Solar System & Interstellar Mission

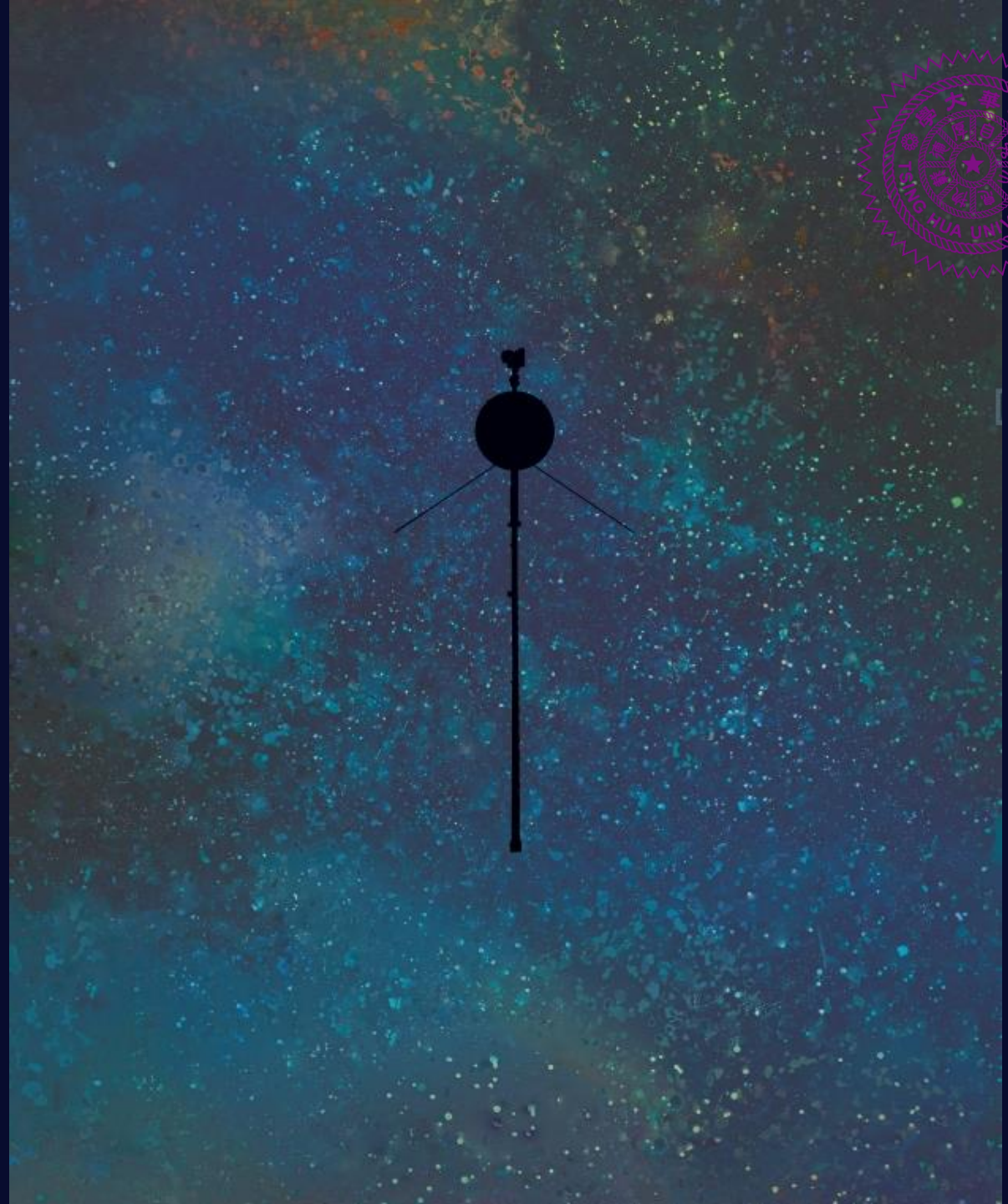
Computational Astrophysics

Final project

NTHU PHYS Wei Hsiang Yu

Outline

- Worksheet
- Prepare for simulation
- Instruction Idea implement
- Problem during coding
- Reference



Worksheet

Prepare for simulation

- Achieve a more real solar system
  - The ephemeris : Time
  - The ephemeris : Location
  - 3-Dimension RK4 method

Problem during coding

- Data manipulation
  - copy() & deepcopy() in Python

Instruction Idea implement

- Interstellar Mission：Voyager 2
  - thruster system
  - Find the minim spend of fuel (unfinished)

Real solar system(1)

1. The ephemeris -> time

Provided by “astropy” package :

UTC time

⬇

JD(Julian Day) time

From Voyager 2 launch time to 10yr

1977/08/20 – 1987/08/20

Time

https://docs.astropy.org/en/stable/time/index.html

[35]:
```
from astropy.time import Time
t         = Time("2021-06-16 03:27")
t_launch = Time("1977-08-20 00:00")
t_arrive = Time("1987-08-20 00:00")
```

[36]:
```
print(t)
```
```
2021-06-16 03:27:00.000
```

[37]:
```
print(t.jd)
```
```
2459381.64375
```

[38]:
```
import numpy as np
time_duration = np.arange(t_launch.jd,t_arrive.jd+1,1)
time_test = np.arange(t_launch.jd,t_launch.jd+365,1)
```

[33]:
```
print(time_duration)
```
```
[2443375.5 2443376.5 2443377.5 ... 2447025.5 2447026.5 2447027.5]
```

Real solar system(2)

2. The ephemeris -> location

Provided by NASA NAIF database (NASA's Navigation and Ancillary Information Facility)

Use “jplephem” can read the file:

given time array

⬇

get each planet location

testforuse.ipynb

Code

Location

https://pypi.org/project/jplephem/

```
[28]: from jplephem.spk import SPK
      kernel = SPK.open('de421.bsp')
      print(kernel)
```

```
File type DAF/SPK and format LTL-IEEE with 15 segments:
2414864.50..2471184.50  Type 2  Solar System Barycenter (0) -> Mercury Barycenter (1)
```

Getting Started With DE421

The DE421 ephemeris is a useful starting point. It weighs in at 17 MB, but provides predictions over the years 1900–2050:

https://naif.jpl.nasa.gov/pub/naif/generic_kernels/spk/planets/a_old_versions/de421.bsp

After the kernel has downloaded, you can use `jplephem` to load this SPK file and learn about the segments it offers:

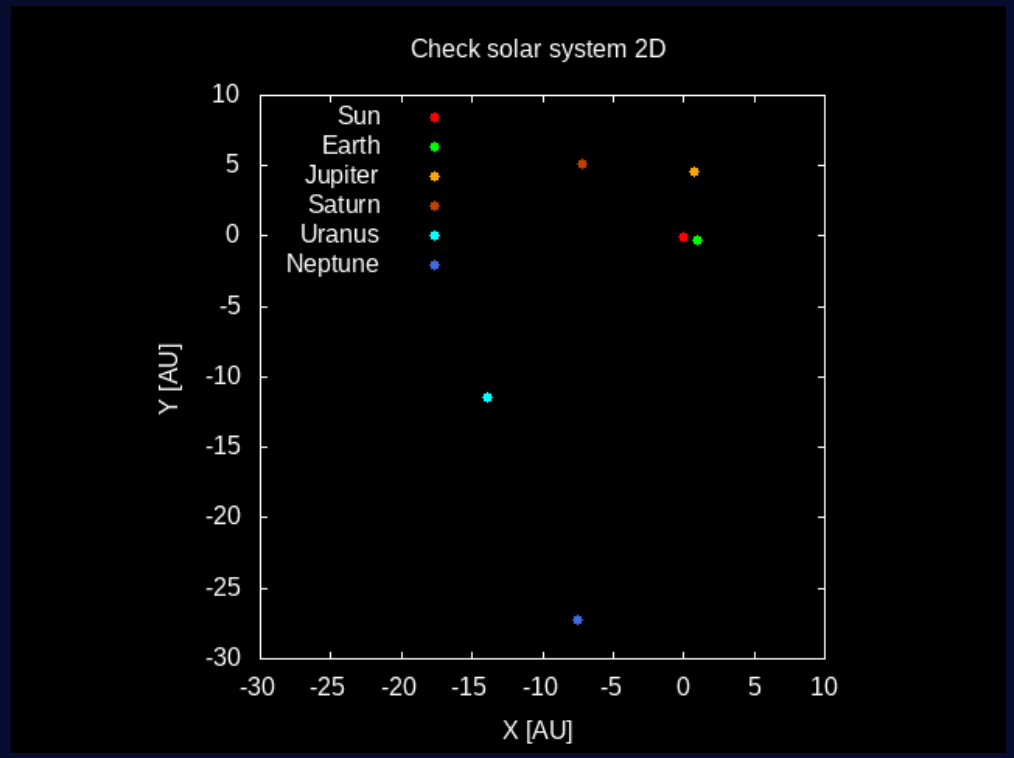
```
2414864.50..2471184.50  Type 2  Mars Barycenter (4) -> Mars (499)
```

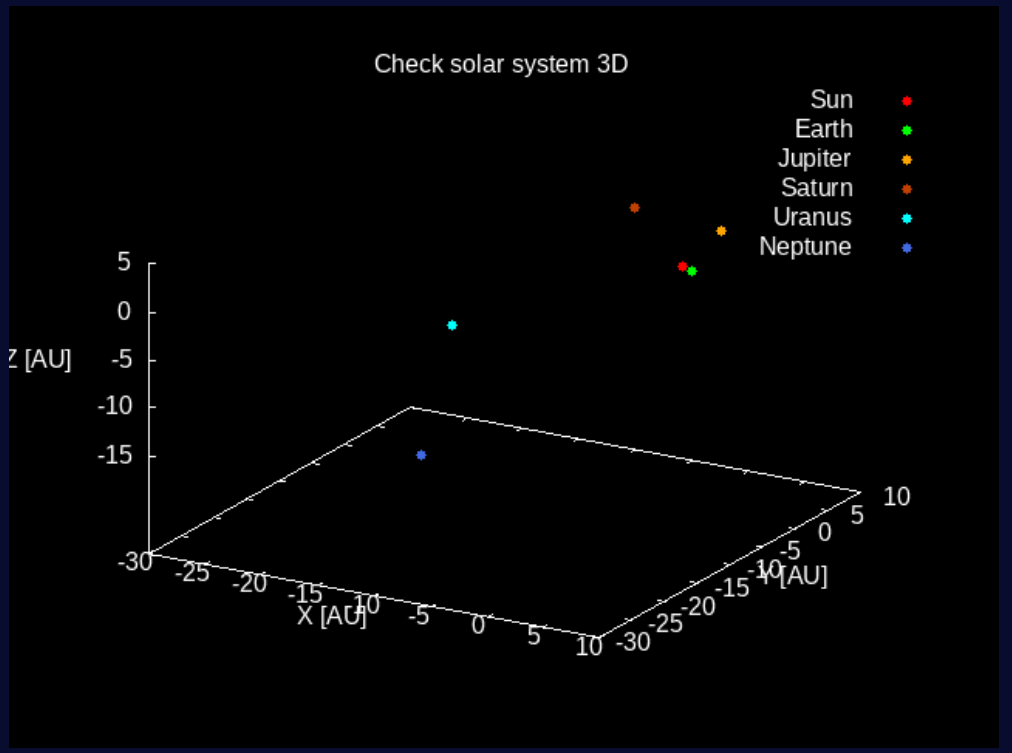
```
[29]: #use Solar System Barycenter(sun index=0) as orgin to get loaction of earth(index=3)
      position = kernel[0,3].compute(time_test)
      position=position/149597870.7
      print(position)
```

```
[[ 0.8525638  0.8614942  0.8701791 ...  0.8246632  0.834374   0.8438483]
 [-0.5070995 -0.4938217 -0.4804038 ... -0.5486526 -0.5358525 -0.5229004]
 [-0.2199454 -0.2141881 -0.2083702 ... -0.2380494 -0.2324993 -0.2268833]]
```

Real solar system – check the solar system

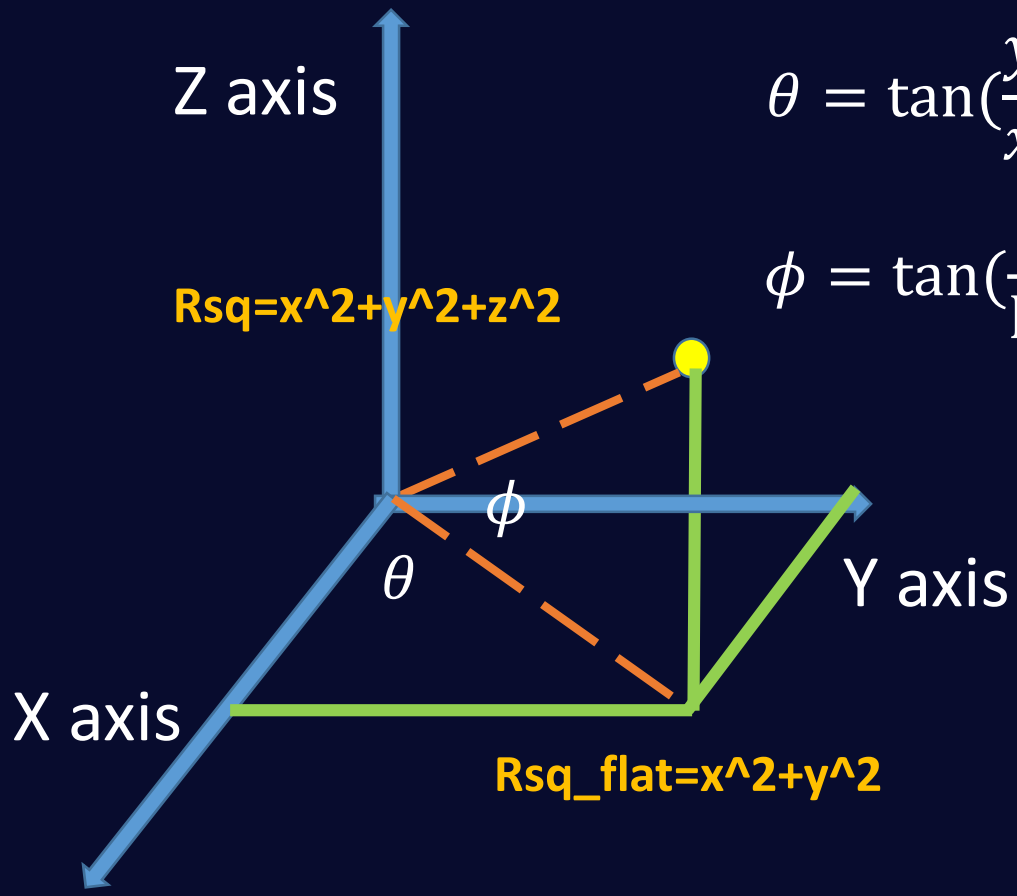
1977/08/20 – 1987/08/20

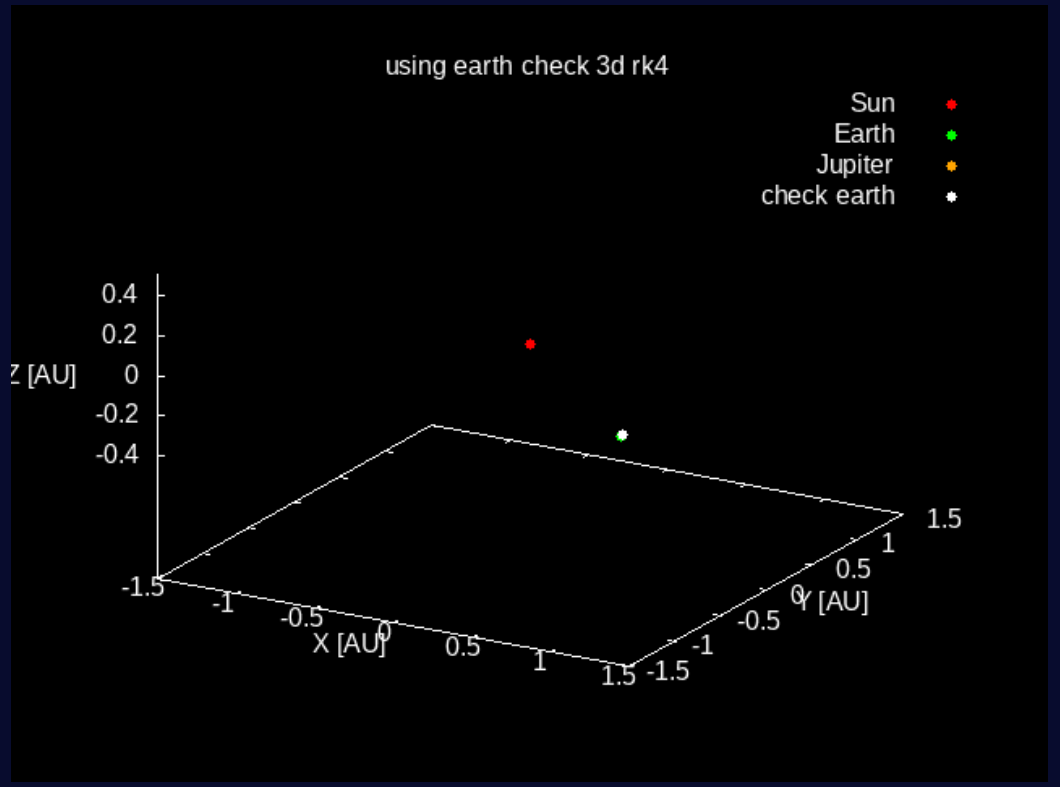




Real solar system(3) : 3D RK4 method

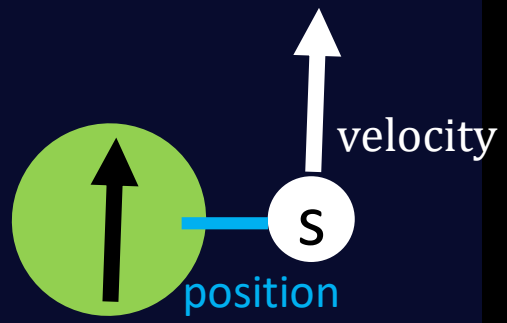
1. Applied spherical coordinate idea



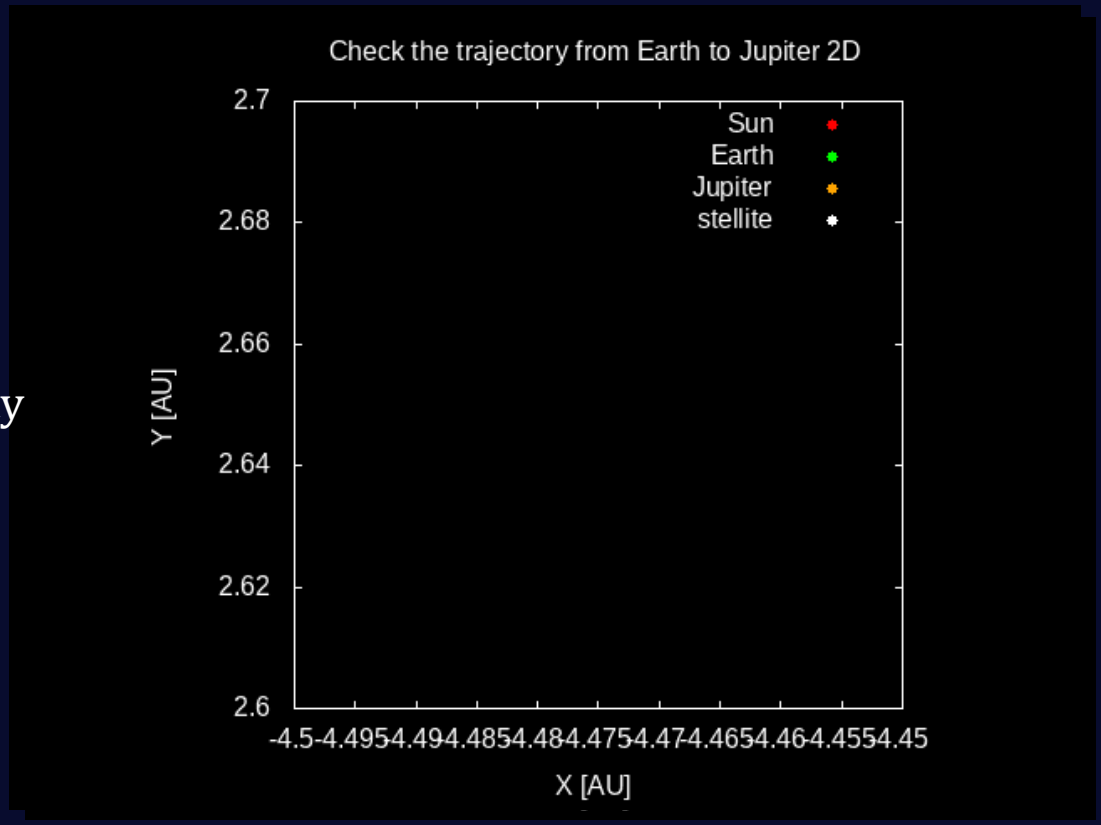


Idea implement(1) : initial condition

$m_{voyager} = 721.9\ kg$
$initial\ velcoity = 38.6\ km/s$
$initial\ position = 100km\ on\ Earth$
$lauch\ time$: 1977.08.22 00: 00



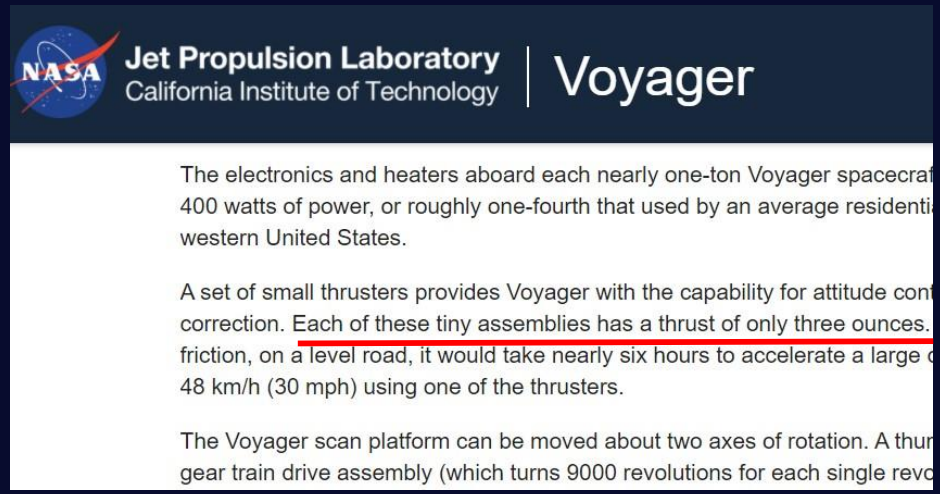
$m_{voyager} = 721.9\ kg$
$initial\ velcoity = 38.621\ km/s$
$initial\ position = 1000km\ on\ Earth$
$lauch\ time$: 1977.09.01 00: 00



Idea implement(2) : thruster $a_{thruster}$

From NASA, Voyager have 16 thrusters (suppose 8 can act in same direction)

Each provide 3 ounces force


Jet Propulsion Laboratory
California Institute of Technology | Voyager

The electronics and heaters aboard each nearly one-ton Voyager spacecraf
400 watts of power, or roughly one-fourth that used by an average residenti
western United States.

A set of small thrusters provides Voyager with the capability for attitude cont
correction. Each of these tiny assemblies has a thrust of only three ounces.
friction, on a level road, it would take nearly six hours to accelerate a large
48 km/h (30 mph) using one of the thrusters.

The Voyager scan platform can be moved about two axes of rotation. A thur
gear train drive assembly (which turns 9000 revolutions for each single revo

$$F_{thruster} = m_{voyager} * a_{thruster}$$

$$8 * \left(\frac{3 * 0.02835[ounces \rightarrow g]}{1000[g \rightarrow kg]} \right) * 9.8 = 721.9[kg] * a_{thruster}$$

Idea implement(3) : thruster dt

speed up time (unit : min)

Simulation update time (unit : hr /day)

$$dt_{update} \; != dt_{speed\ up}$$

$$v = v_0 + dt * a_{planet} + dt' * a_{thruster}$$

$$v = v_0 + dt(a_{planet} + a_{thruster} * \frac{dt'}{dt})$$

dtratio

```python
def euler(dt, yin1, yin2, derive, stell_acc):
    """
    Do one euler update with dt
    """
    dydt    = derive(yin2, stell_acc)
    yout    = yin1 + dt*dydt

    return yout
```

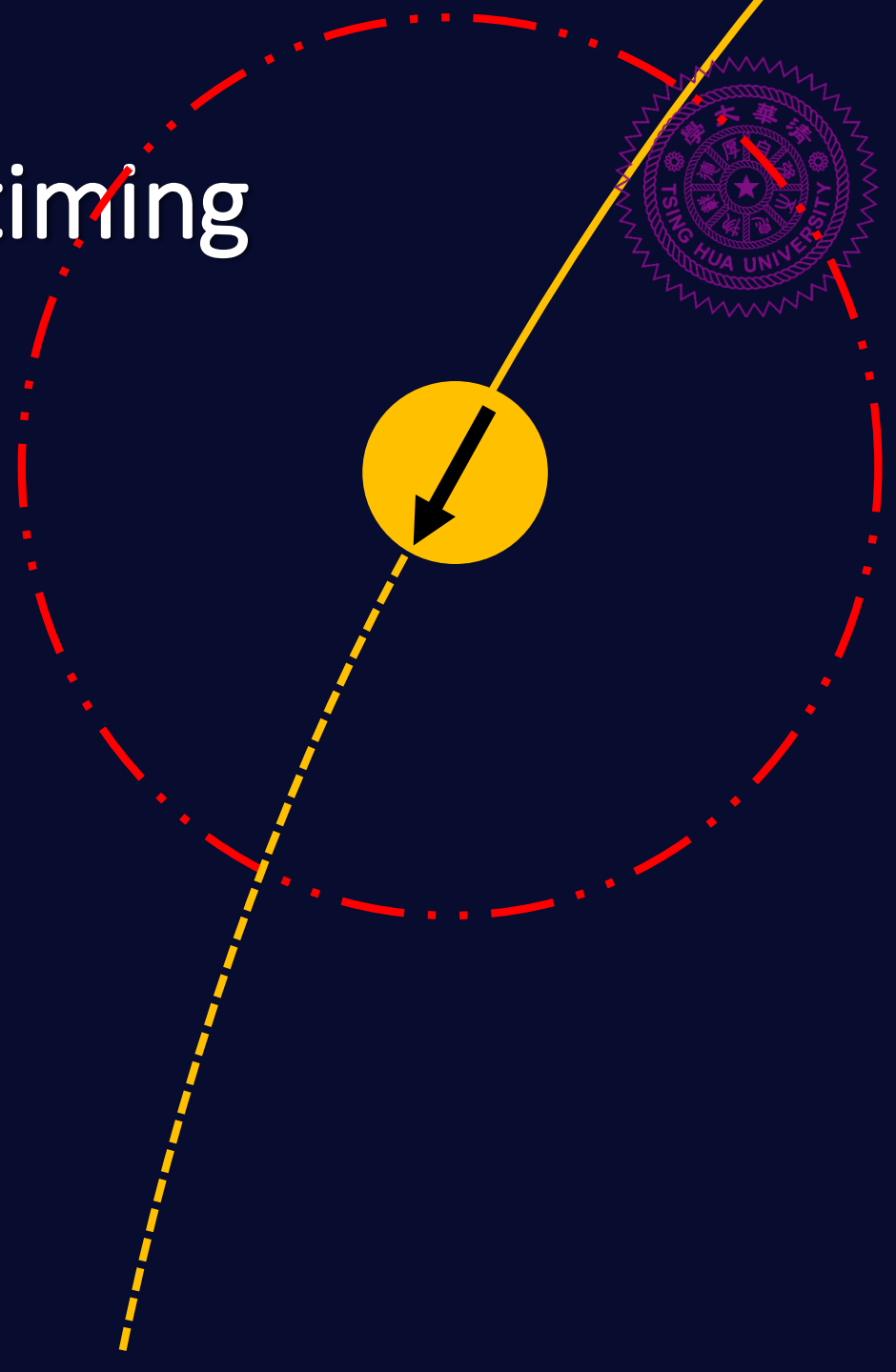
```python
if (speedupinfo.speed_ox):
    speedup_acc_vector = fuel_acc * speedupinfo.direction
    speedup_acc_vector = speedup_acc_vector * speedupinfo.dt_ratio

    stell_acc[0] = stell_acc[0] + speedup_acc_vector[0]
    stell_acc[1] = stell_acc[1] + speedup_acc_vector[1]
    stell_acc[2] = stell_acc[2] + speedup_acc_vector[2]
```

Idea implement(4) : thruster timing

- Can effect when the target planet gravity start to influence our satellite

- Sphere of influence

$$r_{SOI} \approx a\left(\frac{m_{planet}}{M_{sun}}\right)^{\frac{2}{5}}$$

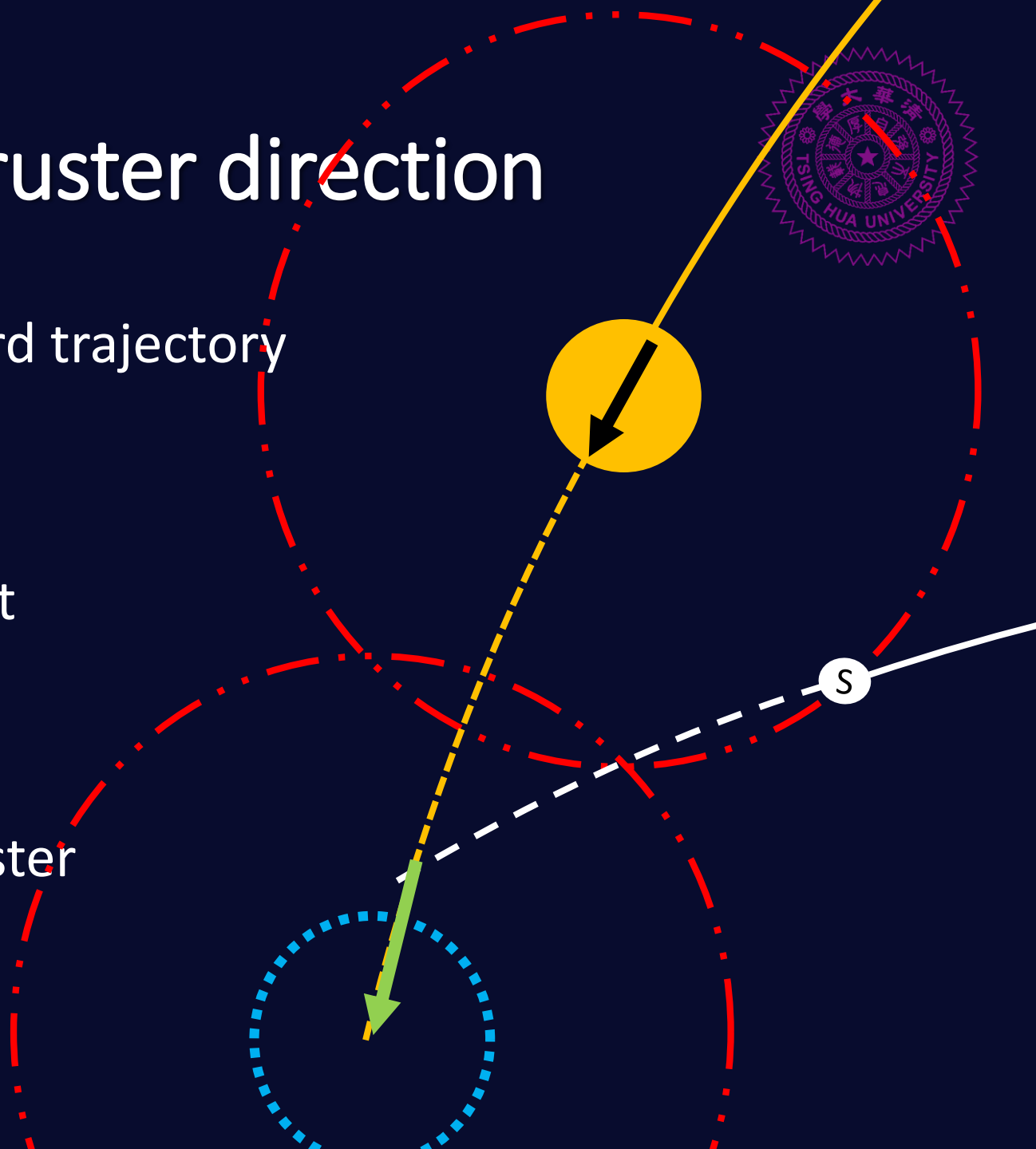


Idea implement(5) : thruster direction

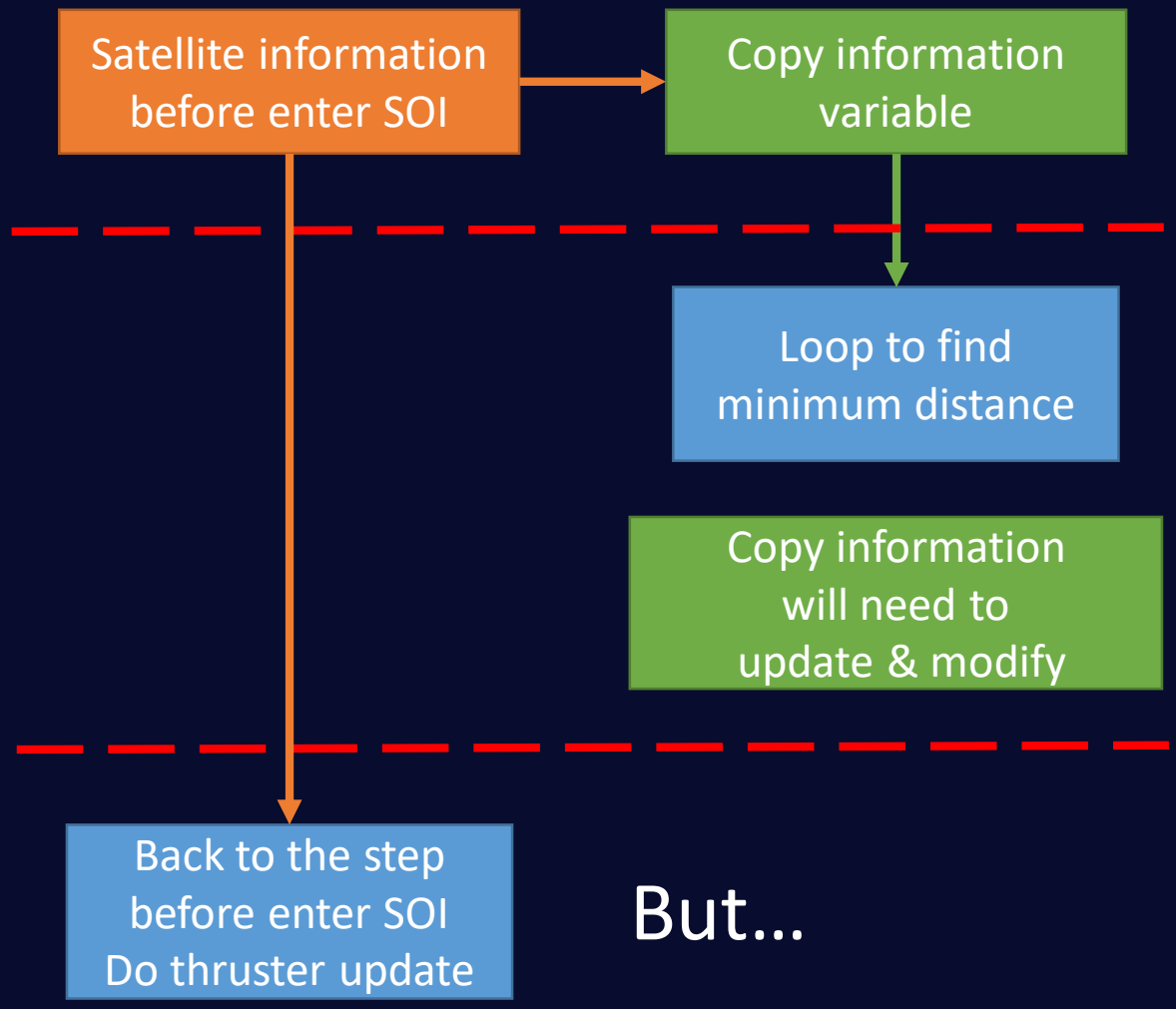
- Use the SOI distance to stop record trajectory
- calculation minimum distance

  voyager2 <-> target planet
- Record the direction angle

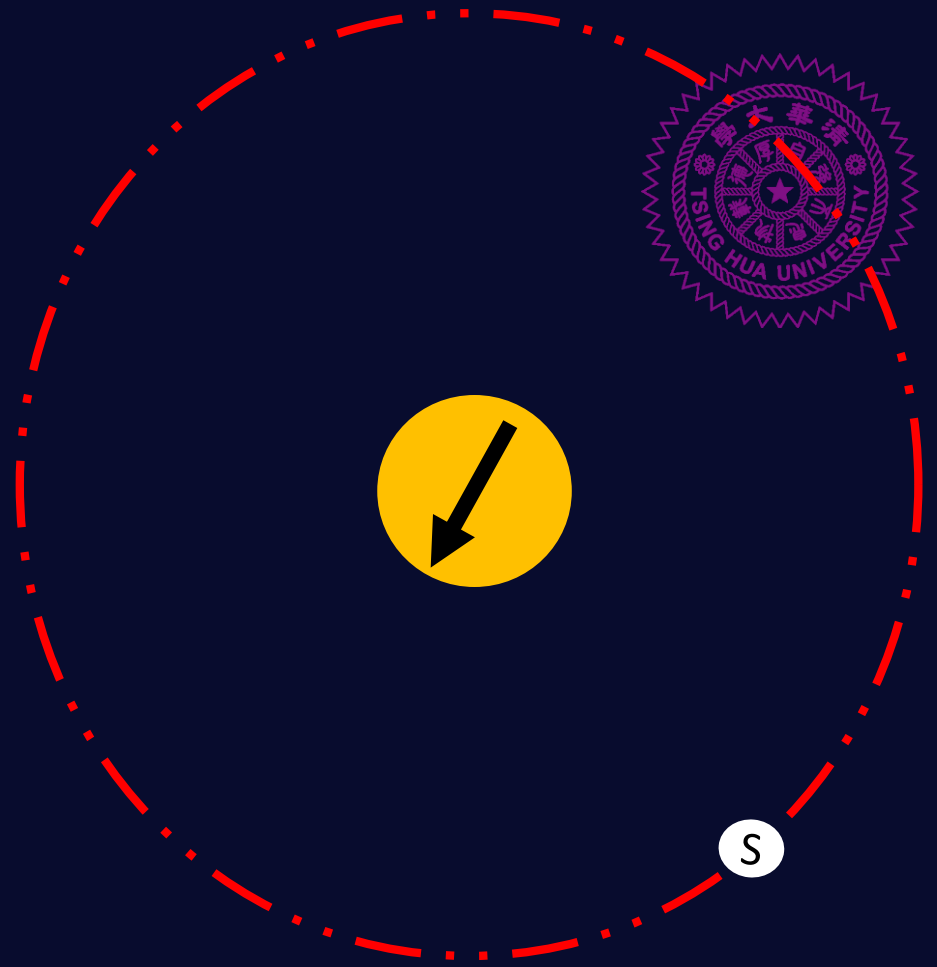
  Back to SOI star point to fire thruster



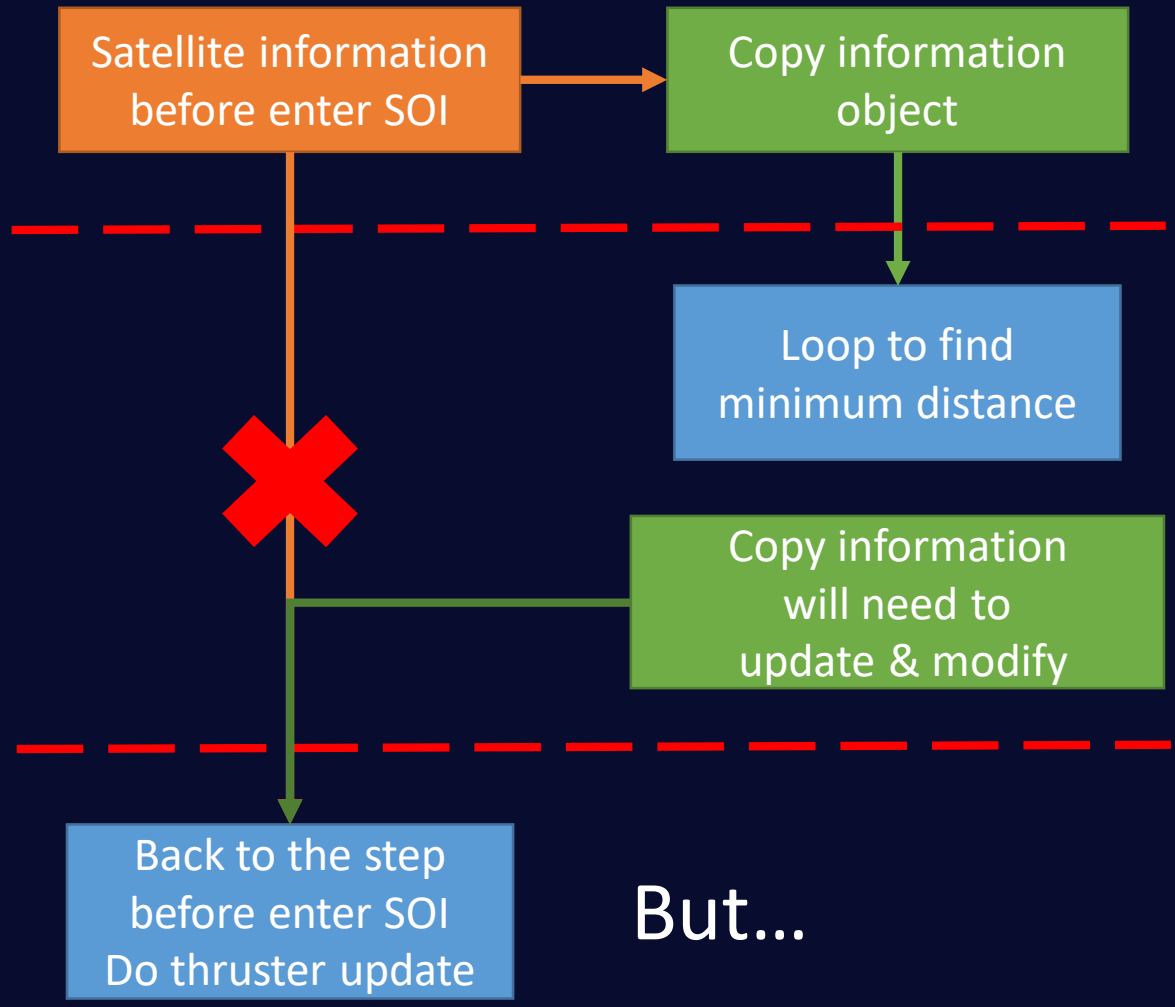
Problem : copy() in Python



But…



Problem : copy() in Python

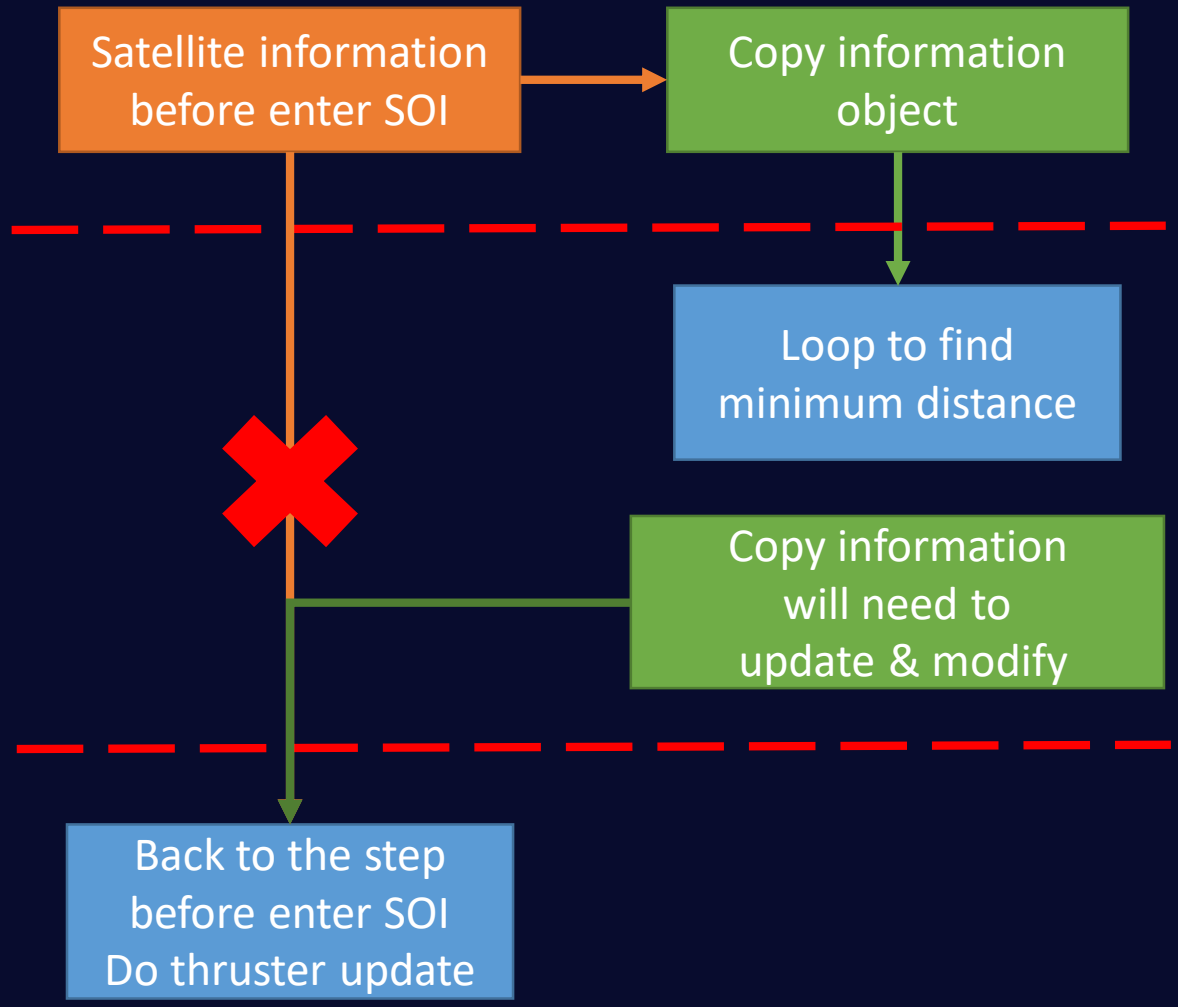


But...

```
x: -622771911.0870203 y: 416138042.42649204 z: 180615442.51379782
find the direction:
time: 2444111.0 x: -623471372.2138661 y: 415888130.4259438 z: 180511336.62062678
time: 2444112.0 x: -624169485.5310107 y: 415637657.61255324 z: 180407102.1324012
time: 2444113.0 x: -624866251.7632319 y: 415386632.8318817 z: 180302747.0527599
time: 2444114.0 x: -625561671.6445698 y: 415135065.12856567 z: 180198279.6698904
```

```
time: 2444210.0 x: -686773621.5519373 y: 388509732.0850351 z: 175801763.68224213
time: 2444211.0 x: -687371288.8545544 y: 388106725.4316633 z: 175893467.3111652
time: 2444212.0 x: -687968006.4191173 y: 387699307.9846047 z: 175984946.90626466
time: 2444213.0 x: -688563747.769249 y: 387287670.8538812 z: 176076100.26585796
time: 2444214.0 x: -689158488.6781838 y: 386871996.61781687 z: 176166840.29239056
time: 2444215.0 x: -689752207.0116308 y: 386452459.0000437 z: 176257093.13340542
direction is find!
back to : 2444110.5 with no speed
x: -690344739.6401566 y: 386030743.2745501 z: 176346930.5982988
Done for simulation!
```

Problem : copy() in Python



	python	Numpy	Tensorflow
b = a	deep copy	deep copy	shallow copy
b = a[:]	shallow copy	deep copy	shallow copy
b = a.copy()	shallow copy	shallow copy	X
b = copy.copy(a)	shallow copy	shallow copy	shallow copy
b = copy.deepcopy(a)	deep copy	deep copy	X
b = tf.identity(a)	X	X	shallow copy

deep copy will **use the address of the variable**

So if we modify the copy one,
modification will act on save address

⬇

will also modify the original one

Idea implement(6) : thruster demo

```
time: 2444194.0 x: -677112585.1588489 y: 394198990.7772657 z: 174476938.62255636
time: 2444195.0 x: -677720056.3957083 y: 393890756.87884665 z: 174538834.75926706
time: 2444196.0 x: -678327238.5998429 y: 393576126.805971 z: 174605447.32445163
time: 2444197.0 x: -678934099.8019776 y: 393254955.5530706 z: 174676327.32943112
time: 2444198.0 x: -679540601.6323012 y: 392927162.37275875 z: 174751017.5425642
time: 2444199.0 x: -680146700.7939484 y: 392592726.8913396 z: 174829063.80248457
time: 2444200.0 x: -680752350.5351741 y: 392251683.2757299 z: 174910024.66204777
time: 2444201.0 x: -681357502.0324658 y: 391904113.127306 z: 174993479.06117046
time: 2444202.0 x: -681962105.6221398 y: 391550137.75010103 z: 175079031.95526236
time: 2444203.0 x: -682566111.8439447 y: 391189910.34813744 z: 175166318.0070777
time: 2444204.0 x: -683169472.2831584 y: 390823608.5799337 z: 175255003.57197046
time: 2444205.0 x: -683772140.2155529 y: 390451427.7639656 z: 175344787.27150825
time: 2444206.0 x: -684374071.0718333 y: 390073574.9060365 z: 175435399.4683076
time: 2444207.0 x: -684975222.7451872 y: 389690263.6187343 z: 175526600.93934396
time: 2444208.0 x: -685575555.7684168 y: 389301709.9280775 z: 175618181.00913325
time: 2444209.0 x: -686175033.3869493 y: 388908128.9119333 z: 175709955.35902524
time: 2444210.0 x: -686773621.5519373 y: 388509732.0850351 z: 175801763.68224213
time: 2444211.0 x: -687371288.8545544 y: 388106725.4316633 z: 175893467.3111652
time: 2444212.0 x: -687968006.4191173 y: 387699307.9846047 z: 175984946.90626466
time: 2444213.0 x: -688563747.769249 y: 387287670.8538812 z: 176076100.26585796
time: 2444214.0 x: -689158488.6781838 y: 386871996.61781687 z: 176166840.29239056
time: 2444215.0 x: -689752207.0116308 y: 386452459.0000437 z: 176257093.13340542
direction is find!
back to : 2444110.5 with no speed
x: -623471372.2138661 y: 415888130.4259438 z: 180511336.62062678
back to : 2444110.5 with speed 1 day
x: -623471370.298339 y: 415888164.7842788 z: 180511338.71796173
Done for simulation!
```

Reference

- https://docs.astropy.org/en/stable/time/index.html
- https://pypi.org/project/jplephem/
- https://voyager.jpl.nasa.gov/mission/did-you-know/
- https://zh.wikipedia.org/wiki/%E5%8F%8D%E6%8E%A8%E5%8A%9B%E7%B3%BB%E7%BB%9F
- https://en.wikipedia.org/wiki/Sphere_of_influence_(astrodynamics)
- https://medium.com/@johnnyliao/python%E9%81%BF%E9%96%8Bdeep-copy%E7%9A%84%E9%99%B7%E9%98%B1-%E5%AF%A6%E9%9A%9B%E5%9C%A8numpy-tensorflow%E7%9A%84%E6%87%89%E7%94%A8-bebbdd247535
- https://www.facebook.com/519265398171665/posts/1212531952178336/
- https://ithelp.ithome.com.tw/articles/10221255

Thank you for listening