EMTG Tutorial: Journey Boundaries

Tim Sullivan *

August 8, 2023

Revision Date	Author	Description of Change
December 2, 2022	Tim Sullivan	Initial revision.
June 30, 2023	Joseph Hauerstein	Conversion to LaTeX.
August 4, 2023	Joseph Hauerstein	Addition of Known Issues section.

*Aerospace Engineer, The Aerospace Corporation

Contents

List of Known Issues

List of Acronyms

EMTG Evolutionary Mission Trajectory Generator

SOI sphere of influence

ICRF International Celestial Reference Frame

SPICE Spacecraft Planet Instrument Camera-matrix Events

NLP Nonlinear Program

MBH Monotonic Basin Hopping

MGAnDSMs Multiple Gravity Assist with n Deep-Space Maneuvers using shooting

GMAT General Mission Analysis Toolkit

DSM deep-space maneuver

1 Introduction

This tutorial assumes you have already completed the OSIRIS-REx and LowSIRIS-REx tutorials. While this tutorial doesn't make use of the Evolutionary Mission Trajectory Generator (EMTG) files created in those tutorials, EMTG options and features covered in past tutorials may not be covered in follow-on tutorials.

As you saw in the OSIRIS-REx tutorials, a typical EMTG workflow is to begin with a low-fidelity solution and use that solution as the initial guess for higher fidelity models. This tutorial will walk you through creating another EMTG options file for a mission launching from Earth, flying past Venus, and arriving at Mars using a patched conics solution. This is called the EVM (Earth-Venus-Mars) scenario. Then you will begin the process of making the mission increasingly realistic by defining specific departure and arrival state parameters. Follow-on tutorials will increase the realism of this and the OSIRIS-REx examples by adding additional forces and more accurate flybys.

1.1 Patched Conics

The method of patched conics approximates interplanetary trajectories in a way that is useful for generating low-fidelity initial guesses for a mission. Patched conics takes a series of conic sections and connects them together between destinations in the solar system. Each conic section is a segment of an orbit around the central body, the Sun in this case.

For example, in the EVM scenario, EMTG attempts to find an ellipse with the Sun at one foci which connects Earth at the launch epoch to Venus at the flyby epoch and another ellipse connecting Venus to Mars at the arrival epoch. These are segments of different orbits around the Sun. Figure 1 shows an example EMTG solution from this tutorial. Shown in a red, dashed ellipse is the solar orbit connecting Venus and Mars which forms part of the patched conic solution.

Notice that there is a deep-space maneuver (DSM) between Earth and Venus. The trajectory from Earth to Venus is not a purely ballistic patched conic solution! Instead, there is a conic section defining the trajectory from Earth to the DSM and another from the DSM to Venus.

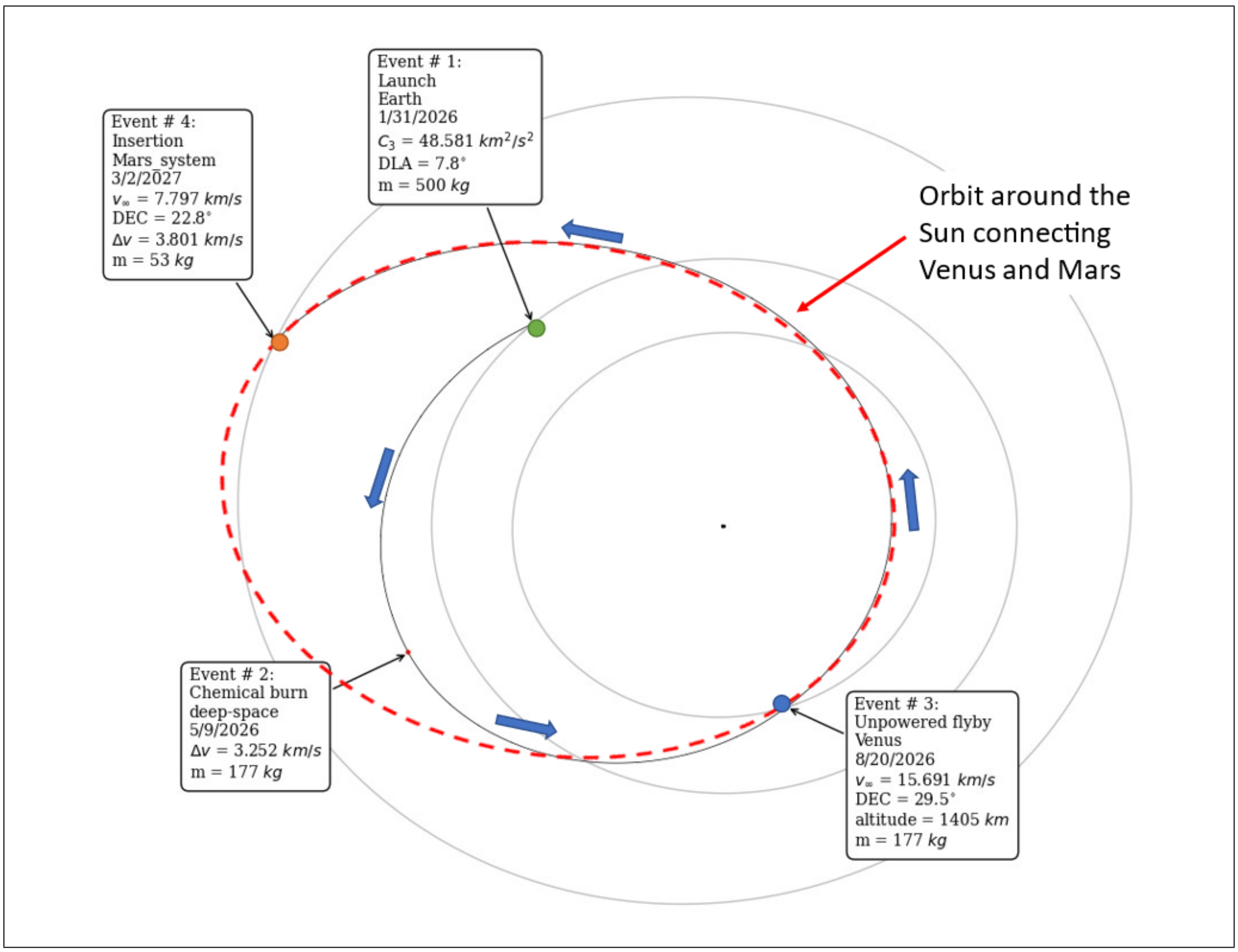

Figure 1: EVM MGAnDSM Patch Conic Example.

Patched conics are a good starting approximation because the gravitational forces exerted on the spacecraft are primarily due to the Sun and can be initially modeled by two-body forces. However, this is not accurate enough for realistic mission analysis. When the spacecraft is close to a planet, it is predominantly affected by the gravity of that planet rather than the Sun. The point where the transition between being predominantly affected by the planet's gravity to predominantly affected by the Sun's is called the sphere of influence (SOI). The approximate radius of the SOI for a body is given by the equation below, where m is the mass of the smaller body, M is the mass of the larger body, and a is the semi-major axis of the smaller body's orbit around the larger body.

$$r_{SOI} \approx a(\frac{m}{M})^{2/5}$$

Additionally, as the spacecraft travels from one body to another, the gravitational forces of multiple bodies influence it. For example, a trajectory from Earth to Venus is affected by Jupiter's gravity. You will add these perturbing gravitational forces and others in the non-two-body force model tutorial.

Patched conics are also not a good high-fidelity solution because the conic sections begin and end at the center of mass of each body. For example, the Earth-to-Venus trajectory starts at Earth's center of mass and ends at Venus's center of mass. Obviously, you do not want the spacecraft to crash into a planet! These and other factors mean that the patched conic solution must be updated in order to fly the mission.

Up to this point, the EMTG missions have used a patched conics approach to specify trajectory leg initial and final points. In this tutorial, the initial step will begin by making the missions more realistic by specifying trajectory leg end points that do not occur at the center of mass of an ephemeris object (i.e., a planet). The next tutorials will continue towards higher fidelity solutions by adding additional perturbing forces and more realistic body flybys.

2 Setup

Let's create the patched conic Earth-Venus-Mars trajectory. This will be the basis for finding an initial solution and then migrating to higher fidelity.

2.1 Global Options

Begin by creating the mission and Universe directory structure discussed in the introductory tutorial. For the EVM series of tutorials, you will only need the default ephemeris files `DE430.bsp`, `naif0012.tls`, and `pck00010.tpc`. Use the EMTG universe file `Sun.emtg_universe`, which is available from the tutorial documents in `docs/0_Users/tutorial/Tutorial_EMTG_Files/EVM_universe`. Place these files into an `EVM_universe` directory using the structure shown in Figure 2. Similarly, the contents of `hardware_models` are in Tutorial_EMTG_Files under `Journey_Boundaries/hardware_models`.

```
+---Journey_Boundaries
    |   EVM.emtgopt
    |
    +---hardware_models
            default.emtg_launchvehicleopt
            default.emtg_powersystemsopt
            default.emtg_propulsionsystemopt
            default.emtg_spacecraftopt
            empty.ThrottleTable

+---EVM_universe
    |   Sun.emtg_universe
    |
    \---ephemeris_files
             de430.bsp
             naif0012.tls
```

Figure 2: Mission Directory Structure.

Start PyEMTG and create a new mission (File->New->Mission or Ctrl+m). Set the following global options as shown in Figure 3:

- **Mission name:** "EVM"
  - Make sure to save the options file.
- **Mission type:** "MGAnDSMs"
- **Objective function:** "0 - minimum deterministic deltaV"
- **Launch window open date:** "1 January 2026" or "61041.0"
- **Enable mission time bounds:** "On"
- **Global flight time bounds:** "0.0" to "395.25" days
- **Forced post-launch coast duration:** "15" days
- **Forced pre-flyby coast duration:** "30" days
- **forced post-flyby coast duration:** "15" days

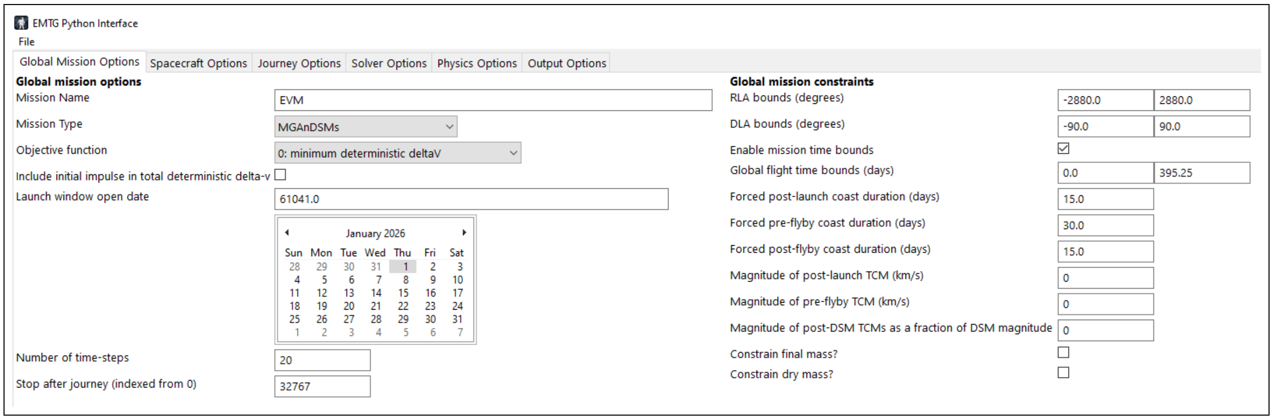

Figure 3: EVM Global Options.

2.2 Spacecraft Options

Select the "Spacecraft Options" tab and make the following changes as shown in Figure 4:

- **Maximum mass:** "500" kg
- **Spacecraft model type:** "Assemble from missionoptions object"
- **Hardware library path:** path to "`hardware_models`"
  - Do not forget the trailing slash.
- **Launch vehicle library file:** "`default.emtg_launchvehicleopt`"
- **Launch vehicle:** "ExampleRocket"
- **Chemical Isp:** "320" seconds
  - Chemical Isp is used for DSMs and assumes a bipropellant thruster.
- **TCM Isp:** "200" seconds
  - TCM Isp is used for pre/post event TCMs and Journey-end delta-v and TCMs. Assumed to be a monopropellant thruster.

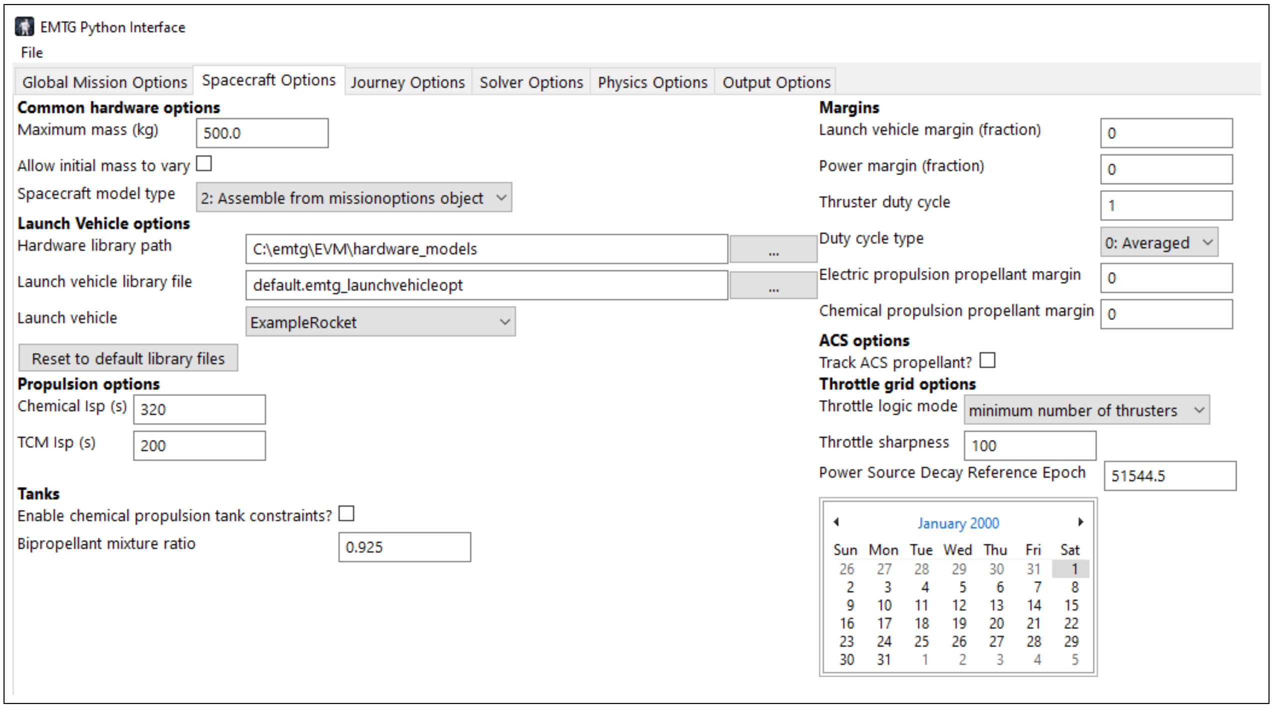

Figure 4: EVM Spacecraft Options.

2.3 Journey Options

Select the "Journey Options" tab. You will create one Journey beginning at Earth, flying past Venus, and ending at Mars. Make the following changes to the default Journey as shown in Figure 5:

- **Central body:** "`Sun.emtg_universe`"
  - If `Sun.emtg_universe` does not appear as an option for the central body, the most likely cause is that the default universe folder was not set to the folder that contains `Sun.emtg_universe`. To override the default, select the "Physics Options" tab, and change the value of the "Universe folder" to the folder that contains `Sun.emtg_universe`. Then, return to the "Journey Options" tab.
- **Start location:** "3" for Earth
- **Final location:** "4" for Mars_system
- **Wait time bounds:** "0" to "30" days
  - Set the launch window open date in the "Global Options" tab to 1 January 2026. Setting 30 days as the upper bound here means that EMTG's solutions must depart Earth by 31 January 2026. If that sounds overly constrained, it is. You will find a better time to launch in a later tutorial.

- **Journey time bounds:** “bounded flight time”
- **Journey flight time bounds:** “0” to “395.25”
  - This means the solution has up to 13 months to arrive at Mars.
- **Journey inital impulse boudns:** “0” to “6.97” km/s
- **Journey departure type:** “0: launch or direct insertion”
- **Journey departure class:** “0: Ephemeris-pegged”
- **Forced initial coast:** “0” days
- **Journey arrival type:** “0: insertion into parking orbit (use chemical Isp)”
- **Journey arrival class:** “0: Ephemeris-pegged”
- **Insertion orbit SMA:** “20000” km
- **Insertion orbit ECC:** “0.7”
- **Forced terminal coast:** “15” days
- **Flyby sequence:** “[2]”
  - Venus is body #2 in the Universe file.

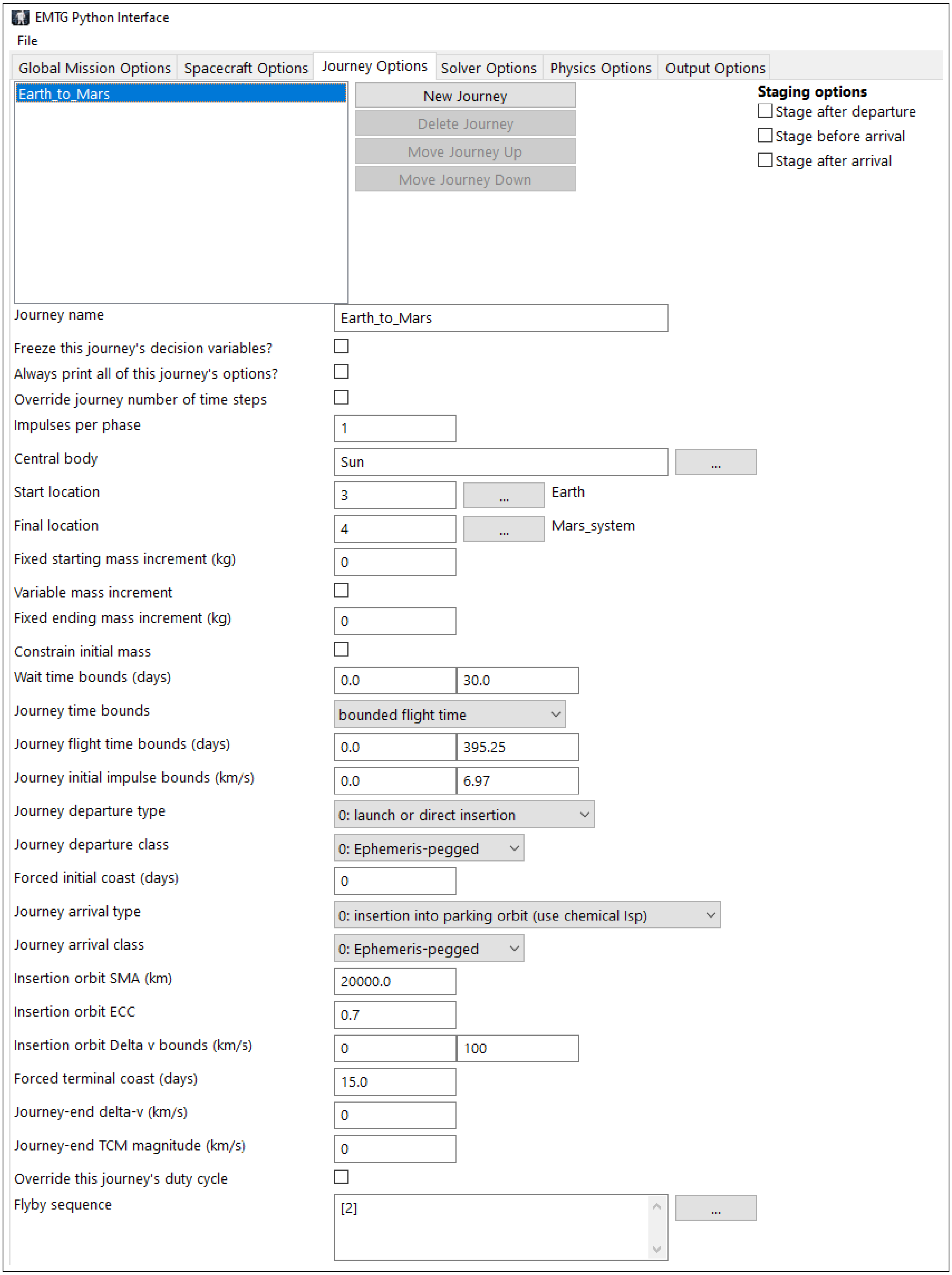

Figure 5: EVM Journey Options.

2.4 Solver Options

Switch to the "Solver Options" tab. Most of these can remain set to their defaults, but it is recommended to change the following settings as shown in Figure 6:

- **NLP Chaperone:** "On"
- **ACE feasible point finder:** "On"
- **Maximum run-time(s):** "120" seconds

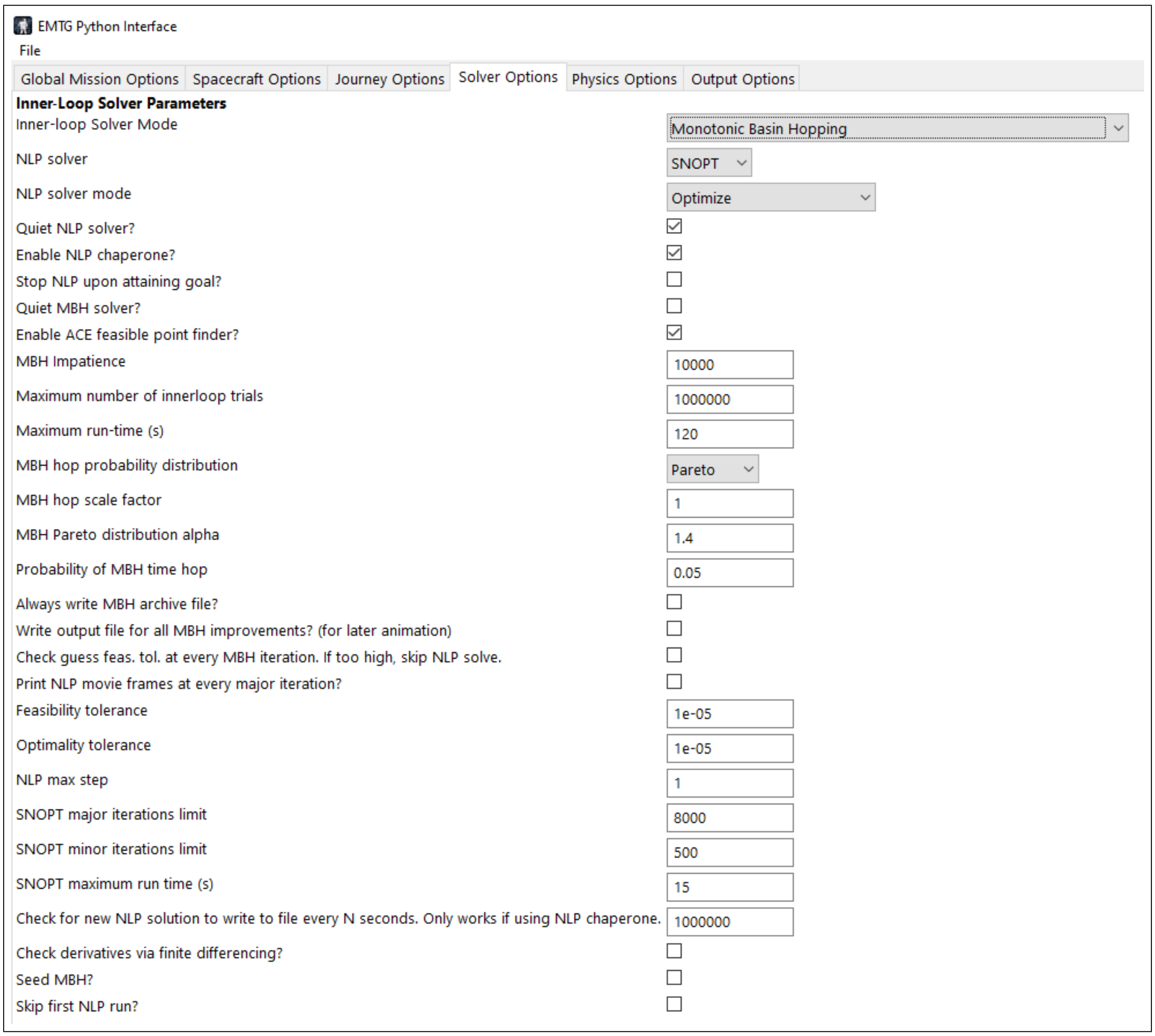

Figure 6: EVM Solver Options.

2.5 Physics Options

Switch to the "Physics Options" tab. Because you're just using the DE430 ephemeris, leave the earliest and latest spline epochs as their default value. Make sure to change the Universe folder path if you haven't already. EMTG will look in the path specified by "Physics Options" for the central bodies set in the Journey tab. If EMTG can't find them, it will throw an error at runtime. Even if you change the path to the central body Universe file in "Journey Options", EMTG will look for a Universe file with that central body name inside the directory specified by "Physics Options", not the one you set in "Journey Options". Make sure to update this path whenever copying missions. The recommended settings are shown in Figure 7, but they should already be set.

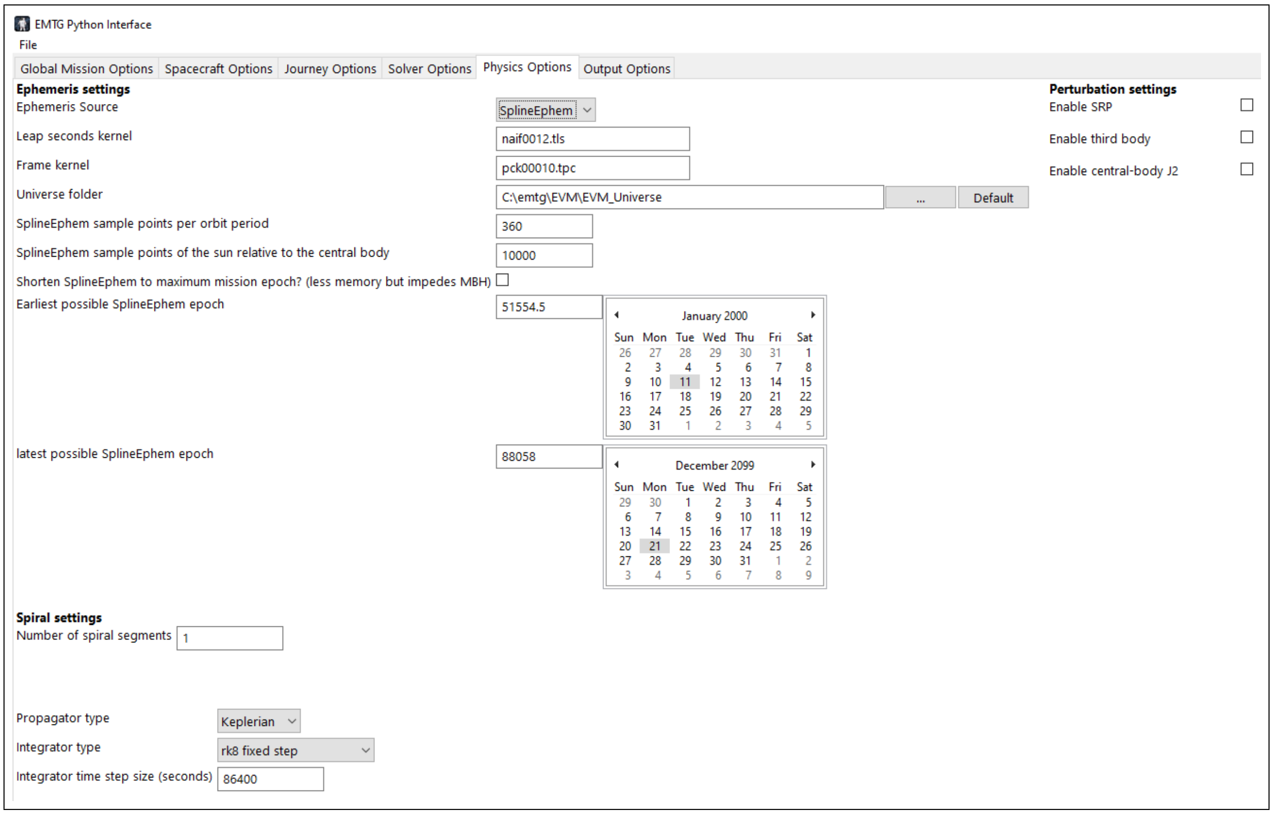

Figure 7: EVM Physics Options.

2.6 Output Options

Following the method used in the OSRIS-REx tutorial, create a `results` folder in the mission directory to hold the EMTG outputs from each run. Switch to the "Output Options" tab and make the following changes as shown in Figure 8:

- **Output file frame:** "ICRF"
- **Override working directory:** "On"

- **Working directory:** path to the `results` directory

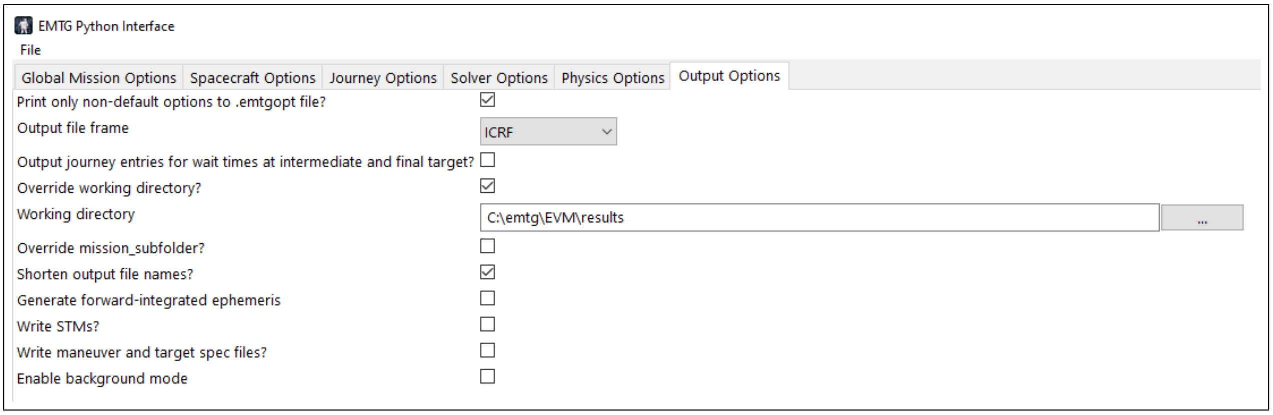

Figure 8: EVM Output Options.

3 Run and Post-Process

Select “File->Run” (Ctrl+r), saving the file as `EVM.emtgopt`. After EMTG finishes running, open the solution `EVM.emtg` file with PyEMTG (“File->Open”, or Ctrl+o) and plot the trajectory. This is the patched conic solution. Note that your trajectory may look different than Figure 9 because EMTG uses stochastic optimization techniques and you only let EMTG run 2 minutes. Next, you’ll modify the mission to have a more realistic arrival at Mars.

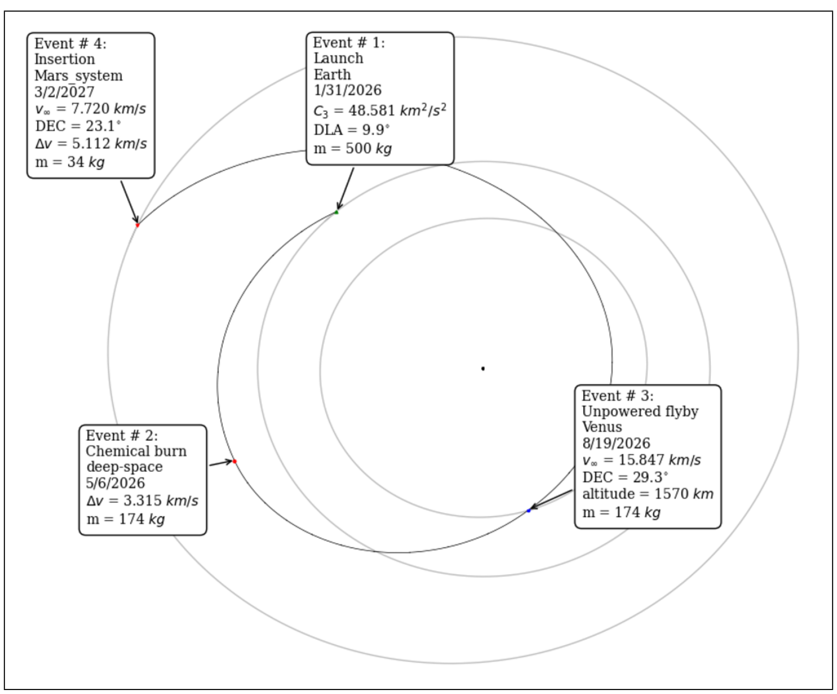

Figure 9: Earth Venus Mars Patched Conic Trajectory.

4 Journey Boundaries

In EMTG, a Boundary State defines the state of the spacecraft at the start (departure) or end (arrival) of a trajectory segment, such as a Phase or a Journey. Examples include the initial departure state from Earth or rendezvous with another body. Boundaries can be defined in several ways, and these definitions can be used to progress a solution from initial low-fidelity models to high-fidelity models.

In general, a user has much more flexibility in defining a *Journey* Boundary than in defining a *Phase* Boundary, and the remainder of this discussion is focused on Journey Boundaries.

4.1 Ephemeris-Pegged Boundary Class

The Boundary Class you've been using so far is "Ephemeris-pegged". This is EMTG's default Boundary Class. Ephemeris-pegged states are tied directly to an ephemeris body (e.g., planet) center of mass and move with that body. The EMTG ephemeris-pegged boundary code references the SPICE ephemeris files to look up the state of the ephemeris body as a function of time. It then uses that state to define the spacecraft state at that time. For example, an "ephemeris-pegged" launch or direct insertion from an Earth Boundary will depart from the center of mass of the Earth with some escape velocity.

4.2 Ephemeris-Referenced Boundary Class

An "Ephemeris-referenced" Boundary is similar to ephemeris-pegged. However, an Ephemeris-referenced Boundary is defined relative to an ephemeris point but not on it. An Ephemeris-referenced Boundary lies on a triaxial ellipsoid centered on an ephemeris point. For example, you might define a spherical ellipsoid corresponding to the sphere of influence of the Earth.

Choosing "Ephemeris-referenced" in PyEMTG will prompt the user to input a list of three floats that define the semi-axes of the ellipsoid (in km) in the ICRF coordinate system. The spacecraft state at the boundary is constrained to lie on the surface of this ellipsoid.

4.3 Free Point Boundary Class

A "Free Point" Boundary occurs at a point in space relative to the central body of a Journey. The point in space may be defined by one of the following state representation: cartesian state, spherical state, classical orbit elements (COE), modified equinoctial elements (MEE), or incoming/outgoing B-plane representations in multiple frames of reference, such as ICRF. Using a Free Point Boundary, you can define a trajectory segment to begin or end at a specific point expressed in any state and frame representation or by a set of state elements that are allowed to be variable between lower and upper bounds. For example, you can specify that EMTG solutions arrive at a boundary within a specific inclination range or conduct a flyby with a minimum and/or maximum distance from the flyby body. Specifying these state parameters can ensure mission requirements are satisfied, such as facilitating the spacecraft's transition to its science mission after orbit insertion.

4.4 Periapse Boundary Class

Periapse Boundary Class is effectively a special case of a Free Point and is used for Boundary events that happen at a periapsis of the spacecraft's orbit around the central body of the Journey, such as a gravity assist close approach. If using a Periapse Boundary, note that the periapse state representation is set on the "Physics Options" tab, under "State Representation Settings": "PeriapseBoundary state representation".

A Free Point may also be constrained to be a periapse in several ways, such as by using a COE state representation and setting an equality constraint that forces the true anomaly at the boundary to be 0.

5 Mars Arrival

Let's use these Boundary states to define a specific arrival orbit at Mars in the EVM mission. Open the `EVM.emtgopt` file in PyEMTG. Switch to the "Global Mission Options" tab and change the following options:

- **Mission name:** "EVM_freepoint"
  - Save the file as `EVM_freepoint.emtgopt`, keeping it in the same directory as `EVM.emtgopt`.
- **Mission type:** "Variable phase type"

5.1 Universe Files

To use the Boundary Class options to create a more realistic arrival at Mars, you need to create a Mars Universe file because you will have a Journey whose central body is Mars. Open the `Sun.emtg_universe` file in a text editor like Notepad++ or VSCode and save a new copy called `Mars.emtg_universe` in the Universe directory. The Universe file contains two sections, one for the central body and one for the "menu of bodies". You need to update the central body section from Sun values to Mars values and add the Sun to the "menu of bodies".

Using the following text as a reference, fill out the central body section of `Mars.emtg_universe`:

```
#Central body name
central_body_name Mars
#Central body SPICE ID
central_body_SPICE_ID 4
#central body radius
```

```
central_body_radius 3389.9
#central body J2
central_body_J2 1960.45e-6
#central_body_J2_reference_radius
central_body_J2_reference_radius 3389.9
#gravitational constant of central body, in km^3/s^2
mu 4.282837362069909E+04
#characteristic length unit, in km
LU 3389.9
#angles defining the local reference frame relative to ICRF, given in degrees
#alpha0, alphadot, delta0, deltadot, W, Wdot
reference_angles 317.68143 -0.1061 52.88650 -0.0609 176.630 350.89198226
#radius of the central body's sphere of influence
r_SOI 578000
#minimum safe distance from the central body
minimum_safe_distance 3559.395
# central body flattening
central_body_flattening_coefficient 0.00589
```

NOTE: `SPICE_ID = 4` is actually the Mars system barycenter; Mars itself is 499. A value of 4 is used here because the Mars barycenter data is in DE430.bsp, and Mars data is not. You would need to add a Mars-specific ephemeris file to the `ephemeris_files` directory to use Mars itself.

The Sun entry in the "menu of bodies" does not need to be as detailed as the other bodies. You only need the name, short name, number, SPICE ID, minimum flyby altitude, gravitational parameter, radius and ephemeris epoch. The other values can just be 0.0. Copy the data below into the top line of the "menu of bodies" section. Delete the line for "Mars_system" in the "menu of bodies" section of the Universe file. Note that you now have a repetition of body "numbers" in the menu of bodies: Both the Sun and Mercury have been assigned body number 1. You need the body number of be unique for each body. So, change Mercury's body number to 2, and increase the number of Venus and Earth by 1. (E.g., Venus's number will now be 3 instead of 2.) Save the file.

```
Sun S 1 10 20000000.0 1.32712440018e+11 4379000.0 0 0 0 0 51544.0 0.0 0.0 0.0 0.0
    0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
```

5.2 Journey Options

Switch to the "Journey Options" tab and make the following changes as shown in Figure 10:

- **Journey name:** "Earth_to_Mars_SOI"
- **Jounry arrival type:** "2: intercept with bounded V_infinity"
- **Journey arrival class:** "2: Ephemeris-referenced"
- **Arrival reference ellipsoid axes:** "[500000.0, 500000.0, 500000.0]"
  - This constrains the Journey to end on a sphere 500,000 km from the Mars body.

- **Journey final velocity bounds:** "0.0" to "20.0"

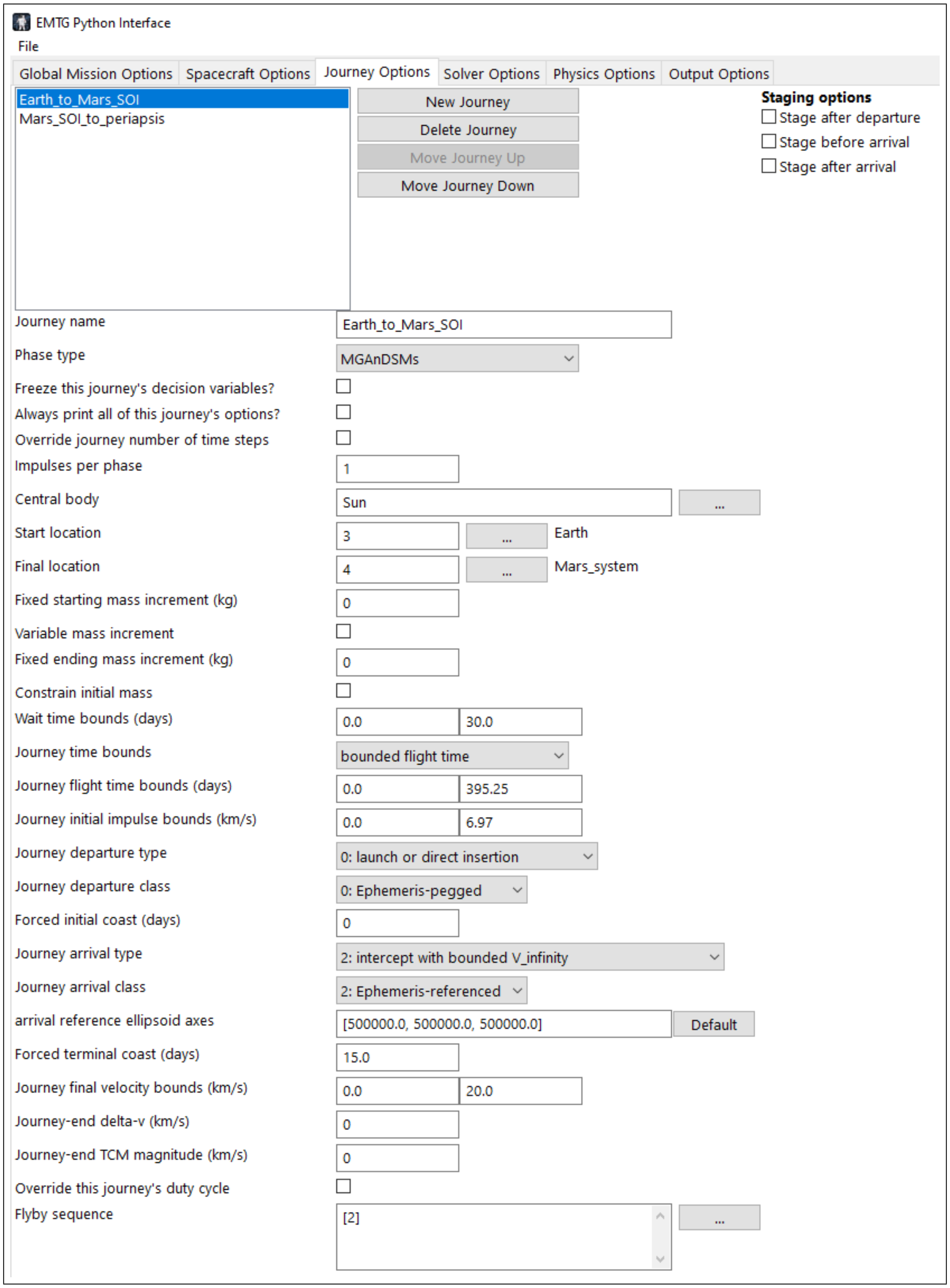

Figure 10: Earth to Mars SOI Journey.

Create a new Journey by clicking the “New Journey” button. Use the “Move Journey Up/Down” buttons to move the new' Journey below the one from Earth to Mars SOI. Make the following additional changes as shown in Figure 11:

- **Journey name:** “Mars_SOI_to_periapsis”
- **Phase type:** “Coast phase”
  - You don't want to do any maneuvers between crossing the Mars SOI until the capture orbit.
- **Wait time bounds (days):** “0” to “0”
  - You require epoch continuity at the departure Boundary of this journey.
- **Central body:** “`Mars.emtg_universe`”
  - You select the Mars Universe file rather than the Sun Universe file because this Journey takes place within the Mars SOI.
- **Journey departure type:** “2: free direct departure”
  - This means that there is no maneuver at the journey departure Boundary.
- **Journey departure class:** “1: Free point”
  - This setting means EMTG is free to move the departure Boundary of this Journey around as necessary. The “Ephemeris-referenced” setting in the previous Journey will ensure the Boundary is on the Mars SOI. The combination of “Free direct departure” from a “Free point” effectively means that you are constraining the Journey departure boundary state to be equal to the arrival boundary state of the previous Journey (full state continuity).
- **Journey arrival type:** “1: rendezvous (with chemical maneuver)”
  - This will be the orbit insertion burn.
- **Journey arrival class:** “1: Free point”
- **Journey final impulse bounds:** “0” to “100.0” km/s
  - You are not limiting this maneuver, so you just set an arbitrarily high upper bound.
- **Print a target spec line for free point arrival:** “On”
  - This is so you can see the maneuver in the output files.
- **Journey elements state representation:** “ 3: COE”
  - Specifies that you want to use classical orbital elements to define the Mars orbit.
- **Journey elements frame:** “0: ICRF”

Now, since you are using a free point arrival, a range of acceptable arrival orbit elements can be specified. Each orbit element can be "fixed" at a particular value or allowed to vary within some bounds. You will specify a near-polar, elliptical arrival orbit which might be a good choice for a Mars orbital science mission. (Remember that polar here is with respect to ICRF because of the arrival elements frame choice!) Set the arrival elements according to Table 1.

Element	Vary	Value	Lower Bound	Upper Bound
SMA (km)	Unchecked	20000.0	N/A	N/A
ECC	Unchecked	0.7	N/A	N/A
INC (degrees)	Checked	N/A	80	90
RAAN (degrees)	Checked	N/A	-360	360
AOP (degrees)	Checked	N/A	-360	360
TA (degrees)	Unchecked	0.0	N/A	N/A

Table 1: Journey Arrival Elements.

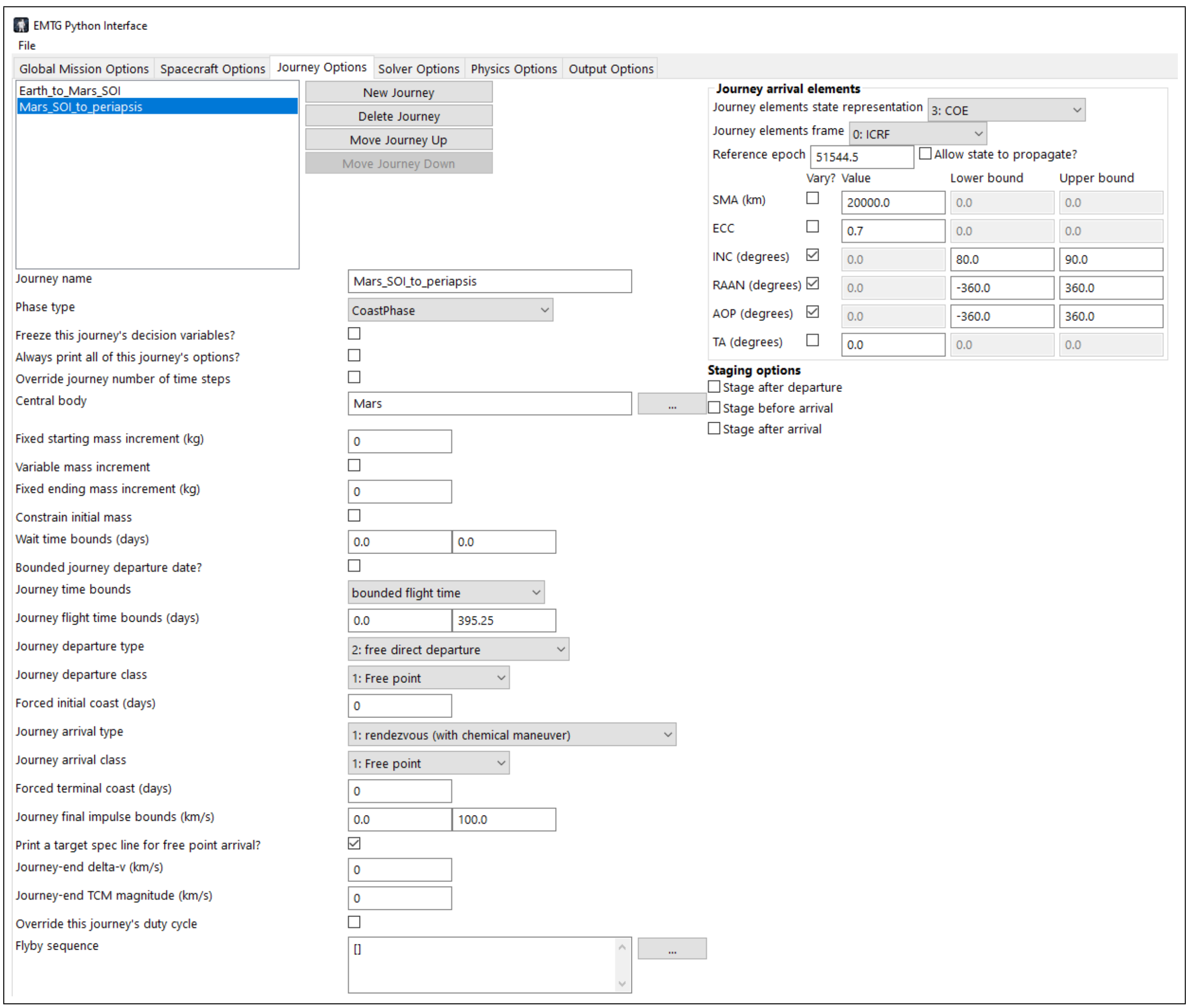

Figure 11: Mars SOI to Periapsis Journey.

6 Run and Post-process

These are all the required changes. Select "File->Run" (Ctrl+r), saving the file as `EVM_freepoint.emtgopt`. If EMTG does not find any feasible solutions, change the run time on the "Solver Options" tab to 180 seconds or higher (adjusting the "MBH Pareto distribution alpha" to 1.3 may also help) and try again. EVM_freepoint is a more difficult problem for EMTG to solve than EVM. There are more decision variables and constraints, and the problem is more sensitive. This demonstrates why you usually begin with ephemeris-pegged boundaries instead of ephemeris-referenced or free-point boundaries when designing interplanetary trajectories.

Open the `EVM_freepoint.emtg` solution file in a text editor and note that there are two sections, one for each Journey. In the "Mars_SOI_to_periapsis" section, Journey 1 (EMTG is starts numbeing at 0), note that the central body is set to Mars. The rows below contain the points defining this Journey, including the cartesian, Mars-centered, ICRF states. Check that the magnitude of initial position of the spacecraft (Journey 1, first row, columns X, Y, and Z) is at the SOI radius set in the Journey Options tab. The last row in this Journey specifies the Mars orbit after performing the insertion burn. Try converting this cartesian state into COE using General Mission Analysis Toolkit (GMAT) or another tool to verify for yourself the correct arrival orbit was reached. Figure 12 shows an example solution visualized using GMAT.

This ends the tutorial on Journey Boundaries. You now know how to specify boundary states in EMTG other than those on an ephemeris body. You will revisit this topic when realistic flybys are created in a later tutorial.

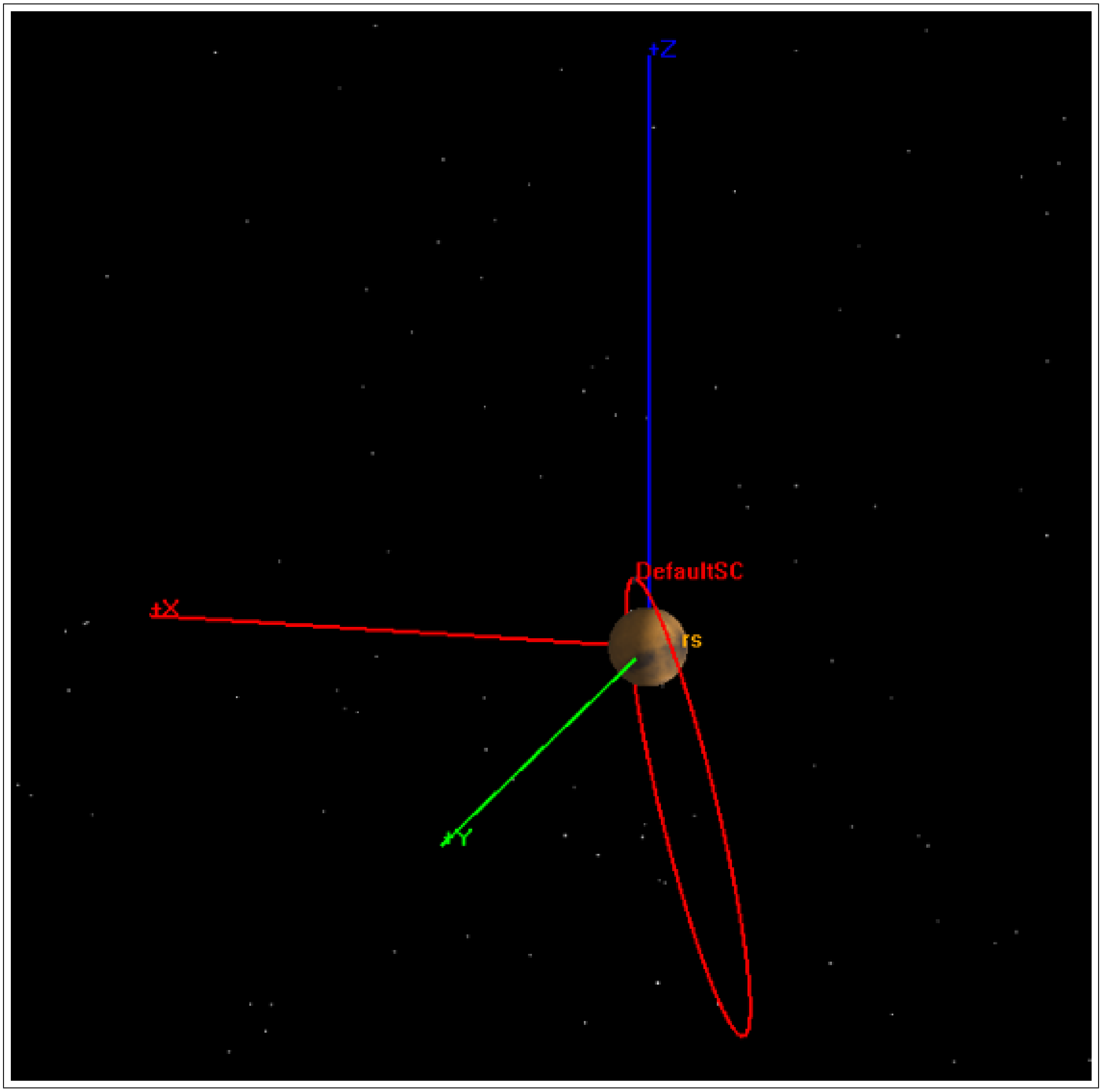

Figure 12: Mars Orbit Visualized with GMAT (Mars ICRF).