

ENAE 404 - 0101
Homework 03: Ground Tracks

Due on March 13, 2025 at 09:30 AM

Dr. Barbee, 09:30

Vai Srivastava

March 11, 2025

Problem 1:

Conceptual questions:

1. Give the semi-major axis, eccentricity and inclination of an orbit whose ground track is a point.
2. Explain why argument of periapsis equal to 0° or 180° produces equatorial symmetry.
3. Consider the Molniya orbit. Which one orbital element would you change so that the spacecraft would spend a long time viewing the Southern hemisphere (rather than the Northern hemisphere)? Identify the orbital element and the value of this orbital element that would preserve the same structure of the ground track (just flipped to observe the Southern hemisphere).

Solution

Part A

For an orbit whose ground track is a fixed point, the spacecraft must be in a geostationary orbit. This requires:

$$\begin{aligned}a &\approx 42\,164 \text{ km} \quad (\text{from Earth's center}), \\e &= 0 \quad (\text{circular orbit}), \\i &= 0^\circ \quad (\text{orbit in the equatorial plane}).\end{aligned}$$

Part B

The argument of periapsis, ω , defines the orientation of the ellipse within the orbital plane relative to the line of nodes. When ω is 0° or 180° , the line of apsides is aligned with the line of nodes. This alignment causes the orbit to be symmetric about the equatorial plane.

Part C

A standard Molniya orbit is designed with an argument of periapsis $\omega \approx 270^\circ$ so that the apogee is in the Northern Hemisphere. To flip the ground track so that the apogee is over the Southern Hemisphere, one would change the argument of periapsis to $\omega = 90^\circ$.

Problem 2:

Plot one day ground tracks for the following orbits:

Spacecraft ID	a(km)	e	i(°)	Ω (°)	ω (°)
A	42164	0.3	40	0	0
B	26562	0.3	40	0	0
C	42164	0.3	60	0	70
D	42164	0.3	30	0	70
E	42164	0.3	120	0	0

Compare and contrast the ground track for Orbit E to that of Orbit A.

Solution

```

1 from astropy import units as u
2 from poliastro.util import time_range
3 from poliastro.bodies import Earth
4 from poliastro.twobody import Orbit
5 from poliastro.earth import EarthSatellite
6 from poliastro.earth.plotting import GroundtrackPlotter

```

Listing 1: P02 Python Code

Part A

Spacecraft A groundtrack

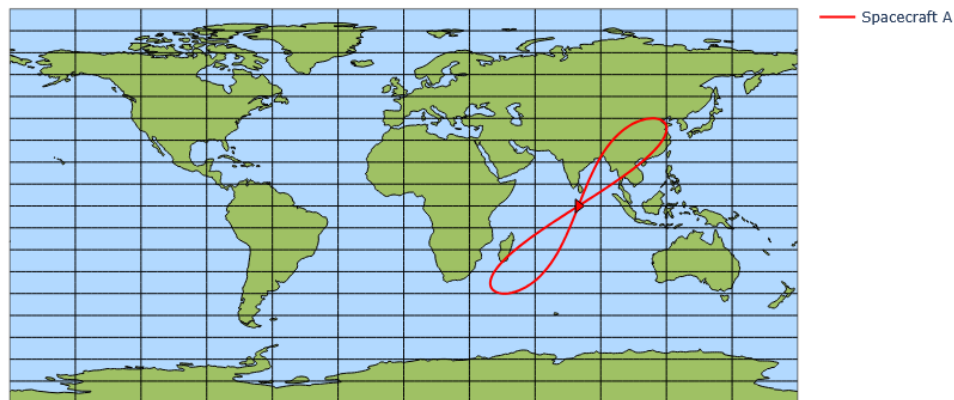


Figure 1: Spacecraft A groundtrack

```

1 a = 42164 * u.km          # semi-major axis
2 ecc = 0.3 * u.one         # eccentricity
3 inc = 40 * u.deg          # inclination
4 raan = 0 * u.deg          # right ascension of the ascending node
5 argp = 0 * u.deg          # argument of perigee
6 nu = 0 * u.deg            # true anomaly (set to 0 for the initial state)

```

```

7
8 Orbit_A = Orbit.from_classical(Earth, a, ecc, inc, raan, argp, nu)
9 Spacecraft_A = EarthSatellite(Orbit_A, None)
10 t_span = time_range(start=Orbit_A.epoch, end=Orbit_A.epoch + 24.0 * u.h, num_values=150)
11
12 gp = GroundtrackPlotter()
13 gp.update_layout(title="Spacecraft A groundtrack")
14
15 gp.plot(
16     Spacecraft_A,
17     t_span,
18     label="Spacecraft A",
19     color="red",
20     marker={
21         "size": 10,
22         "symbol": "triangle-right",
23         "line": {"width": 1, "color": "black"},
24     },
25 )
26
27 # gp.update_geos(projection_type="orthographic")
28 gp.fig.show()

```

Listing 2: P02a Python Code

Part B

Spacecraft B groundtrack

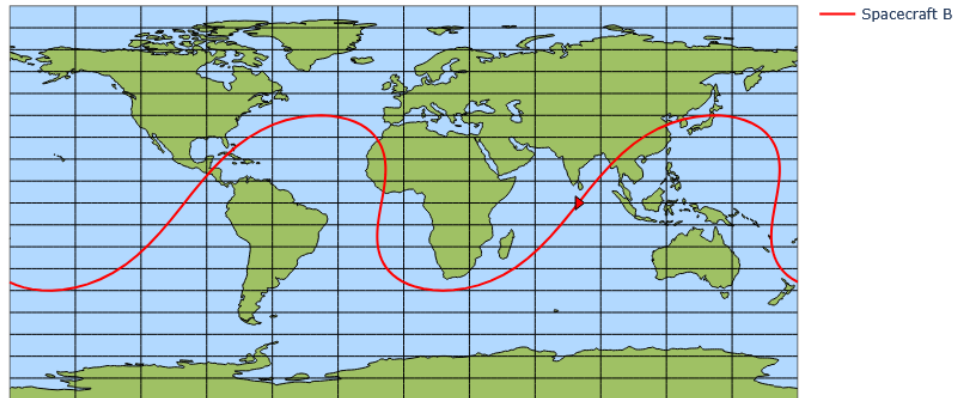


Figure 2: Spacecraft B groundtrack

```

1 a = 26562 * u.km          # semi-major axis
2 ecc = 0.3 * u.one         # eccentricity
3 inc = 40 * u.deg          # inclination
4 raan = 0 * u.deg          # right ascension of the ascending node
5 argp = 0 * u.deg          # argument of perigee
6 nu = 0 * u.deg            # true anomaly (set to 0 for the initial state)
7

```

```

8 Orbit_B = Orbit.from_classical(Earth, a, ecc, inc, raan, argp, nu)
9 Spacecraft_B = EarthSatellite(Orbit_B, None)
10 t_span = time_range(start=Orbit_B.epoch, end=Orbit_B.epoch + 24.0 * u.h, num_values=150)
11
12 gp = GroundtrackPlotter()
13 gp.update_layout(title="Spacecraft B groundtrack")
14
15 gp.plot(
16     Spacecraft_B,
17     t_span,
18     label="Spacecraft B",
19     color="red",
20     marker={
21         "size": 10,
22         "symbol": "triangle-right",
23         "line": {"width": 1, "color": "black"},
24     },
25 )
26
27 # gp.update_geos(projection_type="orthographic")
28 gp.fig.show()

```

Listing 3: P02b Python Code

Part C

Spacecraft C groundtrack

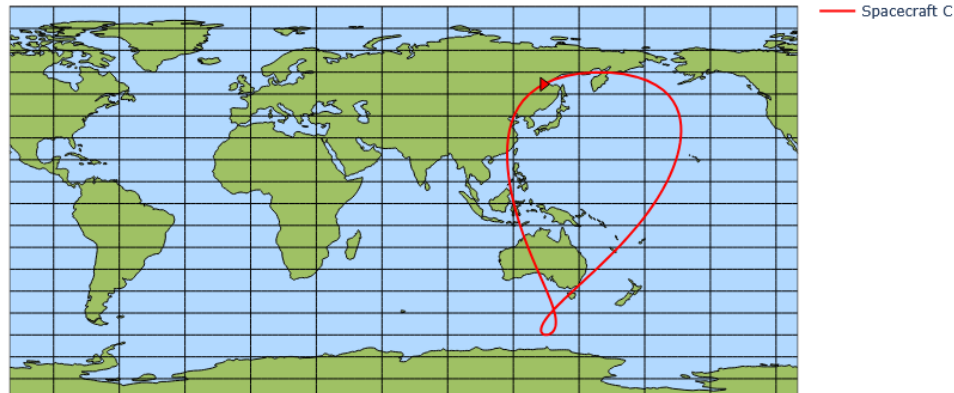


Figure 3: Spacecraft C groundtrack

```

1 a = 42164 * u.km           # semi-major axis
2 ecc = 0.3 * u.one          # eccentricity
3 inc = 60 * u.deg           # inclination
4 raan = 0 * u.deg           # right ascension of the ascending node
5 argp = 70 * u.deg          # argument of perigee
6 nu = 0 * u.deg             # true anomaly (set to 0 for the initial state)
7
8 Orbit_C = Orbit.from_classical(Earth, a, ecc, inc, raan, argp, nu)

```

```

9 Spacecraft_C = EarthSatellite(Orbit_C, None)
10 t_span = time_range(start=Orbit_C.epoch, end=Orbit_C.epoch + 24.0 * u.h, num_values=150)
11
12 gp = GroundtrackPlotter()
13 gp.update_layout(title="Spacecraft C groundtrack")
14
15 gp.plot(
16     Spacecraft_C,
17     t_span,
18     label="Spacecraft C",
19     color="red",
20     marker={
21         "size": 10,
22         "symbol": "triangle-right",
23         "line": {"width": 1, "color": "black"},
24     },
25 )
26
27 # gp.update_geos(projection_type="orthographic")
28 gp.fig.show()

```

Listing 4: P02c Python Code

Part D

Spacecraft D groundtrack

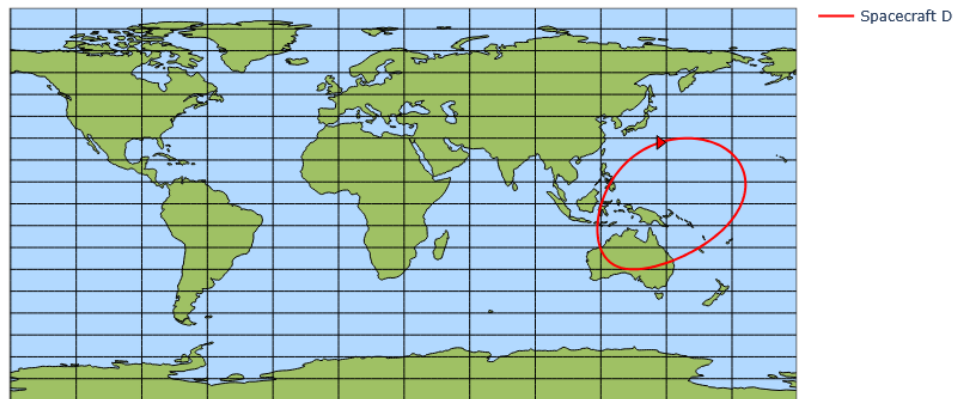


Figure 4: Spacecraft D groundtrack

```

1 a = 42164 * u.km          # semi-major axis
2 ecc = 0.3 * u.one         # eccentricity
3 inc = 30 * u.deg          # inclination
4 raan = 0 * u.deg          # right ascension of the ascending node
5 argp = 70 * u.deg         # argument of perigee
6 nu = 0 * u.deg            # true anomaly (set to 0 for the initial state)
7
8 Orbit_D = Orbit.from_classical(Earth, a, ecc, inc, raan, argp, nu)
9 Spacecraft_D = EarthSatellite(Orbit_D, None)

```

```

10 t_span = time_range(start=Orbit_D.epoch, end=Orbit_D.epoch + 24.0 * u.h, num_values=150)
11
12 gp = GroundtrackPlotter()
13 gp.update_layout(title="Spacecraft D groundtrack")
14
15 gp.plot(
16     Spacecraft_D,
17     t_span,
18     label="Spacecraft D",
19     color="red",
20     marker={
21         "size": 10,
22         "symbol": "triangle-right",
23         "line": {"width": 1, "color": "black"},
24     },
25 )
26
27 # gp.update_geos(projection_type="orthographic")
28 gp.fig.show()

```

Listing 5: P02d Python Code

Part E

Spacecraft E groundtrack

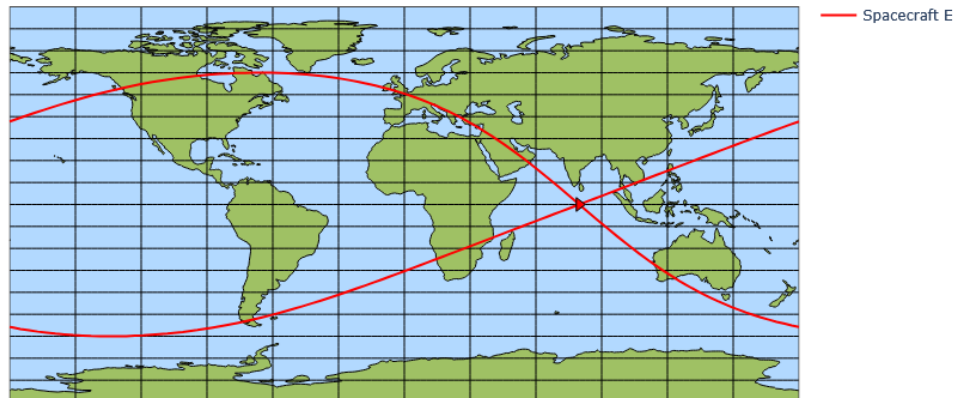


Figure 5: Spacecraft E groundtrack

```

1 a = 42164 * u.km          # semi-major axis
2 ecc = 0.3 * u.one          # eccentricity
3 inc = 120 * u.deg          # inclination
4 raan = 0 * u.deg           # right ascension of the ascending node
5 argp = 0 * u.deg           # argument of perigee
6 nu = 0 * u.deg             # true anomaly (set to 0 for the initial state)
7
8 Orbit_E = Orbit.from_classical(Earth, a, ecc, inc, raan, argp, nu)
9 Spacecraft_E = EarthSatellite(Orbit_E, None)
10 t_span = time_range(start=Orbit_E.epoch, end=Orbit_E.epoch + 24.0 * u.h, num_values=150)

```

```
11
12 gp = GroundtrackPlotter()
13 gp.update_layout(title="Spacecraft E groundtrack")
14
15 gp.plot(
16     Spacecraft_E,
17     t_span,
18     label="Spacecraft E",
19     color="red",
20     marker={
21         "size": 10,
22         "symbol": "triangle-right",
23         "line": {"width": 1, "color": "black"},
24     },
25 )
26
27 # gp.update_geos(projection_type="orthographic")
28 gp.fig.show()
```

Listing 6: P02e Python Code

Part F

The ground tracks for orbits A and E have the same lemniscate shape, though their rotation and scale are vastly different. The similarity is due to A and E sharing many orbital elements, though the differences are due to the varied inclination angle between the two orbits.

Problem 3:

Calculate the ΔV required to execute a Hohmann transfer from a circular orbit with radius 14×10^3 km to a circular orbit with radius 8×10^3 km. Assume the central body is the Earth. State whether the maneuvers increase or decrease the spacecraft's velocity.

Solution

$$a_t = \frac{r_1 + r_2}{2} = \frac{14\,000 \text{ km} + 8\,000 \text{ km}}{2}$$

$$= 11\,000 \text{ km}$$

$$v_{c1} = \sqrt{\frac{\mu}{r_1}} = \sqrt{\frac{398600}{14000}}$$

$$\approx 5.34 \frac{\text{km}}{\text{s}}$$

$$v_{t1} = \sqrt{\mu \left(\frac{2}{r_1} - \frac{1}{a_t} \right)} = \sqrt{398600 \left(\frac{2}{14000} - \frac{1}{11000} \right)}$$

$$\approx 4.55 \text{ km/s}$$

$$\Delta v_1 = v_{c1} - v_{t1} \approx 5.34 \frac{\text{km}}{\text{s}} - 4.55 \frac{\text{km}}{\text{s}}$$

$$= 0.79 \frac{\text{km}}{\text{s}}$$

$$v_{c2} = \sqrt{\frac{\mu}{r_2}} = \sqrt{\frac{398600}{8000}}$$

$$\approx 7.06 \frac{\text{km}}{\text{s}}$$

$$v_{t2} = \sqrt{\mu \left(\frac{2}{r_2} - \frac{1}{a_t} \right)} = \sqrt{398600 \left(\frac{2}{8000} - \frac{1}{11000} \right)}$$

$$\approx 7.97 \frac{\text{km}}{\text{s}}$$

$$\Delta v_2 = v_{t2} - v_{c2} \approx 7.97 \frac{\text{km}}{\text{s}} - 7.06 \frac{\text{km}}{\text{s}}$$

$$= 0.91 \frac{\text{km}}{\text{s}}$$

$$\Delta v_{\text{total}} = \Delta v_1 + \Delta v_2 \approx 0.79 \frac{\text{km}}{\text{s}} + 0.91 \frac{\text{km}}{\text{s}}$$

$$\Delta v_{\text{total}} = 1.70 \frac{\text{km}}{\text{s}} \quad \square$$

The Hohmann transfer requires a total $\Delta V \approx 1.70 \frac{\text{km}}{\text{s}}$. Both maneuvers decrease the spacecraft's velocity.