Institute for Astronomical and Physical Geodesy

ESPACE: Orbit Mechanics

Exercise 1:

«Keplerian Orbits in Space-fixed, Earth-fixed and Topocentric systems»

WS14/15

Prepared by Aleksei Petukhov

aleksei.petukhov@tum.de

Contents

Given Data

Orbits of several satellites are given in an inertial, geocentric reference system (space-fixed) by the Keplerian orbital elements: semi major axis a, eccentricity e, inclination i, right ascension of the ascending node Ω, and argument of the perigee ω.

Satellite	a [km]	e	i [deg]	Ω [deg]	ω [deg]
GOCE	6629	0.004	96.6	257.7	144.2
GPS	26560	0.01	55	60 0	
MOLNIYA	26554	0.7	63	245	270
GEO	geostationary	0	0	0	0
MICHIBIKI	geosynchronous	0.075	41	195	270

For the following computations precession, nutation, polar motion and variations in the length of day are neglected. The Earth fixed reference system then rotates with an angular rate of $\omega_{Earth} = 2\pi/86164s$ about the $\mathbf{e}_3$-axis of the inertial space-fixed reference system. At the time $t_0 = 0$ sec both systems are axis parallel and the satellites are in their perigee, except for MICHIBIKI which mean anomaly at $t_0 = 0$ sec is 30 degrees.

Geocentric gravitational constant $GM = 398.6005 \cdot 10^{12} m^3/s^2$
Earth's radius $R_E = 6371 \cdot 10^3$ m

Brief overview

This exercise gives a practical understanding of orbital mechanics concepts, such as 6 Keplerian Orbital Elements, which are sufficient to describe every orbit. Exercise provides with an opportunity to work within different reference frames – Space-fixed, Earth-fixed and Topocentric system.

Matlab environment is helpful for calculating orbital parameters and plotting the orbits.

The tutorial is carried out under the supervision of Prof. Dr. Urs Hugentobler and Ye Hao.

Problem 1

Create a MATLAB-function kep2orb.m that computes polar coordinates r (radius) and ν (true anomaly) based on input orbital elements. Formulate your program in a way that the time t can be used as input parameter.

To compute *geocentric radius* we use the following formula:

$$r = a \cdot (1 - e \cdot \cos E) \quad (1)$$

where r is the distance from the center of the Earth to the satellite, a – semi major axis and E is the *eccentric anomaly*.

To solve equation (1) we need to obtain E by solving the Kepler's equation. Since its transcendental it cannot be solved analytically - only iteratively for given *mean anomaly M*.

Mean anomaly can be found with the following equation:

$$M = n \cdot (t - T) \quad (2)$$

where n is the *mean motion*, which is obtained from Kepler's 3rd law.

Thus we get:

$$n = \sqrt{G \cdot M / a^3} \quad (3)$$

where $G \cdot M = 398.6005 \cdot 10^{12}$ m^3/s^2 – *geocentric gravitational constant*

and

$$T = 2\pi \cdot \sqrt{a^3 / G \cdot M} \quad (4)$$

(*) for the Michibiki satellite, M at the initial time t_0 is 30 degrees, therefore we define its M as $M+\pi/6$

We iterate the following set of equations:

Initial value of E equals M:

$$E_0 = M \quad (5)$$

$$E_{i+1} = E_0 + e \cdot \sin E \quad (6)$$

We can use these formulas since we have relatively small eccentricities and recursion converges rapidly.

The cycle continues while the difference between the adjacent values of E is larger than 10^{-6}.

Once the desirable value of E obtained we can use it to calculate *geocentric radius r* and *true anomaly v*.

True anomaly v is calculated using the following formula:

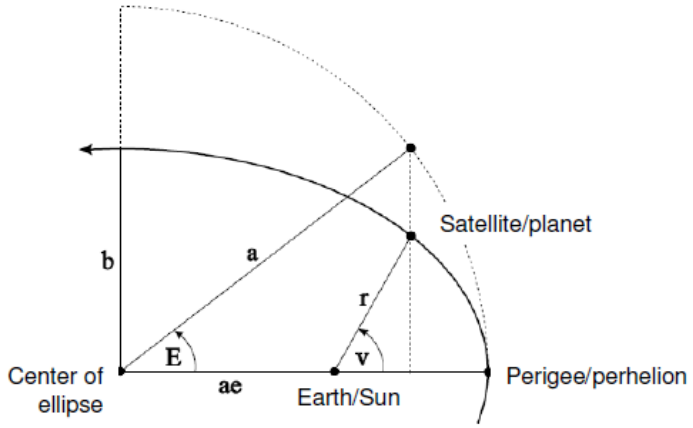
$$\tan\frac{\nu}{2} = \sqrt{\frac{1+e}{1-e}}\tan\frac{E}{2} \qquad (7)$$

Problem 2

Plot the orbit for the 5 satellites in the orbital plane for one orbital revolution.

Time period of satellites is calculated using formula (4).

We enter Keplerian orbital elements to function kep2orb thus obtaining values of r and *v* and plot the orbits for one revolution.



Keplerian orbital elements required are:

a – semi major axis (the longest axis of an ellipse)

e – eccentricity. This parameter defines the shape of the orbit. For elliptical orbits 0< e < 1, for circular orbit e=0, e=1 for parabolic orbits, e>1 for hyperbolic orbits.

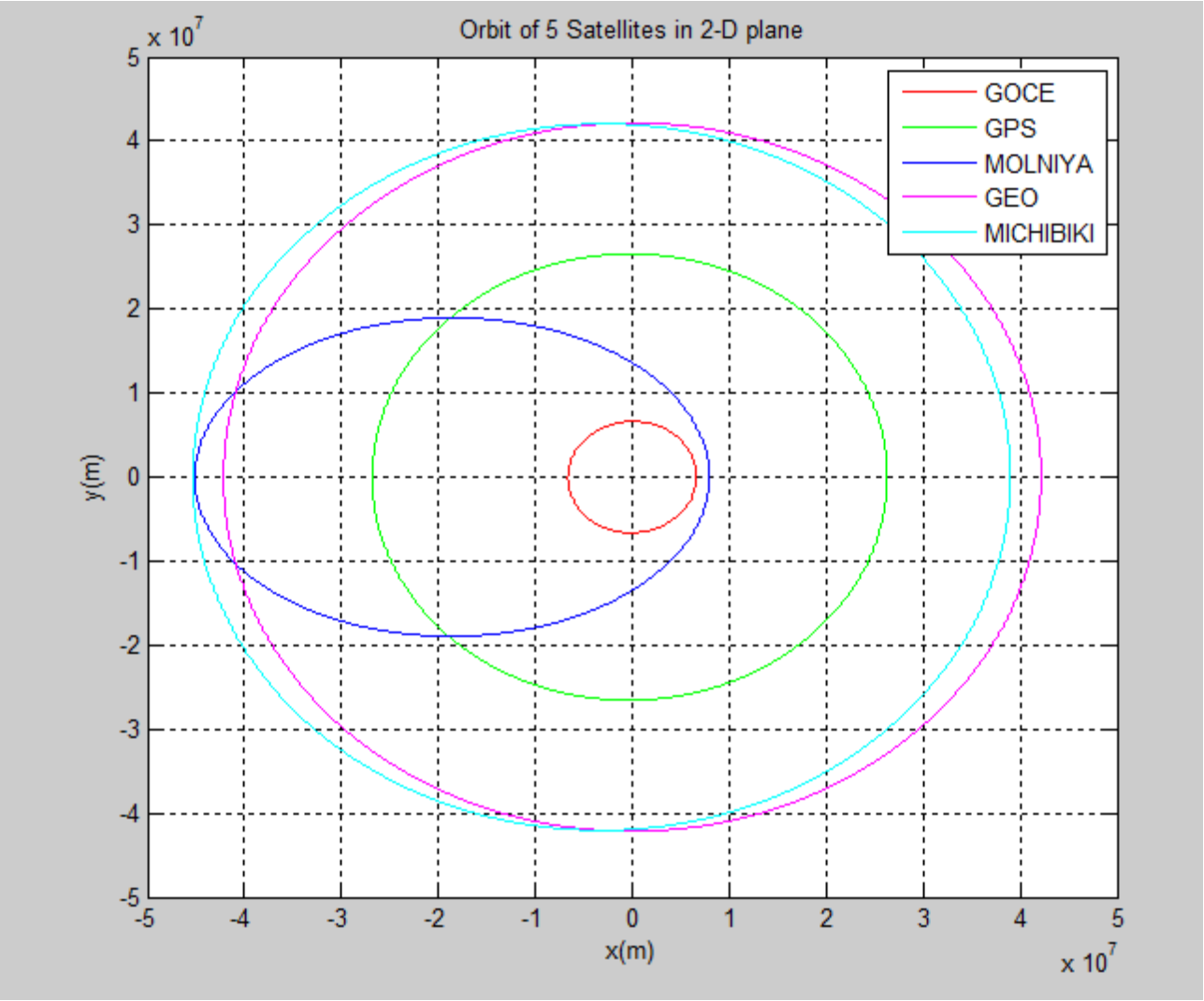
ν – true anomaly. This is the angle measured from the perihelion to the position vector of the satellite. It determines the position of a satellite on the orbital plane.

E – eccentric anomaly. It's an angle between the major axis and the line connecting the center of an ellipse and the vertical projection of satellite's position on the auxillary axis.

Every satellite will have a projection to the equatorial plane. Polar coordinates are:

$x = r \cdot \cos\nu$ and $y = r \cdot \sin\nu$

Plotting from 1 to T gives us the following picture:

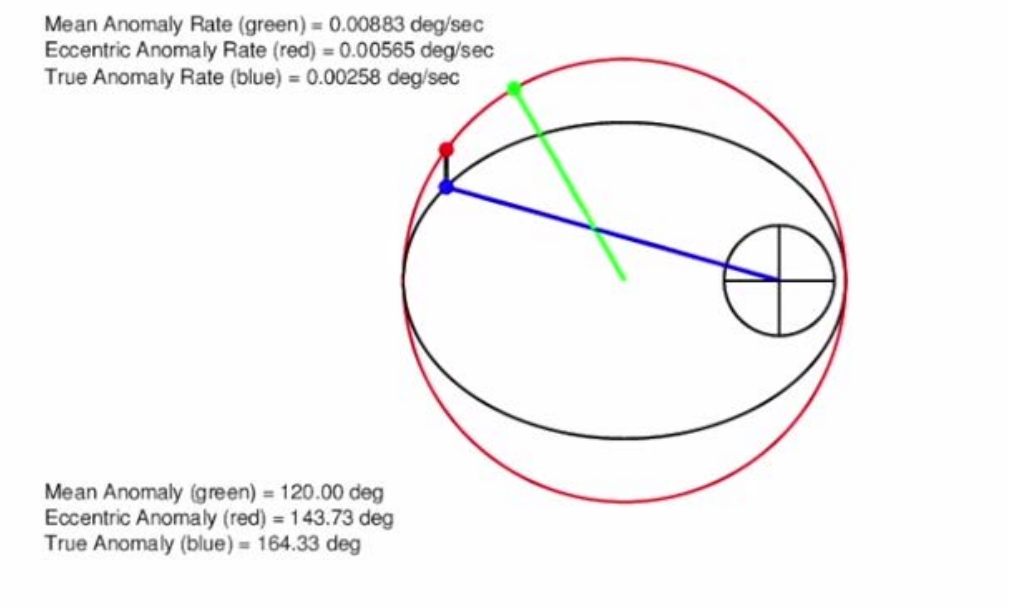


Picture 1: Orbits of 5 satellites

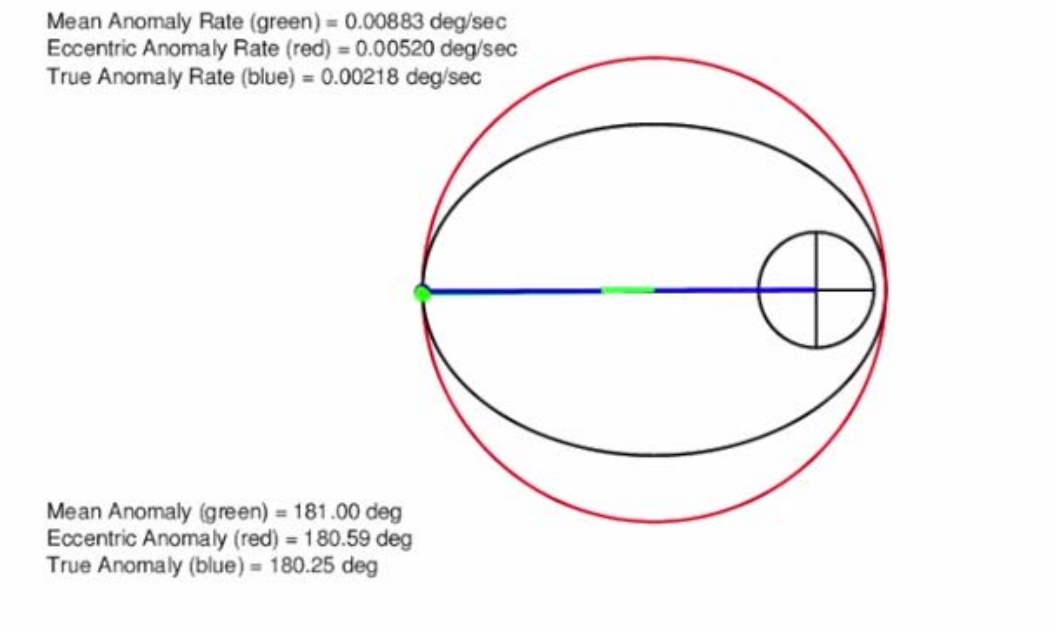
4 out of 5 satellites have elliptical orbits which are consistent with their eccentricities values. GEO satellite, however, has a circular orbit since its eccentricity is zero.

Problem 3

Plot the mean anomaly M, the eccentric anomaly E, and the true anomaly ν as well as the difference ν – M for one orbital revolution for the GPS satellite and the Molniya satellite.



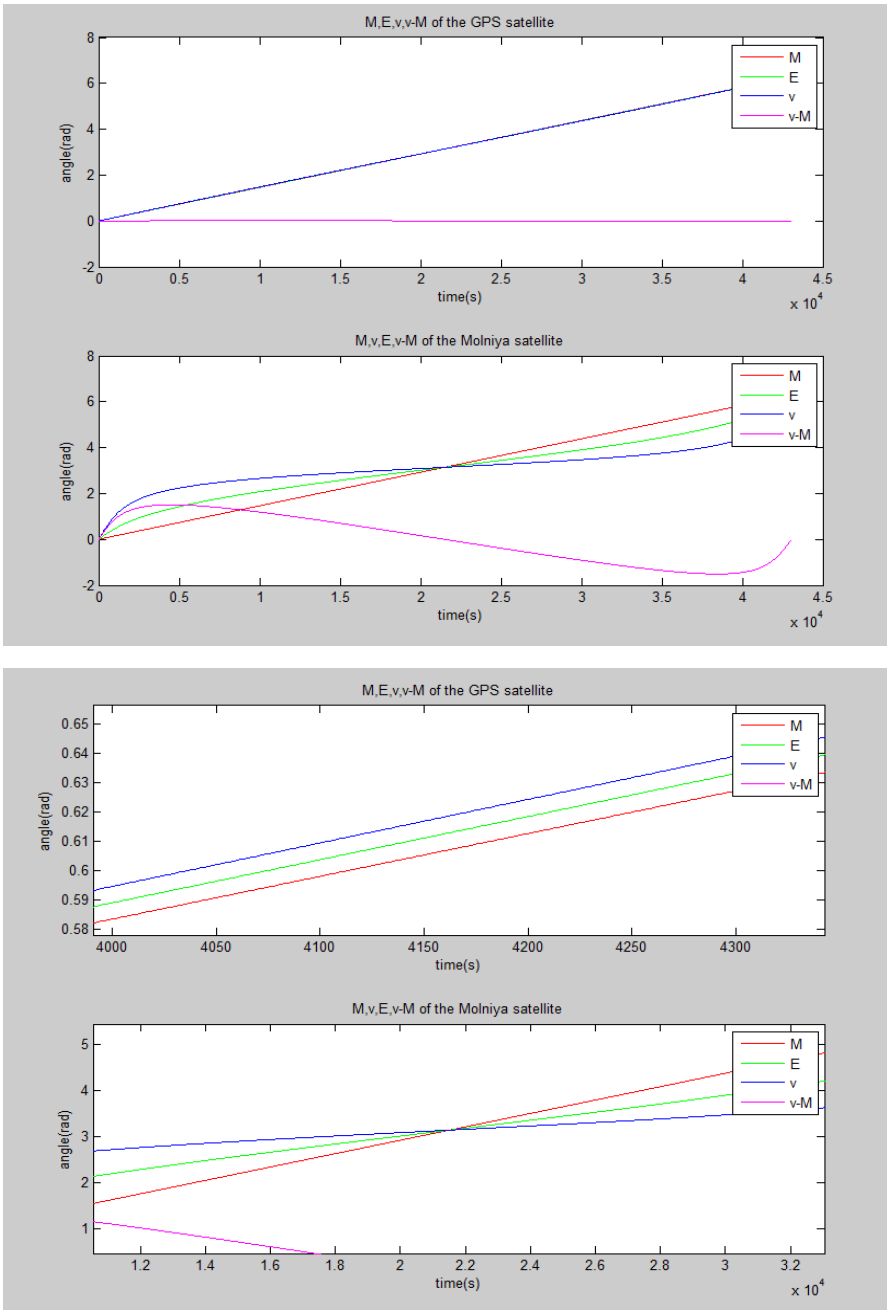
Picture 2: Moving E, ν, M



Picture 3: Moving E, ν, M at the apogee point. All equal to each other.

True anomaly isn’t moving at a constant rate when we deal with elliptical orbits. The rate of true anomaly is the slowest at the apogee and the fastest at the perigee. We need mean anomaly M to be able to predict the position of the satellite at different times. Mean anomaly is moving along the auxillary circle at a constant rate. Eccentric anomaly E is required for associating the true anomaly with the mean anomaly.

As we can see from the plotted graphs, the values of E, ν and M are the same at the perigee (0 rad) and the apogee (π rad).



Picture 4: M, e, ν of GPS and Molniya Satellites. Second picture is zoomed in.

As we can see, values of E, ν, and M for *GPS satellite* are almost always the same (at least they are very close), that is due to the GPS has almost circular orbit (due to very small eccentricity)

Problem 4

Create a MATLAB-function kep2cart.m that uses kep2orb.m, which transforms Keplerian elements to position and velocity in an inertial (space-fixed) system.

Components of the position vector and the velocity vector in the orbit-fixed coordinate system (first axis to the perigee, second axis in the orbital plane, in direction of motion, third axis perpendicular to the orbital plane) can be found using the following equations:

$$\mathbf{r}_b = r(\cos\nu, \sin\nu, 0) \tag{8}$$

$$\dot{\mathbf{r}}_b = \sqrt{\frac{GM}{a(1-e^2)}}(-\sin\nu, e+\cos\nu, 0) \tag{9}$$

Where r and ν are calculated within the function kep2orb.

Next step is to rotate obtained position and velocity (in inertial system) vectors into an equatorial coordinate system. We use the following equations:

$$\mathbf{r} = \mathbf{R}_3(-\Omega)\mathbf{R}_1(-i)\mathbf{R}_3(-\omega)\mathbf{r}_b \tag{10}$$

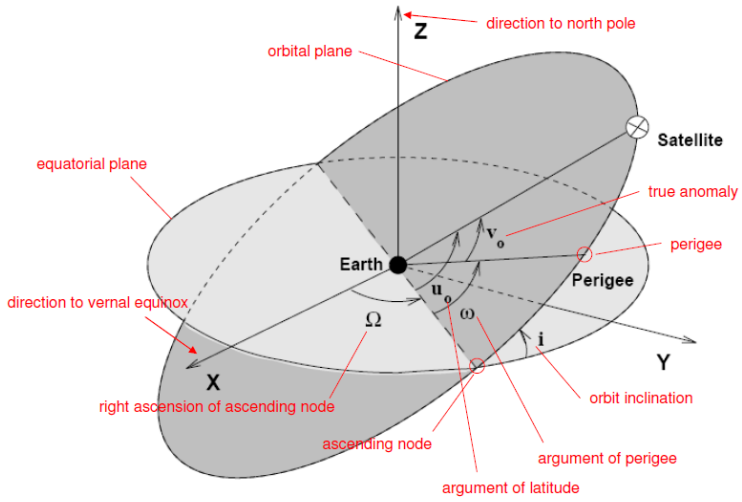
$$\dot{\mathbf{r}} = \mathbf{R}_3(-\Omega)\mathbf{R}_1(-i)\mathbf{R}_3(-\omega)\dot{\mathbf{r}}_b \tag{11}$$

The order of multiplication is from the right to the left.

Where R_1 and R_3 are the rotational matrices defined as follows:

$$R_1(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \qquad R_3(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

We use the following Keplerian elements as arguments in these rotation matrices instead of θ:



ω – argument of perigee,

which basically determines the location of the periapsis on the orbital plane

i – inclination of the orbital plane

This parameter determines the tilt of an orbital plane with respect to the equatorial plane

Ω – right ascention of ascending node, determines the swivel of an orbital plane, measured along the equatorial plane from the vernal equinox to the ascending node.

i, Ω determine the orbital plane orientation.

Magnitude of the velocity we simply determine as $\sqrt{x^2+y^2+z^2}$

Problem 5

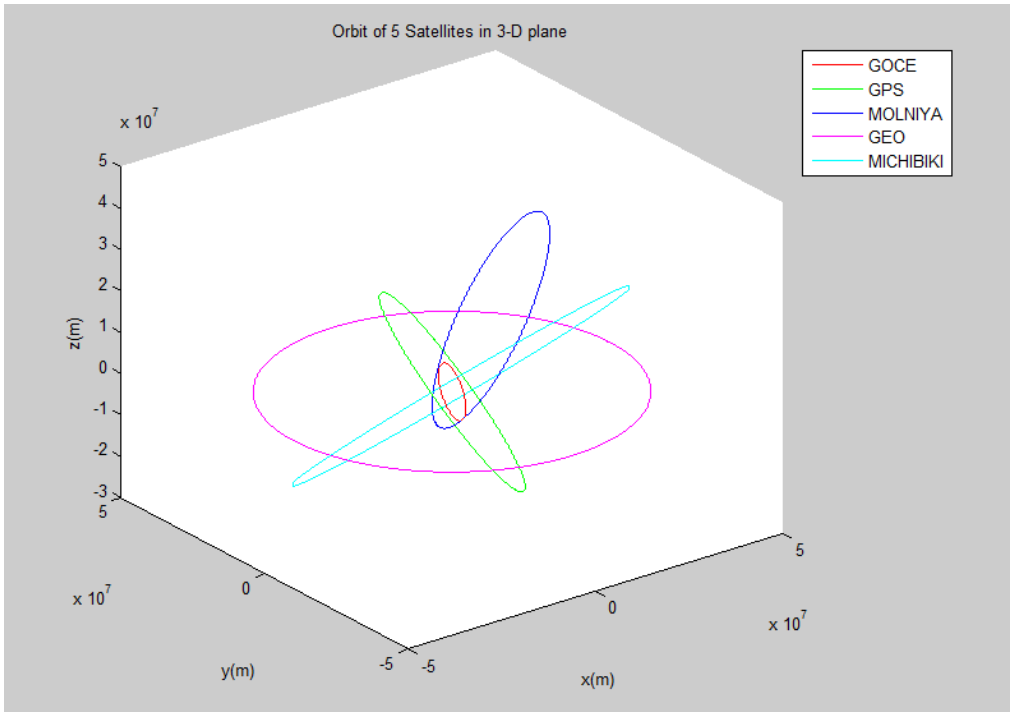
Compute position and velocity vectors of the 5 satellites for a period of one day. Assume true anomaly ν=0 for the beginning of the day. Visualize your results. Plot the trajectory in 3D and 2D (projection to x-y, x-z and y-z planes) as well as a time series of the magnitude of velocity.

We define input parameter t as one day which is 86400 seconds.

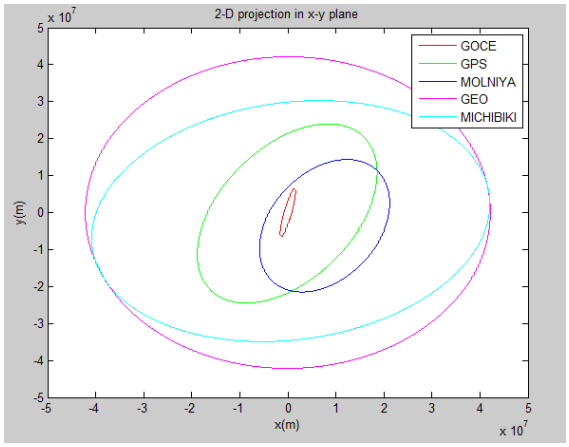
To plot the trajectories in 3D it is necessary to convert orbital elements from degrees to radians:

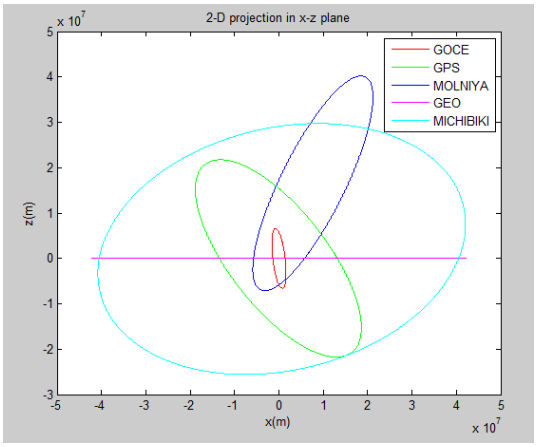
```
perigee_arg  = degtorad([144.2 0 270 0 270]);
inclination_i = degtorad([96.6 55 63 0 41]);
ascending_node= degtorad([257.7 60.0 245 0 195]);
t=1:86400;
```

Using kep2cart function we plot the trajectories in 3D

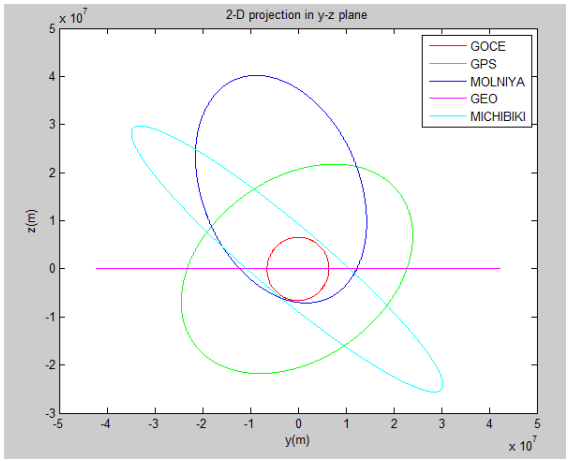


Picture 5: Orbits in space fixed coordinate system. 3D-view





Picture 6,7: x-y, x-z planes

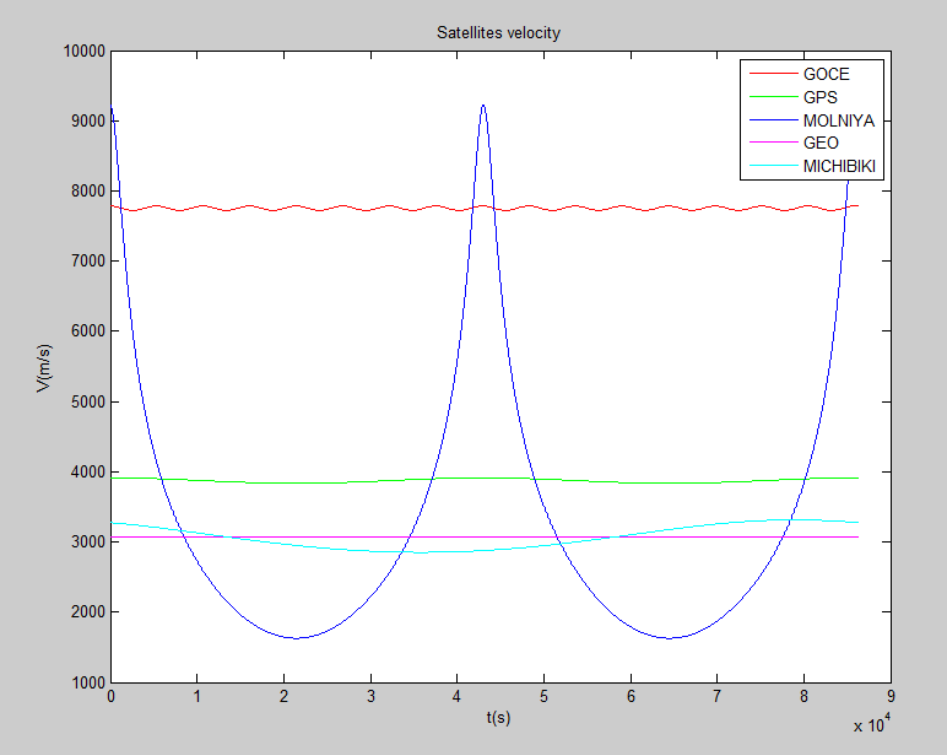


Picture 8: y-z plane

As we can see, for instance, Molniya has relatively big argument of perigee (270 degrees), so its perigee is close to the Earth. For this satellite Earth resides at one the focal points of its elliptical orbit, whereas other satellites have Earth in the center of their orbits.

GOCE satellite has the right ascension of 257.7 degrees, therefore its looks so swiveled with respect to the equatorial plane.

GEO has zero inclination, therefore it looks like a straight line on x-z and y-z 2D planes.



Picture 9: Satellites velocities

As we can see from picture 9, most of the satellites have small speed deviations. That can be easily explained – the Earth is the center of their orbits and distances to perigee and apogee are pretty much the same. Meanwhile, Molniya satellite has its apogee point far from the Earth and slows down on its way to the apogee, whereas at time t=0 (and at approx $4{,}5 \times 10^4$ sec) it resides right at the perigee point and therefore satellite's speed is maximum here.

Michibiki and GEO have relatively small velocities (compared to GOCE for example) since their orbit radiuses are the biggest.

GEO has constant speed because of its circular orbit and hence lacking of perigee and apogee.

Problem 6

Create a MATLAB-function cart2efix.m that transforms position and velocity in a space-fixed system into position and velocity in an Earth-fixed system.

To convert position and velocity to Earth-fixed system we need to rotate position and velocity vectors in space-fixed system about the z-axis at the angle that corresponds to the Greenwich Sidereal Time θ_0 at each epoch t.

Rotating rate: $\dot{\Omega}_E = \frac{2\pi}{86164s}$ (12)

$$\theta_0(t) = \dot{\Omega}_E t \quad (13)$$

Position vector in Earth-fixed system:

$$\mathbf{r}_{\text{Earth-fixed}}(t) = \mathbf{R}_3(\theta_0(t)) \cdot \mathbf{r}_{\text{space-fixed}}(t) \quad (14)$$

Where R_3 is the following matrix:

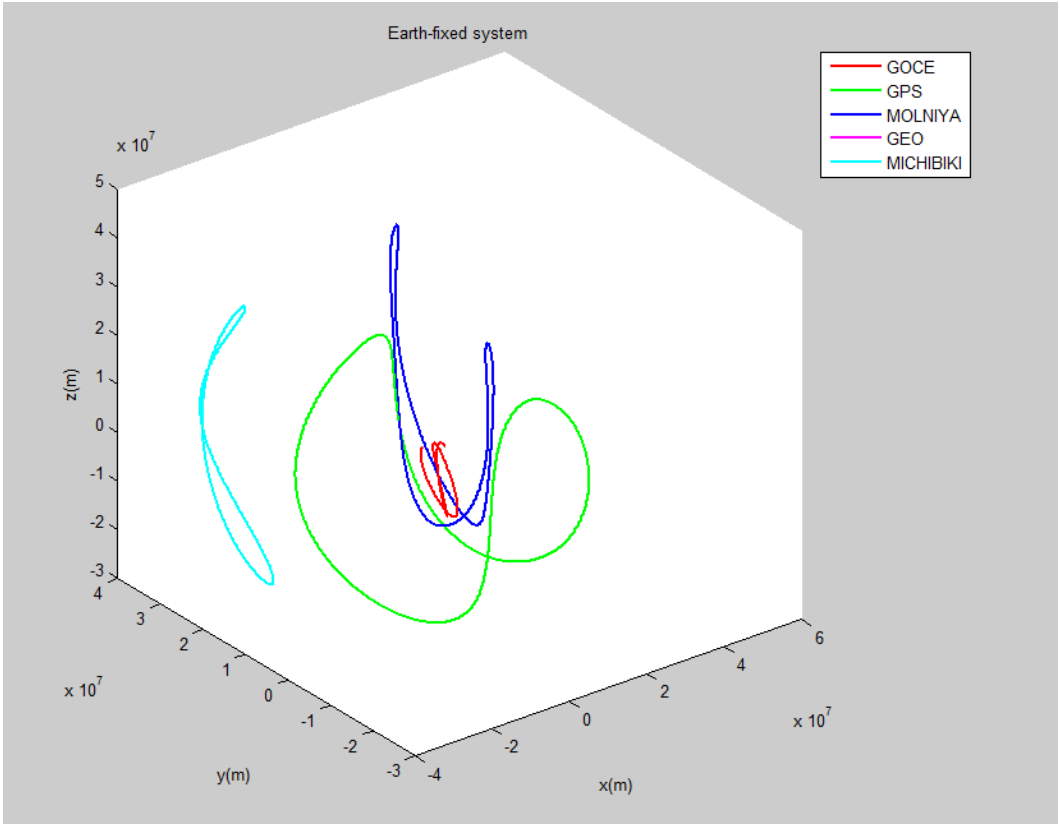
$$R_3(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Velocity vector is obtained the same way, but instead of multiplying rotation matrix by position vector in space-fixed system, we multiply it by velocity vector in space-fixed system.

Problem 7

Plot the trajectory of the satellites in 3D for the first two orbital revolutions.

Again we calculate periods of satellites T and calculate position and velocity vectors in Earth-fixed system from t=1 to t=2·T using cart2efix function.



Picture 10: Orbits in the Earth-fixed system

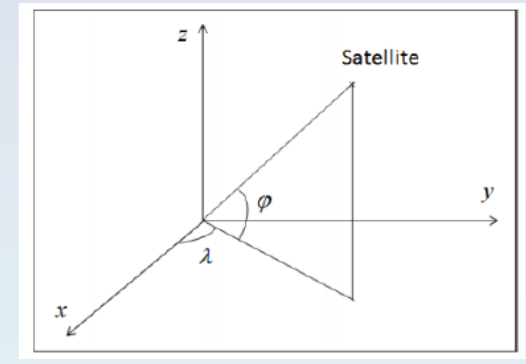
Problem 8

Calculate and draw the satellite ground-tracks on the Earth surface.

The transformed vector components in the Earth-fixed reference frame may be written in polar coordinates:

$$\mathbf{r}_{\text{Earth-fixed}} = \begin{pmatrix} x_1' \\ x_2' \\ x_3' \end{pmatrix} = \begin{pmatrix} r\cos\varphi\cos\lambda \\ r\cos\varphi\sin\lambda \\ r\sin\varphi \end{pmatrix} \qquad (15)$$

From this we obtain longitude φ and latitude λ



$$\tan\lambda = \frac{y_{Earth-fixed}}{x_{Earth-fixed}} \qquad \tan\psi = \frac{z_{Earth-fixed}}{\sqrt{x^2_{Earth-fixed} + y^2_{Earth-fixed}}} \qquad (16)\ (17)$$

For plotting we do the following:

```
x=eposition(1,:);
y=eposition(2,:);
z=eposition(3,:);
```

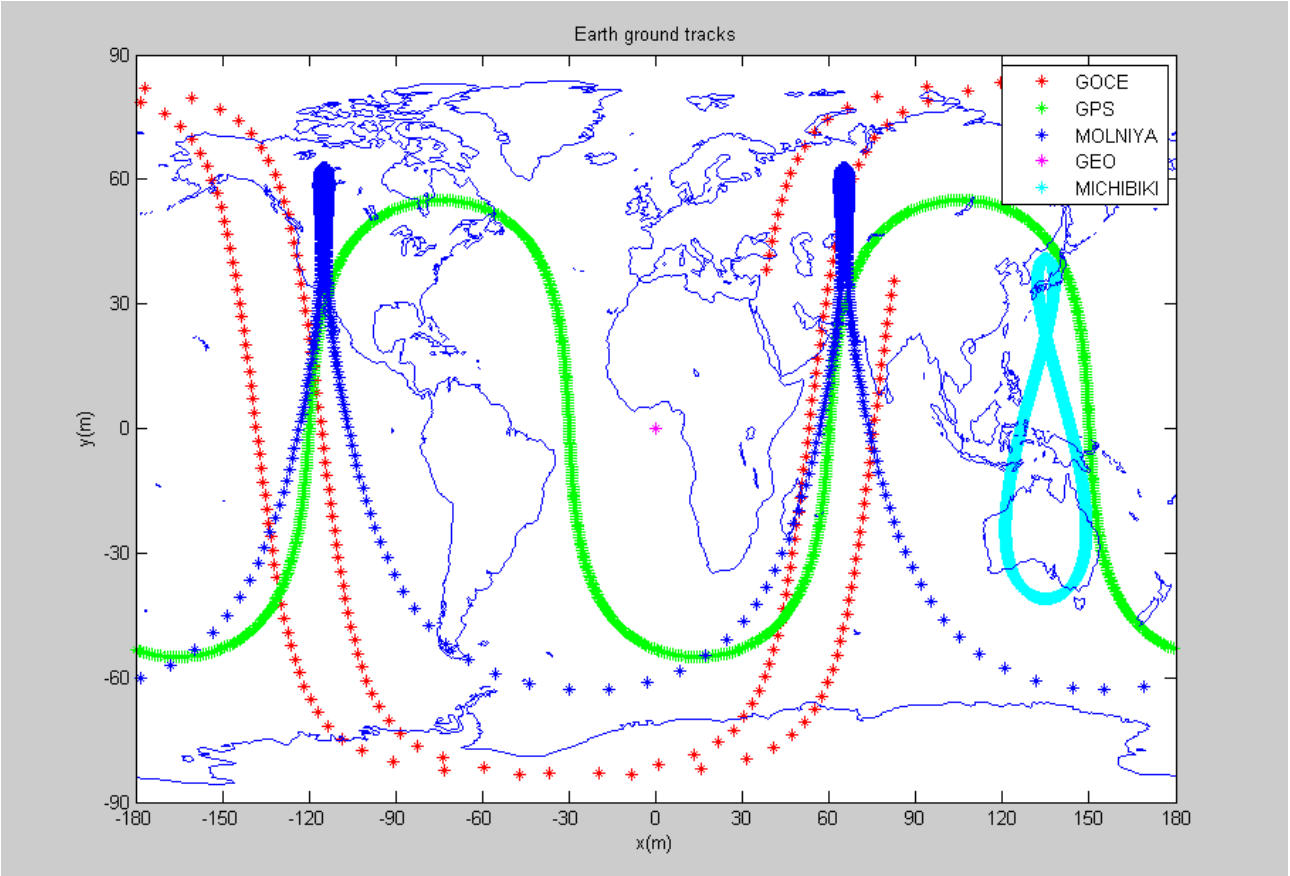
eposition – position in Earth-fixed system

Thus:

```
longitute=radtodeg(atan2(y,x));
latitute=radtodeg(atan(z./sqrt(x.*x+y.*y)));
```

We want to use given function Earth_coast to obtain the Earth outlines.

So, we get the following picture:



Picture 11: Earth ground tracks

Problem 9

Create a MATLAB-function efix2topo.m that transforms position and velocity in an Earthfixed system into position and velocity in a topocentric system centered at the station Wettzell which position vector in an Earth-fixed system is given by: rW = (4075.53022, 931.78130, 4801.61819)T km.

Observations from terrestrial observatories are connected to *a local horizontal coordinate system*. The location of the station is the topocenter.

The *system* is oriented such that:

- the third axis is parallel to the local plumb line, pointing to the zenith,
- the first axis is pointing northwards, in the ellipsoidal tangent plane,

- the second axis is pointing eastwards, in the ellipsoidal tangent plane.

We have to calculate the *longitude and latitude of Wettzell* station using equations 16 and 17 given position vector rW.

Than we want to calculate the Topocentric position vector **d** of the satellite by subtracting the position vector of Wettzell from the position vector of the satellite in Earth-fixed system: d = r – R (18)

Than vector **d** has to be transformed into the local horizontal system by applying a transformation

$$\mathbf{d}_{\text{hori}} = \mathbf{Q}_1 \, \mathbf{R}_2(90° - \varphi) \, \mathbf{R}_3(\lambda) \, \mathbf{d} \tag{19}$$

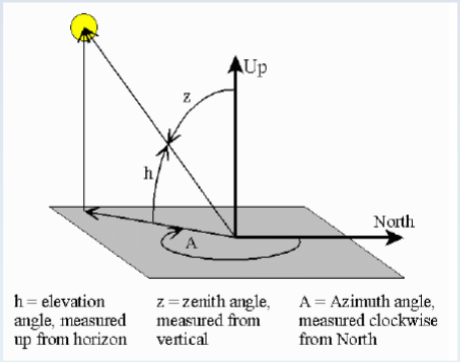
We need matrix Q_1 to transform the system from right-handed to left-handed.

$$\mathbf{Q}_1 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R_2(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}$$

So, we rotate the transformed Topocentric vector along the y-axis at a longitude angle and along the z-axis at a latitude angle. Thus obtaining x,y and z coordinates.

In Topocentric system we need azimuth and elevation angles to locate the object. We calculate them using the following equations:



$$\tan A = \frac{y}{x}$$

$$\cos z = \frac{z}{\sqrt{x^2 + y^2 + z^2}}$$

$$E = \frac{\pi}{2} - z$$

(20), (21)

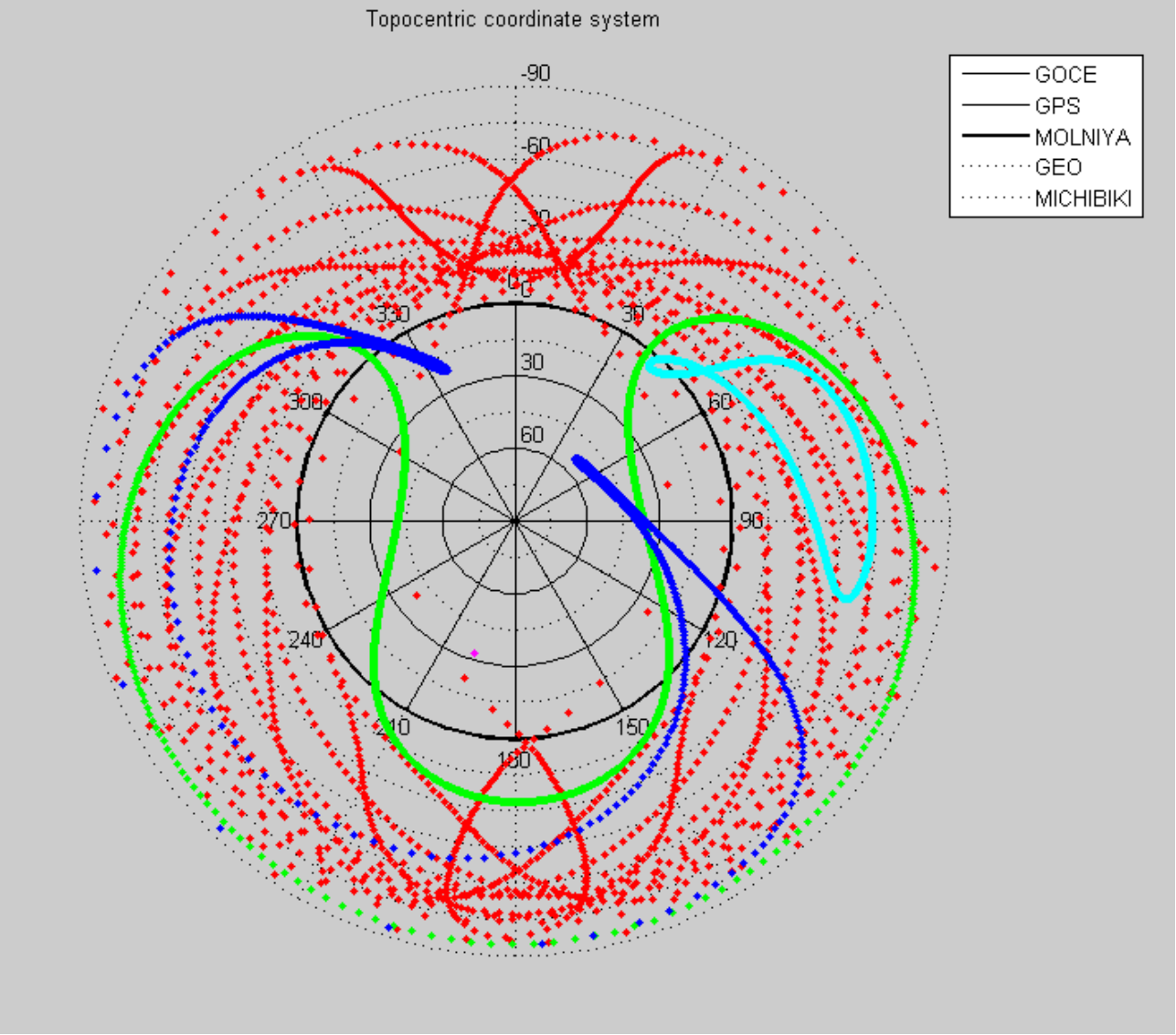
```
x=topoposition(1,:);   topoposition – position of the satellite in a Topocentric system
y=topoposition(2,:);
z=topoposition(3,:);

A=atan2(y,x);
if A(A<0)<0
   A=A+2*pi;
end
azimuth=A*180/pi;
Z=acosd(z./sqrt(x.^2+y.^2+z.^2));
elevation=90-Z;
```

Problem 10

Plot the trajectory of the satellites as observed by Wettzell using the MATLAB-function skyplot.m.

We use calculated azimuth and elevation as input parameters for skyplot function to plot the required trajectory in a Topocentric system.

The picture below shows the result:



Picture 12: Satellites trajectories in a Topocentric system

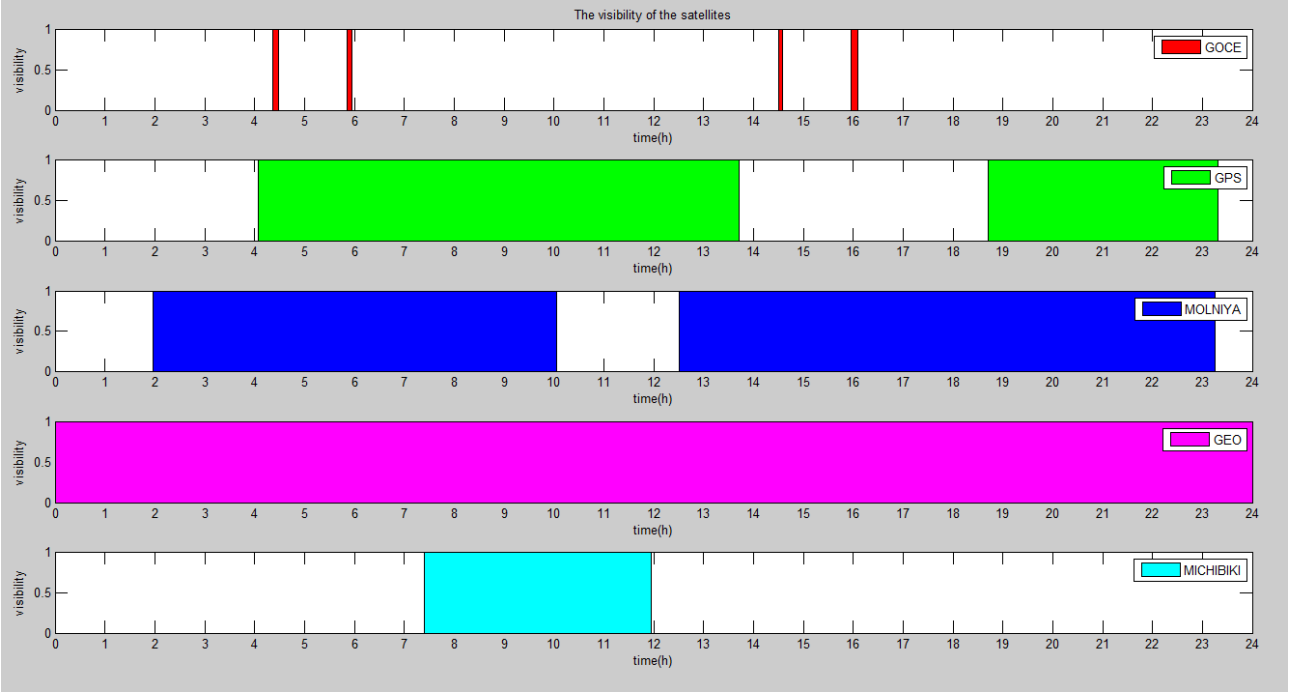
Problem 11

Calculate visibility (time intervals) for the satellites at the station Wettzell and visualize them graphically.

We cannot see the satellite at any given time from the ground station (except for geostationary satellites), therefore it is necessary to determine the visibility periods.

Previously, we have calculated the elevations of the satellites. If elevation is larger than zero then the satellite is visible in this particular moment of time.

Here are the results for given satellites:



Picture 13: Visibility of the satellites

Functions code:

kep2orb

```
function[r, v, M, E]=kep2orb(t,a,e,sat_num)
%% Original parameters
Re = 6371 * 10^3;
GM = 398.6005 * 10^12;
n = sqrt(GM/(a^3));        % mean motion = average angular velocity [rad/s]
T0 = 2*pi*(1/n);           % one revolution
t = 1:T0;                  % points for one revolution,
M = n*t;                   % mean anomaly

%% Michibiki has different mean anomaly:
if sat_num==5
 M=M+pi/6;
end
E(1,length(M))=0;  %Element E from 1st row and last column of M
```

```
for ii=1:length(M)
        E0=M(ii);
        E1=M(ii)+e*sin(E0);
        while abs(E1-E0)>10^(-6)  %%
          E0=E1;
          E1=M(ii)+e*sin(E0);
        end
        E(ii)=E1;
        end

%% Calculating radius and true anomaly
r = a*(1-e*cos(E));
v = 2*atan((sqrt((1+e)/(1-e)))*tan(E/2));
end
```

kep2cart

```
function [position,velocity,vel_magnitude] = kep2cart(t,per,asc,i,e,sat_num,a)
%% Parameters
GM = 398.6005 * 10^12;
%t=1:86400;
%% Polar coordinates function
[r,v] = kep2orb(t,a,e,sat_num)
%% Position Vectors
rbx=r.*cos(v);
rby=r.*sin(v);
rbz=r.*0;
rb=[rbx;rby;rbz];
%% Velocity Vectors
vb_x=sqrt(GM/(a*(1-e^2)))*(-sin(v));
vb_y=sqrt(GM/(a*(1-e^2)))*(e+cos(v));
vb_z=sqrt(GM/(a*(1-e^2)))*zeros(1,length(v));
vb=[vb_x;vb_y;vb_z];
%% Rotational Matrices
R3_per = [cos(-per) sin(-per) 0; -sin(-per) cos(-per) 0; 0 0 1 ];
R1_i = [ 1 0 0; 0 cos(-i) sin(-i); 0 -sin(-i) cos(-i)];
R3_asc = [ cos(-asc) sin(-asc) 0; -sin(-asc) cos(-asc) 0; 0 0 1];
%% Position, Velocity, Velocity Magnitude
position = (R3_asc)*(R1_i)*(R3_per)*rb;
velocity = (R3_asc)*(R1_i)*(R3_per)*vb;
vel_magnitude =
sqrt(velocity(1,:).*velocity(1,:)+velocity(2,:).*velocity(2,:)+velocity(3,:).*velocity(3,:));
end
```

cart2exfix

```
function [eposition, evelocity] = cart2exfix(position, velocity, T)
rotating_rate = (2*pi)/86164;
for k = 1:length(T)
```

```
    theta = (rotating_rate)*k;
    R3_theta = [cos(theta) sin(theta) 0; -sin(theta) cos(theta) 0; 0 0 1]
    eposition(:,k)=R3_theta*position(:,k);
    evelocity(:,k)=R3_theta*velocity(:,k);
end
end
```

exfix2topo

```
function [topoposition topovelocity] = exfix2topo(eposition,evelocity)
% Left hand - Right handed
Q1=[-1 0 0;0 1 0;0 0 1];
% Wettzell coordinates
wettzell=[4075.53022*10^3 931.78130*10^3 4801.61819*10^3]';

n=length(eposition(1,:));
latitude_topo=atan2(wettzell(2),wettzell(1));
longitude_topo=atan(wettzell(3)/sqrt(wettzell(2)^2+wettzell(1)^2));
R2 = [cos((pi/2)-longitude_topo) 0 -sin((pi/2)-longitude_topo); 0 1 0; sin((pi/2)-
longitude_topo) 0 cos((pi/2)-longitude_topo)];
R3 = [cos(latitude_topo) sin(latitude_topo) 0; -sin(latitude_topo) cos(latitude_topo) 0; 0 0
1];
R=Q1*R2*R3;
topoposition=R*(eposition-wettzell*ones(1,n));
topovelocity=R*(evelocity-wettzell*ones(1,n));

end
```