



Comparison of the Legendre–Gauss pseudospectral and Hermite–Legendre–Gauss–Lobatto methods for low-thrust spacecraft trajectory optimization

Sanjeev Narayanaswamy¹ · Christopher J. Damaren¹

Received: 15 August 2019 / Revised: 20 November 2019 / Accepted: 2 December 2019
© Shanghai Jiao Tong University 2020

Abstract

Low-thrust spacecraft propulsion systems enable fuel-efficient trajectories through space but the resulting trajectory optimization problems can be challenging. Such problems often use direct collocation methods for transcribing the optimal control problem into a nonlinear programming problem. In this work, the Hermite–Legendre–Gauss–Lobatto (HLGL) and the Legendre–Gauss pseudospectral (PS) direct collocation methods have been compared for a minimum-time low-thrust Earth-to-Mars transfer problem. It is demonstrated that trajectories with different time domains can be effectively compared using points that correspond to the same normalized times. Various metrics that describe the computational cost as well as the errors with respect to a reference trajectory have been used. For the HLGL method, the number of subintervals (m) and the order of the interpolating polynomial (n), can be varied such that the number of nodes (N) stays constant. The distribution of the number of such (m, n) pairs is analysed using the concept of divisor functions from number theory and an algorithm for determining all the possible combinations for any N value is developed. In addition, a method is presented by which the Edelbaum trajectory is used to set bounds, scale the problem, and generate a suitable initial guess. The primary contribution of this paper involves a detailed comparison of the performance of all the HLGL pairs with the PS method for nodes in the range $N = 6$ to $N = 40$ for the minimum-time low-thrust Earth-to-Mars transfer.

Keywords Low thrust · Spacecraft trajectory optimization · Interplanetary transfer · Pseudospectral method · Divisor function

1 Introduction

Low-thrust electric propulsion for spacecraft provides a fuel-efficient mode of transportation in space. Because of the high specific impulses of these engines, they have been used for a variety of missions with the first interplanetary spacecraft to use this technology being Deep Space I [4]. The primary challenge with the optimization of low-thrust trajectories is due to the large search space caused by the continuous thrusting, the non-Keplerian nature of the orbits, and the long mission durations. As with high-thrust trajectory optimization, a suitable initial guess to an optimization algorithm is required in order to obtain a converged solution. An effective method of gen-

erating these initial guesses is through the use of shape-based methods that use functions with limited parameters to model a family of low-thrust trajectories [17,22,24]. To satisfy the equations of motion and the boundary conditions, different systems of equations corresponding to each shape-based method need to be solved. Petropoulos and Longuski [16] developed the four-parameter exponential sinusoid model to approximate near-planar low-thrust trajectories. Research by Wall and Conway [23,24] developed an inverse polynomial model that can be used for a three-dimensional rendezvous mission with up to a 15° inclination change with a specified starting and ending time. Another type of analytical solution is based on the work by Edelbaum [7], which provides expressions for a minimum-time coplanar circle-to-circle transfer with a low, constant, tangential thrust, for which the terminal phase cannot be specified [6]. Such methods are useful for rapidly searching large areas of the feasible set, which can reduce computational times for problems such as gravity assist optimization [13]. Once an estimate of an optimal trajectory has been found using analytical methods, it can be

✉ Sanjeev Narayanaswamy
sanjeev.narayanaswamy@mail.utoronto.ca

Christopher J. Damaren
damaren@utias.utoronto.ca

¹ University of Toronto Institute for Aerospace Studies,
4925 Dufferin Street, Toronto, ON M3H 5T6, Canada

supplied as an initial guess to a direct transcription method for finer optimization [16].

Direct transcription methods, also called direct collocation methods, convert the optimal control problem into a Non-linear Programming (NLP) problem through transcription onto a discrete set of points in the time domain. Compared to the calculus of variations-based indirect methods, direct methods do not require the derivation of the first-order necessary conditions and are less sensitive to the initial guess [3]. These methods can be further divided into global collocation methods and local collocation methods. Global collocation methods use an approximating function over the entire time interval. An area of active research involves a class of methods known as pseudospectral methods, which have been generally used as global methods [20]. These methods use orthogonal basis polynomials to approximate the state and control and use orthogonal collocation points to achieve accurate quadrature approximations. For smooth problems, pseudospectral methods have the property of *spectral*, or exponential, convergence [21]. Garg et al. [9] have compared the Legendre–Gauss–Lobatto (LGL), Legendre–Gauss (LG), and Legendre–Gauss–Radau (LGR) variants of the pseudospectral method in depth and developed the transformations between the Karush–Kuhn–Tucker multipliers of the NLP problem and the costates of the continuous problem. Local collocation methods break the time interval into segments and within each of these segments, piecewise functions are used to approximate the state and control histories. A well-known method is the Hermite–Simpson method where within each segment, cubic polynomials are used to approximate the state and linear functions are used to approximate the control. Herman and Conway [12] have compared the performance of direct collocation methods that use higher order Gauss–Lobatto quadrature rules. Williams developed a framework known as the Hermite–Legendre–Gauss–Lobatto (HLGL) method for higher odd-ordered Gauss–Lobatto rules [27]. This method avoids the detailed analytical derivation of the constraints needed in the work by Herman and Conway [12] and takes a generalized matrix based approach.

Numerical comparisons of the performance of different direct methods with respect to the number of time discretization nodes, N , is an important research topic. In fact, as expressed by Shirazi et al. [22] in their recent survey paper, there is no widely accepted way to do comparisons of spacecraft trajectory optimization methods. Williams conducted [26,27] a comparison of various direct transcription methods using five different values of N and later using eight different values of N for the HLGL method. Wang et al. [25] compared the LGL, LG, and LGR pseudospectral methods for a minimum-fuel Earth-to-Mars rendezvous mission but restricted their analysis to $N = 42$. García-Heras et al. [8] compared the Hermite–Simpson, the 5th-order HLGL, the Chebyshev–Gauss–Lobatto, and the LGL methods for a

minimum-fuel air traffic management problem but not all of the methods were compared for the same N values.

It is apparent that there is limited research into a comparison of direct methods for a fine mesh of N values with a thorough investigation, for local methods, of all the combinations of the number of subintervals and the order of the interpolating polynomial for a given N value. As a complement to the existing research, and to address this research gap, a detailed comparison of the local Hermite–Legendre–Gauss–Lobatto (HLGL) and the global Legendre–Gauss pseudospectral (PS) direct methods for fixed problem sizes in the range $N = 6$ to $N = 40$ has been performed here for a minimum-time interplanetary spacecraft trajectory optimization problem.

2 Methodology

A formulation of a general continuous optimal control problem stated in the Bolza form [5], is as follows:

$$\begin{aligned} \text{Minimize } J &= \phi(\mathbf{x}_f, t_f) + \int_{t_0}^{t_f} \mathcal{L}(\mathbf{x}(t), \mathbf{u}(t), t) dt \\ \text{Subject To:} \\ \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \\ g_i(\mathbf{x}(t), \mathbf{u}(t), t) &\leq 0, \quad i = 1, 2, \dots, n_g \\ h_i(\mathbf{x}(t), \mathbf{u}(t), t) &= 0, \quad i = 1, 2, \dots, n_h \end{aligned} \quad (1)$$

where $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ is the state, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ is the control, \mathbf{x}_f is the terminal state, t_f is the terminal time, and J is the objective function. The functions g_i and h_i correspond to the inequality and equality constraints, respectively. It is important to note that the general formulation in Eqs. (1) is suitable for both minimum-time as well as minimum-fuel optimizations, due to the presence of t_f and $\mathbf{u}(t)$ in the definition of J .

2.1 Edelbaum transfer

Edelbaum published his analytic equations for a low-thrust minimum-time transfer between two inclined circular orbits of radii r_0 and r_f based on Gauss' variational equations [7]. For the case of a coplanar circle-to-circle transfer, these equations simplify [6] to:

$$\begin{aligned} r(t) &= \frac{\mu}{\left(\sqrt{\frac{\mu}{r_0}} - ft\right)^2} \\ \theta(t) &= \frac{\mu}{4f} \left(\frac{1}{r_0^2} - \frac{1}{r^2(t)} \right) \end{aligned}$$

Table 1 Summary of the Legendre–Gauss pseudospectral and Hermite–Legendre–Gauss–Lobatto methods

	State interpolation strategy	State interpolating polynomial	Enforcement of dynamics	Control interpolation	Quadrature
PS	Lagrange interpolation over the entire time domain using the state values at the set of $N + 1$ interpolating points $\tau_{0:N}$	A single N th-order global polynomial is used to approximate the state over the entire time domain	The \mathbf{h}_{PS} equations constrain the derivatives of the interpolating polynomial to match the vector field at the set of N collocation points $\tau_{1:N}$	Lagrange interpolation over the entire time domain using the control values at $\tau_{1:N}$	LG quadrature performed over the entire time domain
HLGL	Hermite interpolation over the i th subinterval using the state values and its derivatives at the set of $\frac{n+1}{2}$ nodes γ_i	A separate n th-order local polynomial is used to approximate the state for each of the m subintervals	The \mathbf{h}_{HLGL} equations constrain the derivatives of the interpolating polynomial over the i th subinterval to match the vector field at the set of $\frac{n-1}{2}$ collocation points ζ_i	Linear interpolation over the i th subinterval using the control values at γ_i	LGL quadrature performed cumulatively over each subinterval

$$\begin{aligned} \dot{r}(t) &= \frac{2\mu f}{\left(\sqrt{\frac{\mu}{r_0}} - ft\right)^3} \\ \dot{\theta}(t) &= \frac{\mu}{2f} \left(\frac{\dot{r}(t)}{r^3(t)}\right) \\ T_m &= \left(\frac{1}{f}\right) \left(\sqrt{\frac{\mu}{r_0}} - \sqrt{\frac{\mu}{r_f}}\right), \end{aligned} \tag{2}$$

where we assume a small, constant, tangential thrust (f), a small eccentricity throughout the transfer, and a long transfer time (T_m) relative to the orbital period [6,7]. The quantity μ is the standard gravitational parameter and θ is the true anomaly. Since we have constant thrust, this coplanar transfer is optimal with respect to both time and fuel.

The Edelbaum equations are not as flexible as some of the other analytical methods such as exponential sinusoids and inverse polynomials because they only depend on r_0 , r_f , and the thrust magnitude f . For this reason, they cannot be used to generate trajectories for rendezvous missions where the time of flight and terminal position are constraints. However, the real power of Edelbaum’s equations lies in the fact that they can be used as an initial guess for direct transcription methods to solve closely related minimum-time problems.

2.2 Legendre–Gauss pseudospectral method

The continuous optimal control problem in Eqs. (1) can be transformed into an NLP problem using the Legendre–Gauss pseudospectral method [9]:

$$\begin{aligned} \text{Minimize } J &= \phi(\mathbf{X}_{N+1}, t_f) \\ &+ \left(\frac{t_f - t_0}{2}\right) \sum_{i=1}^N w_i \mathcal{L}(\mathbf{X}_i, \mathbf{U}_i, t(\tau_i)) \end{aligned}$$

Subject To:

$$g_i(\mathbf{X}_{1:N+1}, \mathbf{U}_{1:N+1}, t(\tau_{1:N+1})) \leq 0, \quad i = 1, 2, \dots, n_g$$

$$\begin{aligned} h_i(\mathbf{X}_{1:N+1}, \mathbf{U}_{1:N+1}, t(\tau_{1:N+1})) &= 0, \quad i = 1, 2, \dots, n_h \\ \mathbf{h}_{PS} := \mathbf{D}\mathbf{X}_{0:N} - \left(\frac{t_f - t_0}{2}\right) \mathbf{F}(\mathbf{X}_{1:N}, \mathbf{U}_{1:N}, t(\tau_{1:N})) &= \mathbf{0}, \end{aligned} \tag{3}$$

where the problem has been discretized at the set of N Legendre–Gauss collocation points $\tau_{1:N}$. The w_i are the Legendre–Gauss quadrature weights. Lagrange basis polynomials are used to form a global N th-order polynomial to interpolate the state and control. The quantities $\mathbf{X}_{1:N}$ and $\mathbf{U}_{1:N}$ are the matrices composed of rows of the state and control vectors, respectively, at $\tau_{1:N}$. \mathbf{F} is the state derivative matrix at $\tau_{1:N}$ evaluated using the equations of motion, and \mathbf{D} is the differentiation matrix such that $\mathbf{D}\mathbf{X}_{0:N}$ gives the derivative of the interpolating polynomial for the state at each of the collocation points. It should be noted that an additional Nn_x equality constraints, which are due to the PS method, have been added to the problem. These are denoted by \mathbf{h}_{PS} and are enforced at the collocation points $\tau_{1:N}$. The other equality and inequality constraints, g_i and h_i , are enforced at $\tau_{1:N+1}$ with the expectation that a high enough N is likely to cause the entire interpolated solution to satisfy the constraints. For a summary of this method in relation to the HLGL method, please see Table 1. For a detailed derivation of the Legendre–Gauss pseudospectral method, please see Appendix A.

2.3 Hermite–Legendre–Gauss–Lobatto method

The continuous optimal control problem in Eqs. (1) can be transformed into an NLP problem using the Hermite–Legendre–Gauss–Lobatto method [27]:

$$\begin{aligned}
& \text{Minimize } J = \phi(\mathbf{x}_m(1), t_f) \\
& + \sum_{i=1}^m \frac{h_i}{2} \sum_{j=1}^n w_j \mathcal{L}(\mathbf{x}_i(\xi_j), \mathbf{u}_i(\xi_j), t_i(\xi_j)) \\
& \text{Subject To:} \\
& g_i(\mathbf{x}(\boldsymbol{\gamma}), \mathbf{u}(\boldsymbol{\gamma}), t(\boldsymbol{\gamma})) \leq 0, \quad i = 1, 2, \dots, n_g \\
& h_i(\mathbf{x}(\boldsymbol{\gamma}), \mathbf{u}(\boldsymbol{\gamma}), t(\boldsymbol{\gamma})) = 0, \quad i = 1, 2, \dots, n_h \\
& \mathbf{h}_{\text{HLGL}} := \boldsymbol{\Phi}_{d_i}^T \mathbf{b}_i \\
& -\frac{h_i}{2} \mathbf{F}(\boldsymbol{\Phi}_i^T \mathbf{b}_i, \mathbf{u}_i(\zeta_i), t_i(\zeta_i)) = \mathbf{0}, \quad i = 1, 2, \dots, m
\end{aligned} \tag{4}$$

where the time domain has been split into m subintervals. Within each subinterval, Hermite interpolation is used to form an n th-order polynomial to approximate the state while linear interpolation is used to approximate the control. For subinterval i , the n LGL points ξ are divided into $\frac{n+1}{2}$ nodes denoted by $\boldsymbol{\gamma}$ and $\frac{n-1}{2}$ collocation points denoted by ζ . In total, across all subintervals, this problem has N nodes. The term w_j denotes the Legendre–Gauss–Lobatto quadrature weights within each subinterval. The quantities \mathbf{x}_i and \mathbf{u}_i are the state and control approximations at the i th subinterval while \mathbf{x} and \mathbf{u} represent the approximations of the full state and control trajectories, respectively, in $[t_0, t_f]$. The quantity $\boldsymbol{\Phi}_{d_i}^T \mathbf{b}_i$ represents the derivative of the approximating polynomial for the state evaluated at ζ_i while $\boldsymbol{\Phi}_i^T \mathbf{b}_i$ just represents the polynomial evaluated at those points. An additional $mn_x(\frac{n-1}{2})$ equality constraints, due to the HLGL method, have been added to the problem. The \mathbf{h}_{HLGL} constraints are enforced at the collocation points, whereas the other constraints, g_i and h_i , are enforced at the nodal points. For a summary of this method in relation to the PS method, please see Table 1. For a detailed derivation of the Hermite–Legendre–Gauss–Lobatto method, please see Appendix B.

3 Problem formulation: Earth-to-Mars orbital transfer

The dimensional problem statement is formulated with the state $\mathbf{x} = [r \ \theta \ \dot{r} \ \dot{\theta}]$ and the control $\mathbf{u} = [u \ \alpha]$, where u is the thrust magnitude and α is the thrust angle. This can be written as:

$$\begin{aligned}
& \text{Minimize } J = t_f \\
& \text{Subject To:} \\
& \dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{r} & \dot{\theta} & \left(r\dot{\theta}^2 - \frac{\mu_{\text{Sun}}}{r^2} + u \sin \alpha\right) \\ \left(\frac{-2\dot{r}\dot{\theta}}{r} + \frac{u \cos \alpha}{r}\right) \end{bmatrix} \\
& \mathbf{x}_{lb} \leq \mathbf{x} \leq \mathbf{x}_{ub}
\end{aligned}$$

$$\begin{aligned}
& \mathbf{x}_{lb} = [r_{\text{Earth}} \ 0 \ -100 \max(|\dot{r}_{\text{Edel}}|) \ 0] \\
& \mathbf{x}_{ub} = \begin{bmatrix} r_{\text{Mars}} & 2(2\pi) \left[\frac{\theta_{f,\text{Edel}}}{2\pi}\right] & 100 \max(|\dot{r}_{\text{Edel}}|) \\ & 100 \max(|\dot{\theta}_{\text{Edel}}|) \\ & [0 \ -\pi] \leq \mathbf{u} \leq [3 \times 10^{-4} \ \pi] \\ & 1 \leq t_f \leq 10t_{f,\text{Edel}} \\ & \mathbf{x}(0) = \begin{bmatrix} r_{\text{Earth}} & 0 & 0 & \sqrt{\frac{\mu_{\text{Sun}}}{r_{\text{Earth}}^3}} \end{bmatrix} \\ & \mathbf{x}(t_f) = \begin{bmatrix} r_{\text{Mars}} & \frac{3\pi}{2} + k2\pi & 0 & \sqrt{\frac{\mu_{\text{Sun}}}{r_{\text{Mars}}^3}} \end{bmatrix}, \end{bmatrix} \tag{5}
\end{aligned}$$

where the units for \mathbf{x} are [m rad m/s rad/s], the units for \mathbf{u} are [N/kg rad], and the unit of t is [s]. The quantities with subscript $(\)_{\text{Edel}}$ correspond to the Edelbaum transfer. A linear scaling method is chosen where each of the variables and their bounds are scaled by the corresponding upper bounds so that the problem is non-dimensionalized. In addition, the decision vector lies in $[-1, 1]$ because the upper bound of each variable has been chosen to be larger than or equal to the magnitude of the lower bound.

4 PS and HLGL comparison with constant N

The Legendre–Gauss pseudospectral method and the Hermite–Legendre–Gauss–Lobatto method have been compared for an Earth-to-Mars minimum-time transfer problem for the range $N = [6, 40]$. In addition, the $N = 64$ PS method and the $N = 64$ HLGL method with $n = 3$ were analysed to investigate the behaviour at a higher value of N . The $n = 3$ HLGL method, using cubic polynomials, is also known as the Hermite–Simpson (HS) method. The reference trajectory for calculating error metrics for trajectories in the range $N = [6, 40]$ was taken to be the solution of the $N = 64$ PS method as it had a lower objective function value than the $N = 64$ HS method. These methods have been implemented and compared in Matlab using SNOPT [10] as the NLP solver. The Edelbaum trajectory, shown in Fig. 1, was used as an initial guess for the $N = 16$ HS method. To make a fair comparison, the result of this optimization was then used as an initial guess for all runs.

4.1 Optimized Earth-to-Mars trajectory

For $N = 64$, both the HS method and the PS method result in trajectories that are close to each other, as can be seen in Fig. 2. This suggests that close local minima have been found. The optimal transfer time for the PS method was found to be approximately 354 days while for the HS method, it was

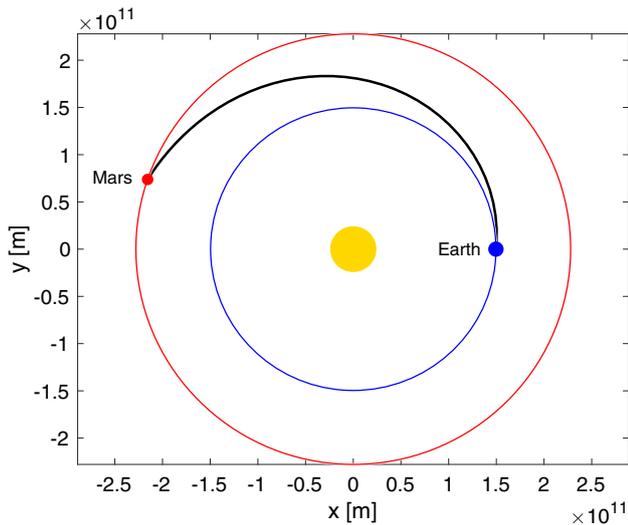


Fig. 1 Earth-to-Mars Edelbaum transfer

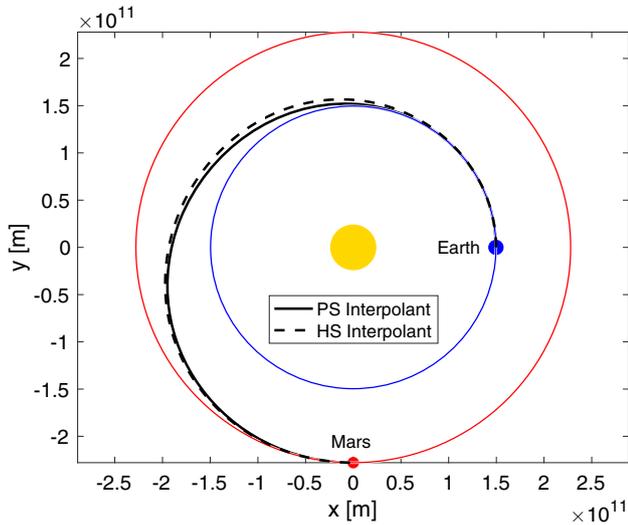


Fig. 2 Earth-to-Mars optimized trajectory— $N = 64$

found to be about 365 days. If we assume a spacecraft dry terminal mass of $m_{dry} = 300$ kg and an engine with a specific impulse of $I_{sp} = 3550$ s, it was found that the PS trajectory expended more fuel in trying to minimize the objective as it consumed $m_{fuel} = 90$ kg while the HS trajectory only consumed $m_{fuel} = 76$ kg. In addition, it was found that the arc length of the PS trajectory was 8.41×10^{11} m while the HS trajectory had a higher arc length of 8.52×10^{11} m. This agrees with the lower transfer duration for the PS trajectory.

4.2 HLGL (m, n) pairs for constant N

To compare the performance of the PS and HLGL methods, it is desirable to keep the problem size approximately constant. This can be done by comparing runs that have the same number of nodes. In this work, we define the problem size

to be the length of the NLP vector passed to the NLP solver. The problem size of the global PS method is directly controlled by N , the number of nodes. In contrast, the problem size of the HLGL method is indirectly controlled by m , the number of subintervals, and n , the order of the interpolating polynomial within each subinterval. It is important to note that for a given N value, the size of the NLP vector for the Legendre–Gauss PS method is given by $N(n_x + n_u) + 1$ while the size of the NLP vector for the HLGL method is given by $N(n_x + n_u) - n_x + 1$. The “+1” terms correspond to adding the transfer time to the set of optimization variables. The size of the NLP vector for the HLGL method is lower by n_x since we omit the initial state from the NLP vector as it is a known quantity. It is, however, augmented internally to evaluate constraints and perform interpolation. For large enough values of N , it is reasoned that the difference of n_x will be negligible when controlling for the problem size. The relation between N , m , and n can be expressed as

$$N = \frac{n + 1}{2} + \left(\frac{n - 1}{2}\right)(m - 1),$$

$$\{n \in 2\mathbb{Z} + 1 \mid 3 \leq n \leq 2N - 1\}, \tag{6}$$

where the first term represents the number of nodes in the first subinterval and the second term accounts for the $(n - 1)/2$ nodes in each of the remaining $(m - 1)$ subintervals. The restrictions on n are necessary to make the number of constraints per interval, $n_x(n - 1)/2$, a positive integer. It is useful to isolate for m in Eq. (6) as a function of N and n and this results in the simple relation

$$m = \frac{2(N - 1)}{n - 1}, \quad \{n \in 2\mathbb{Z} + 1 \mid 3 \leq n \leq 2N - 1\}. \tag{7}$$

For a given value of N , Eq. (7) can be used to generate possible pairs of (m, n) values by iterating through various values of n . To search for valid integer values of m , it is necessary to test positive odd values of n between the lower bound of $n = 3$ and the upper bound of $n = 2N - 1$. A full listing of the possible (m, n) pairs in the range $N = [6, 40]$ is given in Table 2.

At this point, a connection to number theory can be made here using Ramanujan’s work on divisor functions [19] and Highly Composite Numbers (HCN) [2,15,18]. The divisor function

$$\sigma_k(n) = \sum_{d|n} d^k \tag{8}$$

sums over the k th powers of all d such that d is a divisor of n , which is indicated by $d|n$. If $k = 0$, then

$$\sigma_0(n) = \sum_{d|n} d^0 \tag{9}$$

Table 2 HLGL (m, n) pairs and case numbers—bolded case numbers indicate convergence

Case number	Nodes N	HLGL pair (m, n)	Case number	Nodes N	HLGL pair (m, n)	Case number	Nodes N	HLGL Pair (m, n)	Case number	Nodes N	HLGL pair (m, n)
1	6	(5, 3)	37	16	(1, 31)	73	25	(4, 13)	109	33	(4, 17)
2	6	(1, 11)	38	17	(16, 3)	74	25	(3, 17)	110	33	(2, 33)
3	7	(6, 3)	39	17	(8, 5)	75	25	(2, 25)	111	33	(1, 65)
4	7	(3, 5)	40	17	(4, 9)	76	25	(1, 49)	112	34	(33, 3)
5	7	(2, 7)	41	17	(2, 17)	77	26	(25, 3)	113	34	(11, 7)
6	7	(1, 13)	42	17	(1, 33)	78	26	(5, 11)	114	34	(3, 23)
7	8	(7, 3)	43	18	(17, 3)	79	26	(1, 51)	115	34	(1, 67)
8	8	(1, 15)	44	18	(1, 35)	80	27	(26, 3)	116	35	(34, 3)
9	9	(8, 3)	45	19	(18, 3)	81	27	(13, 5)	117	35	(17, 5)
10	9	(4, 5)	46	19	(9, 5)	82	27	(2, 27)	118	35	(2, 35)
11	9	(2, 9)	47	19	(6, 7)	83	27	(1, 53)	119	35	(1, 69)
12	9	(1, 17)	48	19	(3, 13)	84	28	(27, 3)	120	36	(35, 3)
13	10	(9, 3)	49	19	(2, 19)	85	28	(9, 7)	121	36	(7, 11)
14	10	(3, 7)	50	19	(1, 37)	86	28	(3, 19)	122	36	(5, 15)
15	10	(1, 19)	51	20	(19, 3)	87	28	(1, 55)	123	36	(1, 71)
16	11	(10, 3)	52	20	(1, 39)	88	29	(28, 3)	124	37	(36, 3)
17	11	(5, 5)	53	21	(20, 3)	89	29	(14, 5)	125	37	(18, 5)
18	11	(2, 11)	54	21	(10, 5)	90	29	(7, 9)	126	37	(12, 7)
19	11	(1, 21)	55	21	(5, 9)	91	29	(4, 15)	127	37	(9, 9)
20	12	(11, 3)	56	21	(4, 11)	92	29	(2, 29)	128	37	(6, 13)
21	12	(1, 23)	57	21	(2, 21)	93	29	(1, 57)	129	37	(4, 19)
22	13	(12, 3)	58	21	(1, 41)	94	30	(29, 3)	130	37	(3, 25)
23	13	(6, 5)	59	22	(21, 3)	95	30	(1, 59)	131	37	(2, 37)
24	13	(4, 7)	60	22	(7, 7)	96	31	(30, 3)	132	37	(1, 73)
25	13	(3, 9)	61	22	(3, 15)	97	31	(15, 5)	133	38	(37, 3)
26	13	(2, 13)	62	22	(1, 43)	98	31	(10, 7)	134	38	(1, 75)
27	13	(1, 25)	63	23	(22, 3)	99	31	(6, 11)	135	39	(38, 3)
28	14	(13, 3)	64	23	(11, 5)	100	31	(5, 13)	136	39	(19, 5)
29	14	(1, 27)	65	23	(2, 23)	101	31	(3, 21)	137	39	(2, 39)
30	15	(14, 3)	66	23	(1, 45)	102	31	(2, 31)	138	39	(1, 77)
31	15	(7, 5)	67	24	(23, 3)	103	31	(1, 61)	139	40	(39, 3)
32	15	(2, 15)	68	24	(1, 47)	104	32	(31, 3)	140	40	(13, 7)
33	15	(1, 29)	69	25	(24, 3)	105	32	(1, 63)	141	40	(3, 27)
34	16	(15, 3)	70	25	(12, 5)	106	33	(32, 3)	142	40	(1, 79)
35	16	(5, 7)	71	25	(8, 7)	107	33	(16, 5)			
36	16	(3, 11)	72	25	(6, 9)	108	33	(8, 9)			

counts the number of divisors for a given n and is a well-known function in number theory [11]. As the values of $\sigma_0(n)$ are tabulated in the literature [1], if we connect Eq. (7) to this function, then it will be possible to look up the number of valid (m, n) pairs for a given N . Let us, therefore, rewrite Eq. (7) as:

$$m = \frac{2(N-1)}{n-1}, \quad \{n \in 2\mathbb{Z} + 1 \mid 3 \leq n \leq 2N-1\} \quad (10)$$

$$= \frac{N-1}{\left(\frac{n-1}{2}\right)}, \quad \{n \in 2\mathbb{Z} + 1 \mid 3 \leq n \leq 2N-1\} \quad (11)$$

$$= \frac{N-1}{k}, \quad \{k \in \mathbb{Z} \mid 1 \leq k \leq N-1\}, \quad (12)$$

where $k = \frac{n-1}{2}$. In this form, since any integral value of k from 1 to $N-1$ that is a divisor of $N-1$ corresponds to an HLGL (m, n) pair, we see that the number of HLGL pairs corresponds to the number of divisors of $N-1$. To formalize

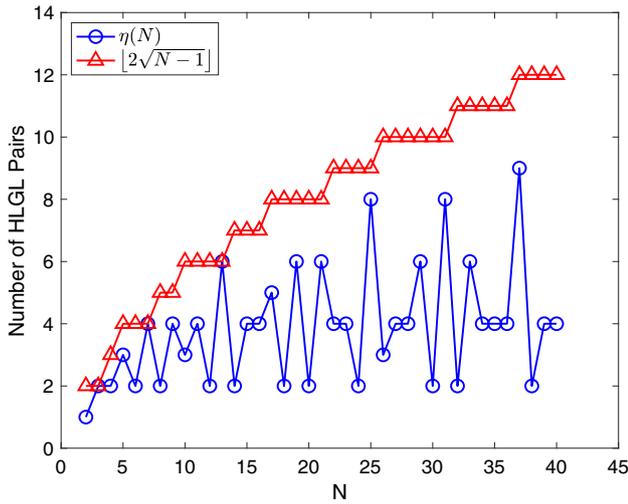


Fig. 3 Number of HLGL pairs and its upper bound vs. N

this, let us denote the number of valid HLGL pairs for a given N by $\eta(N)$. Then we can write

$$\eta(N) = \sigma_0(N - 1). \tag{13}$$

With Eq. (13) we are able to use the values of the divisor function to determine the number of possible HLGL pairs for a given N . Additionally, we know that for a natural number p , we can find an upper bound on the number of divisors using $\sigma_0(p) < 2\sqrt{p}$ and knowing that the number of divisors must be an integral value, we can write $\eta(N) \leq \lfloor 2\sqrt{N-1} \rfloor$. A plot that shows $\eta(N)$ for the range $N = [2, 40]$ along with the upper bound of $\lfloor 2\sqrt{N-1} \rfloor$ is given in Fig. 3.

Through this process, it can be seen that $\eta(N)$ is non-monotonic and that there are some values of N that have more valid (m, n) pairs than others. Odd values of N tend to result in larger values of $\eta(N)$ since this causes $N - 1$ in Eq. (13) to be even, and even numbers tend to have more divisors than odd numbers. In 1915, Ramanujan [18] introduced the concept of a Highly Composite Number (HCN) to describe a number q such that $\sigma_0(p) < \sigma_0(q), \forall p < q$. Looking at the spikes in Fig. 3, we see that a maximum value of $\eta(N) = 9$ occurs at $N = 37$, which is expected since $N - 1 = 36$ is the largest HCN less than 48.

4.3 Calculating errors between spacecraft trajectories with different time domains

For minimum-time spacecraft trajectory optimization problems, since the terminal time is free, the time domains for trajectories resulting from different optimization runs may not be equal. This is certainly the case with using the $N = 64$ PS trajectory as the reference trajectory, since the transfer duration found by this method is generally smaller than the

transfer durations found using the other cases that are being compared. To calculate the Root Mean Square (RMS) errors, corresponding points in time need to be chosen between both trajectories. If we use dimensional time to find the corresponding points, then the reference $N = 64$ PS trajectory needs to be extrapolated and will diverge quite quickly as this involves a 64th-order polynomial.

The solution used in this paper is to calculate RMS errors by comparing points that correspond not to the same points in dimensional time but to the same points in normalized time $s \in [-1, 1]$. For example, the beginning ($s = -1$), middle ($s = 0$), and end ($s = 1$) of each trajectory can be compared with the corresponding points of the reference trajectory. This idea eschews extrapolation and allows the comparison of two spacecraft trajectories with different time domains. As required, the RMS errors found using this method will approach zero as these trajectories converge.

Specifically, in this work, we calculate the RMS errors of trajectories found using either the PS or HLGL method for the range $N = [6, 40]$, with respect to the $N = 64$ PS method reference trajectory. First, a set S of $K_{RMS} = 20000$ linearly spaced points in $[-1, 1]$ is chosen, including the endpoints. Then the RMS error for the quantity p , which can be a position, velocity, or thrust vector represented in Cartesian coordinates, is calculated as

$$E_{RMS,p} = \sqrt{\sum_{s_i \in S} \frac{|p(s_i) - p_{ref}(s_i)|^2}{K_{RMS}}}, \tag{14}$$

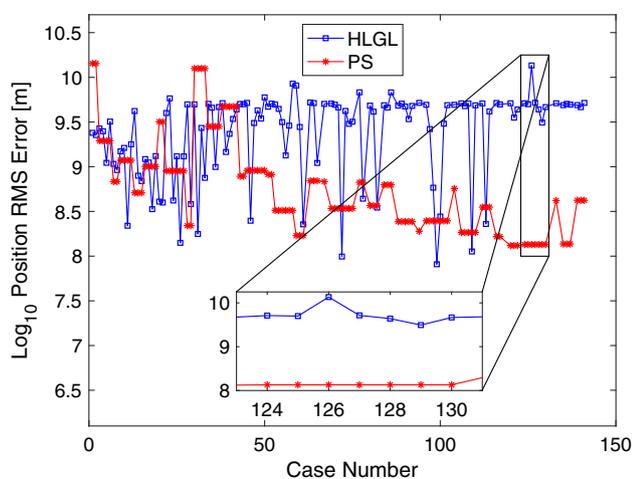
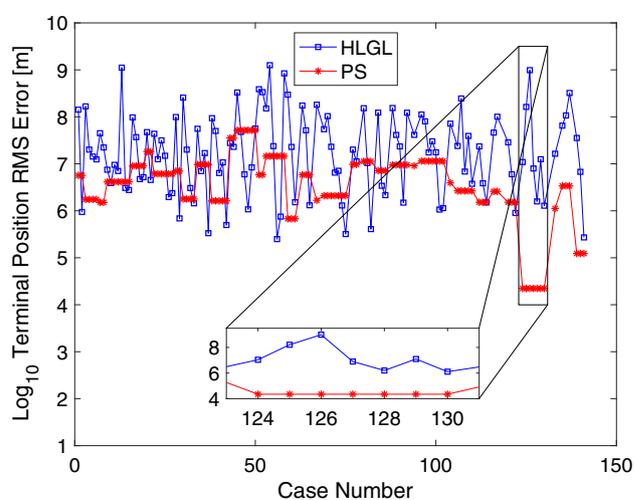
where $|\cdot|$ represents the Euclidean norm operator. The quantities $p(s_i)$ and $p_{ref}(s_i)$ represent the values of p corresponding to s_i at the comparison and reference trajectories, respectively. To determine $p(s_i)$, either Lagrange interpolation or Hermite interpolation needs to be applied depending on whether the PS or HLGL method was used to find the comparison trajectory. We note that Lagrange interpolation is used to determine $p_{ref}(s_i)$ since we are using the $N = 64$ PS trajectory as the reference. For the terminal position error, only the points corresponding to $s = 1$ are used for the calculation.

4.4 Comparison results

The PS method has been compared with the HLGL method for the range $N = [6, 40]$ and each of the HLGL pairs is given a unique case number. If an HLGL run converged, the corresponding case number will be bolded in Table 2 and the data will be included in the figures below. A total of 157 converged trajectories, including the PS runs, have been generated for this analysis. The results for $N = 37$, which corresponds to the largest number of HLGL (m, n) pairs in this range, with $\eta(37) = 9$, are given in Table 3. The

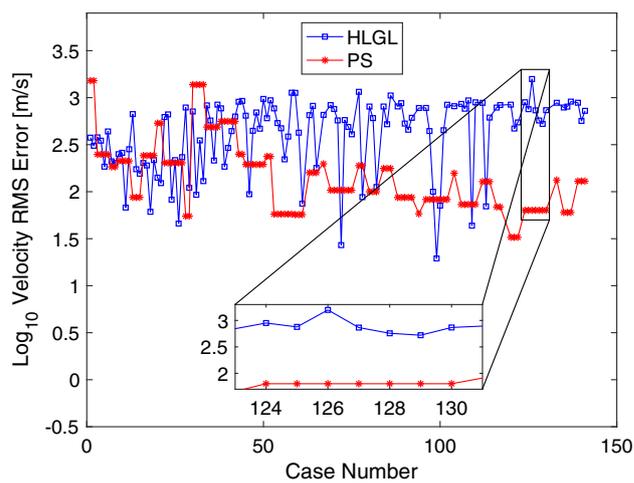
Table 3 Comparison metrics for HLGL (m, n) pairs with the LG pseudospectral method for $N = 37$ —bolded case numbers indicate convergence

Case number	Nodes N	PS or HLGL pair (m, n)	Transfer duration (s)	Position RMS error (m)	Terminal position RMS error (m)	Velocity RMS error (m/s)	Thrust RMS error (N/kg)	Processor time (s)
–	37	PS	3.07×10^7	1.35×10^8	2.23×10^4	6.37×10^1	1.03×10^{-4}	610
124	37	(36, 3)	3.17×10^7	5.13×10^9	1.09×10^7	8.93×10^2	2.45×10^{-4}	43.9
125	37	(18, 5)	3.13×10^7	5.00×10^9	1.63×10^8	7.54×10^2	2.20×10^{-4}	42.1
126	37	(12, 7)	3.32×10^7	1.36×10^{10}	9.87×10^8	1.58×10^3	2.60×10^{-4}	295
127	37	(9, 9)	3.10×10^7	5.19×10^9	7.96×10^6	7.33×10^2	1.74×10^{-4}	130
128	37	(6, 13)	3.10×10^7	4.38×10^9	1.58×10^6	5.72×10^2	1.36×10^{-4}	107
129	37	(4, 19)	3.09×10^7	3.12×10^9	1.25×10^7	5.26×10^2	1.40×10^{-4}	80.3
130	37	(3, 25)	3.13×10^7	4.65×10^9	1.28×10^6	7.36×10^2	2.03×10^{-4}	78.4
131	37	(2, 37)	–	–	–	–	–	–
132	37	(1, 73)	–	–	–	–	–	–

**Fig. 4** Log_{10} of position RMS error for HLGL cases vs. PS**Fig. 5** Log_{10} of terminal position RMS error for HLGL cases vs. PS

complete comparison results for the range $N = [6, 40]$ are given in Table 4 in Appendix C.

The plots in Figs. 4, 5, 6, 7, 8, and 9 show the behaviour of all the HLGL pairs in comparison to the PS method with insets showing the behaviour for $N = 37$. The PS metrics for a particular value of N are repeated such that each of the converged HLGL cases corresponding to that N value has a PS metric for comparison. It is important to note that these metrics are represented in dimensional units. From these plots, one can immediately observe that although the relationship between the metrics and the case number is highly variable and non-monotonic, certain general trends can be found. First, it is seen from Fig. 9 that the processor times for a given N value are in general higher for the PS method than for the corresponding HLGL cases. From Fig. 8, it is seen that the transfer duration, which represents the objective function, tends to be lower for the PS method than for the HLGL method. From Case 3 to Case 15, the RMS errors for

**Fig. 6** Log_{10} of velocity RMS error for HLGL cases vs. PS

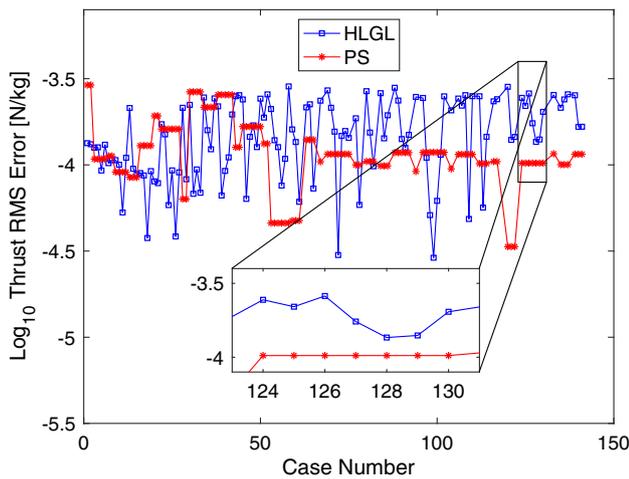


Fig. 7 Log₁₀ of thrust RMS error for HLGL cases vs. PS

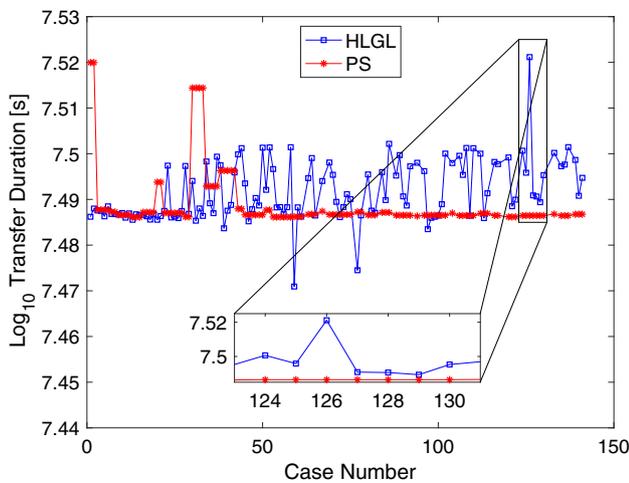


Fig. 8 Log₁₀ of transfer duration for HLGL cases vs. PS

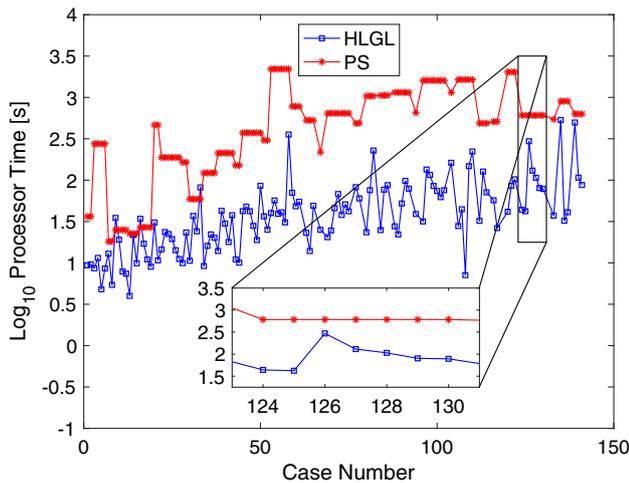


Fig. 9 Log₁₀ of processor time for HLGL cases vs. PS

both the HLGL method and the PS method are similar. For subsequent case numbers, the HLGL errors tend to be higher than the corresponding PS errors and stay close to a certain level as N increases while the PS errors tend to decrease with increasing N . This may occur because as the number of nodes is increased, the HLGL trajectories approach the $N = 64$ HS trajectory and the PS trajectories approach the $N = 64$ PS trajectory, which is used as the reference. However, like Case 72, there are exceptions where certain HLGL cases have errors that are below the corresponding PS errors. Another exception occurs for Case 59, which corresponds to $(m, n) = (21, 3)$ where the transfer duration is only 342 days in contrast to the reference value of 354 days for the $N = 64$ PS method. Perhaps if we use the solution of this run as an initial guess, lower transfer durations would result for all the HLGL pairs. From Table 2, it is seen that problems with convergence for HLGL runs occur for $n \geq 35$. All the PS runs in the range $N = [6, 40]$ have converged but HLGL runs with $n \geq 43$ have not converged. An important analysis is how for a given value of N , the choice of m and n affects the performance of the HLGL method. From the insets, it is seen that Case 126, which corresponds to $(m, n) = (12, 7)$ exhibits higher errors when compared to Case 127 which corresponds to $(m, n) = (9, 9)$, even though they both correspond to $N = 37$. For the specific example of velocity RMS error, looking at Table 3, it is apparent that while Case 126 has an error of 1.58×10^3 (m/s), Case 127 only has an error of 7.33×10^2 (m/s).

5 Conclusion

A minimum-time low-thrust Earth-to-Mars trajectory with a terminal phase constraint has been solved using both the Legendre–Gauss pseudospectral and the Hermite–Legendre–Gauss–Lobatto methods. The phase-free Edelbaum transfer is used to set bounds, scale the problem, and generate a suitable initial guess. A general algorithm for finding the HLGL (m, n) pairs for a given value of N has been formulated. An expression for the number of possible HLGL pairs has been represented using the concept of a divisor function and an upper bound has been derived. Ramanujan’s work on Highly Composite Numbers is used to describe values of N for which many HLGL pairs exist. It is also shown that spacecraft trajectories with different time domains can be compared by calculating errors between points that correspond to the same normalized times. A detailed comparison of the performance of the PS and HLGL methods has been completed for fixed problem sizes in the range $N = [6, 40]$, which addresses a research gap. This reveals significant performance differences between the PS and HLGL methods with respect to N , m , and n . Future adaptive versions of the HLGL method, such as the recent work by Lei et al. [14], may benefit by exploit-

ing those HLGL pairs which offer the best performance for a given problem size. The approach of the present paper can be used as a basis for a systematic comparison of two general local and global direct collocation methods.

Appendix A Legendre–Gauss pseudospectral method derivation

The pseudospectral method is typically formulated as a global method. We will develop and summarize the equations for the N th-order global Legendre–Gauss pseudospectral method using the N th-order LG points as collocation points based on the work by Garg et al. [9].

First we transform the problem into $\tau = [-1, 1]$ from $t = [t_0, t_f]$ space using:

$$t = \frac{t_f - t_0}{2}\tau + \frac{t_f + t_0}{2}. \quad (15)$$

The point τ_0 corresponds to t_0 , $\tau_{1:N}$ represents the transformation of the set of N LG points in the open interval (t_0, t_f) , and the terminal point τ_{N+1} corresponds to t_f . The state and control are then approximated using Lagrange basis polynomials $L_i(\tau)$:

$$\mathbf{x}(\tau) = \sum_{i=0}^N \mathbf{X}_i L_i(\tau) \quad (16)$$

$$\mathbf{u}(\tau) = \sum_{i=1}^N \mathbf{U}_i L_i(\tau), \quad (17)$$

where \mathbf{X}_i and \mathbf{U}_i are row vectors corresponding to τ_i .

Differentiating and evaluating at the k th collocation point τ_k results in

$$\dot{\mathbf{x}}(\tau_k) = \sum_{i=0}^N \mathbf{X}_i \dot{L}_i(\tau_k) = \sum_{i=0}^N D_{ki} \mathbf{X}_i(\tau_k). \quad (18)$$

We note that \mathbf{X}_0 is used for interpolation but not for collocation. Now we can impose constraints that the interpolated values of the state derivative at the collocation points $\tau_{1:N}$ match the exact values of the state derivative at $\tau_{1:N}$, expressed using \mathbf{F} :

$$\mathbf{D}\mathbf{X}_{0:N} = \left(\frac{t_f - t_0}{2} \right) \mathbf{F}(\mathbf{X}_{1:N}, \mathbf{U}_{1:N}, t(\tau_{1:N})). \quad (19)$$

Since we are using the Legendre–Gauss points, we can also employ Gaussian quadrature to write

$$\mathbf{X}_{N+1} = \mathbf{X}_0 + \left(\frac{t_f - t_0}{2} \right) \mathbf{w}^T \mathbf{F}(\mathbf{X}_{1:N}, \mathbf{U}_{1:N}, t(\tau_{1:N})), \quad (20)$$

where \mathbf{X}_{N+1} is the terminal state and \mathbf{w} is the column vector of Legendre–Gauss quadrature weights.

The NLP problem obtained from transcribing the optimal control problem in Eqs. (1) can be written as:

$$\begin{aligned} \text{Minimize } J &= \phi(\mathbf{X}_{N+1}, t_f) \\ &+ \left(\frac{t_f - t_0}{2} \right) \sum_{i=1}^N w_i \mathcal{L}(\mathbf{X}_i, \mathbf{U}_i, t(\tau_i)) \\ \text{Subject To:} \\ g_i(\mathbf{X}_{1:N+1}, \mathbf{U}_{1:N+1}, t(\tau_{1:N+1})) &\leq 0, \quad i = 1, 2, \dots, n_g \\ h_i(\mathbf{X}_{1:N+1}, \mathbf{U}_{1:N+1}, t(\tau_{1:N+1})) &= 0, \quad i = 1, 2, \dots, n_h \\ \mathbf{h}_{PS} := \mathbf{D}\mathbf{X}_{0:N} - \left(\frac{t_f - t_0}{2} \right) \mathbf{F}(\mathbf{X}_{1:N}, \mathbf{U}_{1:N}, t(\tau_{1:N})) &= \mathbf{0}. \end{aligned} \quad (21)$$

We note that an additional Nn_x equality constraints are added to the problem due to the Legendre–Gauss PS method.

Appendix B Hermite–Legendre–Gauss–Lobatto method derivation

The HLGL method is a local method formulated by Williams [27]. We will develop and summarize this method here. The time interval $[t_0, t_f]$ is first subdivided into m subintervals where the $(m+1)$ th-order LGL points are used as the subinterval boundaries in this paper. The time domain for each subinterval is then transformed into $\tau = [-1, 1]$ using the subinterval boundaries. Within each of these subintervals, the state is approximated by an n th-order polynomial [27].

In the HLGL method, n is restricted to be an odd integer such that $n \geq 3$. For the i th subinterval, the state approximation can be written as

$$\mathbf{x}_i(\tau) = \mathbf{a}_0 + \mathbf{a}_1\tau + \mathbf{a}_2\tau^2 + \dots + \mathbf{a}_n\tau^n. \quad (22)$$

In order to form this polynomial, the n LGL points corresponding to the i th subinterval are represented by ξ and are divided into $\frac{n+1}{2}$ nodes denoted by γ and $\frac{n-1}{2}$ collocation points denoted by ζ :

$$\gamma_j \triangleq \xi_{2j-1}, \quad j = 1, \dots, (n+1)/2 \quad (23)$$

$$\zeta_j \triangleq \xi_{2j}, \quad j = 1, \dots, (n-1)/2. \quad (24)$$

The $(n+1)$ pieces of information needed to find the coefficients are obtained from the nodes using both the state values and its derivatives in the τ domain:

$$\begin{bmatrix} 1 & -1 & 1 & \cdots & \gamma_1^{n-1} & \gamma_1^n \\ 1 & \gamma_2^1 & \gamma_2^2 & \cdots & \gamma_2^{n-1} & \gamma_2^n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \gamma_{\frac{n+1}{2}}^1 & \gamma_{\frac{n+1}{2}}^2 & \cdots & \gamma_{\frac{n+1}{2}}^{n-1} & \gamma_{\frac{n+1}{2}}^n \\ 0 & 1 & 2\gamma_1^1 & \cdots & (n-1)\gamma_1^{n-2} & n\gamma_1^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 2\gamma_{\frac{n+1}{2}}^1 & \cdots & (n-1)\gamma_{\frac{n+1}{2}}^{n-2} & n\gamma_{\frac{n+1}{2}}^{n-1} \end{bmatrix} \begin{bmatrix} \mathbf{a}_0 \\ \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix} = \begin{bmatrix} \mathbf{x}(\gamma_1) \\ \mathbf{x}(\gamma_2) \\ \vdots \\ \mathbf{x}\left(\gamma_{\frac{n+1}{2}}\right) \\ \frac{h_i}{2} \mathbf{f}(\gamma_1) \\ \vdots \\ \frac{h_i}{2} \mathbf{f}\left(\gamma_{\frac{n+1}{2}}\right) \end{bmatrix}, \tag{25}$$

where h_i is the length of the i th subinterval in t space. Equation (25) is of the form $\mathbf{A}\mathbf{a} = \mathbf{b}$, so we can now determine the values of the state at the collocation points by inverting this system of equations:

$$\mathbf{x}_i(\zeta_j) = \mathbf{v}_j^T \mathbf{a} = \mathbf{v}_j^T \mathbf{A}^{-1} \mathbf{b}_i = \boldsymbol{\phi}_j^T \mathbf{b}_i, \tag{26}$$

where

$$\mathbf{v}_j = \begin{bmatrix} 1 & \zeta_j & \zeta_j^2 & \cdots & \zeta_j^n \end{bmatrix}^T, \tag{27}$$

$j = 1, \dots, (n-1)/2.$

If we take the derivative of Eq. (22) and evaluate at the collocation points, then we obtain

$$\dot{\mathbf{x}}_i(\zeta_j) = \mathbf{v}_{d_j}^T \mathbf{a} = \mathbf{v}_{d_j}^T \mathbf{A}^{-1} \mathbf{b}_i = \boldsymbol{\phi}_{d_j}^T \mathbf{b}_i, \tag{28}$$

where

$$\mathbf{v}_{d_j} = \begin{bmatrix} 0 & 1 & 2\zeta_j & \cdots & n\zeta_j^{n-1} \end{bmatrix}^T, \tag{29}$$

$j = 1, \dots, (n-1)/2.$

If we form the following matrices for the i th subinterval:

$$\boldsymbol{\Phi}_i = [\boldsymbol{\phi}_1 \ \boldsymbol{\phi}_2 \ \cdots \ \boldsymbol{\phi}_{(n-1)/2}] \tag{30}$$

$$\boldsymbol{\Phi}_{d_i} = [\boldsymbol{\phi}_{d_1} \ \boldsymbol{\phi}_{d_2} \ \cdots \ \boldsymbol{\phi}_{d_{(n-1)/2}}], \tag{31}$$

then the constraints to be enforced within the i th subinterval can now be written as

$$\boldsymbol{\Phi}_{d_i}^T \mathbf{b}_i - \frac{h_i}{2} \mathbf{F}\left(\boldsymbol{\Phi}_i^T \mathbf{b}_i, \mathbf{u}_i(\zeta_i), t_i(\zeta_i)\right) = \mathbf{0}, \tag{32}$$

where \mathbf{u}_i is evaluated at ζ_i through linear interpolation of the values at γ_i , the nodes in the i th subinterval.

The NLP problem obtained from transcribing the optimal control problem in Eqs. (1) can be written as:

Minimize $J = \phi(\mathbf{x}_m(1), t_f)$

$$+ \sum_{i=1}^m \frac{h_i}{2} \sum_{j=1}^n w_j \mathcal{L}(\mathbf{x}_i(\xi_j), \mathbf{u}_i(\xi_j), t_i(\xi_j))$$

Subject To:

$$g_i(\mathbf{x}(\boldsymbol{\gamma}), \mathbf{u}(\boldsymbol{\gamma}), t(\boldsymbol{\gamma})) \leq 0, \quad i = 1, 2, \dots, n_g$$

$$h_i(\mathbf{x}(\boldsymbol{\gamma}), \mathbf{u}(\boldsymbol{\gamma}), t(\boldsymbol{\gamma})) = 0, \quad i = 1, 2, \dots, n_h$$

$$\mathbf{h}_{\text{HLGL}} := \boldsymbol{\Phi}_{d_i}^T \mathbf{b}_i$$

$$- \frac{h_i}{2} \mathbf{F}\left(\boldsymbol{\Phi}_i^T \mathbf{b}_i, \mathbf{u}_i(\zeta_i), t_i(\zeta_i)\right) = \mathbf{0}, \quad i = 1, 2, \dots, m \tag{33}$$

where w_j are the Legendre–Gauss–Lobatto quadrature weights within each subinterval. We note that an additional $mn_x \left(\frac{n-1}{2}\right)$ equality constraints are added to the problem due to the HLGL method.

Appendix C Complete comparison results

Please see Table 4.

Table 4 Detailed metrics for the comparison of HLGL (m, n) pairs with the LG pseudospectral method—bolded case numbers indicate convergence

Case number	Nodes N	PS or HLGL pair (m, n)	Transfer duration (s)	Position RMS error (m)	Terminal position RMS error (m)	Velocity RMS error (m/s)	Thrust RMS error (N/kg)	Processor time (s)
–	6	PS	3.311×10^7	1.429×10^{10}	5.664×10^6	1.518×10^3	2.906×10^{-4}	36.3
1	6	(5, 3)	3.063×10^7	2.396×10^9	1.421×10^8	3.756×10^2	1.337×10^{-4}	9.33
2	6	(1, 11)	3.076×10^7	2.263×10^9	9.425×10^5	3.079×10^2	1.317×10^{-4}	9.62
–	7	PS	3.074×10^7	1.950×10^9	1.745×10^6	2.493×10^2	1.081×10^{-4}	275
3	7	(6, 3)	3.073×10^7	2.684×10^9	1.671×10^8	3.784×10^2	1.258×10^{-4}	8.6
4	7	(3, 5)	3.073×10^7	2.495×10^9	2.007×10^7	3.488×10^2	1.264×10^{-4}	11.5
5	7	(2, 7)	3.064×10^7	1.106×10^9	1.426×10^7	1.846×10^2	9.276×10^{-5}	4.79
6	7	(1, 13)	3.080×10^7	3.200×10^9	1.244×10^7	4.373×10^2	1.309×10^{-4}	8.57
–	8	PS	3.071×10^7	6.812×10^8	1.508×10^6	1.838×10^2	1.127×10^{-4}	18.1
7	8	(7, 3)	3.068×10^7	1.075×10^9	4.487×10^7	2.103×10^2	1.023×10^{-4}	12.9
8	8	(1, 15)	3.067×10^7	9.175×10^8	2.228×10^7	2.018×10^2	1.064×10^{-4}	5.43
–	9	PS	3.066×10^7	1.181×10^9	4.142×10^6	2.123×10^2	9.082×10^{-5}	25
9	9	(8, 3)	3.067×10^7	1.484×10^9	7.484×10^6	2.508×10^2	1.066×10^{-4}	35
10	9	(4, 5)	3.069×10^7	1.623×10^9	3.890×10^6	2.591×10^2	1.002×10^{-4}	19.1
11	9	(2, 9)	3.062×10^7	2.186×10^8	9.508×10^6	6.772×10^1	5.293×10^{-5}	7.85
12	9	(1, 17)	3.069×10^7	1.784×10^9	7.037×10^6	2.829×10^2	1.099×10^{-4}	7.41
–	10	PS	3.063×10^7	5.104×10^8	4.139×10^6	8.701×10^1	8.466×10^{-5}	22.6
13	10	(9, 3)	3.059×10^7	4.199×10^9	1.115×10^9	6.689×10^2	2.138×10^{-4}	3.99
14	10	(3, 7)	3.067×10^7	7.937×10^8	3.066×10^6	1.735×10^2	9.501×10^{-5}	21.7
15	10	(1, 19)	3.066×10^7	6.904×10^8	2.771×10^6	1.553×10^2	8.841×10^{-5}	9.88
–	11	PS	3.070×10^7	1.006×10^9	9.050×10^6	2.422×10^2	1.293×10^{-4}	26.9
16	11	(10, 3)	3.066×10^7	1.219×10^9	9.726×10^7	2.021×10^2	8.962×10^{-5}	34.1
17	11	(5, 5)	3.064×10^7	1.116×10^9	3.697×10^7	1.902×10^2	8.683×10^{-5}	17.1
18	11	(2, 11)	3.059×10^7	3.352×10^8	4.726×10^6	6.112×10^1	3.763×10^{-5}	11
19	11	(1, 21)	3.067×10^7	1.317×10^9	5.192×10^6	2.171×10^2	9.203×10^{-5}	8.98
–	12	PS	3.117×10^7	3.178×10^9	1.819×10^7	5.354×10^2	1.925×10^{-4}	465
20	12	(11, 3)	3.059×10^7	4.071×10^8	4.726×10^7	1.410×10^2	8.021×10^{-5}	30.8
21	12	(1, 23)	3.064×10^7	3.976×10^8	4.538×10^6	1.234×10^2	7.833×10^{-5}	10.7
–	13	PS	3.069×10^7	8.970×10^8	6.112×10^6	2.026×10^2	1.614×10^{-4}	188
22	13	(12, 3)	3.072×10^7	3.986×10^9	4.360×10^7	6.223×10^2	1.722×10^{-4}	14.5
23	13	(6, 5)	3.143×10^7	5.823×10^9	1.251×10^7	6.635×10^2	1.503×10^{-4}	23.6
24	13	(4, 7)	3.062×10^7	4.185×10^8	3.156×10^7	8.217×10^1	5.845×10^{-5}	22.3

Table 4 continued

Case number	Nodes N	PS or HJGL pair (m, n)	Transfer duration (s)	Position RMS error (m)	Terminal position RMS error (m)	Velocity RMS error (m/s)	Thrust RMS error (N/kg)	Processor time (s)
25	13	(3, 9)	3.064×10^7	1.311×10^9	1.485×10^7	2.167×10^2	9.290×10^{-5}	19.4
26	13	(2, 13)	3.061×10^7	1.408×10^8	1.960×10^6	4.582×10^1	3.842×10^{-5}	14.2
27	13	(1, 25)	3.072×10^7	1.305×10^9	2.375×10^6	2.330×10^2	9.050×10^{-5}	11.2
–	14	PS	3.063×10^7	2.192×10^8	7.034×10^6	5.503×10^1	6.337×10^{-5}	165
28	14	(13, 3)	3.143×10^7	4.986×10^9	9.905×10^7	7.855×10^2	2.143×10^{-4}	9.93
29	14	(1, 27)	3.068×10^7	3.814×10^8	6.887×10^5	1.101×10^2	8.252×10^{-5}	23.3
–	15	PS	3.269×10^7	1.255×10^{10}	1.783×10^6	1.376×10^3	2.655×10^{-4}	59.3
30	15	(14, 3)	3.119×10^7	4.979×10^9	2.571×10^8	7.128×10^2	2.229×10^{-4}	10.6
31	15	(7, 5)	3.057×10^7	1.767×10^8	2.006×10^7	9.269×10^1	6.795×10^{-5}	37
32	15	(2, 15)	3.076×10^7	2.717×10^9	3.054×10^6	3.510×10^2	9.410×10^{-5}	24.1
33	15	(1, 29)	3.065×10^7	7.505×10^8	1.443×10^6	1.296×10^2	6.890×10^{-5}	81.7
–	16	PS	3.111×10^7	2.816×10^9	9.673×10^6	4.872×10^2	2.161×10^{-4}	123
34	16	(15, 3)	3.150×10^7	5.054×10^9	5.578×10^7	8.275×10^2	2.461×10^{-4}	9.15
35	16	(5, 7)	3.084×10^7	4.571×10^9	7.010×10^6	5.716×10^2	1.589×10^{-4}	16.1
36	16	(3, 11)	3.069×10^7	9.933×10^8	1.693×10^7	2.144×10^2	1.232×10^{-4}	22.1
37	16	(1, 31)	3.158×10^7	4.656×10^9	3.338×10^5	8.427×10^2	2.430×10^{-4}	20.1
–	17	PS	3.136×10^7	4.705×10^9	1.632×10^6	5.611×10^2	2.554×10^{-4}	213
38	17	(16, 3)	3.144×10^7	5.144×10^9	9.378×10^7	7.697×10^2	2.186×10^{-4}	13.9
39	17	(8, 5)	3.046×10^7	1.461×10^9	5.002×10^7	1.844×10^2	6.641×10^{-5}	42.6
40	17	(4, 9)	3.073×10^7	2.333×10^9	6.374×10^6	2.921×10^2	9.222×10^{-5}	29.9
41	17	(2, 17)	3.082×10^7	3.420×10^9	1.073×10^7	4.409×10^2	1.105×10^{-4}	17.9
42	17	(1, 33)	3.133×10^7	4.369×10^9	4.975×10^5	6.299×10^2	1.961×10^{-4}	37.7
–	18	PS	3.076×10^7	7.823×10^8	3.548×10^7	2.509×10^2	1.267×10^{-4}	151
43	18	(17, 3)	3.161×10^7	5.029×10^9	2.772×10^7	9.046×10^2	2.502×10^{-4}	11
44	18	(1, 35)	3.171×10^7	4.998×10^9	2.293×10^7	9.228×10^2	2.537×10^{-4}	10
–	19	PS	3.066×10^7	9.082×10^8	5.192×10^7	1.950×10^2	1.670×10^{-4}	374
45	19	(18, 3)	3.115×10^7	5.179×10^9	3.298×10^8	6.430×10^2	2.396×10^{-4}	42.2
46	19	(9, 5)	3.056×10^7	2.487×10^8	4.770×10^7	9.402×10^1	6.349×10^{-5}	47.9
47	19	(6, 7)	3.075×10^7	3.085×10^9	5.964×10^6	4.441×10^2	1.454×10^{-4}	42.1
48	19	(3, 13)	3.092×10^7	4.270×10^9	1.074×10^6	6.956×10^2	1.695×10^{-4}	28
49	19	(2, 19)	3.081×10^7	3.460×10^9	8.380×10^6	4.677×10^2	1.268×10^{-4}	18.9

Table 4 continued

Case number	Nodes N	PS or HILGL pair (m, n)	Transfer duration (s)	Position RMS error (m)	Terminal position RMS error (m)	Velocity RMS error (m/s)	Thrust RMS error (N/kg)	Processor time (s)
50	19	(1, 37)	3.172×10^7	5.962×10^9	5.652×10^7	9.694×10^2	2.421×10^{-4}	85.5
–	20	PS	3.074×10^7	8.176×10^8	5.847×10^6	2.375×10^2	1.329×10^{-4}	304
51	20	(19, 3)	3.105×10^7	4.690×10^9	3.875×10^8	6.048×10^2	1.881×10^{-4}	36.5
52	20	(1, 39)	3.172×10^7	5.085×10^9	3.354×10^8	9.379×10^2	2.565×10^{-4}	25.2
–	21	PS	3.063×10^7	3.244×10^8	1.460×10^7	5.787×10^1	4.602×10^{-5}	2208
53	21	(20, 3)	3.138×10^7	4.964×10^9	1.508×10^8	7.687×10^2	2.108×10^{-4}	40
54	21	(10, 5)	3.078×10^7	4.436×10^9	1.258×10^9	5.368×10^2	1.390×10^{-4}	56.9
55	21	(5, 9)	3.078×10^7	3.155×10^9	2.382×10^7	4.724×10^2	1.268×10^{-4}	39.1
56	21	(4, 11)	3.068×10^7	1.338×10^9	2.499×10^5	2.209×10^2	7.586×10^{-5}	41
57	21	(2, 21)	3.078×10^7	2.883×10^9	7.526×10^5	3.855×10^2	1.084×10^{-4}	30.9
58	21	(1, 41)	3.173×10^7	8.494×10^9	8.433×10^8	1.129×10^3	2.852×10^{-4}	356
–	22	PS	3.064×10^7	1.694×10^8	6.789×10^5	5.714×10^1	4.754×10^{-5}	781
59	22	(21, 3)	2.958×10^7	8.105×10^9	2.966×10^8	1.125×10^3	1.606×10^{-4}	70.5
60	22	(7, 7)	3.078×10^7	2.783×10^9	2.289×10^7	4.243×10^2	1.357×10^{-4}	48.4
61	22	(3, 15)	3.064×10^7	2.267×10^8	1.523×10^6	7.459×10^1	6.107×10^{-5}	54.9
62	22	(1, 43)	–	–	–	–	–	–
–	23	PS	3.067×10^7	6.953×10^8	5.855×10^6	1.598×10^2	1.401×10^{-4}	532
63	23	(22, 3)	3.124×10^7	5.235×10^9	1.739×10^8	6.512×10^2	2.155×10^{-4}	23.1
64	23	(11, 5)	3.155×10^7	5.155×10^9	5.176×10^7	8.175×10^2	2.247×10^{-4}	13.9
65	23	(2, 23)	3.065×10^7	1.102×10^9	1.312×10^6	1.790×10^2	7.293×10^{-5}	49.1
66	23	(1, 45)	–	–	–	–	–	–
–	24	PS	3.072×10^7	6.821×10^8	1.675×10^6	1.984×10^2	1.044×10^{-4}	219
67	24	(23, 3)	3.119×10^7	5.050×10^9	1.815×10^8	6.561×10^2	2.357×10^{-4}	25.1
68	24	(1, 47)	–	–	–	–	–	–
–	25	PS	3.067×10^7	3.411×10^8	2.098×10^6	1.038×10^2	1.157×10^{-4}	646
69	25	(24, 3)	3.148×10^7	5.092×10^9	5.451×10^7	8.337×10^2	2.705×10^{-4}	20.4
70	25	(12, 5)	3.129×10^7	4.936×10^9	1.036×10^8	7.579×10^2	2.144×10^{-4}	24.6
71	25	(8, 7)	3.087×10^7	4.597×10^9	2.309×10^7	5.727×10^2	1.559×10^{-4}	45.8
72	25	(6, 9)	3.063×10^7	9.891×10^7	6.489×10^6	2.702×10^1	2.999×10^{-5}	68
73	25	(4, 13)	3.078×10^7	4.209×10^9	7.105×10^6	5.780×10^2	1.472×10^{-4}	37.9
74	25	(3, 17)	3.098×10^7	3.017×10^9	1.297×10^6	4.882×10^2	1.573×10^{-4}	51
75	25	(2, 25)	3.091×10^7	3.188×10^9	3.204×10^5	4.080×10^2	1.434×10^{-4}	42

Table 4 continued

Case number	Nodes N	PS or HILGL pair (m, n)	Transfer duration (s)	Position RMS error (m)	Terminal position RMS error (m)	Velocity RMS error (m/s)	Thrust RMS error (N/kg)	Processor time (s)
76	25	(1, 49)	—	—	—	—	—	—
—	26	PS	3.071×10^7	6.688×10^8	9.738×10^6	1.897×10^2	1.001×10^{-4}	489
77	26	(25, 3)	2.982×10^7	6.801×10^9	2.007×10^7	1.154×10^3	1.863×10^{-4}	82.2
78	26	(5, 11)	3.066×10^7	4.389×10^8	1.132×10^7	8.761×10^1	5.857×10^{-5}	60.2
79	26	(1, 51)	—	—	—	—	—	—
—	27	PS	3.066×10^7	3.667×10^8	1.134×10^7	1.001×10^2	1.049×10^{-4}	1041
80	27	(26, 3)	3.129×10^7	4.826×10^9	1.526×10^8	8.044×10^2	2.677×10^{-4}	23.4
81	27	(13, 5)	3.073×10^7	4.131×10^9	1.040×10^7	6.061×10^2	1.540×10^{-4}	75.7
82	27	(2, 27)	3.070×10^7	3.490×10^8	4.080×10^5	1.122×10^2	9.816×10^{-5}	228
83	27	(1, 53)	—	—	—	—	—	—
—	28	PS	3.070×10^7	6.265×10^8	7.166×10^6	1.772×10^2	9.888×10^{-5}	1062
84	28	(27, 3)	3.133×10^7	4.873×10^9	1.228×10^8	8.049×10^2	2.606×10^{-4}	24.9
85	28	(9, 7)	3.089×10^7	4.588×10^9	3.411×10^6	5.219×10^2	1.426×10^{-4}	76.8
86	28	(3, 19)	3.178×10^7	6.791×10^9	2.156×10^6	1.054×10^3	1.940×10^{-4}	87.1
87	28	(1, 55)	—	—	—	—	—	—
—	29	PS	3.066×10^7	2.433×10^8	9.502×10^6	8.697×10^1	1.181×10^{-4}	1154
88	29	(28, 3)	3.128×10^7	4.828×10^9	1.542×10^8	8.061×10^2	2.801×10^{-4}	27.7
89	29	(14, 5)	3.160×10^7	5.075×10^9	4.118×10^7	8.737×10^2	2.362×10^{-4}	22.1
90	29	(7, 9)	3.095×10^7	4.718×10^9	2.349×10^7	5.305×10^2	1.412×10^{-4}	52.3
91	29	(4, 15)	3.081×10^7	3.405×10^9	1.492×10^6	4.561×10^2	1.256×10^{-4}	98.2
92	29	(2, 29)	3.142×10^7	4.792×10^9	1.223×10^8	6.105×10^2	1.494×10^{-4}	79
93	29	(1, 57)	—	—	—	—	—	—
—	30	PS	3.064×10^7	1.895×10^8	9.118×10^6	5.839×10^1	9.171×10^{-5}	655
94	30	(29, 3)	3.148×10^7	5.180×10^9	4.135×10^7	7.762×10^2	2.478×10^{-4}	39
95	30	(1, 59)	—	—	—	—	—	—
—	31	PS	3.066×10^7	2.479×10^8	1.148×10^7	8.286×10^1	1.185×10^{-4}	1611
96	31	(30, 3)	3.135×10^7	4.930×10^9	1.121×10^8	7.772×10^2	2.444×10^{-4}	31.7
97	31	(15, 5)	3.044×10^7	2.638×10^9	8.007×10^7	4.416×10^2	1.099×10^{-4}	134
98	31	(10, 7)	3.062×10^7	5.829×10^8	1.744×10^7	9.989×10^1	5.111×10^{-5}	116
99	31	(6, 11)	3.063×10^7	8.109×10^7	3.016×10^7	1.950×10^1	2.893×10^{-5}	85.2
100	31	(5, 13)	3.065×10^7	2.774×10^8	1.757×10^7	7.107×10^1	6.185×10^{-5}	74
101	31	(3, 21)	3.083×10^7	3.030×10^9	1.061×10^6	4.505×10^2	1.145×10^{-4}	62.3

Table 4 continued

Case number	Nodes N	PS or HJGL pair (m, n)	Transfer duration (s)	Position RMS error (m)	Terminal position RMS error (m)	Velocity RMS error (m/s)	Thrust RMS error (N/kg)	Processor time (s)
102	31	(2, 31)	3.162×10^7	4.884×10^9	1.143×10^6	8.409×10^2	2.498×10^{-4}	75.6
103	31	(1, 61)	–	–	–	–	–	–
–	32	PS	3.069×10^7	5.690×10^8	3.946×10^6	1.579×10^2	9.508×10^{-5}	1146
104	32	(31, 3)	3.147×10^7	4.933×10^9	7.166×10^7	8.113×10^2	2.071×10^{-4}	162
105	32	(1, 63)	–	–	–	–	–	–
–	33	PS	3.065×10^7	1.838×10^8	2.664×10^6	7.333×10^1	1.154×10^{-4}	1649
106	33	(32, 3)	3.159×10^7	5.114×10^9	2.399×10^7	8.575×10^2	2.426×10^{-4}	27.9
107	33	(16, 5)	3.129×10^7	4.768×10^9	2.461×10^8	7.631×10^2	2.206×10^{-4}	44.5
108	33	(8, 9)	3.171×10^7	5.098×10^9	6.841×10^6	9.347×10^2	2.539×10^{-4}	7.11
109	33	(4, 17)	3.065×10^7	1.127×10^8	3.930×10^7	4.354×10^1	4.854×10^{-5}	148
110	33	(2, 33)	3.171×10^7	4.865×10^9	3.751×10^6	8.917×10^2	2.503×10^{-4}	222
111	33	(1, 65)	–	–	–	–	–	–
–	34	PS	3.069×10^7	3.527×10^8	1.524×10^6	1.284×10^2	1.019×10^{-4}	489
112	34	(33, 3)	3.162×10^7	5.107×10^9	2.352×10^7	8.813×10^2	2.501×10^{-4}	32.2
113	34	(11, 7)	3.061×10^7	2.288×10^8	3.857×10^6	6.981×10^1	5.652×10^{-5}	127
114	34	(3, 23)	3.100×10^7	4.152×10^9	1.505×10^6	6.142×10^2	1.455×10^{-4}	71.2
115	34	(1, 67)	–	–	–	–	–	–
–	35	PS	3.065×10^7	1.653×10^8	2.598×10^6	6.909×10^1	1.048×10^{-4}	511
116	35	(34, 3)	3.149×10^7	5.166×10^9	4.619×10^7	7.846×10^2	2.338×10^{-4}	57.1
117	35	(17, 5)	3.146×10^7	4.974×10^9	1.006×10^8	8.346×10^2	2.404×10^{-4}	26.3
118	35	(2, 35)	–	–	–	–	–	–
119	35	(1, 69)	–	–	–	–	–	–
–	36	PS	3.063×10^7	1.311×10^8	1.528×10^6	3.283×10^1	3.362×10^{-5}	2033
120	36	(35, 3)	3.156×10^7	5.131×10^9	2.865×10^7	8.424×10^2	2.837×10^{-4}	41.6
121	36	(7, 11)	3.080×10^7	3.541×10^9	6.004×10^6	4.694×10^2	1.397×10^{-4}	85.6
122	36	(5, 15)	3.090×10^7	4.371×10^9	8.978×10^5	5.451×10^2	1.452×10^{-4}	102
123	36	(1, 71)	–	–	–	–	–	–
–	37	PS	3.065×10^7	1.347×10^8	2.233×10^4	6.366×10^1	1.026×10^{-4}	610
124	37	(36, 3)	3.167×10^7	5.133×10^9	1.091×10^7	8.934×10^2	2.447×10^{-4}	43.9
125	37	(18, 5)	3.132×10^7	5.001×10^9	1.628×10^8	7.544×10^2	2.197×10^{-4}	42.1

Table 4 continued

Case number	Nodes N	PS or HJGL pair (m, n)	Transfer duration (s)	Position RMS error (m)	Terminal position RMS error (m)	Velocity RMS error (m/s)	Thrust RMS error (N/kg)	Processor time (s)
126	37	(12, 7)	3.320×10^7	1.358×10^{10}	9.870×10^8	1.579×10^3	2.596×10^{-4}	295
127	37	(9, 9)	3.097×10^7	5.193×10^9	7.958×10^6	7.328×10^2	1.744×10^{-4}	130
128	37	(6, 13)	3.095×10^7	4.377×10^9	1.584×10^6	5.717×10^2	1.359×10^{-4}	107
129	37	(4, 19)	3.086×10^7	3.117×10^9	1.251×10^7	5.258×10^2	1.404×10^{-4}	80.3
130	37	(3, 25)	3.128×10^7	4.647×10^9	1.282×10^6	7.361×10^2	2.030×10^{-4}	78.4
131	37	(2, 37)	–	–	–	–	–	–
132	37	(1, 73)	–	–	–	–	–	–
–	38	PS	3.067×10^7	4.178×10^8	1.126×10^6	1.335×10^2	1.163×10^{-4}	546
133	38	(37, 3)	3.164×10^7	5.133×10^9	1.640×10^7	8.820×10^2	2.545×10^{-4}	37.2
134	38	(1, 75)	–	–	–	–	–	–
–	39	PS	3.065×10^7	1.365×10^8	3.408×10^6	6.031×10^1	1.006×10^{-4}	905
135	39	(38, 3)	3.142×10^7	4.883×10^9	6.501×10^7	7.861×10^2	2.145×10^{-4}	534
136	39	(19, 5)	3.145×10^7	5.053×10^9	1.071×10^8	8.030×10^2	2.397×10^{-4}	32.3
137	39	(2, 39)	3.173×10^7	4.953×10^9	3.223×10^8	9.077×10^2	2.567×10^{-4}	41
138	39	(1, 77)	–	–	–	–	–	–
–	40	PS	3.067×10^7	4.203×10^8	1.239×10^5	1.301×10^2	1.157×10^{-4}	633
139	40	(39, 3)	3.152×10^7	4.892×10^9	3.549×10^7	8.832×10^2	2.534×10^{-4}	497
140	40	(13, 7)	3.096×10^7	4.637×10^9	6.756×10^6	5.660×10^2	1.661×10^{-4}	107
141	40	(3, 27)	3.124×10^7	5.175×10^9	2.714×10^5	7.244×10^2	1.664×10^{-4}	87.4
142	40	(1, 79)	–	–	–	–	–	–

References

1. Abramowitz M, Stegun IA (1972) Handbook of mathematical functions with formulas, graphs, and mathematical tables. U.S. Department of Commerce, Washington, D.C.
2. Andrews GE, Berndt BC (2012) Ramanujan's lost notebook. Springer, New York
3. Betts JT (1998) Survey of numerical methods for trajectory optimization. *J Guid Control Dyn* 21(2):193–207
4. Choueiri EY (2009) New dawn for electric rockets. *Sci Am* 300(2):58–65
5. Conway BA (ed.) (2010) Spacecraft trajectory optimization. In: No. 29 in Cambridge Aerospace Series. Cambridge University Press, Cambridge
6. De Ruiter AHJ, Damaren CJ, Forbes JR (2013) Spacecraft dynamics and control: an introduction. Wiley, Chichester
7. Edelbaum TN (1961) Propulsion requirements for controllable satellites. *ARS J* 31(8):1079–1089
8. García-Heras J, Soler M, Sáez FJ (2016) Collocation methods to minimum-fuel trajectory problems with required time of arrival in ATM. *J Aerosp Inf Syst* 13(7):243–265
9. Garg D, Patterson M, Hager WW, Rao AV, Benson DA, Huntington GT (2010) A unified framework for the numerical solution of optimal control problems using pseudospectral methods. *Automatica* 46(11):1843–1851
10. Gill PE, Murray W, Saunders MA (2005) SNOPT: an SQP algorithm for large-scale constrained optimization. *SIAM Rev* 47(1):99–131
11. Hardy GH, Wright EM (1975) An introduction to the theory of numbers, 4th edn. Oxford University Press, Oxford
12. Herman AL, Conway BA (1996) Direct optimization using collocation based on high-order Gauss–Lobatto quadrature rules. *J Guid Control Dyn* 19(3):592–599
13. Izzo D (2006) Lambert's problem for exponential sinusoids. *J Guid Control Dyn* 29(5):1242–1245
14. Lei H, Liu T, Li D, Ye J, Shao L (2017) Adaptive mesh iteration method for trajectory optimization based on hermite-pseudospectral direct transcription. *Math Probl Eng* 2017:1–7
15. Nicolas JL, Robin G (1997) Highly composite numbers by Srinivasa Ramanujan. *Ramanujan J* 1(2):119–153
16. Petropoulos AE, Longuski JM (2004) Shape-based algorithm for the automated design of low-thrust, gravity assist trajectories. *J Spacecr Rockets* 41(5):787–796
17. Petropoulos AE, Sims JA (2002) A review of some exact solutions to the planar equations of motion of a thrusting spacecraft. In: 2nd international symposium on low thrust trajectories. Toulouse
18. Ramanujan S (1915) Highly composite numbers. *Proc Lond Math Soc* 2(1):347–409
19. Ramanujan S (1915) On the number of divisors of a number. *J Indian Math Soc* 7:131–133
20. Rao AV (2009) A survey of numerical methods for optimal control. *Adv Astronaut Sci* 135(1):497–528
21. Ross IM, Fahroo F (2004) Legendre pseudospectral approximations of optimal control problems. In: New trends in nonlinear dynamics and control and their applications. Lecture notes in control and information science. Springer, Berlin, pp 327–342
22. Shirazi A, Ceberio J, Lozano JA (2018) Spacecraft trajectory optimization: a review of models, objectives, approaches and solutions. *Prog Aerosp Sci* 102:76–98
23. Wall B (2008) Shape-based approximation method for low-thrust trajectory optimization. In: AIAA/AAS astrodynamics specialist conference and exhibit. American Institute of Aeronautics and Astronautics, Honolulu
24. Wall BJ, Conway BA (2009) Shape-based approach to low-thrust rendezvous trajectory design. *J Guid Control Dyn* 32(1):95–101
25. Wang Y, Zhu Y, Jiang X, Li S (2014) Comparison of LPM, GPM and RPM for optimization of low-thrust Earth-Mars rendezvous trajectories. In: Proceedings of 2014 IEEE Chinese guidance, navigation and control conference. IEEE, Yantai, pp 2461–2467
26. Williams P (2005) A comparison of differentiation and integration based direct transcription methods. *Adv Astronaut Sci* 120(1):389–408
27. Williams P (2009) Hermite–Legendre–Gauss–Lobatto direct transcription in trajectory optimization. *J Guid Control Dyn* 32(4):1392–1395