Mini-course:

Optimal Control of Space Trajectories using GEKKO

Lecture 3: Reentry gliding

Jhonathan Murcia-Piñeros

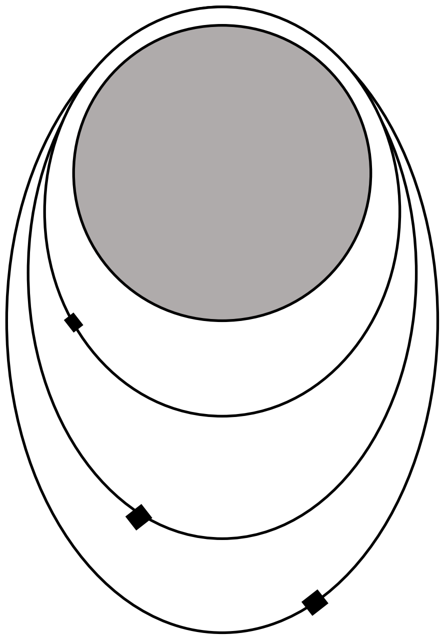
jhonathan.pineros@unifesp.br

CBDO - 2024

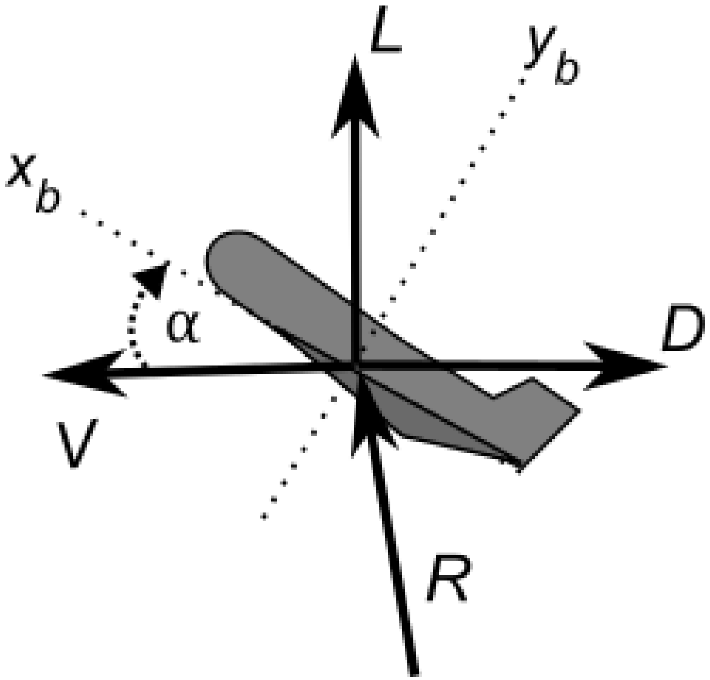
06/12/2024

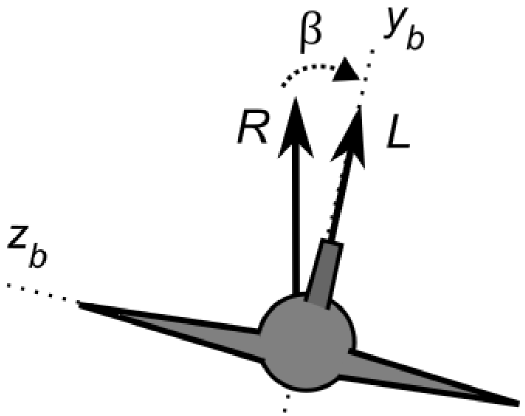
Summary

I. Case of study – Reentry gliding.
II. Final remarks.



I. Space shuttle gliding reentry [2, 3].





$$\dot{R} = V \sin\gamma \tag{1}$$

$$\dot{\theta} = \frac{V \cos\gamma \cos A}{R \cos\varphi} \tag{2}$$

$$\dot{\varphi} = \frac{V \cos\gamma \sin A}{R} \tag{3}$$

$$\dot{V} = \frac{-D_{(\alpha)}}{m} - g \sin\gamma \tag{4}$$

$$\dot{\gamma} = \frac{L_{(\alpha)} \cos\beta}{mV} - \frac{g \cos\gamma}{V} + \frac{V \cos\gamma}{R} \tag{5}$$

$$\dot{A} = \frac{L_{(\alpha)} \sin\beta}{mV \cos\gamma} - \frac{V \tan\varphi \cos A \cos\gamma}{R} \tag{6}$$

$$\boldsymbol{U}\{\alpha(t), \beta(t)\} \tag{7}$$

The Optimal Control Problem - Constraints

$$t_0 < t \leq t_f$$

$$1000\ s \leq t_f \leq 3000\ s$$

$$24384\ km \leq h(t) \leq 79248 km$$

$$V(t) \leq 7802.88$$

$$0\ deg \leq \theta(t) \leq 180\ deg$$

$$-90\ deg \leq \varphi(t) \leq 90\ deg$$

$$-89\ deg \leq \gamma(t) \leq 89\ deg$$

$$0\ deg \leq A(t) \leq 180.0\ deg$$

$$7.4\ deg \leq \alpha(t) \leq 18.0\ deg$$

$$-90 deg \leq \beta(t) \leq 1.0\ deg$$

Additional assumptions

- Non-rotational spherical Earth.
- Static isothermal exponential atmosphere.
- Space shuttle aerodynamic model.

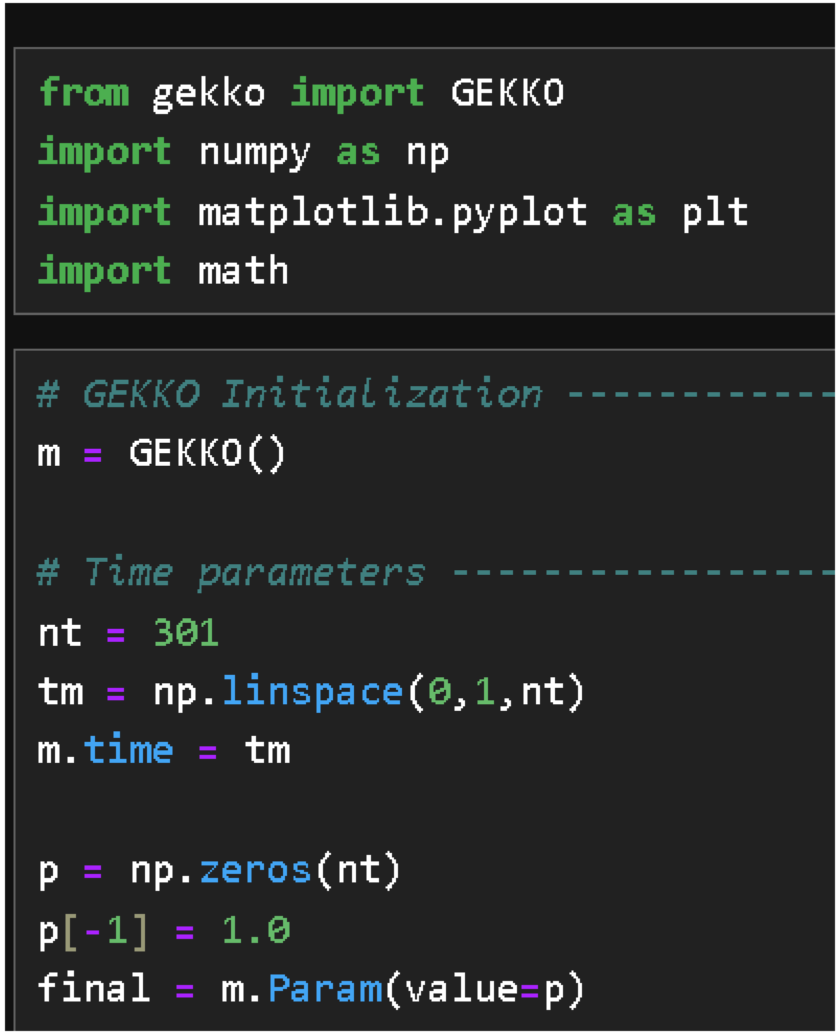
Verification and validation, V&V
The reentry problem.

- Initial conditions:
  - Altitude = 79248 m
  - Velocity = 7802.88 m/s
  - FPA = -1 deg
  - AZI = 90 deg
  - LON & LAT = 0.0 deg

- Final conditions:
  - Altitude: 24384 m
  - Velocity: 762 m/s
  - FPA: -5 deg

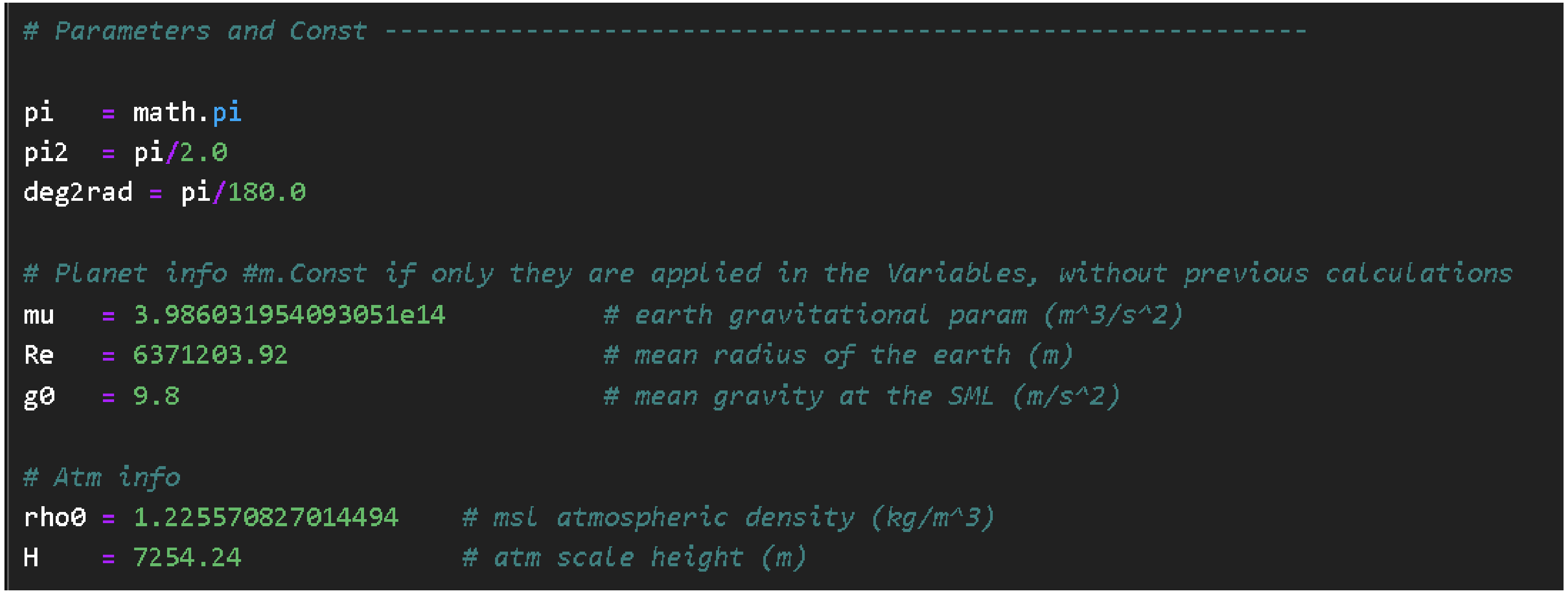
Objective: Maximize cross-range or:

$$\boldsymbol{J} = \max_{t_f}(\varphi)$$

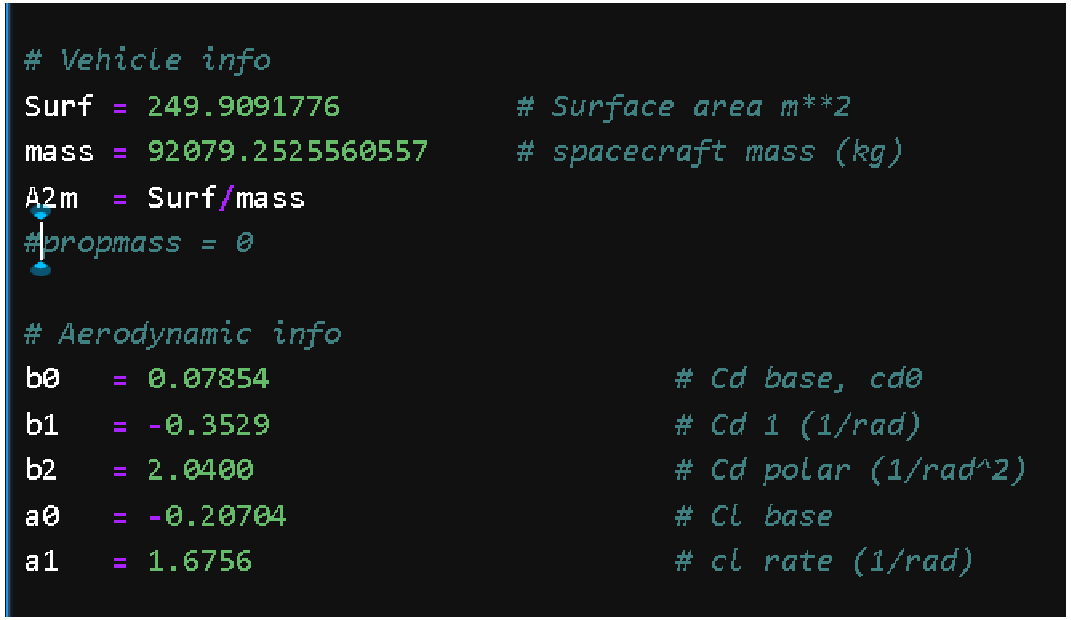


```python
from gekko import GEKKO
import numpy as np
import matplotlib.pyplot as plt
import math
```

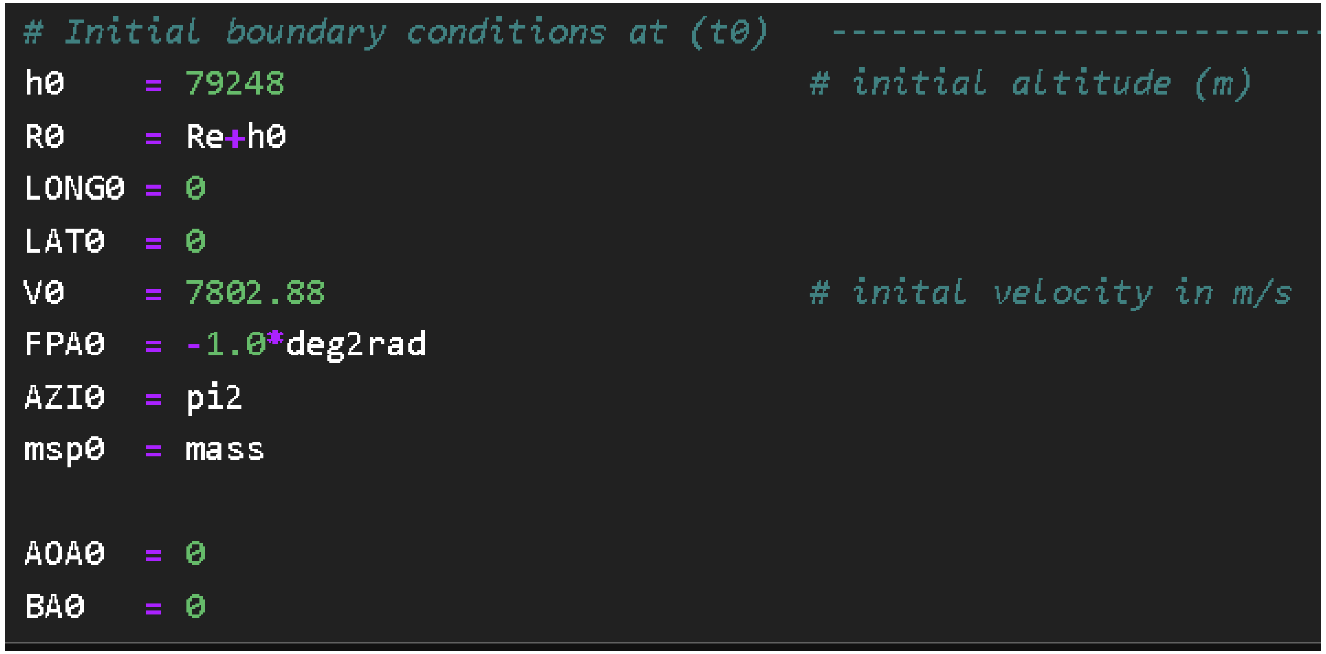
```python
# GEKKO Initialization ------------
m = GEKKO()

# Time parameters -----------------
nt = 301
tm = np.linspace(0,1,nt)
m.time = tm

p = np.zeros(nt)
p[-1] = 1.0
final = m.Param(value=p)
```



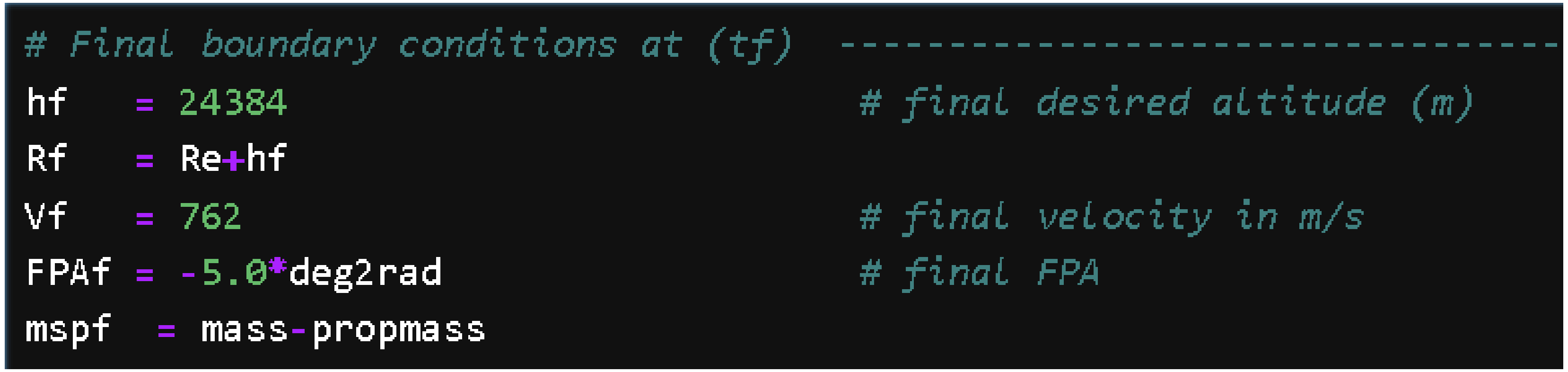
```
# Parameters and Const -----------------------------------------------------------------

pi   = math.pi
pi2  = pi/2.0
deg2rad = pi/180.0

# Planet info #m.Const if only they are applied in the Variables, without previous calculations
mu   = 3.986031954093051e14          # earth gravitational param (m^3/s^2)
Re   = 6371203.92                    # mean radius of the earth (m)
g0   = 9.8                           # mean gravity at the SML (m/s^2)

# Atm info
rho0 = 1.225570827014494    # msl atmospheric density (kg/m^3)
H    = 7254.24              # atm scale height (m)
```



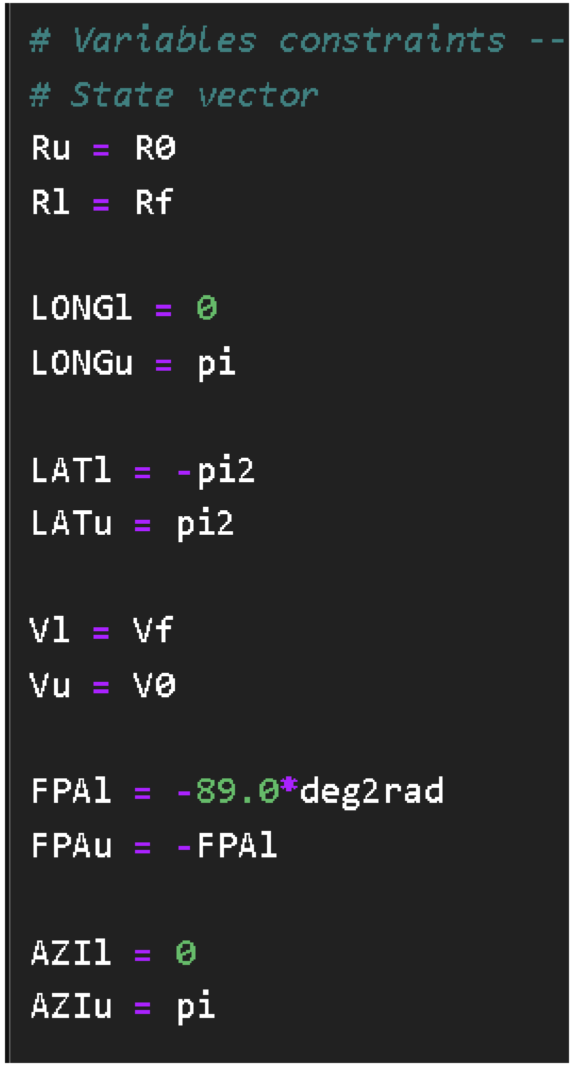
```
# Vehicle info
Surf = 249.9091776          # Surface area m**2
mass = 92079.2525560557     # spacecraft mass (kg)
A2m  = Surf/mass
#propmass = 0

# Aerodynamic info
b0   = 0.07854                       # Cd base, cd0
b1   = -0.3529                       # Cd 1 (1/rad)
b2   = 2.0400                        # Cd polar (1/rad^2)
a0   = -0.20704                      # Cl base
a1   = 1.6756                        # cl rate (1/rad)
```



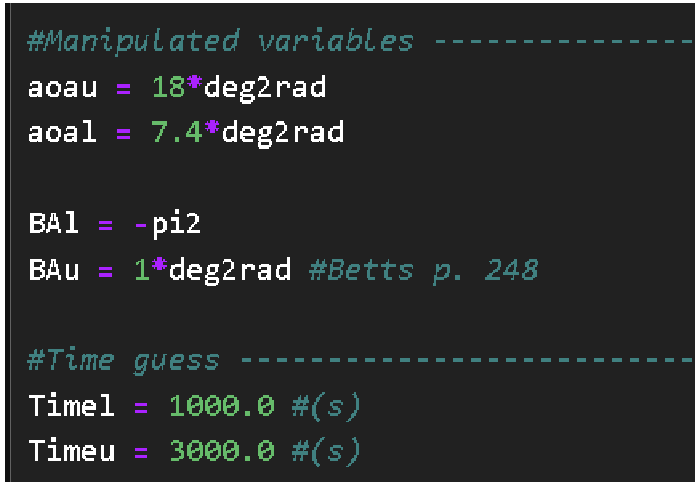
```
# Initial boundary conditions at (t0)   ------------------------
h0     = 79248                          # initial altitude (m)
R0     = Re+h0
LONG0 = 0
LAT0   = 0
V0     = 7802.88                        # inital velocity in m/s
FPA0   = -1.0*deg2rad
AZI0   = pi2
msp0   = mass

AOA0   = 0
BA0    = 0
```



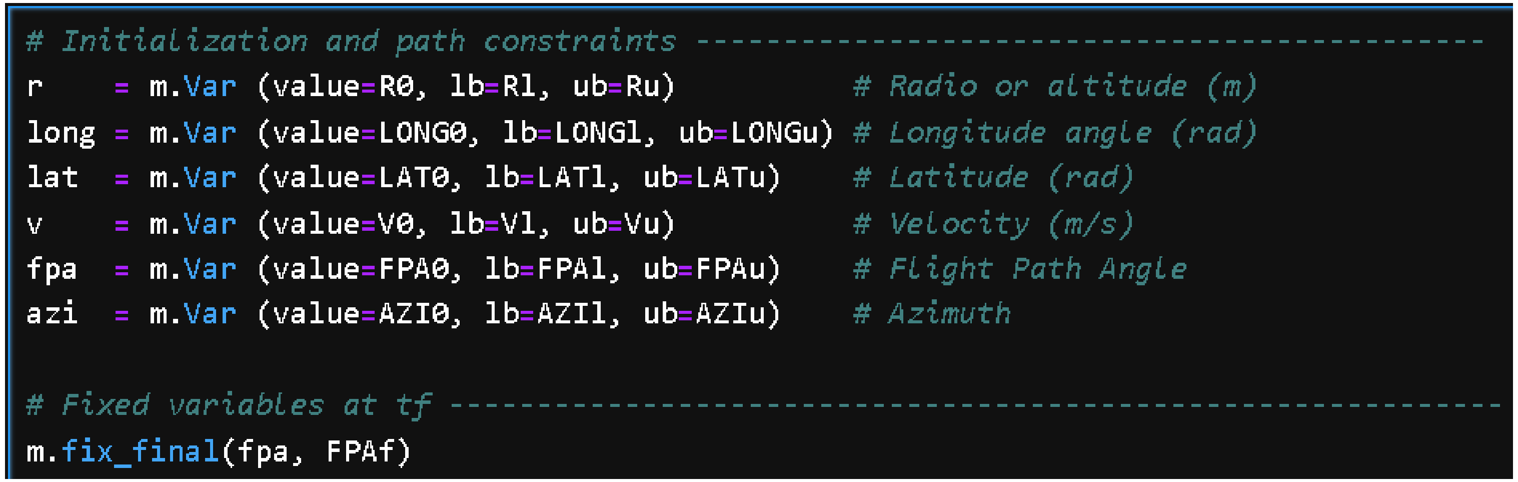
```
# Final boundary conditions at (tf)   ---------------------------------
hf     = 24384                          # final desired altitude (m)
Rf     = Re+hf
Vf     = 762                            # final velocity in m/s
FPAf = -5.0*deg2rad                     # final FPA
mspf   = mass-propmass
```



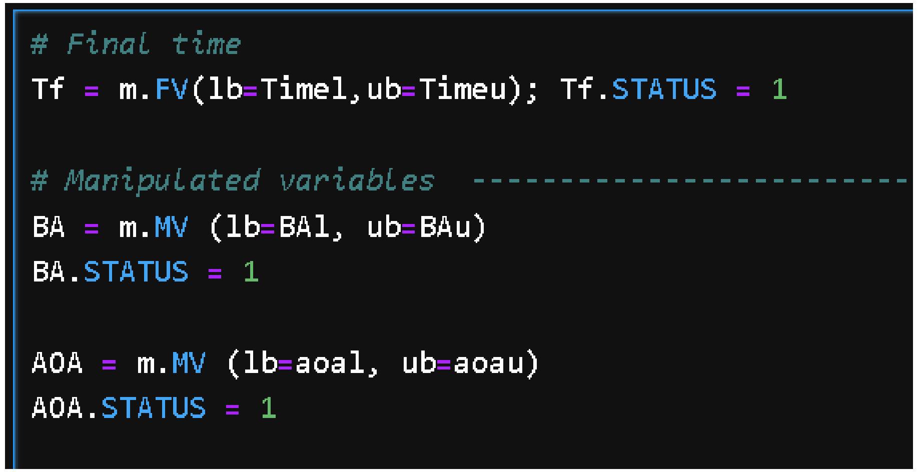
```
# Variables constraints --
# State vector
Ru = R0
Rl = Rf

LONGl = 0
LONGu = pi

LATl = -pi2
LATu = pi2

Vl = Vf
Vu = V0

FPAl = -89.0*deg2rad
FPAu = -FPAl

AZIl = 0
AZIu = pi
```



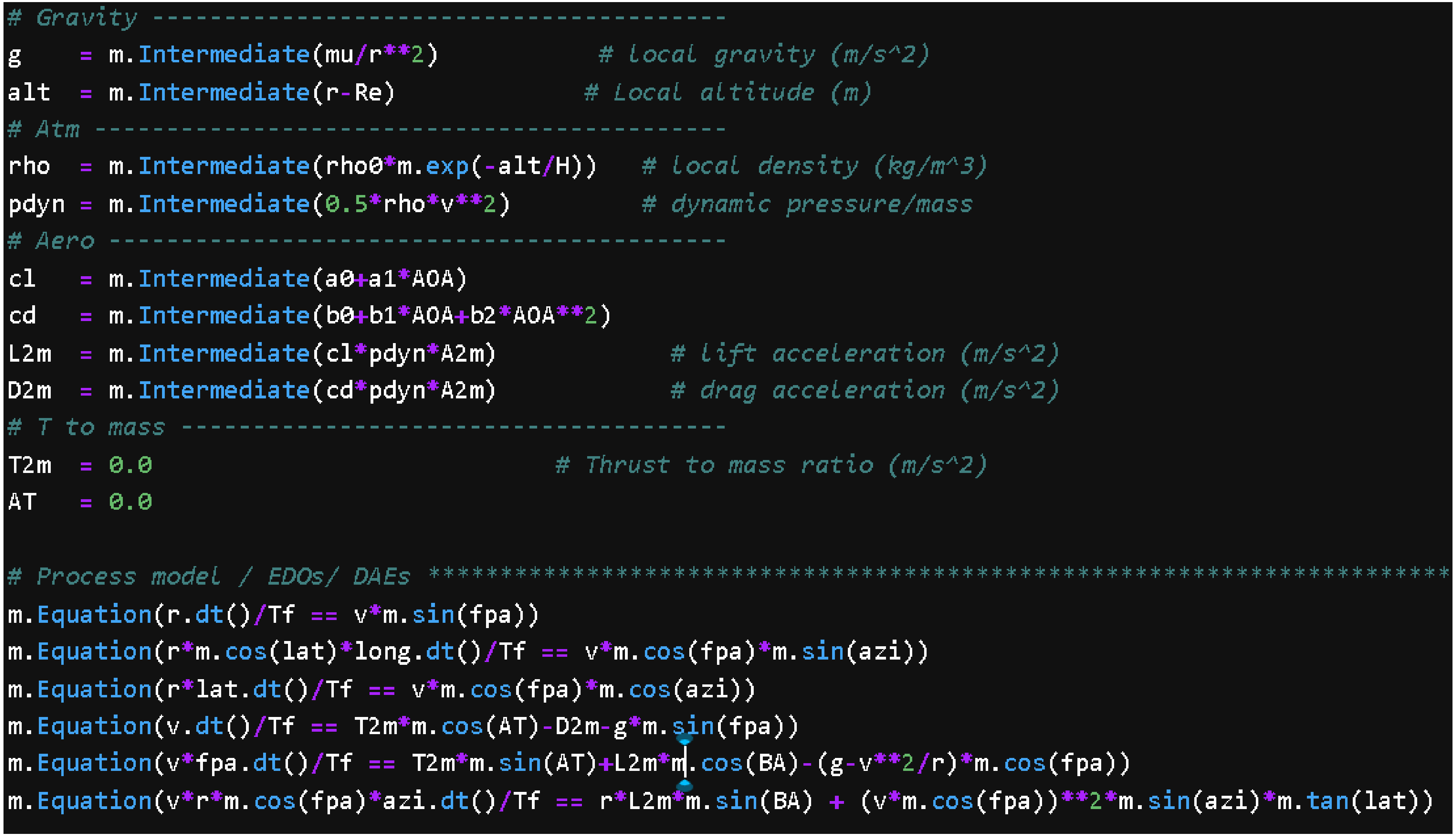
```
#Manipulated variables ---------------
aoau = 18*deg2rad
aoal = 7.4*deg2rad

BAl = -pi2
BAu = 1*deg2rad #Betts p. 248

#Time guess -------------------------
Timel = 1000.0 #(s)
Timeu = 3000.0 #(s)
```



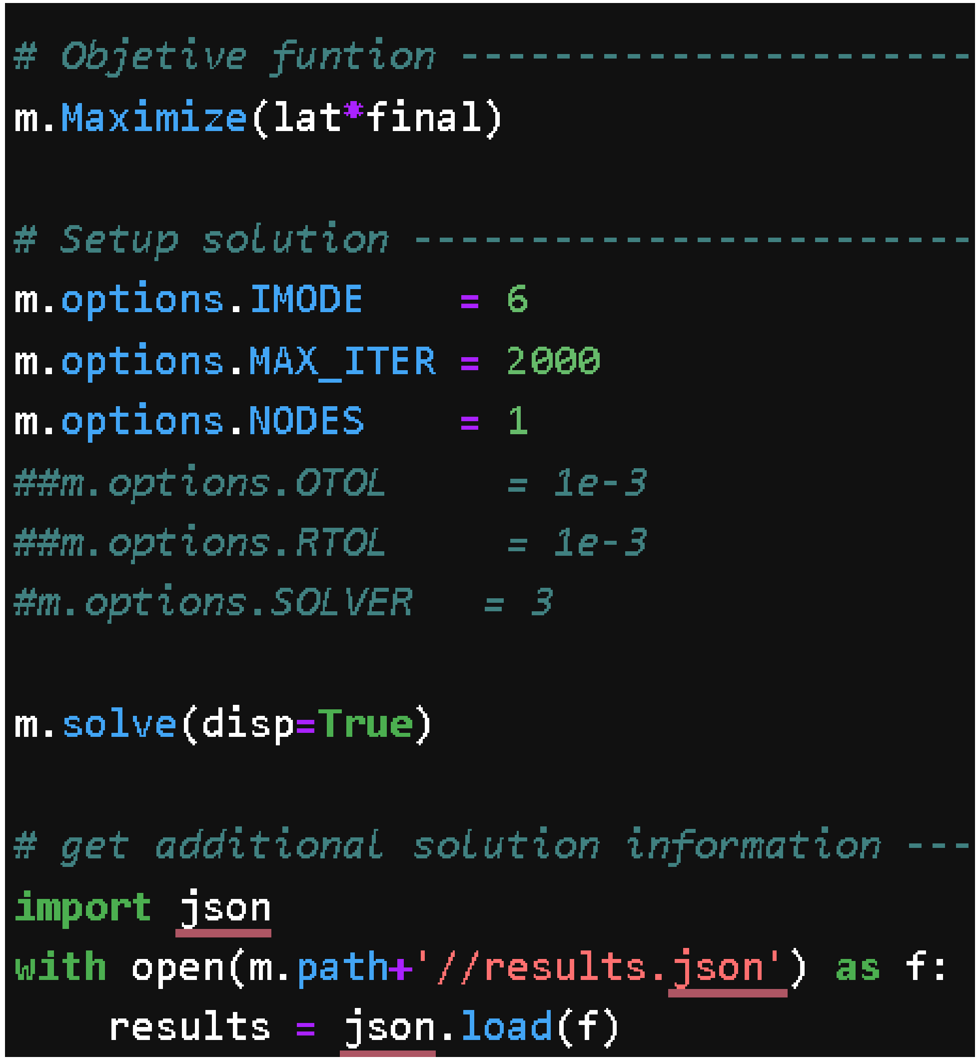
```
# Initialization and path constraints ---------------------------------------------
r    = m.Var (value=R0, lb=Rl, ub=Ru)          # Radio or altitude (m)
long = m.Var (value=LONG0, lb=LONGl, ub=LONGu) # Longitude angle (rad)
lat  = m.Var (value=LAT0, lb=LATl, ub=LATu)    # Latitude (rad)
v    = m.Var (value=V0, lb=Vl, ub=Vu)          # Velocity (m/s)
fpa  = m.Var (value=FPA0, lb=FPAl, ub=FPAu)    # Flight Path Angle
azi  = m.Var (value=AZI0, lb=AZIl, ub=AZIu)    # Azimuth

# Fixed variables at tf -------------------------------------------------------------
m.fix_final(fpa, FPAf)
```



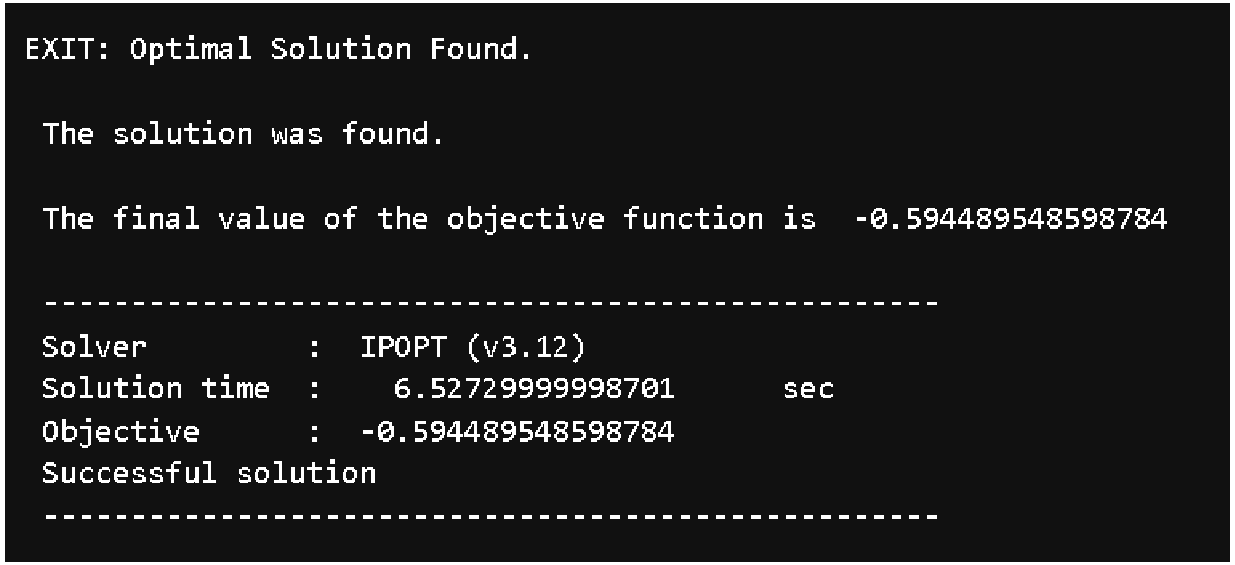
```
# Final time
Tf = m.FV(lb=Timel,ub=Timeu); Tf.STATUS = 1

# Manipulated variables  -------------------------
BA = m.MV (lb=BAl, ub=BAu)
BA.STATUS = 1

AOA = m.MV (lb=aoal, ub=aoau)
AOA.STATUS = 1
```



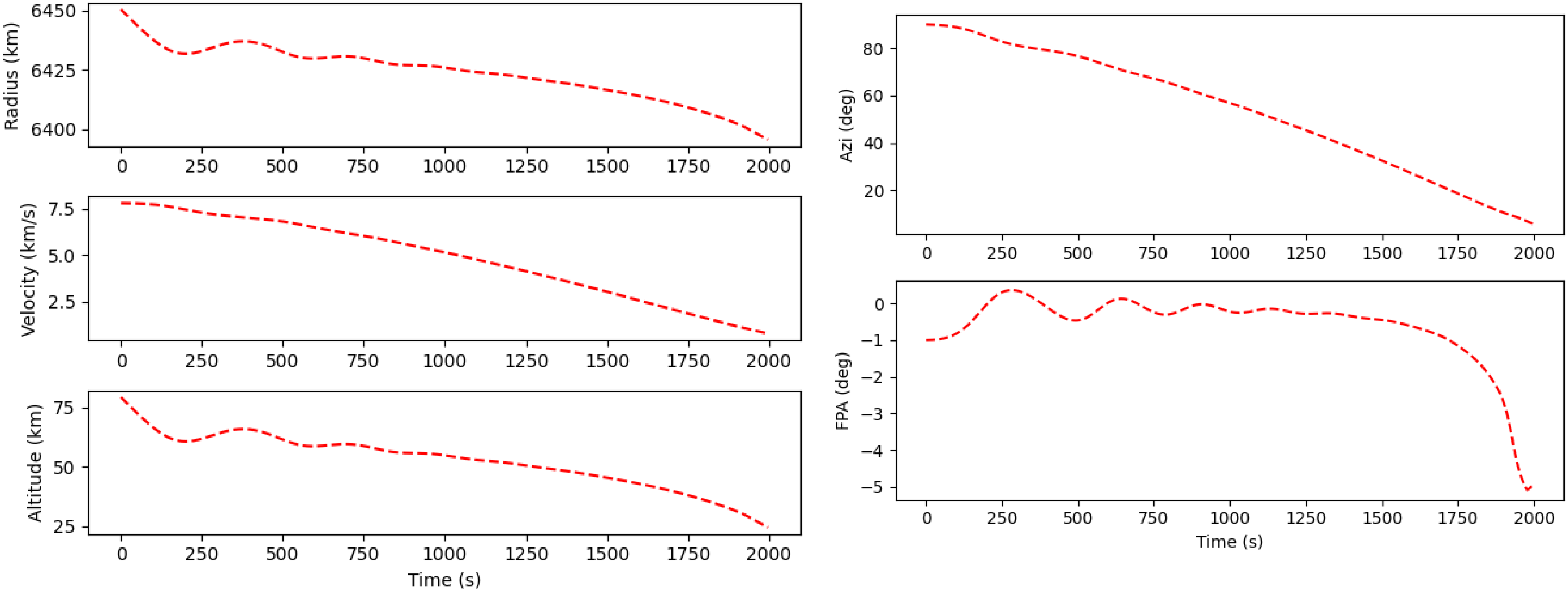
```
# Gravity ----------------------------------------
g    = m.Intermediate(mu/r**2)           # local gravity (m/s^2)
alt  = m.Intermediate(r-Re)             # Local altitude (m)
# Atm --------------------------------------------
rho  = m.Intermediate(rho0*m.exp(-alt/H))   # local density (kg/m^3)
pdyn = m.Intermediate(0.5*rho*v**2)         # dynamic pressure/mass
# Aero -------------------------------------------
cl   = m.Intermediate(a0+a1*AOA)
cd   = m.Intermediate(b0+b1*AOA+b2*AOA**2)
L2m  = m.Intermediate(cl*pdyn*A2m)            # lift acceleration (m/s^2)
D2m  = m.Intermediate(cd*pdyn*A2m)            # drag acceleration (m/s^2)
# T to mass --------------------------------------
T2m  = 0.0                            # Thrust to mass ratio (m/s^2)
AT   = 0.0

# Process model / EDOs/ DAEs ***********************************************************************
m.Equation(r.dt()/Tf == v*m.sin(fpa))
m.Equation(r*m.cos(lat)*long.dt()/Tf == v*m.cos(fpa)*m.sin(azi))
m.Equation(r*lat.dt()/Tf == v*m.cos(fpa)*m.cos(azi))
m.Equation(v.dt()/Tf == T2m*m.cos(AT)-D2m-g*m.sin(fpa))
m.Equation(v*fpa.dt()/Tf == T2m*m.sin(AT)+L2m*m.cos(BA)-(g-v**2/r)*m.cos(fpa))
m.Equation(v*r*m.cos(fpa)*azi.dt()/Tf == r*L2m*m.sin(BA) + (v*m.cos(fpa))**2*m.sin(azi)*m.tan(lat))
```

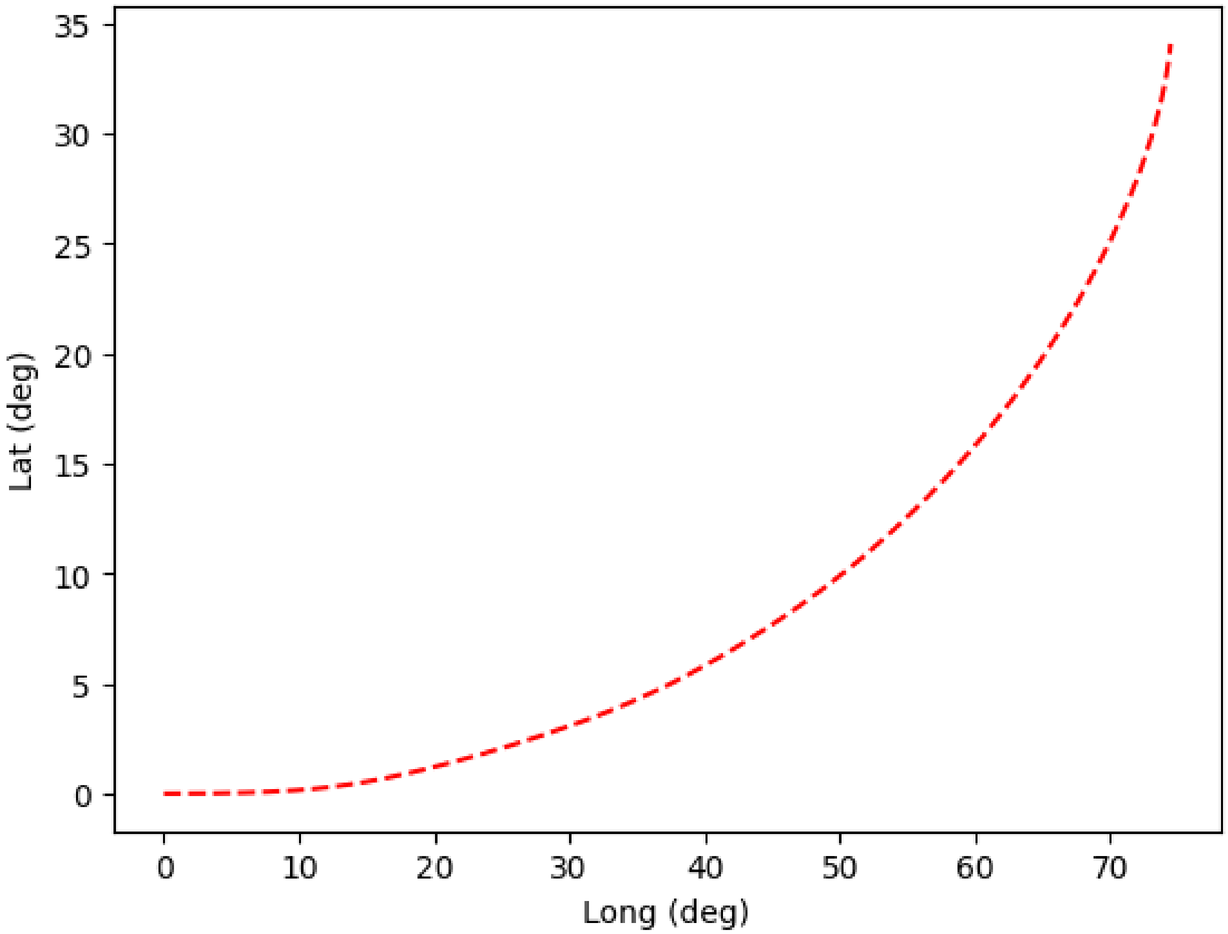


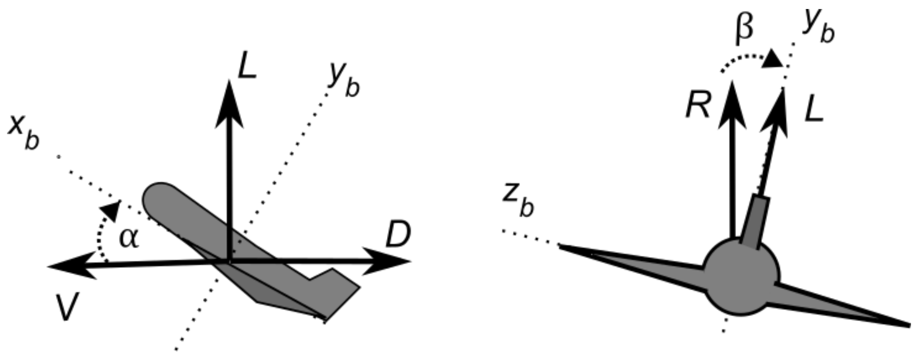
```
# Objetive funtion ----------------------
m.Maximize(lat*final)

# Setup solution ------------------------
m.options.IMODE    = 6
m.options.MAX_ITER = 2000
m.options.NODES    = 1
##m.options.OTOL     = 1e-3
##m.options.RTOL     = 1e-3
#m.options.SOLVER   = 3

m.solve(disp=True)

# get additional solution information ---
import json
with open(m.path+'//results.json') as f:
    results = json.load(f)
```

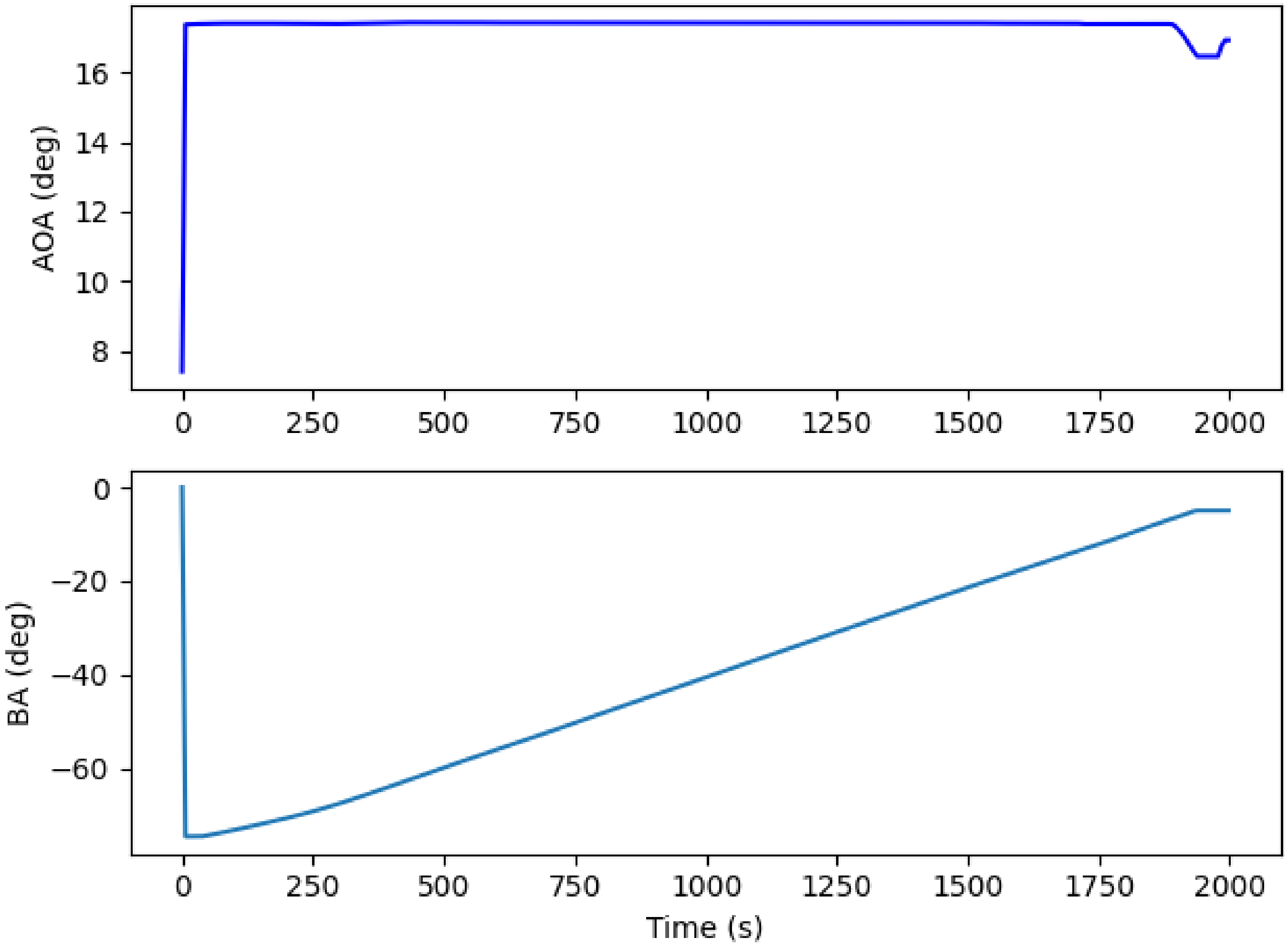


```
EXIT: Optimal Solution Found.

 The solution was found.

 The final value of the objective function is  -0.594489548598784

 ---------------------------------------------------
 Solver         :  IPOPT (v3.12)
 Solution time  :    6.52729999998701      sec
 Objective      :  -0.594489548598784
 Successful solution
 ---------------------------------------------------
```


Radius (km)
6450
6425
6400
Velocity (km/s)
7.5
5.0
2.5
Altitude (km)
75
50
25
0
250
500
750
1000
1250
1500
1750
2000
Time (s)
Azi (deg)
80
60
40
20
FPA (deg)
0
−1
−2
−3
−4
−5
Time (s)


Lat (deg)
35
30
25
20
15
10
5
0
Long (deg)
0
10
20
30
40
50
60
70


x_b
L
y_b
α
D
V
β
y_b
R
L
z_b


AOA (deg)
16
14
12
10
8
0
250
500
750
1000
1250
1500
1750
2000
BA (deg)
0
−20
−40
−60
0
250
500
750
1000
1250
1500
1750
2000
Time (s)

II. Final remarks

I invite you to follow the work of the researchers:

- PhD. Omkar Mulekar – Optimal control for landers based on ML.

Johnson Space Center - NASA

https://scholar.google.com/citations?hl=en&user=5HTQrk4AAAAJ

- PhD(C). Emanuela Gaglio – ML based optimal control for aeromaneuvers.
- Optimal drag-based collision avoidance: Balancing miss distance and orbital decay. Acta Astronautica, 2024.

Scuola Superiore Meridionale - Italy

https://scholar.google.com/citations?hl=en&user=5HTQrk4AAAAJ

- PhD(C). Luis Mendoza Zambrano – Optimal Control on solar sailing and cislunar trajectories.

ADAMUS Lab, ERAU - USA

https://orcid.org/0000-0002-0252-3791

II. Final remarks

I invite you to follow the work of the researchers:

- PhD. Omkar Mulekar – Optimal control for landers based on ML.

Johnson Space Center - NASA

https://scholar.google.com/citations?hl=en&user=5HTQrk4AAAAJ

- PhD(C). Emanuela Gaglio – ML based optimal control for aeromaneuvers.

Scuola Superiore Meridionale - Italy

https://scholar.google.com/citations?hl=en&user=5HTQrk4AAAAJ

- PhD(C). Luis Mendoza Zambrano – Optimal Control on solar sailing and cislunar trajectories.

ADAMUS Lab, ERAU - USA

https://orcid.org/0000-0002-0252-3791

References

[1] Armellin, R., Lavagna, M., and Ercoli-Finzi, A.,(2006) Aero-gravity assist maneuvers: controlled dynamics modeling and optimization, Celestial Mechanics and Dynamical Astronomy, Vol. 95,, pp. 391–405. doi: 10.1007/s10569-006-9024-y

[2] Betts J. (2010) Practical Methods for Optimal Control Using Nonlinear Programming. 3 Ed..

[3] Patterson, M., Rao, A. (2016) GPOPS II Manual. V 2.3. Available at: https://gpops2.com/resources/gpops2UsersGuide.pdf

[4] Murcia-Piñeros, J., Prado, A. F., Dimino, I., & de Moraes, R. V. (2024). Optimal gliding trajectories for descent on Mars. Applied Sciences, 14(17), 7786. Doi: 10.3390/app14177786

[5] Piñeros, J. O. M., Bevilacqua, R., Prado, A. F., & de Moraes, R. V. (2024). Optimizing aerogravity-assisted maneuvers at high atmospheric altitude above Venus, Earth, and Mars to control heliocentric orbits. Acta Astronautica, 215, 333-347. Doi: 10.1016/j.actaastro.2023.12.017

[6] Murcia-Piñeros, J., Bevilacqua, R., Gaglio, E., Prado, A. B., & De Moraes, R. V. (2024). Optimization of Aero-gravity assisted maneuvers for spaceplanes at high atmospheric flight on Earth. In AIAA SCITECH 2024 Forum (p. 1459). Doi: 10.2514/6.2024-1459.

[7] Beal, L.D.R., Hill, D., Martin, R.A., and Hedengren, J. D., “GEKKO Optimization Suite”, Processes, Vol. 6, No. 8, 2018. doi: 10.3390/pr6080106.

[8] Hedengren, J. D., Asgharzadeh Shishavan, R., Powell, K.M., and Edgar, T.F., “Nonlinear Modeling, Estimation and Predictive Control in APMonitor”, Computers and Chemical Engineering, Vol. 70, 2014, pp. 133–148. doi: 10.1016/j.compchemeng.2014.04.013.

Acknowledges

- Organizational committee of the CBDO.
- Professors and participants.
- FAPESP, UNIFESP, INPE.

Thank you and have fun!

Questions?