EMTG Tutorial: Introduction and Setup

Tim Sullivan *

August 4, 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

MONTE Mission analysis, Operations, and Navigation Toolkit Environment

PEATSA Python EMTG Automated Trade Study Application

NASA National Aeronautics and Space Administration

SPICE Spacecraft Planet Instrument Camera-matrix Events

STK Systems Tool Kit

GMAT General Mission Analysis Toolkit

1 Introduction

Welcome to the tutorial series on NASA's Evolutionary Mission Trajectory Generator (EMTG). EMTG is a tool for the design of space missions using either high-thrust chemical or low-thrust electric propulsion and, optionally, planetary flyby maneuvers. EMTG is capable of determining both the optimal flyby sequence (using a Python-based outer loop) and also the optimal trajectory (using a nonlinear programming solver and monotonic basin hopping). EMTG can operate at multiple levels of modeling fidelity that are suitable for trade studies, some proposals, and use as initial guesses for flight-fidelity tools. EMTG is not an operational tool like STK, GMAT, or MONTE.

EMTG is composed of two components. The core EMTG program is written in C++ and is driven by a text script interface. The second component is a Graphical User Interface (GUI) written in Python called PyEMTG, which is used to process EMTG input and output scripts. The two programs are independent but complementary.

2 Learning Objectives

This tutorial series provides seven introductory lessons on EMTG. Each tutorial focuses on a different aspect of EMTG. Each lesson builds on one or more previous lessons, so it is highly

recommended that the lessons be completed in order. The lessons begin with low-fidelity chemical and low-thrust missions. Each tutorial adds additional realism and details to the initial simplistic missions, similar to the way someone might design a real-world mission.

The list of tutorials is provided below, in the order in which they should be performed. Each has an additional directory of the same name in the Tutorial_EMTG_Files directory with the accompanying EMTG files the user will create during the tutorial and example EMTG results. Each tutorial expects the user to have completed all the previous tutorials.

1. **OSIRIS-REx:** Basic introduction to EMTG chemical missions. Creates a low-fidelity, patched-conic, multi-phase mission similar to the OSIRIS-REx trajectory.
2. **LowSIRIS-REx:** Basic introduction to EMTG low-thrust missions. Modifies the OSIRIS-REx mission for low-thrust and introduces two transcription methods for low-thrust missions in EMTG.
3. **Boundary Types:** Explains how EMTG models Journey Boundaries (departure and arrival states).
4. **Propagation and Force Models:** Provides additional details on EMTG "Mission Types" or transcription methods, propagation options, and perturbing forces.
5. **Flybys:** Takes the user through the conversion of a low-fidelity patched conics mission to a high-fidelity mission with realistic flybys.
6. **Config Files:** Explains how to configure EMTG spacecraft and launch vehicle text configuration files to model real-world hardware and vehicles.
7. **PEATSA:** Introduces the Python EMTG Automated Trade Study Application (PEATSA). PEATSA allows users to explore the trade space of mission options and discover how different configurations affect the final trajectory.

3 Conventions

3.1 Environment

All tutorials except the PEATSA tutorial are conducted in a Windows environment using the EMTG Python GUI, PyEMTG, and, occasionally, a text editor such as Notepad++. PEATSA runs are typically conducted on a Linux multi-CPU environment, so the PEATSA tutorial is written assuming the user has access to a Linux workstation.

EMTG files include full paths in several locations. The EMTG files provided were created assuming the working directory is located at `C:\EMTG\Tutorials\NAME-OF-TUTORIAL`. For example, the files for the first tutorial, OSIRIS-REx, expect to be located inside `C:\EMTG\Tutorials\OSIRIS-REx`. You are free to place your tutorial files wherever you want as long as you update paths appropriately.

3.2 Formatting

Throughout the tutorials, EMTG-specific features and naming conventions such as Universes and Journeys will be capitalized. Specific items and values the user should input in PyEMTG will be in quotations, such as "Mission Types". When discussing text in a terminal or EMTG text file, this tutorial series will use a `monospaced font`.

4 Initial EMTG Setup

These tutorials cover using EMTG and assume that EMTG and PyEMTG have already been installed and set up. For more information on these topics, see the files in `docs/0_Users/build_system` and `PyEMTG/docs`.

5 Mission Directory Setup

EMTG will utilize four main directories/folders when run: the directory in which the input text file is located, Universe, Hardware, and results. For these tutorials, a working or mission directory setup for each tutorial will be created like the OSIRIS-REx directory shown in Figure 1.

```
+---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.

This directory will contain the EMTG options file itself, the spacecraft and launch vehicle hardware files in the directory `hardware_models`, and the EMTG results in the `results` directory. The other necessary directory is the universe directory containing the EMTG Universe file and another directory `ephemeris_files` containing all ephemeris files needed for the mission. More information on the Universe directory is provided in the next section and Tutorial 1: OSIRIS-REx. The `ephemeris_files` directory name is specific to EMTG. If any other name is used, EMTG will not be able to find the necessary Spacecraft Planet Instrument Camera-matrix Events (SPICE) files.

5.1 Universe Directory

Each EMTG Journey (Journeys will be explained in detail in the tutorials) uses a `.emtg_universe` file listing the relevant bodies (planets, moons, asteroids, etc.) for that mission, which is kept in a directory referenced by the EMTG options (.etmgopt) file.

EMTG uses ephemeris files provided by the Jet Propulsion Laboratory Navigation and Ancillary Information Facility (JPL NAIF). The standard ephemeris files for any mission are `de430.bsp` (planetary ephemeris), `naif0012.tls` (leap seconds), and `pck00010.tpc` (frame), although other versions from JPL NAIF may be used. To use a body not in the DE430 ephemeris, include its `.bsp` file in the `ephemeris_files` folder. The SPICE ephemeris files can be found on the JPL NAIF site here: DE files, frames, leap seconds.

This tutorial series will create two Universe directories and each tutorial will use one or the other.