Technische Universität München
Institut for Astronomical and Physical Geodesy

Orbit Mechanics Tutorial

Exercise 1: Keplerian Orbits in Space-Fixed, Earth-fixed and Topocentric Systems

Tutor: Mario Fuentes Reyes mario.fuentes@tum.de

Directed by: Prof. Urs Hugentobler urs.hugentobler@bv.tum.de

3 tutorials in the semester:

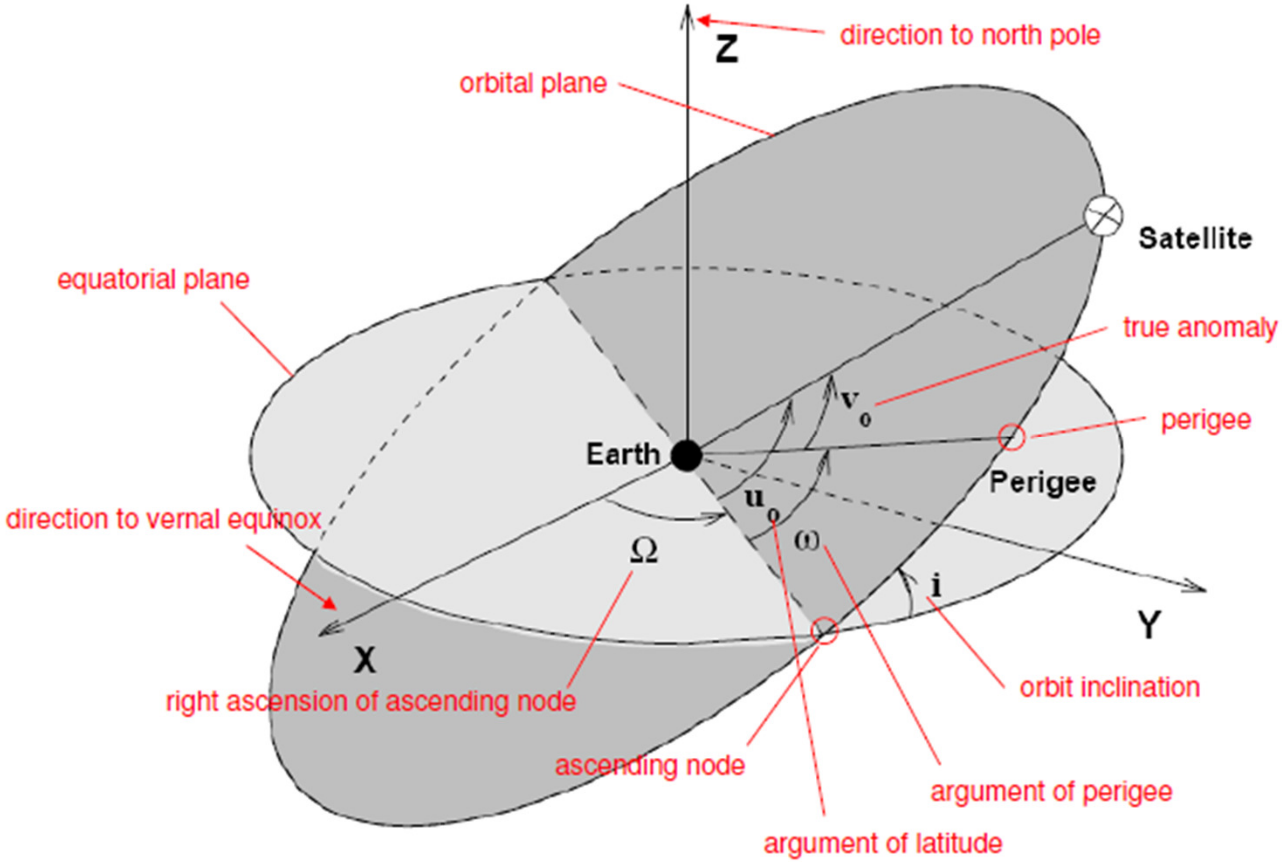
1. Keplerian Orbits in Space-Fixed, Earth-fixed and Topocentric Systems
2. Numerical Integration of Satellite Orbits
3. Integration of Satellite Orbits with Different Force Models

Wednesdays in room 2601 from 13:15 to 14:45.
Doubts and code issues.

Given satellites:

Satellite	a[km]	e	i[deg]	Ω [deg]	Ω [deg]	T_0 [h]
GOCE	6629	0.004	96.6	210	144.2	02:00
GPS	26560	0.01	55	30	30	11:00
MOLNIYA	26554	0.7	63	200	270	07:30
GEO	Geostationary	0	0	0	50	00:00
MICHIBIKI	Geosynchronous	0.075	41	200	270	04:10

Keplerian Elements:



Task 1: Orbits in 2D plane

Create kep2orb function. Polar coordinates.

- Inputs: a, e, t, T_0
- Outputs: r (position), v (velocity), M, E

Useful formulae

To compute mean anomaly: $n = \sqrt{\frac{GM}{a^3}}$ $\quad M(t) = n \cdot (t - T_0)$

Kepler's Equation: $M = E - e \sin E \quad \Delta E_i < 10^{-6}$

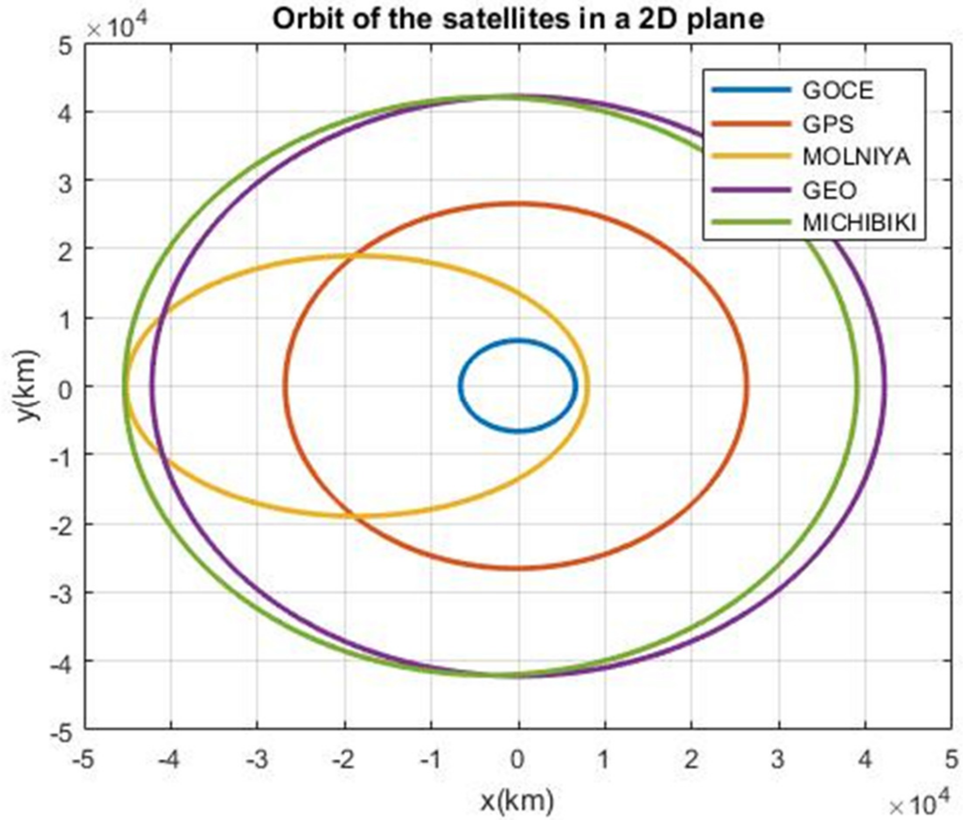
Radius: $r = a(1 - e \cos E)$

True anomaly: $\tan\frac{v}{2} = \sqrt{\frac{1+e}{1-e}} \tan\frac{E}{2}$

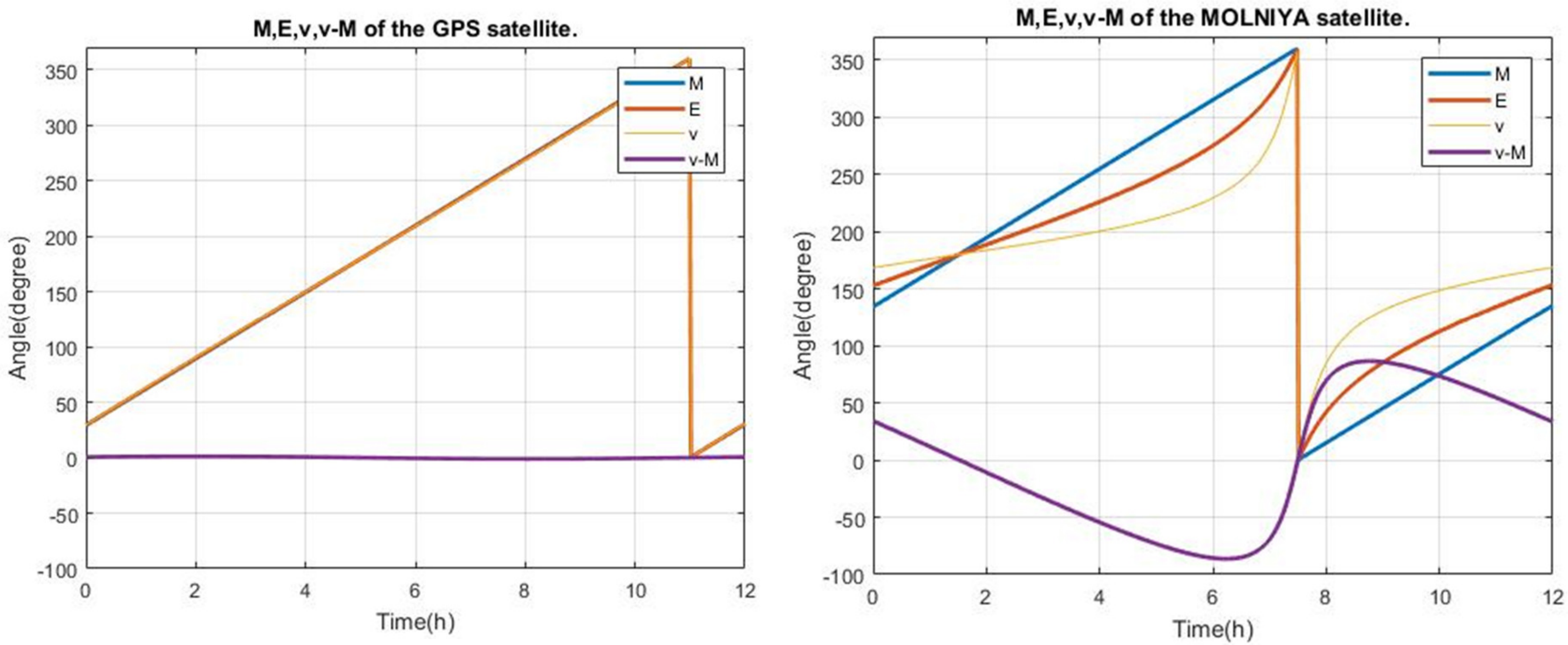
Hint:
$v = 2\,\mathrm{atan2}\left(\sqrt{1+e}\sin\frac{E}{2}, \sqrt{1-e}\cos\frac{E}{2}\right)$

2D coordinates: $x = r\cos v \quad y = r \sin v$

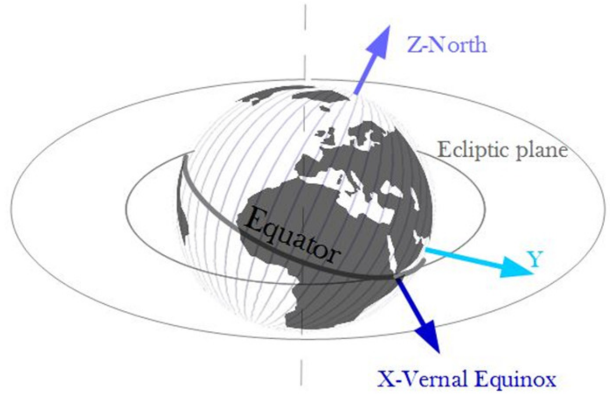
Plot in 2D. Expected result:



Mean, Eccentric and True Anomaly.
GPS and MOLNIYA Satellite. Here for 12 hours.



Task 2: Space-fixed system



Create kep2cart function. Space-fixed system.

- Inputs: a, e, i, Ω, ω, t, T_0
- Outputs: r_2 (position), v_2 (velocity)

Useful formulae

r, v already known

$$\overrightarrow{r_{2'}} = r\begin{bmatrix}\cos v\\ \sin v\\ 0\end{bmatrix} \quad \dot{\overrightarrow{r_{2'}}} = r\begin{bmatrix}\cos v\\ \sin v\\ 0\end{bmatrix}$$

Position:

$$\overrightarrow{r_2} = R_3(-\Omega)R_1(-i)R_3(-\omega)\overrightarrow{r_{2'}}$$

Velocity:

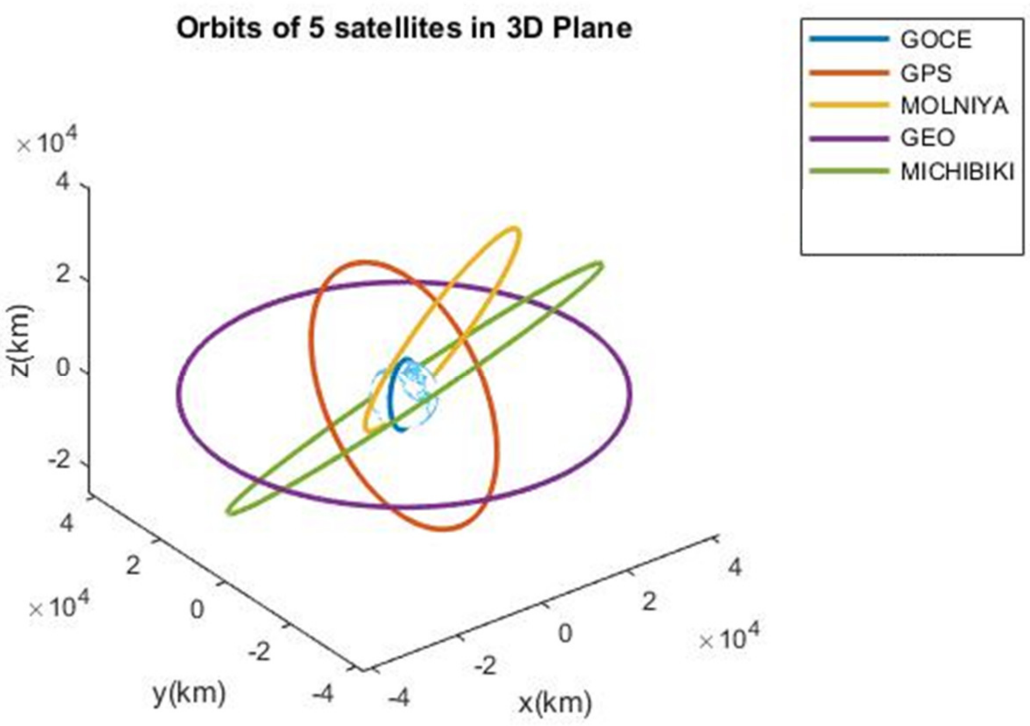
$$\dot{\overrightarrow{r_2}} = R_3(-\Omega)R_1(-i)R_3(-\omega)\dot{\overrightarrow{r_{2'}}}$$

Where:

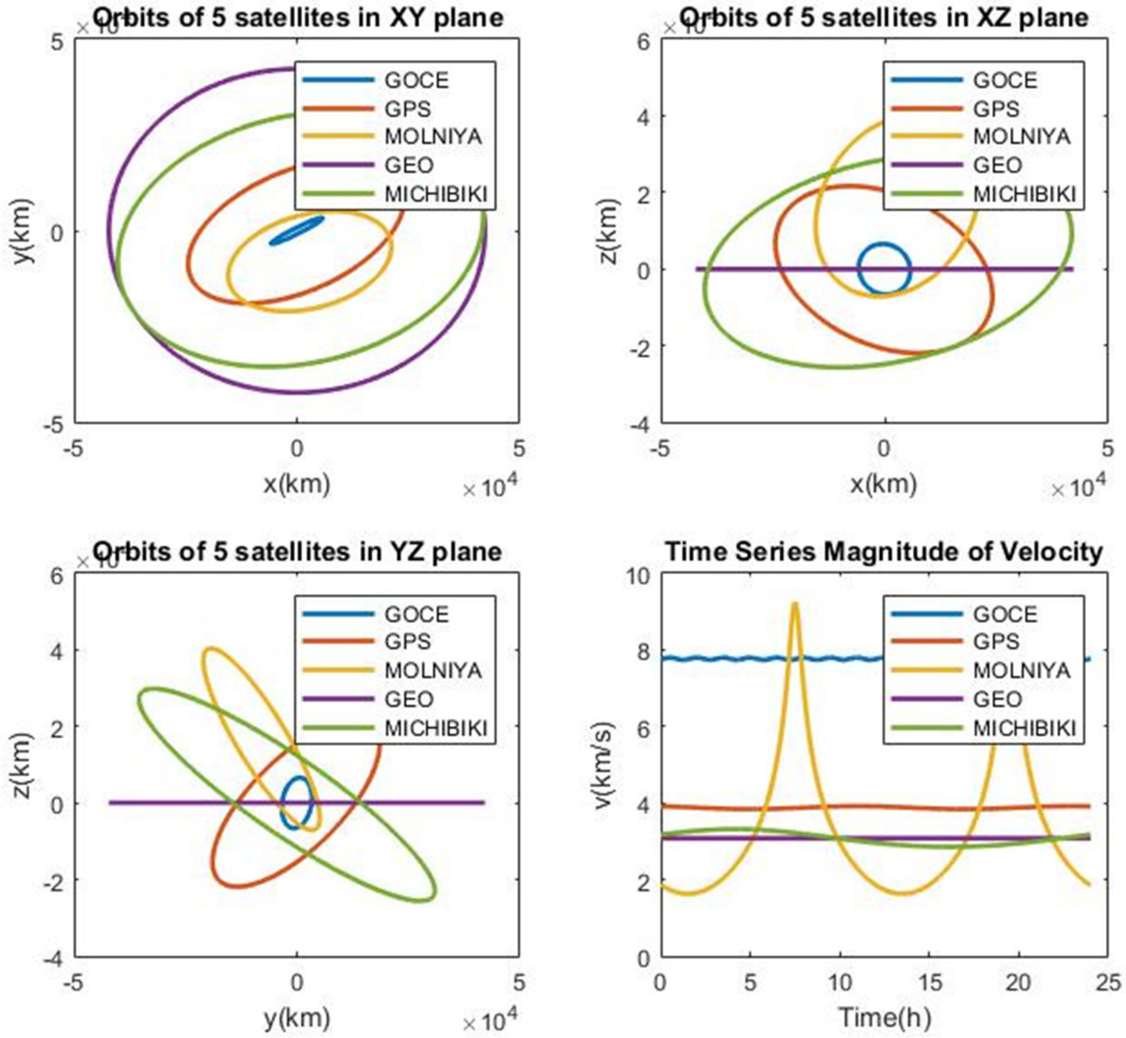
$$R_1(\theta) = \begin{bmatrix}1 & 0 & 0\\ 0 & \cos\theta & \sin\theta\\ 0 & -\sin\theta & \cos\theta\end{bmatrix} \quad R_3(\theta) = \begin{bmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}$$

Plot in 3D.

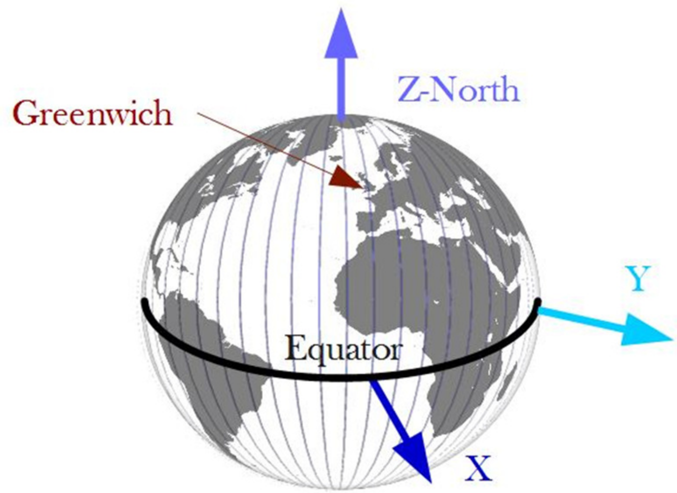
Use Earth_coast(3) from Moodle. Expected result:



Plot in 2D.
Projections.
Expected result:



Task 3: Earth-fixed system



Create cart2efix function. Space-fixed system.

- Inputs: r_2, v_2, t
- Outputs: r_3 (position), v_3 (velocity)

Useful formulae

Earth rotation rate

$$\dot{\Omega}_E = \frac{2\pi}{86164}$$

Rotation angle:

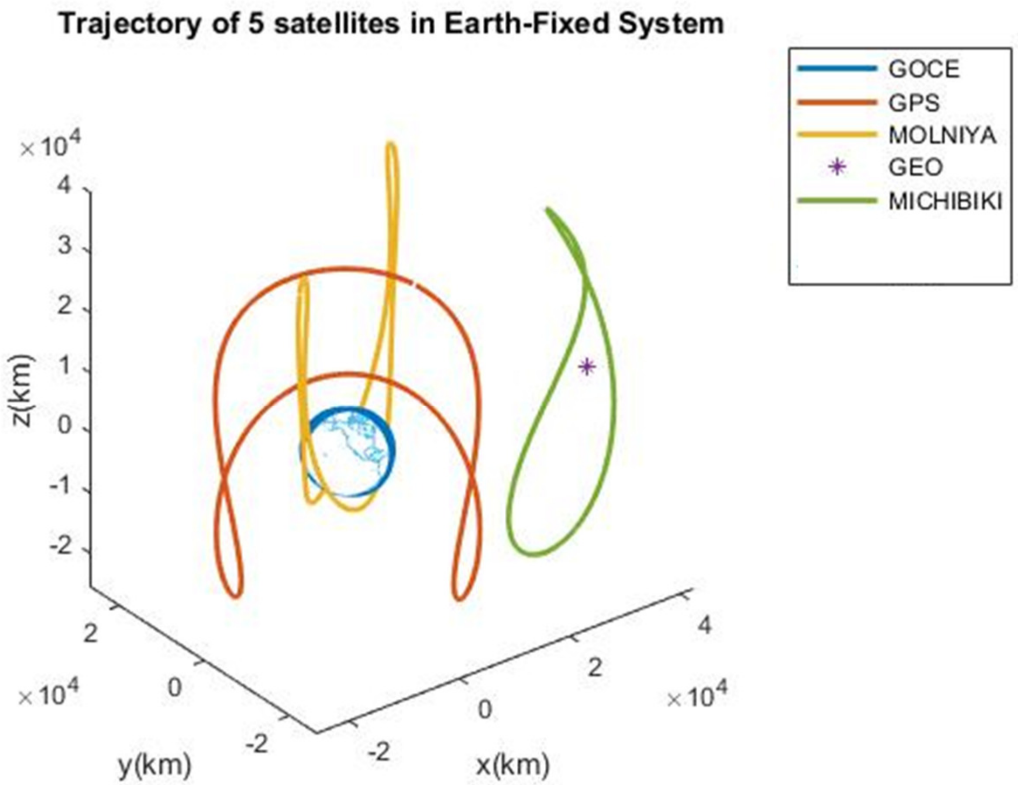
$$\theta_0(t) = \dot{\Omega}_E t + \text{sidereal angle } (03{:}29 \text{ in deg})$$

Position:

$$\vec{r_3}(t) = R_3(\theta_0(t))\vec{r_2}(t)$$

Plot in 3D.

Expected result:

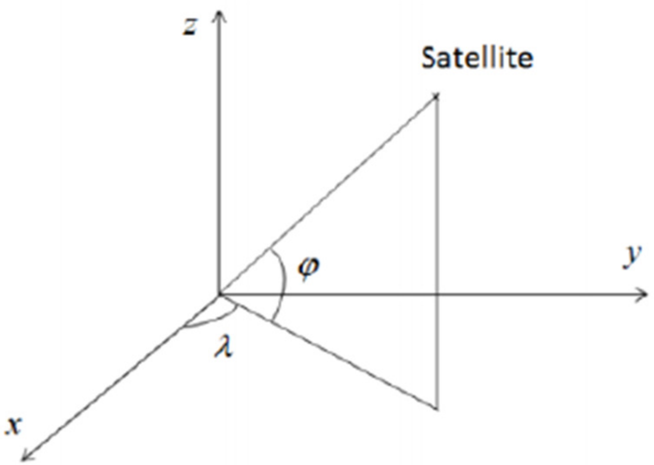


Ground-tracks on Earth-surface

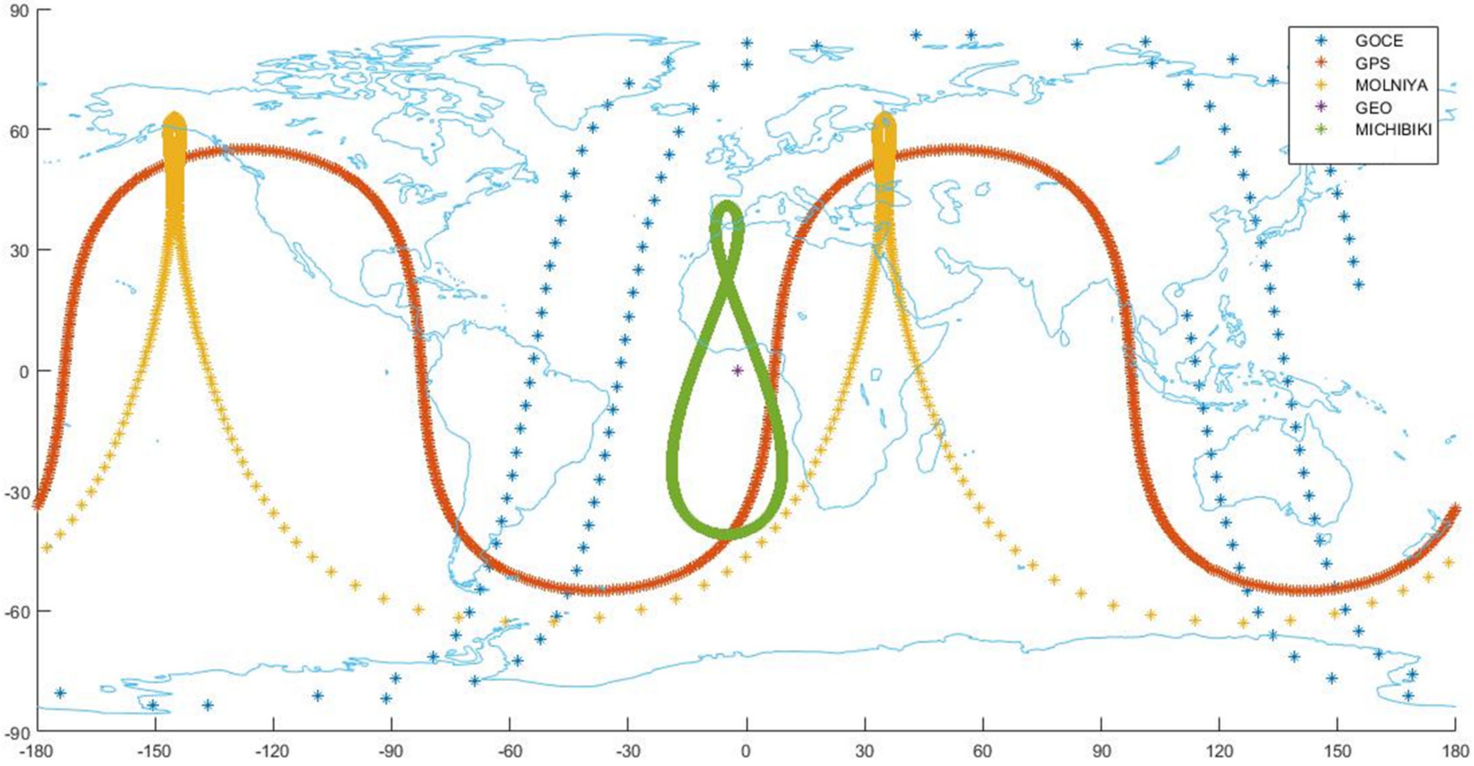
Useful formulae

Latitude $\lambda \in [-180°, 180°]$: $\tan\lambda = \frac{y_3}{x_3}$

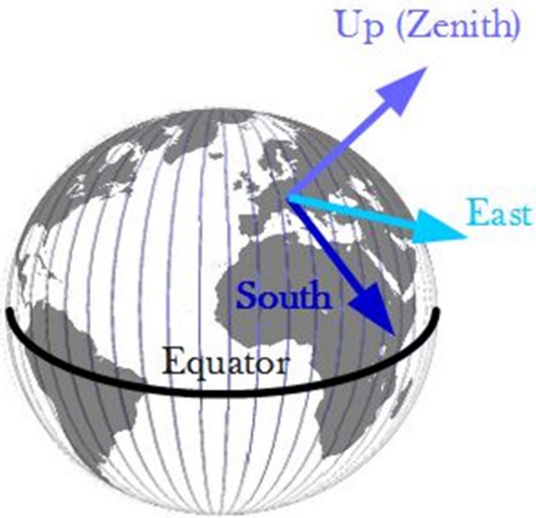
Longitude $\psi \in [-90°, 90°]$: $\tan\psi = \frac{z_3}{\sqrt{x_3^2 + y_3^2}}$



Plot in 2D. Use Earth_coast(2).
Expected result:



Task 4: Topo-centric system



Create efix2topo function. Space-fixed system.

- Inputs: r_3, v_3
- Outputs: r_4 (position), v_4 (velocity), azimuth, elevation

Useful formulae

Translated vector $r_{trans} = r_3 - r_{\text{Wettzell}}$ (check Exercise)

Topocentric vector $r_4 = Q_1 R_2(90 - \text{latitude}_{\text{Wettzell}}) R_3(\text{longitude}_{\text{Wettzell}}) r_{trans}$

Where:

$$Q_1 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

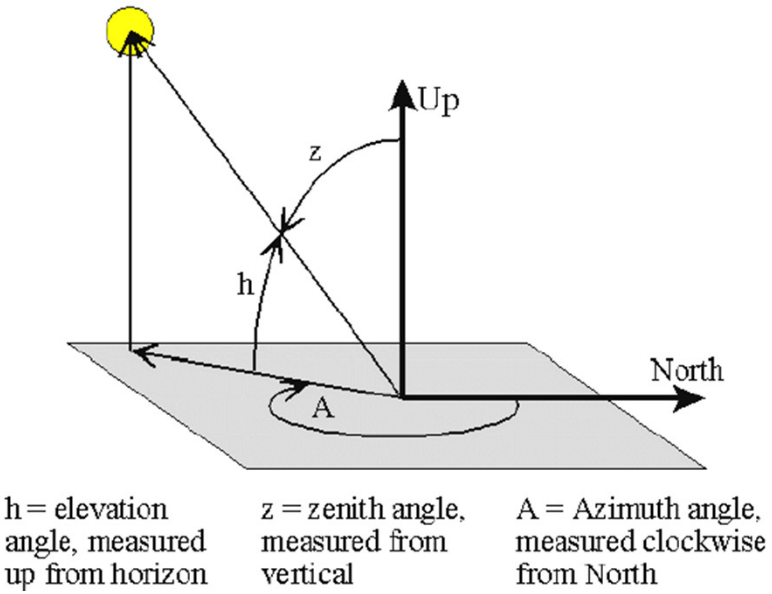
From right to left-handed system

Trajectory observed from Wettzell

Useful formulae

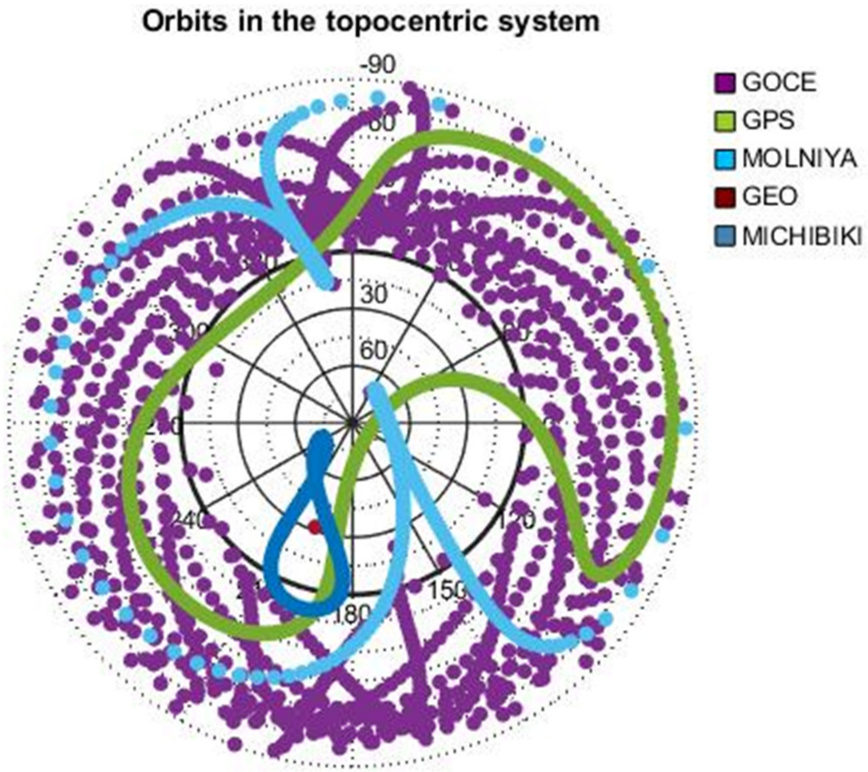
Azimuth
$$\tan A = \frac{y_4}{x_4}$$

Elevation
$$\tan h = \frac{z_4}{\sqrt{x_4^2 + y_4^2}}$$

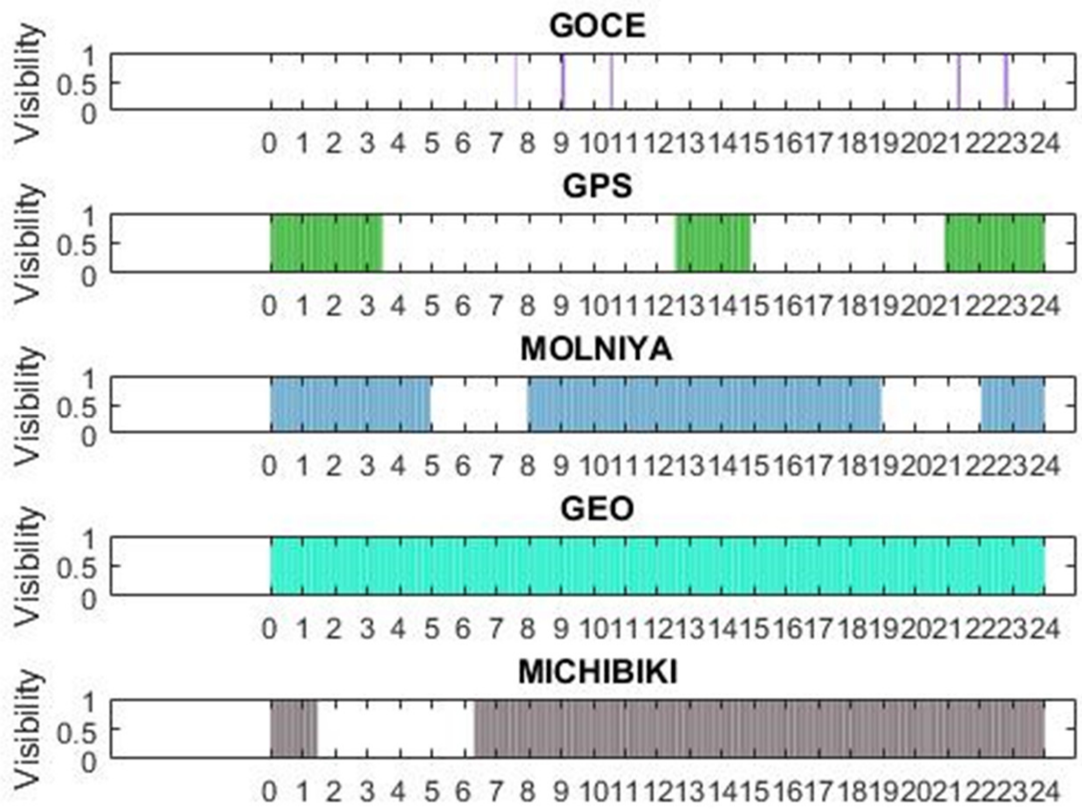


Plot using skyplot.

Syntax: skyplot(azimuth, elevation, Marker_shape(like ‘+b’))



Visualization over Wettzell (consider elevation angle >0)



Suggestions:

- Comment your code.
- Use simple and readable names for variables.
- Avoid nested loops.
- Check angles to be between 0 and 360 or 0 and 2π.
- Separate the code into sections to run just specific parts if required.
- Use atan2 instead of atan.
- Initialize matrices (with zeros())
- Attend on Wednesdays or send an email in case you have doubts.