

Orbit.M - An Open-Source Software for LEO Orbit Maintenance in Python

Sunday, 13 December 2020 14:50 (20 minutes)

Indico rendering error

Could not include image: [404] Error fetching image

What is Orbit.M?

Orbit maintenance analysis tends to be rather rigorous with commercial software like STK, or tedious if you have to develop your own simulator in code. Orbit.M offers to the OSCW audience an open-source, graphical, easy-to-use, and free orbit maintenance simulator and propulsion sizing tool, meant for circular orbits, for anyone and anywhere.

It is a software with a GUI where you can fill in your orbital parameters, satellite masses, satellite areas, drag parameters, and the software will compute your desired Delta-V necessary for the mission, sizing it against your choices of thrusters (based on the Isp).

How do I use Orbit.M specifically?

First, check that you have Python 3 installed, and Python standard libraries: **TKinter**, **NumPy**, **Matplotlib**, and among others (**os**, **datetime**, **comtypes** etc). If you do not have Python, I recommend using the Anaconda installer, and running it on Spyder 4 simply because that's what I'm using right now. At the time of the latest commit and push, NumPy version is 1.18, and Matplotlib is 3.1.3. Older versions should work fine as ORBIT.M doesn't use any of the newer features.

The software can be started by running the Python file **orbitm.py** in the main directory (which is equivalent to the directory you see on the master branch on ORBITM's github page). You should see a GUI that looks like the one below, pop up:

Indico rendering error

Could not include image: [404] Error fetching image

You can fill in your spacecraft and orbit parameters through the GUI.

Next, before you run ORBITM, you should open up the "thruster_shortlist.txt" file on the main ORBITM directory, and fill in any thrusters you wish to size your missions against. Some thrusters I had previously been aware of are written inside this shortlist as an example. The purpose of this shortlist is to graphically compare its Isp and fuel capacity to the suitability of your mission later on.

Do note that the option to use STK10 and STK11 requires a valid legal license for STK Astrogator and STK Integration.

Now, you can run ORBITM.

What can I expect as the output of Orbit.M?

Orbit.M would output three plots - one of altitude decay, one of the mean semi-major axis decay, and finally, your Delta-V sizing profile. The program will also output a time-schedule for orbit maintenance in the main directory of ORBITM, as a text file "deltaV.txt". For the circular orbit @ 500km, with a tolerance band of 5km, an example plot is shown below (blue plot being Sam's simulator, and orange plot generated using STK10):

Indico rendering error

Could not include image: [404] Error fetching image

What else is expected from Orbit.M?

The goal of sharing Orbit.M for the OSCW 2020 is to give satellite and space enthusiasts some simple tools to begin orbit maintenance calculations - especially for aspiring students who cannot afford the time-effort nor money for expensive space simulator software. The accuracies of my model are cross-checked with that of STK's as you can see in the chart above.

In the future, I hope to welcome users experienced in NASA's GMAT program to automate orbit maintenance routines using Python as a 4th simulator option in Orbit.M. If you like open source, and you enjoy orbital mechanics, do consider adding your contributions here for other GMAT users too!

Looking forward to seeing everyone in OSCW 2020!

Primary author: LOW, Samuel

Session Classification: Room #1