A Collocation Approach for Computing Solar Sail Lunar Pole-Sitter Orbits

Martin T. Ozimek, Daniel J. Grebow and Kathleen C. Howell*

School of Aeronautics and Astronautics, Purdue University, Armstrong Hall of Engineering, 701 W. Stadium Ave, West Lafayette, Indiana 47907-2045, USA

Abstract: Implementation of a 12^{th}-order Gauss-Lobatto collocation scheme is detailed, including mesh refinement iterations to meet a user-specified error tolerance. The algorithm is robust and efficient, locating path constrained orbits when little information is available regarding the behavior of the solutions. Using a Fourier series control law, the method is applied to the computation of highly unstable, pole-sitter orbits in the Earth-moon restricted three-body problem. The results are comparable to those obtained with standard explicit propagators.

Keywords: Solar sails, numerical methods, lunar pole-sitter, restricted three-body problem.

INTRODUCTION

Systematically designing a complicated spacecraft trajectory is not always a straightforward procedure. The design problem may involve a nonlinear chaotic dynamical environment, path constraints, multiple types of dynamic phases along the path, and control variables. Often, neither the shape nor the corresponding control history are known *a priori*. Additionally, multiple trajectory design options are desired for trade study purposes and ultimately *optimal* trajectories are sought. In such cases, developing robust numerical techniques is as essential as the resulting trajectories.

Collocation is a powerful approach for solving these types of difficult problems in engineering. In a collocation process, an approximate trajectory is decomposed into many discrete points, or nodes. The nodes are interpolated using piecewise continuous polynomials, and subsequently adjusted until the polynomials satisfy the system differential equations. Collocation best distributes sensitivities over the entire trajectory, allowing for large basins of attraction. Trajectories are computed efficiently by exploiting functional independencies. Doedel successfully used collocation to solve singular perturbation problems, problems with derivative discontinuities, and homoclinic bifurcation problems [1]. In Betts [2], collocation was used to solve over 70 different trajectory optimization problems.

To demonstrate the method, the collocation approach is applied to the problem of computing solar sail lunar pole-sitter orbits. Using the thrust provided by the sail, it is theoretically possible to continually maintain lunar south pole communications with only one spacecraft. Although the technology to support these thrust magnitudes is still in development, the option remains enticing since most studies indicate that at least two satellites are necessary for complete coverage [3, 4]. The solar sail design concept has been thoroughly examined by McInnes [5]. Since the recent lunar initiative announced by NASA in 2004, renewed interest in the lunar pole-sitter has resulted in detailed analyses by Ozimek *et al*. [6] and West [7], and continues to be explored by other researchers as well [8-10].

In this study, the systematic collocation approach for computing orbits is outlined and analyzed within the context of designing periodic solutions for the sail lunar pole-sitter problem. A global Fourier series control law is a natural choice for this application due to the resulting periodic control histories. Elevation angle and altitude bounds are selected to ensure continuous coverage. A simple initial guess is available by exploiting basic knowledge of the dynamical system. After obtaining a preliminary solution with collocation, the distribution of discrete points that approximates the continuous trajectory, or the mesh, is refined until a desired integration accuracy is achieved. The monodromy matrix is computed and used to analyze the stability of the solutions. The solutions are compared to and validated against the accuracy of standard explicit propagators, such as those that appear in MATLAB. In summary, the method serves as a stand-alone procedure for designing complex orbits to high levels of numerical precision.

METHODOLOGY

The general solution procedure for solving problems with collocation is first outlined in detail. The application for this study is the computation of periodic orbits subject to a given control law and constrained path. However, the method is sufficiently general to apply to many different problems in trajectory design. The authors have successfully implemented adaptions of the following procedure to compute flybys in the two-body problem (with or without low-thrust), uncontrolled periodic orbits in the restricted

*Address correspondence to this author at the School of Aeronautics and Astronautics, Purdue University, Armstrong Hall of Engineering, 701 W. Stadium Ave, West Lafayette, Indiana 47907-2045, USA; Tel: +1 765 49-45786; Fax: 1-765-494-0307; E-mail: howell@purdue.edu

1874-1460/10

three-body problem, low-thrust Earth-moon transfers, pole-sitter trajectories supported by electric propulsion, and high fidelity modeling with ephemerides.

The collocation scheme described here was originally published by Herman [11]. Herman demonstrated that higher-order Gauss-Lobatto methods are generally more robust and more efficient than lower-order Simpson and trapezoidal schemes. Lower-order methods also require a much larger discretization and therefore risk a greater chance of accumulating round-off error when step-sizes become small. He notes that, as a general rule, the order of the integration scheme is best selected to equal the desired number of digits of accuracy. Thus, he constructed the collocation scheme with order of accuracy 12 as described below. Although automatic node placement is less important for higher-order methods, it is still essential. The error cannot be bounded without employing some method of mesh refinement. In brief, mesh refinement works to satisfy two objectives: (1) to equally space the error across every segment, and (2) to reduce the error below a user-specified tolerance. Objective (1) supports efficiency and eliminates large round-off errors resulting from small step-sizes. Objective (2) mitigates undesirable behavior in the trajectory that arise as artifacts of numerical error. The refinement procedure outlined in this section and used in the following study is an adaption from de Boor [12].

General Problem

Consider the general objective of computing an orbit $\mathbf{x}(t)$ that satisfies a set of governing ordinary differential equations (ODEs)

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}, \boldsymbol{\mu}) \tag{1}$$

where t is the time, $\mathbf{x}$ is the state vector, $\mathbf{u}$ is the control vector, and $\boldsymbol{\mu}$ is the parameter vector (vectors and matrices are represented by bold-faced non-italicized characters, while non-bold italic characters refer to scalar quantities). The dot in Eq. (1) indicates a derivative with respect to time t. The trajectory may also be required to continuously satisfy the path constraints

$$\mathbf{g}(\mathbf{x}, \boldsymbol{\eta}) = \tilde{\mathbf{g}}(\mathbf{x}) + \boldsymbol{\eta}^2 = 0 \tag{2}$$

Here $\boldsymbol{\eta}^2$ refers to the element-wise square of the vector $\boldsymbol{\eta}$. Equation (2) encompasses inequality constraints of the form $\tilde{\mathbf{g}}(\mathbf{x}) < 0$ by introducing the slack variable vector $\boldsymbol{\eta}$. The periodicity condition is

$$\mathbf{h}(\mathbf{x}) = \mathbf{x}(t+T) - \mathbf{x}(t) = 0 \tag{3}$$

where T is the period of the orbit. A natural solution without any control may not satisfy Eqs. (1)-(3). Thus, solving for $\mathbf{x}(t)$ normally requires a control history or control law $\mathbf{u}(t)$. This study assumes a periodic control law is available. While maximizing or minimizing some objective function of the state variables is often desired, optimality will indirectly be addressed by modifying the prescribed inequality bounds to the limits of available solutions.

Collocation

One approach to solve the problem is to employ seventh-degree piecewise continuous polynomials and the method of collocation. Let n nodes partition the solution into $n-1$ segments consistent with the fixed mesh

$$\Pi : t_1 < t_2 < \ldots < t_{n-1} < t_n \tag{4}$$

The time interval along a given segment can be converted from $[t_i, t_{i+1}]$ to $\tau \in [0,1]$ using

$$\tau = \frac{t - t_i}{\Delta t_i} \tag{5}$$

where $\Delta t_i = t_{i+1} - t_i$. Then the polynomials representing the segment are

$$\mathbf{x}(\tau) = \mathbf{A} \times \left\{1 \quad \tau \quad \tau^2 \quad \tau^3 \quad \tau^4 \quad \tau^5 \quad \tau^6 \quad \tau^7\right\}^T \tag{6}$$

where $\mathbf{A}$ is the matrix of coefficients. Let $\mathbf{x}_j = \mathbf{x}(\tau_j)$, $\mathbf{u}_j = \mathbf{u}(\tau_j)$, and $\mathbf{x}'_j = \mathbf{x}'(\tau_j)$, where the subscript '$j$' is defined in Fig. (**1**). Prime indicates a derivative with respect to normalized time τ, i.e., $\mathbf{x}'_j = \Delta t_i \mathbf{f}(t_j, \mathbf{x}_j, \mathbf{u}_j, \boldsymbol{\mu})$. Only the four points $\mathbf{x}_i$, $\mathbf{x}_{i,2}$, $\mathbf{x}_{i,3}$, $\mathbf{x}_{i+1}$ are necessary to uniquely determine the polynomials represented by Eq. (6). Additionally there are three defect points and, therefore, seven points total are required to construct the Gauss-Lobatto integration constraints. The points are distributed on the normalized time interval according to the set $\{0, \tau_{i,1}, \tau_{i,2}, \tau_{i,c}, \tau_{i,3}, \tau_{i,4}, 1\}$. The values of $\tau_{i,1}$, $\tau_{i,2}$, $\tau_{i,c}$, $\tau_{i,3}$, $\tau_{i,4}$ are the same for every segment and selected to minimize the local truncation error (Table **1**). Recall that the known points on the segment are $\mathbf{x}_i$, $\mathbf{x}_{i,2}$, $\mathbf{x}_{i,3}$, $\mathbf{x}_{i+1}$. From Eq. (6), the points must satisfy

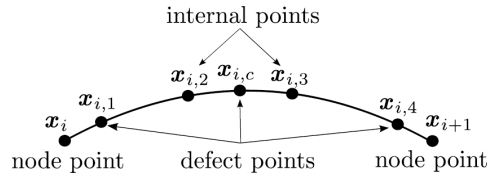


Fig. (1). The seventh-degree Gauss-Lobatto segment.

$$\left\{\mathbf{x}_i \quad \mathbf{x}'_i \quad \mathbf{x}_{i,2} \quad \mathbf{x}'_{i,2} \quad \mathbf{x}_{i,3} \quad \mathbf{x}'_{i,3} \quad \mathbf{x}_{i+1} \quad \mathbf{x}'_{i+1}\right\} = \mathbf{AB} \tag{7}$$

where

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & \tau_{i,2} & 1 & \tau_{i,4} & 1 & 1 & 1 \\ 0 & 0 & \tau_{i,2}^2 & 2\tau_{i,2} & \tau_{i,3}^2 & 2\tau_{i,3} & 1 & 2 \\ 0 & 0 & \tau_{i,2}^3 & 3\tau_{i,2}^2 & \tau_{i,3}^3 & 3\tau_{i,3}^2 & 1 & 3 \\ 0 & 0 & \tau_{i,2}^4 & 4\tau_{i,2}^3 & \tau_{i,3}^4 & 4\tau_{i,3}^3 & 1 & 4 \\ 0 & 0 & \tau_{i,2}^5 & 5\tau_{i,2}^4 & \tau_{i,3}^5 & 5\tau_{i,3}^4 & 1 & 5 \\ 0 & 0 & \tau_{i,2}^6 & 6\tau_{i,2}^5 & \tau_{i,3}^6 & 6\tau_{i,3}^5 & 1 & 6 \\ 0 & 0 & \tau_{i,2}^7 & 7\tau_{i,2}^6 & \tau_{i,3}^7 & 7\tau_{i,3}^6 & 1 & 7 \end{bmatrix} \tag{8}$$

Since the left-hand side of Eq. (7) is given and $\mathbf{B}$ is a known (constant) matrix, the coefficients $\mathbf{A}$ can be computed from Eq. (7). Then, to satisfy the system equations, the time derivatives of the polynomials must also satisfy the ODEs in Eq. (1) at the defect points $\mathbf{x}_{i,1}$, $\mathbf{x}_{i,c}$, and $\mathbf{x}_{i,4}$. (See Fig. (**2**) for a geometric representation of the defect constraints.) Using Eq. (6), the interpolated expressions for these quantities are

$$\mathbf{x}_{i,1} = a_i^1\mathbf{x}_i + a_{i,2}^1\mathbf{x}_{i,2} + a_{i,3}^1\mathbf{x}_{i,3} + a_{i+1}^1\mathbf{x}_{i+1} + \Delta t_i\left(v_i^1\mathbf{f}_i + v_{i,2}^1\mathbf{f}_{i,2} + v_{i,3}^1\mathbf{f}_{i,3} + v_{i+1}^1\mathbf{f}_{i+1}\right) \tag{9}$$

$$\mathbf{x}_{i,c} = a_i^c\mathbf{x}_i + a_{i,2}^c\mathbf{x}_{i,2} + a_{i,3}^c\mathbf{x}_{i,3} + a_{i+1}^c\mathbf{x}_{i+1} + \Delta t_i\left(v_i^c\mathbf{f}_i + v_{i,2}^c\mathbf{f}_{i,2} + v_{i,3}^c\mathbf{f}_{i,3} + v_{i+1}^c\mathbf{f}_{i+1}\right) \tag{10}$$

$$\mathbf{x}_{i,4} = a_i^4\mathbf{x}_i + a_{i,2}^4\mathbf{x}_{i,2} + a_{i,3}^4\mathbf{x}_{i,3} + a_{i+1}^4\mathbf{x}_{i+1} + \Delta t_i\left(v_i^4\mathbf{f}_i + v_{i,2}^4\mathbf{f}_{i,2} + v_{i,3}^4\mathbf{f}_{i,3} + v_{i+1}^4\mathbf{f}_{i+1}\right) \tag{11}$$

where $\mathbf{f}_j = \mathbf{f}(t_j, \mathbf{x}_j, \mathbf{u}_j, \boldsymbol{\mu})$. The resulting defect constraint equations are

$$\boldsymbol{\Delta}_{i,1}(\mathbf{x}_i, \mathbf{x}_{i,2}, \mathbf{x}_{i,4}, \mathbf{x}_{i+1}, \boldsymbol{\mu}) = b_i^1\mathbf{x}_i + b_{i,2}^1\mathbf{x}_{i,2} + b_{i,3}^1\mathbf{x}_{i,3} + b_{i+1}^1\mathbf{x}_{i+1} + \Delta t_i\left(w_i^1\mathbf{f}_i + w_{i,1}^1\mathbf{f}_{i,1} + w_{i,2}^1\mathbf{f}_{i,2} + w_{i,3}^1\mathbf{f}_{i,3} + w_{i+1}^1\mathbf{f}_{i+1}\right) = 0 \tag{12}$$

$$\boldsymbol{\Delta}_{i,c}(\mathbf{x}_i, \mathbf{x}_{i,2}, \mathbf{x}_{i,4}, \mathbf{x}_{i+1}, \boldsymbol{\mu}) = b_i^c\mathbf{x}_i + b_{i,2}^c\mathbf{x}_{i,2} + b_{i,3}^c\mathbf{x}_{i,3} + b_{i+1}^c\mathbf{x}_{i+1} + \Delta t_i\left(w_i^c\mathbf{f}_i + w_{i,2}^c\mathbf{f}_{i,2} + w_{i,c}^c\mathbf{f}_{i,c} + w_{i,3}^c\mathbf{f}_{i,3} + w_{i+1}^c\mathbf{f}_{i+1}\right) = 0 \tag{13}$$

$$\boldsymbol{\Delta}_{i,4}(\mathbf{x}_i, \mathbf{x}_{i,2}, \mathbf{x}_{i,4}, \mathbf{x}_{i+1}, \boldsymbol{\mu}) = b_i^4\mathbf{x}_i + b_{i,2}^4\mathbf{x}_{i,2} + b_{i,3}^4\mathbf{x}_{i,3} + b_{i+1}^4\mathbf{x}_{i+1} + \Delta t_i\left(w_i^4\mathbf{f}_i + w_{i,2}^4\mathbf{f}_{i,2} + w_{i,3}^4\mathbf{f}_{i,3} + w_{i,4}^4\mathbf{f}_{i,4} + w_{i+1}^4\mathbf{f}_{i+1}\right) = 0 \tag{14}$$

Recall that time is fixed consistent with the mesh Π and it is assumed that a control law $\mathbf{u}(t)$ is available for interpolating the control. Therefore t and $\mathbf{u}$ do not appear in the functional dependencies on the left side of Eqs. (12)-(14). The coefficients a, v, b, and w that appear in Eqs. (9)-(14) are the same for every segment, and are listed in Table **1**.

The path and periodicity constraints are discretized in a similar manner. The path constraints are enforced at the node points and internal points such that

$$\mathbf{g}_j(\mathbf{x}_j, \boldsymbol{\eta}_j) = \tilde{\mathbf{g}}_j(\mathbf{x}_j) + \boldsymbol{\eta}_j^2 = 0 \tag{15}$$

Similarly, the periodicity constraint is

$$\mathbf{h}(\mathbf{x}_1, \mathbf{x}_n) = \mathbf{x}_n - \mathbf{x}_1 = 0 \tag{16}$$

The variables $\mathbf{x}_j$, $\boldsymbol{\eta}_j$, and $\boldsymbol{\mu}$ are varied until Eqs. (12)-(16) are satisfied. Since each segment only depends on the adjacent segments, this process generally requires little computational effort.

It is important to note that sometimes a control law is not available. In such cases, either a linearly interpolated or splined control may be employed and, then, the defect constraints will depend on the node point controls as well. For very large dimensioned problems, it is sometimes useful to discretize the problem parameters in the vector $\boldsymbol{\mu}$ by assuming that they are independent for each segment. Additional constraint equations are then necessary to enforce the condition that the parameters are equal from one segment to the next. Although this approach may significantly decrease computation time, for the purposes of this study it is assumed that all segments depend on the same single vector $\boldsymbol{\mu}$.

Satisfying the Constraints

Consider organizing the variables and constraints as follows. The variables are stored in the total variable vector $\mathbf{X}$, i.e.

$$\mathbf{X}^T = (\mathbf{x}_1^T, \mathbf{x}_{1,2}^T, \mathbf{x}_{1,3}^T, \mathbf{x}_2^T, \mathbf{x}_{2,2}^T, \mathbf{x}_{2,3}^T, \ldots, \mathbf{x}_n^T, \boldsymbol{\eta}_1^T, \boldsymbol{\eta}_{1,2}^T, \boldsymbol{\eta}_{1,3}^T, \boldsymbol{\eta}_2^T, \boldsymbol{\eta}_{2,2}^T, \boldsymbol{\eta}_{2,3}^T, \ldots, \boldsymbol{\eta}_n^T, \boldsymbol{\mu}^T) \tag{17}$$

The complete constraint vector $\mathbf{F}(\mathbf{X})$ is

$$\mathbf{F}(\mathbf{X})^T = \left(\boldsymbol{\Delta}_{1,1}^T, \boldsymbol{\Delta}_{1,c}^T, \boldsymbol{\Delta}_{1,4}^T, \boldsymbol{\Delta}_{2,1}^T, \boldsymbol{\Delta}_{2,c}^T, \boldsymbol{\Delta}_{2,4}^T, \ldots, \boldsymbol{\Delta}_{n-1,1}^T, \boldsymbol{\Delta}_{n-1,c}^T, \boldsymbol{\Delta}_{n-1,4}^T, \mathbf{g}_1^T, \mathbf{g}_{1,2}^T, \mathbf{g}_{1,3}^T, \mathbf{g}_2^T, \mathbf{g}_{2,2}^T, \mathbf{g}_{2,3}^T, \ldots, \mathbf{g}_n^T, \mathbf{h}^T\right) = 0 \tag{18}$$

There are many ways to satisfy the constraints, but perhaps the simplest is a least-squares Newton's method. Provided that an initial guess $\mathbf{X}_k$ is available, a first-order

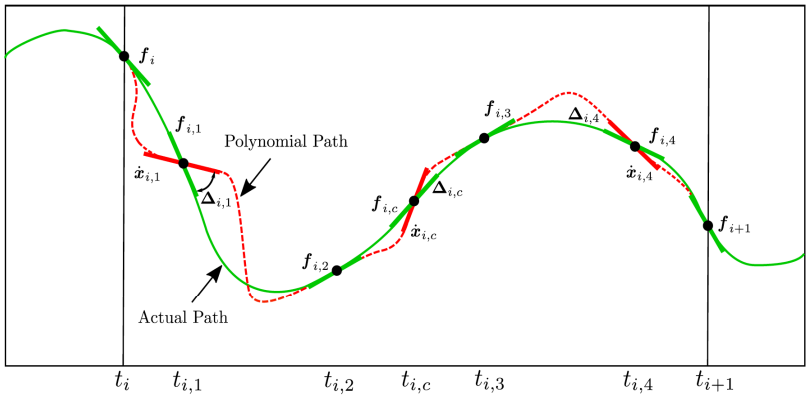


Fig. (2). Defect constraint illustration.

Table 1. List of Constants for Numerical Integration

$\tau_{i,1} = +8.48880518607166\text{d}-2$					
$\tau_{i,2} = +2.65575603264643\text{d}-1$					
$\tau_{i,c} = +5\text{d}-1$					
$\tau_{i,3} = +7.34424396735357\text{d}-1$					
$\tau_{i,4} = +9.15111948139283\text{d}-1$					
a_i^1	=	+6.18612232711785d – 1	b_i^1	=	+8.84260109348311d – 1
$a_{i,2}^1$	=	+3.34253095933642d – 1	$b_{i,2}^1$	=	–8.23622559094327d – 1
$a_{i,3}^1$	=	+1.52679626438851d – 2	$b_{i,3}^1$	=	–2.35465327970606d – 2
a_{i+1}^1	=	+3.18667087106879d – 2	b_{i+1}^1	=	–3.70910174569208d – 2
v_i^1	=	+2.57387738427162d – 2	w_i^1	=	+1.62213410652341d – 2
			$w_{i,1}^1$	=	+1.38413023680783d – 1
$v_{i,2}^1$	=	–5.50098654524528d – 2	$w_{i,2}^1$	=	+9.71662045547156d – 2
$v_{i,3}^1$	=	–1.53026046503702d – 2	$w_{i,3}^1$	=	+1.85682012187242d – 2
v_{i+1}^1	=	–2.38759243962924d – 3	w_{i+1}^1	=	+2.74945307600086d – 3
a_i^c	=	+1.41445282326366d – 1	b_i^c	=	+7.86488731947674d – 2
$a_{i,2}^c$	=	+3.58554717673634d – 1	$b_{i,2}^c$	=	+8.00076026297266d – 1
$a_{i,3}^c$	=	+3.58554717673634d – 1	$b_{i,3}^c$	=	–8.00076026297266d – 1
a_{i+1}^c	=	+1.41445282326366d – 1	b_{i+1}^c	=	–7.86488731947674d – 2
v_i^c	=	+9.92317607754556d – 3	w_i^c	=	+4.83872966828888d – 3
$v_{i,2}^c$	=	+9.62835932121973d – 2	$w_{i,2}^c$	=	+1.00138284831491d – 1
			$w_{i,c}^c$	=	+2.43809523809524d – 1
$v_{i,3}^c$	=	–9.62835932121973d – 2	$w_{i,3}^c$	=	+1.00138284831491d – 1
v_{i+1}^c	=	–9.92317607754556d – 3	w_{i+1}^c .	=	+4.83872966828888d – 3
a_i^4	=	+3.18667087106879d – 2	b_i^4	=	+3.70910174569208d – 2
$a_{i,2}^4$	=	+1.52679626438851d – 2	$b_{i,2}^4$	=	+2.35465327970606d – 2
$a_{i,3}^4$	=	+3.34253095933642d – 1	$b_{i,3}^4$	=	+8.23622559094327d – 1
a_{i+1}^4	=	+6.18612232711785d – 1	b_{i+1}^4	=	–8.84260109348311d – 1
v_i^4	=	+2.38759243962924d – 3	w_i^4	=	+2.74945307600086d – 3
$v_{i,2}^4$	=	+1.53026046503702d – 2	$w_{i,2}^4$	=	+1.85682012187242d – 2
$v_{i,3}^4$	=	+5.50098654524528d – 2	$w_{i,3}^4$	=	+9.71662045547156d – 2
			$w_{i,4}^4$.	=	+1.38413023680783d – 1
v_{i+1}^4	=	–2.57387738427162d – 2	w_{i+1}^4	=	+1.62213410652341d – 2

Taylor series expansion about $\mathbf{X}_k$ yields

$$\begin{aligned}\mathbf{F}(\mathbf{X}_k + \delta\mathbf{X}_k) &\approx \mathbf{F}(\mathbf{X}_k) + D\mathbf{F}(\mathbf{X}_k)\cdot\delta\mathbf{X}_k \\ &\approx 0\end{aligned} \tag{19}$$

where $\delta\mathbf{X}_k = \mathbf{X}_{k+1} - \mathbf{X}_k$. Generally, there are an infinite number of solutions $\delta\mathbf{X}_k$ that satisfy Eq. (19), however, a unique solution is determined by minimizing $\| \delta\mathbf{X}_k \|^2$. The solution is

$$\delta\mathbf{X}_k = -D\mathbf{F}(\mathbf{X}_k)^T \left[D\mathbf{F}(\mathbf{X}_k)\cdot D\mathbf{F}(\mathbf{X}_k)^T \right]^{-1} \mathbf{F}(\mathbf{X}) \tag{20}$$

Newton's method converges quadratically to a nearby solution by iteration over k using the update equation $\mathbf{X}_{k+1} = \mathbf{X}_k + \delta\mathbf{X}_k$, where $\delta\mathbf{X}_k$ is computed from Eq. (20). A solution that minimizes $\| \delta\mathbf{X}_k \|^2$ is a natural choice for a first-order series approximation and ultimately leads to solutions that best preserve the design characteristics that the initial guess may possess. The derivatives in the Jacobian matrix $D\mathbf{F}$ can be computed analytically, from finite-differencing, or a combination of both. In the following, all the derivatives are computed analytically except for the derivatives of the defect constraint equations. Since the expressions for $\boldsymbol{\Delta}_j$ are involved, these derivatives are computed using the complex-step method. The complex-step method is selected for its efficiency and double-precision accuracy. Note that, for large problems, $D\mathbf{F}$ is very large but also very sparse. Efficient algorithms are

available for computing $\left[D\mathbf{F} \cdot D\mathbf{F}^T \right]^{-1} \mathbf{F}$. (See Ozimek *et al.* [6] for a discussion on computing the Jacobian matrix and exploiting sparsity).

Mesh Refinement

An optimal mesh (1) equally distributes the error associated with each segment, and (2) reduces the equally distributed error below a user-specified tolerance. Therefore, mesh refinement is based on an error analysis between the actual and approximate polynomial solutions. The previously described Gauss-Lobatto scheme has an order of accuracy equal to 12. Since the order of the method is greater than eight (one more than the degree), the error for the i^{th} segment is

$$e_i = C\Delta t_i^8 \| \mathbf{x}^{(8)} \|_i + O(\Delta t_i^9) \tag{21}$$

where $\mathbf{x}^{(8)}$ is the eighth time derivative of $\mathbf{x}$. Using the analysis presented in the Appendix of Russell and Christiansen [13], the constant C (dimensionless) for the seventh-degree Gauss-Lobatto scheme is

$$C = 2.93579395141895\text{d} - 9 \tag{22}$$

In general, $\mathbf{x}^{(8)}$ is unknown. In fact, using the seventh-degree polynomial representation in Eq. (6), $\mathbf{x}^{(8)} = 0$. De Boor [12] circumnavigates this potential problem by approximating $\| \mathbf{x}^{(8)} \|$ with the piecewise constant function $\mathbf{x}^{(7)}(\tau)$ and a difference formula. From de Boor

$$\| \mathbf{x}^{(8)} \| \approx \theta(t) \overset{\Delta}{=} \begin{cases} \max\left[2\dfrac{|\mathbf{y}_1 - \mathbf{y}_2|}{\Delta t_1 + \Delta t_2} \right] & , \text{ on } (t_1, t_2) \\ \max\left[\dfrac{|\mathbf{y}_{i-1} - \mathbf{y}_i|}{\Delta t_{i-1} + \Delta t_i} + \dfrac{|\mathbf{y}_{i+1} - \mathbf{y}_{i+2}|}{\Delta t_{i+1} + \Delta t_{i+2}} \right] & , \text{ on } (t_i, t_{i+1}),\ i = 2, \ldots, n-2 \\ \max\left[2\dfrac{|\mathbf{y}_{n-2} - \mathbf{y}_{n-1}|}{\Delta t_{n-2} + \Delta t_{n-1}} \right] & , \text{ on } (t_{n-1}, t_n) \end{cases} \tag{23}$$

where

$$\begin{aligned} \mathbf{y}_i &= \mathbf{x}^{(7)}(\tau) / \Delta t_i^7 \quad \text{on } (t_i, t_{i+1}) \\ &= 7! \left\{ \mathbf{x}_i \quad \mathbf{x}_i^{'} \quad \mathbf{x}_{i,2} \quad \mathbf{x}_{i,2}^{'} \quad \mathbf{x}_{i,3} \quad \mathbf{x}_{i,3}^{'} \quad \mathbf{x}_{i+1} \quad \mathbf{x}_{i+1}^{'} \right\} \times \mathbf{b} / \Delta t_i^7 \end{aligned} \tag{24}$$

The vector $\mathbf{b}$ in Eq. (24) is the last column of $\mathbf{B}^{-1}$.

Equidistribution

A potential solution that satisfies the problem constraints may have a widely varying error Δe_i for the initial mesh Π. The function $\theta(t)$ can be used to determine a new mesh that asymptotically equidistributes the error. The new mesh points are selected such that

$$t_{i+1} = I^{-1}\left[\frac{iI(t_n)}{n-1} \right], i = 1, \ldots, n-2 \tag{25}$$

where

$$I(t) = \int_{t_1}^{t} \theta(s)^{1/8} ds \tag{26}$$

Since $\theta(t)$ is a piecewise constant function, the integral $I(t)$ is a monotonically increasing piecewise linear function and can easily be determined using a rectangle rule. In Eq. (25), I^{-1} represents the inverse integral, and the process reduces to solving for t where $I(t)$ takes the value in the argument. Given the new mesh, the polynomial expressions are used to interpolate the state variables associated with the new times. Interpolating the new state variables minimizes the constraint violation introduced from the new mesh. The slack variables are selected such that the path constraints are initially satisfied. A new solution with a better equidistribution of error is then computed with Newton's method. The process repeats until the error is equally distributed and within a specified equidistribution tolerance.

Meeting an Integration Tolerance

Given a specified number of nodes n, generally one equidistribution iteration sufficiently distributes the error. If the error has been sufficiently equidistributed, the number of nodes is updated with

$$n = \left(\frac{\bar{e}}{\varepsilon / 10} \right)^{1/8} \tag{27}$$

The variable $\bar{e}$ represents the mean error associated with the equidistributed mesh ($\bar{e} \approx \Delta e_i$). The user-defined tolerance is ε. Equation (27) determines the number of nodes required such that the error will reach an order of magnitude less than the specified tolerance. Given the new number of nodes n, a new mesh Π is then constructed with Eq. (25). Again, the state variables for the mesh are interpolated from the polynomial and slack variables are computed that initially satisfy the path constraints. The approximation is reconverged with the Newton's method procedure. A series of equidistribution iterations brings the maximum error (and, therefore, also the maximum difference in error) below ε.

In summary, the algorithm runs as follows:

1. Obtain an initial guess $\mathbf{X}$ and Π.
2. Update $\mathbf{X}$ until $\mathbf{F}(\mathbf{X}) = 0$ with Newton's method.
3. Update Π according to Eq. (25), construct new $\mathbf{X}$.
4. Repeat 2-3 until the error is approximately equidistributed (start with 2, end with 2).
5. Update n using Eq. (27).
6. Repeat 2-3 until the error is below ε (start with 3, end with 2).

The process usually requires only a few refinements to meet the desired tolerance.

Linear Stability Analysis

For a small perturbation initially applied to the trajectory, the stability analysis measures the rate of departure from the reference path when the trajectory returns to a hyperplane. Since the Jacobian matrix for the constraint equations contains sensitivity information, the state-transition matrix can be extracted directly from the Jacobian matrix of the converged

solution. For a converged solution, $\mathbf{F}(\mathbf{X}) = 0$ in Eq. (19). Then by isolating the state variations associated with the i^{th} segment

$$\underbrace{\begin{bmatrix} \frac{d\boldsymbol{\Delta}_{i,1}}{d\mathbf{x}_i} & \frac{d\boldsymbol{\Delta}_{i,1}}{d\mathbf{x}_{i,2}} & \frac{d\boldsymbol{\Delta}_{i,1}}{d\mathbf{x}_{i,3}} & \frac{d\boldsymbol{\Delta}_{i,1}}{d\mathbf{x}_{i+1}} \\ \frac{d\boldsymbol{\Delta}_{i,c}}{d\mathbf{x}_i} & \frac{d\boldsymbol{\Delta}_{i,c}}{d\mathbf{x}_{i,2}} & \frac{d\boldsymbol{\Delta}_{i,c}}{d\mathbf{x}_{i,3}} & \frac{d\boldsymbol{\Delta}_{i,c}}{d\mathbf{x}_{i+1}} \\ \frac{d\boldsymbol{\Delta}_{i,4}}{d\mathbf{x}_i} & \frac{d\boldsymbol{\Delta}_{i,4}}{d\mathbf{x}_{i,2}} & \frac{d\boldsymbol{\Delta}_{i,4}}{d\mathbf{x}_{i,3}} & \frac{d\boldsymbol{\Delta}_{i,4}}{d\mathbf{x}_{i+1}} \end{bmatrix}}_{\mathbf{M}} \begin{Bmatrix} \delta\mathbf{x}_i \\ \delta\mathbf{x}_{i,2} \\ \delta\mathbf{x}_{i,3} \\ \delta\mathbf{x}_{i+1} \end{Bmatrix} = 0 \tag{28}$$

In general, the matrix $\mathbf{M}$ possesses a nullspace with dimension equal to the dimension of $\mathbf{x}$. Therefore, it is possible to eliminate $\delta\mathbf{x}_{i,2}$ and $\delta\mathbf{x}_{i,3}$ in Eq. (28) and solve for $\delta\mathbf{x}_{i+1}$ as a linear combination of $\delta\mathbf{x}_i$. More precisely, Eq. (28) allows computation of a matrix $\Phi(t_{i+1}, t_i)$ such that

$$\delta\mathbf{x}_{i+1} = \Phi(t_{i+1}, t_i)\delta\mathbf{x}_i \tag{29}$$

The matrix $\Phi(t_{i+1}, t_i)$ is known as the state-transition matrix. A general algorithm for computing $\Phi(t_{i+1}, t_i)$ in Eq. (29) applies to every segment. Then, the monodromy matrix is

$$\Phi(t_n, t_1) = \Phi(t_n, t_{n-1}) \cdot \Phi(t_{n-1}, t_{n-2}) \cdots \Phi(t_3, t_2) \cdot \Phi(t_2, t_1) \tag{30}$$

While eigenvalues of $\Phi(t_n, t_1)$ inside the unit circle correspond to locally stable modes, eigenvalues outside the unit circle indicate instability.

APPLICATION TO LUNAR POLE-SITTER ORBIT

Orbits are designed in the Earth-moon restricted three-body problem with the addition of acceleration forces due to a solar sail. In this model, the Earth and moon are assumed to move in circular orbits, and the spacecraft possesses negligible mass in comparison to the Earth and moon. A rotating, barycentric coordinate frame is employed, with the x-axis directed from the Earth to the moon. The z-axis is parallel to the angular velocity vector of the Earth-moon system. Then the equations of motion for the system are

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}) = \begin{pmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{v}} \end{pmatrix} = \begin{Bmatrix} \mathbf{v} \\ \kappa(\mathbf{l}^T\mathbf{u})^2\,\mathbf{u} - 2\boldsymbol{\Omega} \times \mathbf{v} + \nabla^T U(\mathbf{r}) \end{Bmatrix} \tag{31}$$

where the ∇^T operator refers to the gradient-transpose. The components are derivable from the potential function

$$U = \frac{1-\gamma}{\|\mathbf{r} - \mathbf{r}_1\|} + \frac{\gamma}{\|\mathbf{r} - \mathbf{r}_2\|} + \frac{1}{2}(x^2 + y^2) \tag{32}$$

and x, y, and z are the components of the spacecraft's position relative to the rotating, barycentric frame. The mass parameter is γ, the Earth-moon angular velocity is $\boldsymbol{\Omega}$, and $\mathbf{r}_1$ and $\mathbf{r}_2$ are the positions of the Earth and moon, respectively. Equation (31) is also nondimensional, where the characteristic quantities are the total mass m^* of the system, the distance l^* between the Earth and moon, and the familiar characteristic time $t^* = (l^{*3} / Gm^*)^{1/2}$. (The quantity G is the universal gravitational constant.) The magnitude of the solar radiation pressure force supplied by the sail at 1 AU is defined as the constant characteristic acceleration κ, and it is directed along a unit-vector $\mathbf{u}$, the control parameter normal to the surface. An idealized, perfectly reflective sail acceleration model is assumed, where $\mathbf{l}$ is the unit-vector directed from the sun to the spacecraft. This formulation for sail acceleration is valid as long as $\mathbf{l}^T\mathbf{u} \geq 0$. The vector $\mathbf{l}$ is simplified to rotate in a circular orbit within the Earth-moon plane once per synodic lunar month, or with angular rate ω_s, i.e.

$$\mathbf{l} = \{\cos(\omega_s t), -\sin(\omega_s t), 0\}^T \tag{33}$$

Lunar south pole line-of-sight using only one spacecraft requires a path constraint on the minimum elevation angle φ_{lb}. It is also desirable to bound the spacecraft below some maximum altitude d_{ub}. The continuous path constraints as defined by Ozimek *et al.* [6] are

$$\mathbf{g}(\mathbf{r}, \boldsymbol{\eta}) = \begin{Bmatrix} \sin\varphi_{lb} + \dfrac{z + R_m}{a} \\ d - d_{ub} \end{Bmatrix} + \boldsymbol{\eta}^2 = 0 \tag{34}$$

where $d = \sqrt{(x - 1 + \gamma)^2 + y^2 + (z + R_m)^2}$ and R_m is the nondimensional mean radius of the moon. The periodicity constraint is chosen to be synchronous with one lunar synodic month, or period $T = 2\pi / \omega_s \approx 29.64$ days.

Control Law

The components of the control vector $\mathbf{u}$ in Eq. (31) are governed by the angles δ and α. The clock angle δ is the angle between $\mathbf{l}$ and $\mathbf{u}$ projected into the Earth-moon plane. The pitch angle α measures the out-of-plane angle associated with the control direction. Thus, the control $\mathbf{u}$ is

$$\mathbf{u} = \begin{Bmatrix} \cos\alpha\cos(\delta - \omega_s t) \\ \cos\alpha\sin(\delta - \omega_s t) \\ \sin\alpha \end{Bmatrix} \tag{35}$$

In a previous study by the authors [6], the controls were varied at the node points. The angles α and δ were then computed after the determination of a control history that satisfied the problem constraints. Therefore, it was necessary to add a periodicity constraint on the control. For the trajectories previously investigated, the history for the angle α resembled an even sinusoidal function, while the angle δ appeared to closely follow an odd-valued sinusoid as a function of time. Given the problem symmetries and the physical interpretation of these angles, perhaps the most natural control laws for α and δ are even and odd Fourier series. Thus, it can be assumed that α and δ can be represented as

$$\alpha(t) = \alpha_0 + \sum_{k=1}^{N} \alpha_k \cos(k\omega_s t)$$
$$\delta(t) = \sum_{k=1}^{N} \delta_k \sin(k\omega_s t) \tag{36}$$

By construction, the period of $\alpha(t)$ and $\delta(t)$ is $T = 2\pi / \omega_s$. For implementation of the control in the collocation scheme, the Fourier coefficients are varied until the precise control law is uncovered for a given orbit. This result is achieved by storing the coefficients in the problem parameters vector $\boldsymbol{\mu}$ for corrections, i.e.

$$\boldsymbol{\mu}^T = \left(\alpha_0, \alpha_1, ..., \alpha_N, \delta_1, ..., \delta_N\right) \tag{37}$$

Once a solution is computed and the corresponding vector $\boldsymbol{\mu}$ is determined, the angle rates are accessible easily. The control rates $\dot{\alpha}$ and $\dot{\delta}$, which represent changes in the angles relative to $\mathbf{l}$, follow as

$$\dot{\alpha}(t) = -\sum_{k=1}^{N} \alpha_k k\omega_s \sin(k\omega_s t)$$
$$\dot{\delta}(t) = \sum_{k=1}^{N} \delta_k k\omega_s \cos(k\omega_s t) \tag{38}$$

The Fourier series control law provided in Eq. (36) is sufficiently general for implementation in single or multiple shooting schemes with explicit integration.

Initial Guess Strategy

For this application, the initial guess process is based upon approximating the trajectory as a pole-sitter, i.e., $\mathbf{v} = 0$. Then, from Eq. (31), the natural acceleration is entirely dependent upon the position of the spacecraft according to the value of $\|\nabla U\|$. Fig. (**3**) can therefore be used to quickly locate the position where the pole-sitter trajectory would most likely occur, given a specified characteristic acceleration. The nodes comprising the trajectory are all initially assumed to be coincident at a point in space with Π selected to equally distribute the time interval. The slack variables are always initialized such that the path constraints are satisfied. This initial guess requires little *a priori* knowledge. Furthermore, the solution is not biased in favor of a particular orbit, except for those trajectories that may exist near the pole-sitting position. The process to determine solar sail lunar pole-sitter orbits can be summarized as follows:

1. Examine Fig. (**3**) to approximate a feasible region given the value of κ for the sail.
2. Specify desirable values of φ_{lb} and d_{ub} that encompass this region.
3. Within the bounds imposed by φ_{lb} and d_{ub}, approximate an initial trajectory by setting x and z to constant values ($y = \dot{x} = \dot{y} = \dot{z} = 0$ initially for all nodes).
4. Assume the sail acceleration is initially oriented to maximize the out-of-plane component, i.e., set $\alpha_0 = -35.26^\circ$ and set all other coefficients equal to zero.

Once an initial guess for $\mathbf{X}$ and Π is available, the previously defined collocation algorithm locates a nearby solution.

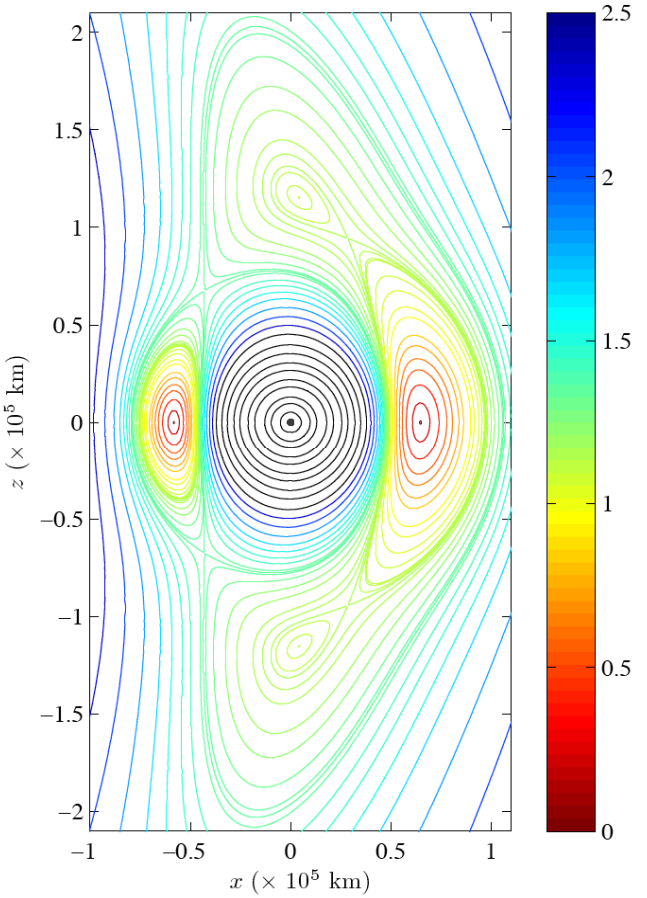


Fig. (3). Contours of $\|\nabla \mathbf{U}\|$ in mm/s^2, moon-centered rotating frame.

NUMERICAL RESULTS

The constants used for all numerical computations are listed in Table **2**.

Table 2. Problem Constants

Parameter	Value	Units
ε	1e-12	
γ	0.012150585609624	
t^*	4.36439991512776	day
l^*	385,692.5	km
R_m	1,734.4	km
ω_s	12.1423770706749	$^\circ$/day
N	5	

Table 3. Data Summary for Near-Optimal Orbits

	$\kappa = 0.58$ mm / s^2		$\kappa = 1.70$ mm / s^2		
Type	L_1	L_2	L_1	L_2	Hover
Color	Blue	Green	Red	Purple	Grey
x_0	+8.334577959416795 d- 1	+1.156211421789636 d- 0	+8.358382331873345 d- 1	+1.133709934961812 d- 0	+1.142606758444961 d- 0
z_0	−1.674447029156162 d- 2	−2.785681461021011 d- 2	−5.080489933754771 d- 2	−7.014221018336932 d- 2	−1.079440386848905 d- 1
$\dot{y}_0$	−2.970274544651623 d- 2	−7.264216970640185 d- 2	−1.035256763465011 d- 1	−1.321776217291153 d- 1	−2.309935244587937 d- 1
α_0	−6.343037134654265 d- 1	−6.532727216254634 d- 1	−6.550832921434782 d- 1	−8.076988446188386 d- 1	−7.176650914056956 d- 1
α_1	−1.639142648497209 d- 2	−3.496431271743909 d- 3	+1.109111936118494 d- 2	−3.760510052379947 d- 2	−9.455764413908209 d- 2
α_2	−1.005162216606042 d- 3	+2.525294068828327 d- 2	−7.204002256366904 d- 3	+9.198677776698262 d- 2	−1.657526877433217 d- 1
α_3	+4.186000055059244 d- 2	−6.282315518132125 d- 3	+7.724704156521697 d- 2	+4.178169912132836 d- 2	+4.081942483021073 d- 2
α_4	+1.276497472641706 d- 2	−2.827459153365777 d- 2	−9.084935129312299 d- 3	−5.380796885631299 d- 3	+9.546331387483088 d- 2
α_5	−7.108087578028680 d- 3	−1.076977595277608 .d- 2	−3.141687535102539 d- 2	+1.457063836041965 d- 2	+1.510595399780250 d- 2
δ_1	+9.923829906300495 d-2	−1.978177053068336 d- 1	+4.240662342949236 d- 1	−4.206338880883266 d- 1	−5.455275819483647 d- 1
δ_2	+1.024195843920122 d- 3	−4.157860433873927 d- 2	−7.898408737806689 d- 4	+6.788375238824441 d- 2	−6.872290868371695 d- 3
δ_3	+3.345688507820322 d- 3	−7.153711065518781 d- 2	−1.269064717140005 d- 1	−1.503551493545528 d- 1	+1.478868620356161 d- 1
δ_4	+8.333198429994938 d-4	−3.504456801267981 d- 3	+1.277367799213988 d- 1	+3.061175727108182 d- 2	+2.429599424843301 d- 2
δ_5	+6.021023233999644 d- 3	−1.440196545516193 d- 2	+1.736235859640061 .d-2	−2.337621947580810 d-2	−5.042661386056409 d-2
$\varphi_{\min}(^\circ)$ *	4.2	6.8	15.8	18.8	15.0
$\varphi_{\min}(^\circ)$	4.2	6.8	15.6	18.6	15.0
$\lambda_{\max}$	3.0×10^8	1.4×10^6	6.9×10^5	2.7×10^5	1.2×10^4
$\dot{\alpha}_{\max}(^\circ/\text{day})$	2.28	2.27	4.60	3.48	9.61
$\dot{\delta}_{\max}(^\circ/\text{day})$	1.76	7.06	12.78	15.14	10.78
Initial n	15	15	15	15	15
Initial Size **X**	355	355	355	355	355
Mesh Refinements	2	2	2	2	2
Final n	51	50	79	68	83
Final Size **X**	1,219	1,195	1,891	1,627	1,987

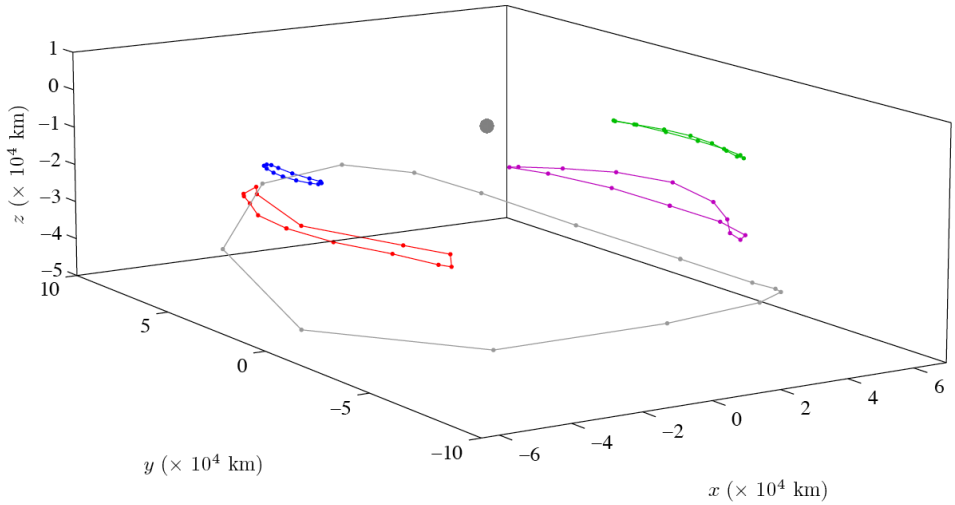
*Values from Ozimek *et al*. [6].

Near-Optimal Orbits

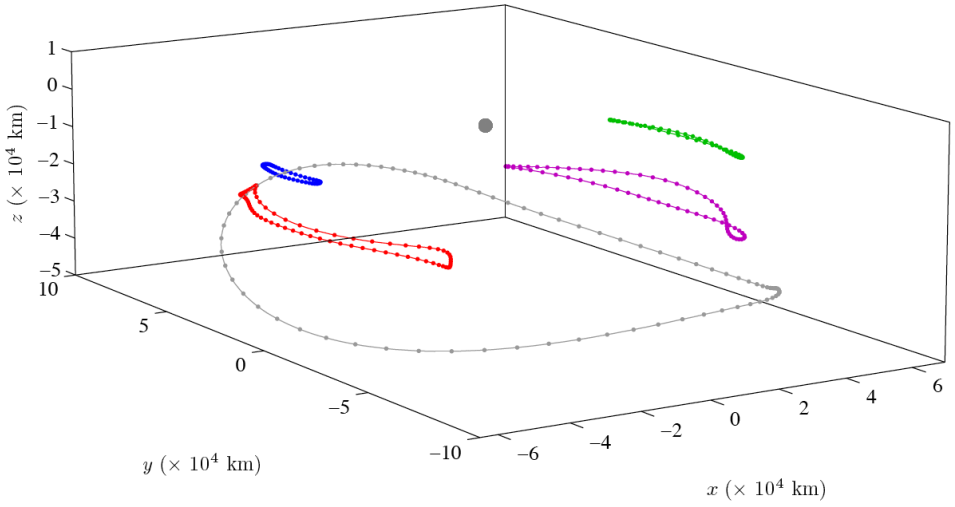
Using the previously mentioned initial guess procedure, five example orbits are generated. (See Fig. (**4**) for the initial and final converged mesh discretizations and Table **3** for initial conditions and coefficient values.) The orbits are near-optimal in the sense that the iteration process evolves with increasing values of minimum elevation angle until the limits of convergence are reached. For all of the final solutions, only two refinement iterations are required. Inspection of one of the resulting control histories reveals that a smooth, periodic control history emerges that closely reflects the angles supplied with the initial guess. (See Fig. **5**). The resulting orbits and control histories bear close resemblance to the trajectories presented in a previous study by the authors [6]. The data summary in Table **3** indicates that with only a small number of Fourier coefficients, the minimum elevation angle $\varphi_{\min}$ along any orbit deviates by at most 0.2° compared to the minimum elevation angle in the previous study. No control rate boundaries are imposed on $\dot{\alpha}$ and $\dot{\delta}$, but all rates remain close to the 12.1°/day baseline rate that is already necessary for the sail to turn to continuously face the sun. Since all of the orbits require control, it is not surprising that all are unstable. The stability analysis, measured in terms of $\lambda_{\max}$, indicates that the most unstable orbit is the L_1, $\kappa = 0.58$ mm/s^2 orbit. The least unstable orbit is the hover orbit.

Comparison to Explicit Schemes

The nondimensional quantities x_0, z_0, and $\dot{y}_0$ ($y_0 = \dot{x}_0 = \dot{z}_0 = 0$) are supplied in Table **3** for the purpose of recreating the orbits with a standard explicit propagator. The control law for implementation with an explicit subroutine is supplied by Eq. (36), where the coefficients for each orbit are provided in Table **3**. The initial states are integrated forward over the interval $[0,T]$ using MATLAB's `ode45` and `ode113`. The propagator



(a) Initial Coarse Mesh



(b) Final Refined Mesh

Fig. (4). Five near-optimal pole-sitter orbits.

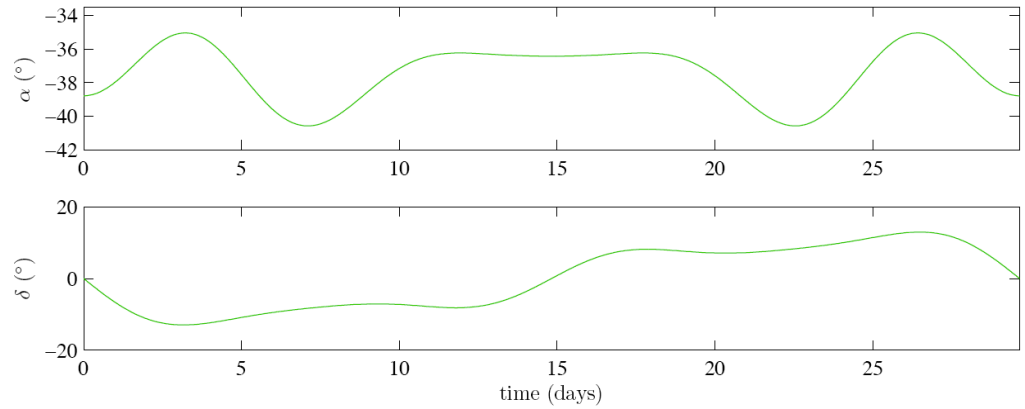


Fig. (5). Control law for the L_2, $\kappa = 0.58$ mm/s^2 orbit.

Table 4. Error with Standard Matlab Propagators

	$\kappa = 0.58$ mm/s^2		$\kappa = 1.70$ mm/s^2		
Type	L_1	L_2	L_1	L_2	Hover
$\|\mathbf{h}\|$ `ode45`	1.35e-06	8.27e-09	2.61e-09	1.34e-09	5.72e-11
$\|\mathbf{h}\|$ `ode113`	1.75e-06	3.59e-08	9.10e-10	5.62e-09	5.68e-11
Comparison of `ode45` and `ode113`	4.02e-07	4.42e-08	1.69e-09	4.28e-09	5.61e-13

`ode45` is a fourth-order Runge-Kutta method with fifth-order error control. The integrator `ode113` is a variable-order Adams-Bashforth-Moulton solver. Both propagators exploit adaptive step-sizing.

Let $\|\mathbf{h}\|$ represent the norm of the difference between the final and initial state along the trajectory. The quantity $\|\mathbf{h}\|$ provides insight into the errors associated with each method. Recall that for the Gauss-Lobatto collocation scheme, $\|\mathbf{h}\|$ is constrained to be less than 1e-12. The values of $\|\mathbf{h}\|$ for the `ode45` and `ode113` runs are available in Table **4**. For all simulations, the absolute and relative errors are set to 1e-12.

To compare `ode45` to `ode113`, the final row of Table **4** corresponds to the norm of the resulting vector from taking the difference of the final state vectors from `ode45` and `ode113`. The table demonstrates that the periodicity violation is consistent with the order of the error associated with each propagator. These errors suggest that the periodicity violations $\|\mathbf{h}\|$ recorded in Table **4** result primarily from dynamical instability (not numerical error associated with the Gauss-Lobatto scheme). Note also that, as expected, greater values of $\|\mathbf{h}\|$ are associated with larger values of $\lambda_{\max}$ in Table **3**. The orbit measure of instability stresses the importance of highly accurate integration subroutines. All the orbits are also propagated for multiple revolutions with `ode45` and `ode113`. The orbit near L_1 with k = 0.58 departs rapidly from the nominal orbit just after one revolution without additional control adjustments. The hover orbit stays close to the baseline periodic orbit the longest, departing after three revolutions. The other orbits depart after approximately two revolutions. The eigenvalues $\lambda_{\max}$ computed from collocation are also verified and match closely with the eigenvalues computed from explicitly propagating the state-transition matrix. If desired, the errors recorded in Table **4** are small enough for x_0, z_0, $\dot{y}_0$, and the Fourier coefficients to serve as initial guesses in a differential corrections procedure with explicit integration. However, given the instability of these orbits, it is likely that reducing the periodicity violation to zero with one integrator will still result in comparable errors to those recorded in Table **4** when integrated explicitly with the other integrator. Due to the dynamical sensitivity of the orbits, slight modifications of the angle histories may be sufficient to retain the trajectory near the baseline.

CONCLUSION

A seventh-degree Gauss-Lobatto scheme with mesh refinement for controlled periodic orbits is detailed. The scheme is successfully applied to compute lunar pole-sitter orbits using a solar sail. Comparison with previous efforts demonstrates that near-optimal elevation angle performance can be achieved by using a Fourier series control law with a small number of coefficients. The numerical behavior of the orbits is consistent with the predicted stability and the method produces results that are accurate to a level consistent with standard numerical integrators.

ACKNOWLEDGEMENTS

The authors wish to thank David Folta and Mark Beckman for assisting the research efforts at NASA Goddard Space Flight Center. This work was supported by the NASA Graduate Student Researchers Program (GSRP) fellowship under NASA Grant No. NNX07A017H and the Purdue Graduate Assistance in Areas of National Need (GAANN) fellowship. Portions of the work were completed at NASA Goddard and at Purdue University.

NOMENCLATURE

t = Time

$\mathbf{x}$ = State vector

$\mathbf{u}$ = Control vector

$\boldsymbol{\mu}$ = Problem dependent parameter vector

T = Orbit period

$\mathbf{g}$ = Path constraint vector

$\mathbf{h}$ = Periodicity constraint vector

τ = Normalized time

n = Number of mesh points (nodes)

Π = Set of mesh points

$\boldsymbol{\Delta}$ = Defect constraint vector

e = Relative integration error

ε = User-defined integration tolerance

Φ = State-transition matrix

λ = Eigenvalue of the state-transition matrix

U = Potential function

κ = Characteristic acceleration due to a solar sail

ω_s = Angular rate of Earth-moon synodic frame

φ = Spacecraft elevation angle

d = Spacecraft altitude

α = Solar sail out-of-plane angle

δ = Solar sail clock angle

REFERENCES

[1] E. Doedel, "Nonlinear numerics," *Int. J. Bifurcat. Chaos*, vol. 7, no. 9, pp. 2127-2143, 1997.

[2] J. Betts and P. Huffman, "Application of sparse nonlinear programming to trajectory optimization," *J. Guid. Control Dyn.*, vol. 15, no. 1, pp. 198-206, 1992.

[3] T. Ely, "Stable constellations of frozen elliptical inclined orbits," *J. Astronaut. Sci.*, vol. 53, no. 3, pp. 301-316, 2005.

[4] D. Grebow, M. Ozimek, and K. Howell, "Multibody orbit architectures for lunar south pole coverage," *J. Spacecr. Rockets*, vol. 45, no. 2, pp. 344-358, 2008.

[5] C. McInnes, *Solar Sailing: Technology, Dynamics and Mission Applications*, Springer-Verlag: Berlin, pp. 171-228, 1999.

[6] M. Ozimek, D. Grebow, and K. Howell, "Design of solar sail trajectories with applications to lunar south pole Coverage," *J. Guid. Control Dyn.*, vol. 32, no. 6, pp. 1884-1897, 2009.

[7] J. West, "The Lunar Polesitter," Paper AIAA-2008-7073, *AIAA/AAS Astrodynamics Specialist Conference*, Honolulu, Hawaii, August 18-21, 2008.

[8] G. Wawrzyniak and K. Howell, "The Solar Sail Lunar Relay Station: An Application of Solar Sails in the Earth-Moon System," *59th IAC Congress*, Paper IAC-08-C1.3.14, Glasgow, Scotland, September 29-October 3, 2008.

[9] D. Grebow, M. Ozimek, and K. Howell, "Design of Optimal Low-Thrust Lunar Pole-Sitter Missions," Paper No. AAS 09-148, *19th AAS/AIAA Space Flight Mechanics Meeting*, Savannah, Georgia, February 8-12, 2009.

[10] J. Simo and C. McInnes, "Asymptotic analysis of displaced lunar orbits," *J. Guid. Control Dyn.*, vol. 32, no. 5, pp. 1666-1670, 2009.

[11] A. Herman, "Improved Collocation Methods with Applications to Direct Trajectory Optimization," Ph.D. Dissertation, Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois, September 1995.

[12] B. De Boor, "Good Approximation by Splines with Variable Knots. II," *Conference on the Numerical Solution of Differential Equations, Lecture Notes in Mathematics*, Springer, New York, vol. 363, 1973.

[13] R. Russell and J. Christiansen, "Adaptive mesh selection strategies for solving boundary value problems," *SIAM J. Numer. Anal.*, vol. 15, no. 1, pp. 59-80, 1978.

Received: December 10, 2009 Revised: June 29, 2010 Accepted: July 9, 2010