MAE3145: PROPOGATE

Description

The objective for this project is to write a computer program which first determines the classical orbital elements of an Earth orbiting body given its current position and velocity vectors. Using these orbital elements your program must then compute the future classical orbital elements given a time of flight using a function called `Update`, which uses the restricted 2-Body assumption. Finally, your program should output these results to a text file for later printing/analysis.

Project Requirements

After completing the project you must submit the hard copies of your work:

- Complete algorithms for the main driver script as well as a seperate algorithm for each sub-function that you develop. Someone totally unfamiliar with astrodynamics should be able to duplicate your program in any computer language.
- Clear, concise and properly documented and tested code
- Correct outputs from your program which matches the test cases
- Any additional test cases you may have used. Explain why you did or did not use any additional test cases.

Authorized Resources

You may consult with your instructor, the course notes or other reference material, and other students. However, you **MAY NOT** copy another student or any other code. The program you develop must be your own work.

Program Specifications

The following is a description of the inputs and outputs for your program:
INPUTS:

- $R \in \mathbb{R}^3$ - the components of the position vector in the Earth Centered Inertial frame given in kilometers (km)
- $V \in \mathbb{R}^3$ - the components of the velocity vector in the Earth Centered Inertial frame given in kilometers per second ($\mathrm{km\,s^{-1}}$)
- $\delta t \in \mathbb{R}^1$ - the time of flight of the spacecraft in minutes (min)

OUTPUTS: The following orbital elements are at the final time $t_0 + \delta t$.

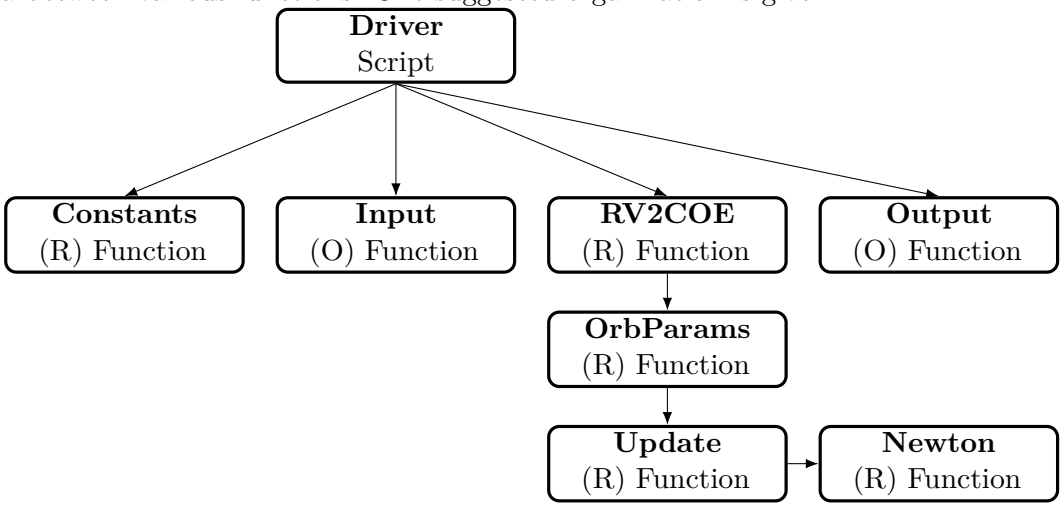
- a - semimajor axis in kilometers (km)
- e - eccentricity (unitless)
- i - inclination in radians (rad)
- Ω - right ascension of the ascending node in radians (rad)

- ω - arguement of perigee in radians (rad)
- ν - true anomaly in radians (rad)

Plus other outputs which are listed later in the description.

Code Organization

Your program should be modular in constructions. This means you have seperate functions which perform **specific** tasks. Furthermore, each function should be tested and properly documented following the example shown in class. You **MAY NOT** use global variables but rather must pass data between various functions. One suggested organization is given:



Required Function Description

The following functions must be written and accept the given inputs and provide the required outputs. Each function should be well tested, with unit tests, and well-documented following the class example.

`Update`

This function "updates" the orbital elements given a time of flight using the restricted two-body assumptions. This function also calls `newton` which solves Kepler's problem using a Newton iteration scheme in order to solve for the future eccentric anomaly E_f.
Input:

- a_i - semimajor axis in kilometers (km)
- e_i - eccentricity (unitless)
- i_i - inclination in radians (rad)
- Ω_i - right ascension of the ascending node in radians (rad)
- ω_i - arguement of perigee in radians (rad)
- ν_i - true anomaly in radians (rad)

OUTPUTS:

- a_i - semimajor axis in kilometers (km)
- e_i - eccentricity (unitless)
- i_i - inclination in radians (rad)
- Ω_i - right ascension of the ascending node in radians (rad)
- ω_i - arguement of perigee in radians (rad)
- ν_i - true anomaly in radians (rad)

newton

This function uses an iterative method to solve Kepler's problem.
Input:

- M - mean anomaly in radians (rad)
- e - eccentricity of orbit

Output:

- E - eccentric anomaly in radians (rad)
- ν - true anomaly in radians (rad)

Test Data

Two test cases are provided with this project. There is a text file named `RV2.txt` which contains the position and velocity vectors of two spacecraft. The data is copied below.

```
8840.0  646.0 5455.0  -0.695  5.25  -1.65 0.0
-3084.7  30.0  6911.0   5.66  -4.07   3.84 0.0
```

The numbers on each line corresponds to the three elements of the position vector and the three elements of the velocity vector followed by the time of flight, i.e. $\begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} & \delta t \end{bmatrix}$. The solution for Case 1, the first line/spacecraft, is given below.

```
R (km)    =  8840.0000 I    646.0000 J   5455.0000 K    Mag= 10407.6866
V (km/s)  =    -0.6950 I      5.2500 J     -1.6500 K    Mag=     5.5469

Radius of Perigee           (km) =  6260.5311
Radius of Apogee            (km) = 11134.4744
Energy              (km^2/sec^2) =   -22.9147
Period                   (hours) =     2.2423
Semimajor Axis              (km) =  8697.5027
Eccentricity                     =     0.2802

Inclination                           (deg) =    33.9987
Right Ascension of the Ascending Node (deg) =   250.0287
```

```
Argument of Perigee                    (deg) =   255.5372
True Anomaly                           (deg) =   214.8548

Time of Flight  (Minutes) =     0.0000
Future True Anomaly (deg) =   214.8548
```

The cases in `RV2.txt` have a time of flight of 0 min which you should immediately convert to seconds. In addition, since the time of flight is zero, the initial and final orbital elements should be identical.

Using your computer program, you should read the vectors, compute the appropriate values and write them to a seperate text file. Do not hardcode the data into your program, but rather read them from the file. While only two cases are given here, your program should be able to convert an arbitrary amount of position and velocity vectors and time of flights.

After verifying both cases shown in `RV2.txt`, modify the time of flight to one orbital period. This should provide another test with the final orbital elements equal to the initial orbital elements. You are encouraged to create your own test cases to further verify you code. For example, you could test your code using a time of flight of $\frac{1}{4}$ orbital period, which should cause a change of the true anomaly by 90°.

Finally, process the data from `RV3.txt` which should give the following

```
CASE 1:  -4525.4103   -2600.2562   6147.2236   4.6461   2.6696   4.5560   300.0

***************** Case    1 *****************
R (km)    = -4525.4103 I   -2600.2562 J    6147.2236 K    Mag=   8064.0578
V (km/s)  =     4.6461 I       2.6696 J       4.5560 K    Mag=      7.0335

Radius of Perigee (km) =   8061.9974
Radius of Apogee  (km) =   8079.4661
Energy    (km^2/sec^2) =    -24.6942
Period         (hours) =      2.0044
Semimajor Axis    (km) =   8070.7317
Eccentricity           =      0.0011

Inclination                            (deg) =     90.0000
Right Ascension of the Ascending Node (deg) =    209.8812
Argument of Perigee                    (deg) =      9.4539
True Anamaly                           (deg) =     40.2134

Time of Flight  (Minutes) =    300.0000
Future True Anamaly (deg) =    218.0948
```