**MONTE-Copernicus Interface Test Summary**

Tool and Case Developer: R. Restrepo (NASA JPL)

Case Tester: P. Valerino (NASA Marshall)

The Monte-Copernicus interface has been developed as part of the NESC Flight Mechanics Analysis Tools Interoperability and Component Sharing project (TI-18-01313). This interface is an application that allows users to use a NAIF SPK trajectory file, read in MONTE software for mission or navigation analysis, and then visualize resulting trajectory in graphics tool, i.e. Copernicus (Ref 1). The Interoperability Tool creates a unique opportunity to utilize any SPK trajectory file given limited knowledge of maneuver size and location to perform further mission design and/or navigation analysis within the GMAT, Copernicus, or MONTE software suites. The following section describes the test objectives, configuration and activities that were performed as part of the testing of the NESC Interoperability Tool.

**Test Objective:** The purpose of testing the MONTE-Copernicus interface was to execute python scripts bsp2cosmic.py and bsp2visualCop.py, some of the main applications of the interface, on a different operating system and environment other than what was used by the tool developer. Successful test criteria included the ability to execute tool with no errors (or minor platform configuration updates).

**In-Scope:** The functional tests performed that are in Scope of Testing include:

Reading and converting trajectory files to other formats; reading-in trajectory files for use in other design software tool sets; using another platform configuration.

**Out-of-Scope:** Generation of multiple-SPK file inputs for testing, and testing tool on other platform than what is described in this document are out of scope for this test.

**Areas not tested: --**

**Test Environment and Platform Configuration**

**System Pre-requisites:** While there are many variations of getting this tool to operate, the following software and system configuration was the selected method for this documentation.

* Windows 10 Enterprise on 64-bit Operating System
* Copernicus 5.2.0
* Docker Engine v20.10.21
* MONTE version 149 software (with SNOPT and DBLSE optimizers)

NOTE: While the MONTE toolkit can be accessed on Linux/RedHat machines, the testing for the NESC Flight Mechanics Analysis Tools Interoperability and Component Sharing project was performed by a software architecture and workflow developed by J. Everett et al. at NASA Marshall for Mission Design and Navigation (MDNav) analysis (Ref 2). Initially prepared for the Solar Cruiser Mission MDNav team, the MDNav Tool Suite utilizes containerization technology to overcome any challenges of having “human in the loop” to maintain any operational dependencies.

In this case, the Docker containerization was chosen to provide access to the MONTE toolkit. Unlike a virtual machine that emulates a physical computer inside a host machine, a Docker container only virtualizes the application layer and runs on top of the host operating system facilitated by a Docker engine. Since Docker is compatible with Windows, MacOS, and Linux, one Docker image that contains all tools and dependences can be deployed and loaded to an users local Docker management system. The MDNav Tool Suite is distributed as a main development image (contains MONTE and its dependencies) and an auxiliary documentation image (includes MONTE documentation HTML files). The MDNav Tool Suite was further developed to support any project such as the testing described in the following section.

**Testing Approach:** Two use cases were identified and underwent integration testing. Scripts, input files, and additional README files are located at the NESC Interoperability Tool project NASA Box.

Use Case 1.0: Human Landing System Lunar Low Orbit to NRHO transfer (LLO-to-NRHO) Trajectory

Use Case 2.0: Copernicus as a Trajectory Visualization Capability

**Files Required:**

* MONTE-Copernicus interface scripts (contained in the python module)
* Initial Copernicus trajectory: gen\_LLO\_to\_NRHO\_imp\_BSP.ideck
* Copernicus generated SPK kernel: gen\_LLO\_to\_NRHO\_imp\_BSP.bsp
* Reference NRHO (15-year Gateway reference orbit produced by JSC): receding\_horiz\_3189\_1burnApo\_DiffCorr\_15yr.bsp

**Test Procedure:**

1. Start with an HLS Copernicus solution transfer trajectory (provided by Marshall Space Flight Center) LLO-to-NRHO (Figure 1).  Create an SPK file containing the transfer trajectory (assigned SPICE id as -30100). The transfer has been modified to implement only impulsive solutions (original solution is modeled with finite burns).
2. Use bsp2cosmic\_HLS.py to generate a pre-converged Cosmic Solution. This is the main script of the interface which converts SPK kernels (\*.bsp files) into MONTE/Cosmic trajectories. Run scripts as

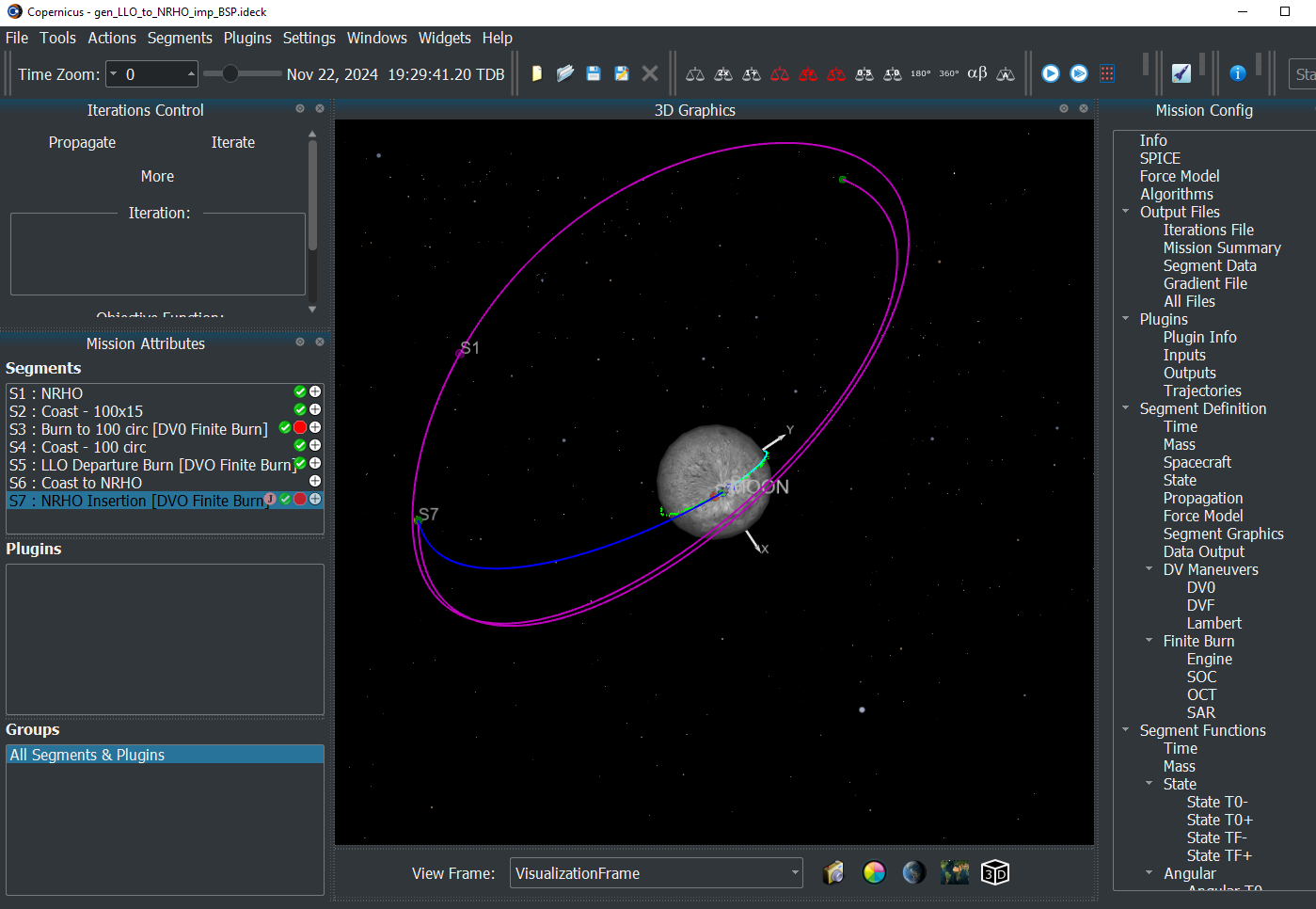
>> bsp2cosmic.py gen\_LLO\_to\_NRHO\_imp\_ext7d\_BSP.bsp -ov -o 3 -tl 1 -dt 10 -dv 20 -n LLO\_to\_NRHO\_Cosmic.py

* 1. Test script, explore the user input options. Test the Velocity discontinuity visualization capability, and parameter tuning (e.g. dvSearch time step, and minDV search) (See Figure 2).
  2. After tuning and finding the 3 Impulsive maneuvers for the LLO to NRHO transfer, save Cosmic solution.

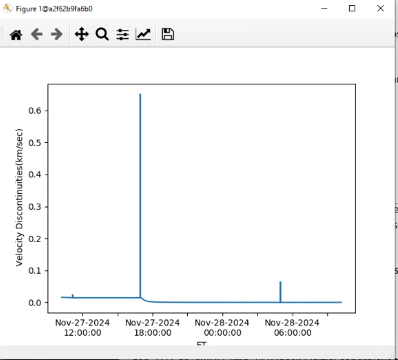
1. Converge cosmic solution to obtain a continues optimized trajectory (use SNOPT or DBLSE optimizers). See Figure 3.
   1. Output should be a MONTE-python file.
   2. Generate new files with continues optimized solution: \*.spk file as LLO\_to\_NRHO\_Cosmic\_OPT.bsp (assigned SPICE id as -30100), Cosmic timeline as LLO\_to\_NRHO\_Cosmic\_OPT.py, and \*.boa file as LLO\_to\_NRHO\_Cosmic\_OPT.boa. Note: BOA files are binary files archives used by MONTE to save all information related with the trajectory, as ephemeris, trajectory itself, partials, spacecraft models, etc. The BOA file is an additional product that will allow Navigation post-processing analysis, like Orbit Determination, or Flight Path Control analysis.
   3. Plot the trajectory in the COSMIC viewer (Figure 4).
2. Use Case 2.0: Copernicus as a Trajectory Visualization Capability:

Using the optimized Cosmic trajectory LLO\_to\_NRHO\_Cosmic\_OPT.bsp, b. See Use Case 2.0 documentation for additional information on bsp2visualCop.py applications

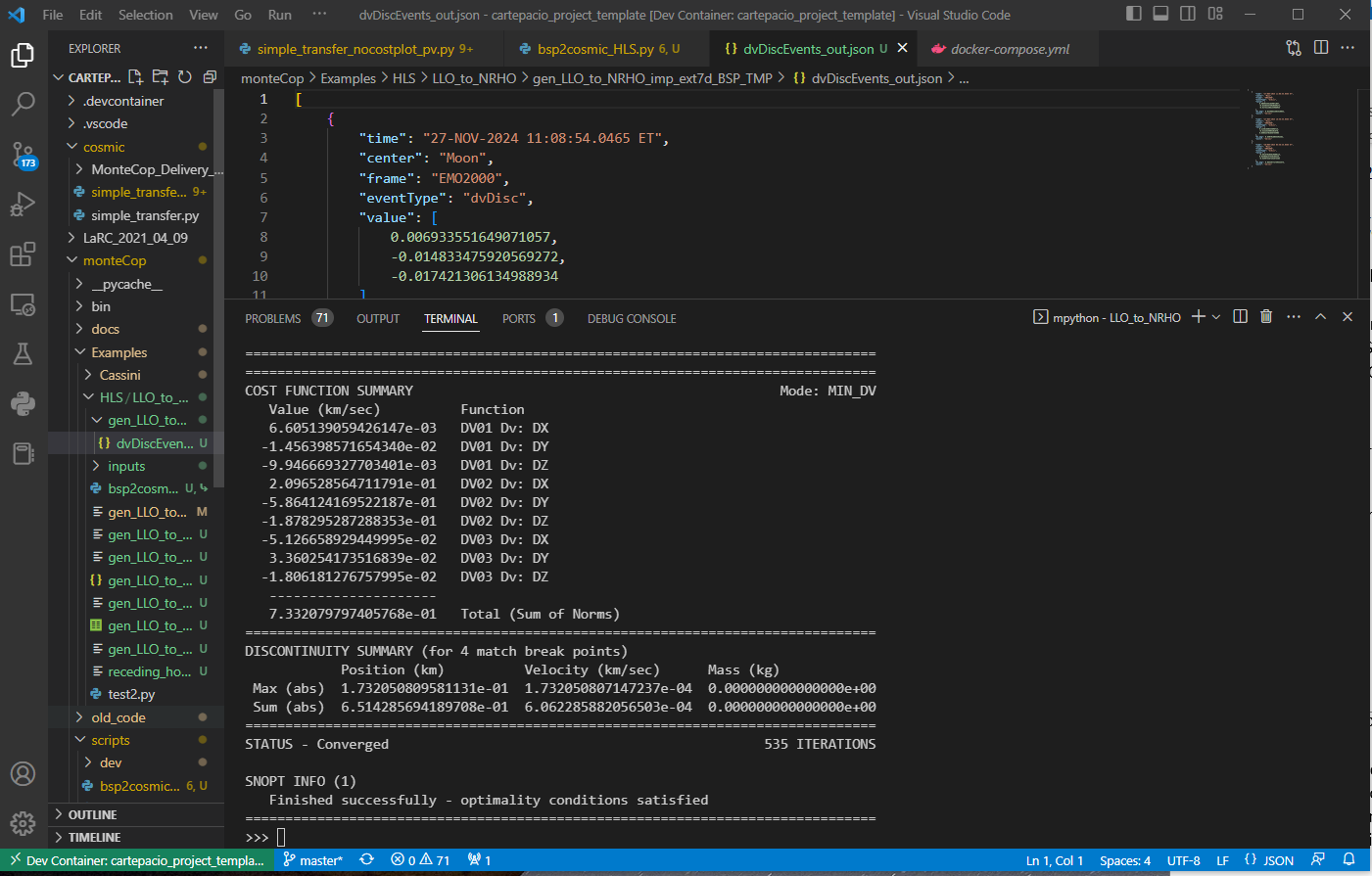
>> bsp2cosmic.py LLO\_to\_NRHO\_Cosmic\_OPT.bsp -c Moon -bl Moon Earth -sc -30100 -f iau\_body\_fixed -n NRHO\_Cosmic\_OPT\_visCop.ideck (Figure 5).



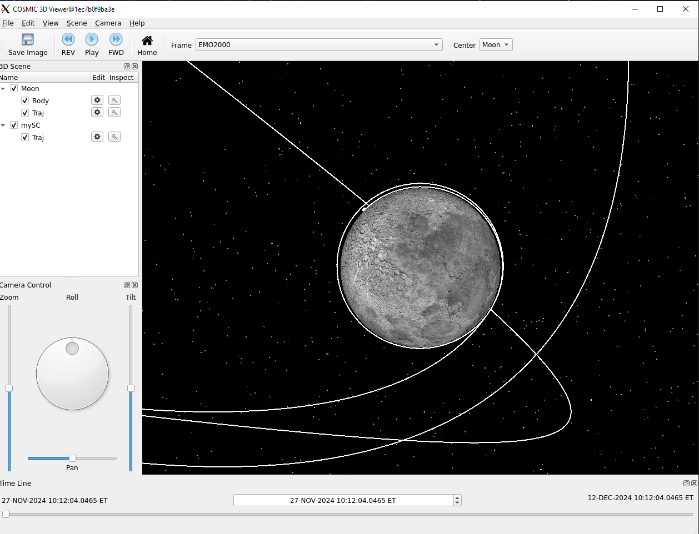
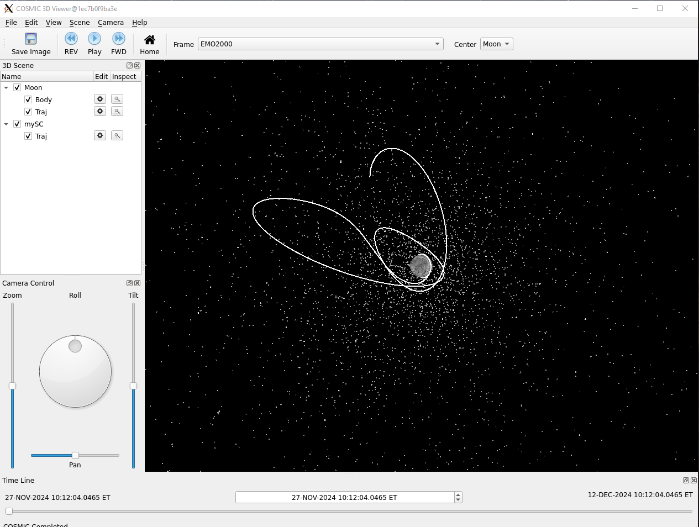
**Figure 1.** HLS ideck trajectory converted to SPK file format in Copernicus.



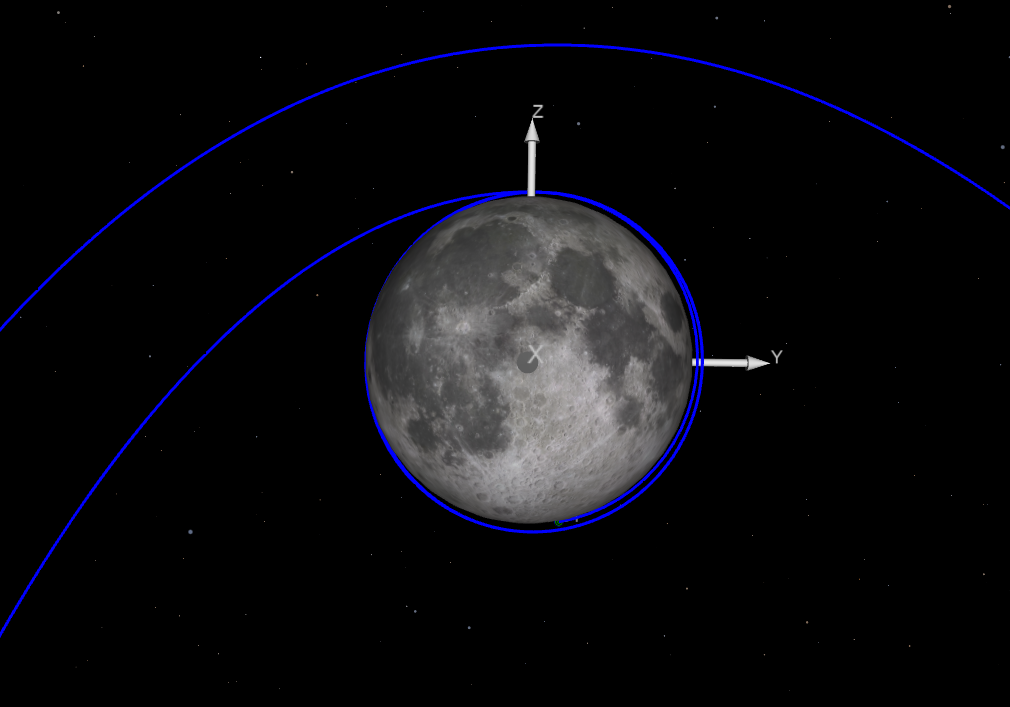
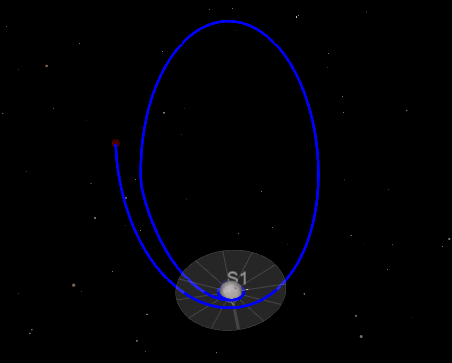
**Figure 2.** Plot of velocity discontinuities (impulsive maneuvers).



**Figure 3.** Successful Convergence for bsp2cosmic.py.



**Figure 4.** Visualization of converged HLS trajectory from MONTE-COSMIC using COSMIC viewer. Visualization frame: J2000 Moon-Centered

**** 

**Figure 4.** Visualization of converged HLS trajectory from MONTE-COSMIC using Copernicus as a trajectory visualization tool. Frame: Moon body fixed

**Overall Summary:** Testing of the NESC Interoperability Tool on another computer platform and environment was successful. In particular, the tester completed the exercise outlined above and provided the expected output planned by the test developer. Minor configuration updates were made based on the tester’s computer environment.

**Acronyms:**

**COSMIC** - Computer Optimization System for Multiple Independent Courses; COSMIC is the MONTE-based replacement software for the legacy Mission Design tool CATO (Computer Algorithm for Trajectory Optimization).

**LLO-to-NRHO** - Lunar Low Orbit to Near Rectilinear Halo Orbit.

MDNav - Mission Design and Navigation.

**MONTE** - Mission Analysis, Operations, and Navigation Toolkit Environment; MONTE is an astrodynamics Python library. MONTE is JPL's signature astrodynamics computing platform and supports all phases of space mission development from early space design and analysis through flight navigation services.

**NAIF** - Navigation and Ancillary Information Facility at JPL.

**NESC** - NASA Engineering and Safety Center.

**SNOPT -** Sparse Nonlinear OPTimizer.  A commercial software package for solving large-scale optimization problems used by MONTE.

**DBLSE** - Double Precision Bounded Least Squares (Optimizer) with equality constraints. Optimizer used by MONTE.

**SPICE** - Spacecraft Planet Instrument C-Matrix Events; provides geometric and some other ancillary information needed to recover the full value of science instrument data, including correlation of individual instrument data sets with data from other instruments on the same or other spacecraft.

**SPK** - SPICE Planetary Kernel; the common file format for NAIF’s ephemeris data.

**References:**

1. R. RESTREPO, *Trajectory Reverse Engineering: A General Strategy for Transferring Trajectories Between Flight Mechanics Tools*, 33th AAS/AIAA Space Flight Mechanics Meeting, Paper AAS-23-312, Austin, TX, January 15-19, 2023.
2. J. Everett, A. Heaton, A. Houin and K. Miller, "*An Integrated Software Architecture for Solar Cruiser Mission Design and Navigation*," 2022 IEEE Aerospace Conference (AERO), Big Sky, MT, USA, 2022, pp. 1-9, doi: 10.1109/AERO53065.2022.9843449.