Hindawi Publishing Corporation
International Journal of Aerospace Engineering
Volume 2011, Article ID 476197, 13 pages
doi:10.1155/2011/476197

Research Article

Generating Solar Sail Trajectories in the Earth-Moon System Using Augmented Finite-Difference Methods

Geoffrey G. Wawrzyniak and Kathleen C. Howell

School of Aeronautics and Astronautics, Purdue University, 701 W. Stadium Avenue, West Lafayette, IN 47907-2045, USA

Correspondence should be addressed to Kathleen C. Howell, howell@purdue.edu

Received 8 November 2010; Accepted 1 February 2011

Academic Editor: Srinivas R. Vadali

Copyright © 2011 G. G. Wawrzyniak and K. C. Howell. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Using a solar sail, a spacecraft orbit can be offset from a central body such that the orbital plane is displaced from the gravitational center. Such a trajectory might be desirable for a single-spacecraft relay to support communications with an outpost at the lunar south pole. Although trajectory design within the context of the Earth-Moon restricted problem is advantageous for this problem, it is difficult to envision the design space for offset orbits. Numerical techniques to solve boundary value problems can be employed to understand this challenging dynamical regime. Numerical finite-difference schemes are simple to understand and implement. Two augmented finite-difference methods (FDMs) are developed and compared to a Hermite-Simpson collocation scheme. With 101 evenly spaced nodes, solutions from the FDM are locally accurate to within 1740 km. Other methods, such as collocation, offer more accurate solutions, but these gains are mitigated when solutions resulting from simple models are migrated to higher-fidelity models. The primary purpose of using a simple, lower-fidelity, augmented finite-difference method is to quickly and easily generate accurate trajectories.

1. Introduction

When a permanent outpost on the Moon to support extended human expeditions is eventually established, the astronauts at the facility will require a continual communications link with the Earth. A leading candidate location for the lunar base is at the south pole, which is not always in view of either antennas on the Earth or space-based transceivers, such as the TDRSS satellites, in geosynchronous orbit. Therefore, at least two spacecraft in traditional polar orbits about the Moon are required for an Earth-Moon link [1].

In contrast to a constellation of multiple spacecraft, alternative communications strategies that rely on only one satellite do exist, using current or near-future technology. Advanced propulsion concepts, such as low-thrust ion engines, as well as solar sails, supply a force in addition to gravity and can actually offset an orbit from the Moon [2–4], that is, the orbit plane is displaced from the gravitational center and the spacecraft appears to hover above the surface. Such an orbit allows one spacecraft to be in view of a location on the lunar surface at all times. This dynamical system is complex due to the gravitational effects of the two primaries and, in the case of solar sails, the fact that the Sun, which influences the direction and magnitude of the resulting sail thrust vector, continually moves relative to a fixed Earth-Moon system. A previous approach, based on an understanding of the dynamical structure and using that knowledge to design an orbit, has been successful in developing some solar sail orbits in the vicinity of the lunar Lagrange points [5–9]. However, motion below one of the lunar poles requires an alternative strategy.

Trajectories can be represented as solutions to boundary value problems (BVPs) in terms of ordinary differential equations (ODEs). For a BVP, conditions are specified at the beginning and, also, at the end of the time domain. The specific values of the states at the extremes may or may not be fixed in a BVP with a periodicity constraint; periodicity only requires that the values at the extremes are equal. Differential-corrections schemes [10] and shooting methods (both adaptations of the initial value problem), collocation approaches, and finite-difference methods [11–13] are common numerical processes to solve BVPs and

are similar in that they all employ linear corrections to a nonlinear system (these three techniques are discussed in greater detail below). Variational methods are also used to solve BVPs when the problem possesses certain minimality properties [14]. Classical perturbation techniques have long supplied analytical approximations for trajectories [5, 15–17]. Finally, recent efforts using evolutionary algorithms to solving optimization problems also provide an approach to solving boundary value problems [18, 19].

Shooting methods [10, 12, 20–23] and collocation schemes [2, 24–31] possess high degrees of accuracy and are frequently employed for trajectory generation and optimization. Shooting methods typically require some knowledge of the design space to initiate the procedure. Collocation schemes do not require a priori knowledge of the solution corresponding to the BVP, but the implementation is often neither intuitive nor straightforward. Furthermore, the partial derivatives of the constraints with respect to the states are nontrivial in collocation schemes and are often approximated numerically [2].

A more rudimentary method for solving BVPs is the finite-difference method (FDM), in which the derivatives that appear in the differential equations are replaced with their respective finite differences and evaluated at node points along the trajectory [12]. The solution process is iterative. The trajectory is discretized, and the equations that represent the relationships at the nodes are solved simultaneously. Using a central difference approximation, the smallest local errors associated with an FDM approach are proportional to Δt^2. However, model accuracy and uncertainty may obviate any precision gains from a high-precision scheme; solutions from those methods may not persevere without additional control in a higher-fidelity model (e.g., a model that includes planetary ephemerides) or in actual flight. If a solution based on a high-precision approach (e.g., shooting or high-degree collocation) is migrated from a model with limited accuracy to one of higher accuracy (e.g., circular restricted three-body model versus ephemeris model or actual flight), it may be no more accurate in that high-fidelity model than a solution from a low-precision approach (e.g., finite-differencing). Moreover, the FDM is simple to formulate and offers an improved understanding of the design space. Partial derivatives are easily obtained via standard analytical techniques, without having to resort to numerical or automatic differentiation (each is often computationally expensive). Once the analyst is familiar with the feasible options, higher-fidelity methods for solving the BVP can be used to refine trajectories or combined with optimization techniques to produce optimal trajectories. Another option is to employ the FDM technique to generate trajectories that fit path constraints to quickly explore a broad design space. Using the MATLAB computing environment, Wawrzyniak and Howell [32] survey over 10 million combinations of initial guesses for the path and control profile, as well as a range of characteristic accelerations and path constraints appropriate to the lunar south pole coverage problem. The time to generate the millions of trajectories for the survey is approximately one week when the algorithms run on eight cores over five platforms.

In the following sections, an FDM is described and applied to solar sail orbits in the Earth-Moon circular restricted three-body (CR3B) system. Also included is a modified FDM for the same application, where the velocity is incorporated as part of the solution at each node along the discretized trajectory. Neither method is strictly a straightforward, "textbook" FDM [11–13], since both are augmented with trajectory and control constraints. Additionally, in contrast to a simple two-point BVP, where the states are fixed at the extremes, the trajectory is required to be periodic; nowhere along the arc is the solution prespecified.

A description of the dynamical model in the Earth-Moon CR3B system is followed by an algorithm using the augmented FDMs for generating trajectories. The strategies are based on minimizing the difference between accelerations from the equations of motion (as well as the velocities as another alternative) and the corresponding values numerically derived from positions along a discretized trajectory. An error analysis is also included. A separate study uses these FDMs for surveying the solution space and assessing the required solar sail and spacecraft characteristics necessary for the lunar south pole (LSP) coverage problem [32].

2. System Dynamical Model

The problem that represents this application is defined within the context of the CR3B system, that is, the problem is formulated in a frame, R, that is rotating with respect to an inertial system, I. A CR3B model that incorporates the gravity contributions of two primary bodies is geometrically advantageous for understanding the problem. Consistent with McInnes [33], the nondimensional vector equation of motion for a spacecraft at a location $\mathbf{r}$ relative to the barycenter (center of mass of the primaries) is

$$^{R}\mathbf{a} + 2\left({}^{I}\boldsymbol{\omega}^{R} \times {}^{R}\mathbf{v}\right) + \nabla U(\mathbf{r}) = \mathbf{a}_s(t), \tag{1}$$

where the first term is the acceleration relative to the rotating frame (more precisely expressed as $^{R}d^2\mathbf{r}/dt^2$, where the left superscript R indicates a derivative in the rotating frame) and the second term is the corresponding Coriolis acceleration, which requires the velocity relative to the rotating frame, $^{R}\mathbf{v}$ (more precisely $^{R}d\mathbf{r}/dt$). Vectors are denoted with boldface. Derivatives of the position vector, $^{R}\mathbf{v}$ and $^{R}\mathbf{a}$, are assumed to be relative to the rotating frame and, consequently, R is dropped. The angular-velocity vector, $^{I}\boldsymbol{\omega}^{R}$ (or $\boldsymbol{\omega}$), relates the rate of change the rotating frame with respect to the inertial frame. The applied acceleration, from a solar sail in this case, is indicated on the right side by $\mathbf{a}_s(t)$. The pseudogravity gradient, $\nabla U(\mathbf{r})$, combines centripetal and gravitational accelerations:

$$\nabla U(\mathbf{r}) = (\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r})) + \left(\frac{(1-\mu)}{r_1^3}\mathbf{r_1} + \frac{\mu}{r_2^3}\mathbf{r_2} \right), \tag{2}$$

where μ represents the mass fraction of the smaller body, or $m_2/(m_1+m_2)$, and r_1 and r_2 are the distances from the larger and smaller bodies, respectively, that is,

$$r_1 = \sqrt{(\mu+x)^2+y^2+z^2},$$
$$r_2 = \sqrt{(\mu+x-1)^2+y^2+z^2}. \tag{3}$$

Solar gravity is neglected in this model. At a distance of 1 AU, an appropriate distance to assume for a sailcraft in the Earth-Moon system, the applied acceleration from a solar sail is modeled as

$$\mathbf{a}_s(t) = \beta\left(\hat{\boldsymbol{\ell}}(t)\cdot\hat{\mathbf{u}}\right)^2\hat{\mathbf{u}}, \tag{4}$$

where $\hat{\mathbf{u}}$ is the sail-face normal, $\hat{\boldsymbol{\ell}}(t)$ is a unit vector in the sun-to-spacecraft direction, and β is the sail characteristic acceleration in nondimensional units. These vectors appear in Figure 1. Observed from the rotating frame, R, the Sun moves in a clockwise direction about the fixed primaries. The sail mass, m_3, is negligible compared to the masses of the Earth and Moon, which are m_1 and m_2, respectively. The term $(\hat{\boldsymbol{\ell}}(t)\cdot\hat{\mathbf{u}})$ is also expressed as $\cos\alpha$, where α is the sail pitch angle, or the angle between the solar incidence direction and the sail normal.

To generate the magnitude of the sail acceleration in dimensional units, a_0, β is multiplied by the system characteristic acceleration, a^*, which is the relationship between the dimensional and nondimensional acceleration in (1). In fact, a^* is the ratio of the characteristic length, L^* (384,400 km for the Earth-Moon distance), to the square of the characteristic time, t^* ($2\pi t^* = 27.321$ days), that is,

$$a^* = \frac{L^*}{t^{*2}} = 2.7307\,\text{mm/s}^2. \tag{5}$$

A recent sailcraft design for NASA's Space Technology competition (ST9) that was built by L'Garde possesses overall characteristic acceleration, a_0, of 0.58 mm/s^2, while the characteristic acceleration of the sail and its support structure alone is closer to 1.70 mm/s^2 (0.212 to 0.623 in units of nondimensional acceleration, resp.) [34].

The sunlight direction is expressed relative to the rotating frame and is a function of time, that is,

$$\hat{\boldsymbol{\ell}}(t) = \cos(\Omega t)\hat{\mathbf{x}} - \sin(\Omega t)\hat{\mathbf{y}} + 0\hat{\mathbf{z}}, \tag{6}$$

where Ω is the ratio of the synodic rate of the Sun as it moves along its path to the system rate, approximately 0.9192. One physical constraint is imposed on the attitude of the spacecraft: the sail normal, $\hat{\mathbf{u}}$, which is coincident with the direction of the resultant force in an ideal model, is always directed away from the Sun. This is written mathematically as

$$\hat{\boldsymbol{\ell}}(t)\cdot\hat{\mathbf{u}} \geq \cos\alpha_{\max}, \tag{7}$$

where $\alpha_{\max}$ is 90°.

The sail modeled here is a perfectly reflecting, flat solar sail. Billowing is not incorporated in this force model; however, $\alpha_{\max}$ could be less than 90°, as sail luffing (i.e., flapping) is assumed to occur at high pitch angles [35]. Higher fidelity models include optical models [33], parametric models that incorporate billowing in addition to optical effects [33, 36], and realistic models based on finite-element analysis that incorporates optical properties and manufacturing flaws [37]. Optical effects can represent a nonperfectly reflecting solar sail; some energy is absorbed, and some is reflected diffusely as well as specularly. An ideal sail reflects only specularly. In all of these models, the resulting acceleration from a solar sail is not perfectly parallel to the sail-face normal but, instead, is increasingly offset from the sail-face normal as the sail is pitched further from the sunlight direction [33]. Fully accounting for realistic solar sail properties attenuates the sail characteristic acceleration by nearly 25% and places an upper limit on the pitch angle between 50° and 60°, depending on the properties of the sail [37]. Nevertheless, this analysis will employ an ideal sail to lend insight into the technology level that is required to solve the LSP coverage problem and into the use of augmented finite-difference methods for generating solar sail trajectories.

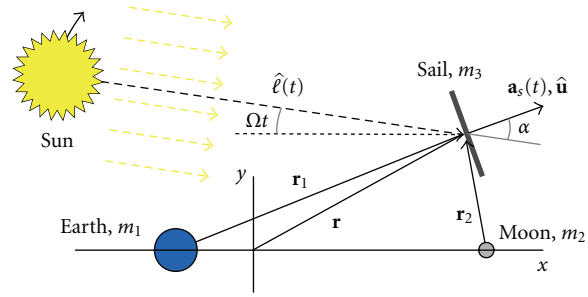


Figure 1: Earth-Moon system model.

3. Augmented Finite-Difference Methods

The derivation for an augmented finite-difference method begins with generic, nonlinear, second-order, two-point BVP that can be represented as

$$\ddot{\mathbf{r}} = \mathbf{f}(t,\mathbf{r},\dot{\mathbf{r}}), \tag{8}$$

where $\mathbf{r}(t_1)$ and $\mathbf{r}(t_n)$ are fixed at the extremes of the time span, $[t_1, t_n]$ in interval notation [11–14]. The solution of (8) is a trajectory, $\mathbf{r}(t)$. To solve the BVP, the classic FDM discretizes the domain into nodes, or epochs, $t_1, t_2, \ldots, t_n$ and replaces the derivatives in (8) by their respective finite differences. Central-difference approximations (CDAs) are commonly used because they are more accurate than forward or backward differences. The first and second time derivatives of $\mathbf{r}(t)$ are, of course, the velocity and acceleration vectors and are approximated by CDAs as

$$\tilde{\mathbf{v}}_i = \frac{\mathbf{r}_{i+1}-\mathbf{r}_{i-1}}{t_{i+1}-t_{i-1}}, \tag{9}$$

$$\tilde{\mathbf{a}}_i = 2\frac{(\mathbf{r}_{i+1}-\mathbf{r}_i)(t_i-t_{i-1})-(\mathbf{r}_i-\mathbf{r}_{i-1})(t_{i+1}-t_i)}{(t_i-t_{i-1})(t_{i+1}-t_{i-1})(t_{i+1}-t_i)}, \tag{10}$$

where $\mathbf{r}(t_i)$ is defined as $\mathbf{r}_i$ and $t_i \in (t_{i-1}, t_{i+1})$ and the symbol "$\tilde{\cdot}$" indicates a numerical approximation. Equation (10) arises from three central-differences: one for the velocity at $t_{i-1/2}$, which is midway between t_{i-1} and t_i, another at $t_{i+1/2}$, midway between t_i and t_{i+1}, and a third using the first two intermediate velocities resulting in the acceleration at t_i. The FDM does not require uniform node placement (mesh refinement and nonuniform node spacing may improve the accuracy of the FDM), but when the time steps between t_{i-1}, t_i, and t_{i+1} are fixed at a value of Δt, then the midpoint of $[t_{i-1}, t_{i+1}]$ is t_i and (9) and (10) simplify to

$$\tilde{\mathbf{v}}_i = \frac{\mathbf{r}_{i+1} - \mathbf{r}_{i-1}}{2\Delta t}, \tag{11}$$

$$\tilde{\mathbf{a}}_i = \frac{\mathbf{r}_{i+1} - 2\mathbf{r}_i + \mathbf{r}_{i-1}}{\Delta t^2}. \tag{12}$$

Note that $\mathbf{r}_{n-1}$ is used for $\mathbf{r}_0$ when calculating $\tilde{\mathbf{v}}_1$ for a periodic solution. Similarly, $\mathbf{r}_1$ substitutes for $\mathbf{r}_n$ in $\tilde{\mathbf{v}}_{n-1}$, so the problem does not include redundant constraints. The calculations for $\mathbf{a}_1$ and $\mathbf{a}_{n-1}$ are similar. The differences between the actual velocity and acceleration and their respective CDAs at t_i are

$$\mathbf{v}_i - \tilde{\mathbf{v}}_i = -\frac{\Delta t^2}{6}\mathbf{r}'''(\rho_v), \tag{13}$$

$$\mathbf{a}_i - \tilde{\mathbf{a}}_i = -\frac{\Delta t^2}{12}\mathbf{r}^{\mathrm{iv}}(\rho_a), \tag{14}$$

where $\rho_{v,a} \in (t_{i-1}, t_{i+1})$ and $\mathbf{r}'''(t)$ and $\mathbf{r}^{\mathrm{iv}}(t)$ are continuous on $[t_{i-1}, t_{i+1}]$. Derivations of these truncation errors are available in the literature [11–13]. The approximations in (11) and (12) replace the first and second time derivatives in (8), and the result is an equation at each epoch of the form

$$\frac{\mathbf{r}_{i-1} + 2\mathbf{r}_i + \mathbf{r}_{i+1}}{\Delta t^2} = \mathbf{f}\left(t, \mathbf{r}, \frac{\mathbf{r}_{i+1} - \mathbf{r}_{i-1}}{2\Delta t}\right). \tag{15}$$

Equation (15) may still be nonlinear, but it can be linearized into a system of equations, one for each t_i, excluding the extremes, and solved iteratively. Due to the approximations in (11) and (12), the classic FDM results in a solution that will approximate each component of the true solution to the BVP by an error proportional to Δt^2 at each node [11, 13].

For the current problem, the FDM is augmented to incorporate path and system constraints, as well as a control history. Two variations of this FDM are developed. In the first, the position is the only explicitly discretized trajectory state (FDM-R). The second FDM is formulated to directly solve for position *and* velocity along the discretized trajectory (FDM-RV).

The FDM-R formulation is based on approximating (1) via the CDAs for velocity and acceleration given by (11) and (12), respectively, that is,

$$\tilde{\mathbf{a}}_i + 2(\boldsymbol{\omega} \times \tilde{\mathbf{v}}_i) + \nabla U(\mathbf{r}_i) - \mathbf{a}_s(t_i) = 0. \tag{16}$$

Note that the velocity term in (16) is replaced with its CDA, $\tilde{\mathbf{v}}_i$. To solve the BVP, the trajectory is discretized at $t_1, t_2 \ldots, t_n$. Position is expressed in terms of Cartesian coordinates at each node:

$$\mathbf{r}_i = \left\{x_i \quad y_i \quad z_i\right\}^{\mathrm{T}}, \tag{17}$$

where "T" indicates a vector transpose. The control, $\mathbf{u}_i$, that is, the sail pointing vector in this application, as well as a set of m slack variables, η_i, are also included at each node. Note that the control, $\mathbf{u}_i$, is not necessarily a unit vector until the FDM converges on a solution. The positions, control, and slack variables at node i for the FDM-R are collected into the subvector,

$$\mathbf{q}_i = \left\{\mathbf{r}_i^{\mathrm{T}} \quad \mathbf{u}_i^{\mathrm{T}} \quad \boldsymbol{\eta}_i^{\mathrm{T}}\right\}^{\mathrm{T}}_{(6+m)}, \tag{18}$$

where the length of $\mathbf{q}_i$ is $(6+m)$. The subvectors corresponding to each node are collected into a column vector,

$$\mathbf{X} = \left\{\mathbf{q}_1^{\mathrm{T}} \quad \mathbf{q}_2^{\mathrm{T}} \quad \cdots \quad \mathbf{q}_{n-1}^{\mathrm{T}} \quad \mathbf{q}_n^{\mathrm{T}}\right\}^{\mathrm{T}}_{(6+m)n}, \tag{19}$$

which represents the discretized trajectory as well as the associated control history and slack variables. The velocities associated with the trajectories generated by the FDM-R can be reconstructed with CDAs of $\mathbf{r}(t)$ during postprocessing.

The FDM-RV process is similarly formulated, but there are two distinct sets of ODEs to solve, that is,

$$\tilde{\mathbf{v}}_i - \mathbf{v}_i = 0, \tag{20}$$

$$\tilde{\mathbf{a}}_i + 2(\boldsymbol{\omega} \times \mathbf{v}_i) + \nabla U(\mathbf{r}_i) - \mathbf{a}_s(t_i) = 0. \tag{21}$$

Note the difference in the velocity terms in (16) and (21). With this FDM-RV alternative, the subvector is

$$\mathbf{q}_i = \left\{\mathbf{r}_i^{\mathrm{T}} \quad \mathbf{v}_i^{\mathrm{T}} \quad \mathbf{u}_i^{\mathrm{T}} \quad \boldsymbol{\eta}_i^{\mathrm{T}}\right\}^{\mathrm{T}}_{(9+m)}, \tag{22}$$

since $\mathbf{v}_i$ is now explicitly part of the solution. This formulation results in an $\mathbf{X}$ vector with length $(9+m)n$.

Both the FDM-R and the FDM-RV algorithms constrain the difference between the evaluated EOMs at node i and the associated numerically derived approximations to be zero. Additional constraints are required for periodicity, control-direction magnitude, and path characteristics. All of these constraints are dependent on $\mathbf{X}$ and are collected in the column vector $\mathbf{F}(\mathbf{X})$. An expansion of $\mathbf{F}(\mathbf{X})$ about $\mathbf{X}^j$ yields a linear approximation for $\mathbf{F}(\mathbf{X}^{j+1})$, that is,

$$\mathbf{F}\left(\mathbf{X}^{j+1}\right) = \mathbf{F}\left(\mathbf{X}^j\right) + D\mathbf{F}\left(\mathbf{X}^j\right)\left(\mathbf{X}^{j+1} - \mathbf{X}^j\right) + \mathcal{O}\left(\Delta\mathbf{X}^2\right), \tag{23}$$

where the $D\mathbf{F}(\mathbf{X}^j)$ is the Jacobian, $\partial\mathbf{F}(\mathbf{X}^j)/\partial\mathbf{X}^j$. Superscripts on vectors refer to iteration number. The initial guess is denoted $j = 0$; a converged solution by $j = f$. Partial derivatives used in the Jacobian are derived analytically, but are not specifically presented here. By assuming that $\mathbf{X}^j$ is in the neighborhood of $\mathbf{F}(\mathbf{X}) = 0$ (i.e., $\|\mathbf{X}^{j+1} - \mathbf{X}^j\| \ll 1$), (23) is rearranged into a least-norm problem, that is,

$$\mathbf{X}^{j+1} = \mathbf{X}^j - D\mathbf{F}\left(\mathbf{X}^j\right)^{\mathrm{T}}\left[D\mathbf{F}\left(\mathbf{X}^j\right) \cdot D\mathbf{F}\left(\mathbf{X}^j\right)^{\mathrm{T}}\right]^{-1}\mathbf{F}\left(\mathbf{X}^j\right). \tag{24}$$

Equations (23) and (24) reflect the Newton-Raphson method of root finding for several variables. For efficient computation, $D\mathbf{F}(\mathbf{X})$ is preallocated and stored in memory as a sparse matrix [38]. Iterations based on (24) continue until some convergence tolerance is met, that is,

$$\frac{\|\mathbf{X}^{j+1} - \mathbf{X}^{j}\|}{\|\mathbf{X}^{j}\|} \leq \text{tol}. \tag{25}$$

When an initial guess is in the neighborhood of a solution, convergence is quadratic. Specifying tol $\approx 1 \times 10^{-7}$ is sufficient, since any further iteration approaches double precision. If the initial guess is not in the neighborhood of a solution, $\mathbf{X}^{j+1}$ bears little resemblance to $\mathbf{X}^{j}$; the step from $\mathbf{X}^{j}$ to $\mathbf{X}^{j+1}$ may even be chaotic. However, the new $\mathbf{X}^{j+1}$ might be in the neighborhood of an alternate solution and subsequently lead to convergence. The converged discretized solution, $\mathbf{X}^{f}$, that satisfies these constraints is the trajectory that solves the EOMs to within a theoretical error.

Algebraically, these FDMs are equivalent. In practice, the two formulations yield slightly different results and, sometimes, dramatically different results. This problem is sensitive to the initial guess; because the initial condition may or may not explicitly include velocity, the vector of initial guesses is not the same for the two FDMs. Additionally, the least-norm update in (24) minimizes the norm of the correction to the state vector; the state vectors for the two formulations are different, and the least-norm update for an initial guess that includes velocity states may move the solution to a different region when compared to an update that does not contain velocity states between iterations. The convergence criterion (25) also depends on the norm of the update to the state vector. Because the size of the state vectors differs between the methods, the vector norm of the update will also differ for converged solutions. The current formulations do not control specified states for convergence (e.g., position and not velocity), and other variations on the implementation of these methods will yield different results. Given that the objective is a simple and quick method to approximate the design space, neither strategy should be considered superior to the other; both achieve the stated goal and results from both are insightful.

At each epoch, the converged subvector $\mathbf{q}_i$ differs from the true solution, $\overline{\mathbf{q}}_i$, due to the truncation error associated with the FDMs and the limited machine precision. The proceeding error analysis is described by Kincaid and Cheney [13, pages 589–592]. For a step size Δt that is greater than some value, the error due to truncation in (9)–(14) will dominate the machine error. Assuming that there exists a true set of position states, $\overline{\mathbf{r}}$, that solve (1) and (13) as formulated for the FDM-R approach, (8) is written as

$$\begin{aligned}\frac{\overline{\mathbf{r}}_{i-1} + 2\overline{\mathbf{r}}_i + \overline{\mathbf{r}}_{i+1}}{\Delta t^2} &- \frac{\Delta t^2}{12}\overline{\mathbf{r}}^{\text{iv}}(\rho_a) \\ &= -\frac{\boldsymbol{\omega} \times (\overline{\mathbf{r}}_{i+1} - \overline{\mathbf{r}}_{i-1})}{\Delta t} + \frac{\Delta t^2}{6}\boldsymbol{\omega} \times \overline{\mathbf{r}}'''(\rho_v) \\ &\quad - \nabla U(\overline{\mathbf{r}}_i) + \mathbf{a}_s(t_i).\end{aligned} \tag{26}$$

The pseudogravity gradient associated with the true solution, $\overline{\mathbf{r}}_i$, is expanded as a function of an approximate solution, $\mathbf{r}_i$, that is,

$$\nabla U(\overline{\mathbf{r}}_i) = \nabla U(\mathbf{r}_i) + \mathbf{M}(\overline{\mathbf{r}}_i)\mathbf{e}_i + \mathcal{O}\left(\mathbf{e}_i^{\mathrm{T}}\mathbf{e}_i\right), \tag{27}$$

where $\mathbf{e}_i = \overline{\mathbf{r}}_i - \mathbf{r}_i$ and $\mathbf{M}(\overline{\mathbf{r}}_i)$ is the Hessian of U. Using the solution from the FDM-R algorithm, (15) is subtracted from (26), resulting in

$$\begin{aligned}&\Delta t^{-2}(\mathbf{e}_{i-1} + 2\mathbf{e}_i + \mathbf{e}_{i+1}) \\ &\quad = -\Delta t^{-1}(\boldsymbol{\omega} \times (\mathbf{e}_{i+1} - \mathbf{e}_{i-1})) - \mathbf{M}_i\mathbf{e}_i + \Delta t^2\mathbf{h}_i,\end{aligned} \tag{28}$$

where $\mathbf{h}_i = (1/12)\overline{\mathbf{r}}^{\text{iv}}(\rho_a) + (1/6)\overline{\mathbf{r}}'''(\rho_v)$. It is assumed that $\overline{\mathbf{r}}(t)$ is continuously differentiable to fourth order. Rearranging and multiplying by Δt^2 yields

$$\begin{aligned}&(-\mathbf{I} + \Delta t\boldsymbol{\omega}\times)\mathbf{e}_{i-1} + (2\mathbf{I} - \Delta t^2\mathbf{M}_i)\mathbf{e}_i + (-\mathbf{I} - \Delta t\boldsymbol{\omega}\times)\mathbf{e}_{i+1} \\ &\quad = -\Delta t^4\mathbf{h}_i,\end{aligned} \tag{29}$$

where $\mathbf{I}$ is the identity matrix and $\boldsymbol{\omega}\times$ is the skew-symmetric gyroscopic matrix. Some terms in (29) can be simplified to a form that will be useful later, that is

$$(-\mathbf{I} + \Delta t\boldsymbol{\omega}\times) + (2\mathbf{I} - \Delta t^2\mathbf{M}_i) + (-\mathbf{I} - \Delta t\boldsymbol{\omega}\times) = -\Delta t^2\mathbf{M}_i. \tag{30}$$

Now, let $\boldsymbol{\lambda}$ correspond to the difference element $\mathbf{e}_i$ that possesses the largest magnitude. Equation (29) reduces to an upper bound on $\boldsymbol{\lambda}$,

$$(-\mathbf{I} + \Delta t\boldsymbol{\omega}\times)\boldsymbol{\lambda} + (2\mathbf{I} - \Delta t^2\mathbf{M}_i)\boldsymbol{\lambda} + (-\mathbf{I} - \Delta t\boldsymbol{\omega}\times)\boldsymbol{\lambda} \geq -\Delta t^4\mathbf{h}_i. \tag{31}$$

Using (30), (31) simplifies to

$$-\Delta t^2\mathbf{M}_i\boldsymbol{\lambda} \geq -\Delta t^4\mathbf{h}_i, \tag{32}$$

$$\mathbf{M}_i\boldsymbol{\lambda} \leq \Delta t^2\mathbf{h}_i, \tag{33}$$

$$\boldsymbol{\lambda} \leq \Delta t^2\mathbf{M}_i^{-1}\mathbf{h}_i. \tag{34}$$

The inequality in (34) represents three scalar equations. Thus, the position error corresponding to the FDM-R formulation is $\mathcal{O}(\Delta t^2)$ as $\Delta t \to 0$. To calculate the velocity from the FDM-R algorithm, the CDA from (11) is used, also with an error $\mathcal{O}(\Delta t^2)$. The error analysis for the alternate FDM-RV algorithm incorporates the fact that the velocity, $\mathbf{v} = \mathbf{f}(t, \mathbf{r})$, possesses errors that can also be expressed by (13), and, therefore, the derivation continues with the same steps as those for the FDM-R formulation beginning with (26).

The augmented finite-difference methods sacrifice precision for simplicity. At each node, the error in the trajectory is proportional to Δt^2, significantly less precise than common collocation methods [26, 39]. In this analysis, $\Delta t = 0.067$, or $n = 101$, which translates to an error of 0.452% of the Earth-Moon distance, approximately 1740 km, in each direction. To improve the precision of the FDM, the step size between nodes, Δt, can be decreased. The Jacobian $D\mathbf{F}(\mathbf{X})$ becomes quite large, imposing a significant computational

cost. Furthermore, the smaller the step size, the larger the machine error [12]. Alternatively, Richardson extrapolation, or extrapolation to the limit, could be employed to further minimize the error of the solutions [12]. Incorporating a process to determine the specific step size that minimizes some combination of truncation and machine errors or incorporating an additional process, like Richardson extrapolation, adds complexity to a method that is formulated to be simple to understand and implement, as well as quick to yield results.

Prior to any analysis, the design space for this problem is not well known, so a trajectory that is precise to within Δt^2 relative to an actual solution is meaningful. If the goal is a general understanding of the design space, accomplished in a relatively quick analysis, then these results can be very insightful.

4. Algebraic Constraint Vector: F(X)

At the core of the augmented finite-difference methods is the algebraic constraint vector, $\mathbf{F}(\mathbf{X})$, which contains the algebraic approximations to the equations of motion and other constraints necessary for the desired periodic solar sail trajectory. In this formulation, each element in $\mathbf{F}(\mathbf{X})$ should be zero to simultaneously satisfy the ODEs and path constraints. For a periodic solar sail trajectory, subject to path constraints $\mathbf{g}(\mathbf{X})$,

$$\mathbf{F}(\mathbf{X}) = \left\{\Delta\mathbf{a}(\mathbf{X})^{\mathrm{T}} \quad \left[\Delta\mathbf{v}(\mathbf{X})^{\mathrm{T}}\right] \quad \mathbf{T}(\mathbf{X})^{\mathrm{T}} \quad y_1 \quad \mathbf{N}(\mathbf{X})^{\mathrm{T}} \quad \mathbf{g}(\mathbf{X})^{\mathrm{T}}\right\}^{\mathrm{T}}. \tag{35}$$

The presence of $\Delta\mathbf{v}(\mathbf{X})$ depends on whether the formulation is the FDM-R or FDM-RV, and the length of the $\mathbf{F}(\mathbf{X})$ vector is either $(2 + (4 + m)n)$ or $(2 + (7 + m)n)$, accordingly. The order of these elements within the vector is arbitrary and no difference in performance is apparent when $\mathbf{F}(\mathbf{X})$ and $\mathbf{X}$ are rearranged such that $D\mathbf{F}(\mathbf{X})$ is a sparse, banded-diagonal matrix. Using the configuration in (35), each element set appears as block diagonal in the corresponding Jacobian, $D\mathbf{F}(\mathbf{X})$. Each set is subsequently discussed, and the corresponding size of each set is indicated in a subscript outside of the braces.

The first element set in (35) is the difference in acceleration. The acceleration at a given epoch, i, as evaluated from the equations of motion, depends only on the position, velocity, and control, $\mathbf{a}_i = \mathbf{f}(\mathbf{r}_i, \mathbf{v}_i, \mathbf{u}_i)$. The numerically derived acceleration, $\tilde{\mathbf{a}}_i$ from (10) or (12), depends on the state at multiple epochs. A valid trajectory possesses the same acceleration whether computed from the EOMs or from the CDAs, within the truncation error. Therefore, the difference between accelerations resulting from the EOMs and those from numerical calculation should nominally be zero for a converged solution; thus, the difference forms the first set of equality constraints in the composite constraint vector:

$$\Delta\mathbf{a}(\mathbf{X}) = \begin{Bmatrix} \mathbf{a}_1(\mathbf{r}_1, \mathbf{v}_1, \mathbf{u}_1) - \tilde{\mathbf{a}}_1 \\ \vdots \\ \mathbf{a}_{n-1}(\,(\mathbf{r}, \mathbf{v}, \mathbf{u})|_{n-1}) - \tilde{\mathbf{a}}_{n-1} \end{Bmatrix}_{3(n-1)}. \tag{36}$$

Likewise, the difference between the velocity from the EOMs and the numerically derived velocity is zero for a valid trajectory. Velocity from the EOMs at each epoch, $\mathbf{v}_i$, is available from $\mathbf{X}$ (see (19) and (22)). This difference forms a second set of equality constraints (only in the FDM-RV algorithm) in $\mathbf{F}(\mathbf{X})$:

$$\Delta\mathbf{v}(\mathbf{X}) = \begin{Bmatrix} \mathbf{v}_1 - \tilde{\mathbf{v}}_1 \\ \vdots \\ \mathbf{v}_{n-1} - \tilde{\mathbf{v}}_{n-1} \end{Bmatrix}_{3(n-1)}, \tag{37}$$

where $\mathbf{v}_i$ may be approximated as $\tilde{\mathbf{v}}_i$. These differences in acceleration and velocity may also be denoted as *defects* and are located at the node corresponding to each t_i. In an initial guess for the trajectory state vector, the defects $\Delta\mathbf{a}_i$ and $\Delta\mathbf{v}_i$ at each node will most likely not be zero. The Newton-Raphson iteration process adjusts the path of the trajectory to resolve these differences.

The next constraints enforce periodicity and unit length of the control vector. To enforce periodicity, the goal is an originating and final state vector that are equal, which is represented as a set of constraints in the algebraic constraint vector, $\mathbf{F}(\mathbf{X})$, that is,

$$\mathbf{T}(\mathbf{X}) = \begin{Bmatrix} \mathbf{r}_n - \mathbf{r}_1 \\ [\mathbf{v}_n - \mathbf{v}_1] \\ \mathbf{u}_n - \mathbf{u}_1 \\ \boldsymbol{\eta}_n - \boldsymbol{\eta}_1 \end{Bmatrix}_{(6|9+m)}, \tag{38}$$

where m is the number of path constraints at the first and last node, expressed as slack variables. The length of $\mathbf{T}(\mathbf{X})$ depends on the application of an FDM-R or an FDM-RV approach. Although the only boundary condition is periodicity, for consistency the position at the boundary is constrained to be in the xz-plane. Therefore, the y-coordinate at the first point along the trajectory is constrained, $y_1 = 0$. In practice, without this constraint, y_1 remained very close to zero, approximately 40 km from the xz plane once the solution is converged. With this constraint, y_1 is reduced to 0.004 mm from the plane. Next, the magnitude of the control vector must remain of unit length in the iteration process

$$\mathbf{N}(\mathbf{X}) = \begin{Bmatrix} \mathbf{u}_1^{\mathrm{T}}\mathbf{u}_1 - 1 \\ \vdots \\ \mathbf{u}_{n-1}^{\mathrm{T}}\mathbf{u}_{n-1} - 1 \end{Bmatrix}_{(n-1)}. \tag{39}$$

Formulating the constraint in terms of $\mathbf{u}_i^{\mathrm{T}}\mathbf{u}_i - 1$ as opposed to $\sqrt{\mathbf{u}_i^{\mathrm{T}}\mathbf{u}_i} - 1$ results in a simpler partial derivative while retaining the intended effect; this constraint is incorporated into the implementation of the FDM since the control history in this analysis is described in terms of a unit vector.

Path constraints are included as inequality constraints, which are converted to equality constraints by use of slack variables, a successful numerical adaptation from nonlinear programming [2, 40]. The slack variables are incorporated into $\mathbf{X}$ and are associated with the other state elements at their particular epoch, i. In this analysis, the elevation angle

constraint, $E_{\min}$, maintains the visibility of the spacecraft from an outpost located near the south pole of the Moon, and the spacecraft altitude constraint, $A_{\max}$, is imposed for radio power restrictions. Altitude is defined as the distance from the lunar south pole, that is,

$$A_i = \sqrt{(x_i - 1 + \mu)^2 + y_i^2 + (z_i + R_m)^2}. \tag{40}$$

The third path constraint requires that the sail-face normal, or control $\mathbf{u}_i$, is always directed away from the Sun (the sunlight vector is $\hat{\boldsymbol{\ell}}_i$), or $\alpha_{\max} = 90°$ in (7). Of the inequality constraints, only this attitude requirement is mandated. Together, these three inequality constraints are written as

$$E_{\min} \le E_i \equiv \arcsin\left(-\frac{z_i + R_m}{A_i}\right), \tag{41a}$$

$$A_{\max} \ge A_i, \tag{41b}$$

$$\cos\alpha_{\max} \le \hat{\boldsymbol{\ell}}_i \cdot \mathbf{u}_i. \tag{41c}$$

For the given problem and model, adding a path constraint to avoid the penumbra and umbra of the Earth or the Moon's shadows is unnecessary because of the elevation constraint. A shadow constraint could be added for another application or shadowing effects could be incorporated into the dynamical model directly [41]. Additional inequality path constraints could include limits on the body turn rates and accelerations governed by the attitude control system of the spacecraft. The associated slack variables are squared and added to the inequality constraints, resulting in the following equality constraints:

$$\mathbf{g}_i\left(\mathbf{r}_i, \boldsymbol{\eta}_i\right) = \begin{Bmatrix} \sin E_{\min} + \dfrac{z_i + R_m}{A_i} + \eta_{E,i}^2 \\ A_i - A_{\max} + \eta_{A,i}^2 \\ \cos\alpha_{\max} - \left(\hat{\boldsymbol{\ell}}_i \cdot \mathbf{u}_i\right) + \eta_{\ell,i}^2 \end{Bmatrix}_m. \tag{42}$$

Combining the path constraints at all epochs results in $m(n-1)$ total elements in $\mathbf{g}(\mathbf{X})$. When using the path constraints in (42), $m = 3$. The first two path constraints in (42) as well as the periodicity constraint in (38) are illustrated in Figure 2. The $\cos\alpha_{\max}$ constraint appears in Figure 1. The sailcraft in Figure 2 orbits below the Moon. Feasible solutions could exist that do not orbit below the Moon, but do orbit below either the L_1 or the L_2 point.

Only the first $(n-1)$ epochs are required in formulating $\Delta\mathbf{a}$, $\Delta\mathbf{v}$, $\mathbf{N}$, and $\mathbf{g}(\mathbf{X})$ in the constraint vector $\mathbf{F}(\mathbf{X})$. Due to periodicity, the first and nth epochs are identical. Activating the nth epoch yields a problem that is overconstrained and the Jacobian, $D\mathbf{F}(\mathbf{X})$, is not full rank.

As mentioned, the Jacobian is a sparse matrix. A diagram of the sparsity pattern based on partial derivatives of $\mathbf{F}(\mathbf{X})$ with respect to $\mathbf{X}$ for the FDM-RV formulation appears in Figure 3. Diagonals corresponding to the specific element sets in $\mathbf{F}(\mathbf{X})$ are labeled in the figure. For a scenario employing 101 nodes, the size of the Jacobian is 1013 rows by 1212 columns where approximately 0.38% of the elements are nonzero. The sparsity pattern for the FDM-R formulation is similar to that of the FDM-RV, except that the diagonal

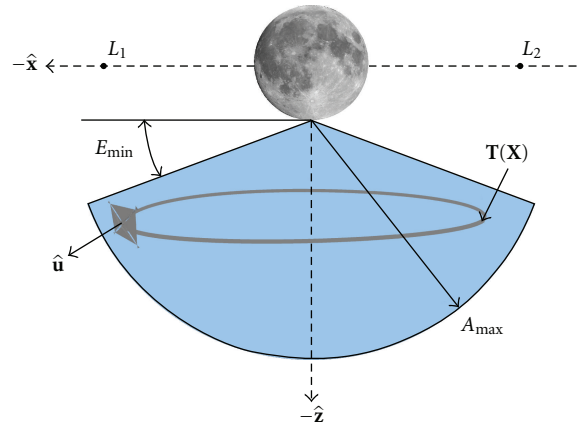


Figure 2: Path constraints for an orbit below the Moon (Moon image from nasa.gov).

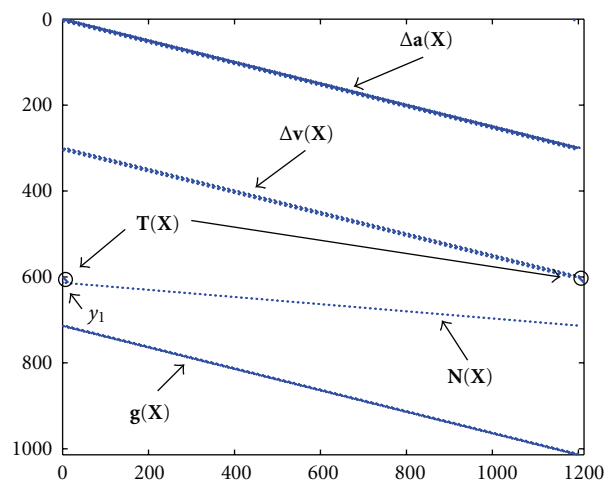


Figure 3: Sparsity pattern for the Jacobian $D\mathbf{F}(\mathbf{X})$ in the FDM-RV formulation.

corresponding to $\mathbf{v}(\mathbf{X})$ does not exist in the FDM-R option. The size of the Jacobian for the FDM-R option associated with 101 nodes is 710 rows by 909 columns; approximately 0.62% of the entries are nonzero.

The advantage of a finite-difference approach is its ease of implementation and the speed improvements enabled by its simplicity. Partial derivatives are easily accessible via analytical derivation, especially for an idealized force model such as (1). As complexity of the formulation increases, which is the case with collocation, for example, analysts rely on numerical or automatic differentiation to generate partial derivatives. MATLAB supplies the function NUMJAC.M, which numerically approximates a derivative using a forward difference approximation [42]. Third-party software for MATLAB exists for automatic (a.k.a. algorithmic) differentiation. TOMLAB Optimization's MAD (MATLAB Automatic Differentiation) suite employs automatic differentiation [43], and Shampine's PMAD (Poor Man's Automatic

Table 1: Computation times for various differentiation strategies.

Strategy	Time
Analytic	0.003269 seconds
PMAD	0.013798 seconds
NUMJAC	0.014924 seconds
TOMLAB/MAD	0.051849 seconds

Differentiation) exploits complex step differentiation [44]. Both MAD and PMAD result in highly accurate derivatives, on the order of machine precision when compared to the partial derivatives determined analytically, while derivatives calculated by NUMJAC.M are accurate to 10^{-8}. Because analytical derivatives are directly encoded into the algorithm, computation times are significantly faster when compared to the other options listed. Table 1 lists the computations times for generating $\partial\nabla U/\partial\mathbf{r}$, the most computationally intensive step in constructing an entry in the Jacobian, for the four differentiation strategies. Clearly, if analytical derivatives are easily available, they should be employed. Furthermore, if the subroutine to generate the derivatives analytically is programmed in a lower-level language (e.g., C or FORTRAN) and then complied as a MATLAB Executable (MEX), further time savings are realized (with compiled code the computation of $\partial\nabla U/\partial\mathbf{r}$ takes less than 0.001 seconds). The formulation of the FDMs for this analysis exploits the computational advantage of analytical derivatives and compiled MEX code.

In summary, a finite-difference method yields a solution for the path by replacing the path derivatives with their finite-difference approximations. The differences between the approximate derivatives and the derivatives determined by evaluating the equations of motion at a specific point along a path are minimized by iteratively solving a linearized system of equations. Inequality path constraints are added to this set of equality constraints by way of slack variables, and subsequently augmenting the constraint vector and linearized system of equations, resulting in a feasible path.

5. Results

The objective when employing a finite-difference method is to quickly generate a feasible trajectory where system constraints and sail characteristics are given. Because model simplification into the CR3B system ignores lunar librations, the inclined orbits of the primaries, additional perturbations to the orbit, and lunar surface features, a conservative minimum elevation angle of $E_{\min} = 15°$ is selected. A generous maximum-altitude constraint of one Earth-Moon distance (384,400 km) is also selected (in practice most results are an order of magnitude closer to the lunar south pole and this constraint is not active) to establish the constraints in (42).

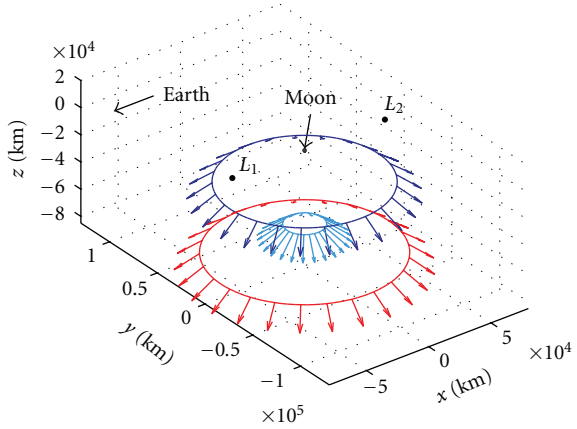
As mentioned, a recent sail design possesses a characteristic acceleration of 0.58 mm/s^2 [34]. JAXA's IKAROS is the first mission to have flown using a sail as its only propulsive device, and NASA recently deployed the NanoSail-D2. Both of these sailcraft are relatively small (200 m^2 and 10 m^2, resp.) and are designed to demonstrate in-space deployment of the sail. Designs for larger sailcraft exist, and future solar sail spacecraft are likely to be hybridized with other propulsion devices [45]. However, it is instructive to demonstrate the FDMs with characteristic accelerations for sail spacecraft that may be available with near-future technological improvements. An assessment of the technology level (i.e., characteristic acceleration) is required for a sail-only LSP mission. The focus of this analysis is not to demonstrate that a sail-only LSP mission is feasible based on current technology. Instead, the results of this investigation demonstrate the use of finite-difference methods for the sample application of an LSP mission. Therefore, a high characteristic acceleration, that is, 1.70 mm/s^2, is used to generate example orbits.

To demonstrate the FDM-R and FDM-RV approaches, three sets of initial guesses for the path and control history are selected to initialize the process. The initial guesses corresponding to the slack variables are always established such that $\mathbf{g}^0(\mathbf{X}) = \mathbf{0}$ (the superscript "0" indicates an initial guess). For each orbit, the initial guess of the control history is one that maximizes the out-of-plane force contributed by the sail along the initial trajectory. Derived analytically by McInnes [33, pages 115–118, 223-224], the sail-pitch angle that maximizes the out-of-plane thrust below the Moon is $\alpha^* = -35.26°$. The initial guess for the thrust vector is then

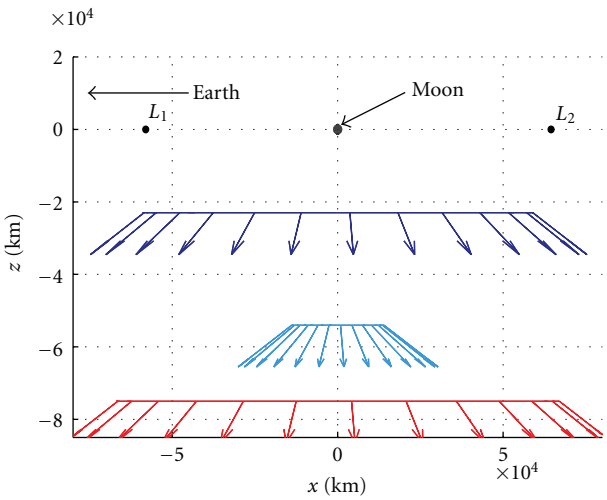
$$\hat{\mathbf{u}}_i^0 = \cos(\Omega t_i)\cos\alpha^*\hat{\mathbf{x}} + \sin(\Omega t_i)\cos\alpha^*\hat{\mathbf{y}} - \sin\alpha^*\hat{\mathbf{z}}. \quad (43)$$

The initial guess for each trajectory is a circle offset below the Moon. Three sample initial guesses appear in Figure 4. The dark-blue circle with a radius of 59,000 km is offset below the Moon by 23,000 km. The radius of the light-blue circle is 14,000 km and the z-offset is 54,000 km. The dimensions of the red circle are 67,000 km by 75,000 km. As previously described, the system is formulated in the Earth-Moon circular restricted three-body system, where the reference frame is fixed to the Earth and the Moon. Therefore, the Sun is initially along the negative x-axis and *moves* in this reference frame clockwise around the Earth and the Moon, per (6). In this frame, the spacecraft is initially on the opposite side of the Moon from the Sun and moves clockwise around the Moon. The arrows on the orbits represent the initial guess for the direction of the sail-face normal, $\hat{\mathbf{u}}$, at the specified point along the orbit. Note that their direction is generally away from the Sun. The arrows are spaced in time by a little more than one day.

When the estimates from Figure 4 initialize the FDM-R algorithm, the three orbits in Figure 5 result. While the dark-blue orbit is not far from its initial approximation, the red orbit and the light-blue orbit are obviously dissimilar from their respective initial guesses. This indicates that a good initial guess for the finite-difference method is not essential for converging on a trajectory that solves the equations of motion. When the same initial guesses are used as inputs for the FDM-RV process, the trajectories in Figure 6 result. For both the FDM-R and FDM-RV algorithms, the dark-blue orbits converged in less than 10 iterations. The other two orbits required between 15 and 20 iterations for both

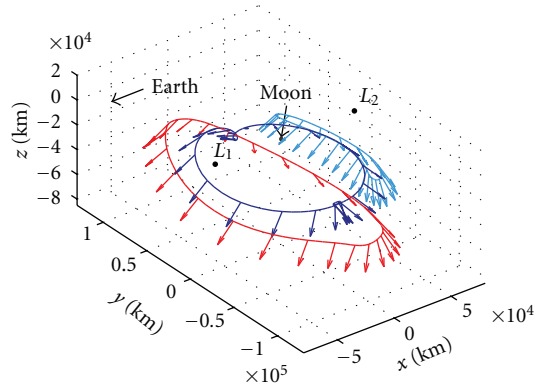


(a) 3D view

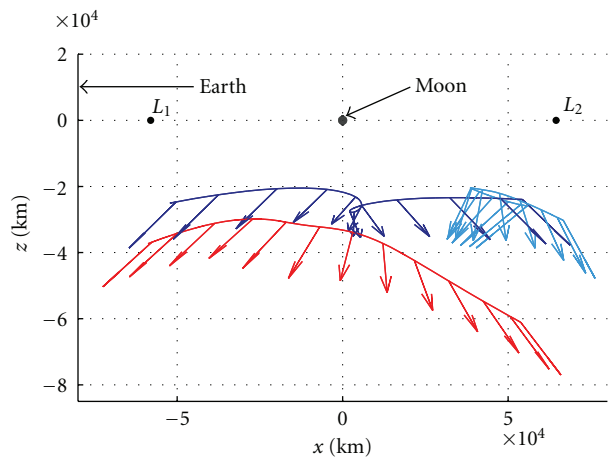


(b) Side view

Figure 4: Three example initial guesses.



(a) 3D view



(b) Side view

Figure 5: Three converged orbits from the FDM-R process.

methods. When individual iterations are examined for these two orbits, it is observed that the general shape of the orbit emerges quickly from the initial guess and the large number of iterations results from small attitude and path changes at the extremities in the $\pm y$ directions to accommodate the elevation constraint. All solutions converge quadratically over the final four or five iterations.

For all trials of the FDM-R and FDM-RV formulations (and in the collocation formulation described in the next section) it is observed that the corrections to the components of the initial guess of the control history are not smooth at the first and nth point; this phenomenon is apparent in the control history from the seventh-degree Gauss-Lobatto collocation formulation presented in Figure 7 of Ozimek et al. [2]. All corrections to the discretized path and attitude profiles are done simultaneously, and the first point and nth points are the only two points in this simulation that are constrained to be equal to each other to ensure periodicity. Because the remaining points in the attitude profile do appear to be continuous and smooth, a postfit modification is applied such that the first and nth points are shifted by interpolating with neighboring points in the periodic profile. This postfit modification is only applied when the Newton-Raphson solver is not within the radius of convergence (i.e., when the convergence criterion in (25) exceeds 0.1). Control profiles from subsequent iterations appear smooth thereafter.

Although the dark-blue and red orbits appear to be similar in both the FDM-R and FDM-RV results, they differ by up to 8000 km and 1800 km, respectively, at certain epochs along the respective paths. The difference in the light-blue orbits in Figures 5 and 6 strongly suggests that the different methods, FDM-R and FDM-RV, can produce different solutions to the equations of motion using the same initial guess. While the theoretical errors in the solutions from the respective algorithms are equivalent, the differences in the formulation of the FDM (either -R or -RV) drive the evolution of the trajectory from iteration to iteration. It is not appropriate to suggest that the solution from one method is more "correct" than the other; both methods theoretically solve the equations of motion to within their errors bounds, and both methods produce orbits that are not obvious when the design space is not well understood.

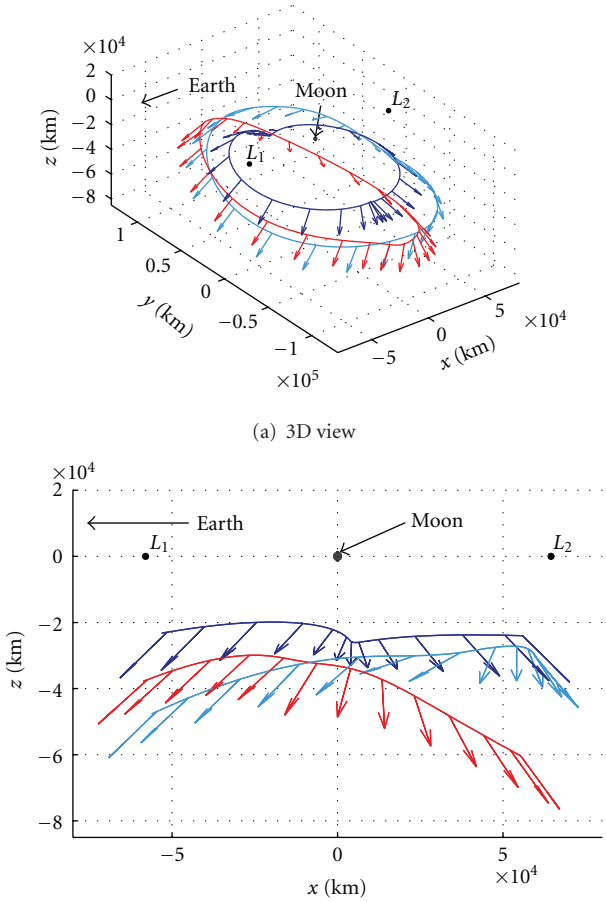


Figure 6: Three converged orbits from the FDM-RV process.

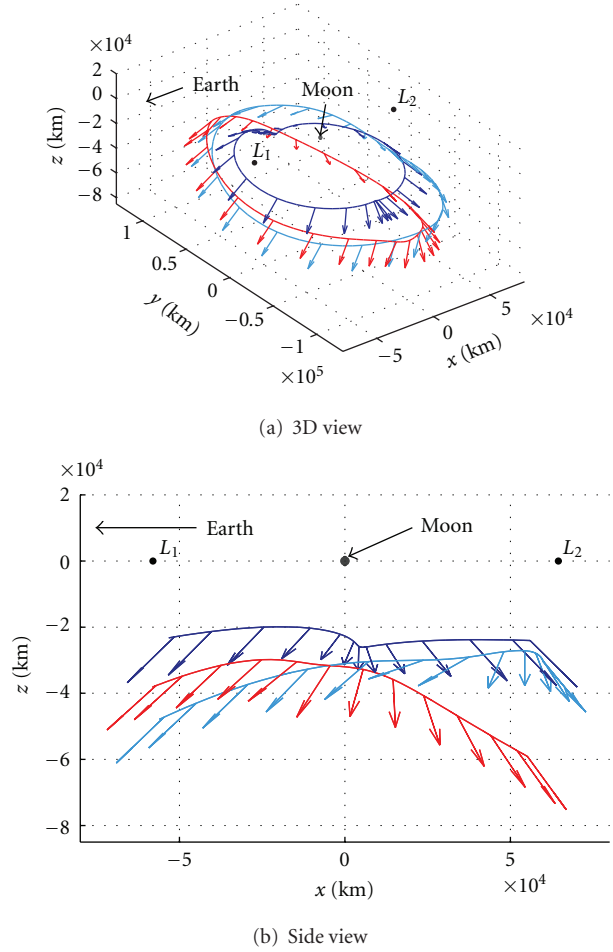


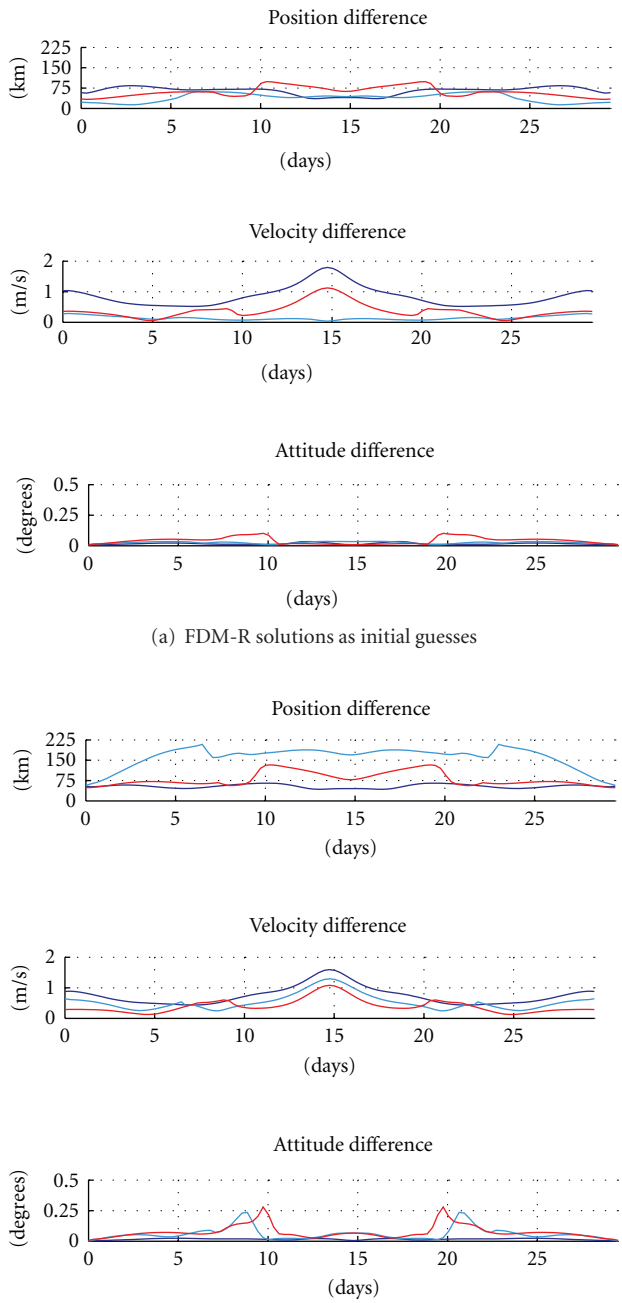
Figure 7: Three converged orbits from the Hermite-Simpson collocation process.

6. Comparison to a Collocation Method

As stated, prior to any analysis, the design space for this problem is unknown. Entry into the design space via analytic techniques is difficult because of coupling of the sail attitude and thrust vector. However, based on the FDM algorithms, solutions precise to $\mathcal{O}(\Delta t^2)$, or 1740 km for this step size, in each direction emerge. In work by Ozimek et al. [2], a Hermite-Simpson collocation method for the LSP coverage problem that has a theoretical local error on the order of $\mathcal{O}(\Delta t^5)$, or 0.5 km for the same step size as that from the FDM algorithms. The Hermite-Simpson collocation method is adapted here as a basis to evaluate the accuracy of the FDM algorithms.

Using the same initial guesses as those represented in Figure 4 to seed the collocation method, the three orbits in Figure 7 result. The three orbits from the Hermite-Simpson approach are clearly similar to those produced by the FDM-RV process and only differ by, at most, 3000 km for the red orbit, 450 km for the light-blue orbit, and 70 km for the dark-blue orbit. The dark-blue and red paths from the Hermite-Simpson process are also similar to the dark-blue and red paths from the FDM-R approach (differing by, at most, 8000 km and 3500 km, resp.), but the light-blue paths are significantly different between the two strategies. Since the theoretical local error in the Hermite-Simpson method is on the order of $\mathcal{O}(\Delta t^5)$, or 0.5 km for this step size, these differences between the similar orbits are not surprising [30]. Given that the initial guesses for the three sample orbits do not resemble their associated converged trajectories, it is remarkable that the converged trajectories from all three numerical processes are similar.

There is one noteworthy difference between the Hermite-Simpson collocation method employed by Ozimek et al. and the adaptation used for this analysis. Ozimek et al. employ a small step along the imaginary axis to calculate the partial derivatives related to the defects. For the current analysis, a simple central difference approximation (CDA) represents the same partial derivatives. Both are numerical schemes for computing difficult partial derivatives and subject to roundoff errors, but the CDA is also subject to truncation error, while the imaginary-step method is not. Thus, the error for the CDA employed in the Hermite-Simpson collocation method is minimized at a step size greater than zero (on the order of 10^{-8}), but the error for the imaginary-step method approaches the roundoff error as the step size approaches zero (on the order of machine precision) [46]. This difference in precision should not have large



(a) FDM-R solutions as initial guesses

(b) FDM-RV solutions as initial guesses

Figure 8: Differences between the solutions from the Hermite-Simpson method and their initial guesses supplied by the FDM-R (a) and FDM-RV (b) solutions.

ramifications for the process to converge, but must be noted. Furthermore, because the partial derivatives are numerically formulated for the Hermite-Simpson collocation scheme in this analysis, the three orbits in Figure 7 took approximately 2 seconds to generate. The orbits in Figures 5 and 6 took approximately 1 second to generate.

The results from any procedure can be used to seed another, usually more precise, process. The solutions produced by the FDM-R and FDM-RV algorithms are used to initialize the Hermite-Simpson scheme. All orbits reconverge quadratically in three or four steps when the Hermite-Simpson strategy is incorporated. The differences in position and velocity, as well as the direction of the sail-face normal, along the 29.5 day trajectories appear in Figure 8. Each color in the figure corresponds to the respective "initial" orbits appearing in Figures 5 and 6. The differences are well within the theoretical errors for the finite-difference methods, and, as such, the augmented finite-difference methods produce accurate and appropriate reference orbits for preliminary mission analysis.

7. Conclusions

Augmented finite-difference methods are useful tools for generating trajectories in poorly understood dynamical regimes, such as the mechanics of flying a solar sail in a multibody system. These schemes are simple to understand and implement, notably in the presence of path constraints. While the theoretical errors of finite-difference methods may be larger than other similar methods that examine the trajectory as a whole, such as collocation, finite-difference methods can provide a reasonable entrance to the design space of a complicated nonlinear problem. They also quickly return results in a regime where little intuition concerning the trajectory and sail-angle history exist. Because of its speed and simplicity, this method may also serve as the basis for generating a large survey of orbit options. Once a viable trajectory is uncovered by the finite-difference method, other techniques may be employed to refine it further. The FDM approach developed for this analysis need not be applied to just solar sail problems, but should be applicable to a large variety of mission design problems.

Acknowledgments

The authors would like to thank Purdue Professor William Crossley for his explanation of optimization techniques in the literature used to solve similar problems and Professor Ahmed Sameh, also from Purdue University, for his insights on numerical methods used to solve boundary-value problems. Former Purdue graduate students Daniel Grebow and Zubin Olikara were invaluable to the initial development and understanding of the numerical methods examined in this paper. The authors also are grateful for the comments from the Editor and reviewers.

References

[1] A. Ely and E. Lieb, "Constellations of elliptical inclined lunar orbits providing polar and global coverage," in *Advances in the Astronautical Sciences*, vol. 123, pp. 1447–1462, Lake Tahoe, Calif, USA, 2005, paper no. AAS 05-343.

[2] M. T. Ozimek, D. J. Grebow, and K. C. Howell, "Design of solar sail trajectories with applications to lunar south pole coverage," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 6, pp. 1884–1897, 2009.

[3] G. G. Wawrzyniak and K. C. Howell, "The solar sail lunar relay system: an application of solar sails in the Earth–Moon

system," in *Proceedings of the 59th International Astronautical Congress*, Glasgow, Scotland, September 2008, paper no. IAC-08.C1.3.14.

[4] D. J. Grebow, M. T. Ozimek, and K. C. Howell, "Design of optimal low-thrust lunar pole-sitter missions," in *Advances in the Astronautical Sciences*, vol. 134, pp. 741–760, Savannah, Ga, USA, 2009, paper no. AAS 09-148.

[5] R. W. Farquhar, "The control and use of libration point satellites," Tech. Rep. TR-R-346, National Aeronautics and Space Administration, February 1970.

[6] C. R. McInnes, "Solar sail trajectories at the lunar L_2 Lagrange point," *Journal of Spacecraft and Rockets*, vol. 30, no. 6, pp. 782–784, 1993.

[7] C. R. McInnes, A. J. McDonald, J. F. L. Simmons, and E. W. MacDonald, "Solar sail parking in restricted three-body systems," *Journal of Guidance, Control, and Dynamics*, vol. 17, no. 2, pp. 399–406, 1994.

[8] J. Simo and C. R. McInnes, "Solar sail trajectories at the Earth–Moon Lagrange points," in *Proceedings of the 59th International Astronautical Congress*, Glasgow, Scotland, 2008, paper no. IAC-08.C1.3.13.

[9] J. Simo and C. R. McInnes, "Solar sail orbits at the Earth-Moon libration points," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 12, pp. 4191–4196, 2009.

[10] K. C. Howell and H. J. Pernicka, "Numerical determination of Lissajous trajectories in the restricted three-body problem," *Celestial Mechanics*, vol. 41, no. 1–4, pp. 107–124, 1987.

[11] H. B. Keller, *Numerical Methods for Two-Point Boundary-Value Problems*, Blaisdell, Waltham, Mass, USA, 1968.

[12] S. D. Conte and C. de Boor, *Elementary Numerical Analysis*, McGraw-Hill, New York, NY, USA, 1980.

[13] D. R. Kincaid and E. W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing*, American Mathematical Society, Providence, RI, USA, 2002.

[14] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer, New York, NY, USA, 3rd edition, 2002.

[15] E. W. Brown, *An Introductory Treatise on Lunar Theory*, Cambridge University Press, London, UK, 1896.

[16] D. L. Richardson, "Analytic construction of periodic orbits about the collinear points," *Celestial Mechanics*, vol. 22, no. 3, pp. 241–253, 1980.

[17] S. H. Strogatz, *Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry, and Engineering*, Westview Press, Cambridge, Mass, USA, 2001.

[18] D. Izzo, "Global optimization and space pruning for spacecraft trajectory design," in *Spacecraft Trajectory Optimization*, B. A. Conway, Ed., Cambridge University Press, New York, NY, USA, 2010.

[19] M. Pontani and B. A. Conway, "Swarming theory applied to space trajectory optimization," in *Spacecraft Trajectory Optimization*, B. A. Conway, Ed., Cambridge University Press, New York, NY, USA, 2010.

[20] R. S. Wilson and K. C. Howell, "Trajectory design in the Sun-Earth-Moon system using lunar gravity assists," *Journal of Spacecraft and Rockets*, vol. 35, no. 2, pp. 191–198, 1998.

[21] B. G. Marchand, K. C. Howell, and R. S. Wilson, "Improved corrections process for constrained trajectory design in the n-body problem," *Journal of Spacecraft and Rockets*, vol. 44, no. 4, pp. 884–897, 2007.

[22] O. von Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Annals of Operations Research*, vol. 37, no. 1, pp. 357–373, 1992.

[23] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 22, no. 2, pp. 193–207, 1998.

[24] R. D. Russell and L. F. Shampine, "A collocation method for boundary value problems," *Numerische Mathematik*, vol. 19, no. 1, pp. 1–28, 1972.

[25] E. J. Doedel, "Finite difference collocation methods for nonlinear two point boundary value problems," *SIAM Journal on Numerical Analysis*, vol. 16, pp. 173–185, 1979.

[26] C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 4, pp. 338–342, 1987.

[27] P. J. Enright and B. A. Conway, "Optimal finite-thrust spacecraft trajectories using collocation and nonlinear programming," *Journal of Guidance, Control, and Dynamics*, vol. 14, no. 5, pp. 981–985, 1991.

[28] P. J. Enright and B. A. Conway, "Discrete approximations to optimal trajectories using direct transcription and nonlinear programming," *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 4, pp. 994–1002, 1992.

[29] S. Tang and B. A. Conway, "Optimization of low-thrust interplanetary trajectories using collocation and nonlinear programming," *Journal of Guidance, Control, and Dynamics*, vol. 18, pp. 599–604, 1995.

[30] A. L. Herman and B. A. Conway, "Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules," *Journal of Guidance, Control, and Dynamics*, vol. 19, no. 3, pp. 592–599, 1996.

[31] R. G. Melton, "Comparison of direct optimization methods applied to solar sail problems," in *Proceedings of the AIAA/AAS Astrodynamics Specialists Conference and Exhibit*, Monterey, Calif, USA, August 2002, paper no. AIAA 2002-4728.

[32] G. G. Wawrzyniak and K. C. Howell, "Investigating the design space for solar sail trajectories in the Earth—Moon system using augmented finite-difference methods," Purdue University Document, School of Aeronautics and Astronautics, February 2011.

[33] C. R. McInnes, *Solar Sailing: Technology, Dynamics and Mission Applications. Space Science and Technology*, Springer, New York, NY, USA, 1999.

[34] D. Lichodziejewski, B. Derbes, D. Sleight, and T. Mann, "Vacuum deployment and testing of a 20m solar sail system," in *Proceedings of the 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Newport, RI, USA, May 2006, paper no. AIAA-2006-1705.

[35] B. Derbes, D. Lichodziejewski, and G. Veal, "A "yank and yaw" control system for solar sails," in *Advances in the Astronautical Sciences*, vol. 19, pp. 2893–2907, 2004.

[36] L. Rios-Reyes and D. J. Scheeres, "Solar-sail navigation: estimation of force, moments, and optical parameters," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 3, pp. 660–668, 2007.

[37] B. Campbell, *An analysis of thrust of a realistic solar sail with focus on a flight validation missionin a geocentric orbit*, Ph.D. Thesis, The George Washington University, Washington, DC, USA, 2010.

[38] The MathWorksTM, Inc, "MATLAB R, Version 7.11.0 (R2010b)," August 2010, http://www.mathworks.com/.

[39] A. L. Herman, *Improved collocation methods with application to direct trajectory optimization*, Ph.D. Thesis, University of Illinois at Urbana-Champaign, 1995.

[40] J. T. Betts, *Practical Methods for Optimal Control Using Nonlinear Programming*, SIAM, Philadelphia, Pa, USA, 2001.

[41] C. R. Ortiz Longo and S. L. Rickman, "Method for the calculation of spacecraft umbra and penumbrashadow terminator points," Tech. Rep. 3547, Lyndon B. Johnson Space Center, Houston,Tex, USA, 1995.

[42] D. E. Salane, "Adaptive routines for forming Jacobians numerically," Tech. Rep. SAND86-1319, Sandia National Laboratories, Albuquerque, NM, USA, 1986.

[43] S. A. Forth and M. Edvall, *User Guide for MAD—A Matlab Automatic Differentiation Package, TOMLAB/MAD, Version 1.4 The Forward Mode*, Engineering Systems Department, Defence College of Management & Technology, Cranfield University, Shrivenham, UK, 2007.

[44] L. F. Shampine, "Accurate numerical derivatives in MATLAB," *ACM Transactions on Mathematical Software*, vol. 33, no. 4, Article ID 1268781, 2007.

[45] M. Macdonald and C. McInnes, "Solar sail mission applications and future advancement," in *Proceedings of the 2nd International Symposium on Solar Sailing*, pp. 1–26, New York City College of Technology, City University of New York, New York, NY, USA, July 2010.

[46] J. Martins, I. Kroo, and J. Alonso, "An automated method for sensitivity analysis using complex variables," in *Proceedings of the 38th Aerospace Sciences Meeting and Exhibit*, Reno, Nev, USA, January 2000, paper no. AIAA 2000-0689.