

Keplerian Analysis for Versatile Evaluation of Arbitrary Trajectories

Katherine T. McBrayer¹

NASA Langley Research Center, Hampton, VA, 23666

Stephen J. Edwards²

NASA Marshall Spaceflight Center, Huntsville, AL, 35808

Nikita Birbasov³

Georgia Institute of Technology, Atlanta, GA, 30332

Designing interplanetary missions is an iterative process, with a tight coupling between mission analysis and vehicle design. The ability to simultaneously process both analyses early on can provide significant benefits in both the concept formulation and in overall mission feasibility. To implement such a simultaneous analysis process, there is a need to model interplanetary trajectories rapidly while maintaining an acceptable level of fidelity. Many simple patched conics tools make too many assumptions to satisfactorily address all relevant constraints and objectives, while full-fidelity trajectory design tools often require more time and expertise than is available in early conceptual studies. Hence, some form of middle-ground modeling tool would provide value.

This paper introduces the development of such a tool using Python. This tool, Keplerian Analysis for Versatile Evaluation of Arbitrary Trajectories (KAVEAT), implements a patched conics method that enforces continuity of state (time, mass, position, velocity) through all transitions between various spheres of influence on the trajectory. The trajectory is constructed in an object-oriented manner, dynamically assembling unique maneuvers such as departure burns, heliocentric transfers (ballistic or with electric propulsion), and planetary flybys (powered or unpowered). Each maneuver has its own specific set of basic inputs needed to define it. The tool then takes these defined maneuvers, strings them together creating all the interdependencies required, and solves for the ideal trajectory using a gradient-based optimizer.

I. Nomenclature

DSM	=	deep space maneuver
i	=	inclination
m	=	mass
r_a	=	radius of apoapse
r_p	=	radius of periapse
r_{soi}	=	radius of sphere of influence
SOI	=	sphere of influence
V_∞	=	hyperbolic excess speed
Δv	=	change in velocity
ϑ	=	true anomaly
ω	=	argument of periapsis
Ω	=	right ascension of the ascending node, or, longitude of the ascending node

¹ Aerospace Engineer, Space Mission Analysis Branch, NASA Langley Research Center

² Mission Architect, Advanced Concepts Office, NASA Marshall Space Flight Center, AIAA Member

³ Graduate Research Assistant, Aerospace Systems Design Lab, AIAA Student Member

II. Introduction

Interplanetary spacecraft missions are some of the most complicated types of missions to design. They are highly constrained by the positions of the planets, launch windows, launch vehicle capabilities, extreme deep space environments and more. Designers must balance all of these constraints in order to create a spacecraft capable of completing the target mission. In this process, vehicle modeling is typically mostly decoupled from the mission trajectory. Mission analysis will be run, a few key parameters will be brought over to use in vehicle sizing, and then results from that sizing are used to update the mission trajectory. This leads to a highly iterative workflow for tightly coupled problems such as sizing the thermal and power systems of an interplanetary spacecraft or designing an electric propulsion spacecraft. Being able to simultaneously size vehicle and trajectory design would lead to faster, more efficient spacecraft designs.

On the mission analysis side, mission designers have many different tools at their disposal. Many are similar variations of each other, tailored to the specific needs of the given mission. Most of these tools are specialized, focused on modeling one type of trajectory, like solving Lambert problem or low thrust maneuvers or lining up sequences of planetary flybys. With this specialization and the increasing complexity of mission concepts, it is often necessary to use multiple tools to create one cohesive trajectory.

The challenge with needing to use multiple tools to create one trajectory is that establishing consistency across the different tools can be very difficult. While trajectory tools are all based off of the same fundamental physics relationships, different tools use different formulations, parameterizations, or optimizations. Tying several different tool outputs together can be quite the task in and of itself. The other option is to use a high-fidelity tool like Copernicus which can model the majority of all types of trajectories, but which also requires a significant learning curve to utilize beneficially and produces detailed results that are excessive for a pre-Phase A study.

The goal of the Keplerian Analysis for Versatile Execution of Arbitrary Trajectories (KAVEAT) tool is to address this capability gap. KAVEAT supports the intermixing of a wide variety of different maneuver classes, including low and high thrust burns, to support diverse advanced concept vehicles and missions. KAVEAT provides a unified environment for seamless mission modeling in a single tool, without requiring the advanced knowledge required to perform a more detailed mission analysis.

Being able to model the advanced trajectories is the first step in being able to design the vehicle and trajectory simultaneously. The second step is to be able to size a vehicle based on that trajectory. Previous research has worked to alleviate the need to run both disciplines simultaneously through the training of neural nets on various finite burn models. There is a gap in the research for a tool that is capable of sizing in-space vehicles and designing their trajectories simultaneously for any type of interplanetary mission.

The DYnamic Rocket EQUation Tool (DYREQT) is a tool that has the potential to be able to do such an interdisciplinary analysis. Co-developed between Georgia Tech's ADSL and NASA Marshall's Advanced Concepts Office (ACO), DYREQT is capable of providing generalized synthesis of space transportation architectures. It allows users to bring in and call modules that represent different aspects of the spacecraft architecture design process. This ability allows for all forms of models, physics or empirical based, to be brought in to represent any type of space vehicle. It uses openMDAO to enable this flexibility as well as provides efficient numerical algorithms to aid in tool efficiency [1], [2]. DYREQT can be implemented for a diverse range of uses, including technology assessment [3] and was built to be able to support the later implementation of the integration of vehicle and trajectory optimization [1]. KAVEAT is being built specifically with future integration into DYREQT in mind. KAVEAT and DYREQT working together will enable simultaneous mission and vehicle sizing.

This paper will discuss the development process of KAVEAT, both its current status and anticipated future progress, as well as demonstrate some of the benefits of the tool's unique capabilities.

III. Background

When designing a trajectory, there are two main approaches to take. The first is to use a low fidelity patched conics approach, which makes several simplifying assumptions to solve each separate leg of the trajectory as a two-body problem, with the bodies represented as point masses [4]. The second is to use a fully implemented trajectory design tool such as AGI's Systems Tool Kit (STK) or NASA's Copernicus that can model all of the forces the spacecraft is subject to.

A. Low Fidelity Tools

The advantage of the simple patched conics approach is that it involves a set of simple equations that can be implemented and solved rapidly. The disadvantage of this approach is that it is impossible to know the exact state vector (time, position, velocity) for a given point on the trajectory. There are discontinuities in the trajectory that come

from assumptions regarding spheres of influence (SOI). A planet's SOI is assumed to be negligible when dealing with a sun-centric frame but is considered infinite when it is the center of the frame [4]. Additionally, the patched conics approximation cannot handle any type of maneuver other than impulsive. This leaves the patched conics method unsuited for modeling finite burn trajectories or for plotting any sort of trajectory.

MIDAS is one such low fidelity patched conics interplanetary trajectory optimization tool developed by JPL. It is meant for early concept mission design and to verify other trajectory optimization methods. MIDAS focuses on optimizing "event times" – ballistic coast durations, or when to perform a specific maneuver. The main limitations of MIDAS are that it can only handle up to 8 deep space maneuvers and 8 intermediate flyby or rendezvous occurrences, and it can only analyze impulsive maneuvers [5].

In addition to basic patched conics equations, there are also many tools designed to solve low thrust electric propulsion trajectories. These are specialized tools specifically for such maneuvers and are often limited in their abilities to solve the rest of a trajectory.

The Heliocentric Interplanetary Low Thrust Trajectory Optimization Program (HILTOP) is a Phase A tool for the optimization of low thrust electric propulsion interplanetary missions. It is capable of optimizing for a wide range of design targets for all types of electric propulsion systems, including both nuclear and solar systems. There can be up to 4 celestial bodies included in the primary trajectory, with 1 primary target and 3 intermediate targets. However, the 3 intermediate targets cannot be modeled to have mass or to impact the trajectory so they must be relatively small for the trajectory to remain accurate. An additional 5 small targets can be included after a flyby of the primary target as well. HILTOP is also only capable of modeling interplanetary trajectories as low thrust. While it can represent orbit insertion and flyby maneuvers, each of those are done impulsively. HILTOP cannot simulate a low thrust orbit insertion or powered flyby [6].

The Mission Analysis Low-Thrust Optimization Tool (MALTO) is a medium fidelity tool capable of modeling low thrust interplanetary missions with some high thrust capability [7]. The tool uses the Sims-Flanagan model to represent a continuous burn trajectory around a single central body. The outputs from this tool can be integrated with the Variational calculus Trajectory Optimization Program (VARITOP), which is the base program for HILTOP [8].

Higher fidelity tools designed for low thrust trajectories include Mystic, the Optimal Trajectories by Implicit Simulation program (OTIS), and the Spacecraft N-body Analysis Program (SNAP). SNAP is generally built to be planetocentric, with limited heliocentric capabilities. Mystic has the special capability to identify and utilize gravity assist maneuvers without the user explicitly including them. All of these tools are capable of solving n-body problems [7] and use more advanced analysis methods than required for Phase A studies.

B. High Fidelity Tools

In addition to the specialized tools, there are tools capable of modeling mixed-type trajectories that include both high and low thrust maneuvers as well as both planetocentric and heliocentric travel.

A pair of these tools are IPREP and IPOST. IPOST is a tool capable of supporting a wide range of analysis phases that gives the user a wide range of knobs to turn to control the level of detail in the analysis. IPOST allows users to select a sequence of events, identify the independent and dependent variables, set constraints and outline the cost function. The level of fidelity can also be altered for different phases of flight. Since IPOST requires more detailed initial inputs to help define a mission, IPREP was created to serve both as an input generator for IPOST but also as a stand-alone pre-Phase A tool. IPREP can also solve ballistic, high- and low-thrust trajectories but at a lower fidelity level. It can be used to perform quick scans of the trade space to find optimal opportunities. IPREP uses simple conic equations and the Lambert method for ballistic transfers, and the Chebytop polynomial representation for low thrust maneuvers. Flybys are treated as impulsive ΔV maneuvers. LPREP is similar to IPREP but enables lunar mission trajectories. Both can be used to feed into IPOST for better initial conditions [9].

High fidelity tools like STK or Copernicus are valuable because they can model all the forces of the solar system to a very high degree of accuracy. They are capable of modeling all sorts of different trajectories, including those for low thrust electric propulsion systems [10], [11]. However, these tools require a great deal of knowledge of orbital mechanics to correctly design a trajectory, and they require a wide variety of inputs and parameters to be set in the tool. The requirement for such advanced knowledge to operate these tools sets a high barrier of entry for an engineer in the conceptual design phase who is trying to size the overall mission and vehicle.

There is a demonstrated need for a tool with a fidelity level somewhere between the two abovementioned approaches, that is capable of being integrated with a vehicle sizing tool. This paper will discuss the development of such a tool, using a modified patched conics approach to generate a wide variety of trajectories with the state vector available at every point.

IV. Formulation

The purpose of KAVEAT is to be a medium fidelity, rapidly converging tool capable of solving a diverse range of interplanetary trajectories, including hybrid missions, with minimal user input required. KAVEAT allows users to piece together many different trajectory elements and maneuvers in a variety of different ways in order to create their ideal trajectory. It is based in object-oriented programming to enable this flexibility and requires very few initial parameters from the user.

KAVEAT models the trajectories using a modified patched conics approach. The basic patched conics approach breaks interplanetary trajectories into different segments, where each segment, or leg, is around a different central body. For example, a one-way mission to Mars would include an Earth-centric leg, followed by a sun centric leg, and end with a Mars centered leg. Each leg is solved using basic physics equations that assume central bodies are point masses, have infinite spheres of influence, and are the only force acting on the spacecraft for that leg. These equations often only deal with magnitudes of the quantities in question and not direction. All maneuvers are modeled as impulsive. Impulsive burns assume that the change in velocity is instantaneous, and that the spacecraft immediately changes from the initial orbit to the final one.

The modifications to the patched conics approach include the implementation of a finite SOI. This SOI, which is defined in Eq. (1) with m representing mass and r representing radius, identifies the point in the trajectory where the spacecraft transitions between having the primary body force be the sun or be the nearby planet [4]. In the tool, the spacecraft moves between heliocentric and planet centric orbits, connecting position, velocity, and time at the SOI. This reduces the inaccuracies in the ΔV calculations from the patched conics method that come from assuming the velocity at the SOI is equivalent to the v_∞ of the hyperbolic escape orbit. It also allows a fully continuous orbit to be plotted as the spacecraft moves from one central body to another. The standard patched conics method results in discontinuities in the interplanetary trajectory when the spacecraft encounters a planet, but when a finite SOI is used, the trajectory can be modeled continuously throughout the mission.

$$r_{soi} \approx \left(\frac{m_{planet}}{m_{sun}} \right)^{2/5} r_{planet/sun} \quad (\text{Ref. [4]}) \quad (1)$$

Each leg or maneuver performed along a trajectory is modeled as an event. Events can include a departure or arrival burn, flyby maneuvers, and interplanetary transfers. In KAVEAT, the calculations required for each maneuver are contained within an event class. When a user defines a trajectory, KAVEAT automatically strings together the series of events required to model it, using the input parameters provided by the user. Each event requires only a few simple input parameters, many of which can become optimization variables. Connecting two consecutive events is a state. Each state corresponds to a specific 3D position, velocity and date. The end state of the first event must match the initial state of the second event. These states ensure trajectory continuity through different events.

This formulation is enabled by OpenMDAO, an open source Python framework developed by NASA to enable multidisciplinary analysis. It provides the ability to define arbitrary arrangements of engineering analyses and then implements advanced numerical algorithms to enhance convergence and global gradient estimation. The OpenMDAO library provides the overall structure of the code as well as the optimizer and constraints manager. It provides the mechanism for the state construct, connecting the event classes. Constraints are set to ensure that all the events connect and both correspond to the shared state in between them. Once the problem has been constructed using the OpenMDAO framework, it is solved using the constraints and a user defined optimization function with the goal of minimizing some value, like ΔV or time of flight.

The base modeling functionality of KAVEAT uses the PyKEP library written by ESA. It has a number of built in functions useful for computing a trajectory, including a Lambert solver, a simple Lagrangian propagator, a Taylor propagator (for high thrust finite burns), and a Sims-Flanagan propagator (for low thrust finite burns). As KAVEAT develops, users will be able to bring in their own classes and libraries to define a trajectory, with KAVEAT providing the connection points and the optimization capability. This will enable the user to define their own maneuver classes, expanding and enhancing KAVEAT's base capabilities, provided by PyKEP.

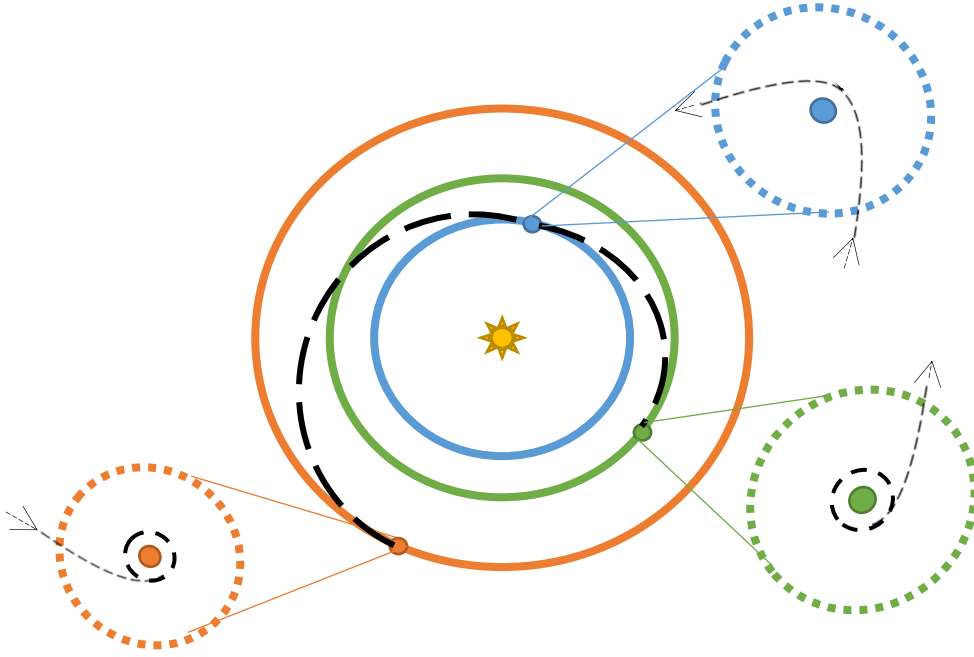


Figure 1 Sample trajectory representation

To run the code, the user sets their desired maneuvers, planets to visit, start date, as well as a few other basic orbital parameters for each maneuver. KAVEAT then reads in these inputs and sets up the optimization problem automatically. Each event definition has its own set of orbit calculations, design variables, and constraints. OpenMDAO connects all of the event inputs and outputs and handles the various constraints and overall optimization. A mission to Mars for example, may have an Earth starting orbit, a departure burn, an interplanetary coast, a Venus flyby, a second interplanetary transfer, an arrival burn, and then a Mars final orbit. The user defines these events and then KAVEAT strings them together into a complete trajectory, with the final state of one maneuver being the same as the initial state of the next. A graphical representation of this structure can be seen in Figure 1. The following subsections define how KAVEAT models the different types of events.

Once the complete trajectory has been created, it is optimized. OpenMDAO iterates on the given design variables until a feasible, optimal solution is found. This solution is then passed to a final class which breaks each maneuver leg into many smaller segments and generates the state vector (position, velocity, and time) for each sub-state so that the position of the spacecraft can be plotted semi-continuously throughout the trajectory, its position known at small enough intervals to provide valuable information for vehicle sizing.

C. Planet-centric Orbits

Parking Orbits

An interplanetary trajectory typically both begins and ends in a parking orbit. An orbit is defined by six independent parameters, often defined using the Keplerian elements: radius of periapsis (r_p), radius of apoapsis (r_a), inclination (i), argument of perigee (ω), right ascension of the ascending node (Ω), and true anomaly (θ). Often, several of these are known or selected a priori, such as r_p , r_a , and i ; for some, though, it is preferable for the simulation to simply find the best value. Specifically, it is often desirable for ω and Ω to be solved for as a function of other trajectory segments; for example, if an interplanetary departure requires a specific departure velocity vector to be achieved, ω and Ω are typically solved for as dependent variables. The fact that KAVEAT implements a finite SOI and computes interplanetary trajectories between distinct points on each planet's SOI means an analytic solution is not possible; however, OpenMDAO's efficient numerical framework is able to solve for these through an iterative patched conic method. [12]

Planetary Flyby

Flybys are important parts of many interplanetary missions, to change orbit planes or to increase or decrease heliocentric orbit energy. Many missions would not have been able to reach their destinations without the ability to perform at least one flyby.

A flyby connects two interplanetary legs of a trajectory that terminate at the flyby planet's SOI. KAVEAT solves for the incoming position vector of the flyby and then propagates the spacecraft around the planet on a hyperbolic trajectory until it exits the SOI on the other side. Constraints ensure that the spacecraft does not travel too close to the planet in question and that the flyby does in fact connect the two interplanetary legs correctly.

Orbit Transfers

As part of the future development planned for KAVEAT, orbit change maneuvers will be added. These will allow the user to specify a number of different maneuvers, including orbit raises, circularizations, and inclination changes, among others. The spacecraft will be able to transfer between one orbit and another around the same central body.

D. Interplanetary Transfers

Interplanetary transfers are the paths that a spacecraft takes between two planets' spheres of influence. The simplest examples of these are solutions to Lambert's problem, where the initial and final positions are known along with the approximate time it takes to travel between those two points, but the corresponding initial and final velocities are unknown. KAVEAT is able to solve Lambert's problem using one of PyKEP's built in functions to solve for those velocities. The output velocities are needed to compute the C_3 for the hyperbolic trajectories around the local planets on each end.

Deep Space Maneuvers

In future iterations of KAVEAT, there will be the ability to add a deep space maneuver (DSM) to a basic interplanetary transfer. The DSM will be optimally placed along the trajectory as to best meet the user defined objective function (e.g. minimize total mission ΔV including arrival, departure and DSM burns). It will be modeled as an impulsive maneuver, an instantaneous change in the interplanetary trajectory.

Low Thrust Arcs

For concepts that employ low-thrust propulsion systems, propulsive forces act over prolonged periods of time, taking advantage of long interplanetary transit times and high efficiency to dramatically reduce the required departure and arrival energies. To model these, KAVEAT leverages a formulation of the method of Sims and Flanagan, breaking trajectory arcs into multiple discrete segments and applying propulsive impulses at the midpoints of each segment. [13] The specific implementation of this algorithm provided in the PyKEP library incorporates several enhancements to improve both the fidelity and the formulation of the optimization that must be solved. [14]

E. Departure and Arrival Maneuvers

Impulsive Departure

Departure burns provide the required ΔV to a spacecraft to place it on an escape orbit. The magnitude of the ΔV comes from the C_3 value determined by the required velocity at the beginning on the subsequent interplanetary leg at the planet's SOI. The ΔV is applied impulsively to the spacecraft while it's at periapse, instantaneously changing the orbit to a hyperbolic escape trajectory. The spacecraft then propagates out to the SOI, where it connects to the interplanetary leg, creating a continuous trajectory.

Impulsive Arrival

Impulsive arrivals are functionally very similar to impulsive departures but done in reverse. The incoming C_3 is used along with the desired final parking orbit to determine the magnitude of the ΔV required to enter the orbit. From the parking orbit, the ΔV is applied and then the orbit is propagated backwards through time along the incoming hyperbolic trajectory to the planet's SOI. The trajectory connects the arrival maneuver with the previous interplanetary trajectory at the SOI.

Impulsive Powered Flyby

By adding a burn at or near periapsis of a flyby, a spacecraft is able to alter its trajectory even more significantly than could be provided by an unpowered flyby. An impulsive powered flyby is a combination of an unpowered flyby and a departure burn. Like an unpowered flyby, it connects two interplanetary legs of a trajectory that terminate at the flyby planet's SOI. Also similar to an unpowered flyby, KAVEAT solves for the incoming position vector and then

propagates the spacecraft through the orbit. However, unlike an unpowered flyby, the orbit is only propagated to periapse. At periapse, an impulsive maneuver is applied, the magnitude of the ΔV being determined from the energy of the incoming hyperbola and the required C_3 of the outgoing interplanetary leg, similar to a departure burn. Then, like as for a departure burn, the new orbit is propagated out to the SOI. Finally, similar to an unpowered flyby, constraints are placed to ensure that the spacecraft does not travel too close to the planet and that the flyby does in fact connect the two interplanetary legs correctly.

Finite Maneuvers

The trajectories of high-thrust systems are often modeled with an impulsive approximation which assumes that the durations over which propulsive forces act are short enough to incur negligible gravity losses. However, “high-thrust” is not actually what drives this, but rather “high-acceleration.” Many high-thrust systems have relatively low acceleration in the early maneuvers of their missions, and so the impulsive assumption becomes less accurate.

While they have not yet been implemented, finite burn maneuvers are a prominent capability on the near-term roadmap for KAVEAT development. The approach will leverage the Taylor integration scheme as formulated in PyKEP, which allows for the rapid propagation of a spacecraft under a perturbing force.

V. Demonstration

As development proceeds on the KAVEAT core source code, a set of test cases is being developed for verification, validation, and benchmarking purposes. Initially these are being taken from basic trajectories documented in literature but will expand to include known good solutions from other analyses of similar or higher fidelity in the future. Presented here are two early examples; the first is a very simple minimum energy transfer from Earth to Mars, the second a more involved trajectory modeling a hybrid transportation system which includes both chemical and electric propulsion subsystems. Both test cases were selected because they contain key aspects that make them valuable for continued testing as KAVEAT matures.

A. Minimum Energy High-Thrust Earth-Mars Transfer

The simplest interplanetary trajectory to model is the heliocentric coasting arc of a high-thrust transfer, which is found via a Lambert solution process. This is the logical starting place for building up a set of verification cases, as it tests some of the most basic functionality within a trajectory modeling setup. For this test case, a solution from the 1998 *Interplanetary Mission Design Handbook* was selected. [15] This trajectory models a high-thrust system departing from a 400 km circular Earth orbit and inserting into a 250 x 33,793 km 1-sol orbit at Mars.

Table 1: Verification results for a high-thrust Earth-Mars transfer

Source	Departure Date	Departure ΔV	ΔV Losses	Arrival Date	Transit Time	V_∞ Earth Departure	V_∞ Mars Arrival
George [15]	7/19/2020	3,899 m/s	120 m/s	1/15/2021	180 d	3,665 m/s	3,154 m/s
KAVEAT	7/19/2020	3,749 m/s	0	1/15/2021	180 d	3,684 m/s	3,186 m/s

Table 1 shows close, though not exact, agreement between KAVEAT and the published result. The largest discrepancy between these two is the gravity losses that were computed in [15]; ignoring these, the results are much closer. As discussed previously, finite burn maneuvers appear early on the roadmap for KAVEAT, and so this test case will also serve as verification when that capability comes online.

Another difference that may explain the small discrepancies in the V_∞ values is that KAVEAT computes the Lambert solution between distinct points on the SOI of each planet rather than from the planet centers. Also, the time of flight from [15] is modeled from departure burn to arrival burn, so the time of flight used in KAVEAT’s Lambert solver is reduced by the transit time between periapsis and SOI at both planets.

B. Hybrid Propulsion System Earth-Mars Transfer

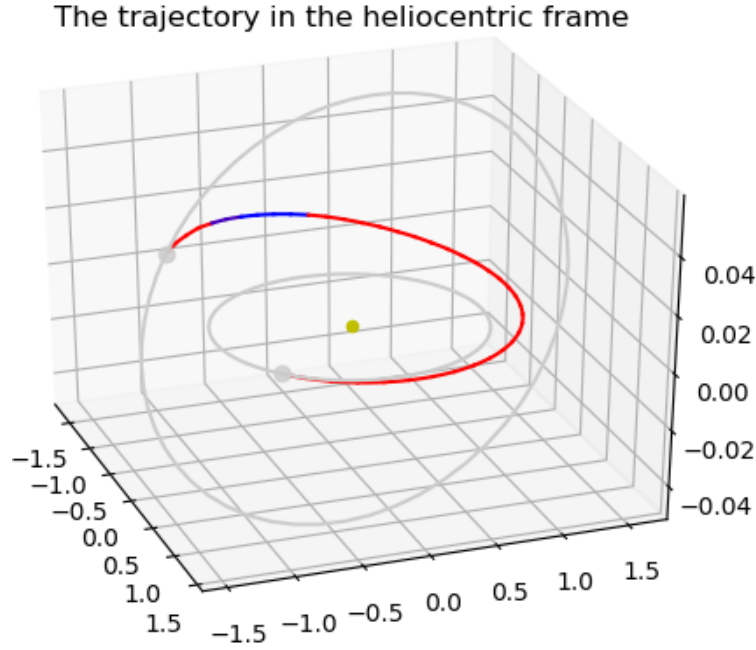
Although KAVEAT is still in the preliminary stages of development, it was desirable to identify an early test case that would demonstrate some of the intended modeling flexibility that motivated its development in the first place. For this purpose, the trajectory of a hybrid propulsion system published by Chai et. al. was selected due to its combination of both high-thrust and low-thrust maneuvers. [16] The specific trajectory selected for comparison was a 2039 crewed mission to Mars, defined at a high level in Table 2. The parking orbit at Earth was assumed as a 400 x 400,000 km Lunar Distant High Earth Orbit (LDHEO), while at Mars a 5-sol parking orbit with periapse of 250 km altitude was assumed. The low thrust propulsion system was assumed to be a solar electric propulsion system.

Table 2: Verification results for a hybrid propulsion Earth-Mars transfer

Source	Departure Date	Departure Mass, kg	Arrival Date	Transit Time	V_{∞} Earth Departure	V_{∞} Mars Arrival	Inserted Mass, kg
Chai [16]	8/3/2039	99590	9/6/2040	400	1,414 m/s	1,201 m/s	75,444
KAVEAT	8/3/2039	99590	9/6/2040	400	1,414 m/s	1,201 m/s	78,666

A qualitative comparison of the trajectory solution from [16] and the KAVEAT computed trajectory depicted in Figure 2 and Figure 3 shows general agreement. However, as can be seen in Table 2, the final mass predicted by KAVEAT is noticeably higher than the published result. This comes as no surprise to the authors of this paper, as the relatively immature modeling capabilities of KAVEAT prevent it from fully reproducing the intended result. Foremost among the discrepancies in current modeling capabilities is that KAVEAT's low-thrust trajectory capabilities do not yet implement a solar power model in which power available to the electric propulsion system varies by the inverse square of distance from the sun. In effect, the lack of such a model means KAVEAT's version of this trajectory could be interpreted as that of a nuclear electric propulsion system rather than solar electric propulsion.

Other factors that may also affect these results even once a solar power model is implemented include assumptions regarding electric thruster duty cycles and any contingency coasting periods imposed. Also, since KAVEAT models the interplanetary trajectory between distinct points on the SOI of each planet, modeling the lunar gravity assist explicitly would be required to provide an estimate of where the spacecraft would exit the Earth's SOI on departure

**Figure 2: Interplanetary trajectory, with thrusting (red) and coasting (blue) periods**

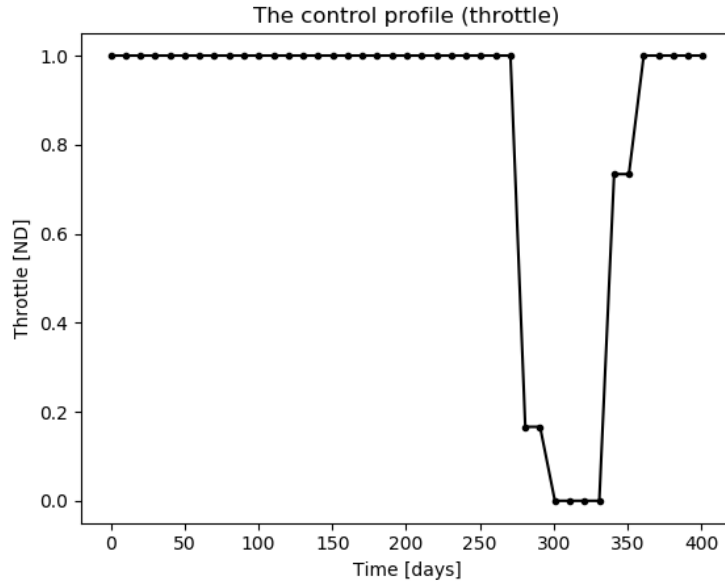


Figure 3: Interplanetary throttle profile for electric propulsion system

VI. Conclusion

Interplanetary mission design is a tightly coupled process between the trajectory and vehicle design process. Current approaches to developing a new mission involve an iterative process between the two disciplines, with incremental changes being made each time until a solution is converged upon. There is a capability gap for a tool capable of simultaneous sizing of both the trajectory and vehicle elements. KAVEAT, introduced in this paper, aims to be one half of that tool. KAVEAT will be capable of modeling a wide range of trajectory options with relatively few user inputs and coming to a mid-fidelity solution quickly. The tool is currently under development, with basic maneuvers already included in the code. These maneuvers include ballistic interplanetary transfers, impulsive arrivals and departures, and planet orbits and planetary flybys. Future development will add the ability to include low thrust interplanetary transfers, high powered DSMs, finite arrival and departure burns that take gravity and other losses into account, as well as orbit change maneuvers. These will enable KAVEAT to become a flexible multi-use tool capable of modeling a diverse range of trajectories. When KAVEAT is combined with DYREQT, a vehicle sizing tool, they will be able to work in tandem to simultaneously size and optimize both the spacecraft vehicle and its trajectory.

References

- [1] D. J. Trent, "Integrated Architecture Analysis and Technology Evaluation for Systems of Systems Modeled at the Subsystem Level," Ph.D Thesis, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, 2017.
- [2] S. J. Edwards, M. J. Diaz, D. N. Mavris and D. J. Trent, "A Model-Based Framework for Synthesis of Space Transportation Architectures," in *AIAA Space Forum*, Orlando, 2018.
- [3] A. Sudol, S. J. Edwards and D. N. Mavris, "Technology Assessment for Launch Vehicle Concepts Using the DYnamic Rocket EQUation Tool (DYREQT)," in *AIAA Space Forum*, Orlando, 2018.
- [4] J. E. Prussing and B. A. Conway, "Chapter 9 Interplanetary Mission Analysis," in *Orbital Mechanics*, 2nd ed., New York, Oxford University Press, 2013, pp. 154-172.
- [5] C. Sauer, "Preliminary Draft of Users Guide to MIDAS," NASA JPL, Pasadena, 1991.
- [6] F. I. Mann and J. L. Horsewood, "Program Manual for HILTOP, A Heliocentric Interplanetary Low Thrust Trajectory Optimization Program. Part 1 - User's Guide," NASA-CR-143894, 1974.

- [7] L. D. Kos, T. Polsgrove, R. C. Hopkins, D. Thomas and J. A. Sims, "Overview of the Development for a Suite of Low-Thrust Trajectory Analysis Tools," in *AIAA/AAS Astrodynamics Specialist Conference*, Keystone, CO, 2006.
- [8] T. Polsgrove, L. Kos, R. Hopkins and T. Crane, "Comparison of Performance Predictions for New Low-Thrust Trajectory Tools," in *AIAA/AAS Astrodynamics Specialist Conference*, Keystone, CO, 2006.
- [9] P. E. Hong, P. D. Kent, D. W. Olson and C. A. Vallado, "Interplanetary Program to Optimize Simulated Trajectories (IPOST). Volume 1 - User's Guide," NASA-CR-189653, 1992.
- [10] "STK Help," Analytical Graphics Inc, 2020. [Online]. Available: <https://help.agi.com/stk/index.htm>. [Accessed 11 September 2020].
- [11] J. Williams, "Copernicus Version 4.6 User Guide," NASA JSC, Houston, 2018.
- [12] S. P. Parvathi and R. V. Ramanan, "Direct transfer trajectory design options for interplanetary orbiter missions using an iterative patched conic method," *Advances in Space Research*, vol. 59, pp. 1763-1774, 2017.
- [13] J. A. Sims and S. N. Flanagan, "Preliminary Design of Low-Thrust Interplanetary Missions," in AAS 99-338, 1999.
- [14] D. Izzo, "PYGMO AND PYKEP: OPEN SOURCE TOOLS FOR MASSIVELY PARALLEL OPTIMIZATION IN ASTRODYNAMICS (THE CASE OF INTERPLANETARY TRAJECTORY OPTIMIZATION)," in *Proceedings of the International Conference on Astrodynamics Tools and Techniques*, 2012.
- [15] L. E. George and L. D. Kos, "Interplanetary Mission Design Handbook: Earth-to-Mars Mission Opportunities and Mars-to-Earth Return Opportunities 2009-2024," NASA/TM-1998-208533, 1998.
- [16] P. R. Chai, R. G. Merrill and M. Qu, "Mars Hybrid Propulsion System Trajectory Analysis Part I: Crew Missions," in *AIAA SPACE Forum*, Pasadena, CA, 2015.