JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 31, No. 5, September–October 2008

# Two-Timescale Discretization Scheme for Collocation

Prasun N. Desai*
*NASA Langley Research Center, Hampton, Virginia 23681*
and
Bruce A. Conway†
*University of Illinois at Urbana–Champaign, Urbana, Illinois 61801*

DOI: 10.2514/1.33974

**The development of a two-timescale discretization scheme for collocation is presented. This scheme allows a coarser discretization to be used for slowly varying state variables and a second finer discretization to be used for state variables having higher-frequency dynamics. That is, the discretization scheme can be tailored to the dynamics of the particular state variables. Consequently, the size of the overall nonlinear programming problem can be reduced significantly. Two two-timescale discretization architecture schemes are described. Comparison of results between the two-timescale method and conventional single-timescale collocation shows very good agreement. When the two-timescale discretization is used in combination with the sparse nonlinear optimizer SNOPT, a significant reduction (by more than 60%) in the number of nonlinear programming parameters required for the transcription of the problem and iterations necessary for convergence can be achieved without sacrificing solution accuracy.**

## I. Introduction

IN THE collocation method of transcribing a continuous optimal control problem into a discrete problem, a single-timescale discretization scheme is used for all the state variables. This approach works very well for solving a great variety of problems. However, if the dynamics of one or some of the state variables are at a high frequency, a small discretization timescale is required (i.e., greater number of segments needed) to capture the system dynamics. However, with the present available collocation methodologies, even if only one state variable requires a small discretization timescale, all the state variables must use a small timescale due to the architecture of collocation. This limitation results in the size of the nonlinear programming (NLP) problem that must be solved becoming enormous due to the fine discretization employed. If a separate discretization scheme can be employed, one for the lower-frequency state variables (using a larger timescale) and another one for the higher-frequency state variables (at a finer timescale), the size of the NLP problem can be greatly reduced. Consequently, the finer discretization timescale can be tailored to only those state variables that have higher-frequency dynamics in the governing equations, whereas the rest of the state variables can use a much larger discretization timescale. This two-timescale methodology is well suited for solving six-degree-of-freedom trajectory optimization problems, where the dynamics of the translational motion state variables are at a low frequency, although the dynamics of the rotational motion state variables are at a high frequency [1].

This paper describes the development of a two-timescale discretization scheme for collocation. Before the discussion of the two-timescale collocation methodology, an overview of the standard single-timescale collocation method is provided. Then, the results from a test case using the two-timescale methodology are presented, and the solutions obtained from this new method and from conventional standard singe-timescale collocation are compared.

Received 13 August 2007; revision received 5 April 2008; accepted for publication 6 April 2008. 

*Senior Aerospace Engineer, Atmospheric Flight and Entry Systems Branch, 1 North Dryden Street, MS 489; prasun.n.desai@nasa.gov. Associate Fellow AIAA.

†Professor, Department of Aerospace Engineering, 306 Talbot Laboratory; bconway@uiuc.edu. Associate Fellow AIAA.

## II. Standard Collocation Methodology

The optimal control problem minimizes a cost function

$$J = J(x, u, t) \tag{1}$$

subject to a set of state equations of motion for the dynamical system

$$\mathrm{d}x/\mathrm{d}t = \bar{f}(x, u, t) \tag{2}$$

having initial $\chi$, final boundary $\Psi$, and path constraints $g$

$$\bar{\chi}(x(t_o), t_o) = \bar{0} \tag{3}$$

$$\bar{\Psi}(x(t_f), t_f) = \bar{0} \tag{4}$$

$$\bar{g}(x, u, t) \leq \bar{0} \tag{5}$$

where $x$ and $u$ are vectors representing the state and control variables. In the method of direct collocation with nonlinear programming (DCNLP) the total time ($T = t_f - t_o$) is discretized into $n$ segments (and $n + 1$ nodes or segment boundaries). The size of each segment can be a constant or of varying lengths. For this application, each segment length has been set to a constant ($t = T/n$).

The state differential Eq. (2) is approximated within each segment using an integration formula. The formulation for the approximate integration of the system equations transforms them into a set of discrete algebraic constraints (referred to as defect equations) imposed at each segment in the discretization. Within each segment, evaluation points are selected at which $x$ and $u$ must satisfy Eqs. (2–5). When satisfied, an approximate solution to the system equations is obtained. In this manner, for the overall NLP problem, the state and control parameters are chosen to minimize the performance function while representing an approximate solution to the system equations. The discrete algebraic constraint equations can have different forms depending upon the implicit integration formulation used. A detailed description of both the collocation method and several different implicit integration methods can be found in [2–4].

Figure 1 illustrates the system constraint formulation using a higher-order integration formula based on a fifth-degree Gauss–Lobatto polynomial [3,4]. In this version of the DCNLP method, each state variable is represented by a quintic polynomial (in time) within each segment. Six parameters are required to define the polynomial uniquely; the polynomial can be determined using the values of the states $(x_i, x_{i+1}, x_{i+2})$ at the segment boundaries and at the center point, together with the values of the time derivatives $f_i$, $f_{i+1}$, and $f_{i+2}$, which correspond to values of $\mathrm{d}x/\mathrm{d}t$ [Eq. (2)]

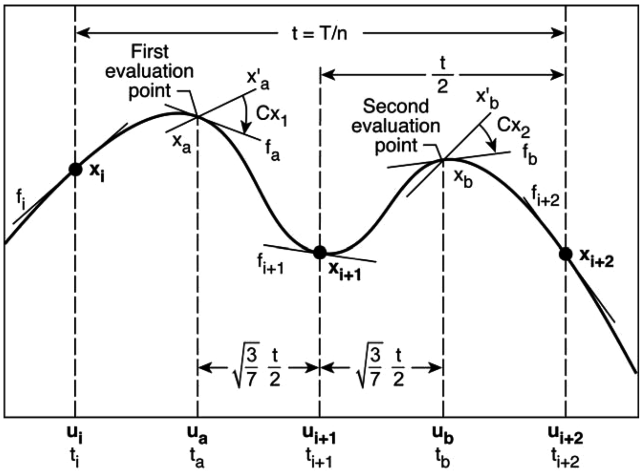


**Fig. 1 Fifth-degree Gauss–Lobatto system constraint formulation.**

evaluated at $t_i$, $t_{i+1}$, and $t_{i+2}$, respectively. Note that evaluation of $f_i$, $f_{i+1}$, and $f_{i+2}$ requires specification of the control variables at the same times (i.e., requires $u_i$, $u_{i+1}$, $u_{i+2}$). Making use of $x_i$, $x_{i+1}$, $x_{i+2}$, $u_i$, $u_{i+1}$, $u_{i+2}$, $f_i$, $f_{i+1}$, and $f_{i+2}$, a quintic polynomial representing the state time history between the endpoints can be constructed, as shown in Fig. 1.

In the collocation method using the fifth-degree Gauss–Lobatto polynomial, this quintic polynomial is forced to satisfy two additional constraints within each segment at the collocation points $t_a$ and $t_b$ (shown in Fig. 1). That is, at the two interior points $t_a$ and $t_b$, the quintic polynomial must satisfy the system differential equations. The location of these two points is not arbitrary; no other choice of two such points yields a more precise implicit integration. The algebraic constraints take the form [3,4]

$$Cx_1 = f_a - x'_a = \frac{1}{360}[(32\sqrt{21} + 180)x_i - 64\sqrt{21}x_{i+1} + (32\sqrt{21} - 180)x_{i+2} + t\{(9 + \sqrt{21})f_i + 98f_a + 64f_{i+1} + (9 - \sqrt{21})f_{i+2}\}] = 0 \quad (6)$$

$$Cx_2 = f_b - x'_b = \frac{1}{360}[(-32\sqrt{21} + 180)x_i + 64\sqrt{21}x_{i+1} + (-32\sqrt{21} - 180)x_{i+2} + t\{(9 - \sqrt{21})f_i + 98f_b + 64f_{i+1} + (9 + \sqrt{21})f_{i+2}\}] = 0 \quad (7)$$

where $f_a = f(x_a, u_a, t_a)$ and $f_b = f(x_b, u_b, t_b)$. The states $(x_i, x_{i+1}, x_{i+2})$ and controls $(u_i, u_a, u_{i+1}, u_b, u_{i+2})$ shown in bold in Fig. 1 are discrete NLP parameters. The values of these states and controls are selected to force the algebraic constraints to zero. In so doing, the polynomial is made to satisfy the differential equation at the collocation points $t_a$ and $t_b$ of the segment (in addition to being implicitly satisfied at $t_i$, $t_{i+1}$, and $t_{i+2}$). When Eqs. (6) and (7) are satisfied, an approximate solution to the system equations is obtained. The method is discussed in much greater detail in [3,4].

## III. Two-Timescale Collocation Discretization Methodology

A two-timescale discretization scheme for collocation uses a standard timescale discretization scheme for the smoothly varying lower-frequency state variables and another finer timescale discretization scheme for the higher-frequency state variables. Two different two-timescale discretization schemes are presented: 1) using two segments to represent the higher-frequency state variables for every one segment of the low-frequency state variables (i.e., a two-to-one discretization architecture), and 2) using four segments to represent the higher-frequency state variables for every one segment of the low-frequency state variables (i.e., a four-to-one discretization architecture). Greater details are offered in [5] on the development of this two-timescale methodology.

### A. Ratio of Two Segments to One Segment (2:1) Discretization Architecture

The system state variables have previously been represented by the vector $x$. For the two-timescale discretization, the elements of the state vector need to be divided into low- and high-frequency categories. Let the low-frequency state variables contained in $x$ be represented by the vector $y$, and the high-frequency state variables contained in $x$ be represented by the vector $z$. The control variables are collected into the vector $u$. Figure 2 illustrates the Gauss–Lobatto system constraint formulation for the original single-timescale and the proposed two-to-one discretization architecture. The part of this figure illustrating the original discretization has been simplified to contain only the salient features of Fig. 1. The center part of the figure shows that the discretization in time for the low-frequency $y$ states and is the same as the original discretization. For the high-frequency $z$ states, two segments (each having the same form as the original discretization) span the same total time, as seen at the bottom of the figure. The variables in bold are NLP parameters.

As seen in Fig. 2, some of the parameters, both states and controls highlighted in *italics* ($u_a$, $u_b$, $y_{1a}$, $y_{2a}$, and $u_{1a}$ through $u_{2b}$), that are required to evaluate the defect constraint equations ($Cy_i$ and $Cz_i$) do not occur at the same times in the low- and high-frequency segments. For example, to evaluate the defect equations $Cy_1$ at the left collocation point of the center diagram, a knowledge of (or estimate of) the high-frequency $z$ states is required at that location. However, the $z$ states are specified as NLP parameters $z_i$ through $z_{i+4}$ at the node endpoints, midpoint, quarter point, and three-quarter point of the bottom diagram. None of these points are coincident with the left collocation point of the middle diagram. The same difficulty occurs,

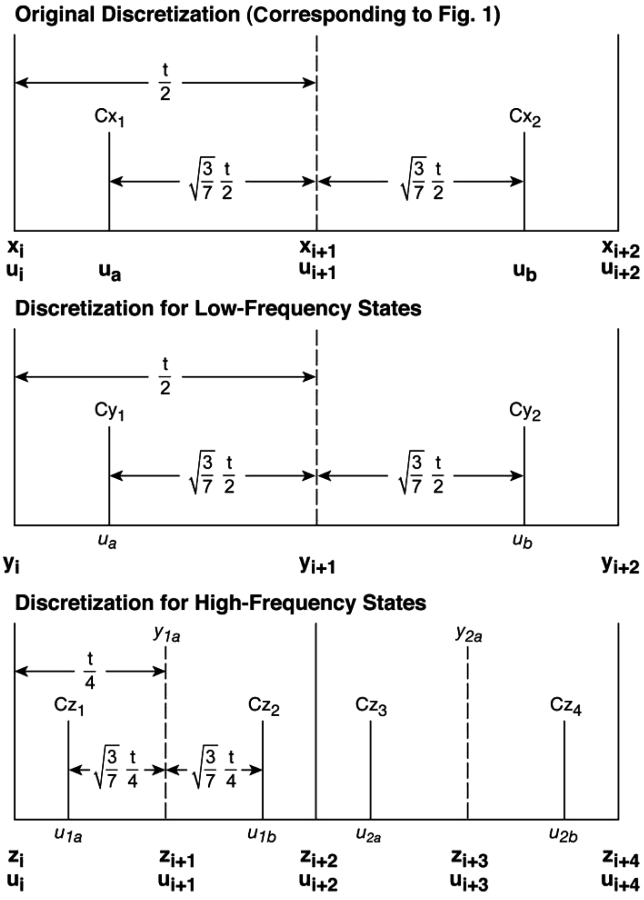


**Fig. 2 Gauss–Lobatto two-to-one two-timescale discretization scheme compared with original Gauss–Lobatto discretization (parameters in bold are NLP parameters).**

for example, when the defect equations $Cz_1$ needs to be evaluated in the bottom diagram. A knowledge of the values of the low-frequency states is required, but the $y$ states are specified as NLP parameters $y_i$, $y_{i+1}$, and $y_{i+2}$ at times that do not coincide with the time at which $Cz_1$ is evaluated. A similar problem arises with the control variables. They are included as NLP parameters $u_i$ through $u_{i+4}$ defined at the node endpoints, midpoint, quarter point, and three-quarter point of the bottom diagram but are needed at completely different times for evaluating the six sets of defect equations $Cy_1$ and $Cy_2$ in the low-frequency discretization segment and $Cz_1$ through $Cz_4$ in the high-frequency discretization segments.

To evaluate all of the defect constraint equations ($C$), the values for all of these state and control parameters (highlighted in *italics* in Fig. 2) somehow need to be determined. The obvious solution, although at the cost of increasing the size of the NLP problem, is to define all of them as additional NLP parameters. Unfortunately, this first approach produced incorrect results, because there was insufficient coupling between the low- and high-frequency segments. That is, parameters closely spaced in time (e.g., $u_i$, $u_{1a}$, $u_a$, and $u_{i+1}$) could have widely different values, leading to an incorrect solution. An analysis of this incorrect solution revealed that allowing the NLP solver the freedom to choose these parameters without any direct relation to one another provided insufficient coupling between the low-frequency discretization of the middle diagram and the high-frequency discretization of the bottom diagram. Therefore, an approach was sought that would strengthen the coupling between the low- and high-frequency segments so as to rigidly bind them together for this two-timescale discretization architecture.

For the low-frequency segment, the Gauss–Lobatto defect algebraic constraint equations are formulated as described in Sec. II. All the state and control variables ($y_i, y_{i+1}, y_{i+2}, z_i, z_{i+2}, z_{i+4}, u_i, u_{i+2}, u_{i+4}$) at the left-hand side, center, and right-hand side of the segment (shown in bold) that are necessary to evaluate the defect constraint equations ($Cy_1$ and $Cy_2$) are available (because they all are NLP parameters), except for the control variables $u_a$ and $u_b$ at the left-of-center and right-of-center locations within the segment. In the standard collocation architecture, these control variables were NLP parameters. However, in the two-to-one two-timescale formulation, they have been eliminated. As a result, their values need to be determined.

Several approaches were considered for determining these derived variables, using polynomials and splines, with the objective of providing sufficient coupling between adjacent values of the parameters and between the low- and high-frequency discretization segments. For the control parameters, a good solution was obtained by forming a quartic polynomial defined by the control variables $u_i$, $u_{i+1}$, $u_{i+2}$, $u_{i+3}$, and $u_{i+4}$ from the high-frequency segments, and evaluating this quartic polynomial at the specific times corresponding to $u_a$ and $u_b$. Appendix A provides the quartic polynomial function for determining $u_a$ and $u_b$, given $u_i$, $u_{i+1}$, $u_{i+2}$, $u_{i+3}$, and $u_{i+4}$. Consequently, all the variables that are required for evaluating the defect constraints equations $Cy_1$ and $Cy_2$ using Eqs. (6) and (7) are available for the standard or lower-frequency portion of the scheme.

Similarly, for the high-frequency discretization portion of the scheme, the Gauss–Lobatto defect algebraic constraint equations $Cz_1$ through $Cz_4$ need to be determined. As described in the preceding paragraphs, the following variables are needed that are not NLP parameters: the controls $u_{1a}$, $u_{1b}$, $u_{2a}$, and $u_{2b}$, along with two additional unknowns, the state variables $y_{1a}$ and $y_{2a}$ at the center points of the two segments. The values for these controls can be obtained by evaluating the same quartic polynomial described in the preceding paragraphs, by using $u_i$, $u_{i+1}$, $u_{i+2}$, $u_{i+3}$, and $u_{i+4}$, at their appropriate times to yield $u_{1a}$, $u_{1b}$, $u_{2a}$, and $u_{2b}$. The state variables $y_{1a}$ and $y_{2a}$ (which are functions of the low-frequency segment) are also needed. For the state parameters a good solution was obtained by forming a quintic polynomial using the NLP parameters at the left-hand ($y_i, z_i, u_i$), center ($y_{i+1}, z_{i+2}, u_{i+2}$), and right-hand ($y_{i+2}, z_{i+4}, u_{i+4}$) sides of the segment, along with their function evaluations $f_i$, $f_{i+1}$, and $f_{i+2}$ using Eq. (2) and evaluating this polynomial at their respective times. Appendix B provides the quintic polynomial function for determining $y_{1a}$ and $y_{2a}$, given $y_i$, $y_{i+1}$, $y_{i+2}$ and their associated slopes $f_i, f_{i+1}, f_{i+2}$. Consequently, all the variables that are required for evaluating the defect constraint equations $Cz_1$ and $Cz_2$ in the first segment and $Cz_3$ and $Cz_4$ in the second segment, using Eqs. (6) and (7), are available for the higher-frequency portion of the scheme. This process can be repeated for any number of segments.

### B. Ratio of Four Segments to One Segment (4:1) Discretization Architecture

If additional refinement is required for the high-frequency state variables, a four-to-one discretization architecture can be used. The Gauss–Lobatto system constraint formulation for the four-to-one discretization architecture is illustrated in Fig. 3. The overall scheme is very similar to that of the two-to-one architecture. However, there are four Gauss–Lobatto segments defined for the finer discretization state variables $z$ for every one Gauss–Lobatto segment for the standard or low-frequency state variables $y$. The variables in bold are again NLP parameters, although the variables in italics ($u_a$, $u_b$, $y_{1a}$ through $y_{6a}$, and $u_{1a}$ through $u_{4c}$) are derived parameters.

The process for determining the Gauss–Lobatto defect constraints equations $Cy_1$ and $Cy_2$ in the standard or low-frequency portion of the scheme is the same as that described in the two-to-one, two-timescale architecture (Sec. III.A). The control variables $u_a$ and $u_b$ are obtained by forming a quartic polynomial (see Appendix A) using the control variables $u_i$, $u_{i+1}$, $u_{i+2}$, $u_{i+3}$, and $u_{i+4}$ in the same manner described in Sec. III.A and evaluating this quartic polynomial at their appropriate times to yield $u_a$ and $u_b$. Consequently, all the variables that are required for evaluating the defect constraints equations $Cy_1$ and $Cy_2$ using Eqs. (6) and (7) are available.

For the finer discretization portion of the scheme, the Gauss–Lobatto algebraic constraints equations $Cz_1$ through $Cz_8$ are formulated similarly as described in the two-to-one architecture (Sec. III.A) as well. However, because there are four segments for this architecture, there are additional variables that are unknown and need to be determined. All the controls ($u_{ia}, u_{ib}, u_{ic}$) interior to each of the four finer discretization segments are obtained by evaluating the same quartic polynomial defined by $u_i$, $u_{i+1}$, $u_{i+2}$, $u_{i+3}$, and $u_{i+4}$ at their respective times. Similarly, all the interior state variables ($y_{ia}$) are obtained by forming a quintic polynomial (see Appendix B) using the NLP parameters at the left-hand ($y_i, z_i, u_i$), center

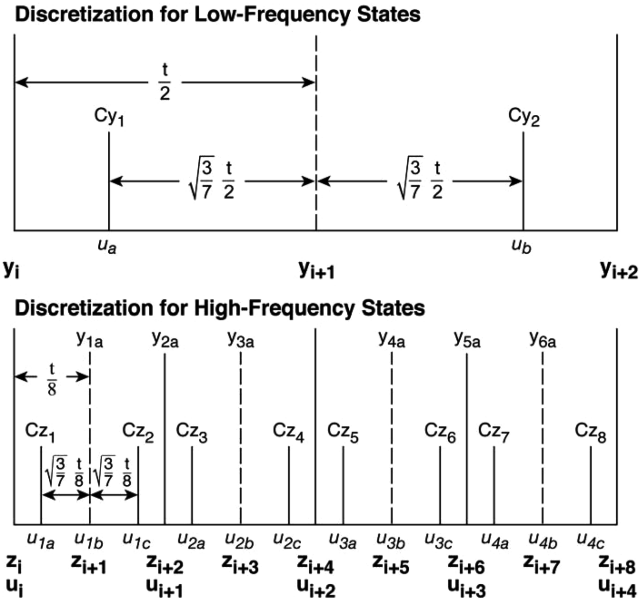


**Fig. 3 Gauss–Lobatto four-to-one two-timescale discretization scheme (parameters in bold are NLP parameters).**

$(y_{i+1}, z_{i+4}, u_{i+2})$, and right-hand $(y_{i+2}, z_{i+8}, u_{i+4})$ sides of the interior segments, along with their function evaluations $f_i$, $f_{i+1}$, and $f_{i+2}$ using Eq. (2) and evaluating this polynomial at their respective times. Consequently, all the variables that are required for evaluating the defect constraint equations $Cz_1$ through $Cz_8$ using Eqs. (6) and (7) within the four segments are available for the high-frequency portion of the scheme. This process can be repeated for any number of segments.

### C. Numerical Accuracy of the Two-Timescale Architecture

When using any numerical method for optimization, both a determination of the accuracy of which the method is capable and a knowledge of the parameters that influence this accuracy are important. The method of direct collocation uses implicit integration. That is, if the defect constraint equations are satisfied, then the integration of the system equations is equivalent to numerical integration across the segment using the specified rule. Before discussing the accuracy to be expected of the two-timescale discretization methodology developed in this research, a review is first presented of the closely related subject of accuracy in explicit numerical integration.

The order of accuracy of any approximate integration formula is the power $n$ for which the truncation error is proportional to $(\Delta t)^n$, where $\Delta t$ is the discretization step size. As such, the greater the order of accuracy of the integration formula, the greater is the reduction in truncation error as the step size is reduced. The derivation of the local truncation error is discussed in detail in [6,7]. For example, the local truncation errors for the well-known trapezoid and Simpson's explicit integration rules are proportional to $(\Delta t)^3$ and $(\Delta t)^5$, respectively. The power of $\Delta t$ of the Simpson's rule error is higher than that of the error for the trapezoid rule. Therefore, the Simpson's integration polynomial is of higher order than the trapezoid polynomial. As the $\Delta t$ is reduced (thereby increasing the number of subintervals or segments for the total time discretization), the Simpson's polynomial approximation improves by $(\Delta t)^5$, whereas the trapezoid polynomial approximation only improves cubically. However, as stated by Conte and de Boor [6], there is a lower limit to which the step size can be reduced before the accumulation of rounding error becomes a problem.

As described by Herman and Conway [3,4], for the integration of ordinary differential equations, the discretization error of concern is the global truncation error that results from using a discretization scheme. This global truncation error is a consequence of discretizing a function using a polynomial over an interval that has been subdivided. The sum of this discretization error over the total length of the integration produces the global truncation error. Because the step size and the number of local truncation-error values to be summed are both proportional to the number of subintervals or segments, the global truncation error is shown to be of order one less than the local truncation error [3,4]. This observation yields an order of accuracy of four for the Simpson's integration rule. For the fifth-degree Gauss–Lobatto integration rule, the order of accuracy is eight [3]. Consequently, the order of accuracy of the two-timescale architecture is also eight due to its use of the fifth-degree Gauss–Lobatto polynomial for the approximation of each state within each segment. Moreover, because the order of accuracy of any method is driven by the selection of the $\Delta t$ or step size, the *largest* length used for the subintervals or segments will dictate the accuracy achieved. Thus, for the two-timescale architecture, the width of the low-frequency segment (which is either twice or four-times the width of the high-frequency segment) will determine the overall accuracy achieved by this method.

Note that the driving factor in the selection of the segment width for the high-frequency states is the need to appropriately model their dynamical behavior; hence, this time interval needs to be small in comparison with the period of variation of the high-frequency states. For a problem for which little a priori information about the character of the solution is available, trial and error may be required to determine what an appropriately small $\Delta t$ should be. Herman and Conway [3,4] suggest as a rule of thumb (provided the segment width is chosen appropriately for the problem), that the number of digits of accuracy in the solution will equal the order of accuracy of the implicit integration method, which is eight for the fifth-degree Gauss–Lobatto rule used in this research.

## IV. Results

A sample problem (the lunar ascent problem of Bryson and Ho [8]) is used to validate the two-timescale collocation architecture. The two-dimensional lunar ascent problem has four state variables ($X$ and $Y$ Cartesian coordinates, $U$ and $V$ velocities) and one control variable, a thrust pointing angle $\theta$. The state equations are

$$\dot{X} = U \tag{8}$$

$$\dot{Y} = V \tag{9}$$

$$\dot{U} = a^* \cos\theta \tag{10}$$

$$\dot{V} = a^* \sin\theta - g \tag{11}$$

where $a$ is a constant thrust acceleration and $g$ is gravity. The objective is to minimize the time to achieve a desired orbit altitude and velocity. The solution history using the standard collocation scheme is first provided. Then, a comparison is made with the results obtained using both the two-to-one and four-to-one two-timescale architectures.

Figures 4–6 show the position, velocity, and control histories using the standard collocation scheme.

Two cases are used to illustrate the two-timescale discretization architecture. Case 1 exercises the two-to-one two-timescale architecture. In this case, the $X$ and $Y$ state variables use a standard discretization, whereas $U$ and $V$ state variables use a finer discretization. Specifically, two segments are employed for states $U$ and $V$ for every one segment for states $X$ and $Y$. Note that for this test case $U$ and $V$ are chosen arbitrarily to act as the high-frequency states for the two-timescale discretization for illustrative purposes only; in fact, there is no significant difference in the rate at which $U$ and $V$ change in comparison with the other states. The intention is simply to show how well the solution of the problem can be accomplished using two timescales rather than just one. For an equivalent comparison to the two-timescale solution, the standard one-timescale collocation architecture uses 20 segments. Table 1 summarizes the results.

As seen, the number of NLP parameters is significantly reduced from 246 for the standard one-timescale collocation architecture to 166 for the two-to-one two-timescale discretization collocation architecture. The number of algebraic constraint equations is reduced as well. Consequently, the resulting number of iterations required for

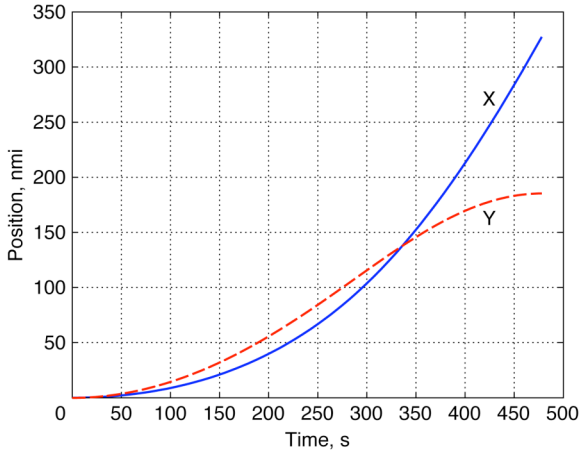


**Fig. 4 Time history of $X$ and $Y$ position.**

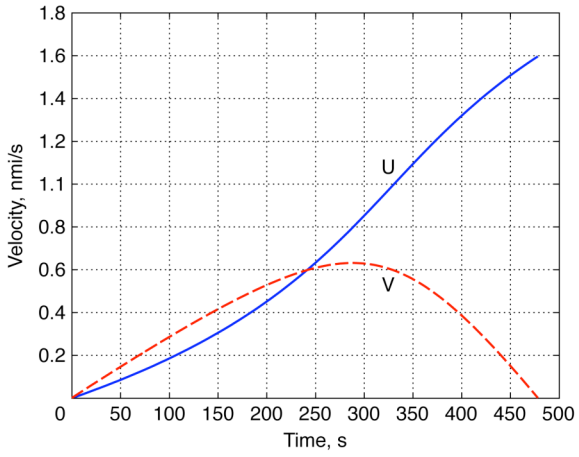


**Fig. 5 Time history of *U* and *V* velocities.**

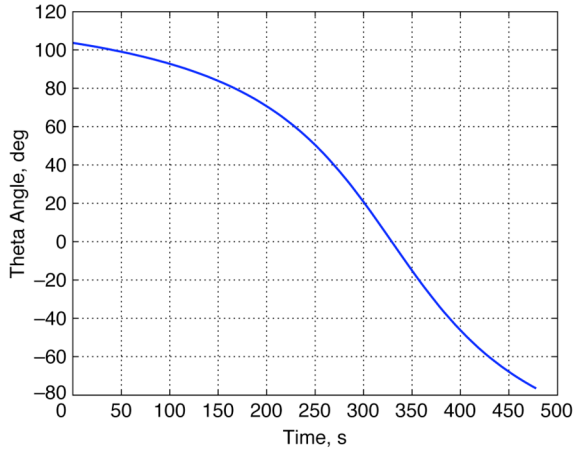


**Fig. 6 Time history of theta control angle.**

solution convergence is significantly reduced from 104 to 58, respectively. Furthermore, nearly identical results are obtained with the two-timescale collocation architecture. The maximum difference observed in any of the state variables is less than 0.07% between the two methods. The difference in the control variable is only slightly higher (less than 0.5%). These results illustrate that a significant reduction in the number of NLP parameters and iterations required for solution convergence can be achieved without sacrificing solution accuracy.

Case 2 exercises the four-to-one two-timescale architecture. In this case, the $X$, $Y$, and $U$ state variables use a standard discretization, whereas the $V$ state variable uses a finer discretization. Specifically, four segments are employed for the state $V$ for every one segment for states $X$, $Y$, and $U$. Note that for this test case $V$ is chosen arbitrarily to act as the high-frequency state for the two-timescale discretization for illustrative purposes only. Again, there is no significant difference in the rate at which $V$ changes in comparison with the other states. The intention is simply to show how well the solution of the problem can be accomplished using two timescales rather than just one. For an equivalent comparison with the two-timescale solution, the standard one-timescale collocation architecture uses 40 segments. Table 2 summarizes the results.

As seen, the number of NLP parameters is significantly reduced from 486 for the standard one-timescale collocation architecture to 186 for the two-to-one two-timescale discretization collocation architecture. The number of algebraic constraint equations is reduced considerably as well from 320 to 140. Consequently, the resulting number of iterations required for solution convergence is significantly reduced from 199 to 64. Furthermore, nearly identical results are obtained with the two-timescale collocation architecture. The maximum difference observed in any of the state variables is less than 0.08% between the two methods. The maximum difference observed in the control variable is only slightly higher (less than 0.3%). These results again illustrate that a significant reduction in the number of NLP parameters, and hence solution convergence, can be achieved without sacrificing solution accuracy. The obvious application of this two-timescale architecture is to a problem that exhibits two distinct classes of states with different rates of variation (e.g., the coupled translational/rotational dynamics of an aircraft or spacecraft). Such a problem has been successfully solved and is the subject of [5,9], which provide a comparison for a more complex problem, specifically a six-degree-of-freedom trajectory optimization of the approach and landing solution for a winged vehicle.

## V. Conclusions

The development of a two-timescale discretization scheme for collocation is presented. This scheme allows a coarser discretization to be used for slowly varying state variables and a second finer discretization to be used for state variables having higher-frequency dynamics. As such, the finer discretization timescale can be tailored to only those state variables that have higher-frequency dynamics in the governing equations, whereas the rest of the state variables can use a much larger discretization timescale. Two two-timescale architecture discretization schemes are described as follows: 1) using two segments to represent the higher-frequency state variables for every one segment of the low-frequency state variables and 2) using four segments to represent the higher-frequency state variables for every one segment of the low-frequency state variables.

Results from two test cases are presented to validate the two-timescale collocation architecture and then compared with the solution obtained from the conventional single-timescale collocation method. Very good agreement between the two methods is obtained with a maximum difference of less than 0.08% for the state variables and less than 0.5% for the control variable. Thus, the two-timescale architecture accurately reproduces results compared with using the conventional single-timescale collocation. The number of NLP parameters and iterations required for convergence for the two-timescale scheme can be reduced by approximately two-thirds. Consequently, a significant reduction in the number of NLP parameters, and hence improved solution convergence can be

**Table 1 Comparison of the standard and two-to-one two-timescale architectures**

| | Number of segments | Number of NLP parameters | Number of defect equations | Number of iterations | Maximum difference |
|---|---|---|---|---|---|
| Standard | 20 | 246 | 160 | 104 | NA |
| Case 1 | 10/20 | 166 | 120 | 58 | 0.07%, 0.5% |

**Table 2 Comparison of the standard and four-to-one two-timescale architectures**

| | Number of segments | Number of NLP parameters | Number of defect equations | Number of iterations | Maximum difference |
|---|---|---|---|---|---|
| Standard | 40 | 486 | 320 | 199 | NA |
| Case 2 | 10/40 | 186 | 140 | 64 | 0.08%, 0.3% |

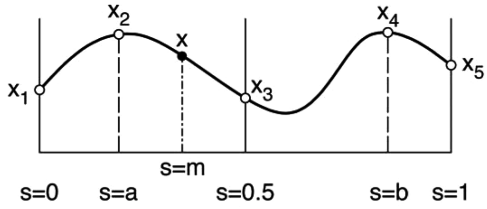


**Fig. A1 Quartic polynomial interpolation formulation.**

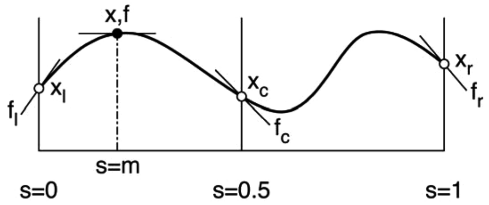


**Fig. B1 Quintic polynomial interpolation formulation.**

achieved without sacrificing solution accuracy. The obvious application of this two-timescale architecture is to a problem that exhibits two distinct classes of states with different rates of variation (e.g., the coupled transitional–rotational dynamics of an aircraft or spacecraft). Such a problem has been successfully solved and is the subject of a companion paper.

## Appendix A: Quartic Polynomial Interpolation Function Formulation

The quartic polynomial interpolation function used to obtain an interpolated value is shown in Fig. A1. Given five values of $x_i$ at their associated locations $s$, the interpolated value for $x$ at any $s = m$ can be obtained, where $s$ is defined between 0 and 1. The parameter $x_1$ is defined at $s = 0$, $x_3$ at $s = 0.5$, $x_5$ at $s = 1$. The parameter $x_2$ is defined at any $s = a$ between 0 and 0.5 and $x_4$ is defined at any $s = b$ between 0.5 and 1.

Given $x_1$ through $x_5$, $a$, and $b$, the interpolated value $x$ at any $s$ is given by

$$x = c_0 + c_1 s + c_2 s^2 + c_3 s^3 + c_4 s^4$$

where

$$c_0 = x_1$$

$$\begin{aligned} c_1 = {} & \{(b^2 - 3b^3 + 2b^4)(x_1 - x_2) \\ & + a^3[3x_1(1 - 5b^2 + 4b^4) + 16x_3(b^2 - b^4) \\ & + x_5(4b^4 - b^2) - 3x_4] - 2a^4[x_1(1 - 7b^2 + 6b^3) \\ & + 8x_3(b^2 - b^3) + x_5(2b^3 - b^2) - x_4] + a^2[x_1(15b^3 - 1 - 14b^4) \\ & + 16x_3(b^4 - b^3) + x_5(b^3 - 2b^4) + x_4]\}/\{(a - 3a^2 + 2a^3) \\ & \times (a - b)(b - 3b^2 + 2b^3)\} \end{aligned}$$

$$\begin{aligned} c_2 = {} & \{(7b^3 - b - 6b^4)(x_1 - x_2) + 2a^4[x_1(3 - 7b + 4b^3) \\ & + 8x_3(b - b^3) + x_5(4b^3 - b) - 3x_4] + a[x_1(1 - 15b^3 + 14b^4) \\ & + 16x_3(b^3 - b^4) + x_5(2b^4 - b^3) - x_4] + a^3[x_1(15b - 7 - 8b^4) \\ & + 16x_3(b^4 - b) + x_5(b - 8b^4) + 7x_4]\}/\{(a - 3a^2 + 2a^3) \\ & \times (a - b)(b - 3b^2 + 2b^3)\} \end{aligned}$$

$$\begin{aligned} c_3 = {} & \{(3b - 7b^2 + 4b^4)(x_1 - x_2) + a^2[x_1(7 - 15b + 8b^4) \\ & + 16x_3(b - b^4) + x_5(8b^4 - b) - 7x_4] - 4a^4[x_1(1 - 3b + 2b^2) \\ & + 4x_3(b - b^2) + x_5(2b^2 - b) - x_4] + a[3x_1(5b^2 - 1 - 4b^4) \\ & + 16x_3(b^4 - b^2) + x_5(b^2 - 4b^4) + 3x_4]\}/\{(a - 3a^2 + 2a^3) \\ & \times (a - b)(b - 3b^2 + 2b^3)\} \end{aligned}$$

$$\begin{aligned} c_4 = {} & \{(6b^2 - 2b - 4b^3)(x_1 - x_2) + 4a^3[x_1(1 - 3b + 2b^2) \\ & + 4x_3(b - b^2) + x_5(2b^2 - b) - x_4] + 2a[x_1(1 - 7b^2 + 6b^3) \\ & + 8x_3(b^2 - b^3) + x_5(2b^3 - b^2) - x_4] + 2a^2[x_1(7b - 3 - 4b^3) \\ & + 8x_3(b^3 - b) + x_5(b - 4b^3) + 3x_4]\}/\{(a - 3a^2 + 2a^3) \\ & \times (a - b)(b - 3b^2 + 2b^3)\} \end{aligned}$$

## Appendix B: Quintic Polynomial Interpolation Function Formulation

The quintic polynomial interpolation function used to obtain an interpolated value is shown in Fig. B1. Given $x_1$, $x_c$, and $x_r$ and their slopes $f_1$, $f_c$, and $f_r$ at their associated locations $s$, the interpolated value for $x$ and $f$ at any $s = m$ can be obtained, where $s$ is defined between 0 and 1. The variable $t$ is the length of the segment in seconds. The parameters $x_1$ and $f_1$ are defined at $s = 0$; $x_c$ and $f_c$ at $s = 0.5$; $x_r$ and $f_r$ at $s = 1$.

$$\begin{aligned} x = {} & (1 - 23s^2 + 66s^3 - 68s^4 + 24s^5)x_l \\ & + (16s^2 - 32s^3 + 16s^4)x_c + (7s^2 - 34s^3 + 52s^4 - 24s^5)x_r \\ & + t(s - 6s^2 + 13s^3 - 12s^4 + 4s^5)f_l \\ & + t(-8s^2 + 32s^3 - 40s^4 + 16s^5)f_c \\ & + t(-s^2 + 5s^3 - 8s^4 + 4s^5)f_r \end{aligned}$$

$$\begin{aligned} f = {} & (-46s + 198s^2 - 272s^3 + 120s^4)x_l/t \\ & + (32s - 96s^2 + 64s^3)x_c/t \\ & + (14s - 120s^2 + 208s^3 - 120s^4)x_r/t \\ & + (1 - 12s + 39s^2 - 48s^3 + 20s^4)f_l \\ & + (-16s + 96s^2 - 160s^3 + 80s^4)f_c \\ & + (-2s + 15s^2 - 32s^3 + 20s^4)f_r \end{aligned}$$

## References

[1] Powell, R. W., "Six-Degree-of-Freedom Guidance and Control-Entry Analysis of the HL-20," *Journal of Spacecraft and Rockets*, Vol. 30, No. 5, 1993, pp. 537–542. doi:10.2514/3.25563

[2] Hargraves, C. R., and Paris, S. W., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 4, 1987, pp. 338–342. doi:10.2514/3.20223

[3] Herman, A. L., and Conway, B. A., "Direct Optimization Using Collocation Based on High-Order Gauss–Lobatto Quadrature Rules," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 3, 1996, pp. 592–599. doi:10.2514/3.21662

[4] Herman, A. L., "Improved Collocation Methods with Applications to Direct Trajectory Optimization," Ph.D. Dissertation, Aerospace Engineering Dept., Univ. of Illinois, Urbana, IL, 1995.

[5] Desai, P. N., "Improved Collocation Methods with Applications to Six-Degree-of-Freedom Trajectory Optimization," Ph.D. Dissertation, Aerospace Engineering Dept., Univ. of Illinois, Urbana, IL, 2005.

[6] Conte, S. D., and de Boor, C., *Elementary Numerical Analysis: An Algorithmic Approach*, McGraw–Hill, New York, 1980, Chap. 8.

[7] Hammerlin, G., and Hoffman, K. H., *Numerical Mathematics*, translated by L. Schumaker, Springer–Verlag, New York, 1991, Chap. 7.
[8] Bryson, A. E., and Ho, Y. C., *Applied Optimal Control*, Hemisphere, New York, 1975, p. 59.
[9] Desai, P. N., and Conway, B. A., "Six-Degree-of-Freedom Trajectory Optimization Utilizing a Two-Timescale Collocation Architecture," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, 2008, pp. 1308–1315.
doi:10.2514/1.34020