

# REVISITING TRAJECTORY DESIGN WITH STK ASTROGATOR PART 1

**Cody Short\***, **Pradipto Ghosh<sup>†</sup>**  
and **Austin Claybrook<sup>‡</sup>**

Despite more than six decades of ongoing presence in space, the process of spacecraft trajectory design remains challenging. As orbital regimes become more crowded and attainable from an expanding industry, new challenges arise. Research efforts to better understand previously unexploited strategies and drive innovation in the face of new constraints continue to increase. At the same time, all of these research and design efforts are restricted by the complication of a “lab” where designers cannot physically interact with their designs; they are confined to a virtual workbench. Indeed, it is not uncommon for trajectory designers to spend much of their time creating tools for that workspace. This situation is strange as most industries that involve design and analysis typically do not build their own tools. The Systems Tool Kit (STK) Astrogator module from Analytical Graphics, Inc. (AGI) is one such tool intended to improve the trajectory design and analysis process, both for the engineer and for the solution. Astrogator’s evolving development is the subject of this paper.

## INTRODUCTION

As missions progress through the phases of the space mission lifecycle, various analyses are required. From the earliest stages of determining feasibility, to operational maneuver planning, execution and processing, designers and analysts need information. Various means are available to perform the necessary simulations and produce the required data products. The driving consideration for the associated tool is generally, “Can it do what I need?” This question is often followed with, “Can it be made to do what I need?” While a resounding affirmative to the former question is always desirable (and generally most cost effective<sup>1</sup>), unique concepts often immediately promote to seeking the answer to the latter. The history of Astrogator reads as a clear reflection of incremental development and improvement to incorporate capabilities in response to both questions. One of the goals of its continuing development is to make it do what is needed before a designer may even realize that such a need exists.

Analytical Graphics, Inc. (AGI) produces the Astrogator trajectory design and maneuver planning module for its flagship Systems Tool Kit (STK) software. Astrogator shares heritage with other trajectory design software, its roots tracing back to the commercialized Swingby tool and associated precursors. Development of Swingby began in 1989 by the former Computer Sciences Corporation (CSC) for NASA to migrate trajectory design efforts to the personal computer. Swingby was marketed and sold by CSC in partnership with AGI as Navigator, and Navigator eventually transferred

---

\*Astrodynamics Software Engineer, Analytical Graphics, Inc., 220 Valley Creek Blvd., Exton, PA 19341-2380.

<sup>†</sup>Senior Computational Astrodynamicist, Analytical Graphics, Inc., 220 Valley Creek Blvd., Exton, PA 19341-2380.

<sup>‡</sup>Systems Engineer, Analytical Graphics, Inc., 220 Valley Creek Blvd., Exton, PA 19341-2380.

wholly to AGI. Renewed development efforts at AGI resulted in the current incarnation of the software, Astrogator. Vestiges of Astrogator’s precursors can be observed in other tools, and similar paradigms are observed throughout various space mission design software packages. This year, 2019, marks the 30th anniversary of the shared ancestry initiated with Swingby.

Since its acquisition by AGI, redevelopment and rebranding,\* Astrogator entered into cycles of research and development both independently and as part of STK. Currently, Astrogator is experiencing a new such cycle. Its ongoing evolution is the topic of this paper. As with any evolving entity, future enhancements, revisions and modernizations are expected; consequently, this paper is denoted Part 1 in anticipation of additional installments. This paper also represents an extension of previous, similar works by Carrico et al.<sup>3,4</sup> as well as Berry et al.<sup>5-7</sup> The current development team for Astrogator has elicited input from numerous interested parties, and transformed that input into an ongoing development roadmap. Items of interest include integrated tools to define initial low-fidelity designs, improved capabilities to transition solutions through various phases of the mission lifecycle, better support for low-thrust designs, trajectory optimization and quality of life improvements. Each of these concepts has received some attention in the past 2–3 years and continue to be points of focus. Other areas of interest are also being pursued. The various sections of this paper describe the theoretical foundations and practical implementations surrounding many aspects of Astrogator with particular focus on recent development efforts.

Goals of this paper include describing the nature and background of Astrogator, enumerating the various functional aspects of the tool, capturing a snapshot of its current state and providing additional insight into the technical details of particular capabilities. The document is structured into sections as follows: Background, Capabilities, Recent Advancements, Selected Examples, Concluding Remarks and Acknowledgments. Less detail is given for long-standing capabilities that may be better known in the field, and for which extensive documentation exists,<sup>8</sup> in favor of expanded information regarding more recent additions.

## BACKGROUND

Astrogator’s roots lie in a strong lineage of tools with incarnations dating back to 1989.<sup>4</sup> Many elements and algorithms preceded these formal offerings, and others have arisen over time. The tool has been used for analysis, design and operations on missions ranging from LEO to GEO,<sup>4</sup> from the Sun<sup>9</sup> to Ultima Thule,<sup>10,11</sup> and many places in between. Most of these applications faced unique requirements, and software enhancements resulted in response.<sup>5-7</sup> With particular respect to operational capacities, Astrogator has been employed to support numerous programs. The earliest mission utilizing the software was the Wilkinson Microwave Anisotropy Probe (WMAP) mission.<sup>12</sup> Many current missions in various phases of the mission design lifecycle utilize Astrogator in some capacity. Several specific missions for which Astrogator has been used in an operational sense are tabulated in Table 1,<sup>†</sup> and categorized by mission regime.

---

\*The term astrogator is a portmanteau of *astronomical navigator* (or, perhaps, *astronautical navigator*) used by Robert A. Heinlein in the novel *Starman Jones*.<sup>2</sup> The word may refer to a tool used for astrogation or as a functional title (read: divine calling) for a person.

<sup>†</sup>Table 1 is compiled from input of various AGI personnel and in cooperation with AGI partner Space Exploration Engineering, LLC principals Mike Loucks and John Carrico. The authors welcome feedback with respect to errors and omissions. Corrections and/or additions will be reflected in later installments of this paper series.

**Table 1: Missions Utilizing Astrogator During Operations Categorized by Orbital Regime**

	LEO	GEO/GTO, HEO	Lunar, Cislunar, Libration	Deep Space
ORS-1	OCO-2	AsiaSat 3	WMAP Sun-Earth L2	NEAR Asteroid
Chandra	JPSS	Artemis	LRO Moon	CONTOUR Comet
STSS	STS	DSCS III	LCROSS Moon	ISEE-3 Comet
AMOS	GeoEye 1-2	UFO - all	ARTEMIS Earth-Moon L1, L2, Moon	New Horizons Pluto, Ultima Thule
Cygnus	OrbView 2-3	FLTSAT - all	LADEE Moon	Messenger Mercury
Skysat 3-14	Suomi NPP	MUOS - all	DSCOVR Sun-Earth L1	STEREO Heliocentric
Cloudsat	Glory	GEOSat series ascent	SELENE Moon	MAVEN Mars
STPSat-2	KOMP Sat 1-3	COMS	Beresheet Moon	MOM Mars
IKONOS		IBEX Lunar Resonant HEO		OSIRIS-Rex Asteroid
		Van Allen Probes/RBSP HEO		Parker Solar Probe Heliocentric

The role of Astrogator in the spacecraft mission design lifecycle is largely reflected in the trajectory design and analysis phases, although it is also relevant to other stages such as early concept design and selection as well as operations. Astrogator's integration within STK leads to significant synergies that enable trade studies where the environment is modeled accurately to capture many important contributions to the system. For example, inherent system awareness of periods of time when a spacecraft has line-of-sight intervisibility (or access) to other resources will affect other aspects of the mission design. Considerations such as the access example are also critical during operational phases of a spacecraft mission. Further, the integration with STK allows for feedback through graphics and data product reporting.

While the STK system lends some support, Astrogator displays several independent strengths. In turn, Astrogator extends the capabilities of the STK system. Particular strengths of Astrogator are reflected in the principles of modularity and configurability. These principles are implemented in the STK Component Browser and associated component technology.<sup>13</sup> The various components in the browser represent individual aspects of the mission model. A particular component may reflect part of the force model or characteristics associated with a central body for the system. Some components represent calculations, and provide a mechanism for reporting various data, while others interact together to produce a functional unit or even a sequence of segments to define an Astrogator satellite. These capabilities offer extensive “out-of-the-box” options, while also providing plugin points for users to supply their own models when needed. Additional mechanisms exist to execute the simulation. The Mission Control Sequence (MCS) is an interactive environment for designing and configuring the evolution of spacecraft trajectories. The MCS uses various segments and other objects to capture the design—each of these pieces is a *component*.

Often, the interactive desktop environment is employed to construct and configure a particular solution strategy. Indeed, this rapid design architecture accompanied by graphical feedback enables users to quickly arrive at a particular solution and then refine that solution. It is sometimes the case that the design begins with external analysis and a particular trajectory is migrated into Astrogator for refinement, but it is also possible for the design to progress through levels of model fidelity entirely within STK/Astrogator. The interactive environment is well-poised to support both types of operations. While the MCS provides an interactive environment, both internal and external scripting capabilities allow for even greater configurability and automation. Regular tasks that require re-execution of a previously computed Astrogator trajectory under updated conditions are typically automated through a combination of internal scripting and external API wrapping. Astrogator can be operated by scripting in MATLAB, through compiled Microsoft COM programs or native and extended Windows/Linux scripting. Thus, the high-fidelity solutions previously computed in support of some operational goal can be adjusted and recomputed as required. Additionally, large-scale research analyses are supported through the same concepts. This option between operating modes allows for a large breadth of applicability, and some combination of Astrogator's capabilities can be employed to address most astrodynamical problems. A closer look at some of the specific aspects of the tool in terms of their functionality and technical implementation follows.

## **CAPABILITIES**

The Astrogator trajectory design and analysis system is available as an interactive environment supporting rapid design and extensive specificity of configuration. The architecture also supports high-level scripting and automation. Regardless of the particular method of use, the functional aspects of the tool are presented through higher-, intermediate- or lower-level constructs. The general

interface is reflected through the Mission Control Sequence (MCS). In the MCS, individual segments are available to meet particular mission requirements. Intermediate level algorithms and flow control strategies are available on particular segments or other components. The lower-level configuration and control occurs through the component model associated with the Component Browser. While these three aspects may have some distinction, they are heavily interdependent. A particular Astrogator component is perfectly valid by itself, but it typically can't produce meaningful data without execution from the MCS. The MCS requires segments and other components to be executed. The mechanism for this execution is prescribed by the intermediate-level algorithms and control structures. Some elaboration of long-standing Astrogator capabilities follows, and additional details are available in Astrogator documentation.<sup>8</sup>

### **High-level Control: The MCS**

The Astrogator Mission Control Sequence represents the executive layer of the tool. Individual components in the form of MCS segments are configured, often invoking additional subcomponents for particular functions, and strategically arranged to produce the flow for a particular spacecraft sequence. The MCS provides sequence-wide configuration options and mechanisms to author and execute the associated simulation. The functions and interactions between segments combine to produce a spacecraft ephemeris consistent with the constructed sequence. Complex logical operations, event-triggered transitions, and subroutine-like execution patterns are available to meet the needs of a particular application.

The fundamental components of the MCS are *segments*. Associated with each segment is an underlying concept aligned with the trajectory design. For example, a propagate segment reflects the evolution of a spacecraft along a path from some initial state to another state later in time (or from a state later in time to some earlier state when evolved in reverse time). The segments that may be included in a particular trajectory design include variations that enable configuration of initial conditions, control MCS flow or that produce a spacecraft path (i.e., state ephemeris). These segments reflect names that seek to encapsulate their associated functions. The functionality of the segment types is summarized in Table 2, where they are classified by whether they *initialize ephemeris*, *generate ephemeris* or explicitly *control execution*.

While the MCS represents the higher level behavior associated with the overall trajectory evolution, each independent segment and its constituent components may be configured to increasing levels of specificity. Returning to the previously mentioned example case of propagation, the associated model for propagation may be a default or may be defined by the user, and may incorporate any number of environmental or dynamical agents. Environmental considerations include fundamental effects such as two-body gravitation, specialized notions like Yarkovsky acceleration for considering the evolution of comets or asteroids and many layers of model fidelity in between. Not every individual driving mechanism is appropriate for each spacecraft simulation. For example, in the early stages of mission lifecycle exploration, the environment for modeling the spacecraft evolution might only include the dominant two-body acceleration. Later stages may include force contributions such as albedo, a small but significant effect, that may only be relevant toward the end of the design process and moving into operations. By constructing the overall force model from independent components, extensive combinations are possible. These same principles apply to varying extents to other components. Configuration of all of these aspects occurs at the individual component level in the Component Browser or in place for certain embedded components.

**Table 2:** Astrogator Segments

Segment	Init. Ephem.	Gen. Ephem.	Control Exec.	Description/Function
Initial State	✓	✓		Interface to set spacecraft initial conditions
Launch	✓	✓		Simple modeling to connect launch and burnout states by splining a quarter-ellipse
Follow	✓	✓		Inherit from a section of the path of another spacecraft
Hold		✓		Maintain a fixed position in a given reference frame until specified conditions are met
Update		✓		Interface to manually modify various spacecraft state values
Propagate		✓		Numerically evolve the spacecraft consistent with a configurable force model
Impulsive Maneuver		✓		Apply an instantaneous change to the vehicle's velocity vector
Finite Maneuver		✓		Incorporate thrust from specific engine modeling and pointing into the numerical evolution
Optimal Finite Maneuver		✓	✓	Optimize the pointing direction along the path of a finite maneuver using direct transcription
Sequence			✓	A container for other segments that includes a scripting interface and other execution control
Backward Sequence			✓	A sequence with members that are numerically evolved in reverse time
Target Sequence			✓	A sequence that adjusts members' parameters consistent with various search/control algorithms
Return			✓	Returns execution control from a sequence to its parent; may be en/disabled by profile
Stop			✓	Halts execution of the MCS; may be en/disabled by a target sequence profile

### Low-level Configuration: Components

The functional notions of Astrogator exist as configuration options for an STK satellite object. However, the associated algorithms and techniques require adjustments to specific spacecraft properties and the larger modeling environment. Particular Astrogator components impart varying levels of consequence for the overall system. The following subsections are ordered roughly by the associated level of influence over the modeling (greatest to least) as well as by functional relationship.

*Central Bodies* A fundamental concept of the Systems Tool Kit is the central body or bodies associated with a simulation. Most analyses in space and time, orbital or otherwise, happen in or around our local stomping ground, Earth. Other analyses might require enabling STK's interplanetary options for studies at or near some other solar system body. In some cases, users may require a central body that is not offered by default in STK (e.g. some asteroids or comets, or an alternately defined, existing central body). STK relies on a set of central bodies to enable the function of many of the low-level capabilities of the tool. However, a convenient means for adding a new central body emerged from Astrogator's requirement for user-defined central bodies (driven by a particular mission need), and this came through central body components.

Central body components represent a wrapping of a low-level STK central body, and they hold other embedded components or constituent pieces. These pieces include gravity models, whose definitions are taken from a file and allow configuration of lower-order harmonics from the user interface. Central body components also hold file-based or otherwise simply-configurable shape models. Notions of central body "attitude" and motion may be defined by specified models through the interface or through external files. A user may configure an STK scenario to model a system that is completely unique with respect to its corner of the Universe, by copying and modifying an existing central body. User-defined central bodies may then be used as the primary body for force models or to define departure and/or arrival conditions for an interplanetary spacecraft. They may also serve as a basis for updated analyses of their selves as missions to characterize the bodies feed back new information. The capability to adapt the primary environment of the simulation by supplying user-defined central body information enables effective planning options.

*Propagator Functions (Force Model Constituents) and Propagators* For use with Astrogator, a propagator component consists of a set of equations of motion in the form of propagator functions as well as a set of numerical integrator configuration parameters. The numerical integrator configuration is also a component that is embedded directly into the propagator component, and does not exist independently in the Component Browser (a good example of such an embedded component). The propagator requires a central body and this is supplied as a link to one of the central bodies in the Component Browser. Propagator functions are included in the list of functions for the propagator and while they derive from existing components elsewhere, they are embedded into the propagator's list and configured in place. An appropriately configured propagator may be used to evolve a trajectory with a propagate or finite maneuver segment, or to serve as the force model for the direct transcription optimization process of the optimal finite maneuver.

Propagator functions represent some of the most powerful pieces of the Astrogator environment. These functions reflect the underlying dynamical environment and capture the effects that allow for the concept of High Precision Orbit Propagation (HPOP) a term used in different contexts throughout STK as well as in AGI's Orbit Determination Tool Kit (ODTK). Each force model component has configurable parameters and may rely on other components or inputs. General force model components are available at the top-level category of the propagator function library, while other propagator functions are categorized by their particular force-modeling concept. These categories include atmospheric density models that enable computation of orbital regime appropriate drag forces\*; gravity models where one of two-body, configurable field or Circular Restricted 3-body Problem (CR3BP) gravity models may be selected for an appropriate propagator\*; plugin force models accommodating multiple programming paradigms; various solar radiation pressure force models\*; and third-body perturbation forces including individual solar system bodies as well as sev-

---

\*Recent additions exist for the marked categories and will be elaborated in the Expanded Model Support section.

eral planetary systems taken at their respective system barycenters. All of the propagator functions allow for a uniquely configurable environment to support high-precision propagation. At the same time, multiple force model configurations may be established for simulations that evolve between force model regimes.

*Engine Models and Thruster Sets* Astrogator models spacecraft thrust using maneuver segments composed of several functional elements. One of the critical elements for maneuvers is the engine model that supplies the thrust magnitude for the spacecraft. Several engine models exist as components for use in finite maneuvers with underlying implementations that supply thrust and specific impulse values as functions of the spacecraft state. Specific options include constant thrust or acceleration variants, plugin or custom function options, an ion model that describes thrust parameters using polynomials in power and a polynomial option for thrust as a function of tank temperature and pressure. Each of these offerings may be configured to match a particular mission requirement—if an existing type isn't sufficient, a plugin may be used. As with any aspect of the Astrogator architecture, new additions to the default repertoire are always a possibility as evidenced by the recently added throttle-table engine model—this new engine model is elaborated in the Throttle-table Engine Model section.

For a particular spacecraft configuration, a single engine may not adequately describe the thrust configuration. To accommodate different thrust geometries, Astrogator allows for an engine to be implemented as a thruster or multiple engines combined in a thruster set. These options are available as components and follow the existing paradigm of “configure unique variations and use as needed.” Thus, different mission phases can use different thrust configurations consistent with constraints or other strategies.

*Power Sources* Some engine models may require power as an input for their functions. This is true for the ion engine, but might also be necessary for a user-provided custom/plugin engine model. Power source components are configurable and include internal or solar power, each allowing for a power processing unit in the power chain.

*Stopping Conditions and Constraints* Numerous preconfigured event triggers are available for use within Astrogator and user-defined options that take advantage of calculation object components are also available. Common stopping conditions such as orbit peri/apoapsis, plane crossings, duration, epoch, true anomaly, and several others are all available. Conditions enabled by the STK environment are also available such as lighting or access conditions. An extremely powerful but well-kept secret of stopping conditions is the availability of the *before* stopping condition, as in “stop when a certain condition is met prior to some other condition.” For example, a particular before stopping condition might follow the logic to stop propagation on the last entrance into sunlight prior to line-of-sight (access) to another object. At which point, an appropriate action for the subsequent evolution can be taken. Stopping conditions allow for the number of times the condition should be met before activating, the direction in which to count an event as applicable and the number of times to repeat the event finding in the case that the stopping condition triggers a subroutine (autosequence) that returns control back to the current operation upon completion. All of these event finding operations also allow for additional constraints to be placed on them. For example, “stop on periapsis *if* altitude is less than or equal to some value.” Individual constraint components and user-selected stopping conditions are all configurable as needed.

*Calculation Objects* As a spacecraft state evolves, many useful values may be calculated as a function of the current state. Calculation object components, the most numerous component type,

provide options for various values to be evaluated at a given time along the associated spacecraft ephemeris. These calculation objects are configurable to varying degrees depending upon their associated computation, and entirely unique values are possible through scripting calculation object components. The calculation objects may be added to a particular segment within an Astrogator MCS and evaluated after the segment has been executed. They also serve as data providers for reports and graphs in the STK Report & Graph system. Some calculation objects operate as a bridge to the dynamic geometric system of STK, the Analysis Workbench. Finally, calculation object components can serve as inputs for stopping conditions, constraints, targeter goals and other operations. The calculation object catalog is continuously growing with several new additions made recently. Calculation objects are a critical, robust and powerful mechanism of the Astrogator paradigm.

*MCS Segments* The primary functional element of the mission control sequence is the segment. Some discussion of the various segments with respect to their functions was offered previously. Here, the focus is on the component nature of the segments. As components, any particular type of segment may be duplicated and customized for reuse in various contexts. Indeed, many applications of Astrogator rely on sequence segments that have been preconfigured with parameters for their operation available as inputs to their associated scripting interface. Some previously configured sequences are available as templates in the “Examples” subcategory within the software.

*Design Tools* Additional discussion with respect to the design tool components will be given in the Expanded Initial Guess Support section. The general concept of these components is that of configurable calculators of varying complexity. For example, the first installment of the design tool components is a relatively simple Lambert solver. This tool supplies options for computing certain Lambert solutions, and provides a mechanism for exporting a sequence segment for use on an Astrogator satellite. These tools are intended to offer starting points for various analyses.

### **Intermediate Level: Algorithms**

Between the individual components and the “go button”, various sophisticated algorithms are employed to control the system execution. Propagation is controlled by a variety of numerical integration techniques, modular force model definition and robust event detection. Maneuvers involve a range of fidelity, with modeling as simple as instantaneous impulses to complex engine model and thrust pointing schemes and beyond to thrust-pointing optimization. MCS constituent sequences and target sequences offer a scripting interface for specialized behavior and preconditioning, search algorithms and additional run-time logical control strategies.

*Coordinate System Transformations* Throughout STK object state data is evolved through space and time. In every case, what is meant by “space” is critical to the simulation. Astrogator leverages extensive native capabilities built into STK that define coordinate systems and transformations between these systems. It also utilizes the powerful Analysis Workbench framework for many transformations. In a few, limited cases, unique Astrogator transformations are built directly into the tool. Often transformations between systems are nontrivial, for example, the underlying algorithms for transformation between fixed and inertial coordinate systems implements the complex and sophisticated International Celestial Reference Frame (ICRF)<sup>14,15</sup> and associated theory. Other transformations may be as simple as translation or involve more complex, multiple chained transformations. One prominent area where coordinate systems are applicable within Astrogator is in the initial state segment, where a state vector may be defined in multiple reference systems and then converted between these systems. The combination of options available within Astrogator for performing transformations adds significantly to its effectiveness.

*Numerical Integration and Event Finding* A modular numerical integration strategy is implemented for use within Astrogator. User-selected propagation functions correlate with independent equations of motion. When combined as a set of propagator functions, the equations of motion become interdependent. The numerical propagation system utilized by Astrogator manages the associated dependencies. This system evaluates the force model and supplies the results to the numerical integrator. Each step along the process must reconcile what reference system is required to evaluate the associated equations, and so the system must be aware of any prerequisites. The numerical integration options within Astrogator are implemented from typical algorithms and offer a number of choices for evolving the spacecraft state. Throughout evaluation, the numerical propagation system employs stopping conditions as previously described for event finding. When a particular trigger condition is traversed, integration is stepped back to the state prior to the trigger. A new step is proposed to refine the crossing, and then the process is repeated until the trigger is met to prescribed accuracy. All relevant triggers are found during the evolution and refined as needed. While the implementation of the event-finding process is fairly standard, the available trigger options are expansive and extendable given utilization of the calculation object component paradigm.

*Floating End-point Targeting* Refined through extensive practical application, the Astrogator differential corrector offers a means for solving a generic targeting problem where adjustments in control parameters are made to achieve some desired goal. Astrogator currently uses a system of numerical derivatives to perform a single-shooting loop to execute the floating end-point targeting of the differential corrector.<sup>16</sup> This system has evolved from earlier incarnations<sup>3</sup> and has incorporated numerous adjustments and improvements<sup>7</sup> over a number of years. A specialized system is implemented for reconciling the dependencies of Astrogator result values based on problem control variables. Additional, advanced options are available for the differential corrector, and the tool has proven robust for its purpose given an appropriately posed problem. A single parameter bisection algorithm is also available to solve simple targeting problems.

*Numerical Optimization Profiles* Astrogator has previously offered various forms of numerical optimization, and these options have evolved over time. In recent years, powerful numerical optimizers including the Sparse Nonlinear Optimizer (SNOPT)<sup>17,18</sup> and Interior Point Optimizer (IPOPT)<sup>19,20</sup> have been added to function as target sequence profiles. These profiles allow extensive options for constructing optimization problems following the paradigm of the differential corrector's choices for decision variables, objective functions and constraint functions. This flexibility has advantages and disadvantages, and additional discussion is provided in the Expanded Optimization Support section.

*MCS Reconfiguration Profiles* Yet another set of options for the target sequence are the MCS reconfiguration/control profiles. These include profiles that will toggle whether certain segments or stopping conditions are enabled or disabled, a profile that changes the propagator for a particular segment, profiles that enable transitions from impulsive to finite maneuvers, scripting profiles, and a profile to initialize the target sequence's segments. All of these profiles allow for more complex logical patterns, and provide options to meet various objectives. The added flexibility and automatable characteristics of these profiles introduces additional solution strategies.

*Scripting Profiles* Eventually the necessity to directly interact with the user interface reaches a point of diminishing returns. Once a baseline process or solution has been established, repetitive variations on the process can be tedious. For this reason (and others), Astrogator offers scripting capabilities at certain points in the process. User-created calculation objects can be made through scripting, and the overall STK/Astrogator simulation can be controlled through external scripting.

In the middle of the process, however, are the capabilities to script on sequences. The forward and backward sequences offer an interface to set up parameters and perform other operations on the sequence when it executes. The target sequence offers a scripting profile that can be used to initialize or adjust values in preparation for a subsequent profile. Some profiles also offer scripting capabilities that run in concert with the other functions of the profile. These scripting capabilities allow for additional levels of automation and customization.

*Prescribed Attitude Control* Astrogator has two mechanisms for defining attitude for the spacecraft. One option is common between STK vehicles, and involves invoking an attitude profile that the vehicle will follow. This option is usually a function of the vehicle's state or geometry with other simulation objects, or it can be defined with a precomputed attitude history saved to file. The other optional mode for defining the attitude of an Astrogator satellite is to inherit the attitude from its maneuver pointing. When employed, the maneuver pointing will also define the attitude of the satellite for a time period associated with the maneuver. Several profiles exist to determine the maneuver pointing for both impulsive and finite maneuvers, while the optimal finite maneuver will optimize the maneuver pointing in terms of spherical angles or unit vector components. Both the impulsive and finite maneuvers allow for specification of the maneuver pointing to be set by the user consistent with certain vector or alignment specifications or from a file. In the case of the finite maneuver, the options are expanded, still allowing for specification as before, but now offering options to maintain the maneuver alignment with respect to the body frame throughout the maneuver or to hold the direction inertially. The finite maneuver also allows for thrust-pointing through a plugin or by following a functional evaluation in time, fit to the duration of the maneuver. This last, time-varying, option offers polynomial and sinusoidal terms to define the functional form. Additional discussion of the time-varying thrust pointing model may be found in the Expanded Maneuver Pointing Support section.

*Thrust Pointing Optimization through Direct Transcription* Extended discussion regarding the technical details associated with the direct transcription method for thrust-pointing optimization are available in the Optimal Finite Maneuver section. Generally speaking the method allows for specification of an initial guess along the path either by inheriting from a standard finite maneuver or from a file. Given an initial guess, numerical optimization and direct transcription are employed along the path to find the optimal thrust pointing consistent with a few common objective functions.

*Simple Launch Modeling* While generally of lower fidelity when compared to many of the other capabilities offered for use within Astrogator, the launch segment does involve unique modeling. The launch mechanism essentially attempts to connect launch and burnout states with a section of an ellipse. The ellipse is fit with either a cubic or quartic spline and ephemeris for the path is generated by numerically integrating along the ellipse from launch to burnout. A user-specified time step is employed for the resulting ephemeris data.

## RECENT ADVANCEMENTS

During the most recent Astrogator development renewal at AGI, several new or expanded capabilities have emerged. Some of these capabilities represent simple extensions of existing options or new alternative options for previous capabilities. Other advancements consist of entirely new behavior. In many cases, the associated feature-sets remain active areas of ongoing development, and Astrogator developers are happy to receive input on implementations.

*Expanded Model Support* Propagator functions represent an interface for lower-level mathematical (sometimes empirical) force models implemented as equations of motion. Several recent additions have been incorporated within the set of models offered through the propagator functions. Each is enumerated along with some combination of technical details, unique implementation considerations or standard references for the associated models.

Among the atmospheric density models, a blended density model offers linear blending across a range of altitudes. This model takes as input two other models and the blending range for the model. The blending range begins at the lowest extent of the upper model and continues upward to the specified distance. The blending is a function such that, at the bottom of the blending range, the density is entirely that of the lower model. At the top of the range, the density is entirely that of the upper density model. This relationship is captured in Equation (1) for the altitude spanning the blending range, where  $\rho$  is density,  $\alpha$  and  $\beta$  are scalars.

$$\rho(\text{alt}) = \alpha(\text{alt}) \rho_{\text{lower}}(\text{alt}) + \beta(\text{alt}) \rho_{\text{upper}}(\text{alt})$$

$$\{\text{alt} : \text{alt}_{\min} \rightarrow \text{alt}_{\max}\} : \{(\alpha : 1 \rightarrow 0), (\beta : 0 \rightarrow 1)\} \quad (1)$$

The blended model is included to account for atmospheric density models that terminate at some minimum altitude. While orbital drag is an important effect that will act on a spacecraft, the typical assumption is that the satellite will generally remain on orbit. Once the satellite dips low enough into the atmosphere, the aerodynamics quickly dominate the force model. These forces are a function of the atmospheric density which becomes increasingly small at higher altitudes. Given these considerations, many atmospheric density models intended for use with orbiting spacecraft set some minimum altitude as their lower bound of reasonable modeling. Other atmospheric density models reach all the way to the ground, and yield varying results. Regardless, it is sometimes of interest to use one model at higher altitudes, and additional consideration must be given for what to do below its effective altitude. The blended atmospheric density model is included to meet these use cases.

The Drag Temperature Model (DTM) 2012<sup>21</sup> and Jacchia–Bowman 2008<sup>22</sup> models represent more modern atmospheric density models. The former is a semi-empirical thermospheric model developed by Centre National d’Etudes Spatiales (CNES) while the latter is an empirical thermospheric density model developed by Space Environment Technologies (SET). Both of these models require additional space weather indices beyond those required for other models. These models offer alternative options for atmospheric density which may yield more accurate results depending on the simulation. The associated references detail the supporting research for these models.

A plugin atmospheric density model is available following existing AGI plugin coding paradigms. This plugin point also supports augmented space weather data similar to the Jacchia–Bowman models. This plugin capability allows users to bring their own atmospheric density models for use within STK/Astrogator when such a need might arise. An example density plugin ships with STK.

The Global Reference Atmospheric Model (GRAM) series consists of various “engineering-level atmospheric models applicable for engineering design analyses, mission planning, and operational decision making”.<sup>23</sup> For many years, several versions of the Mars-GRAM models (v3.7, 2000, 2001 and 2005)<sup>24</sup> were available through a non-standard external DLL plugin system for use in STK/Astrogator. Recently, these previous versions have been refactored to be available out-of-the-box for use within STK consistent with licensing terms from NASA. Further, the Mars-GRAM 2010 model has also been added to the available models. All five versions have also been made available for use with STK/HPOP, a propagation-only satellite type offering high-precision propagation within STK aside from Astrogator, and the Analysis Workbench Force vector.

The N-Plate solar radiation pressure model has been added for use in Astrogator in efforts to reach and maintain parity with ODTK.<sup>25,26</sup> This model is common to both systems, although some parameters are relevant only to the orbit determination process. Driven by file, the model allows for significant user configuration. A particular definition file for this SRP formulation allows for sets of plates defined in three ways. The plates may be designated as Sun-pointing plates, where their normals are taken to be always aligned with the Sun direction. They may be constrained, Sun-pointing plates where the alignment is toward the Sun to the extent that some other constraint vector allows; the result is a coning motion for the plate normals. Lastly, the plates may be fixed in the body frame. Some combination of plate definitions allows for reasonably effective modeling of a particular spacecraft for the purposes of solar radiation pressure forces.

The Circular Restricted 3-body Problem (CR3BP) models the evolution of a particle (i.e., a small body assumed to have negligible mass) under the direct, simultaneously influence of two massive bodies. The model is formulated in a rotating reference frame such that the two primaries are fixed in the frame. A number of simplifying assumptions are made to construct the CR3BP model including modeling the primaries as point-masses and evolving the less-massive primary (or *secondary*) body on a circular orbit. The mathematical formulation of the problem is often posed in nondimensional quantities with the origin of the rotating reference frame taken at the barycenter of the primaries. It admits an integral of the motion, reflects fixed-point solutions (i.e., the libration points), periodic orbits and other ordered and chaotic dynamics. The model is often utilized as a sandbox for dynamical systems theory applications, and its principal features, while emerging from a lower-fidelity model, often effectively capture the governing behavior in regimes where two massive bodies generally interact in the specified configurations. The well-known libration points associated with some common space applications exist as time-dynamic higher-fidelity realizations of their underlying CR3BP analogs. The CR3BP model has recently been incorporated into Astrogator's set of available force models implementing a formulation similar to Szebehely's<sup>27</sup> except that the formulation is taken with the more massive primary to the left of the origin, the less massive to the right, as in Howell<sup>28</sup> and others. The implementation within Astrogator has been executed in such a way that, given an appropriately configured STK scenario, the CR3BP dynamics can be integrated independently or with other appropriate propagator functions included. This recent addition is currently undergoing validation, and additional extensions of the capabilities are forthcoming.

*Expanded Initial Guess Support* A recent expansion of the Component Browser model introduces components that are intended for use as preliminary, calculator-like operations. These tools are found in the Design Tools library of the Component Browser, where an initial installment has been added in the form of a Lambert solver. The various operations associated with the Lambert solver are implemented following Battin,<sup>29</sup> as these methods have been previously employed within STK for other vehicle types. Additional details associated with the Lambert solver tool are available in the STK documentation.<sup>30</sup> The Lambert solver itself is intended to be a relatively simple tool to help establish the overall design tool paradigm. These tools are included in the Component Browser to build on the patterns existing there, namely that a user can create their own particular "flavors" of any of the tools. For example, the Lambert solver offers two modes, one mode that allows for explicit specification of departure and arrival states and another that derives these states in relation to STK central bodies. A user can copy the default Lambert solver, configure the copy as desired and then save it along with a default scenario (or place it in their user configuration area). Thus, reuse of a uniquely configured Lambert solver is enabled. Other design tools are planned for inclusion in the library including additional Lambert operations. The driving motivation for these tools is to

help with the initial stages of design, and answer the frequent question of, “where do I start?”

*Expanded Maneuver Pointing Support* An improvement to maneuver pointing that seeks to simplify user workflows has been incorporated for Astrogator’s finite maneuver. A pointing strategy denoted “Time Varying” allows for thrust direction to be specified with two spherical angles in terms of polynomial and sinusoidal components. The burn direction can be varied in both the azimuth and elevation of a specified set of thrust axes. The azimuth and elevation may be varied according to a 4<sup>th</sup>-order polynomial with the addition of a periodic term controlled by amplitude, frequency and phase as expressed in Equation (2). This option has been added in response to requests from users

$$\begin{aligned} \text{Azimuth}(t) &= \text{Az}_i t^i + \text{Az}_{\text{amp}} \sin(\text{Az}_{\text{freq}} t + \text{Az}_{\text{phase}}), \quad i = [0..4] \\ \text{Elevation}(t) &= \text{El}_i t^i + \text{El}_{\text{amp}} \sin(\text{El}_{\text{freq}} t + \text{El}_{\text{phase}}), \quad i = [0..4] \end{aligned} \quad (2)$$

who migrate designs and solutions back and forth between Astrogator and NASA’s Copernicus software. Astrogator’s developers have sought to recreate the functionality of Copernicus’ parameter steering model as published by Ocampo.<sup>31,32</sup> While the internal implementation is undoubtedly different between the two tools, minimal conversion is necessary for transferring solutions. For Astrogator users, the incorporation of these capabilities alleviates the previous necessity of user plugins to accomplish the same results. An example application is included in the Matching the Thrust-pointing for the Apollo 11 Lunar Descent section.

*Expanded Optimization Support* Numerical optimization is an expansive field unto itself. Of course, the field exists primarily because optimization is typically always beneficial to the associated process. For trajectory applications, optimization problems are usually concerned with achieving some mission objective in the most economic way possible. This economy might be in terms of fuel, time or some other metric. Sometimes, trajectory optimization problems require the capability to answer these questions along the entire trajectory—one particular formulation of this type is offered in the Optimal Finite Maneuver section. Often, however, the problem is posed as, “given current resources, what can be achieved sometime later than now?” The Astrogator option to answer these questions is similar in form to the differential corrections process implementing single shooting to solve floating end-point problems. Astrogator’s Target Sequence offers two relatively recent additions in the form of numerical optimization search profiles. These profiles serve as interfaces to two industry standard numerical optimization packages: the Sparse Nonlinear Optimizer (SNOPT)<sup>17</sup> and the Interior Point Optimizer (IPOPT).<sup>19</sup>

The Astrogator optimization profiles are implemented to allow flexible problem definition. As profiles for a target sequence, the optimizers have access to all control variables, as well as any calculation object configured as a result for a particular segment, included in the sequence. The control variables can include maneuver properties, stopping condition trigger values, initial state values and numerous other options. The results are used to form constraint and objective functions for the optimization problem. Given the latitude with which problem variables may be selected, it is possible to easily construct an optimization problem that is poorly posed for the optimizer. This drawback comes at the benefit of general flexibility. Given a well-posed problem, both optimizers solve local optimization problems and often do so very well. However, a good initial guess for a solution is critical for such local optimizers. Further, both optimizers currently rely on manual or heuristic scaling of variables as well as numerical derivatives. The addition of these optimization capabilities represents a first step toward solving some very challenging problems. Astrogator developers, given extensive user feedback, are currently initiating a new phase of improvements with respect to aspects of optimizer implementation.

*Optimal Finite Maneuver* The mission use of electric propulsion (EP) systems providing continuous, low thrust has steadily increased over the past several years. The higher momentum-transfer efficiency, or Specific Impulse, of low-thrust engines results in greater effective-payload-to-propellant ratio, or alternately, smaller launch costs for a given payload size, making these engines attractive for a variety of mission types. As EP technology continues to mature, the upward trend in its efficient utilization is expected to persist. Even though the primary advantage of low thrust levels, namely, a large accumulated  $\Delta V$  with the expenditure of relatively little propellant mass compared to chemical propulsion systems affords ample motivation for the adoption of EP, the underlying mathematical problem of trajectory design of low-thrust-propelled vehicles is challenging, see, for example, reference [33] and the sources cited therein. The Optimal Finite Maneuver (OFM) feature of Astrogator’s Maneuver MCS segment, introduced in STK 11.5, is designed to meet many of these challenges.

The Astrogator OFM is a collocation-based trajectory optimization tool which solves a single-phase trajectory optimization problem. Multi-phase problems may be posed by concatenating multiple Maneuver segments, and phase-matching constraints enforced through well-documented mechanisms.<sup>8</sup> Trajectory optimization using Direct Transcription, of which the collocation method is an incarnation, has been extensively studied, and as such, is only briefly discussed here; interested readers may turn to references [34, 35, 36] for details.

The OFM minimizes a certain mission objective, such as the orbit transfer time of flight, propellant consumption, etc. by appropriately selecting the thrust attitude program, the initial/final vehicle orbit states, and possibly also the maneuver duration. Mathematically, this is typically posed as the following Optimal Control problem: find the state-control function pair  $\{\mathbf{x}(\tau), \mathbf{u}(\tau)\} \in \mathbb{R}^n \times \mathbb{R}^u$  and epoch(s)  $\tau_0$  and/or  $\tau_f$  such that the cost functional:

$$J[\mathbf{x}(\cdot), \mathbf{u}(\cdot), \tau_0, \tau_f] = \psi(\mathbf{x}_0, \mathbf{x}_f, \tau_0, \tau_f) + \int_{\tau_0}^{\tau_f} \mathbb{L}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau \quad (3)$$

is minimized subject to the differential-algebraic system of equations:

$$\dot{\mathbf{x}}(\tau) = \mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) \quad (4)$$

$$\Psi(\mathbf{x}_0, \mathbf{x}_f, \tau_0, \tau_f) = \mathbf{0} \quad (5)$$

where  $\mathbf{x}_0 = \mathbf{x}(\tau_0)$ ,  $\mathbf{x}_f = \mathbf{x}(\tau_f)$ . For an Astrogator satellite, the state  $\mathbf{x}(\tau)$  comprises its orbital elements and mass, the control functions  $\mathbf{u}(\tau)$  are the thrust attitude direction cosines or spherical angles in a specified reference frame, and the dynamical equations are those modeled by the available propagator force models, such as HPOP, Earth J2, Heliocentric, etc. Users may choose to solve the above problem either in terms of Equinoctial Elements (EE) or Modified Equinoctial Elements (MEE). The vector function  $\Psi$  represents the terminal constraints at the conclusion of the maneuver, and will typically involve an equality on five (all but the true/mean longitude) or all six elements.

A standard technique for solving an Optimal Control or Trajectory Optimization problem as posed by Equations (3)–(5) is to convert or *transcribe* it into a standard numerical optimization problem of the following type:

$$\begin{aligned} \min_{\xi} \quad & \mathcal{J}(\xi) \\ \text{s.t.} \quad & \mathcal{C}(\xi) \leq \mathbf{0} \\ & \mathbf{a} \leq \xi \leq \mathbf{b} \end{aligned} \quad (6)$$

where  $\mathcal{J}$ ,  $\mathcal{C}$  are the cost and constraint functions, respectively,  $\xi$  are the decision variables, and  $\mathbf{a}$ ,  $\mathbf{b}$  are the variable bounds. This parameter optimization problem is solved using a nonlinear programming (NLP) library. In Astrogator, *pseudospectral methods*, in particular, the *Legendre Pseudospectral Method* (LPM) and *Radau Pseudospectral Method* (RPM), are employed to accomplish the transcription,<sup>34,36</sup> and SNOPT is used for solving the resulting NLP. The OFM user-interface allows sophisticated NLP scaling options to be applied; thus, users may choose to scale the orbit transfer problem using the well-known Canonical dimensions, use SNOPT’s native scaling options, or a combination of both to facilitate convergence.

The solution of finite-maneuver optimization problems using the direct transcription method requires that an *initial guess* or approximation for the optimization decision variables  $\xi$  be supplied to the transcription-optimization system. Following LPM and RPM, the decision variables are the orbit states and the control functions (thrust attitude in terms of spherical angles or unit vectors) at a set of nodes, and this set may also include the maneuver start/end time(s). Astrogator users can either import a pre-generated initial guess in the form of an Initial Guess File, or use a pre-run Finite Maneuver, as a source for the initial guess. Once an initial guess has been provided to Astrogator at a set of generic points following either of the above mechanisms, the “raw” guess data are internally re-sampled/interpolated at the discretization nodes (e.g., the Legendre-Gauss-Lobatto nodes for LPM and Legendre-Gauss-Radau nodes for RPM) before numerical optimization is initiated. Additionally, upon the execution of an OFM, the solution is internally stored, and may be exported as an ASCII file, thus supporting a continuation-type paradigm for solving related problems of progressively increasing fidelity/complexity.

*Throttle-table Engine Model* A throttle table is a tabular representation of an EP system performance (typically Hall/Ion thrusters) as a function of the thruster input power. The electrical power available to a spacecraft’s propulsion system changes with varying distance from the Sun, or with fluctuating lighting conditions as it encounters umbra/penumbra regions while orbiting in the vicinity of an obscuring body. A throttle table captures the variation of propulsion parameters such as Thrust ( $T$ ), Specific Impulse ( $I_{sp}$ ), and Mass Flow Rate ( $\dot{m}$ ) at various thruster input power levels ( $P$ ), or throttle levels. Examples of throttle tables available in the open literature include the NEXT (ion), NSTAR (ion) and BHT (Hall) engine tables.<sup>37-39</sup> Astrogator’s recently-added Throttle Table Engine Model accepts, through a GUI, user-provided throttle table data in an ASCII file containing enumerations of  $\{P, I_{sp}, T\}$  or  $\{P, \dot{m}, T\}$ . Users can then choose to create a “data model” from this information for propagation/optimization. Currently, the following three options are supported for data model creation:

- *Continuous with regression polynomial fit*: Fits a regression polynomial of user-specified degree to interpolate  $T(P)$ ,  $I_{sp}(P)$ ,  $\dot{m}(P)$ .
- *Continuous with piecewise linear fit*: Interpolates  $T(P)$ ,  $I_{sp}(P)$ ,  $\dot{m}(P)$  by fitting a piecewise-linear interpolant through the data points.
- *Discrete power levels*: Uses zero-order-hold interpolation to compute  $T(P)$ ,  $I_{sp}(P)$ ,  $\dot{m}(P)$ , i.e., if the current available thruster power lies in  $[P_i, P_{i+1}]$ ,  $P_{i+1} > P_i$ , then the quantities of interest are evaluated at  $P_i$ .

## SELECTED EXAMPLES

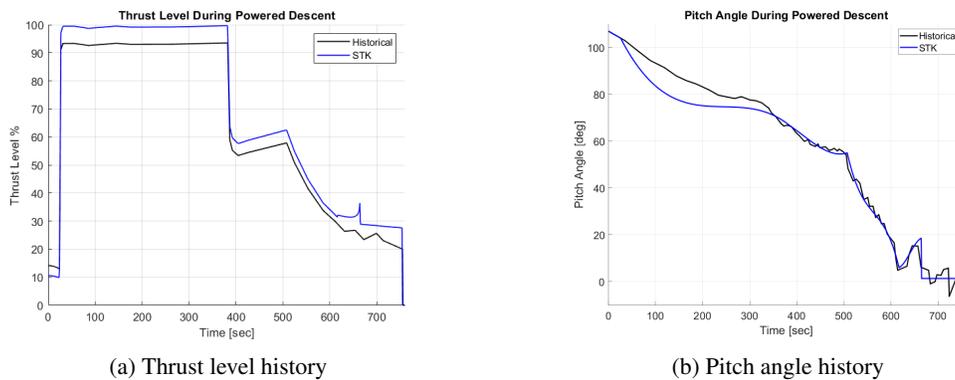
While the goal of this paper is not necessarily to provide extensive application details, two cases have been highlighted to capture: (1) improved ease of use from a new feature, and (2) a simple

example of powerful new capabilities that take Astrogator in a different direction. The first example is associated with a recent recreation of the Apollo 11 lunar landing, illustrating reduced user onus to accomplish the simulation. The second example is related to the new optimal finite maneuver, intimating the possibilities associated with this feature.

### Matching the Thrust-pointing for the Apollo 11 Lunar Descent

The release of STK 11.4 introduced a new attitude control definition for finite maneuvers. The new attitude profile, called “Time Varying”, allows the direction of the finite maneuver to vary as a function of time over the maneuver’s duration as described in the Expanded Maneuver Pointing Support section. As part of the celebration of the 50<sup>th</sup> anniversary of the first Moon landing, AGI decided to pursue a reconstruction of the Apollo 11 mission utilizing Astrogator. The goal of the reconstruction was to rebuild the entire Apollo 11 trajectory as technically and historically accurate as possible from publicly available sources. A particular section of interest that highlights Astrogator’s modeling capabilities and flexibility was the powered lunar descent. The last twelve and a half minutes of the descent were powered by the Lunar Module, which took the Apollo 11 crew from an altitude of 50,000 ft down to the lunar surface. Apollo-like lunar powered descents have been modeled previously in Astrogator with a series of finite maneuvers. These finite maneuver segments had a fixed burn attitude aligned with the spacecraft’s anti-velocity direction in the Moon fixed frame.<sup>40</sup> While this approach was sufficient for the associated analysis, it did not fully capture the varied attitude profile from the historic mission. When rebuilding the Apollo 11 mission, the time-varying finite maneuver was used to more closely follow the pitch attitude of the flown mission.

The Apollo 11 powered descent consisted of six main burn phases characterized by different thrust levels of the lunar module descent engine. The burn phases are powered descent initiation, max throttle, throttle recovery, high gate, low gate, and manual control until touchdown.<sup>41</sup> These burn phases also roughly coincide with changes in the lunar module’s burn attitude. The thrust and pitch attitude are shown in Figure 1, which compares the historical data to the values calculated during the Astrogator recreation.<sup>42</sup>



**Figure 1:** Apollo 11 lunar powered descent pitch and thrust level comparison

In order to capture the varied pitch profile during the descent, each of the burn phases was modeled as a target sequence containing a time-varying finite maneuver. A 3<sup>rd</sup>-order polynomial was used to control the burn attitude by adjusting the polynomial coefficients of the azimuth in the RIC frame. For each burn phase, the target sequence was configured to adjust the thrust efficiency and

the coefficients for the 1<sup>st</sup>-, 2<sup>nd</sup>- and 3<sup>rd</sup>-order azimuth terms to meet the boundary conditions for the lunar module’s altitude, altitude descent rate, lunar velocity and pitch angle. The target sequences utilized Astrogator’s differential corrector to meet the boundary conditions within the specified tolerance. Although the polynomial fit does not exactly reproduce the historical data, it is sufficient to meet the boundary conditions consistent with the associated constraints of the thrust parameters. This example illustrates the capability to successfully capture the salient aspects of the pitch profile for a reasonable reproduction without significant additional programming, modeling or analysis.

### Optimization of a LEO-to-GEO Orbit Transfer

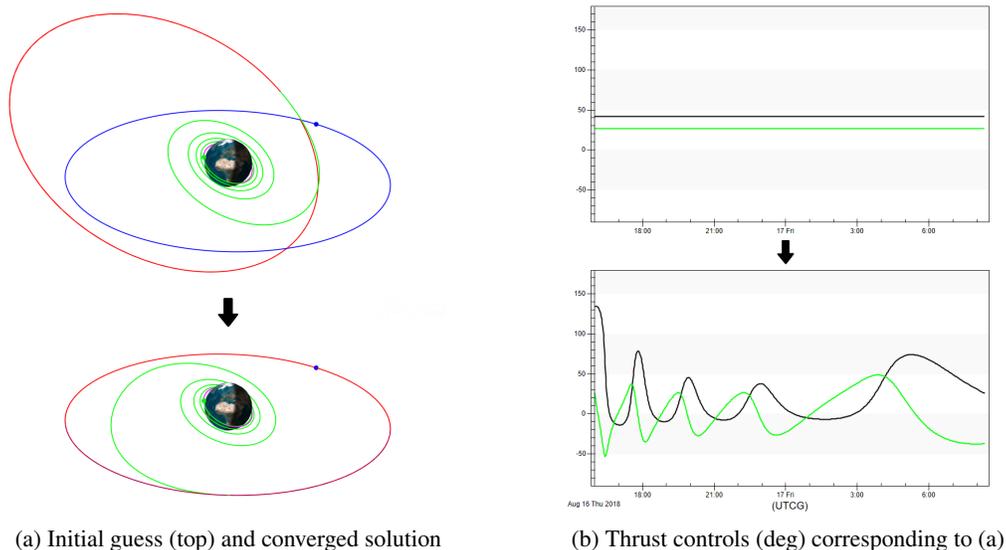
The capabilities of the newly-introduced OFM feature are demonstrated by solving a well-known test case from the literature, namely a minimum-time LEO-to-GEO orbit transfer with a constant thrust acceleration of  $10^{-2}g$ .<sup>43</sup> Briefly, the problem is to transfer a satellite from:

$$[a_0 \ e_0 \ i_0 \ \Omega_0 \ \omega_0 \ M_0] = [7000 \text{ km } 0 \ 28.5^\circ \ 0^\circ \ 0^\circ \ -220^\circ]$$

to:

$$[a_f \ e_f \ i_f \ \Omega_f \ \omega_f] = [42000 \text{ km } 0.001 \ 1^\circ \ 0^\circ \ 0^\circ]$$

in minimum time. However, while reference [43] solves this problem under a two-body + thrust force model, the results presented here are based on Astrogator’s *Earth HPOP* model, which models Earth gravity and perturbations with arbitrarily high fidelity. For this problem, an initial guess was generated by simply propagating the vehicle under constant azimuth and elevation thrust angles of  $42^\circ$  and  $27^\circ$  in the ICR frame (i.e. “Thrust Vector” attitude control for a Constant Acceleration and Isp engine) for a time duration computed from Edelbaum’s approximation.<sup>43</sup> Both the initial guess as well as the converged, final trajectory and controls appear in Figure 2. A 60-node solution with Equinoctial working variables and Canonical units-based scaling is shown, although a Modified Equinoctial Elements parameterization yields identical results in this case.



**Figure 2:** A constant-thrust LEO-to-GEO transfer solved with collocation in Astrogator

## CONCLUDING REMARKS

Since trajectory designers and analysts are restricted to a virtual workbench, software applications to assist in their processes have nearly\* always been necessary. Increasingly, software tools support shifting routine tasks to the background, and allow for more effective complex operations. As one option, Astrogator is supported by proven algorithms and has helped to enable numerous spacecraft missions. Newer advancements to the tool seek to further expand the efficiency of mission planners and operators. Ongoing development seeks to improve numerous aspects of the tool to meet the needs of its users and assist in pursuing their goals in space flight.

## ACKNOWLEDGMENTS

The authors acknowledge all AGI contributors to the various tables and documentation, especially Jonathan Lowe, as well as external partners Mike Loucks, John Carrico and others of Space Exploration Engineering, LLC (SEE). Additional thanks is given to members of the AGI team who help our users understand the machinations of the developers, especially veteran supporter Jens Ramrath.

## REFERENCES

- [1] W. J. Larson and J. R. Wertz, *Space Mission Analysis and Design, Third Edition*. Microcosm, 1999.
- [2] R. A. Heinlein, *Starman Jones*. Scribner's, 1953.
- [3] J. P. Carrico, H. L. Hooper, L. Roszman, and C. Gramling, "Rapid Design of Gravity Assist Trajectories," *ESA Symposium on Spacecraft Flight Dynamics*, Darmstadt, Germany, September–October 1991.
- [4] J. Carrico and E. Fletcher, "Software Architecture and Use of Satellite Tool Kit's Astrogator Module for Libration Point Orbit Missions," *Libration Point Orbits and Applications*, Girona, Spain, June, 2002.
- [5] M. M. Berry and V. T. Coppola, "Correct Modeling of the Indirect Term for Third Body Perturbations," *AAS/AIAA Astrodynamics Specialist Conference*, Mackinac Island, Michigan, August, 2007.
- [6] M. M. Berry and V. T. Coppola, "Integration of Orbit Trajectories in the Presence of Multiple Full Gravitational Fields," *AAS/AIAA Space Flight Mechanics Meeting*, Galveston, Texas, January, 2008.
- [7] M. M. Berry, "Comparisons Between Newton–Raphson and Broyden's Methods for Trajectory Design Problems," *AAS/AIAA Astrodynamics Specialist Conference*, Girdwood, Alaska, August, 2011.
- [8] AGI Documentation Team and Subject Matter Experts, "Astrogator Documentation," March 2019. Online: <http://help.agi.com/stk/index.htm#gator/astrogator.htm>, Accessed: 25 July 2019.
- [9] Johns Hopkins University Applied Physics Laboratory, "Solar 60: Parker Solar Probe Mission Design," July 2018. Online: <https://www.youtube.com/watch?v=rd7YwY0uDf4>, Accessed: 5 April 2019.
- [10] Y. Guo and R. Farquhar, "New Horizons Mission for the Pluto–Kuiper Belt Mission," *AAS/AIAA Astrodynamics Specialist Conference*, Monterey, California, August, 2002.
- [11] E. Lu, M. Loucks, and J. Carrico, "The Distribution of Required Deflection Impulses as a Function of Time Before Impact for Earth Impacting Asteroids," *IAA Planetary Defense Conf.*, Tokyo, Japan, 2017.
- [12] M. A. Mesarch and S. F. Andrews, "The Maneuver Planning Process for the Microwave Anisotropy Probe (MAP) Mission," *AAS/AIAA ASC*, Monterey, California, August, 2002.
- [13] AGI Marketing, "Astrogator Specifications Sheet," March 2019. Online: <https://agi.com/astrogator>, Accessed: 24 July 2019.
- [14] C. Ma et al., "IERS Technical Note 23," Online: <https://www.iers.org/IERS/EN/DataProducts/ICRF/icrf.html>, Accessed: 26 July 2019.
- [15] A. L. Fey et al., "IERS Technical Note 35," Online: <https://www.iers.org/IERS/EN/DataProducts/ICRF/icrf.html>, Accessed: 26 July 2019.
- [16] AGI Documentation Team and Subject Matter Experts, "Astrogator Differential Corrector Documentation," Online: <http://help.agi.com/stk/index.htm#gator/eq-diffcorr.htm>, Accessed: 26 July 2019.

---

\*A compelling chronicle of the efforts of manual trajectory design prior to modern computing is effectively captured in the context of accompanying social challenges faced by several pioneers in the field in Margot Lee Shetterly's recent biographical history: *Hidden Figures: The American Dream and the Untold Story of the Black Women Who Helped Win the Space Race*<sup>44</sup>

- [17] P. E. Gill, W. Murray, and M. A. Saunders, “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization,” *SIAM Review*, Vol. 47, No. 1, 2005, pp. 99–131.
- [18] AGI Documentation Team and Subject Matter Experts, “Astrogator SNOPT Profile Documentation,” Online: <http://help.agi.com/stk/index.htm#gator/ab-SNOPTprofile.htm>, Accessed: 26 July 2019.
- [19] A. Wächter and L. Biegler, “On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming,” *Mathematical Programming*, Vol. 106-1, 2005, pp. 22–57.
- [20] AGI Documentation Team and Subject Matter Experts, “Astrogator IPOPT Profile Documentation,” Online: <http://help.agi.com/stk/index.htm#gator/ab-IPOPTprofile.htm>, Accessed: 26 July 2019.
- [21] S. Bruinsma, “The DTM-2013 thermosphere model,” *J. Space Weather Space Clim.*, Vol. 5-A1, 2015.
- [22] B. R. Bowman, W. K. Tobiska, F. A. Marcos, C. Y. Huang, C. S. Lin, and W. J. Burke, “A New Empirical Thermospheric Density Model JB2008 Using New Solar and Geomagnetic Indices,” *AIAA/AAS Astrodynamics Specialist Conference*, Honolulu, Hawaii, August 2008.
- [23] H. L. Justh, “Overview of Global Reference Atmospheric Model (GRAM) Upgrades,” *15<sup>th</sup> International Planetary Probe Workshop*, Boulder, Colorado, June 2018.
- [24] H. L. Justh, C. G. Justus, and V. Keller, “Global Reference Atmospheric Models, Including Thermospheres, for Mars, Venus and Earth,” *AIAA/AAS Astrodynamics Specialist Conference*, Keystone, Colorado, August 2006.
- [25] AGI Documentation Team and Subject Matter Experts, “Astrogator N-Plate File Format Specification,” Online: [http://help.agi.com/stk/index.htm#gator/NPlate\\_File\\_Format.htm](http://help.agi.com/stk/index.htm#gator/NPlate_File_Format.htm), Accessed: 28 July 2019.
- [26] AGI Documentation Team and Subject Matter Experts, “ODTK N-Plate File Format Specification,” Online: <http://help.agi.com/ODTK/index.htm#od/NPlateFileFormat.htm>, Accessed: 28 July 2019.
- [27] V. Szebehely, *Theory of Orbits: The Restricted Problem of Three Bodies*. Academic Press, 1967.
- [28] K. C. Howell, “Three-Dimensional, Periodic, ‘Halo’ Orbits,” *Celestial Mechanics*, Vol. 32, No. 1, 1984.
- [29] R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics*. AIAA, 1987.
- [30] AGI Documentation Team and Subject Matter Experts, “Astrogator Lambert Solver Documentation,” Online: <http://help.agi.com/stk/index.htm#gator/ab-lambert.htm>, Accessed: 29 July 2019.
- [31] C. Ocampo, J. S. Senent, and J. Williams, “Theoretical Foundation of Copernicus: A Unified System for Trajectory Design and Optimization,” *NASA Technical Reports Server, Document ID: 20100017708; Report Number: JSC-CN-20552*, 2010.
- [32] C. Ocampo, “Finite Burn Maneuver Modeling for a Generalized Spacecraft Trajectory Design and Optimization System,” *Annals of the New York Academy of Science*, Vol. 1017, 2004, pp. 210–233.
- [33] P. Ghosh, “A Survey of the Methods Available for the Design of Many-Revolution Low-Thrust Planeto-centric Trajectories,” *2019 AAS/AIAA Space Flight Mechanics Meeting*, Maui, Hawaii, 2019.
- [34] Q. Gong, F. Fahroo, and I. M. Ross, “Spectral Algorithm for Pseudospectral Methods in Optimal Control,” *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 460–471.
- [35] B. A. Conway, “A Survey of Methods Available for the Numerical Optimization of Continuous Dynamic Systems,” *Journal of Optimization Theory and Applications*, Vol. 152, No. 2, 2012, pp. 271–306.
- [36] D. Garg, M. Patterson, W. W. Hager, A. V. Rao, D. A. Benson, and G. T. Huntington, “A Unified Framework For The Numerical Solution of Optimal Control Problems Using Pseudospectral Methods,” *Automatica*, Vol. 46, No. 11, 2010, pp. 1843–1851.
- [37] S. Oleson, L. Gefert, S. Benson, M. Patterson, M. Noca, and J. Sims, “Mission Advantages of NEXT: NASA’s Evolutionary Xenon Thruster,” *38<sup>th</sup> AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, Indianapolis, Indiana, 2002.
- [38] M. Patterson and S. Benson, “NEXT Ion Propulsion System Development Status and Performance,” *43<sup>rd</sup> AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, Cincinnati, Ohio, 2007.
- [39] J. Szabo, B. Pote, L. J. Byrne, S. Paintal, V. J. Hruby, R. A. Tedrake, G. Kolencik, C. W. Freeman, and N. A. Gatsonis, “Eight Kilowatt Hall Thruster System Characterization,” *33<sup>rd</sup> International Electric Propulsion Conference*, The George Washington University, USA, 2013.
- [40] M. Loucks, J. Carrico, T. Carrico, and C. Deiterich, “A Comparison of Lunar Landing Trajectory Strategies Using Numerical Simulations,” *International Lunar Conference*, Toronto, Canada, Sept., 2005.
- [41] F. V. Bennett, “Apollo Lunar Descent and Ascent Trajectories,” *Tech. Report, NASA/TM-X-58040*, 1970.
- [42] G. M. Low et al., “Apollo 11 Mission Report (MSC 1969 Version),” *Technical Report, MSC-00171, NASA/TM-X-62633*, 1969.
- [43] J. Kechichian, “Optimal LEO-GEO Intermediate Acceleration Orbit Transfer,” *AAS/AIAA Space Flight Mechanics Meeting*, Cocoa Beach, Florida, 1994.
- [44] M. L. Shetterly, *Hidden Figures: The American Dream and the Untold Story of the Black Women Who Helped Win the Space Race*. William Morrow and Company, 2016.