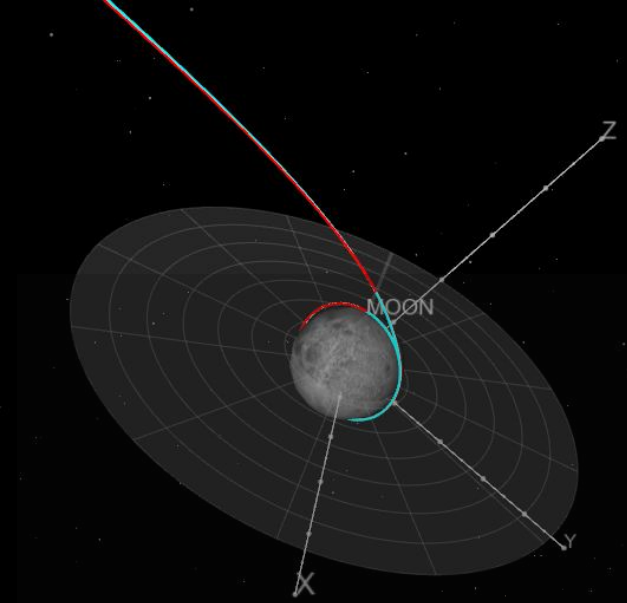
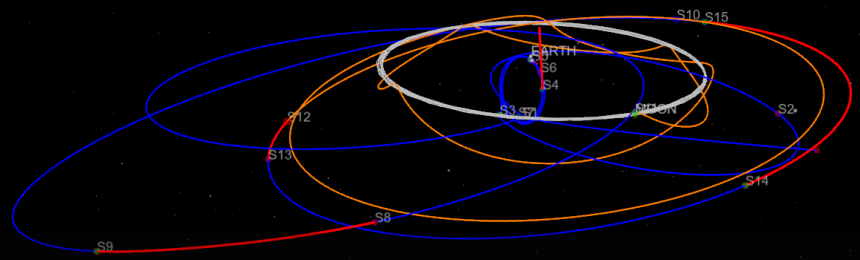
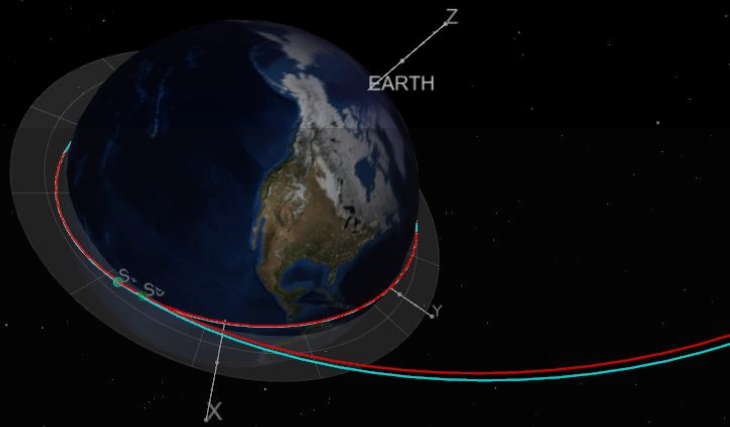






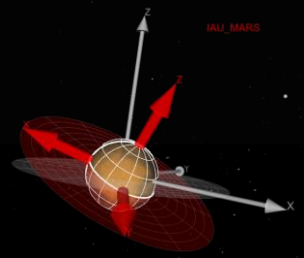
FortranCon 2020

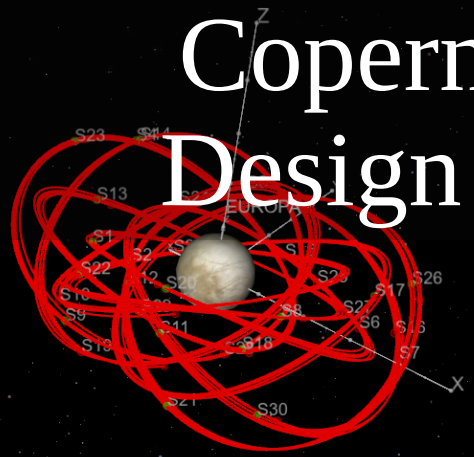
Copernicus Spacecraft Trajectory Design and Optimization Program

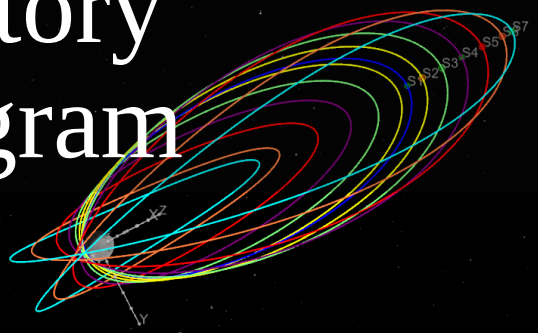
Jacob Williams

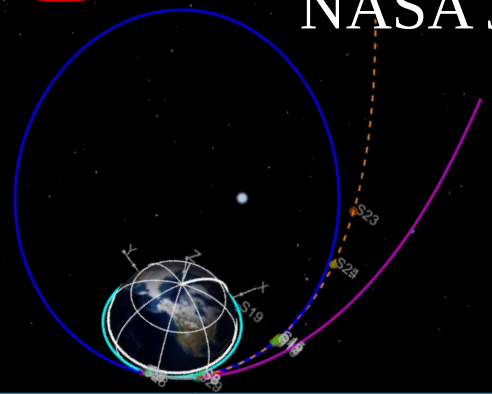
NASA Johnson Space Center

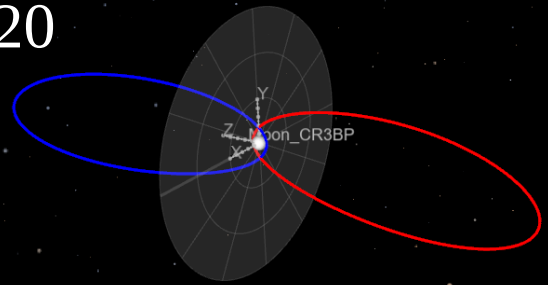
July 2, 2020



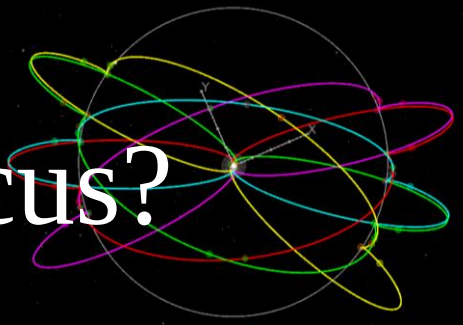


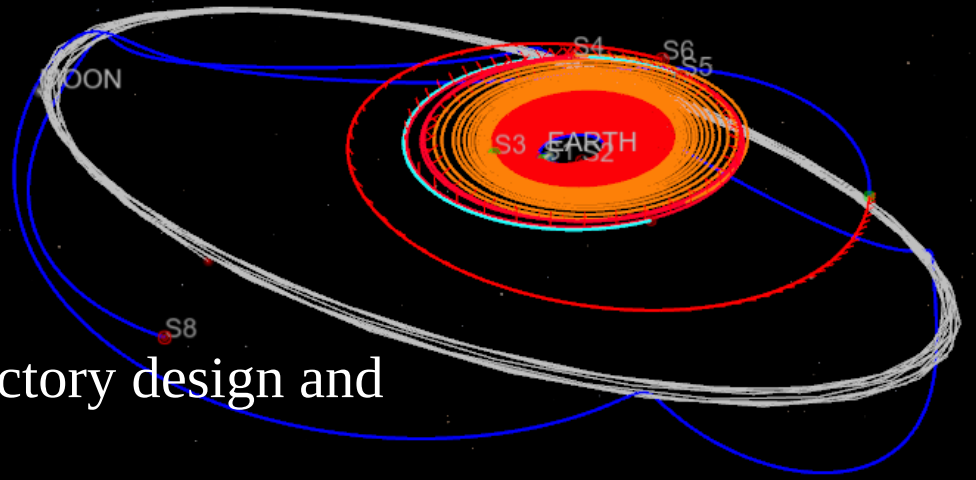




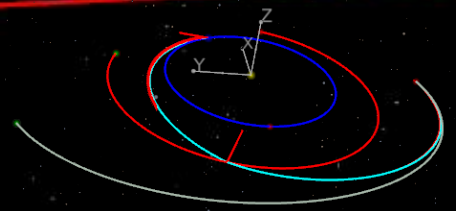


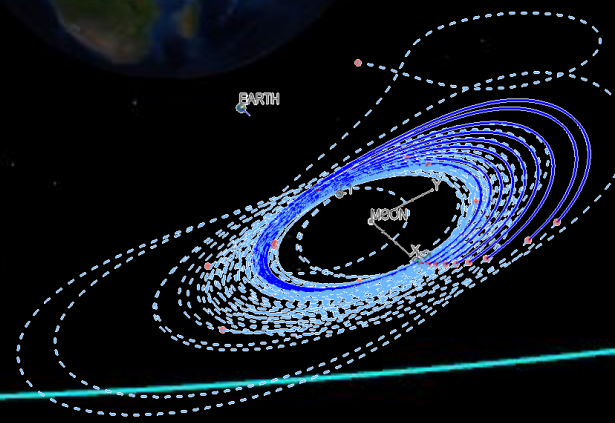
What is Copernicus?





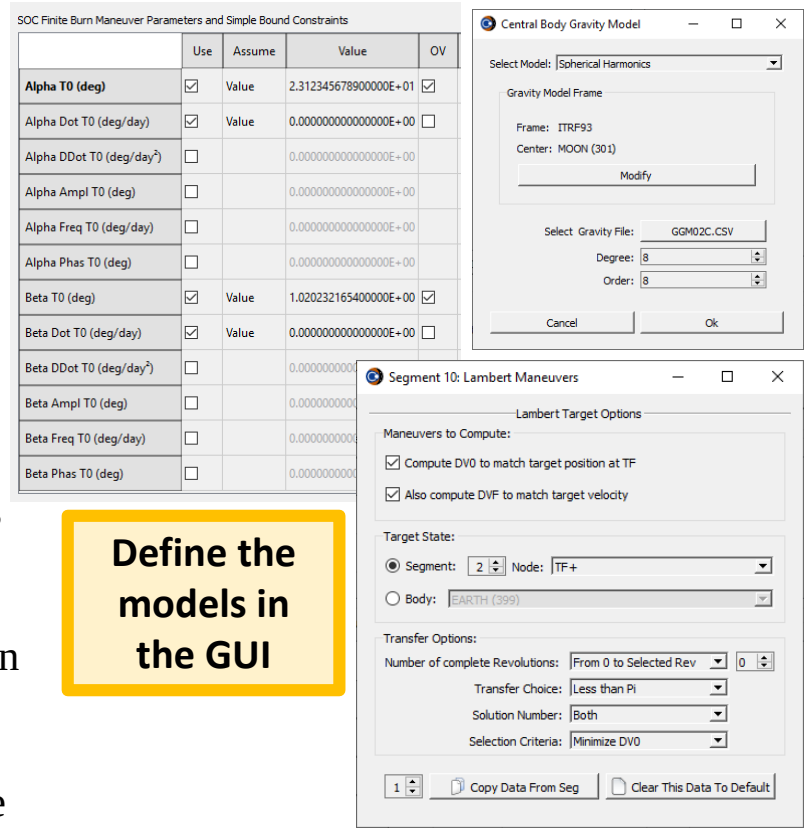
- Copernicus is a generalized, interactive, spacecraft trajectory design and optimization application
- Multiple spacecraft and propulsion systems, integrated GUI and 3D graphics, flexible segment & plugin architecture, selectable mission fidelity (simple to complex)
- Extensive range of missions: impulsive/low/high thrust, multi-body, planet centered/inter-planetary, multi-body transfers/trajectories
- Developed at JSC, and available for use by any NASA employee or U.S. government contractor
- Evolutionary and expandable
- Copernicus can be scaled from a single desktop or laptop computer using the GUI, to computer clusters where no user interaction or graphical feedback are required



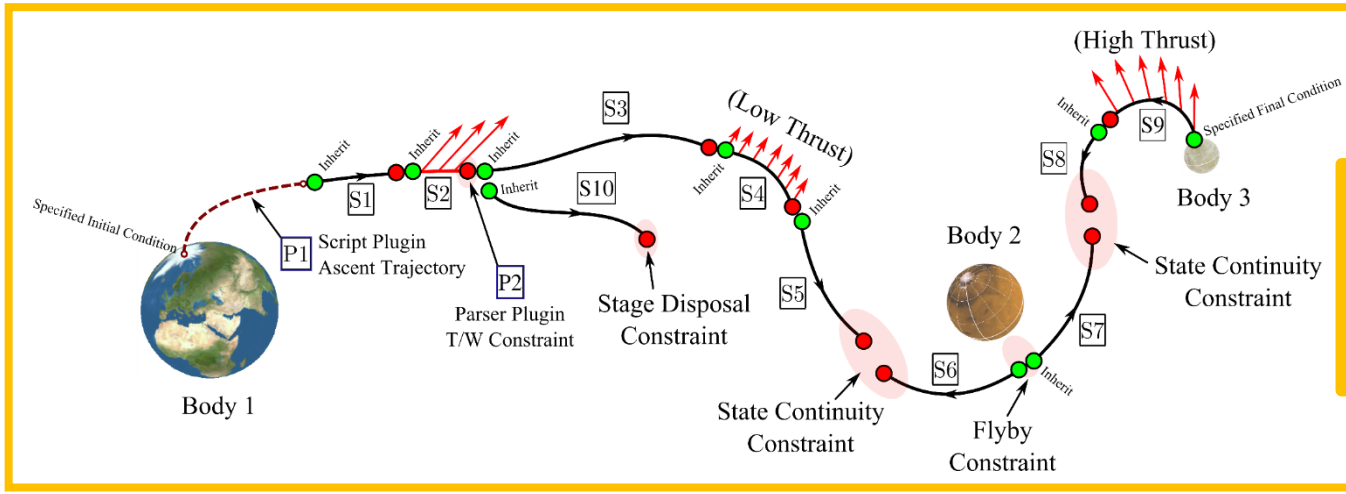


Copernicus Models

- Low and high fidelity models in the same tool
  - Mission Segments
  - Integrators/Propagators
  - Optimal Control Theory
  - Parameter Optimization
  - Numerical Differentiation
  - Ephemerides
  - Reference Frames
  - Finite Burn Engine Models
  - Finite Burn Maneuver Models
  - Impulsive Maneuvers
  - Lambert Targeting
  - State Parameterizations
  - Maneuver Parameterizations
  - Gravity Assists
  - Halo Orbits
  - Gravity Models
  - Interpolation
  - 3D Visualization
  - Data Output
  - Plugin Interface
  - Scripting Interface
- Interactive, flexible architecture
- More than one way to design/optimize a mission



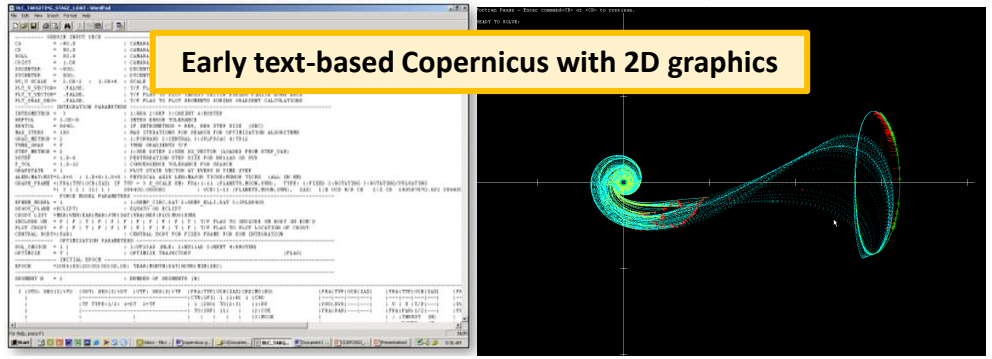
Define the models in the GUI

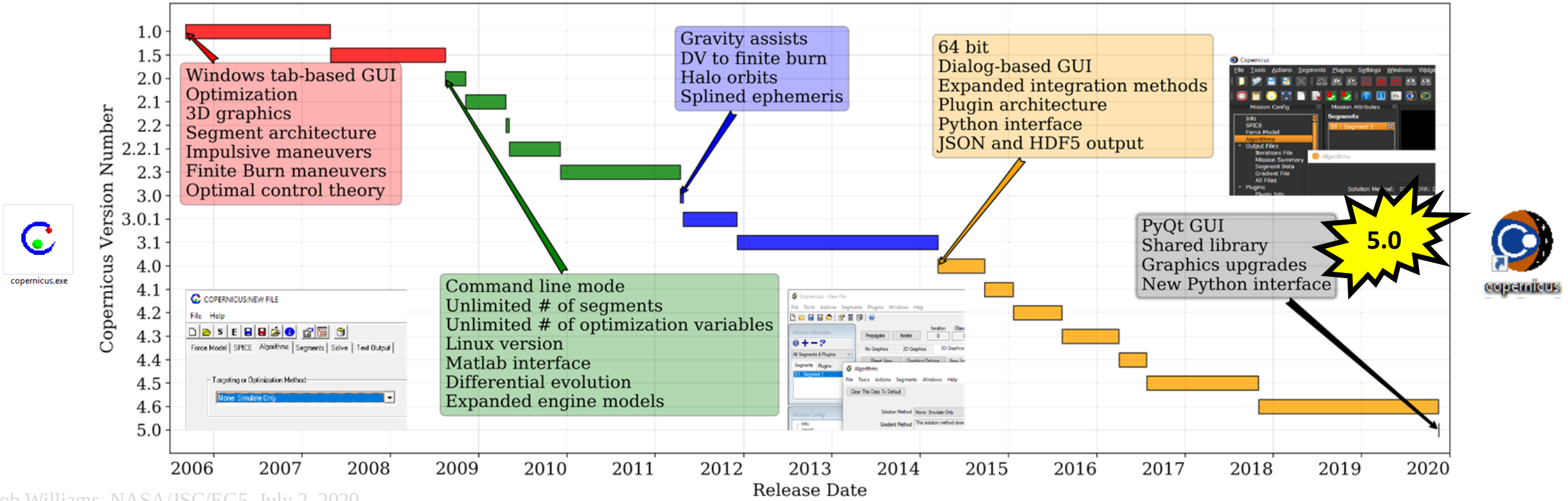


Many classes of problems can be modeled with the segment concept.

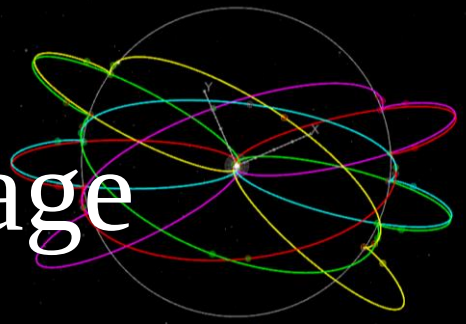
Copernicus History

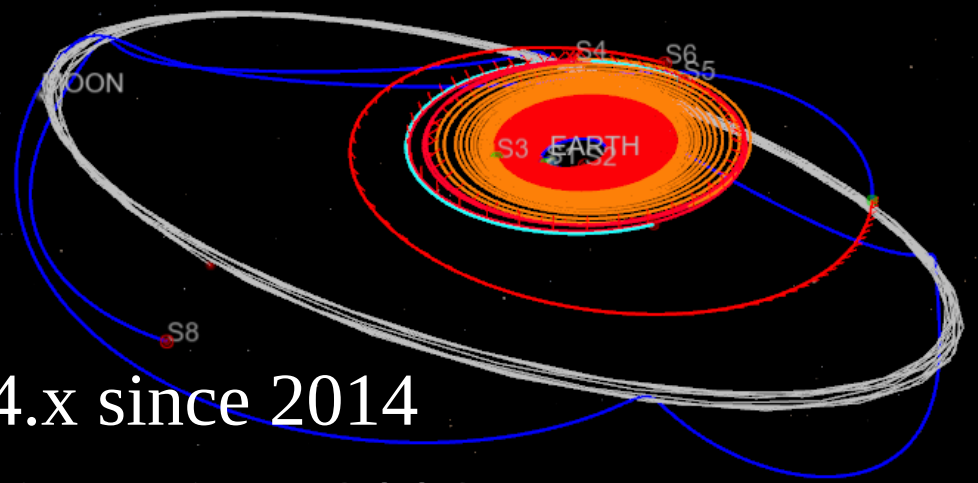
- Copernicus has been continuously developed for nearly 20 years
- 5 major releases (5.0 is one of the most significant updates we have done)
- 2001-2002: University of Texas at Austin (UT)
- 2003-2006: JSC/UT Collaboration
- **First official release: March 2006**
- 2007-Present: Primary Development at JSC



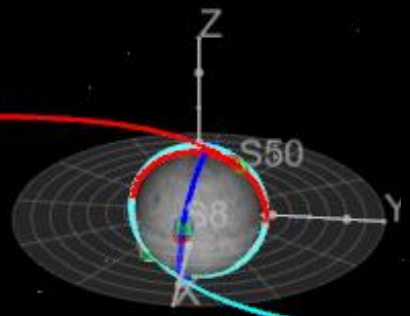


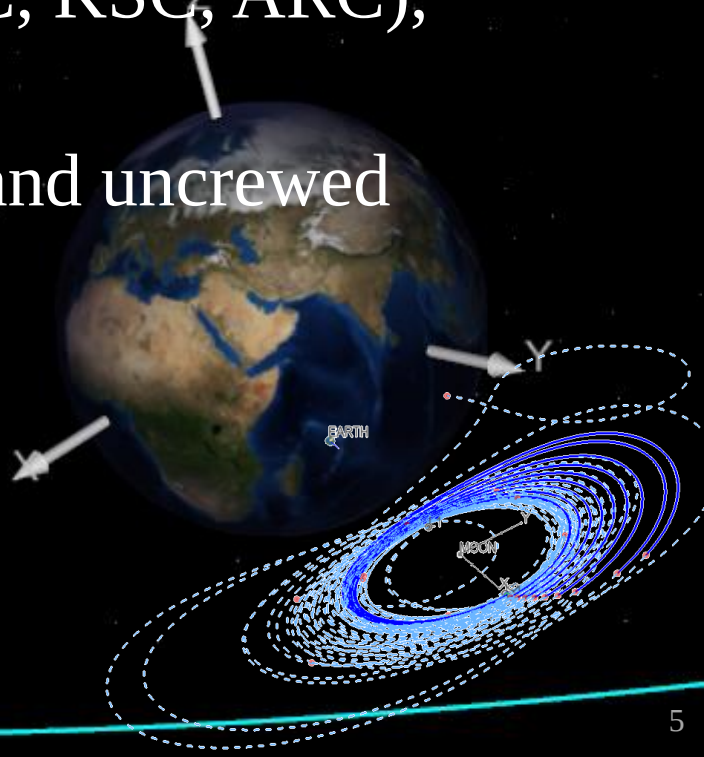
Copernicus Usage





- About 190 licenses issued for Copernicus 4.x since 2014
- About 300 licenses issued in all for all versions since 2006
- Used at NASA (JSC, MSFC, GRC, GSFC, JPL, LaRC, KSC, ARC), numerous government contractors, and universities
- Copernicus has become a workhorse tool for crewed and uncrewed spaceflight mission design at JSC
- What are we using it for?...

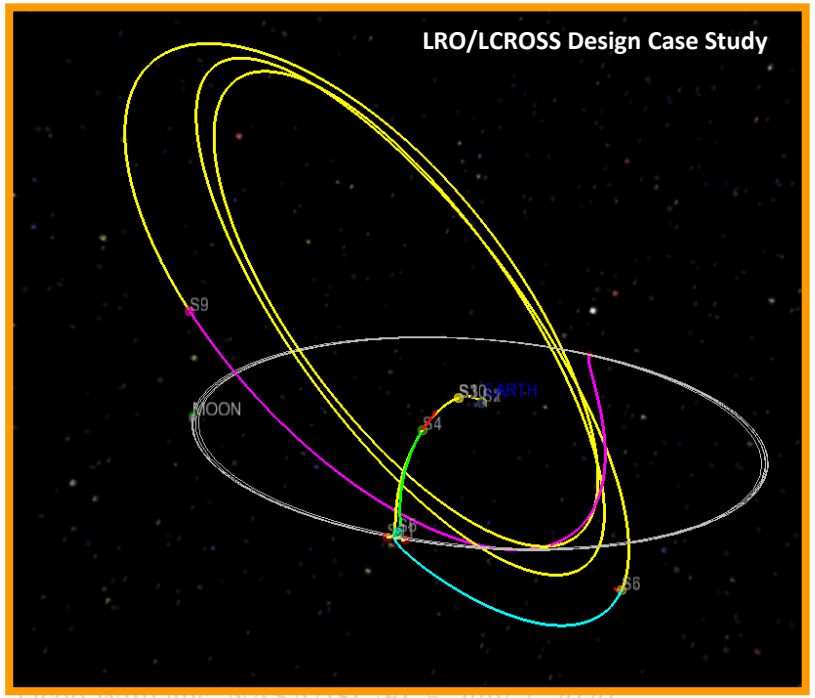


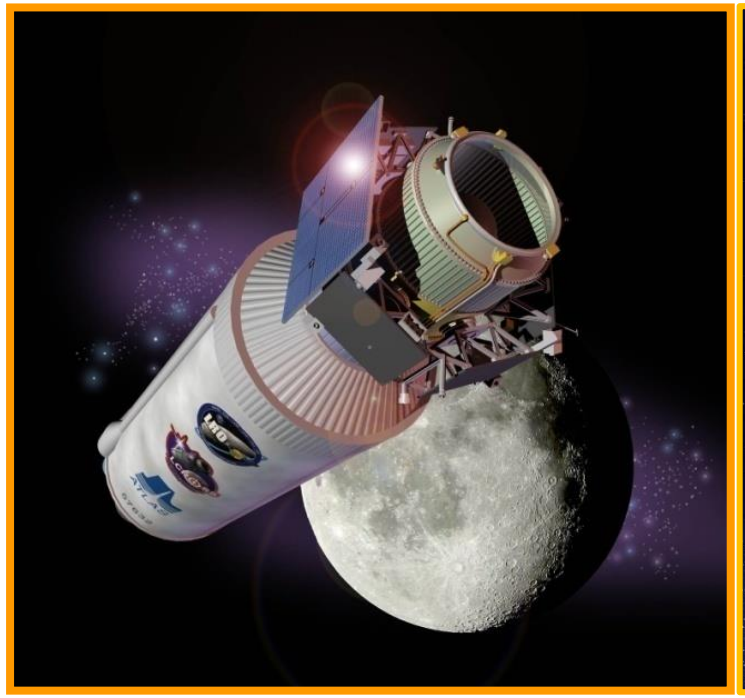


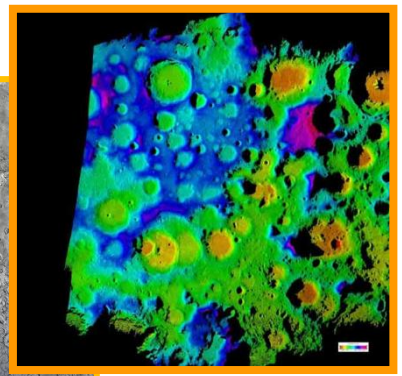
LCROSS Mission

(Lunar Crater Observation and Sensing Satellite)

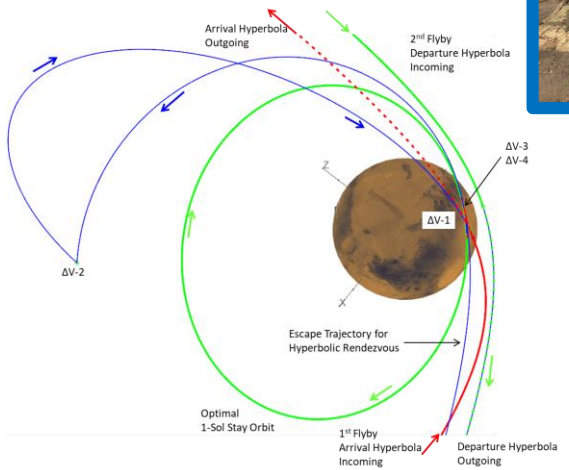
- Copernicus was used to construct hundreds of optimal Earth-Lunar flyby-to-Lunar impact trajectories including the separation phase from the original LRO trajectory which was bound for Lunar orbit
- Also used post-launch to examine under/over burns en route
- LCROSS and its Centaur stage impacted the Moon on Oct. 9, 2009

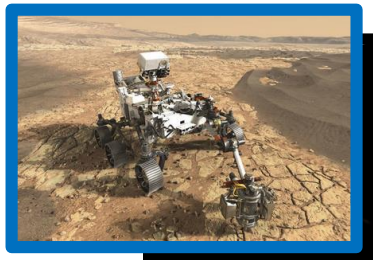




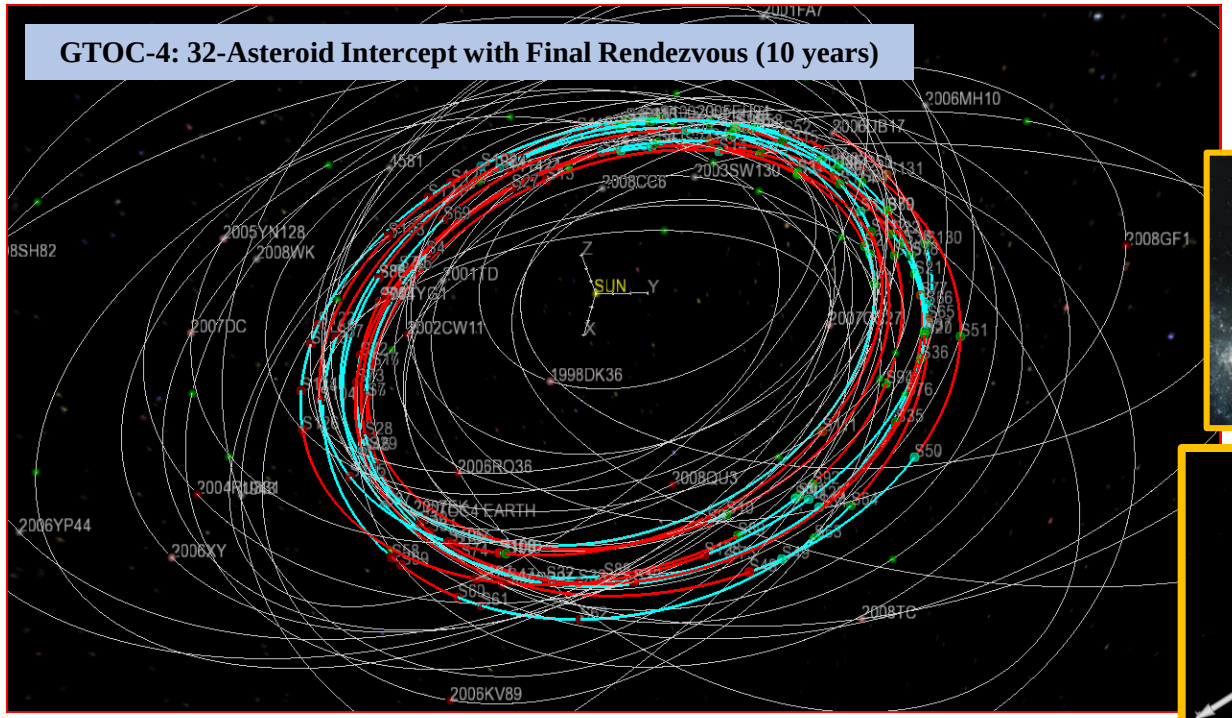


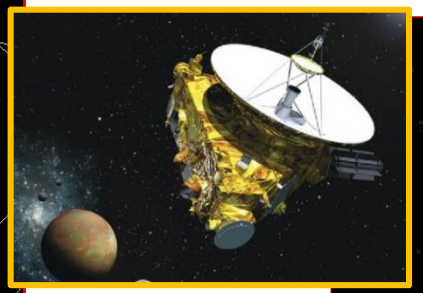
Asteroids, Mars & Outer Planet Studies

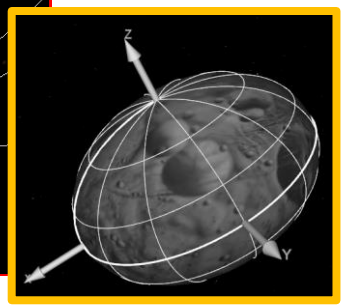


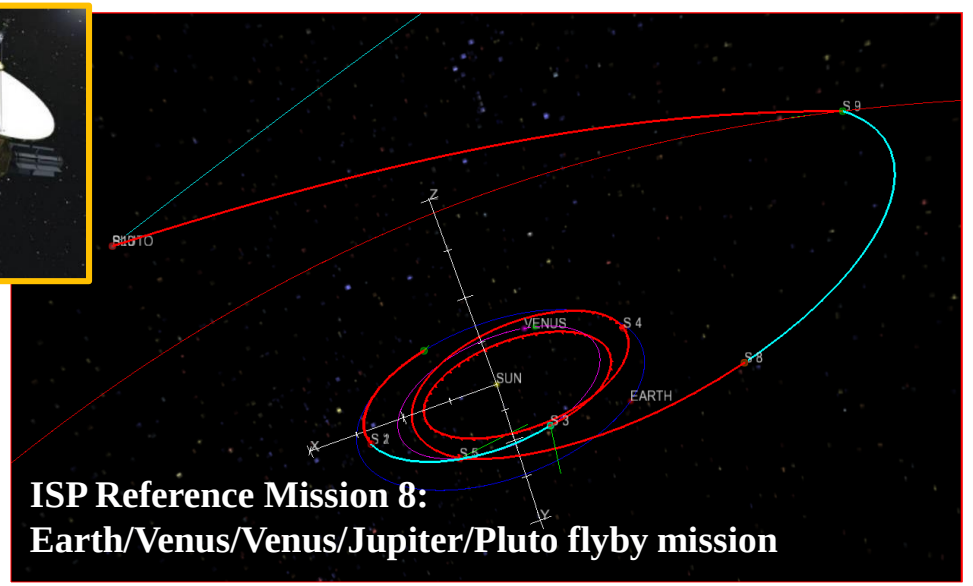


ISP Reference Mission 12: Mars Sample Return Mission [Using low thrust engine and optimal control theory]



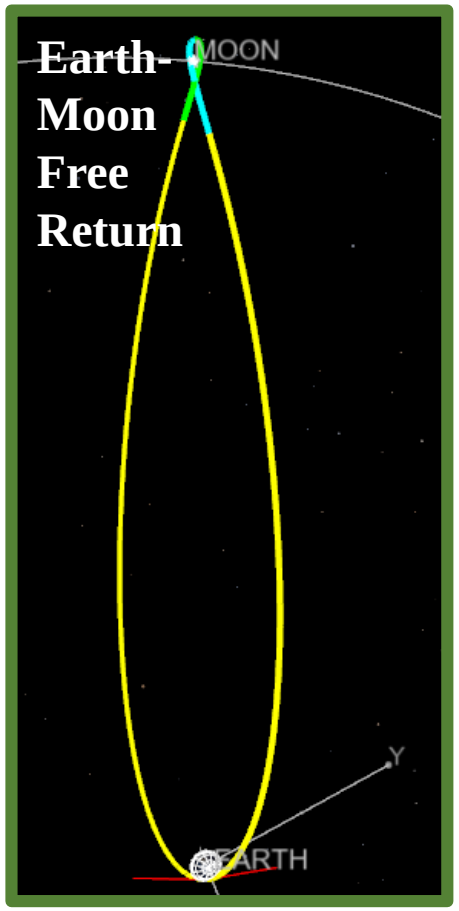


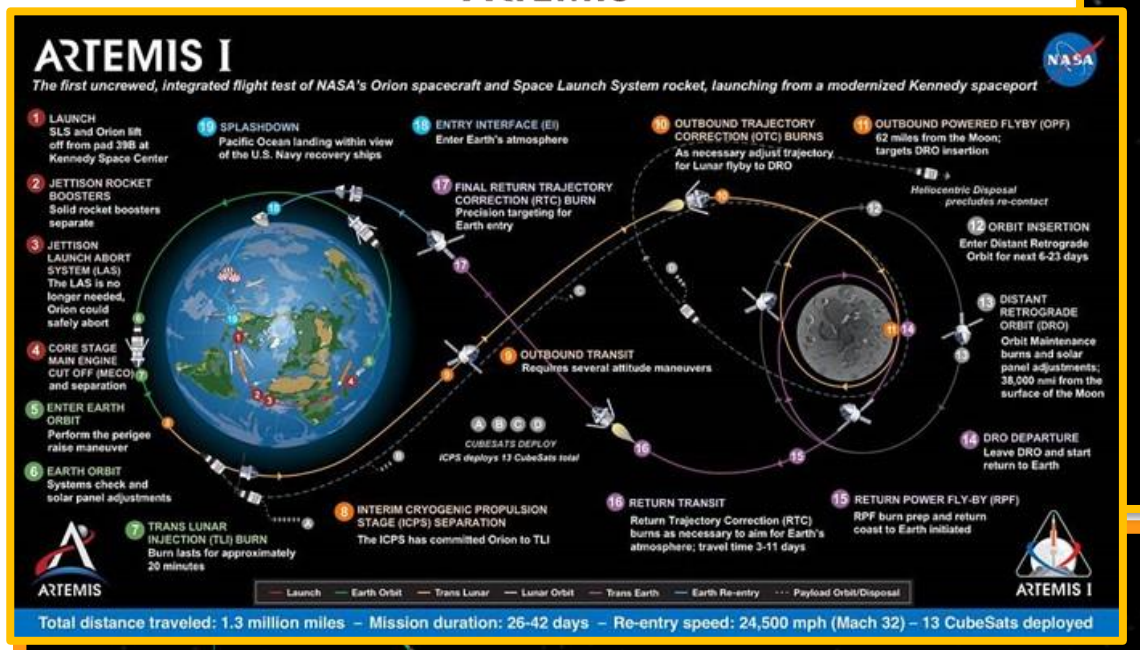


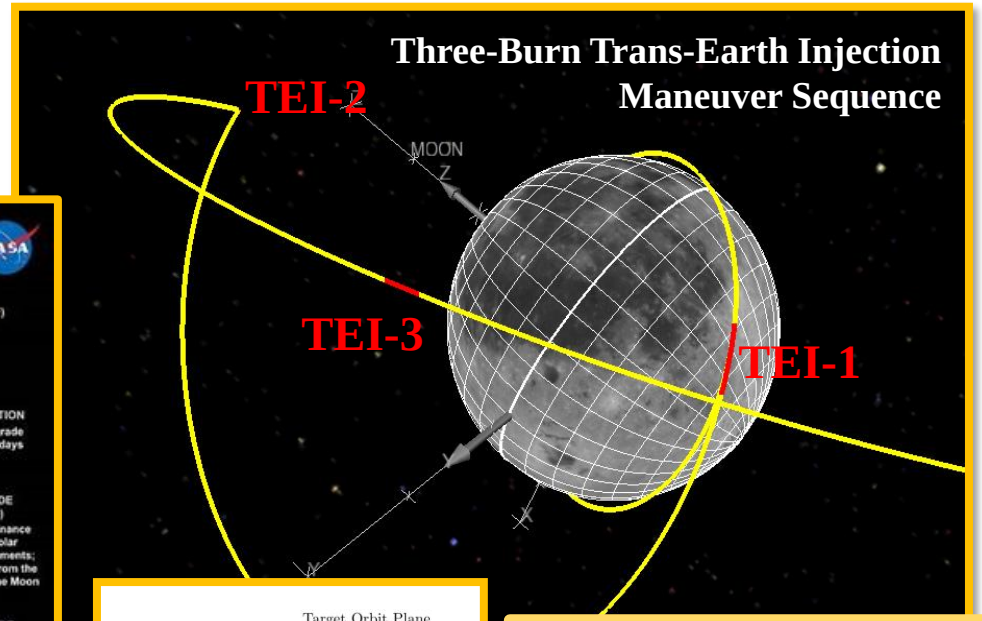


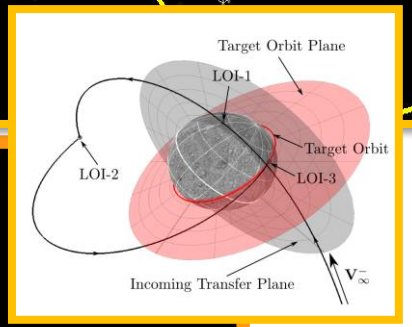
ISP Reference Mission 8: Earth/Venus/Venus/Jupiter/Pluto flyby mission

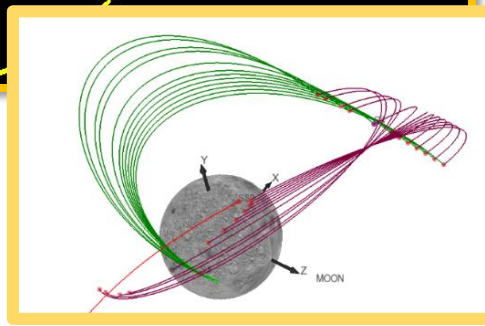
Lunar Missions

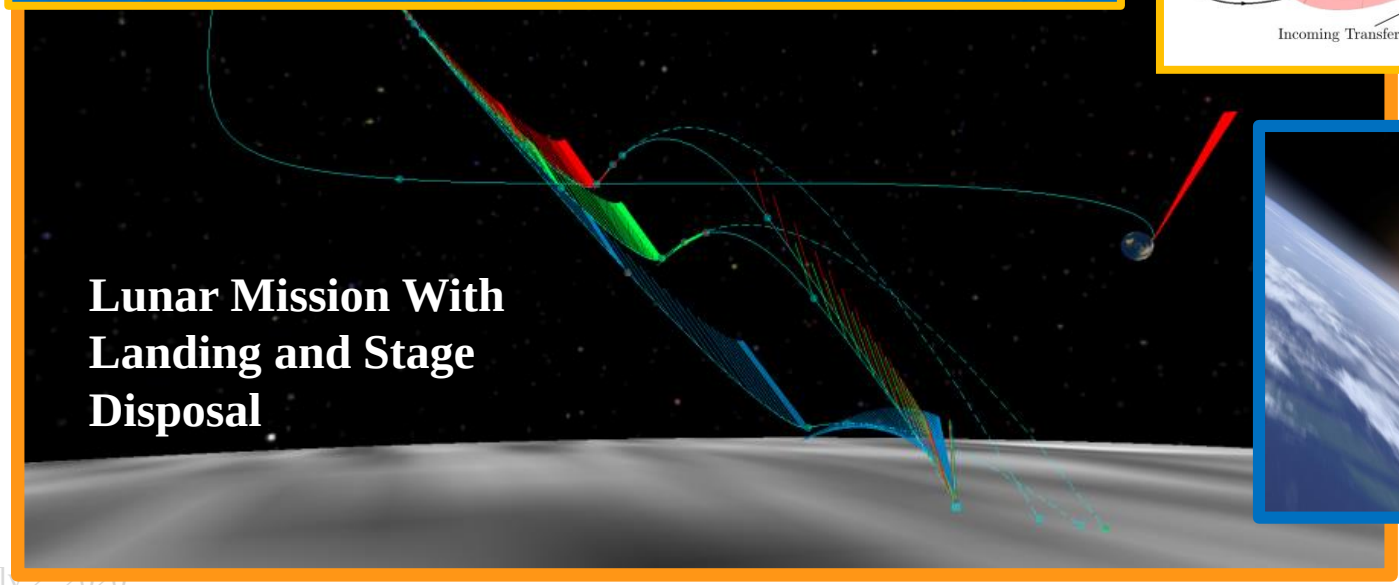






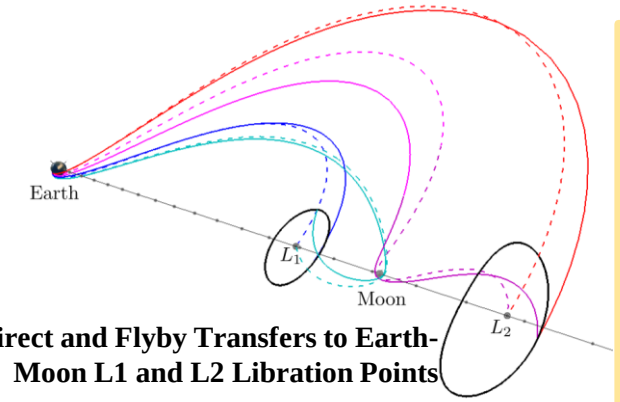




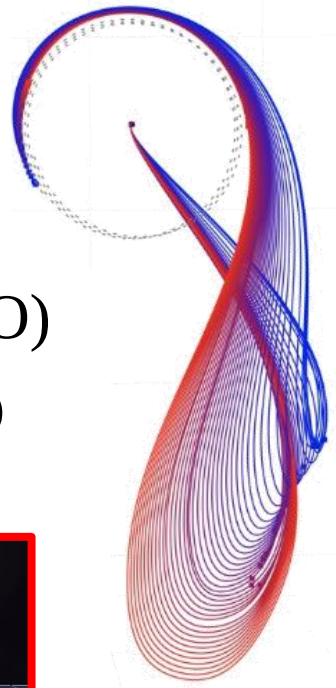


Three-Body, Halo Orbits, DRO, NRHO, etc.

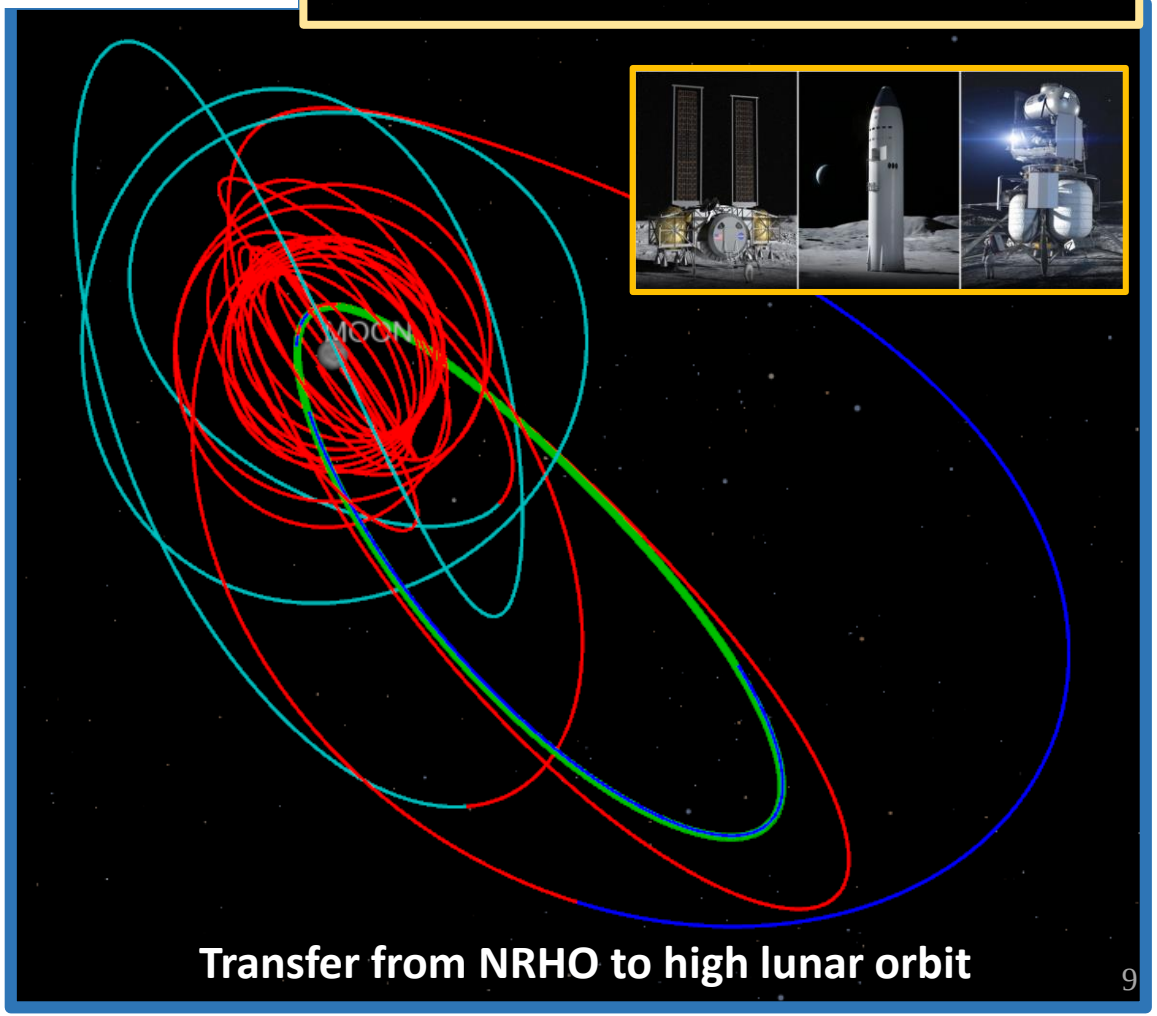
- Halo Orbits & Weak Stability Boundary/Ballistic Capture
- Artemis I, II, III
- Human Landing System (HLS)
- Near-Rectilinear Halo Orbits (NRHO)
- Deep Space Gateway / PPE / HALO

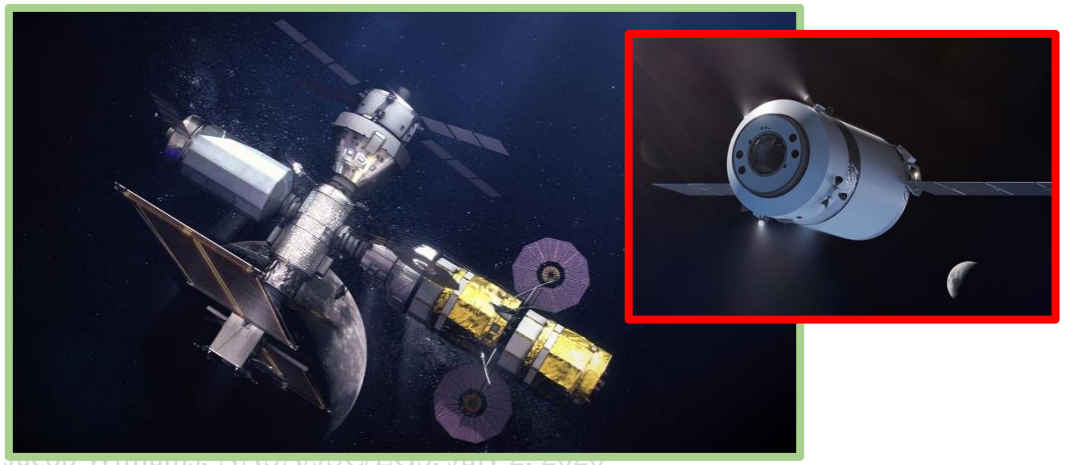


Direct and Flyby Transfers to Earth-Moon L1 and L2 Libration Points



Ballistic Transfers from Earth to NRHO

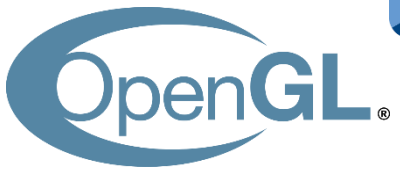


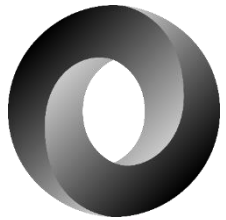


Copernicus Software Development

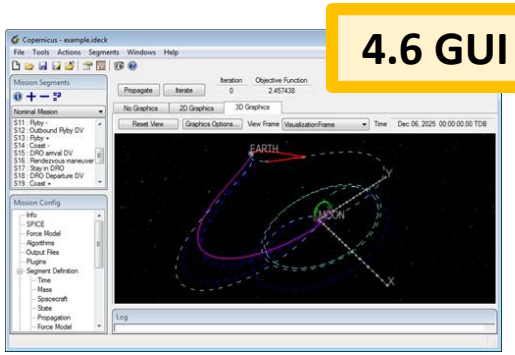
- Copernicus started in 2001 as Fortran 77/90, Compaq Visual Fortran (Windows only)
- Transitioned to Intel Fortran circa 2007
- Cross platform: Windows, macOS, Linux
- Continuous improvement & modernization, keep up with the latest Fortran standards and tools
  - If Intel supports a feature, we will use it.
  - Copernicus is never finished.
- Some of the tools/libraries we are currently using: Intel Fortran 2019, CMake 3.12.3, Git, Python 3.7, Anaconda 2019.07, MS Visual Studio 2017, VS Code, Qt 5.12.3, OpenSceneGraph 3.6.2, HDF5 1.10.4



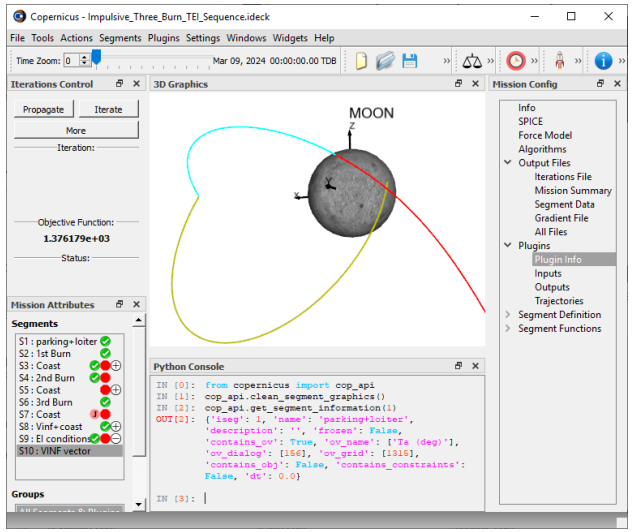


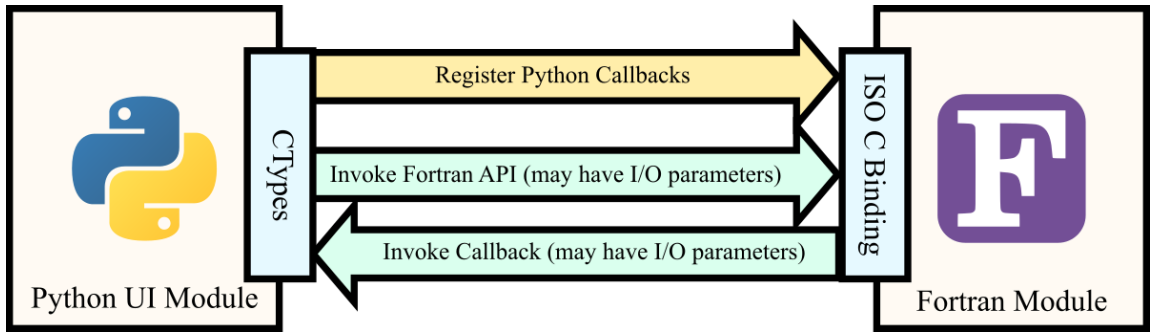



Software Architecture

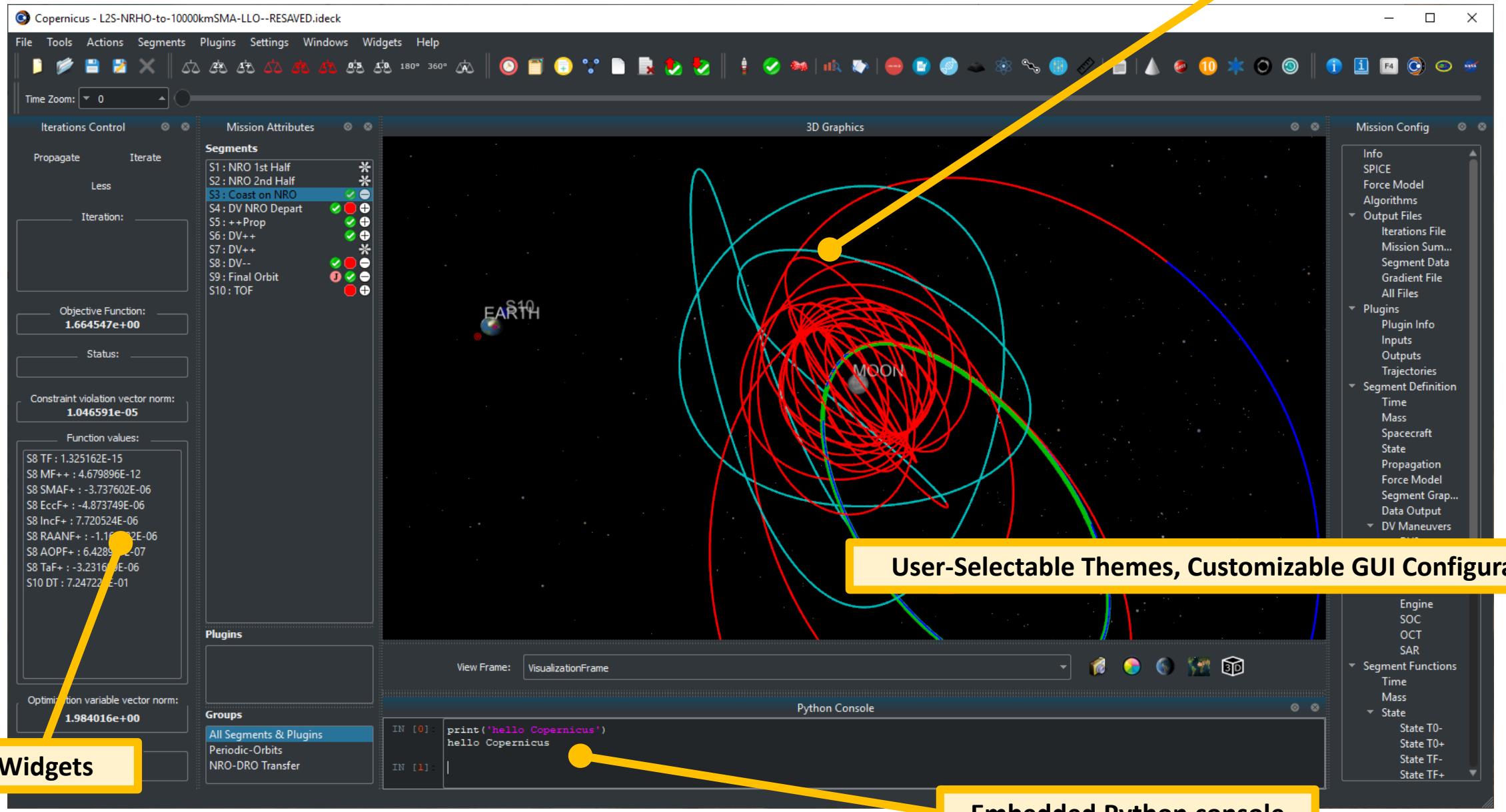


- Formerly, the entire program was Fortran (used the Winteracter library for the GUI)
- Significant refactoring as Fortran 2003+ became available
  - Mostly standard Fortran with some Intel extensions and MKL routines
  - 327 modules, about 218,505 lines of code (not counting 3rd party F77)
- Copernicus is now (v5.0) implemented as a shared library that is called from a Python GUI
  - Core Copernicus code (Fortran) and the GUI (Python) are now completely decoupled
  - Extensive use of C Interoperability – Callbacks to/from Fortran & Python
  - The Copernicus shared library can also be used by other scripts, tools, etc.
- Interactive 3D graphics using OpenSceneGraph and OpenFrames libraries – Fortran interface to C++ code
- Dynamic equations/functions/models user-input:
  - Custom internal function parser written in Fortran
  - User created shared library plugins (DLLs)
  - Eventually: Python code executed by callback.





Copernicus GUI: Example



Interactive graphics: Iterate, pan, zoom, rotate

User-Selectable Themes, Customizable GUI Configuration

Widgets

Embedded Python console

3D Party Fortran Components

- SNOPT (optimization)
  - Fortran 77, Waiting for “modern” version
- SPICELIB (JPL, solar system ephemeris, geometry, and time)
  - Fortran 77, Recently announced rewriting it in C++
- JSON-Fortran (configuration files, data output, data exchange)
- Bspline-Fortran (interpolation)
- SLSQP (optimization)
- Hairer (ODE IVP)
- FLINT (ODE IVP)
- And more!

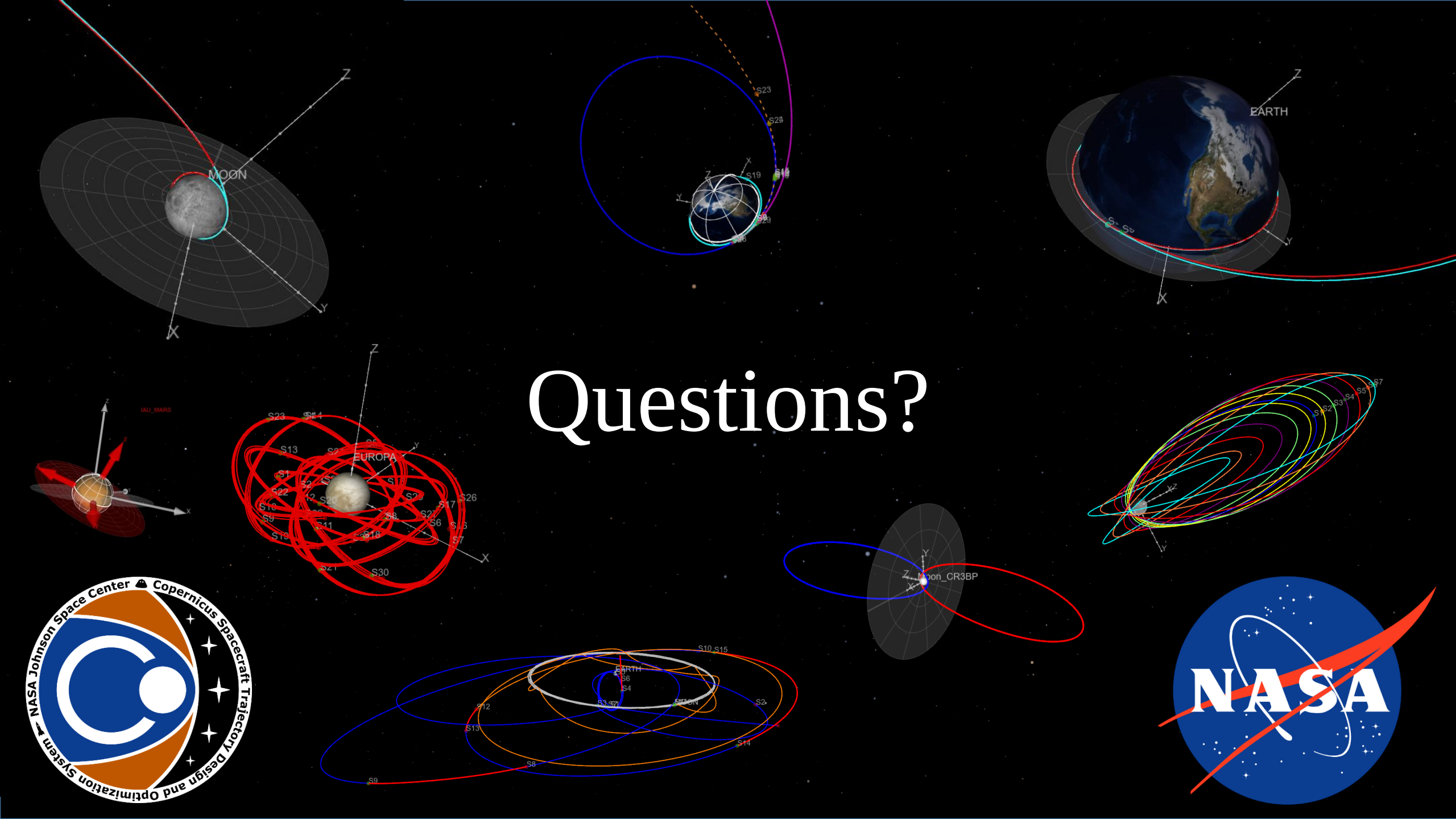
Open source

Conclusions

- Copernicus is an example of an actively developed modern Fortran application with a wide user base at NASA
- Critical software tool for JSC & NASA-wide
- Greatly expanded capabilities and use cases with recent (v5.0) Python GUI and scripting integration
- Copernicus Fortran wish list
  - Better ecosystem & cross-platform tooling, linting, etc.
  - Generic programming, differentiable programming
  - Exception handling
  - Built-in modern string class
  - Dynamic, interactive capability (think Python, Julia, Jupyter)

References

- J. Williams, A. Kamath, R. Eckman, G. Condon, R. Mathur, D. Davis, "**Copernicus 5.0: Latest Advances in JSC's Spacecraft Trajectory Optimization and Design System**", 2019 AAS/AIAA Astrodynamics Specialist Conference, Portland, ME, August 11-15, 2019, AAS 19-719
- J. Williams, R. D. Falck, and I. B. Beekman. "**Application of Modern Fortran to Spacecraft Trajectory Design and Optimization**", 2018 Space Flight Mechanics Meeting, AIAA SciTech Forum, (AIAA 2018-1451)
- J. Williams, "**A New Plugin Architecture for the Copernicus Spacecraft Trajectory Optimization Program**", AAS/AIAA Astrodynamics Specialist Conference, Vail, Colorado, Aug. 2015. AAS 15-606.
- J. Williams, J. S. Senent, D. E. Lee, "**Recent Improvements to the Copernicus Trajectory Design and Optimization System**", Advances in Astronautical Sciences, 2012.
- J. Williams, J. S. Senent, C. Ocampo, R. Mathur, E. C. Davis. "**Overview and Software Architecture of the Copernicus Trajectory Design and Optimization System**", 4th International Conference on Astrodynamics Tools and Techniques, 3-6 May 2010, Madrid, Spain
- C. A. Ocampo, J. S. Senent, J. Williams, "**Theoretical Foundation of Copernicus: A Unified System for Trajectory Design and Optimization**", 4th International Conference on Astrodynamics Tools and Techniques, May 2010.
- R. Mathur, C. A. Ocampo, "**An Architecture for Incorporating Interactive Visualizations into Scientific Simulations**", Advances in the Astronautical Sciences, Feb. 2007.
- C. A. Ocampo, J. Senent, "**The Design and Development of Copernicus: A Comprehensive Trajectory Design and Optimization System**", Proceedings of the International Astronautical Congress, 2006. IAC-06-C1.4.04.
- Kos, Larry D. et al. "**Overview of the Development for a Suite of Low-Thrust Trajectory Analysis Tools**", AIAA AAS Astrodynamics Specialist Conference, August, 2006.
- C. A. Ocampo, "**Finite Burn Maneuver Modeling for a Generalized Spacecraft Trajectory Design and Optimization System**", Annals of the New York Academy of Science, May 2004.
- C. A. Ocampo, "**An Architecture for a Generalized Trajectory Design and Optimization System**", Proceedings of the International Conference on Libration Points and Missions, June, 2002.


MOON
EARTH
EUROPA
IAU_MARS
Moon_CR3BP
NASA Johnson Space Center
Copernicus Spacecraft Trajectory Design and Optimization System
NASA

Questions?