Acta Astronautica 211 (2023) 416–430

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

Research paper

Particle Swarm Optimization-based co-state initialization for low-thrust minimum-fuel trajectory optimization

Grant R. Hecht, Eleonora M. Botta [*]

Department of Mechanical and Aerospace Engineering, University at Buffalo, Buffalo, NY, United States

ARTICLE INFO

Keywords:
Low-thrust trajectory optimization
Minimum-fuel trajectory optimization
Homotopic continuation
Particle Swarm Optimization
Optimal control

ABSTRACT

In this paper, Particle Swarm Optimization with energy-to-fuel continuation is proposed for initializing the co-state variables for low-thrust minimum-fuel trajectory optimization problems in the circular restricted three-body problem. Particle Swarm Optimization performs a search of the solution space by minimizing the weighted sum of squares of the two-point boundary-value problem final boundary condition residuals for the minimum-energy problem. Next, an energy-to-fuel homotopy is employed to transition the minimum-energy trajectory to a minimum-fuel trajectory, starting from the generated guess. The proposed methodology is applied to two low-thrust transfer problems in the Earth–Moon system: a transfer from a geostationary transfer orbit to an L1 halo orbit, and a transfer from an L2 halo orbit to an L1 halo orbit. The resulting minimum-fuel trajectories are validated with the literature. It is demonstrated that the methodology can successfully generate guesses for the initial co-state variables which converge to a solution for both scenarios. A strategically chosen particle swarm size is shown to improve the efficiency of the methodology. The proposed approach is of simple implementation, can be easily extended to other trajectory optimization problems, facilitates the discovery of multiple candidate trajectories, and does not require a user-provided guess, all of which are advantageous features for the preliminary phase of mission design.

1. Introduction

Low-thrust propulsion has gained much attention in recent years and is now often considered the best option for a wide range of space missions. The associated reduction in required propellant mass compared to traditional chemical propulsion results in cost savings and an increase in the feasible payload ratio. Low-thrust propulsion does, however, require long, continuous thrusting arcs and trajectories often spanning many revolutions about a large body, which can lead to high-dimensional optimization problems with many local minima. Low-thrust spacecraft trajectory optimization problems most commonly focus on minimizing the time of flight, fuel required, or combinations of these conflicting objectives. As the name suggests, minimum-time trajectory optimization is necessary when mission requirements specify that the spacecraft must reach its destination in a timely manner, regardless of the fuel required, as is often the case when transporting human life or trying to acquire time-sensitive scientific data. On the other hand, minimum-fuel trajectory optimization is useful when time is not such an important constraint and it is desired to reduce the required fuel, which allows for increasing the feasible payload ratio and decreasing mission costs. Many approaches for optimizing low-thrust spacecraft trajectories have been proposed and applied successfully. Typically, approaches can be separated into two categories known as *direct* and *indirect* methods. Hybrid approaches also exist which combine the characteristics of direct and indirect methods [1]. A broad review of the state-of-the-art techniques employed for spacecraft trajectory optimization can be found in work by Chai et al. [2].

Direct trajectory optimization methods formulate the trajectory optimization problem by parameterizing the state and control variables, forming a nonlinear programming (NLP) problem [1]. These methods can be further categorized into shooting, collocation, dynamic programming, or differential inclusion-based methods [2]. When considering trajectories spanning tens to hundreds of days with many separate, continuous thrusting arcs – as is common for low-thrust propulsion –, this parameterization can lead to very high-dimensional problems which may be intractable if not strategically formulated, due to the approaching curse of dimensionality. Additionally, this parameterization generally enforces the control to take some predetermined form, which can result in suboptimal solutions. Nevertheless, direct methods continue to be an attractive approach for solving low-thrust trajectory optimization problems for several reasons. When compared to indirect methods, direct methods can more easily incorporate complicated constraints and tend to be much less sensitive to the quality of the initial

[*] Corresponding author.
E-mail addresses: granthec@buffalo.edu (G.R. Hecht), ebotta@buffalo.edu (E.M. Botta).

https://doi.org/10.1016/j.actaastro.2023.06.021
Received 4 October 2022; Received in revised form 14 March 2023; Accepted 13 June 2023
Available online 17 June 2023

guess supplied to the algorithm [1]. Furthermore, physical intuition can be exploited when an initial guess for the solution is determined and supplied to an NLP solver. Application of direct methods to the solution of low-thrust trajectory optimization problems is discussed in Refs. [3–5].

Indirect methods apply Calculus of Variations (COV) and Pontryagin's Maximum Principle (PMP) to analytically derive the necessary conditions for optimality, introducing additional co-state variables and formulating the problem as a Two-Point Boundary-Value Problem (TPBVP) [6]. Collocation or finite element-based methods can then be used to approximate the solution of the TPBVP, or a shooting-based method may be used to solve the TPBVP directly [2]. Collocation or finite element-based approaches can lead to useful physical insight about the problem while providing approximate solutions [1]. Alternatively, shooting-based methods result in a fully continuous optimal control law for the low-thrust spacecraft trajectory. Unfortunately, the convergence of the indirect shooting-based approach is typically not guaranteed and is highly dependent on the selection of the initial co-state variables used to initialize the algorithm [1,6].

Typically, the co-state variables are abstract quantities with no clear physical significance and the convergence radius of indirect shooting methods is often very small. Therefore, determining an initial choice of co-state variables to initialize a shooting routine can be a daunting task. To mitigate some of this difficulty, several methods have been proposed to assist in initializing the co-state variables. One such approach, first proposed by Dixon and Biggs [7,8] and termed Adjoint Control Transformation (ACT), involves relating the co-state variables to alternative variables of physical significance, thereby decreasing the sensitivity of the problem and allowing for the use of physical insight when selecting a guess. Application of ACT to problems in astrodynamics is discussed, for example, in Refs. [9–12]. An alternative approach involves formulating an approximation of the initial co-state variables to avoid the need for a user-provided guess. For example, within the field of astrodynamics, Thorne and Hall approximated the initial co-state variables by solving a similar problem with an analytical solution by assuming zero gravity and a constant spacecraft mass [13], whereas Lee et al. derived an approximate relationship between the co-state variables and the initial constrained physical state based on the behavior of the initial co-states for a range of optimal trajectories [14,15]. Unfortunately, these techniques are restricted to planar motion or transfers of less than one revolution. Another possibility to estimate the initial co-state variables involves employing a solution acquired via direct trajectory optimization. Examples of such methods include a technique that exploits approximate solutions acquired via shape-based methods and nonlinear least squares [6], and an approach that allows one to relate the Lagrange multipliers of the Karush–Kuhn–Tucker equations associated with various collocation methods to the co-states of the optimal control problem [16–20]. Of course, a primary difficulty of such approaches is that a solution must be found through the appropriate direct trajectory optimization method first, a task that can itself be difficult and requires a guess to be formulated for the state and control time histories in most cases. Furthermore, once a solution is discovered via direct trajectory optimization, successful estimation of the co-state variables is not guaranteed.

A particularly interesting approach, proposed by Pontani and Conway, employs PSO to select co-state variables which minimize the sum of the absolute value of the final boundary condition residuals [21,22], solving the TPBVP directly. Such an approach is highly desirable as it can, in principle, be applied to any TPBVP and does not require a user to provide any guess (although box constraints must be defined to limit the search space). Unfortunately, due to the extreme sensitivity of the final boundary condition residuals to perturbations in the initial co-state variables for many indirect trajectory optimization problems, this initial application of PSO was limited to planar minimum-time transfers in the two-body problem, which reduces the number of unknown variables and constraints.

In this paper, a method is proposed which builds upon the work by Pontani and Conway [21,22] and allows for solving non-planar low-thrust minimum-fuel trajectory optimization problems with the indirect shooting approach by leveraging PSO to initialize the co-state variables. PSO co-state initialization and an energy-to-fuel homotopy technique are coupled [23] to widen the convergence radius of the TPBVP and reduce the difficulty of the parameter optimization problem to be solved by PSO. PSO generates co-states for the easier-to-solve minimum-energy problem, before transitioning the solution to minimum-fuel through continuation, all without the need of a user-provide guess. Herein, the proposed methodology is presented in the framework of the Circular Restricted Three Body Problem (CR3BP), and important considerations in regard to solution optimality and computational performance are discussed. Case studies, based on previous works [3,24], are presented which shed light on the promising performance of the proposed methodology while allowing for its validation. Guidance on the selection of the particle swarm size, a heuristic parameter of the optimization algorithm which has a significant effect on co-state initialization performance, is also provided.

The remainder of this paper is organized as follows. In Section 2, the problem is formulated, beginning with the CR3BP dynamics, and then deriving the indirect minimum-fuel TPBVP with an energy-to-fuel homotopy through the application of COV and PMP. The proposed method of co-state initialization is discussed in Section 3, along with the single shooting algorithm used to robustly transition PSO-initialized co-state values, corresponding to the minimum-energy problem, to a solution of the minimum-fuel problem. In Section 4, two minimum-fuel trajectory optimization scenarios (i.e., a transfer from a geostationary transfer orbit to L1 halo orbit, and a transfer from an L2 halo orbit to an L1 halo orbit) are employed to validate the methodology, demonstrate the performance of the proposed method, and analyze the effect of the particle swarm size on convergence. Finally, conclusions are made which summarize the achievements of this work and future steps.

2. Indirect optimal control formulation

In this Section, the indirect optimal control TPBVP is formulated for the minimum-fuel problem with terminal state constraints and a fixed time of flight. This is done using the CR3BP dynamics model, which is provided and discussed. Throughout the TPBVP formulation, an energy-to-fuel homotopy technique is incorporated by introducing a perturbed energy term to the minimum-fuel cost function, before the Hamiltonian is formed and the analytical necessary conditions of optimality are derived.

2.1. Circular restricted three body problem dynamics

The CR3BP dynamics describe the motion of a body in a synodic (rotating) reference frame under the influence of two primary bodies which travel along co-planar circular paths centered about the barycenter of the system [25]. The mass of the third body is indicated with m and is assumed to be much lower than those of the primary bodies m_1 and m_2; therefore it has no effect on their motion. In the following, all variables are non-dimensionalized such that the angular velocity of both primary bodies, the sum of the masses of the primary bodies, and the distance between the primary bodies are set to unity to improve numerical stability. If the third body is a spacecraft with the capability to apply propulsive thrust in any direction, the equations of motion describing the evolution of the spacecraft's position and velocity in a synodic reference frame, along with its mass are given by [12]

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \hat{\boldsymbol{\alpha}}, u) = \begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{v}} \\ \dot{m} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{g}(\mathbf{r}) + \mathbf{h}(\mathbf{v}) + uT_{max}\hat{\boldsymbol{\alpha}}/m \\ -uT_{max}/c \end{bmatrix} \tag{1}$$

where $\mathbf{r}$ and $\mathbf{v}$ are the non-dimensionalized position and velocity vectors of the spacecraft, T_{max} is its maximum capable thrust, and c is the propellant exhaust velocity, which is assumed to be constant and given by $c = I_{sp} g_0$, with I_{sp} the specific impulse of the propellant and $g_0 = 9.81\ \mathrm{m/s^2}$. The control variables are u and $\hat{\boldsymbol{\alpha}}$, where $u \in [0, 1]$ is the thrust throttling factor and $\hat{\boldsymbol{\alpha}}$ is a unit vector representing the direction at which thrust is applied. Functions $\mathbf{g}(\mathbf{r})$ and $\mathbf{h}(\mathbf{v})$ are given by [24]

$$\mathbf{g}(\mathbf{r}) = \begin{bmatrix} r_x - \frac{(1-\mu)(r_x+\mu)}{r_1^3} - \frac{\mu(r_x+\mu-1)}{r_2^3} \\ r_y - \frac{(1-\mu)r_y}{r_1^3} - \frac{\mu r_y}{r_2^3} \\ -\frac{(1-\mu)r_z}{r_1^3} - \frac{\mu r_z}{r_2^3} \end{bmatrix} \tag{2}$$

$$\mathbf{h}(\mathbf{v}) = \begin{bmatrix} 2v_y & -2v_x & 0 \end{bmatrix}^T \tag{3}$$

where subscripts x, y, and z denote the components of the position and velocity vector along the corresponding Cartesian direction of the synodic reference frame and $\mu = m_2/(m_1 + m_2)$ is the non-dimensionalized mass of the second primary body. Additionally, r_1 and r_2 denote the non-dimensionalized distance of the spacecraft from the first and second primary bodies respectively and are given by

$$r_1 = \sqrt{(r_x + \mu)^2 + r_y^2 + r_z^2} \tag{4}$$

$$r_2 = \sqrt{(r_x + \mu - 1)^2 + r_y^2 + r_z^2} \tag{5}$$

2.2. Indirect minimum-fuel trajectory optimization using energy-to-fuel homotopy

For the minimum-fuel problem, it is desired to find a solution of Eq. (1) which minimizes the cost function

$$J_{mf} = \frac{T_{max}}{c} \int_{t_i}^{t_f} u\, dt \tag{6}$$

where t_i and t_f correspond to the initial and final time of the trajectory. A known characteristic of minimum-fuel trajectories is the bang–bang nature of their control profile [12,23,24], which results in discontinuities in the throttling factor u along the trajectory and can severely restrict the convergence radius of numerical methods. To mitigate this issue, Bertrand and Epenoy proposed the use of an *energy-to-fuel* homotopy continuation for transforming solutions from the discontinuity-free minimum-energy problem to the minimum-fuel problem [23]. A perturbed energy term is introduced into the cost function by means of the perturbation parameter ϵ, which is gradually reduced until its effect is nulled. Employing this approach, the new cost function is given by

$$J = \frac{T_{max}}{c} \int_{t_i}^{t_f} [u - \epsilon u(1-u)]\, dt \qquad \epsilon \in [0, 1] \tag{7}$$

To begin the derivation of the TPBVP, we proceed by forming the Hamiltonian as [24]

$$H = \boldsymbol{\lambda}^T \mathbf{f}(\mathbf{x}, \hat{\boldsymbol{\alpha}}, u) + \frac{T_{max}}{c} [u - \epsilon u(1-u)] \tag{8}$$

$$= \boldsymbol{\lambda}_r^T \mathbf{v} + \boldsymbol{\lambda}_v^T \left[\mathbf{g}(\mathbf{r}) + \mathbf{h}(\mathbf{v}) + \frac{u T_{max}}{m} \hat{\boldsymbol{\alpha}} \right] - \lambda_m \frac{u T_{max}}{c} + \frac{T_{max}}{c} [u - \epsilon u(1-u)] \tag{9}$$

where $\boldsymbol{\lambda}^T = \left[\boldsymbol{\lambda}_r^T, \boldsymbol{\lambda}_v^T, \lambda_m \right]$ is the introduced co-state vector, which evolves in time according to

$$\dot{\boldsymbol{\lambda}} = \begin{bmatrix} \dot{\boldsymbol{\lambda}}_r \\ \dot{\boldsymbol{\lambda}}_v \\ \dot{\lambda}_m \end{bmatrix} = -\left(\frac{\partial H}{\partial \mathbf{x}} \right)^T = \begin{bmatrix} -\mathbf{G}^T \boldsymbol{\lambda}_v \\ -\boldsymbol{\lambda}_r - \mathbf{H}^T \boldsymbol{\lambda}_v \\ \frac{u T_{max}}{m^2} \boldsymbol{\lambda}_v^T \hat{\boldsymbol{\alpha}} \end{bmatrix} \tag{10}$$

$$\mathbf{G} = \frac{\partial \mathbf{g}(\mathbf{r})}{\partial \mathbf{r}} \tag{11}$$

$$\mathbf{H} = \frac{\partial \mathbf{h}(\mathbf{v})}{\partial \mathbf{v}} \tag{12}$$

where the non-zero terms of $\mathbf{G}$ and $\mathbf{H}$ are

$$G_{1,1} = 1 - \frac{1-\mu}{r_1^3} + \frac{3(1-\mu)(x+\mu)^2}{r_1^5} - \frac{\mu}{r_2^3} + \frac{3\mu(x+\mu-1)^2}{r_2^5} \tag{13}$$

$$G_{2,2} = 1 - \frac{1-\mu}{r_1^3} + \frac{3(1-\mu)y^2}{r_1^5} - \frac{\mu}{r_2^3} + \frac{3\mu y^2}{r_2^5} \tag{14}$$

$$G_{3,3} = -\frac{1-\mu}{r_1^3} + \frac{3(1-\mu)z^2}{r_1^5} - \frac{\mu}{r_2^3} + \frac{3\mu z^2}{r_2^5} \tag{15}$$

$$G_{1,2} = G_{2,1} = \frac{3(1-\mu)(x+\mu)y}{r_1^5} + \frac{3\mu(x+\mu-1)y}{r_2^5} \tag{16}$$

$$G_{1,3} = G_{3,1} = \frac{3(1-\mu)(x+\mu)z}{r_1^5} + \frac{3\mu(x+\mu-1)z}{r_2^5} \tag{17}$$

$$G_{2,3} = G_{3,2} = \frac{3(1-\mu)yz}{r_1^5} + \frac{3\mu yz}{r_2^5} \tag{18}$$

$$H_{1,2} = -H_{2,1} = 2 \tag{19}$$

Applying the weak form of PMP, the optimal thrust direction unit vector is characterized by [6]

$$\hat{\boldsymbol{\alpha}}^* \in \arg\min_{\hat{\boldsymbol{\alpha}}} H \tag{20}$$

which, noting that $u T_{max}/m \geq 0$ and $\boldsymbol{\lambda}_v \neq 0$ in general, gives

$$\hat{\boldsymbol{\alpha}}^* = -\frac{\boldsymbol{\lambda}_v}{\lambda_v} \tag{21}$$

where λ_v is the Euclidean norm of $\boldsymbol{\lambda}_v$, and $-\boldsymbol{\lambda}_v$ is defined as the primer vector [26]. Substituting this result into Eq. (9) and defining a switching function S as [24]

$$S = -\frac{c}{m} \lambda_v - \lambda_m + 1, \tag{22}$$

the Hamiltonian can now be written as

$$H = \boldsymbol{\lambda}_r^T \mathbf{v} + \boldsymbol{\lambda}_v^T \left[\mathbf{g}(\mathbf{r}) + \mathbf{h}(\mathbf{v}) \right] + \frac{u T_{max}}{c} (S - \epsilon + \epsilon u) \tag{23}$$

Again applying the weak form of PMP, the optimal thrust throttling factor is given by [24]

$$u^* = \begin{cases} 0 & S > \epsilon \\ \frac{\epsilon - S}{2\epsilon} & -\epsilon \leq S \leq \epsilon \\ 1 & S < -\epsilon \end{cases} \tag{24}$$

Note that the relations between the co-states and optimal control variables $\hat{\boldsymbol{\alpha}}^*$ and u^*, given in Eqs. (21) and (24), are equivalent to Lawden's primer vector control law when $\epsilon = 0$.[1]

To complete the formulation of the trajectory optimization problem, terminal constraints placed on the physical state variables are given by

$$\begin{matrix} \mathbf{r}(t_i) - \mathbf{r}_i = 0 & \mathbf{v}(t_i) - \mathbf{v}_i = 0 & m(t_i) - 1 = 0 \\ \mathbf{r}(t_f) - \mathbf{r}_f = 0 & \mathbf{v}(t_f) - \mathbf{v}_f = 0 & \end{matrix} \tag{25}$$

where $\mathbf{r}_i$, $\mathbf{v}_i$, $\mathbf{r}_f$, and $\mathbf{v}_f$ are the desired initial and final position and velocity vectors, and the initial mass has been scaled to one. Noting that the final mass of the spacecraft is unconstrained, a transversality condition provides the last constraint, which is given by [27]

$$\lambda_m(t_f) = 0 \tag{26}$$

Substituting Eq. (21) into Eq. (1) and combining with Eq. (10), a 14-dimensional system of first order ordinary differential equations is obtained

$$\dot{\mathbf{y}} = \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{g}(\mathbf{r}) + \mathbf{h}(\mathbf{v}) - \boldsymbol{\lambda}_v u T_{max}/(\lambda_v m) \\ -u T_{max}/c \\ -\mathbf{G}^T \boldsymbol{\lambda}_v \\ -\boldsymbol{\lambda}_r - \mathbf{H}^T \boldsymbol{\lambda}_v \\ -\lambda_v u T_{max}/m^2 \end{bmatrix} \tag{27}$$

[1] Lawden's primer vector control law is a well-known relationship between the co-states, optimal throttling factor and thrust direction when the indirect optimal control approach is employed to solve minimum-fuel problems [26].

which, when paired with the optimal throttling factor given in Eq. (24) and the constraints given in Eqs. (25) and (26), fully define the TPBVP. The solution of the TPBVP is the set of initial co-state variables which satisfy the terminal constraints. Here, it is important to note that solutions of the derived TPBVP only imply satisfaction of the first-order necessary conditions of optimality [12]. Therefore, without further investigation of the second-order sufficient conditions of optimality, solutions are only guaranteed to be stationary or extremal solutions of the optimal control cost function given in Eq. (7), also known as candidates of optimality [28]. Furthermore, one should note that multiple local extrema may exist, resulting in multiple solutions of the TPBVP.

3. TPBVP solution methodology

Solving the derived TPBVP requires a good guess of the initial co-state variables $\lambda(t_i)$. The methodology proposed here involves the use of the PSO algorithm to first initialize the co-state variables for the minimum-energy problem, corresponding to the perturbation parameter $\epsilon = 1$ in Eq. (24). The PSO-initialized co-states are then used to seed a single shooting procedure with homotopy continuation, where a trust-region nonlinear solver [29] iteratively discovers the solution to the TPBVP for decreasing values of the perturbation parameter until a solution corresponding to $\epsilon = 0$ is found. A flowchart for the algorithm is provided in Appendix.

In this Section, the methodology employed to integrate the state, co-state, and State Transition Matrix (STM) differential equations while performing switching detection is discussed. The proposed method for initializing the co-state variables with PSO for the minimum-energy problem (i.e., for $\epsilon = 1$) is also presented, along with an approach to perform single shooting with a homotopy continuation. Along the way, important considerations in regard to programming language selection and computational efficiency are made.

3.1. Numerical integration with switching detection

When integrating the state and co-state differential equations, in addition to the STM differential equations when single shooting is performed, it is important to ensure that a sufficiently high degree of accuracy is maintained, so that convergence can be achieved. Due to the piecewise continuous function which defines the optimal throttling factor (see Eq. (24)), integration error can grow about the points at which switching of the throttling factor occurs, and can severely restrict the convergence of a shooting algorithm. However, this can be avoided if the points at which the switching occurs are explicitly determined and numerically integrated up to. Additionally, both the PSO co-state initialization and single shooting phases require numerous repeated numerical integrations of the differential equations. Performing numerical integration in a programming language which exhibits high computational performance with a differential equation solving suite that provides high-order Runge–Kutta algorithms and robust event detection is thus desirable. For these reasons, the Julia [30] programming language and the Julia-implemented differential equation solving package, *DifferentialEquations.jl* [31], were chosen for all numerical computation and integration in this work.

Julia is a dynamically typed language with syntax similar to that of MATLAB or Python, thereby lending itself to quick development. Historically, a known characteristic of this category of languages is the slow execution time due to the required use of an interpreter, an issue which Julia circumvents through the use of a just-in-time (JIT) LLVM-based compiler. Therefore, although Julia is a dynamic language with easy-to-read syntax, it allows producing software with performance that rivals statically typed languages like C and FORTRAN. Furthermore, *DifferentialEquations.jl* is highly optimized and provides a plethora of modern differential equation solvers and features, including robust event detection schemes, termed *callbacks*, which can employ high-order interpolants to check whether an event has occurred at multiple points backwards in time. Through the application of *DifferentialEquations.jl* and its *continuous callback* feature (i.e., continuous event detection), numerical integration can be performed efficiently by exploiting an adaptive time-stepping method while robustly detecting throttling factor switching to within 64-bit floating point number precision.

3.2. Co-state initialization with PSO

Beginning with the inception of the PSO algorithm detailed in the work of Kennedy and Eberhart [32], many have proposed modifications with improved convergence characteristics [33,34]. The term PSO has thus evolved to encompass a class of heuristic global optimization algorithms rather than any one strictly defined algorithm. For the sake of reproducibility, in this work, an implementation of PSO in the Julia programming language is employed, which is based on the MATLAB PSO algorithm [35].

To employ PSO to initialize the co-state variables for the minimum-energy problem (See step (A) in the Appendix), a nonlinear, unconstrained optimization problem is formulated as

$$\underset{\lambda(t_i)}{\text{Minimize}} \quad J_{PSO}(\lambda(t_i)) = \mathbf{e}^T(t_f)\mathbf{W}\mathbf{e}(t_f) \tag{28}$$

where $\mathbf{W} \in \mathbb{R}^{7\times7}$ is a diagonal weighting matrix and $\mathbf{e}(t_f)$ is the final time boundary condition residual given by

$$\mathbf{e}(t_f) = \begin{bmatrix} \mathbf{r}(t_f) - \mathbf{r}_f \\ \mathbf{v}(t_f) - \mathbf{v}_f \\ \lambda_m(t_f) \end{bmatrix} \tag{29}$$

which is computed by integrating the state and co-state differential equations from t_i to t_f as described before, starting from an initial state vector constructed as a concatenation of the constrained initial state and the current guess for the co-state values at the initial time

$$\mathbf{y}(t_i) = \begin{bmatrix}\mathbf{r}_i^T & \mathbf{v}_i^T & 1.0 & \lambda^T(t_i)\end{bmatrix}^T \tag{30}$$

Through computational investigation, the implemented single shouting routine, discussed in the following Section, was found to be most sensitive to large residuals in the final position. Therefore, the weighting matrix was defined as

$$\mathbf{W} = \text{Diag}\left(\begin{bmatrix}10 & 10 & 10 & 1 & 1 & 1 & 1\end{bmatrix}\right) \tag{31}$$

such that the squared position residuals are weighted by a factor of ten, while the remaining weights are left at unity.

Due to the stochastic nature of the PSO algorithm, some particles may travel to positions that result in trajectories that pass below the surface of a primary body, which is physically non-realizable. Therefore, the integration routine was set up to halt the integration of a given trajectory if the distance between the spacecraft and the surface of a primary body was smaller than zero at any time.

When employing PSO for co-state initialization, especially when a large swarm size is required and/or the cost function is expensive to evaluate, it is important to take advantage of the parallelizable nature of the algorithm to reduce compute time. In fact, for each iteration of PSO, the cost function must be re-evaluated for each particle. In general, the evaluation of the cost function for any given particle is independent of all other particles in the swarm, which makes the process well suited for shared or distributed memory parallelism, which was taken advantage of in this work.

3.3. Single shooting with continuation

After co-state initialization with PSO, a single shooting routine based on a trust-region nonlinear solver provided by *NLsolve.jl* [29] is employed for solving the TPBVP (See step (C) in the Appendix). To do so, the nonlinear 7-dimensional vector function which is desired to be solved is defined as

$$\mathbf{Z}(\lambda(t_i), \epsilon) = \mathbf{e}(t_f) \tag{32}$$

where a solution corresponds to $\mathbf{e}(t_f) = \mathbf{0}_{7\times1}$. In other words, it is desired to find the initial co-state vector $\lambda(t_i)$ which, for a given value of $\epsilon \in [0,1]$ – integrating Eq. (27) from t_i to t_f as described before – results in a final time boundary condition residual of $\mathbf{e}(t_f) = \mathbf{0}_{7\times1}$. Henceforth, $\mathbf{Z}(\lambda(t_i),\epsilon)$ in Eq. (32) is referred to as the *shooting function*. The trust-region algorithm is used to iteratively solve Eq. (32) for values of the perturbation parameter decreasing from one to zero according to the continuation law [24]

$$\epsilon_j = \frac{j^2-1}{N^2-1} \qquad j = N, N-1, \ldots, 2, 1 \tag{33}$$

where N is the number of iterations that are desired to be performed for transitioning the perturbation parameter from one to zero. This value can be chosen somewhat arbitrarily, but it is important to consider that higher values of N result in smaller changes in the perturbation parameter at each iteration, providing a smoother transition from the minimum-energy to the minimum-fuel cost function, albeit at a higher computational cost. Throughout this work, a value of $N = 25$ was chosen to balance numerical stability and computational efficiency. The PSO-initialized co-states are used to seed the first step in the continuation process only (see step (B) in the Appendix). Then, subsequent steps in the continuation process employ the solution from the previous step as the initial guess to begin iteration of the trust-region method.

It is important to note that the trust-region algorithm requires that the Jacobian of the shooting function, i.e., $\partial\mathbf{Z}(\lambda(t_i),\epsilon)/\partial\lambda(t_i)$, also be computed. Multiple methods exist to compute this Jacobian (e.g., automatic differentiation and finite difference), among which the STM was selected for this work. The STM is defined as

$$\boldsymbol{\Phi}(t_f,t_i) = \frac{\partial\mathbf{y}(t_f)}{\partial\mathbf{y}(t_i)} \tag{34}$$

To compute the STM, the 14^2 STM differential equations must also be integrated along with the state and co-state differential equations in Eq. (27) according to

$$\dot{\boldsymbol{\Phi}}(t,t_i) = \mathbf{F}\boldsymbol{\Phi}(t,t_i) = \left.\frac{\partial\dot{\mathbf{y}}}{\partial\mathbf{y}}\right|_t \boldsymbol{\Phi}(t,t_i), \qquad \boldsymbol{\Phi}(t_i,t_i) = \mathbf{I}_{14\times14} \tag{35}$$

where $\mathbf{F}$ is the Jacobian of Eq. (27) evaluated at time t, the analytical expression of which is provided by Zhang et al. [24]. Additionally, when discontinuities are present in the shooting function, as is the case for $\epsilon = 0$ when switching occurs (see Eq. (24)), the STM across the discontinuity must also be computed as [12]

$$\boldsymbol{\Psi}(t_n) = \frac{\partial\mathbf{y}(t_n^+)}{\partial\mathbf{y}(t_n^-)} = \mathbf{I}_{14\times14} + \left(\dot{\mathbf{y}}\Big|_{t_n^+} - \dot{\mathbf{y}}\Big|_{t_n^-}\right)\left(\frac{\partial S}{\partial\mathbf{y}}\frac{1}{\dot{S}}\right) \tag{36}$$

where t_n^- and t_n^+ represent the time immediately before and after the discontinuity occurred, respectively, and

$$\frac{\partial S}{\partial\mathbf{y}} = \begin{bmatrix}\mathbf{0}_{1\times6} & \frac{c}{m^2}\lambda_v & \mathbf{0}_{1\times3} & -\frac{c}{m}\frac{1}{\lambda_v}\lambda_v^T & -1\end{bmatrix} \tag{37}$$

$$\dot{S} = \frac{c}{m}\left(\lambda_r + \mathbf{H}^T\lambda_v\right)^T\frac{\lambda_v}{\lambda_v} \tag{38}$$

For a trajectory with M discontinuities between t_i and t_f, the state transition matrix is then given by [24]

$$\boldsymbol{\Phi}(t_f,t_i) = \boldsymbol{\Phi}(t_f,t_M^+)\boldsymbol{\Psi}(t_M)\boldsymbol{\Phi}(t_M^-,t_{M-1}^+)\boldsymbol{\Psi}(t_{M-1})\ldots\boldsymbol{\Phi}(t_2^-,t_1^+)\boldsymbol{\Psi}(t_1)\boldsymbol{\Phi}(t_1^-,t_i) \tag{39}$$

Once the STM has been propagated from t_i to t_f along with the state and co-state variables, the Jacobian of the shooting function is constructed from components of the STM, such that

$$\frac{\partial\mathbf{Z}(\lambda(t_i),\epsilon)}{\partial\lambda(t_i)} = \begin{bmatrix}\frac{\partial\mathbf{r}(t_f)}{\partial\lambda_r(t_i)} & \frac{\partial\mathbf{r}(t_f)}{\partial\lambda_v(t_i)} & \frac{\partial\mathbf{r}(t_f)}{\partial\lambda_m(t_i)} \\ \frac{\partial\mathbf{v}(t_f)}{\partial\lambda_r(t_i)} & \frac{\partial\mathbf{v}(t_f)}{\partial\lambda_v(t_i)} & \frac{\partial\mathbf{v}(t_f)}{\partial\lambda_m(t_i)} \\ \frac{\partial\lambda_m(t_f)}{\partial\lambda_r(t_i)} & \frac{\partial\lambda_m(t_f)}{\partial\lambda_v(t_i)} & \frac{\partial\lambda_m(t_f)}{\partial\lambda_m(t_i)}\end{bmatrix} \tag{40}$$

Table 1
Constant parameters.

Constant	Value	Units
Mass Parameter μ	1.21506038×10^{-2}	–
Grav. Constant g_0	9.81	m/s^2
Time Unit (TU)	3.75162997×10^5	s
Length Unit (LU)	3.84400000×10^5	km
Velocity Unit (VU)	1.02462131	km/s
Mass Unit (MU)	m_i	kg

4. Results

In this Section, case studies are presented to validate the methodology, analyze the performance of the proposed method for co-state initialization, as well as to assess the convergence of the PSO-initialized co-states to solutions of the minimum-fuel TPBVP for varying particle swarm sizes. The proposed methodology is applied to determine low-thrust minimum-fuel transfers in the following two scenarios in the Earth–Moon system: (1) from a geostationary transfer orbit (GTO) to an L1 halo orbit and (2) from an L2 halo orbit to a L1 halo orbit. Constant parameters used for both scenarios, including the time, length, speed, and mass units employed in the non-dimensionalization of the CR3BP dynamics, are provided in Table 1. Note that the mass unit (which is a unique value for a given scenario) is defined as the initial mass of the spacecraft m_i. The chosen scenarios provide problems of moderate difficulty to benchmark our approach while allowing for validation and comparison to previous works [3,24,36–39].

Due to the stochastic nature of PSO, 100 co-state initialization trials are performed for swarm sizes ranging from 250 to 5000 particles in order for statistical conclusions to be made about the performance of the algorithm. The position of the kth particle in the swarm is defined in the 7-dimensional co-state space as

$$\lambda_k(t_i) = \begin{bmatrix}\lambda_{r,k}^T(t_i) & \lambda_{v,k}^T(t_i) & \lambda_{m,k}(t_i)\end{bmatrix}^T \tag{41}$$

The swarm is initialized with particle positions uniformly distributed between upper and lower bounds such that $\lambda_r(t_i) \in [-40,40]$, $\lambda_v(t_i) \in [-2,2]$, and $\lambda_m(t_i) \in [0,2]$. This is motivated by previous work that showed that the initial co-states generally lie near the origin when the non-dimensionalized CR3BP dynamics are employed [12,24]. Through experimentation, it was confirmed that initializing the PSO swarm such that the particles are initially located closer to the origin results in more favorable convergence characteristics. It is important to note that these are not constraints placed on the positions of particles, but are rather bounds used to define the distribution from which starting particle positions are sampled. Once the swarm is initialized, particles are then free to travel anywhere in the search space. As there is – in general – no guarantee that all solutions to a given TPBVP will lie within boundaries that are known *a priori*, the search space must not be too restrictive. Therefore, in this work the search space is defined by $\lambda_r(t_i) \in [-100,100]$, $\lambda_v(t_i) \in [-10,10]$, and $\lambda_m(t_i) \in [0,10]$.

Convergence criteria are chosen such that a PSO trial is halted if the objective function value does not improve by more than 1×10^{-6} over the course of 50 iterations, or if a maximum run time of 30 min is met. Throughout the following case studies, the effect of different PSO swarm sizes only is analyzed for the discussion to remain tractable. The remaining PSO parameters are held at their default values (as defined by MathWorks [35]), with the exception of the minimum adaptive neighborhood size fraction, which is set to 0.05.

For all numerical integration performed throughout the case studies, Verner's "most efficient" Runge–Kutta 9(8) method was employed, with an absolute and relative tolerance of 10^{-14}, along with its corresponding 9-th order interpolant for switching detection [40]. All computations were performed on a workstation PC with specifications relevant to this work provided in Table 2. At each iteration of PSO, the evaluations of the cost function for each particle were computed in parallel using shared memory parallelism on 32 of the available 64 physical cores within the CPU.

Table 2
PC specifications.

Specification	Value
CPU	AMD Ryzen Threadripper Pro 3995WX
RAM	32 GB DDR4 1600 MHz
Operating System	Windows 10 Enterprise
Julia	Version 1.7.0

4.1. Scenario 1: GTO to L1 halo orbit

For the GTO to L1 halo orbit transfer problem, the spacecraft was chosen to have an initial mass $m_i = 1500$ kg, a maximum thrust $T_{max} = 10$ N, and a specific impulse $I_{sp} = 3000$ s.[2] In the minimum-fuel problem, the time of flight (TOF) was fixed at 8.6404 days. The terminal physical boundary conditions for the TPBVP, which identify the initial position of the spacecraft and the final point targeted on the halo orbit, were defined as [24]

$$\mathbf{r}_i = \begin{bmatrix}-0.0194885115 & -0.0160334798 & 0.0\end{bmatrix}^T \text{ LU}$$
$$\mathbf{v}_i = \begin{bmatrix}8.9188819237 & -4.0817936888 & 0.0\end{bmatrix}^T \text{ VU}$$
$$\mathbf{r}_f = \begin{bmatrix}0.8233851820 & 0.0 & -0.0222775563\end{bmatrix}^T \text{ LU}$$
$$\mathbf{v}_f = \begin{bmatrix}0.0 & 0.1341841703 & 0.0\end{bmatrix}^T \text{ VU}$$

such that the GTO had periapsis and apoapsis altitudes of $h_p = 400$ km and $h_a = 35864$ km, respectively, and the L1 halo orbit had an out-of-plane amplitude of 8000 km.

4.1.1. Minimum-fuel TPBVP solutions

As discussed before, multiple local extrema of the optimal control cost function, and therefore multiple solutions of the TPBVP may exist. A total of six solutions of the minimum-fuel TPBVP (corresponding to $\epsilon = 0$) were found. These are labeled Trajectories A through F, and are shown in Fig. 1, with thrusting arcs depicted in red and coasting arcs depicted in blue. It is clear that each of the trajectories follows different paths and thrusting and coasting arcs are placed at different locations along the transfers. It is observed that Trajectories A, E, and F (in Figs. 1(a), 1(e), and 1(f)) exhibit much shorter coasting arcs during either the initial spiraling about the Earth or the final leg of the trajectory when approaching the L1 halo orbit, compared to the remaining trajectories. It is also observed that each trajectory exhibits a unique number of revolutions about the Earth before the final transfer to the halo orbit.

Table 3 displays the initial co-state variables corresponding to each of the solutions, along with the total change in mass, i.e. fuel required, which is computed as $\Delta m = [1 - m(t_f)] \times 1500$ kg. As can be seen by observing the Δm column of Table 3, Trajectory B corresponds to the most optimal minimum-fuel trajectory found, while the remaining solutions correspond to other extremal solutions of the minimum-fuel cost function. The total number of revolutions about the Earth is shown in the final column of Table 3. The unique number of revolutions for each solution suggests that each trajectory may be locally optimal for a given number of revolutions about the Earth. It is noted that the initial co-states corresponding to Trajectory B, along with Δm, align with a previously found solution by Zhang et al. [24].

For Trajectory B, the time histories of the Keplerian orbital elements and vehicle mass along the transfer are presented in Fig. 2. Here, variations of the orbital elements and mass that occur while the spacecraft is thrusting or coasting are shown in red or blue, respectively. The effect that the applied thrust has on the evolution of the orbital elements can be clearly appreciated; for example, the semi-major axis undergoes important increases during thrusting periods. The existence of 5 revolutions about the Earth is confirmed from the true anomaly θ results. Furthermore, the increasing influence of the lunar gravity on the motion of the spacecraft as it nears the L1 halo orbit is observed as the orbital elements visibly change along the final coasting arc.

4.1.2. Co-state initialization

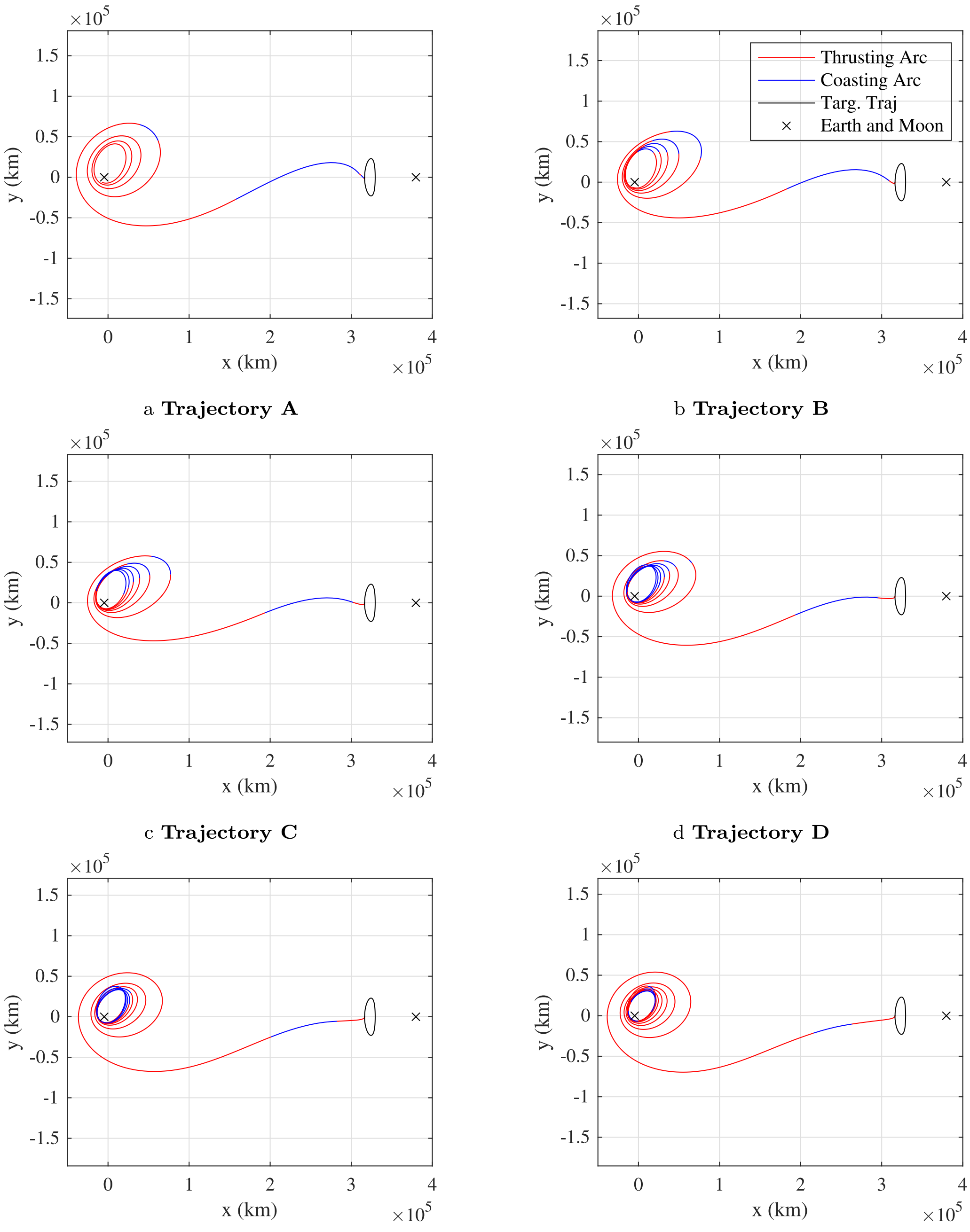
In this Section, the performance of the proposed method for converging to solutions of the derived TPBVP which satisfy all boundary conditions is analyzed. For the purpose of this discussion, a trial is considered to have converged if the ∞-norm of the TPBVP final boundary condition residual is reduced to less than 1×10^{-10}, i.e., $\|\mathbf{e}(t_f)\|_\infty = \max\{|e_i(t_f)| : i = 1, 2, \ldots, 7\} < 1 \times 10^{-10}$.

For each of the investigated swarm sizes, Table 4 displays the percentage of trials for which the initialized co-states converged to a solution of the minimum-fuel ($\epsilon = 0$) TPBVP, the average final value of the PSO objective function (see Eq. (28)), the average time and objective function evaluations before PSO convergence, and the total number of trials which were stopped prematurely due to reaching the time limit of 30 min. The percentage of converged trials was found to be lowest for the smallest number of particles and to increase with swarm size, reaching a maximum of 64% converged trials with a swarm size of 4000 particles, before reducing slightly for the largest swarm size. A similar trend of improvement was observed in the average value of the final PSO objective function and the number of function evaluations. The average time to converge increases with the swarm size, approaching the maximum time allocated to PSO (i.e., 30 min per trial) for larger swarm sizes. Similarly, the number of trials which were halted prematurely due to the enforced PSO evolution time limit increases as the number of particles within the swarm increases.

These results suggest that for swarm sizes smaller than 1000 particles, early stagnation of objective function improvement (due to the limited number of particles exploring the solutions space) results in premature convergence of the PSO algorithm. For larger swarm sizes, ranging from 2000 to 4000 particles, PSO does not suffer from premature convergence and the ability of the algorithm to produce guesses for the initial co-states which converge to solutions of the TPBVP is improved greatly. These larger swarm sizes are more effective at exploring the solution space and are able to reduce the PSO objective function further, before improvement stalls and the algorithm converges or the PSO evolution time limit is reached. It is also important to note that these larger swarm sizes do come with an additional computational cost: the objective function must be evaluated many more times per PSO iteration, resulting in longer periods spent waiting for a guess to be generated. Furthermore, this increased computational expense reduces the number of PSO iterations (i.e., updates of the particle's positions) which can be performed within the swarm evolution time limit, which can lead to a poor exploration of the solution space and more frequent premature halting of the optimization (depending on the maximum time limit imposed). For a swarm size of 5000 particles, we see the greatest effect of this computational cost: 82 of the 100 trials performed were stopped prematurely due to attainment of the maximum PSO evolution time of 30 min. It is expected that this trend would continue for even larger swarm sizes unless more time was allocated for PSO co-state initialization.

If it is desired to select a swarm size which maximizes the percentage of guesses that converge to a solution in a unique run, it is clear that 4000 particles is the optimal choice of the swarm sizes investigated. However, it should be noted that higher probabilities of convergence could be achieved performing multiple trials with lower numbers of particles, in a comparable time. For example, it was found from Table 4 that 22% of trials could produce a converged solution in an average of 2.33 min when using 250 particles. In line of principle, a total of 12 trials could be performed with 250 particles in 28 min

[2] It should be noted that the thrust-to-mass ratio is much greater than what can be achieved with current state-of-the-art electric propulsion systems. This scenario was chosen to allow validation with previous work by Zhang et al. [24]. Results remain relevant, as an additional homotopy continuation process can by employed to reduce the maximum thrust, as discussed in [24].



a **Trajectory A**

b **Trajectory B**

c **Trajectory C**

d **Trajectory D**

e **Trajectory E**

f **Trajectory F**

Fig. 1. Minimum-fuel trajectories for scenario 1.

Table 3
Minimum-Fuel TPBVP solutions for Scenario 1.

Trajectory	$\lambda_r^T(t_i)$	$\lambda_v^T(t_i)$	$\lambda_m(t_i)$	Δm (kg)	# Revs.
A	$[23.2524, 50.6272, -0.08489]$	$[-0.1546, 0.0706, -0.0002]$	0.1385	140.0	4
B	$[15.6850, 33.0313, -0.0938]$	$[-0.1020, 0.0450, -0.0002]$	0.1334	134.4	5
C	$[7.7397, 14.5669, -0.1239]$	$[-0.0466, 0.0180, -0.0001]$	0.1535	139.1	6
D	$[1.1404, -0.6176, -0.1535]$	$[-0.0006, -0.0042, -0.0002]$	0.1935	153.5	7
E	$[-9.4937, -25.2095, -0.2092]$	$[0.0738, -0.0401, -0.0002]$	0.2916	178.5	8
F	$[-24.3124, -60.4677, -0.2883]$	$[0.1801, -0.0928, -0.0002]$	0.5613	219.7	9

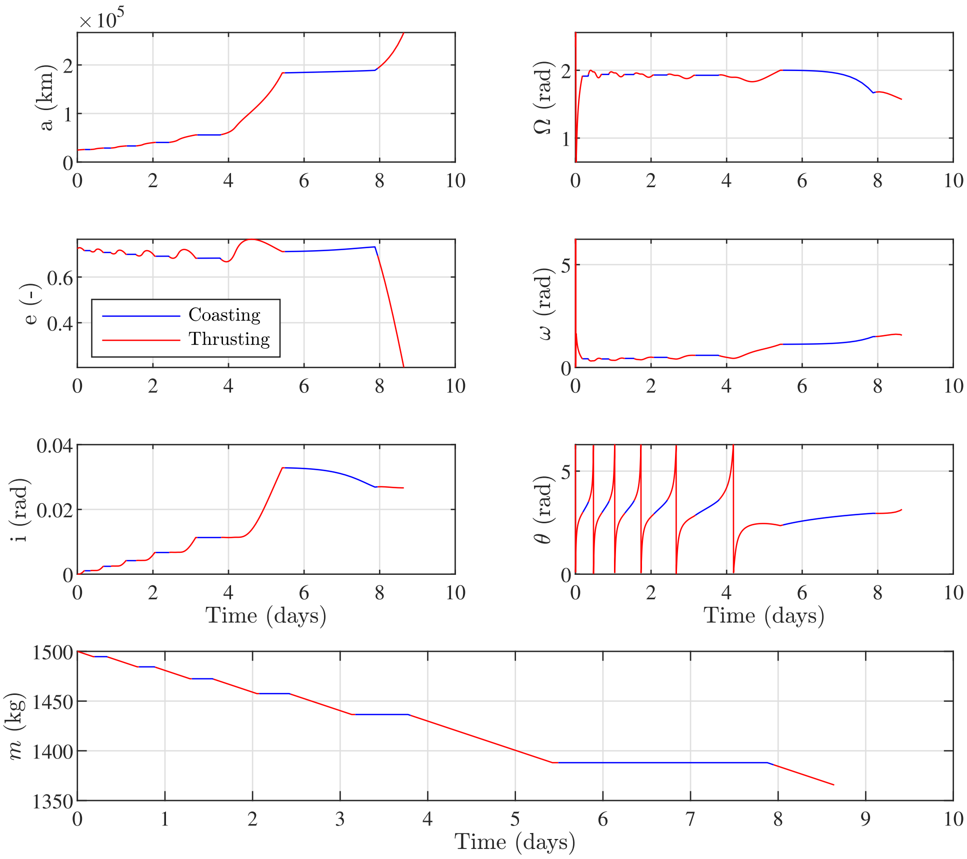


Fig. 2. Orbital elements and vehicle mass time histories for Trajectory B.

Table 4
Scenario 1 convergence characteristics for different swarm sizes.

Convergence characteristic	Swarm size							
	250	500	750	1000	2000	3000	4000	5000
Min. Fuel % Converged	22	42	41	50	63	58	64	54
Avg. Final Obj. Func.	7.93	5.74	5.10	4.28	3.84	3.50	2.45	2.97
Avg. Time to Converge (min)	2.33	7.75	13.48	15.66	22.88	25.91	28.04	28.84
Avg. Function Evaluations	3.02E5	9.24E5	1.56E6	2.00E6	2.82E6	3.06E6	3.71E6	3.58E6
Reached Time Limit	0	0	8	22	48	63	76	82

(i.e., in approximately the same time required to perform just 1 trial with 4000 particles). If one considers the approximate probability of a trial converging to a solution when employing 250 particles to be $p_{250} \approx 0.22$, the probability of at least one successful trial occurring within 12 attempted trials is given by:

$$p_{250,12} = \bigcup_{k=1}^{12} p_{250}(k) = \sum_{k=1}^{12} p_{250}(k) = 0.96 \tag{42}$$

where

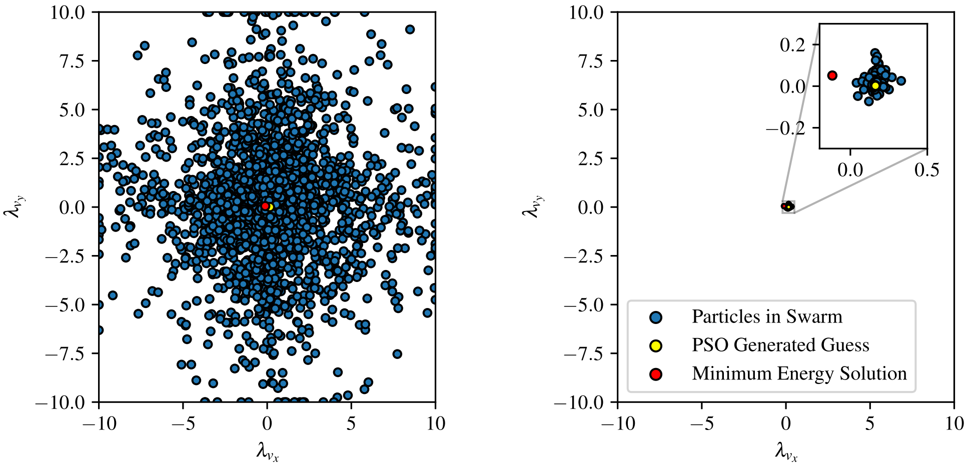
$$p_{250}(k) = \frac{12!}{k!(12-k)!} p_{250}^{k} (1 - p_{250})^{12-k} \tag{43}$$

In other words, if 12 trials are performed employing 250 particles, there is approximately a 96% chance of computing at least one solution (and a nonzero chance of computing multiple solutions) in the same time that is required to perform a single trial using 4000 particles. Thus, performing many trials using fewer particles is preferred over performing fewer trials using more particles.

4.1.3. Illustration and analysis of the solution generation process

Within this Section, the process of generating a solution of the minimum-fuel TPBVP, from co-state initialization to single shooting with homotopy continuation, is illustrated and analyzed. The discussion is focused on a single trial that converged to the most fuel-optimal solution found throughout the Scenario 1 case study, i.e., Trajectory B.

By way of example, Fig. 3 displays the distribution of 2000 particles (each represented by a blue dot) in the $x - y$ velocity co-state space after 5 PSO iterations and at PSO convergence, along with the location of the PSO-generated guess (represented by a yellow dot) and the minimum-energy solution which the generated guess converged to after performing single shooting (represented by a red dot). Additionally, Fig. 4 displays the PSO objective function value as the number of PSO iterations increases. Observing Fig. 3(a), it can be seen that the particles quickly distributed themselves across the space from their initial placement of $\lambda_{v_x}, \lambda_{v_y} \in [-2, 2]$. The particles then spent the next few hundred iterations traveling across the full search space, rapidly decreasing the PSO objective function as is shown in Fig. 4. After approximately 200 iterations, once significantly better solutions (i.e., co-states that resulted in a significantly smaller objective function value) were discovered, the rate at which the objective function improved is seen to decrease significantly and the swarm began to contract, spending more time searching the area near the better-known solutions. This slow improvement of the objective function continued until one of the



a **After 5 iterations**

b **At convergence**

Fig. 3. Distribution of PSO particles.

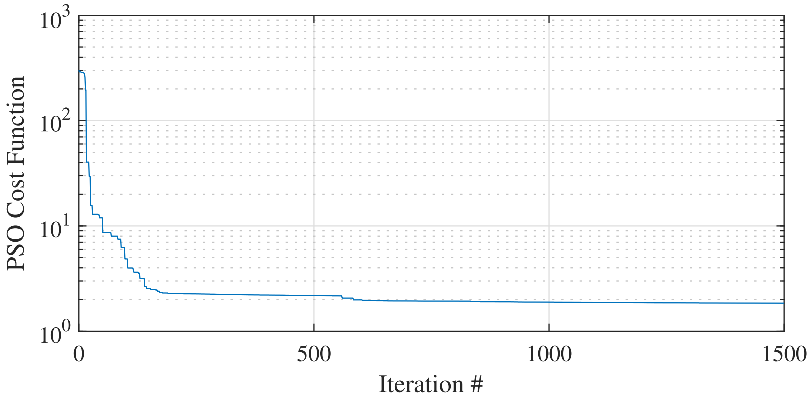


Fig. 4. PSO objective function versus iteration number.

stopping criteria of PSO was met. From Fig. 3(b), it can be observed that, upon halting of PSO, the particles had traveled to lie within a small region, which happens to be centered about the generated guess. It can also be seen that the generated guess lies near the solution to the minimum-energy problem. This confirms the capability of the algorithm of generating a guess for the co-state variables which lies near a solution of the minimum-energy TPBVP.

After PSO-based co-state initialization, the next step in the solution generation process requires employing single shooting to first converge to a solution of the minimum-energy TPBVP (i.e., with $\epsilon = 1$) starting from the PSO-generated guess, before repeatedly resolving the TPBVP while decreasing ϵ according the Eq. (33). Fig. 5 displays the evolution of Trajectory B through the continuation process as it is transitioned from minimum-energy to minimum-fuel with $\epsilon = 1.0, 0.72, 0.49, 0.25$ and 0.0. One should note that 25 steps in total are taken during the continuation process, and only 5 are selected here for the sake of illustration. Also, Fig. 6 displays the switching function and throttling factor in time for each of the trajectories displayed in Fig. 5. As can be seen in Fig. 5, all trajectories start from and arrive at the same positions as expected. The minimum-energy and minimum-fuel trajectories follow nearly identical paths until the final spiral and transfer to the halo orbit. However, it is observed that coasting arcs appear to increase as the continuation approaches a minimum-fuel trajectory. Fig. 6 illustrates the gradual introduction of the bang–bang throttling

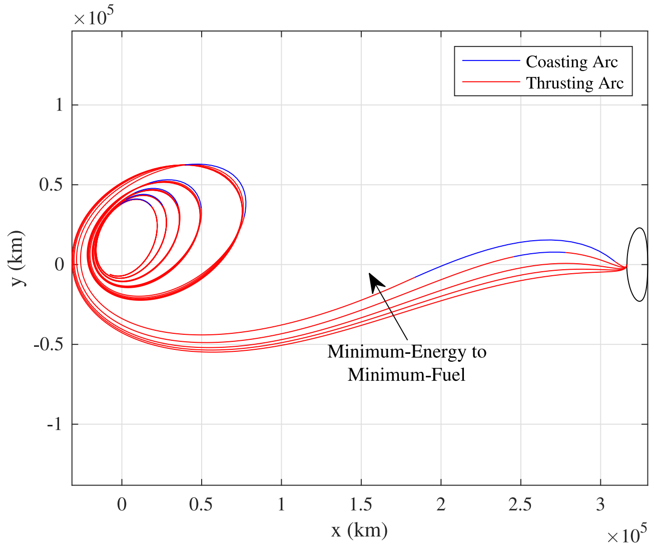


Fig. 5. Trajectory B evolution through homotopy continuation.

Table 5
Minimum-Fuel TPBVP solutions.

Trajectory	$\lambda_r^T(t_i)$	$\lambda_v^T(t_i)$	$\lambda_m(t_i)$	Δm (kg)
α	[0.12603, −0.07665, −0.05635]	[0.03999, −0.00518, −0.06410]	0.02236	35.34
β	[−0.01486, 0.01215, −0.07936]	[0.01015, 0.04457, 0.01256]	0.07632	81.28
γ	[−0.02195, 0.00659, 0.07490]	[−0.04314, 0.03615, 0.03842]	0.03489	61.27

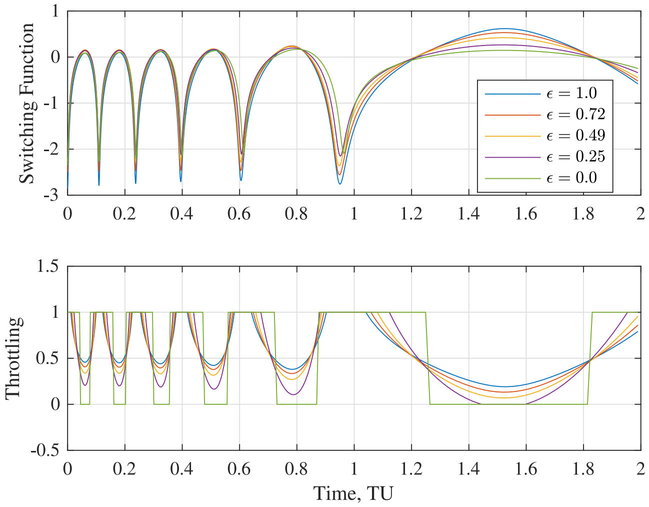


Fig. 6. Switching function and throttling factor through Trajectory B continuation.

discontinuity throughout the continuation, which is the core strength of the energy-to-fuel homotopy, which widens the convergence radius and improves the probability of PSO-generated guesses producing a solution of the minimum-fuel cost function.

4.2. Scenario 2: L2 halo orbit to L1 halo orbit

For the L2 halo orbit to L1 halo orbit transfer problem, the spacecraft was chosen to have an initial mass m_i = 2000 kg, a maximum thrust T_{max} = 1.5 N, and a specific impulse I_{sp} = 2000 s, as in Ref. [3]. The transfer duration was fixed to 12.7 days and the terminal physical boundary conditions for the TPBVP, identifying the initial state of the spacecraft on an L2 halo orbit and the target point on an L1 halo orbit, were defined as [3]

$$\mathbf{r}_i = \begin{bmatrix} 1.1599795702248494 & 0.009720428035815552 & -0.12401864915284157 \end{bmatrix}^T \text{ LU}$$

$$\mathbf{v}_i = \begin{bmatrix} 0.008477705130550553 & -0.20786307954141953 & -0.010841912833115475 \end{bmatrix}^T \text{ VU}$$

$$\mathbf{r}_f = \begin{bmatrix} 0.8484736688482315 & 0.005064888634636 82 & 0.17343680487577373 \end{bmatrix}^T \text{ LU}$$

$$\mathbf{v}_f = \begin{bmatrix} 0.005241131023638693 & 0.26343491250951045 & -0.008541420325316247 \end{bmatrix}^T \text{ VU}$$

These correspond to initial and target halo orbits with periods of 14.2 and 11.2 days respectively.

4.2.1. Minimum-fuel TPBVP solutions

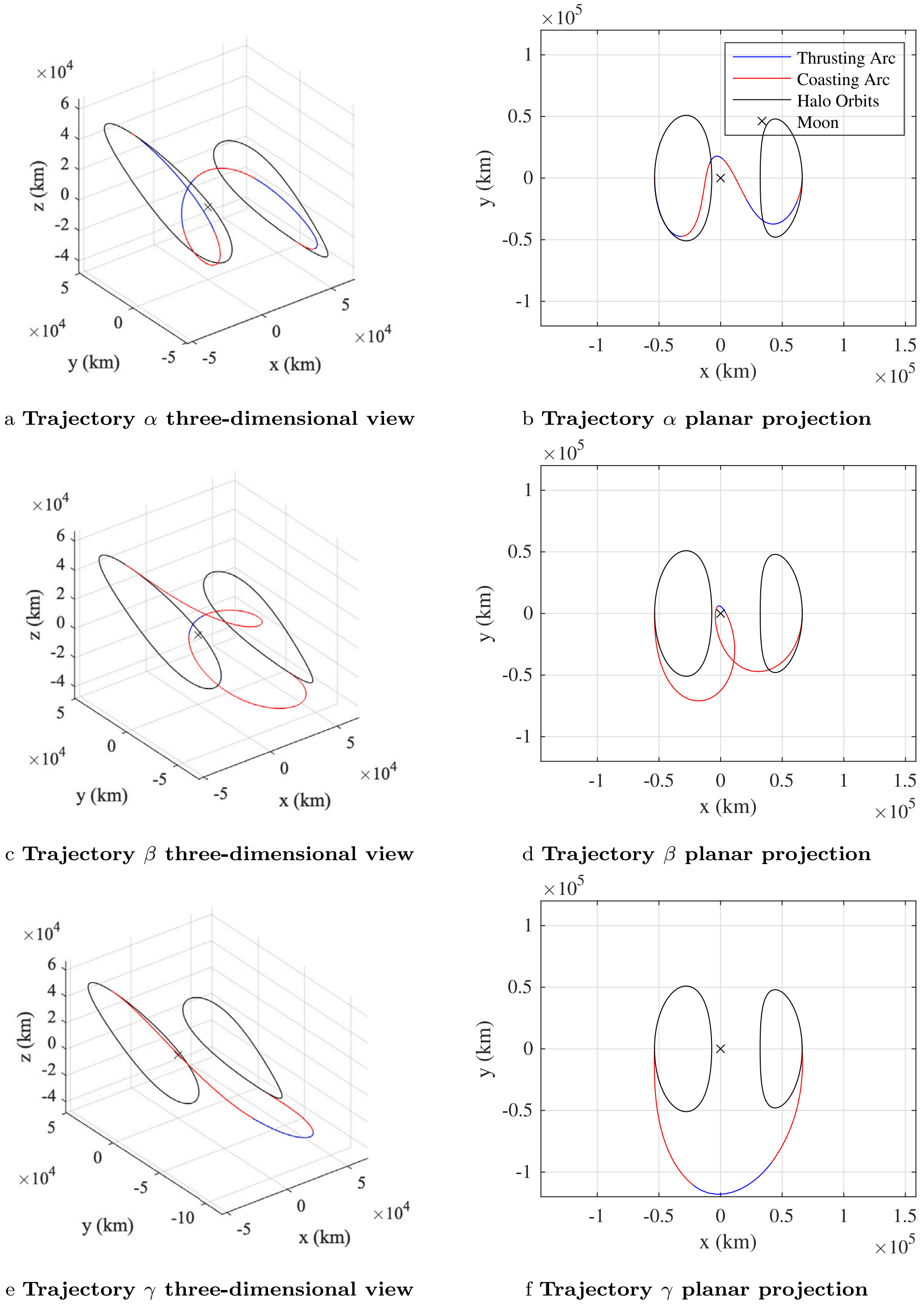
Similarly to Scenario 1, multiple solutions of the TPBVP were discovered by applying the proposed method to solve the L2 halo orbit to L1 halo orbit transfer problem. Figs. 7 and 8 display the resultant trajectories and the corresponding vehicle mass time histories, respectively. In Fig. 7, the trajectories are shown both in a three-dimensional view and projected onto the x–y plane of the synodic reference frame with the origin shifted to lie at the center of the Moon. It is evident that the three discovered solutions differ drastically. Both Trajectory α and β (shown in Figs. 7(a), 7(b), 7(c), and 7(d)) appear to take advantage of the lunar gravitational attraction with a close approach of the Moon while transferring between halo orbits. Whereas a vehicle traveling along Trajectory α efficiently performs the transfer, requiring the least amount of thrusting among the found solutions, traversing Trajectory β requires thrust to be applied for nearly the full duration of the transfer (see Fig. 8). Employing an entirely different strategy, Trajectory γ (shown in Figs. 7(e) and 7(f)) does not use a close approach to the lunar surface, and instead thrusts away from the Moon, then follows a coasting arc placed at the greatest distance from the Moon, before thrusting for the final leg of the transfer to insert into the L1 halo orbit.

Table 5 reports the initial co-state variables corresponding to each of the solutions, along with the fuel required to traverse each trajectory. As expected from the previous discussion, Trajectory α is the most fuel-efficient, requiring just 35.34 kg of fuel, while Trajectory γ requires nearly double the fuel mass (i.e., 61.27 kg), and Trajectory β is the most fuel expensive (at 81.28 kg). It should be noted that the fuel-optimal trajectory discovered through the application of the proposed method (i.e., Trajectory α) agrees with results published by Aziz et al. [3]. A slightly lower fuel requirement is found here, which is explained by the fact that the Hybrid Differential Dynamic Programming (HDDP) algorithm applied by Aziz et al. enforced a constant direction of thrust over each integration step and can lag in switching thrust on or off [3], whereas indirect methods allow thrust to be applied in the optimal direction of the primer vector at every point along the trajectory and enforce throttling on or off at the optimal times.

4.2.2. Co-state initialization

The performance of the proposed method at converging to solutions of the minimum-fuel L2 halo orbit to L1 halo orbit transfer TPBVP is analyzed in the following. For each of the investigated swarm sizes, Table 6 displays the percentage of trials that converge to a solution of the minimum-fuel TPBVP, along with the final PSO objective function value, time to converge, the number of function evaluations averaged over the 100 trials, and the total number of trials which were stopped due to the maximum PSO evolution time limit of 30 min.

Overall, the proposed methodology performs more favorably at generating a guess of the initial co-state values for the L2 to L1 transfer problem, compared to Scenario 1. In this case, the best performance was observed when a swarm size of 500 particles was employed, with 86% of trials converging to a solution of the TPBVP. A swarm size of 250 particles performed nearly as well, with 80% of trials converging to a TPBVP solution while only requiring 2.3 min on average to generate the guess. Swarm sizes increasing up to 1000 particles continued to produce quality guesses frequently, converging in 79% and 81% of trials for 750 or 1000 particles respectively, albeit at a higher computational cost. Interestingly, for a swarm size greater than 1000 particles (aside from 3000 particles), co-state initialization performance appeared to taper off, whereas the average time to converge in all cases was well below the time limit of 30 min; the number of trials which were stopped prematurely was relatively small when compared



a **Trajectory α three-dimensional view**

b **Trajectory α planar projection**

c **Trajectory β three-dimensional view**

d **Trajectory β planar projection**

e **Trajectory γ three-dimensional view**

f **Trajectory γ planar projection**

Fig. 7. Minimum-fuel trajectories for Scenario 2.

Table 6
Convergence characteristics for different swarm sizes in Scenario 2.

Convergence characteristic	Swarm size							
	250	500	750	1000	2000	3000	4000	5000
Min. Fuel % Converged	80	86	79	81	67	79	73	75
Avg. Final Obj. Func. (10^{-3})	14.6	8.89	9.57	7.37	9.33	7.56	7.61	8.30
Avg. Time to Converge (min)	2.30	3.76	4.77	6.20	10.46	11.28	12.69	14.61
Avg. Function Evaluations	7.78E5	1.31E6	1.58E6	2.02E6	3.21E6	3.25E6	3.24E6	3.43E6
Reached Time Limit	0	0	0	0	10	15	14	27

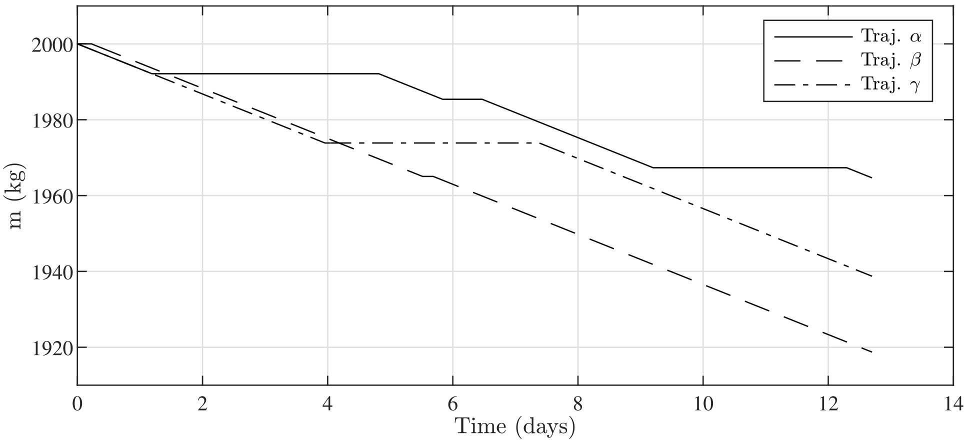


Fig. 8. Vehicle mass time histories for Scenario 2 solutions.

to the results presented for Scenario 1. Furthermore, on average, larger swarm sizes successfully reduced the PSO objective function further compared to the smaller, more often successful swarm sizes. This is counter-intuitive, as a smaller average final objective function value indicates that the generated co-state guesses are closer to satisfying the final time boundary condition constraints and are therefore expected to converge to a solution of the TPBVP with a higher frequency, as was observed in Scenario 1.

It was discovered that multiple local minima of the PSO cost function exist for the problem investigated in Scenario 2. The necessary condition for a minimum of the PSO cost function is given by

$$\left(\frac{\partial J_{PSO}}{\partial \lambda(t_i)}\right)^T = 2\left(\frac{\partial \mathbf{e}(t_f)}{\partial \lambda(t_i)}\right)^T \mathbf{W}\mathbf{e}(t_f) = \mathbf{0}_{7\times 1} \tag{44}$$

which is satisfied either (1) when the final boundary condition residual $\mathbf{e}(t_f)$ is reduced to the zero vector or (2) when $\left(\partial\mathbf{e}(t_f)/\partial\lambda(t_i)\right)^T$ becomes rank deficient and $\mathbf{We}(t_f)$ lies in its null space. Clearly, global minima of the PSO cost function correspond to a final boundary condition residual of zero, and local minima of the PSO cost function exist in regions where the Jacobian of the final boundary condition residual with respect to the initial co-state vector is rank deficient.

It is expected that the counter-intuitive phenomenon of larger PSO swarm sizes resulting in lower convergence rates observed in Scenario 2 is due to PSO converging to regions near local minima of the PSO cost function more frequently than in Scenario 1, possibly due to the chaotic behavior of the cislunar regime in this case. In trials when PSO produced a guess which did not converge to a solution through single shooting, the PSO-generated guess was confirmed to lie in a region where $\partial\mathbf{e}(t_f)/\partial\lambda(t_i)$, and therefore the Jacobian of the shooting function with respect to the initial co-states (see Eq. (40)), is rank deficient. This rank deficiency then results in updates of Newton's method in the trust-region algorithm which produce little to no improvement during single shooting and therefore fail to converge to a solution of the TPBVP.

An example of a trial that encountered this difficulty is shown in Fig. 9. Represented with a dashed line is the trajectory corresponding to the PSO-generated guess of

$$\lambda_{PSO}(t_i) = \begin{bmatrix} -0.75848 & 0.28231 & -9.76567 & -1.27696 & 2.08707 & -3.41462 & 4.57662 \end{bmatrix}^T$$

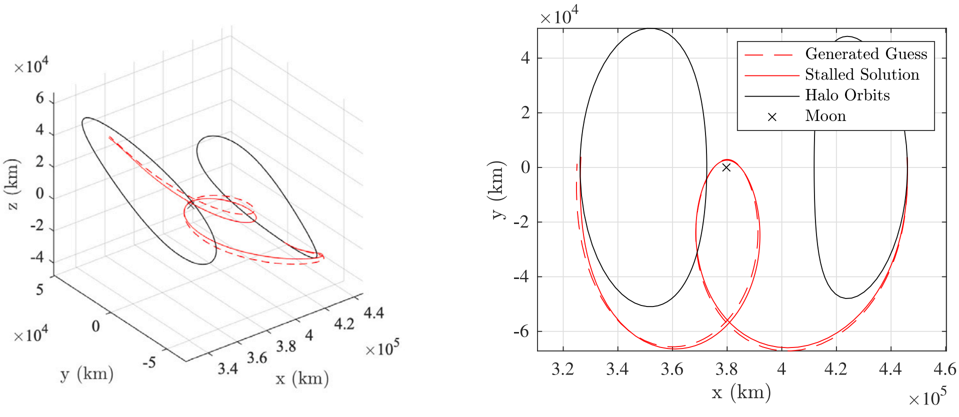
After performing single shooting and failing to converge, the trajectory shown as a solid red line corresponds to the values of the initial co-states at which single shooting stalled, which are given by

$$\lambda(t_i) = \begin{bmatrix} -0.15992 & -0.23058 & -5.42275 & -0.57888 & 0.97406 & -1.98589 & 2.11896 \end{bmatrix}^T$$

It is important to note that both trajectories are entirely shown in red because no coasting arcs were exploited by either; however, it should be recalled that the minimum-energy problem does not require optimal trajectories to maintain a bang–bang thrust profile (see Eq. (24) with $\epsilon = 1.0$). The ∞−norm of the TPBVP final boundary condition residual was $\|\mathbf{e}(t_f)\|_\infty = 3.95 \times 10^{-2}$ and $\|\mathbf{e}(t_f)\|_\infty = 9.72 \times 10^{-3}$ for the generated guess and the co-states returned after stalled single shooting respectively, which does confirm slight improvement during single shooting. Furthermore, it was found that both the generated guess and single shooting-returned co-states result in a Jacobian of the shooting function with respect to the initial co-states of rank six, which corresponds to rank deficiency. This example illustrates a trial where the final boundary condition residuals were nearly satisfied for both cases but rank deficiency of the Jacobian of the shooting function prevented convergence to a solution of the TPBVP.

5. Conclusion

This paper proposed a method for initializing the co-state variables for solving low-thrust minimum-fuel trajectory optimization problems with the indirect shooting approach by employing PSO paired with an energy-to-fuel homotopy technique. The method was applied successfully to solve two low-thrust transfer problems in the Earth–Moon system. It was demonstrated that the methodology is able to successfully generate guesses for the initial co-state variables which converge to a solution for both investigated scenarios. The resulting minimum-fuel trajectories were validated by comparison with published solutions of the optimal low-thrust transfer for both scenarios. Analysis was performed to determine the effect that varying the number of particles in the swarm has on the performance of the proposed method. It was found that for problems involving several low-thrust spirals about a primary body, a larger swarm size of 2000 to 4000 particles resulted in the highest percentage of guesses which converged to a solution of the TPBVP. However, due to the increase of the computational expense with a large number of particles, it was shown that using less particles (e.g., 250) and performing more trials ensures a higher probability of computing at least one solution in a certain time. For problems involving short transfers between orbits in cislunar space, a smaller swarm size of 500 particles is preferred and is capable of rapidly generating solutions in less than 4 min. It should be noted that, due to the heuristic nature of the PSO algorithm, multiple solutions to the derived minimum-fuel TPBVP were found, which correspond to extremal solutions of the minimum-fuel objective function, providing interesting physical insight into the investigated trajectory optimization scenarios. Considerations on the existence of local minima and Jacobian rank deficiency were also discussed.



a **Three-dimensional view**

b **Planar projection**

Fig. 9. Initial guess and stalled solution for minimum-energy trajectory.

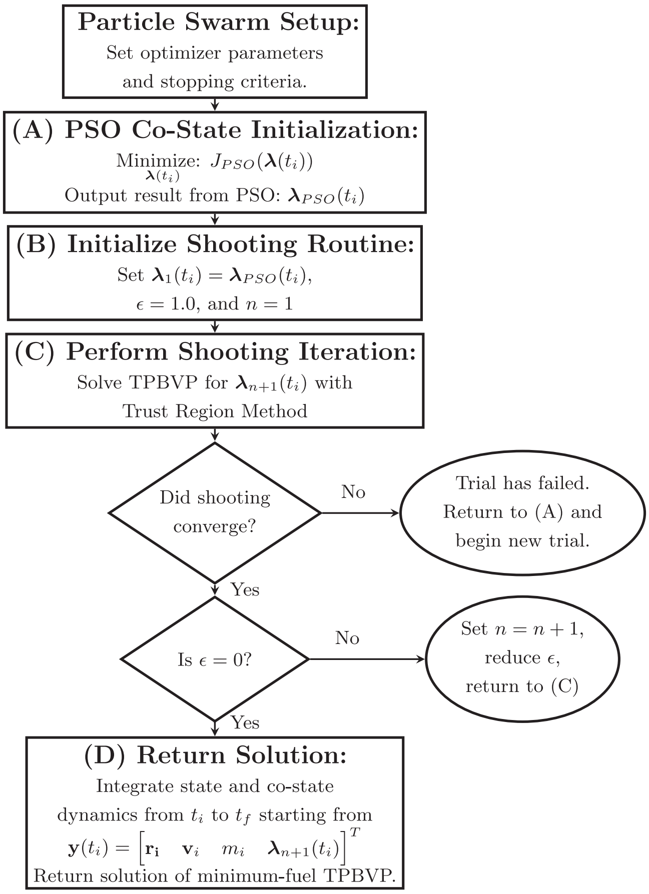

Particle Swarm Setup:
Set optimizer parameters
and stopping criteria.
(A) PSO Co-State Initialization:
Minimize: $J_{PSO}(\boldsymbol{\lambda}(t_i))$ over $\boldsymbol{\lambda}(t_i)$
Output result from PSO: $\boldsymbol{\lambda}_{PSO}(t_i)$
(B) Initialize Shooting Routine:
Set $\boldsymbol{\lambda}_1(t_i) = \boldsymbol{\lambda}_{PSO}(t_i)$,
$\epsilon = 1.0$, and $n = 1$
(C) Perform Shooting Iteration:
Solve TPBVP for $\boldsymbol{\lambda}_{n+1}(t_i)$ with
Trust Region Method
Did shooting
converge?
No
Trial has failed.
Return to (A) and
begin new trial.
Yes
Is $\epsilon = 0$?
No
Set $n = n + 1$,
reduce ϵ,
return to (C)
Yes
(D) Return Solution:
Integrate state and co-state
dynamics from t_i to t_f starting from
$\mathbf{y}(t_i) = \begin{bmatrix}\mathbf{r_i} & \mathbf{v}_i & m_i & \boldsymbol{\lambda}_{n+1}(t_i)\end{bmatrix}^T$
Return solution of minimum-fuel TPBVP.

Fig. A.1.

The primary benefits of applying PSO for minimum-fuel co-state initialization are the simplicity and versatility of the approach. Once the TPBVP is derived through the application of COV and PMP, little additional work is required to employ PSO to initialize the co-state variables for a range of trajectory optimization problems in similar dynamical environments. Additionally, PSO removes the need for a guess to be provided by the end-user of the algorithm, a task that is often nontrivial and requires experience. Furthermore, removing the requirement for a user-provided guess is an advantageous feature for preliminary mission design efforts, when a large search for many candidate trajectories is required. A large search for candidate trajectories could also be accelerated drastically through the exploitation of a high-performance computing environment, as the methodology developed herein lends itself well to parallel computing and could be distributed across many CPUs to generate solutions for many initial and target states, flight times, and spacecraft parameters. Furthering this cause, the proposed methodology was shown to facilitate the discovery of multiple solutions of the derived TPBVP, most of which could be considered candidate trajectories during preliminary mission design efforts once local optimality is verified through an investigation of the second-order sufficient conditions of optimality. Finally, while not investigated in this work, the multiple solutions discovered through the application of the proposed methodology could be improved through further homotopy continuation on the fixed flight time or thrust magnitude to produce even more candidate trajectories.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 88983. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Grant Hecht also acknowledges reception of the NASA Space Grant Fellowship.

Appendix. Co-state initialization and homotopy continuation flowchart

See Fig. A.1.

References

[1] R. Woollands, E. Taheri, J. Junkins, Efficient computation of optimal low thrust gravity perturbed orbit transfers, in: 29th AAS/AIAA Spaceflight Mechanics Meeting, Univelt Inc., 2019, pp. 899–918.

[2] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, Y. Xia, A review of optimization techniques in spacecraft flight trajectory design, Prog. Aerosp. Sci. 109 (2019) 100543, http://dx.doi.org/10.1016/j.paerosci.2019.05.003.

[3] J.D. Aziz, D.J. Scheeres, G. Lantoine, Hybrid differential dynamic programming in the circular restricted three-body problem, J. Guid. Control Dyn. 42 (5) (2019) 963–975, http://dx.doi.org/10.2514/1.G003617.

[4] J.L. Shannon, M.T. Ozimek, J.A. Atchison, C.M. Hartzell, Q-law aided direct trajectory optimization of many-revolution low-thrust transfers, J. Spacecr. Rockets 57 (4) (2020) 672–682.

[5] C. Hofmann, F. Topputo, Rapid low-thrust trajectory optimization in deep space based on convex programming, J. Guid. Control Dyn. 44 (7) (2021) 1379–1388, http://dx.doi.org/10.2514/1.G005839.

[6] E. Taheri, N.I. Li, I. Kolmanovsky, Co-state initialization for the minimum-time low-thrust trajectory optimization, Adv. Space Res. 59 (9) (2017) 2360–2373, http://dx.doi.org/10.1016/j.asr.2017.02.010.

[7] L.C.W. Dixon, M.C. Biggs, The advantages of adjoint–control transformations when determining optimal trajectories by Pontryagin's maximum principle, Aeronaut. J. (1968) 76 (735) (1972) 169–174, http://dx.doi.org/10.1017/S0001924000042731.

[8] L.C.W. Dixon, M.C. Biggs, Adjoint–control transformations for solving practical optimal control problems, Optim. Control Appl. Methods 2 (4) (1981) 365–381, http://dx.doi.org/10.1002/oca.4660020405.

[9] J. Senent, C. Ocampo, A. Capella, Low-thrust variable-specific-impulse transfers and guidance to unstable periodic orbits, J. Guid. Control Dyn. 28 (2) (2005) 280–290, http://dx.doi.org/10.2514/1.6398.

[10] C.L. Ranieri, C.A. Ocampo, Optimization of roundtrip, time-constrained, finite burn trajectories via an indirect method, J. Guid. Control Dyn. 28 (2) (2005) 306–314, http://dx.doi.org/10.2514/1.5540.

[11] C.L. Ranieri, C.A. Ocampo, Indirect optimization of spiral trajectories, J. Guid. Control Dyn. 29 (2006) 1360–1366, http://dx.doi.org/10.2514/1.19539.

[12] R.P. Russell, Primer vector theory applied to global low-thrust trade studies, J. Guid. Control Dyn. 30 (2) (2007) 460–472, http://dx.doi.org/10.2514/1.22984.

[13] J.D. Thorne, C.D. Hall, Approximate initial Lagrange costates for continuous–thrust spacecraft, J. Guid. Control Dyn. 19 (1996) 283–288, http://dx.doi.org/10.2514/3.21616.

[14] D. Lee, H. Bang, Efficient initial costates estimation for optimal spiral orbit transfer trajectories design, J. Guid. Control Dyn. 32 (2009) 1943–1947, http://dx.doi.org/10.2514/1.44550.

[15] D. Lee, H. Bang, H.-D. Kim, Optimal Earth-Moon trajectory design using new initial costate estimation method, J. Guid. Control Dyn. 35 (2012) 1671–1676, http://dx.doi.org/10.2514/1.55863.

[16] F. Fahroo, I.M. Ross, Costate estimation by a Legendre pseudospectral method, J. Guid. Control Dyn. 24 (2) (2001) 270–277, http://dx.doi.org/10.2514/2.4709.

[17] D.A. Benson, G.T. Huntington, T.P. Thorvaldsen, A.V. Rao, Direct trajectory optimization and costate estimation via an orthogonal collocation method, J. Guid. Control Dyn. 29 (2006) 1435–1440, http://dx.doi.org/10.2514/1.20478.

[18] D. Garg, M.A. Patterson, C. Francolin, C.L. Darby, G.T. Huntington, W.W. Hager, A.V. Rao, Direct trajectory optimization and costate estimation of finite-horizon and infinite-horizon optimal control problems using a radau pseudospectral method, Comput. Optim. Appl. 49 (2011) 335–358, http://dx.doi.org/10.1007/s10589-009-9291-0.

[19] C.C. Françolin, D.A. Benson, W.W. Hager, A.V. Rao, Costate approximation in optimal control using integral Gaussian quadrature orthogonal collocation methods, Optim. Control Appl. Methods 36 (2015) 381–397, http://dx.doi.org/10.1002/oca.2112.

[20] J.D. Eide, W.W. Hager, A.V. Rao, Modified Radau collocation method for solving optimal control problems with nonsmooth solutions part II: Costate estimation and the transformed adjoint system, in: 2018 IEEE Conference on Decision and Control, CDC, 2018, pp. 1651–1656, http://dx.doi.org/10.1109/CDC.2018.8619426.

[21] M. Pontani, B.A. Conway, Particle swarm optimization applied to space trajectories, J. Guid. Control Dyn. 33 (2010) 1429–1441, http://dx.doi.org/10.2514/1.48475.

[22] M. Pontani, B. Conway, Optimal low-thrust orbital maneuvers via indirect swarming method, J. Optim. Theory Appl. 162 (2014) 272–292, http://dx.doi.org/10.1007/s10957-013-0471-9.

[23] R. Bertrand, R. Epenoy, New smoothing techniques for solving Bang–Bang optimal control problems – Numerical results and statistical interpretation, Optim. Control Appl. Methods 23 (4) (2002) 171–197, http://dx.doi.org/10.1002/oca.709.

[24] C. Zhang, F. Topputo, F. Bernelli-Zazzera, Y.-S. Zhao, Low-thrust minimum-fuel optimization in the circular restricted three-body problem, J. Guid. Control Dyn. 38 (8) (2015) 1501–1510, http://dx.doi.org/10.2514/1.G001080.

[25] V. Szebehely, Chapter 1 – description of the restricted problem, in: V. Szebehely (Ed.), Theory of Orbit, Academic Press, 1967, pp. 7–41, http://dx.doi.org/10.1016/B978-0-12-395732-0.50007-6.

[26] D.F. Lawden, Optimal trajectories for space navigation, Math. Gazette 48 (366) (1964) 478–479.

[27] A.E. Bryson, Applied Optimal Control : Optimization, Estimation, and Control, Rev. Print. ed., Hemisphere Pub. Corp., Washington, 1975.

[28] J.E. Prussing, S.L. Sandrik, Second-order necessary conditions and sufficient conditions applied to continuous-thrust trajectories, J. Guid. Control Dyn. 28 (4) (2005) 812–816, http://dx.doi.org/10.2514/1.4040.

[29] P.K. Mogensen, K. Carlsson, S. Villemot, S. Lyon, M. Gomez, C. Rackauckas, T. Holy, D. Widmann, T. Kelman, D. Karrasch, A. Levitt, A.N. Riseth, C. Lucibello, C. Kwon, D. Barton, J. TagBot, M. Baran, M. Lubin, S. Choudhury, S. Byrne, S. Christ, T. Arakaki, T.A. Bojesen, Benneti, M.R.G. Macedo, JuliaNLSolvers/NLsolve.jl: v4.5.1, 2020, URL https://github.com/JuliaNLSolvers/NLsolve.jl. (Accessed 04 October 2022).

[30] J. Bezanson, A. Edelman, S. Karpinski, V.B. Shah, Julia: A fresh approach to numerical computing, SIAM Rev. 59 (1) (2017) 65–98, http://dx.doi.org/10.1137/141000671.

[31] C. Rackauckas, Q. Nie, DifferentialEquations.jl – A performant and feature-rich ecosystem for solving differential equations in Julia, J. Open Res. Software 5 (1) (2017) 15, http://dx.doi.org/10.5334/jors.151.

[32] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, Vol. 4, IEEE, 1995, pp. 1942–1948.

[33] E. Mezura-Montes, C.A.C. Coello, Constraint-Handling in nature-inspired numerical optimization: Past, present and future, Swarm Evol. Comput. 1 (4) (2011) 173–194, http://dx.doi.org/10.1016/j.swevo.2011.10.001.

[34] M.E.H. Pedersen, Good Parameters for Particle Swarm Optimization, Tech. Rep. HL1001, Hvass Lab. Copenhagen, Denmark, 2010, pp. 1551–3203.

[35] Particle swarm optimization algorithm, 2022, URL https://www.mathworks.com/help/gads/particle-swarm-optimization-algorithm.html. (Accessed: 11 May 2022).

[36] G. Mingotti, F. Topputo, F. Bernelli-Zazzera, Combined optimal low-thrust and stable-manifold trajectories to the Earth-Moon Halo orbits, AIP Conf. Proc. 886 (1) (2007) 100–112, http://dx.doi.org/10.1063/1.2710047.

[37] R. Armellin, F. Topputo, A sixth-order accurate scheme for solving two-point boundary value problems in astrodynamics, Celestial Mech. Dynam. Astronom. 96 (3–4) (2006) 289–309, http://dx.doi.org/10.1007/s10569-006-9047-4.

[38] C. Martin, B.A. Conway, Optimal Low-Thrust Trajectories using Stable Manifolds, in: Cambridge Aerospace Series, Cambridge University Press, 2010, pp. 238–262, http://dx.doi.org/10.1017/CBO9780511778025.010.

[39] M. Rasotto, R. Armellin, P.D. Lizia, F. Bernelli-Zazzera, Optimal low-thrust transfers in two-body and three-body dynamics, in: 64th International Astronautical Congress, Vol. 7, 2014, pp. 5230–5244, URL https://epubs.surrey.ac.uk/840753/.

[40] J.H. Verner, Numerically optimal Runge–Kutta pairs with interpolants, Numer. Algorithms 53 (2–3) (2010) 383–396, http://dx.doi.org/10.1007/s11075-009-9290-3.

Grant R. Hecht is a Ph.D. candidate in the Department of Mechanical and Aerospace Engineering at the University at Buffalo. He is also an NSF GRFP fellow and a NASA Pathways Student. He holds a B.Sc. in Aerospace Engineering from the Missouri University of Science and Technology. His main research interests are in spacecraft trajectory optimization, formation guidance, and inertial navigation.

Dr. Eleonora M. Botta is an assistant professor in the Department of Mechanical and Aerospace Engineering at the University at Buffalo, NY. She holds a Ph.D. in Mechanical Engineering from McGill University, a B.Sc. in Aerospace Engineering and a M.Sc. in Space Engineering from Politecnico di Milano, and a M.Sc. in Aerospace Engineering from Politecnico di Torino. Her main research interests are the dynamics and control of tethered space systems, spacecraft trajectory optimization, and space situational awareness. She is a member of the AAS Space Flight Mechanics Committee and the vicechair of the AIAA Space Tethers Technical Committee.