lon = sin(rec\_llh.lon);

cos\_lon = cos(rec\_llh.lon);

% Projections of vector of range in topocentric coordinate system:

top.e = -sin\_lon \* rx1 + cos\_lon \* ry1;

top.s = cos\_lon \* sin\_lat \* rx1 + sin\_lon \* sin\_lat \* ry1 - cos\_lat \* rz1;

top.z = cos\_lat \* cos\_lon \* rx1 + cos\_lat \* sin\_lon \* ry1 + sin\_lat \* rz1;

% azimut: отсчет по часовой стрелке от оси направленной на Север (N or -S) (-top.s)

eps = 10e-10;

if ( (abs(top.e) < eps) || (abs(top.s) < eps))

top.az = 0.0;

else

top.az = atan2(top.e,-top.s);

end;

if (top.az < 0.0)

top.az = top.az + pi \* 2;

end;

% elevation:

cos\_el\_top = (rec\_xyz.x \* rx + rec\_xyz.y \* ry + rec\_xyz.z \* rz) / (r\_sat \* r\_rec);

if ( cos\_el\_top >= 1.00 )

el = 0.0;

else

if ( cos\_el\_top <= -1.00 )

el = pi;

else

el = acos(cos\_el\_top);

end;

end;

top.el = pi / 2.0 - el;

PRIM\_TOP\_COORD

%Имя m-файла: prim\_top\_coord.m

%Пример расчета

a=6378137.0; b=6356752.314; % для WGS-84;

% Коэффициенты перевода градусов в радианы и обратно

A2R = pi/180;

R2A = 180/pi;

%Входные данные координаты, например, приемника

rec\_deg.lon = 45.5042;

rec\_deg.lat = 73.2583;

rec\_deg.h = 100;

%Входные данные координат объекта

nlo\_deg.lon = 40.9194 ;

nlo\_deg.lat = 78.8681 ;

nlo\_deg.h = 200;

%Преобразование градусов в радианы

rec\_llh.lon = rec\_deg.lon \* A2R;

rec\_llh.lat = rec\_deg.lat \* A2R;

rec\_llh.h = rec\_deg.h;

nlo\_llh.lon = nlo\_deg.lon \* A2R;

nlo\_llh.lat = nlo\_deg.lat \* A2R;

nlo\_llh.h = nlo\_deg.h;

%Преобразование координат приемника и объекта систему ECEF

[rec\_xyz] = ECEFLLH(a, b, rec\_llh);

[nlo\_xyz] = ECEFLLH(a, b, nlo\_llh);

%Преобразование координат приемника и объекта в топоцентрическую

%систему координат

[top] = top\_coord(rec\_llh, rec\_xyz, nlo\_xyz);

%Вывод данных приемника в топоцентрической системе координат

fprintf('e=%22.16e s=%22.16f z=%22.16f az=%f el=%f r=%f \n', top.e, top.s, top.z, top.az\*R2A, top.el\*R2A, top.r);

%Вывод данных объекта в топоцентрической системе координат

fprintf('e=%22.16e s=%22.16f z=%22.16f az=%f el=%f r=%f \n', top.e, top.s, top.z, top.az\*R2A, top.el\*R2A, top.r);

ECEFLLH

function [R] = ECEFLLH(a, b, llh)

%Имя функции: ECEFLLH

%Назначение- вариант функции ECEFLLH\_N

a2 = a \* a;

b2 = b \* b;

n = a2 / sqrt(a2 \* cos(llh.lat)\*cos(llh.lat) + b2 \* sin(llh.lat) \* sin(llh.lat));

R.x = (n + llh.h) \* cos(llh.lat) \* cos(llh.lon);

R.y = (n + llh.h) \* cos(llh.lat) \* sin(llh.lon);

R.z = (b2 / a2 \* n + llh.h) \* sin(llh.lat);