Orbit.M

Orbit Maintenance & Propulsion Sizing Tool

In the Spirit of the Open Source Cube-Sat Workshop (OSCW), 12 – 13 December 2020

Why should orbit maintenance analysis be fast?

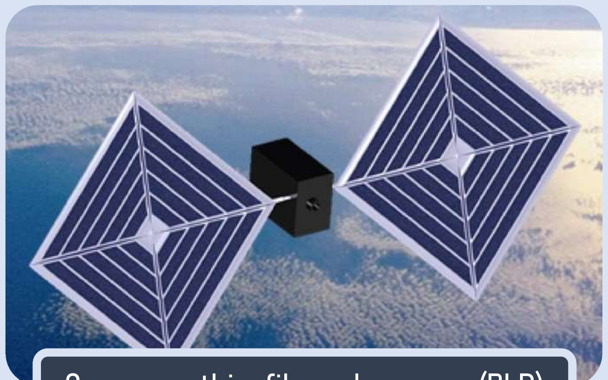
How can orbit maintenance analysis be faster?

What does OrbitM offer to your analysis?

Why should orbit maintenance analysis be fast?

Agile development ➔ agile requirements; requiring frequent mission life comparisons between multiple structures.

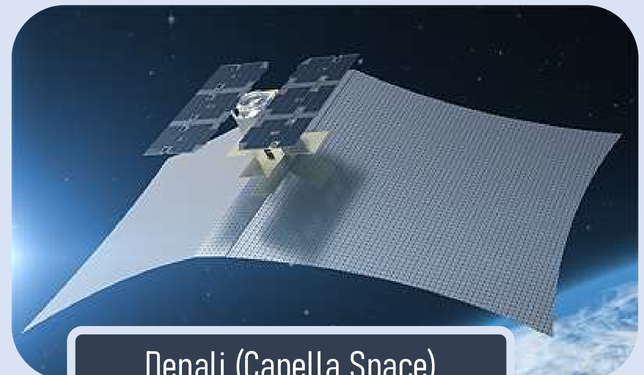
Drag Effects on Different Solar Array Types


Gossamer thin-film solar array (DLR)


S2TEP Small Sat Bus (DLR)

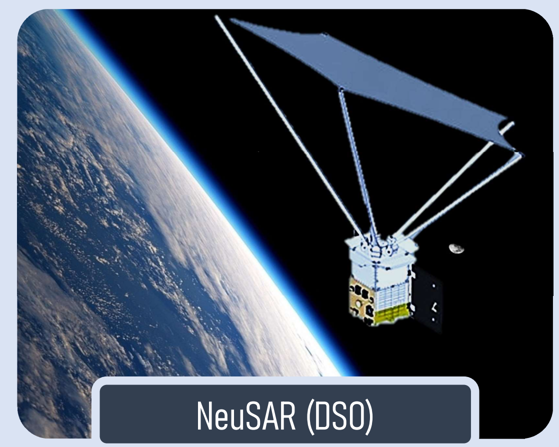

Traditional Mechanical Solar Array Deployer (SpaceX)

Drag Effects on Different Antenna Reflectors


Denali (Capella Space)

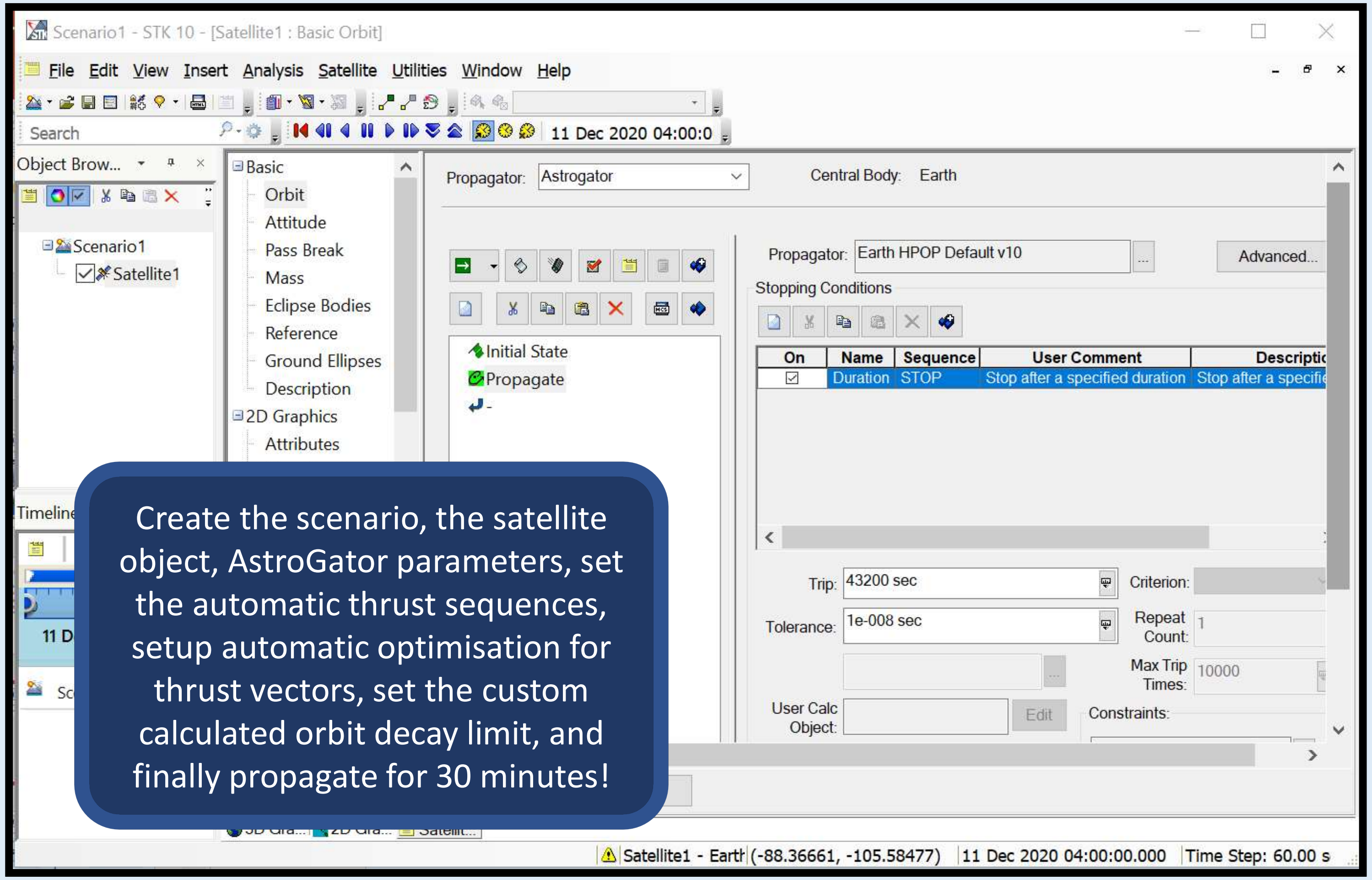

Sequioa (Capella Space)

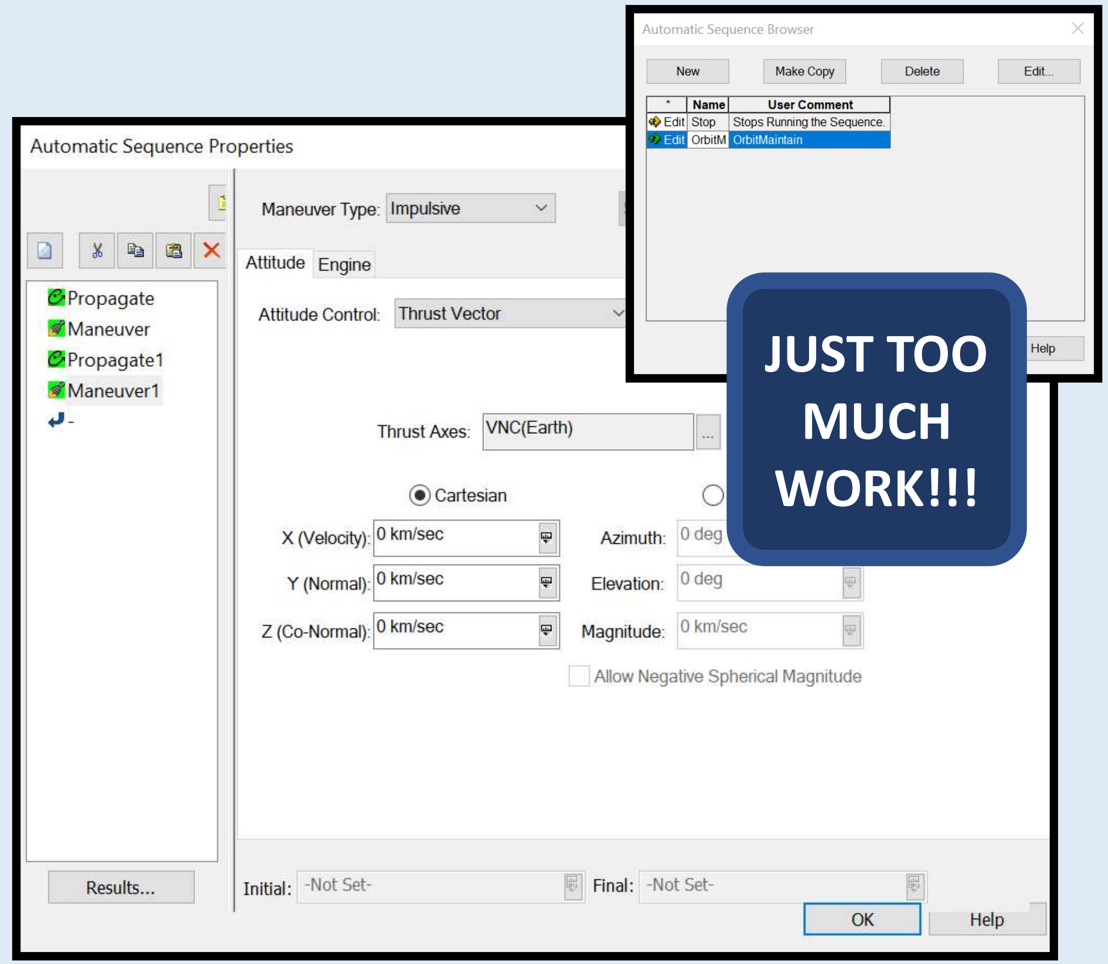

TeLEOS-2 (DSO)


NeuSAR (DSO)

How can orbit maintenance analysis be faster?

Commercial astrodynamics software like STK and GMAT can compute with high precision the orbit state vectors at any point in time, but that may be too much work for a simplified orbit lifetime analysis… (and not everyone has access to STK!)





How can orbit maintenance analysis be faster?

Key Idea: You only need the altitude to compute the decay rate, no need for all 6 state vectors with high precision orbit propagation, and definitely no need for such a complicated series of steps!

Starting from some high school physics… let's see how we can characterise energy loss in orbit…

Total Orbital Energy

$$U = KE + GPE = \left(-\frac{GM_E m}{2R}\right)$$

Let's differentiate with R

Radial Derivative

$$\frac{dU}{dR} = \frac{GM_E m}{2R^2}$$

Let's revisit first principles of "energy and work done"

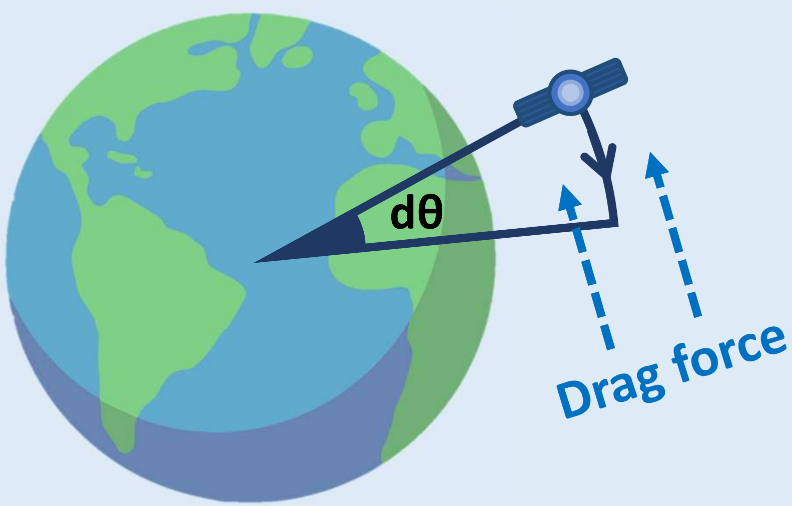
Time Derivative

$$\frac{dU}{dt} = F\frac{ds}{dt} = F \cdot R \cdot \frac{d\theta}{dt} = F \cdot R\sqrt{\frac{GM_E}{a^3}}$$

Drag

Radius

Mean Motion



How can orbit maintenance analysis be faster?

Now, we can actually already derive the orbit decay rate easily!

Atmospheric Density

US Standard Atmosphere 1976 (Look-Up Table)

Radial Derivative

$$\frac{dU}{dR} = \frac{GM_E m}{2R^2}$$

Time Derivative

$$\frac{dU}{dt} = F \cdot R\sqrt{\frac{GM_E}{a^3}}$$

Drag Equation

$$\alpha_{drag} = -\frac{1}{2}\rho C_d \left(\frac{A}{m}\right) V^2$$

$$\frac{dR}{dt} = \left(\left(\frac{dU}{dR}\right)^{-1} \cdot \frac{dU}{dt}\right) = 2\alpha_{drag}\sqrt{\frac{a^3}{GM_E}} = \frac{\alpha_{drag} T}{\pi}$$

$T = 2\pi\sqrt{\frac{a^3}{GM_E}}$ ➔ Kepler's Third Law (Circular Orbits)

How can orbit maintenance analysis be faster?

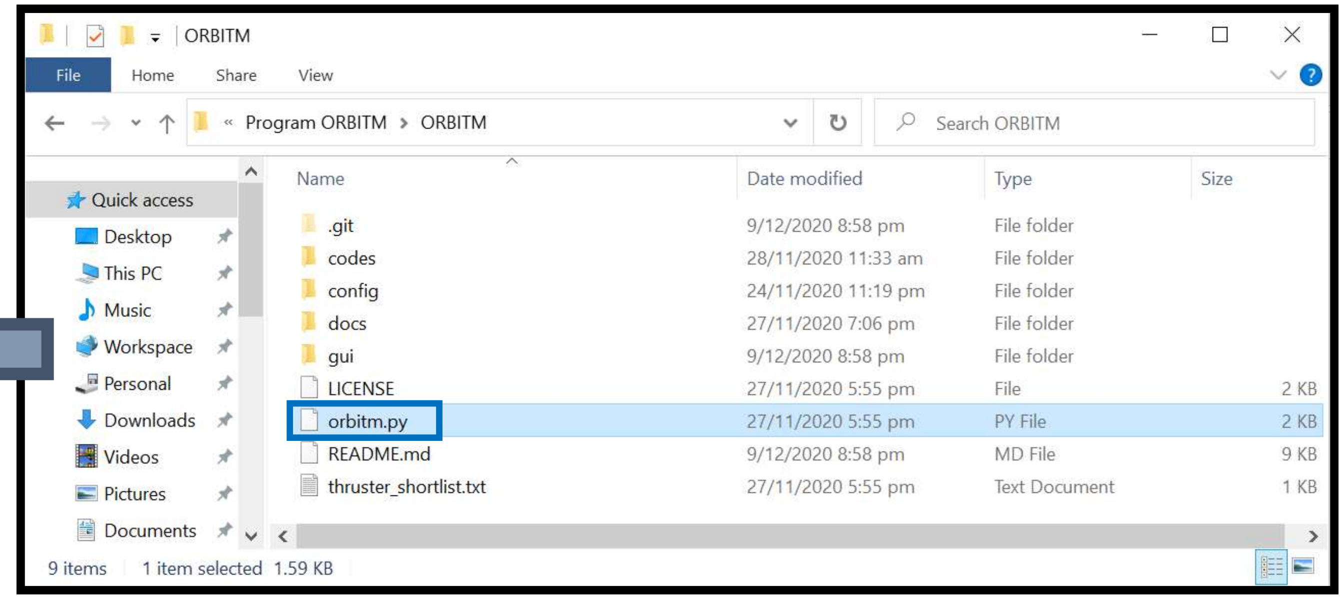
Now, we can actually already derive the orbit decay rate easily!

$$\frac{dR}{dt} = \frac{\alpha_{drag} T}{\pi}$$

In other words, the rate at which the orbit decays can be solved in closed-form without any full orbit propagation of orbit states!

The decay rate = product of the <u>drag deceleration</u> and the <u>Keplerian period</u>, divided by <u>Pi</u>!

What does OrbitM offer to your analysis?

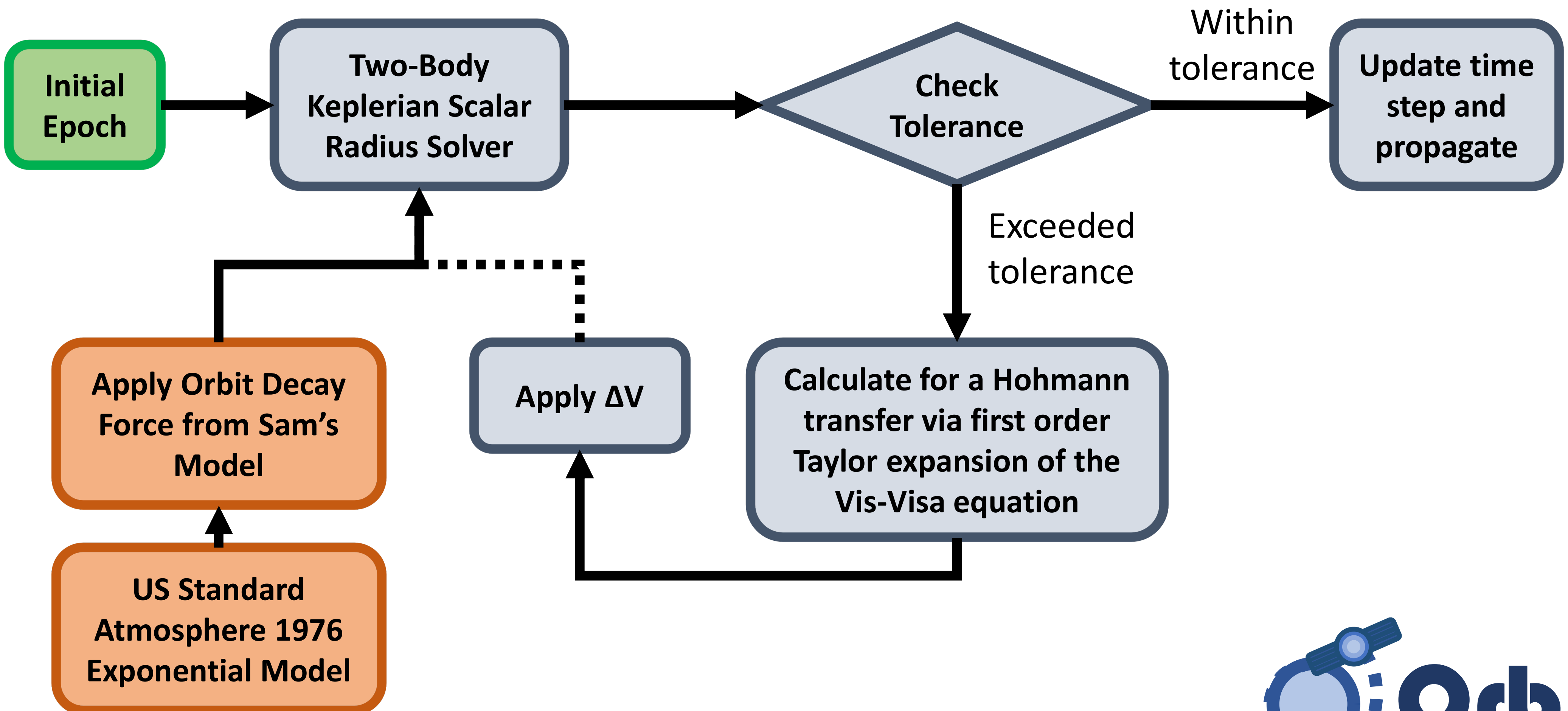


- **Choose your orbit simulator.**
- **Specify your mission duration.**
- **Specify your spacecraft and orbit parameters.**
- **Specify your mission tolerances and margins.**

Outputs a propulsion sizing chart and a ΔV report.

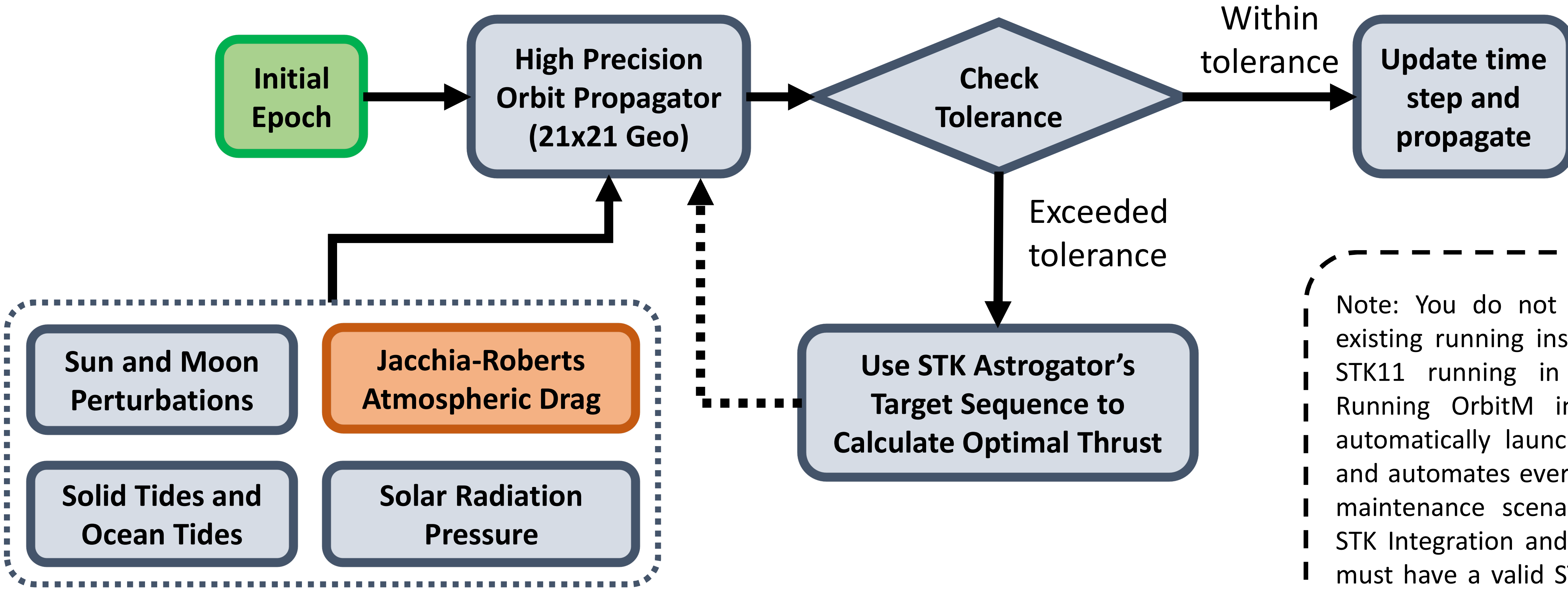
What does OrbitM offer to your analysis?

Sam's (Fast) Computation



What does OrbitM offer to your analysis?

STK's (Precise) Computation



Note: You do not need to have an existing running instance of STK10 or STK11 running in the background. Running OrbitM in the STK mode automatically launches STK, the GUI, and automates every step of the orbit maintenance scenario for you using STK Integration and STK Connect. You must have a valid STK Integration and Astrogator license!

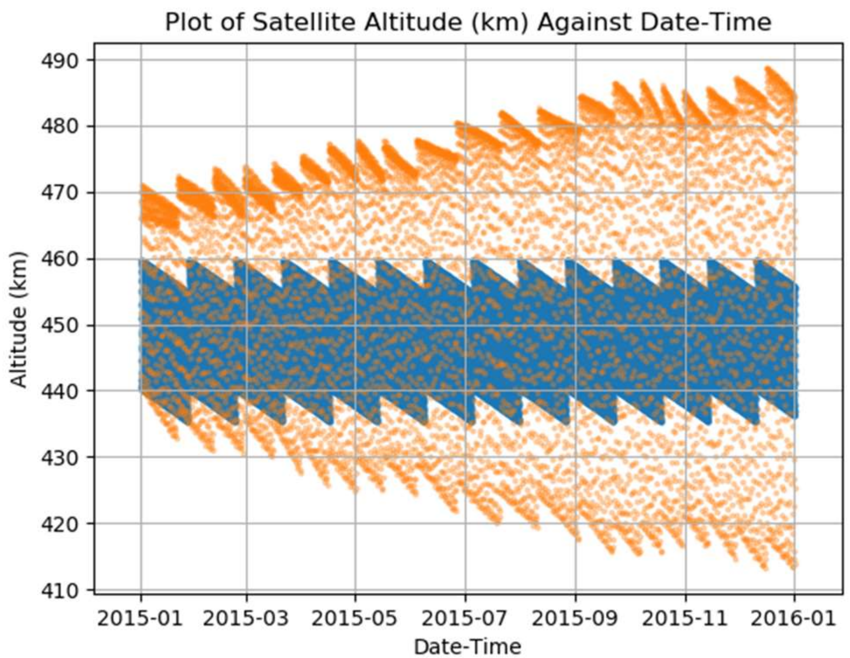
What does OrbitM offer to your analysis?

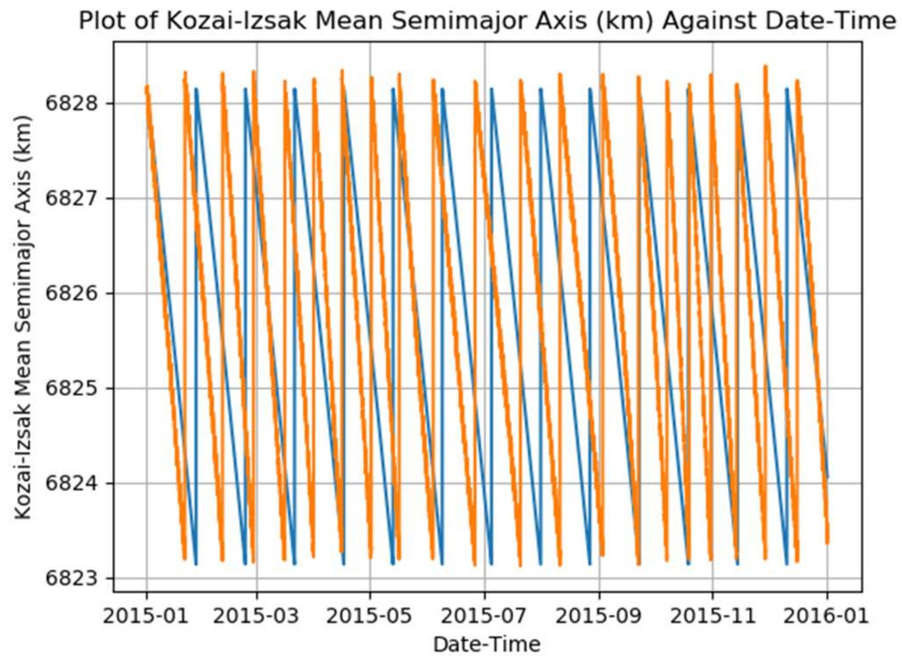
Orbit Maintenance Scheduling Report

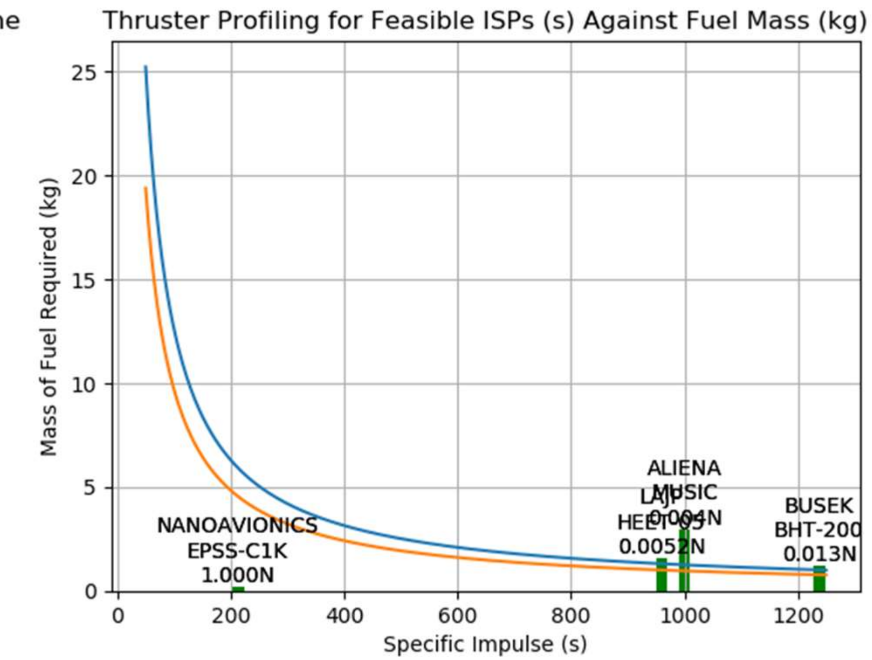
```
1 Maintain.ThrustApo 8 Jan 2012 03:45:24.438158324 0.13666080028002783
2 Maintain.ThrustPeri 8 Jan 2012 04:33:35.359392809 0.13666080028002783
3 Maintain.ThrustApo 16 Jan 2012 23:46:19.843702020 0.13666080028002783
4 Maintain.ThrustPeri 17 Jan 2012 00:33:44.859483133 0.13666080028002783

... ... ...

293 Maintain.ThrustApo 22 Dec 2014 05:13:52.359890580 0.13666080028002783
294 Maintain.ThrustPeri 22 Dec 2014 06:01:07.847235188 0.13666080028002783
295 Maintain.ThrustApo 30 Dec 2014 14:04:11.517531291 0.13666080028002783
296 Maintain.ThrustPeri 30 Dec 2014 14:52:41.602837637 0.13666080028002783
```

Number | Thrust Location | Time of Impulsive Thrust | Delta-V

What does OrbitM offer to your analysis?







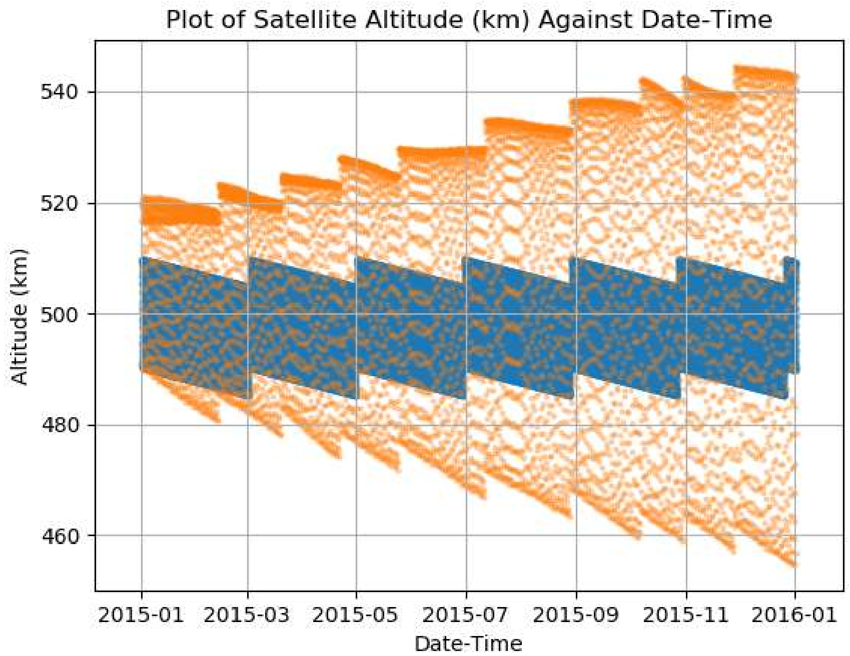
Circular Orbit @ 450km Mean Altitude

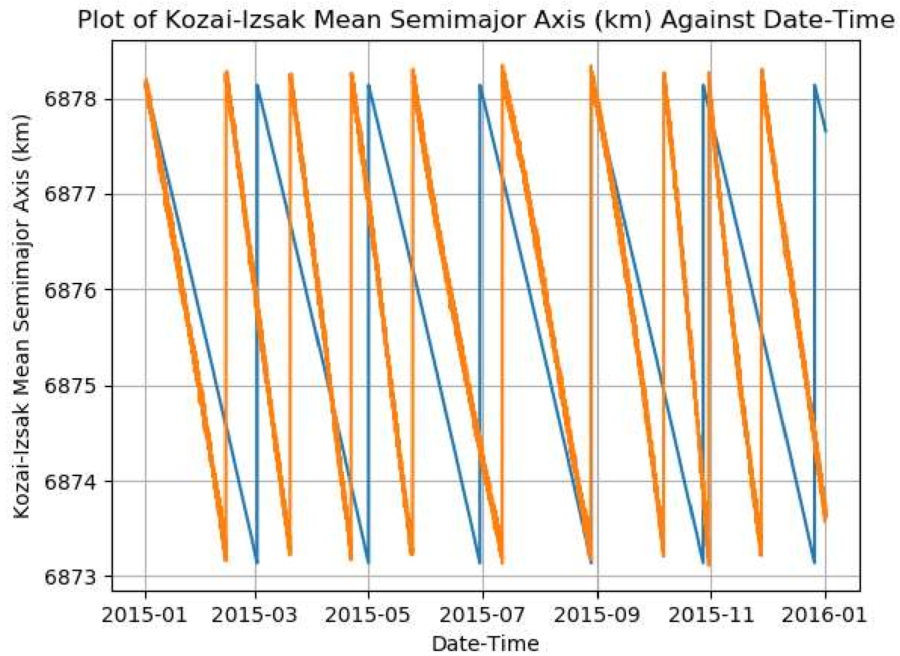
Cd = 2.3; Mass = 170kg; Area ~ 2.5m^2

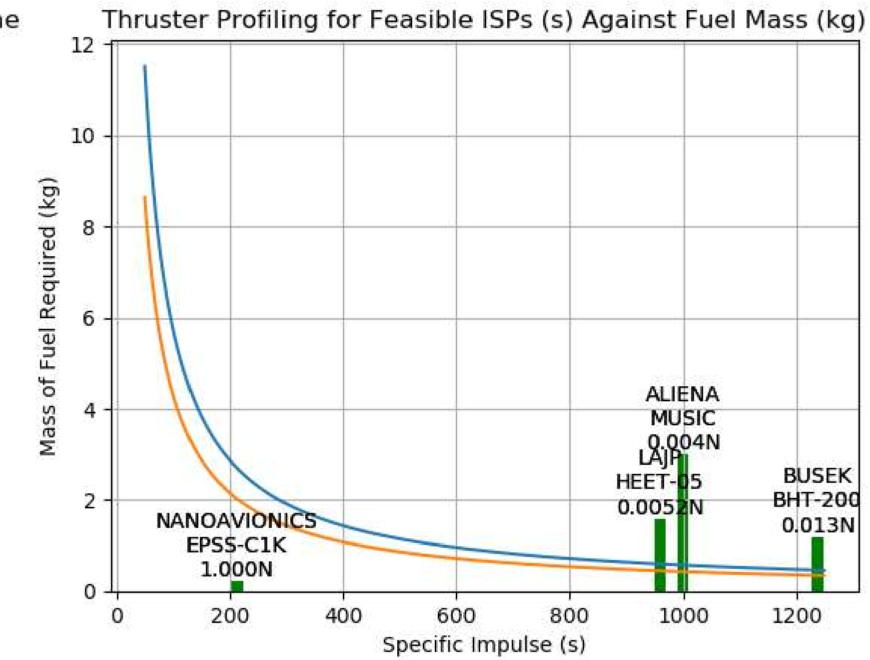
Sam's (Fast) Solver

STK10 AstroGator (HPOP)

What does OrbitM offer to your analysis?







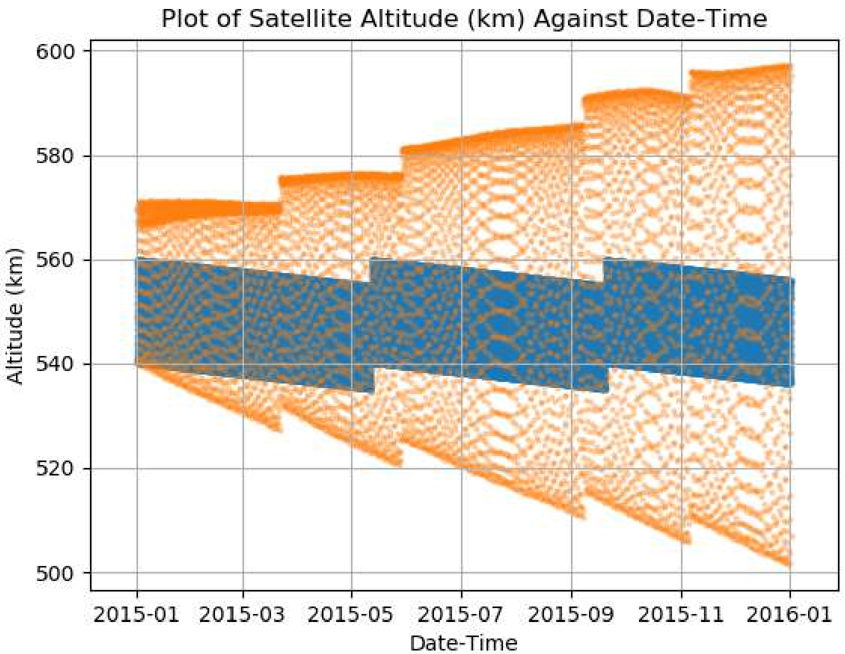
Circular Orbit @ 500km Mean Altitude

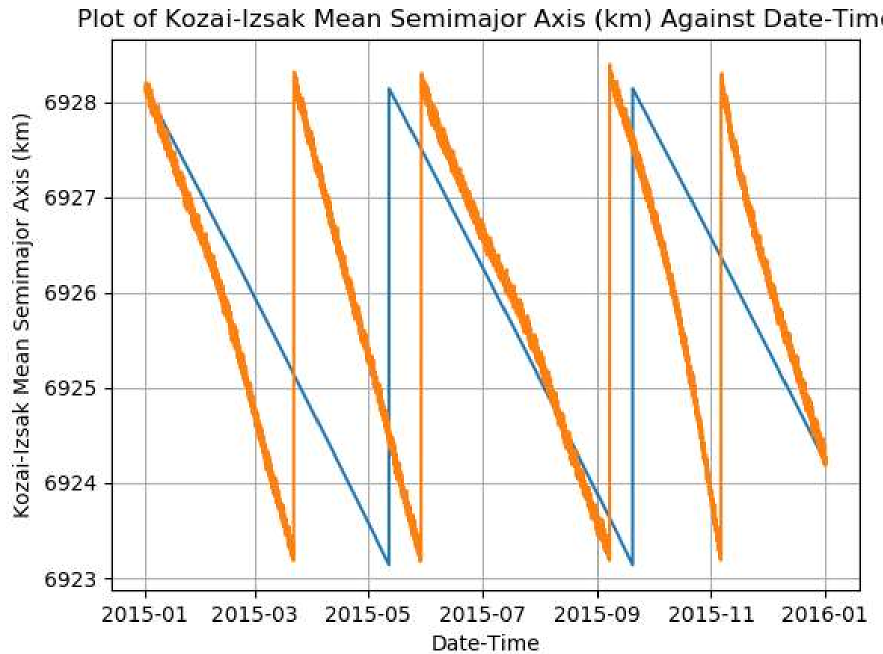
Cd = 2.3; Mass = 170kg; Area ~ 2.5m^2

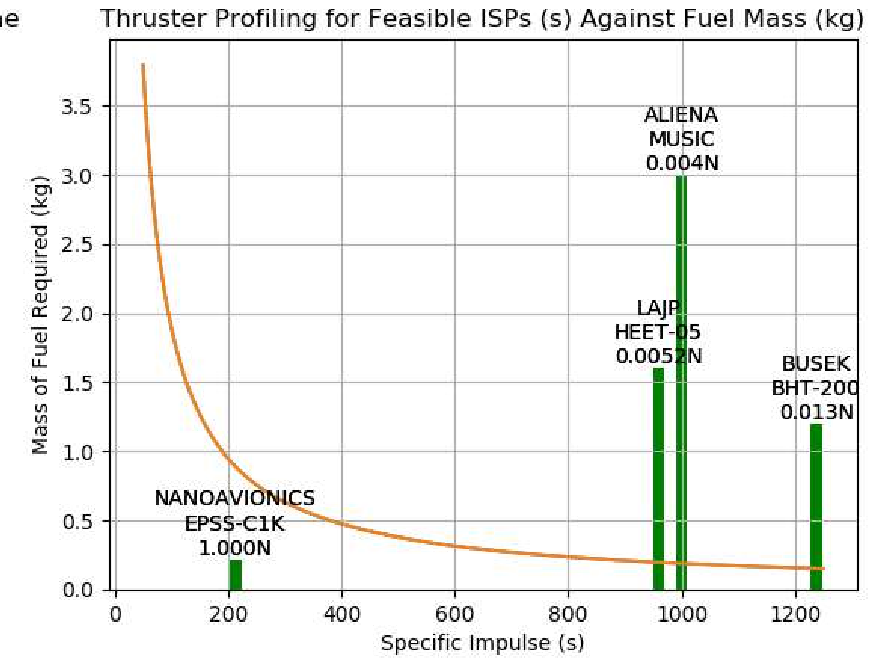
Sam's (Fast) Solver

STK10 AstroGator (HPOP)

What does OrbitM offer to your analysis?







Circular Orbit @ 550km Mean Altitude

Cd = 2.3; Mass = 170kg; Area ~ 2.5m^2

Sam's (Fast) Solver

STK10 AstroGator (HPOP)

Summary and Conclusion

Orbit.M is a Python-based orbit maintenance simulator, which helps you size your mission lifetime quickly!

Orbit.M can also help you determine if your short-listed propulsion units are suitable for your mission.

Orbit.M is most useful, if your satellite design has many physical iterations with changing area-to-mass.

Orbit.M is also looking for collaborators versed in GMAT (since STK is not free!)

https://github.com/sammmlow/ORBITM.git

THANK YOU!

[1] Low, S. Y. W., &; Chia, Y. X. (2018). “Assessment of Orbit Maintenance Strategies for Small Satellites”, 32nd Annual AIAA/USU Conference on Small Satellites, Logan, Utah, Utah State University, USA.

**** Please cite if you found OrbitM useful!***