EMTG Tutorial: OSIRIS-REx

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

GUI Graphical User Interface

SNOPT Sparse Nonlinear OPTimizer

ICRF International Celestial Reference Frame

SPICE Spacecraft Planet Instrument Camera-matrix Events

STK Systems Tool Kit

NLP Nonlinear Program

MBH Monotonic Basin Hopping

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

GMAT General Mission Analysis Toolkit

1 Introduction

This first tutorial will go over the basics of Evolutionary Mission Trajectory Generator (EMTG) and design a version of the OSIRIS-REx mission to acquire a sample from the asteroid Bennu and return to Earth.

2 Directory Setup

As mentioned in the `0_EMTG_Tutorials_Intro`, a specific directory setup for each EMTG mission will be created. Begin by creating a working mission directory called `OSIRIS-REx` containing a directory called `hardware_models`. As mentioned in the `0_EMTG_Tutorials_Intro`, correctly configured EMTG files for each tutorial are provided. Copy the default EMTG hardware options files and the throttle table file from the example `Tutorial_EMTG_Files/OSIRIS-REx/hardware_models` directory into your own. Tutorial 6, Config Files, will discuss these in more detail. Separately, create an `OSIRIS_universe` directory containing an `ephemeris_files` directory (the name of this subdirectory is important: EMTG will throw an error if it cannot find a directory with this name in your Universe directory). Refer to Figure 1 as you work through your directory setup.

```
+---OSIRIS-REx
 |  |   OSIRIS-REx.emtgopt
 |  |
 |  +---hardware_models
 |          default.emtg_launchvehicleopt
 |          default.emtg_powersystemsopt
 |          default.emtg_propulsionsystemopt
 |          default.emtg_spacecraftopt
 |          empty.ThrottleTable
 +---results
       OSIRIS-REx_DATE_TIME

+---OSIRIS_universe
 |     Sun_OREx.emtg_universe
 |
 \---ephemeris_files
         bennu_refdrmc_v1.bsp
         de430.bsp
         naif0012.tls
         pck00010.tpc
```

Figure 1: Example EMTG Mission Directories.

3 Ephemeris Setup

EMTG is most accurate when a SPICE kernel is provided for each body of interest but can alternatively use Keplerian orbit elements. Using Keplerian elements will allow EMTG to run faster but reduce the accuracy of the solution. In practice, SPICE kernels are used most often.

For this OSIRIS-REx example, all required SPICE ephemeris files can be copy/pasted from `Tutorial_EMTG_Files/OSIRIS_universe/ephemeris_files`. Place them in your new `OSIRIS_universe/ephemeris_files`, as shown in Figure 1.

3.1 Create Universe File

An EMTG Universe file contains all the body information necessary for a particular Journey of a mission. An EMTG Universe file can be created by manually creating a `.emtg_universe` text file. For this tutorial, an initial step will be on how to modify an existing Universe text file to include the asteroid Bennu in addition to the planets, Sun, and Moon. The Journey Boundaries and Flybys tutorials will discuss further modification of Universe files. A correctly configured Universe file for all tutorials using the OSIRIS-REx mission is provided in the location: `Tutorial_EMTG_Files/OSIRIS_universe/Sun_OREx.emtg_universe`. This Universe file has the Sun as its central body and the planets, the Moon, and Bennu as secondary bodies.

4 Options

Now that a sun-centered Universe file with the target asteroid as a body has been created, configure the other mission options by creating and editing an EMTG Options file. Open the PyEMTG Graphical User Interface (GUI) and, in the upper left corner, select File ->New ->Mission to open the Mission Options tabs for a new mission. Save this file (Ctrl + s) as "OSIRIS-REx.emtgopt" in the top-level mission directory you created. Tip: Changing the "Mission Name" on the "Global Mission Options" tab pre-populates the file name in the save-mission dialog. Thus, it is recommended to change the Mission Name, then save the mission. Similarly, it is recommended to always save your EMTG options files with the same name as the mission.

4.1 Global

Select "Global Mission Options" if it is not already selected. Then, set the following options as shown in Figure 2:

- **Mission type:** "MGAnDSMs"
  - This mission type transcribes chemical maneuvers as impulsive burns along the trajectory and can be used as a low-fidelity model for chemical-propulsion trajectories.
- **Objective function:** "0: minimum deterministic deltaV"
- **Launch window open date:** "1 January 2016" (MJD "57388.0")
- **Global flight time bounds(days):** upper bound of "2256.75" days
- **Forced post-launch coast duration (days):** "60" days
  - This is good practice for spacecraft checkout.
- **Forced pre-flyby coast duration (days):** "45" days
- **Forced post-flyby coast duration (days):** "15" days
  - The above forced coast periods for flybys allow for cleanup of trajectory deviations.

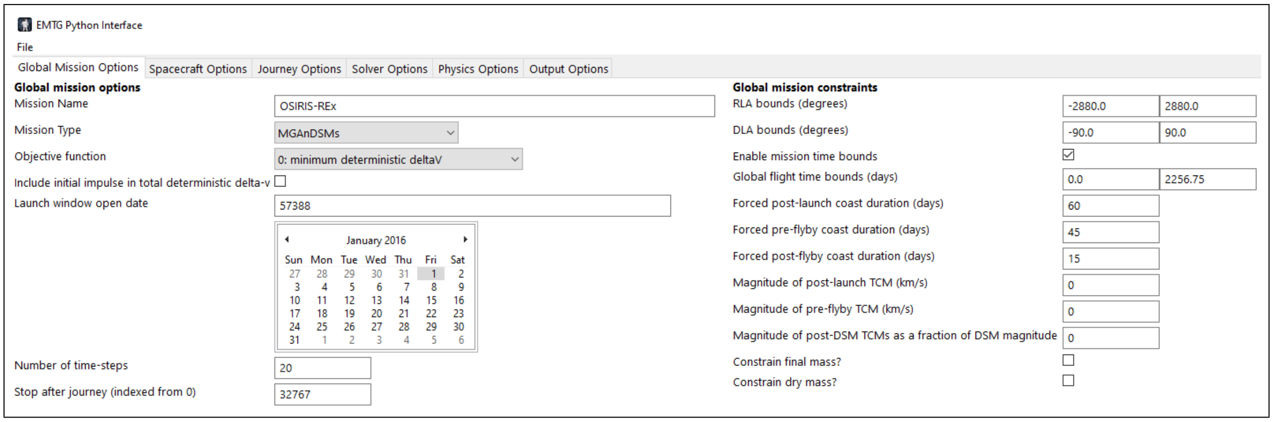

Figure 2: OSIRIS-REx Global Options.

Try saving the options file again and opening it with a text editor. Opening the options file with a text editor can be done from the GUI using the Ctrl + e hotkey. You should see a line with a key-value pair for each of the options that were changed from the default so far.

Note: There is currently a bug in PyEMTG where if you have assigned a program to open *.emtgopt files, the file will be opened in that program rather than a text editor.

4.2 Spacecraft

Change to the "Spacecraft Options" tab, which sets configuration options for the spacecraft and launch vehicle. Set the following options as shown in Figure 3:

- **Chemical Isp:** "230" seconds
- **Maximum mass:** "10000" kg
  - The maximum mass field in this case just scales the optimization problem; the actual launch mass is limited to whatever the launch vehicle can carry to the C3 that the optimizer chooses. (The function relating injected mass to C3 is set in the EMTG launch vehicle options file.)
- **Hardware library path:** the path to the `hardware_models` directory in the overall mission directory
  - Tip: Clicking on the "..." button to the left of the input will open a file explorer window where you can select the folder.
  - Be sure to include a trailing slash in the path.
- **Launch vehicle library file:** "default.emtg_launchvehicleopt"
- **Launch vehicle:** "ExampleRocket"

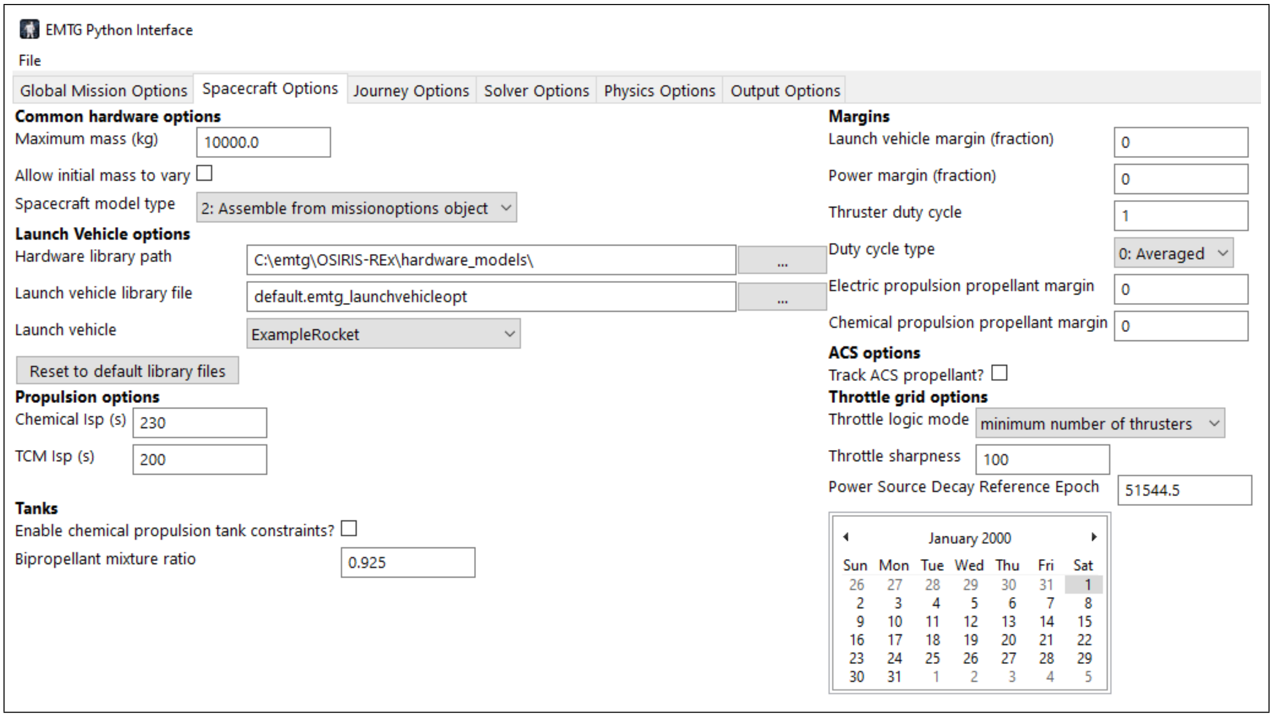

Figure 3: Spacecraft Options.

4.3 Journey

Select the "Journey Options" tab. Each Journey represents a set of events with user-defined boundary conditions. There are many types of Journey departure and arrival conditions; the ones shown here are appropriate for a small body rendezvous. Change the following options on this tab as shown in Figure 4:

- **Journey name:** "Earth_to_Bennu"
- **Central body:** "Sun_OREx.emtg_universe"
  - Note: If Sun_OREx 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_OREx.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_OREx.emtg.universe. Then, return to the "Journey Options" tab.
- **Start location:** "3" (Earth)
- **Final location:** "11" (Bennu)
  - These numbers correspond to the integer IDs listed in the Sun_OREx EMTG Universe file.
- **Wait time bounds (days):** upper bound of "365.25" days

  - Earth departure can occur up to 365.25 days after the value set for “Launch window open date” on the “Global Mission Options” tab.

- **Journey inital impulse bounds (km/s):** upper bound of “5.4102” km/s
- **Journey arrival type:** “1: rendezvous (with chemical maneuver)”
  - In addition to any DSMs between Earth departure and Bennu arrival, a chemical maneuver will be placed at the final time of the Journey to force the spacecraft to match the position and velocity of Bennu.
- **Flyby sequence:** “[3]”
  - This is a list of Universe body integer IDs. It can have zero, one, or more than one entry. In this case, the list has one element, 3, for Earth because OSIRIS-REx does an Earth flyby on its way to Bennu. Each entry added to the list adds a Phase to the Journey: If a Journey has more than one Phase, they are separated by planetary flybys. In this case, one Earth flyby has been selected, and the Journey has two Phases.

Tip: PyEMTG text boxes can perform basic math operations, so a value of 730.5 can also be achieved by typing “365.25 * 2” and deselecting the text input box.

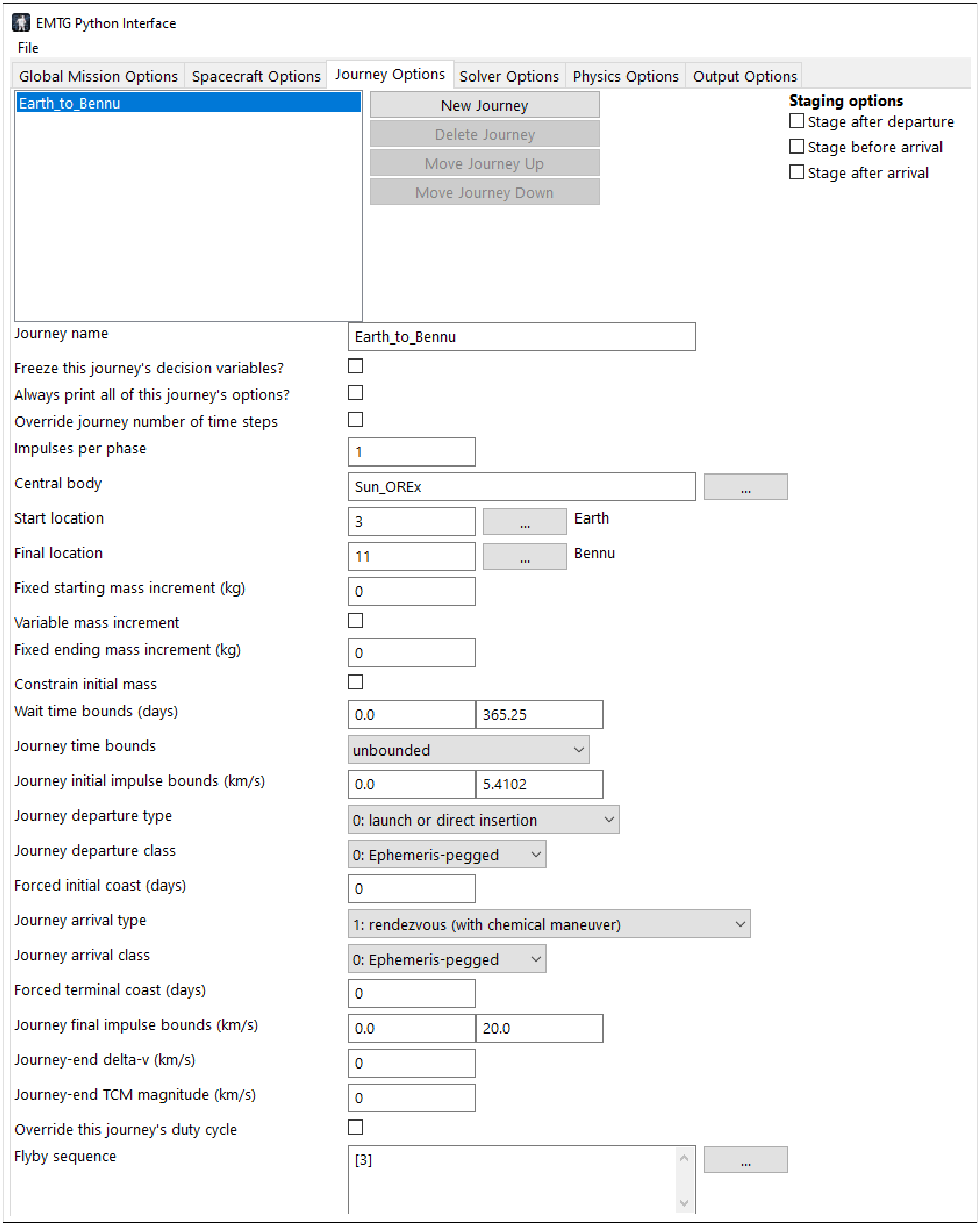

Figure 4: Earth to Bennu Options.

Create a new Journey for the return to Earth by clicking “New Journey” and then clicking on “default” in the list of Journeys to see the new Journey’s properties. Then, set the following Journey options as shown in Figure 5:

- **Journey name:** “Bennu_to_Earth”
- **Central body:** “Sun_OREx.emtg_universe”
- **Start location:** “11” (Earth)
- **Final location:** “3” (Bennu)
- **Wait time bounds(days):** lower bound of “730.5” days and upper bound of “1461.0” days
  - This constrains OSIRIS-REx to spend between 2 and 4 years at Bennu.
- **Journey inital impulse bounds (km/s):** upper bound of “10” km/s
  - The initial impulse is an impulsive chemical maneuver applied at the beginning of the Journey. Like the maneuvers at the beginning and end of the “Earth_to_Bennu” journey, this maneuver does not count against the “Impulses per phase” count.
- **Journey arrival type:** “2: intercept with bounded V_infinity”
  - Here, an Earth arrival as an intercept with bounded v-infinity is being simulated so that the Journey will end with the spacecraft matching the position of the Earth, and the magnitude of the difference between the Earth’s heliocentric velocity and the spacecraft’s heliocentric velocity at this point is constrained. EMTG can model the flight path of OSIRIS-REx from the boundary of Earth’s sphere of influence to its interface with the atmosphere and supports a variety of entry constraints. However, this is out of scope for the tutorial. See the scripted constraints document in the EMTG repo in `docs/0_Users/constraint_scripting` for more details.
- **Journey final veclocity:** “6” km/s
  - This is the bound on the V_infinity of the spacecraft with respect to the arrival body (Earth) at the end of the Journey.

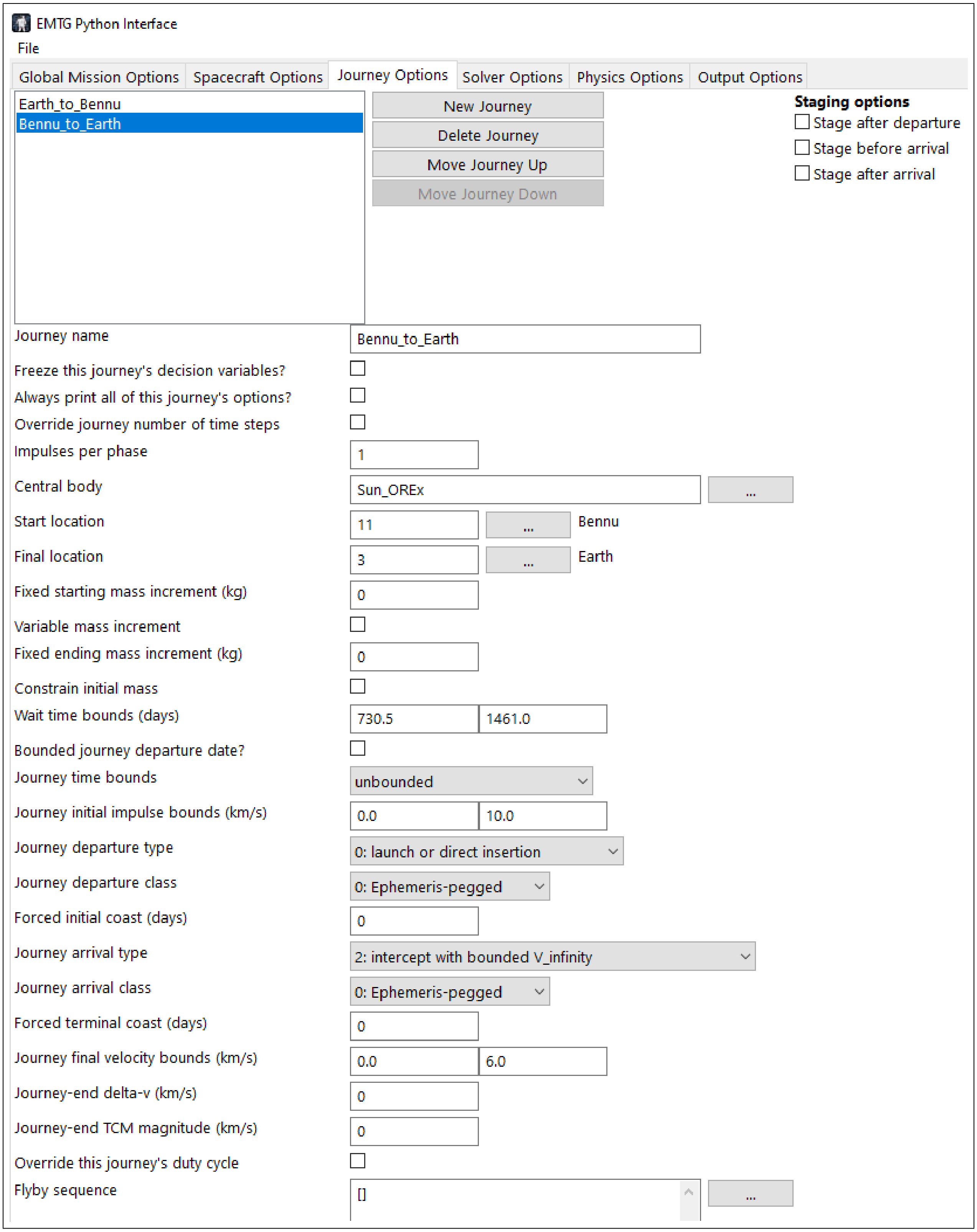

Figure 5: Bennu to Earth Options.

4.4 Physics

The Physics tab specifies the SPICE kernels and configuration to use, force model settings, and state propagation settings. Set the following options as shown in Figure 6:

- **Universe folder:** the path to the `OSIRIS-REx/OSIRIS_Universe` directory
- **Propogator type:** "Keplerian"
  - At this stage of design, "Keplerian" is being used because it is much faster than "Integrator".
- **Earliest possible SplineEphem epoch:** "57037.0"
- **Latest possible SplineEphem epoch:** "60079.0"
  - These epoch settings are due to the shortened Bennu ephemeris.

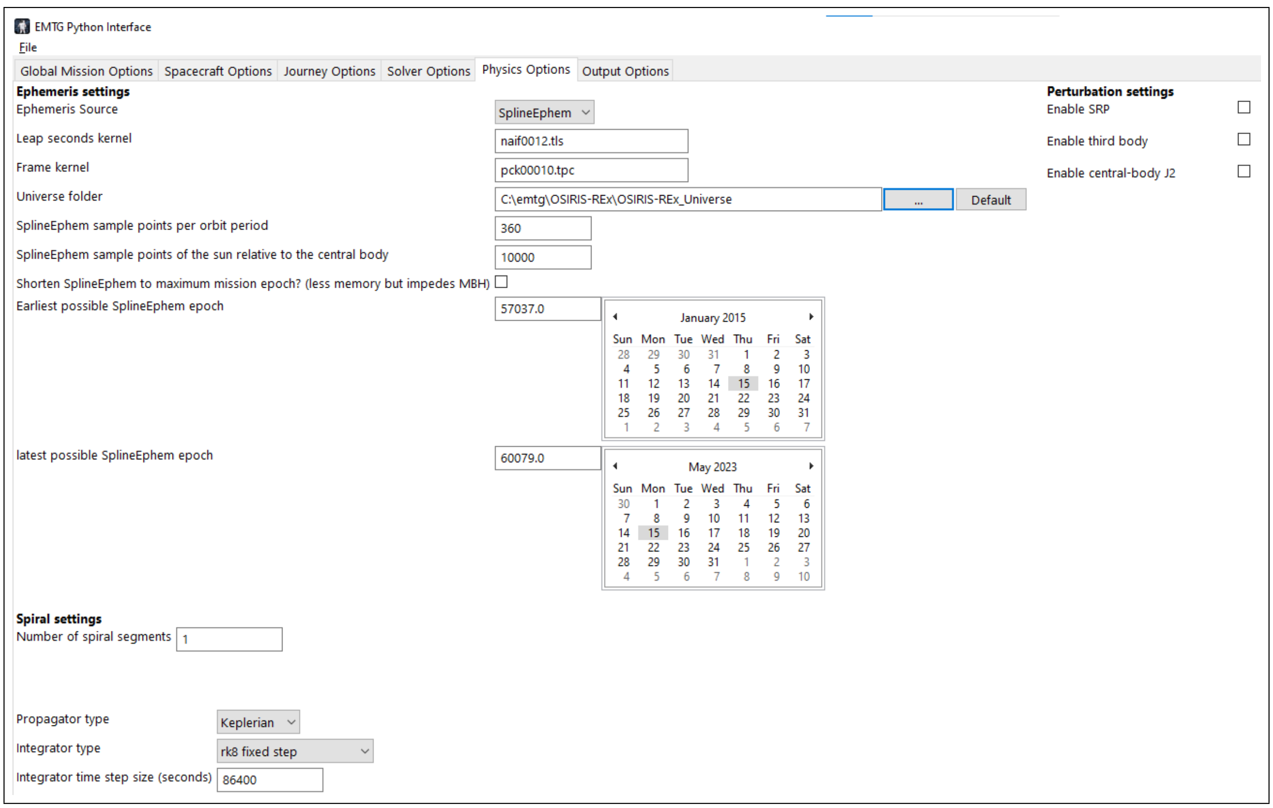

Figure 6: Physics Options.

The "Perturbation settings" checkboxes do not work for Multiple Gravity Assist with n Deep-Space Maneuvers using shooting (MGAnDSMs) Phases with the "Keplerian" propagator, so do not check any of them. (Propagation and Phase types will be covered in more detail in later tutorials.)

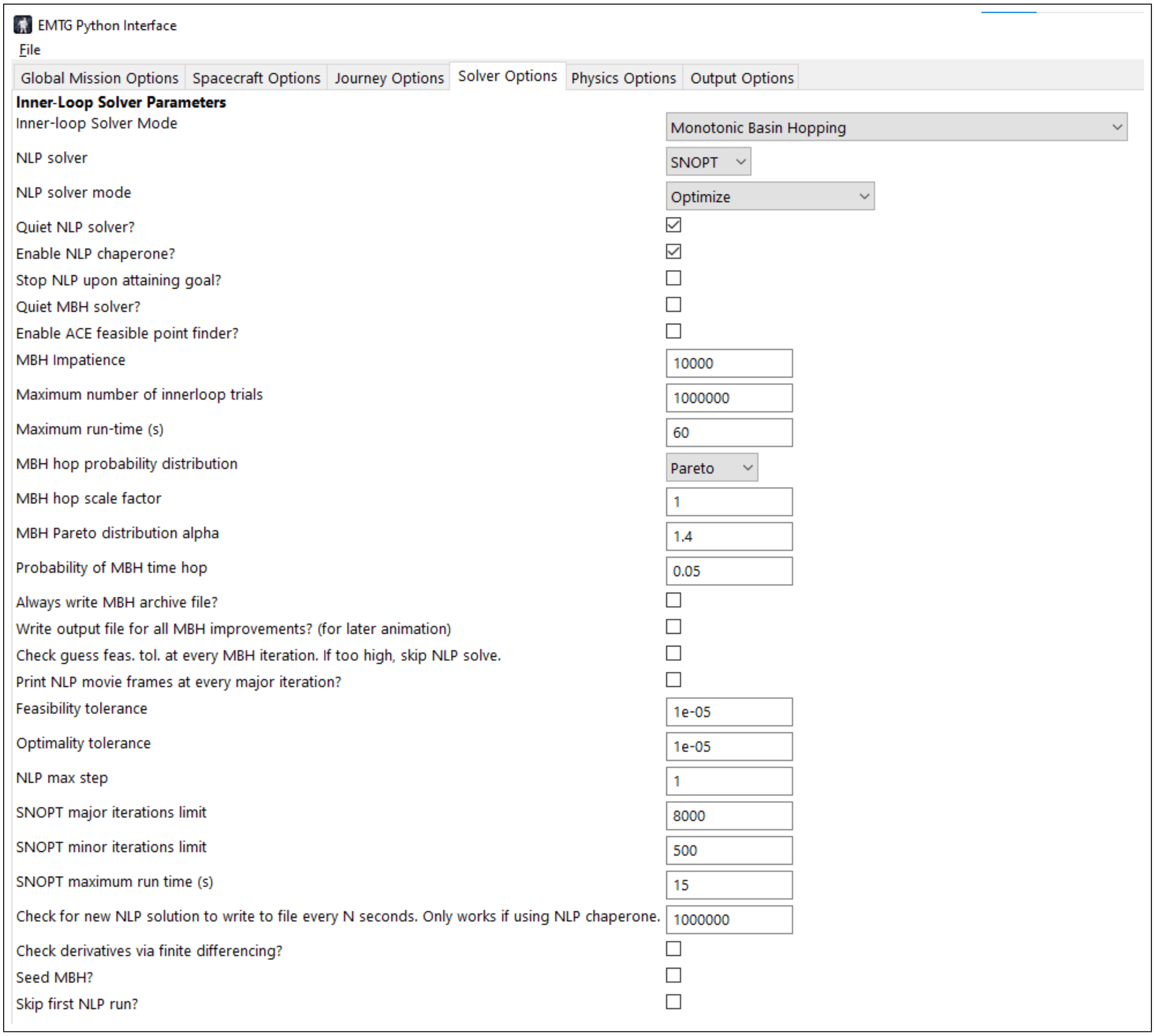

Figure 7: Solver Options.

4.5 Solver

Switch to the Solver Options tab and set the following options as shown in Figure 7:

- **Inner-loop Solver Mode:** "Monotonic Basin Hopping (MBH)"
- **MBH hop probability distribution:** "Pareto"
- **Pareto alpha:** "1.4"
  - For chemical missions like this one, the recommendation is to set the "Pareto alpha" to 1.4 (the default) or, if the problem has many local optima, 1.3.

- **ACE feasible point finder:** “On”
  - This settings allows MBH to compare infeasible solutions and search for minimum infeasibility before it finds its first feasible solution and begins to search for optimality. In most cases, it is recommended to have the ACE feasible point finder on.
- **Nonlinear Program (NLP) chaperone:** “On”
  - This settings enables EMTG to recover the best solution found in a run of Sparse Nonlinear OPTimizer (SNOPT) even if SNOPT does not converge. In most cases, it is recommended to have the NLP chaperone on.

4.6 Output

Switch to the Output Options tab and set the following options as shown in Figure 8:

- **Background mode:** “Off”
  - Background mode means “close EMTG as soon as it is done executing”. If you are running EMTG from PyEMTG, then you should leave background mode off so that you can see your results more easily.
- **Print only non-default options to .emtgopt file:** “On”
  - This settings means write a short-form emtgopt file that takes up less space on your hard drive. Options that have their default value are not written to the file. If this option is unchecked, all options are written to the emtgopt file, regardless of whether or not they have their default value.
- **Output file frame:** “ICRF”
- **Override working directory:** “On”
- **Working directory:** create the `OSIRIS-REx/results` directory and set this to the path to that directory
  - This will cause EMTG to place its results in this folder instead of the default location inside the EMTG repo, and lets you keep the results with the EMTG options files that created them.

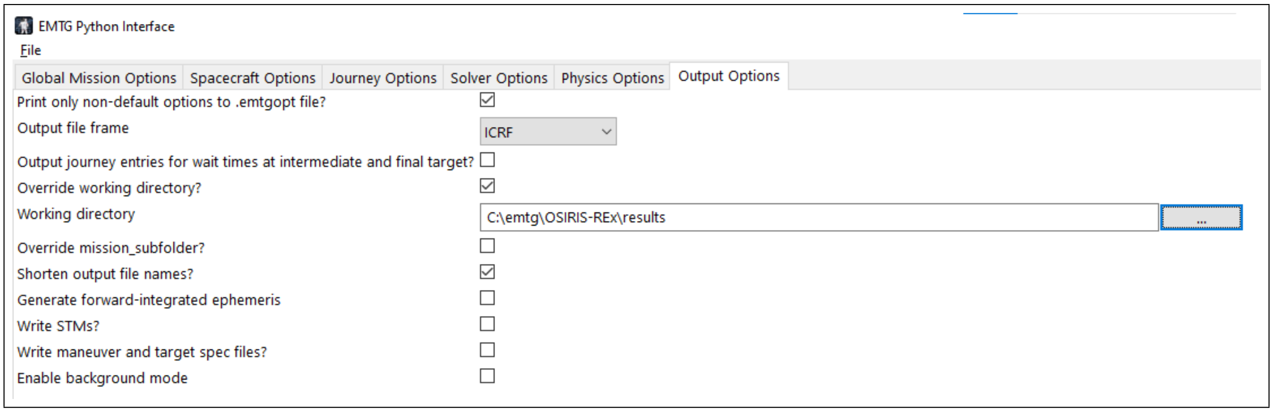

Figure 8: Output Options.

5 Run Mission

Select File->Run as shown in Figure 9. PyEMTG will prompt you to save an emtgopt options file to ensure that your emtgopt file is saved to disk prior to executing EMTG. (The same prompt as if you selected File->Save.) EMTG will then execute, create a new timestamped subdirectory for your results within your results directory, and begin solving your problem. You can also save and run the mission from the command line by passing the emtgopt file as a command-line argument to the executable. For example, replacing "<EMTG-FOLDER>" and "<OSIRIS-REx>" folder with the correct paths and running:
`C:\<EMTG-FOLDER>\bin\EMTGv9.exe C:\<OSIRIS-REx-FOLDER>\OSIRIS-REx.emtgopt`

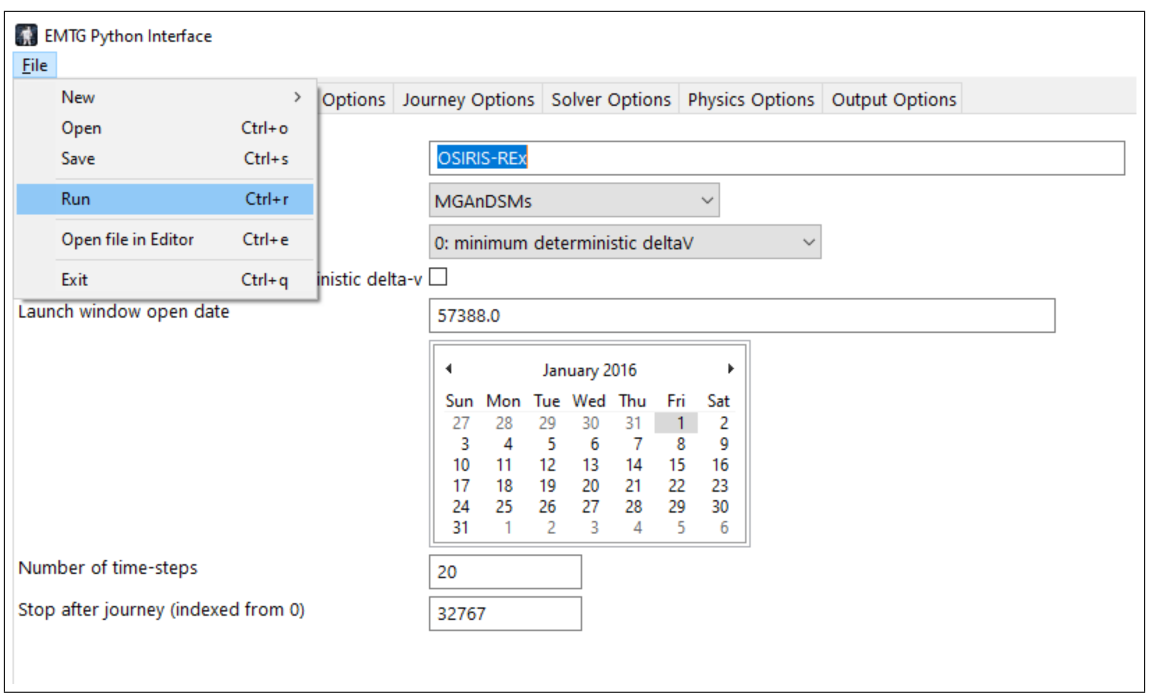

Figure 9: Running a mission.

While running, EMTG outputs information about its progress to the command window in which it is executing, showing information about the current decision vector feasibility and optimality compared to the best solution found so far. Since the Bennu SPICE kernel used covers a smaller range of time than DE430, you may see occasional SPICE errors, but the run should still finish. (Even if a given SNOPT run fails, the next MBH iteration will start normally.) If everything is configured correctly, you should see output similar to Figure 10 when the EMTG run ends. (The EMTG run will last about 1 minute; this can be configured in the "Solver Options" tab.) Note that, due to the short run time used for EMTG and the stochastic nature of MBH, your "Best value found" may not be the same as that shown in Figure 10, but it is very likely that EMTG will at least find a feasible solution. Your results subdirectory will contain, among other files, a .emtg file that describes the solution.

```
Hop evaluated mission Sun(EEB)_Sun(BE) with fitness 2.12125
NLP incumbent point and exit point are feasible and incumbent point is
superior to exit point.
Worst constraint is F[21]: j0p1MGAnDSMs: match point xdot
with violation -4.27227e-07
Hop evaluated mission Sun(EEB)_Sun(BE) with fitness 7.07911

Best value found was 2.12118
EMTG run complete.
Press enter to close window.
```

Figure 10: Example EMTG run output.

6 Post-Process

The .emtg solution file contains the final trajectory, objective value, scaling information, Journey events, and other information to describe the trajectory. It can also be used as an initial guess for other EMTG runs. This feature is covered in other tutorials. For now, open the .emtg file in the results directory in a text editor and look through the information contained inside. An example file is shown in Figure 11.

Mission: OSIRIS-REx
Written by EMTG_v9 core program compiled Jun 9 2023 09:42:05

Journey: 0
Journey name: Earth_to_Bennu
Central Body: Sun
Radius (km): 695700
mu (km^3/s^2): 132712440040.94459534
Characteristic length unit (km): 149597870.69100001454
Frame: ICRF

#	JulianDate (ET/TDB)	MM/DD/YYYY	event type	location	step size (days)	altitude (km)	BdotR (km)	BdotT (km)	RA degrees	DEC degrees	C3 (km^2/s^2)
1	2457655.06575168	9/23/2016	launch	Earth	0.00000000	-	-	-	-41.016	50.394	29.27026
2	2457664.38770065	10/2/2016	coast	deep-space	18.64389793	-	-	-	0.000	0.000	-
3	2457683.03159857	10/21/2016	coast	deep-space	18.64389793	-	-	-	0.000	0.000	-
4	2457701.67549650	11/9/2016	coast	deep-space	18.64389793	-	-	-	0.000	0.000	-
5	2457720.31939443	11/27/2016	coast	deep-space	18.64389793	-	-	-	0.000	0.000	-
6	2457738.96329235	12/16/2016	coast	deep-space	18.64389793	-	-	-	0.000	0.000	-
7	2457757.60719028	1/4/2017	coast	deep-space	18.64389793	-	-	-	0.000	0.000	-
8	2457776.25108821	1/22/2017	coast	deep-space	18.64389793	-	-	-	0.000	0.000	-
9	2457794.89498613	2/10/2017	coast	deep-space	18.64389793	-	-	-	0.000	0.000	-
10	2457813.53888406	3/1/2017	coast	deep-space	18.64389793	-	-	-	0.000	0.000	-
11	2457832.18278199	3/19/2017	coast	deep-space	18.64389793	-	-	-	0.000	0.000	-
12	2457850.82667991	4/7/2017	coast	deep-space	18.64389793	-	-	-	0.000	0.000	-
13	2457860.14862888	4/16/2017	chem_burn	deep-space	0.00000000	-	-	-	-72.255	-44.958	-
14	2457860.14862888	4/16/2017	match_point	deep-space	0.00000000	-	-	-	0.000	0.000	-
15	2457870.15890332	4/26/2017	coast	deep-space	20.02054888	-	-	-	0.000	0.000	-
16	2457890.17945220	5/16/2017	coast	deep-space	20.02054888	-	-	-	0.000	0.000	-
17	2457910.20000109	6/5/2017	coast	deep-space	20.02054888	-	-	-	0.000	0.000	-
18	2457930.22054997	6/25/2017	coast	deep-space	20.02054888	-	-	-	0.000	0.000	-
19	2457950.24109886	7/15/2017	coast	deep-space	20.02054888	-	-	-	0.000	0.000	-
20	2457970.26164774	8/4/2017	coast	deep-space	20.02054888	-	-	-	0.000	0.000	-
21	2457990.28219663	8/24/2017	coast	deep-space	20.02054888	-	-	-	0.000	0.000	-
22	2458010.30274551	9/13/2017	coast	deep-space	20.02054888	-	-	-	0.000	0.000	-
23	2458020.31301995	9/23/2017	upwr_flyby	Earth	0.00000000	300.26838356	11725.99590455	-9399.19472601	-171.419	35.098	29.37506
24	2458031.94465546	10/5/2017	coast	deep-space	23.26327101	-	-	-	0.000	0.000	-
25	2458055.20792647	10/28/2017	coast	deep-space	23.26327101	-	-	-	0.000	0.000	-
26	2458078.47119748	11/20/2017	coast	deep-space	23.26327101	-	-	-	0.000	0.000	-
27	2458101.73446849	12/14/2017	coast	deep-space	23.26327101	-	-	-	0.000	0.000	-
28	2458124.99773950	1/6/2018	coast	deep-space	23.26327101	-	-	-	0.000	0.000	-
29	2458148.26101051	1/29/2018	coast	deep-space	23.26327101	-	-	-	0.000	0.000	-
30	2458171.52428152	2/22/2018	coast	deep-space	23.26327101	-	-	-	0.000	0.000	-
31	2458194.78755253	3/17/2018	coast	deep-space	23.26327101	-	-	-	0.000	0.000	-
32	2458218.05082354	4/9/2018	coast	deep-space	23.26327101	-	-	-	0.000	0.000	-
33	2458229.68245905	4/21/2018	chem burn	deep-space	0.00000000	-	-	-	115.150	7.643	-

Figure 11: Selection from .emtg solution file.

PyEMTG can open .emtg files as shown in Figure 12. PyEMTG can create basic trajectory plots and systems summary plots, and process various other EMTG outputs. For this example, the trajectory plot will be focused on. PyEMTG can also search for targets of opportunity along your trajectory. This is out of scope for the tutorial.

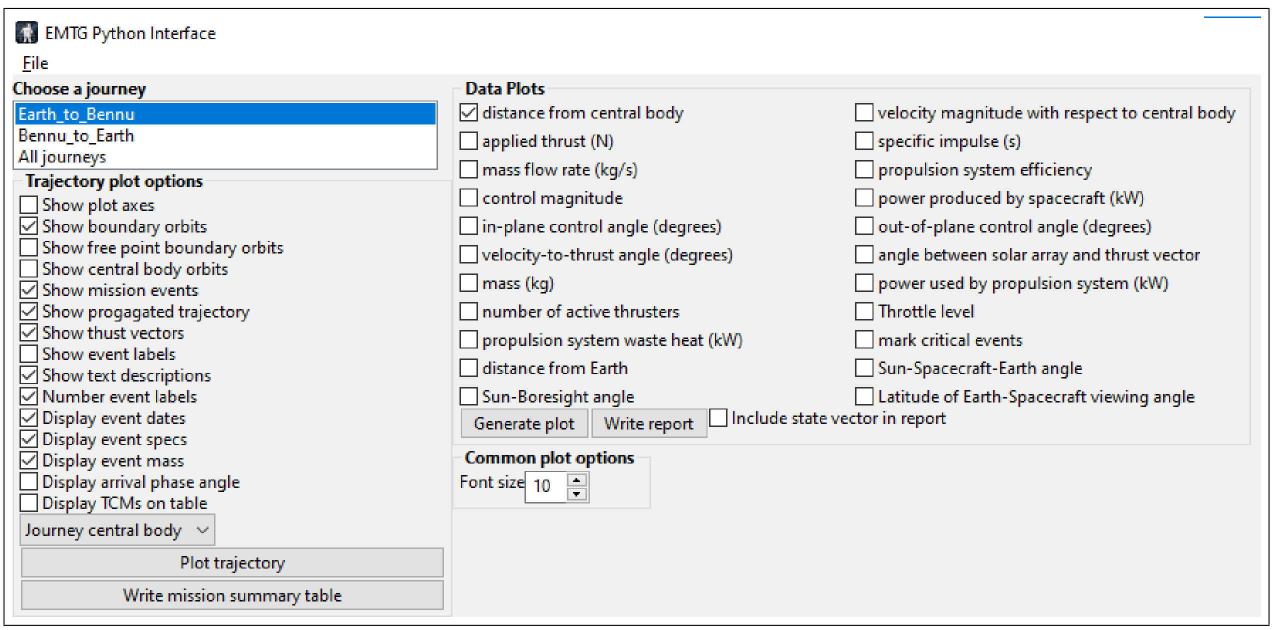

Figure 12: A .emtg file in PyEMTG.

Open the .emtg file in PyEMTG by selecting File->Open and navigating to the file. Try plotting the results by selecting “All journeys” and “Show event labels”, then clicking “Plot trajectory”.

You should see something similar to Figure 13. Note that PyEMTG's plotting utilities are quite basic. For more advanced plotting, the workflow is to use an EMTG solution to create a bsp and import the bsp into a tool like General Mission Analysis Toolkit (GMAT) or Systems Tool Kit (STK). Using an EMTG solution to create a bsp is beyond the scope of this tutorial.

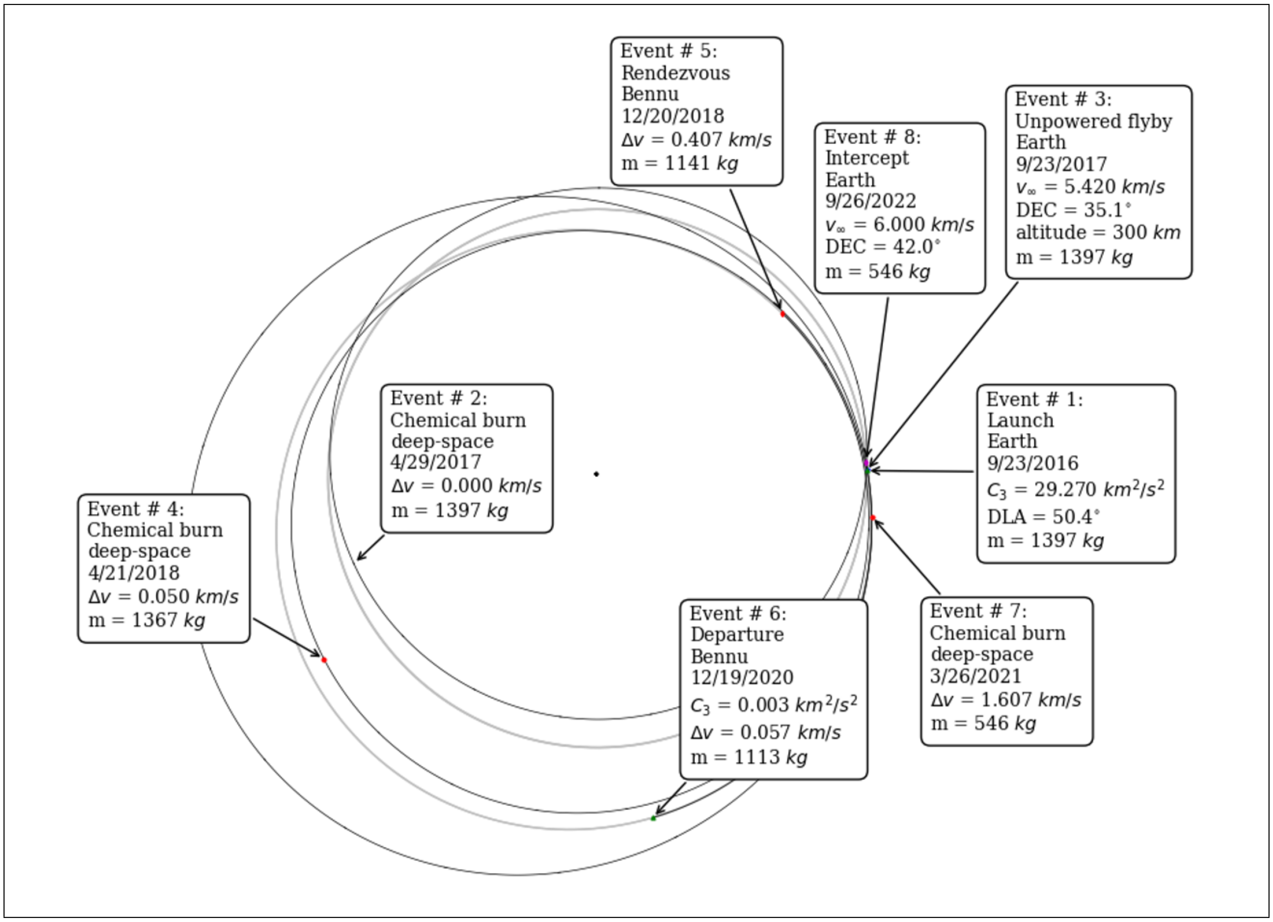

Figure 13: Example EMTG trajectory plot.

This concludes the first EMTG tutorial! This mission is the basis for the low-thrust OSIRIS-REx tutorial, LowSIRIS-REx.