JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 15, No. 4, July-August 1992

Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming

Paul J. Enright* and Bruce A. Conway†
University of Illinois at Urbana-Champaign, Urbana, Illinois 61801

A recently developed method for solving optimal trajectory problems uses a piecewise-polynomial representation of the state and control variables, enforces the equations of motion via a collocation procedure, and thus approximates the original calculus-of-variations problem with a nonlinear programming problem, which is solved numerically. This paper identifies this method as being of a general class of direct transcription methods and proceeds to investigate the relationship between the original optimal control problem and the approximating nonlinear programming problem, by comparing the optimal control necessary conditions with the optimality conditions for the discretized problem. Attention is focused on the Lagrange multipliers of the nonlinear programming problem, which are shown to be discrete approximations to the adjoint variables of the optimal control problem. This relationship is exploited to test the adequacy of the discretization and to verify optimality of assumed control structures. The discretized adjoint equation of the collocation method is found to have deficient accuracy, and an alternate scheme that discretizes the equations of motion using an explicit Runge-Kutta parallel-shooting approach is developed. Both methods are applied to finite-thrust spacecraft trajectory problems, including a low-thrust escape spiral, a three-burn rendezvous, and a low-thrust transfer to the moon.

Introduction

ALTHOUGH optimal trajectory problems usually originate as function-space optimization problems, they are almost exclusively solved by finite-dimensional approximations. Essentially, two approaches have emerged over the past 35 years. In the first, the necessary conditions for optimality are derived using calculus-of-variations techniques. The resulting two-point boundary-value problem (TPBVP) is solved numerically, where some sort of discretization is introduced. We refer to this approach as "indirect." The second approach is to discretize the variational form of the original problem, i.e., to approximate the optimal control problem with a discrete optimization problem, which is then solved numerically. We refer to this approach as "direct."

Many methods have been developed to solve the TPBVP that results from the indirect approach to the optimal trajectory problem. The list includes neighboring extremal methods, the method of gradients,[1] quasilinearization,[2] finite difference methods,[3] and collocation techniques.[4] The main drawbacks of the indirect methods are the requirement for an initial guess for the adjoint variables of the TPBVP, the sensitivity of the Euler-Lagrange equations, and the frequent occurrence of discontinuities in the optimal control.

A variety of direct methods have been proposed and implemented. They are best categorized by their handling of the discretization of the equations of motion, which appear as function-space constraints in the original optimal control problem. The most straightforward approach to discretizing the equations of motion is to use a numerical integration procedure. The integration procedure may be a single step or multistep, explicit or implicit. The so-called "mathematical programming" approach developed in the sixties and early seventies[5-7] usually employed an explicit single-step integration formula to convert the equation-of-motion constraint into a set of discrete algebraic constraints (one set for each integration step). The problem becomes a "multistage" discrete optimization problem.[8] In addition, the interior state variables were usually eliminated by propagating the integration scheme forward, using the initial state and the control sequence. This reduced the problem to a discretized-control-space optimization problem, which is relatively unconstrained. The alternative to this state-elimination procedure is to leave the interior states as variables, retaining the constraints that the integration formula be satisfied at each step. The result is a programming problem that is fairly large and includes a large number of constraints. This approach was referred to as "direct transcription" by Canon et al.[5] For nonlinear problems, the difficulty associated with solving the resulting nonlinear programming (NLP) problem made this approach unattractive for many years. However, other direct methods were and continue to be used. One popular approach is to assume a simple parameterization of the control history (e.g., polynomials) and integrate the equations of motion forward (sometimes called "direct shooting"). Another approach is to represent the state and control by polynomials, using an integral penalty function scheme to convert the problem into a sequence of unconstrained discrete optimization problems.[9-11]

In 1987, Hargraves and Paris[12] reintroduced the direct transcription method in a new form, discretizing the equations of motion using a collocation scheme. The collocation scheme used was one that Dickmanns and Well[13] employed to solve the TPBVP of the indirect method, which was a particular implementation of a general collocation method for TPBVPs presented by Russell and Shampine.[4] Note that Dickmanns and Well used the collocation scheme to reduce the indirect TPBVP (in the state and adjoint variables) to a set of nonlinear equations, which they solved using a modified Newton's method. Hargraves and Paris used the collocation scheme to reduce the equations of motion (state only) to a set of nonlinear constraints for the direct-approach NLP problem, which they solved using a sequential quadratic programming (SQP) technique.[14] A direct collocation method had been used by

Presented as Paper 90-2963 at the AIAA/AAS Astrodynamics Conference, Portland, OR, Aug. 20-22, 1990; received Oct. 12, 1990; revision received June 12, 1991; accepted for publication July 3, 1991.

*Graduate Research Assistant, Department of Aeronautical and Astronautical Engineering; currently Member of the Technical Staff, Jet Propulsion Laboratory, MS 198-326, Pasadena, CA 91109. Member AIAA.

†Associate Professor, Department of Aeronautical and Astronautical Engineering, 101 Transportation Building, 104 South Mathews Avenue. Associate Fellow AIAA.

Neuman and Sen[15] for linear-quadratic problems, which requires solution of a quadratic programming problem that is much less difficult than the nonlinear problem. Neuman and Sen,[15] Renes,[16] and Kraft[17] all proposed methods similar to that of Hargraves and Paris for nonlinear problems, but none gave numerical results.

It can be shown that the collocation integration methods are equivalent to implicit Runge-Kutta methods,[18,19] and thus the direct collocation method falls into the category of direct transcription methods described earlier. However, this relationship is not obvious. In addition, the NLP technology required to solve complicated nonlinear optimal control problems using this type of method has only recently become available. These factors have obscured the relationship between the direct collocation method and the older mathematical programming methods.

In the following, we reinterpret the direct collocation method as a direct transcription method using an implicit integration formula. We also implement a new method that is essentially an adaptation of the parallel-shooting method for TPBVPs[20,21] for the discretization of the equations of motion in the direct transcription approach to the optimal trajectory problem.

The parallel-shooting method uses an explicit Runge-Kutta integration formula that has been used previously in conjunction with the state-elimination method described earlier. Abadie,[22] Hager,[23] and Kelley and Sachs[24] compared the necessary conditions for the original optimal control problem to the necessary conditions for the discretized NLP problem to obtain convergence information. This analysis carries over directly to our parallel-shooting implementation. For the collocation method, we present a similar analysis, concluding that the method suffers from a deficient accuracy in the discretization of the adjoint equation that is imposed by the optimality conditions.

Transcription Using Hermite Interpolation and Simpson Quadrature

Consider the discretization of the differential equation

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x) \qquad \text{for } t \in [0, t_f] \tag{1}$$

Introduce the partition $[t_0, t_1, \ldots, t_N]$, with $t_0 = 0$ and $t_N = t_f$, and $t_0 < t_1 < \ldots < t_N$, and let $h_i = t_i - t_{i-1}$ for $i = 1, \ldots, N$. The mesh points t_i are referred to as "nodes," whereas the intervals $[t_{i-1}, t_i]$ are referred to as "segments" (after Hargraves and Paris). The $x(t)$ is to be approximated by values at the nodes, $x_i \cong x(t_i)$ for $i = 0, \ldots, N$. The conditions that approximate the differential equation (1) are derived as follows. Obtain estimates of $x(t)$ at the segment centers $y_i \cong x(t_{i-1} + h_i/2)$ using Hermite-cubic interpolation[25] from the adjacent nodes, derivatives being provided by evaluation of the differential equation at the nodes,

$$y_i = \frac{1}{2}(x_{i-1} + x_i) + \frac{h_i}{8}[f(x_{i-1} - f(x_i)] \tag{2}$$

Next evaluate the derivative at the interpolated center value $f(y_i)$, and integrate across the segment using Simpson's quadrature rule. We require

$$x_i = x_{i-1} + \frac{h_i}{6}[f(x_{i-1}) + 4f(y_i) + f(x_i)] \tag{3}$$

or, alternatively, we require that the "Hermite-Simpson defects,"

$$\Delta_i = x_{i-1} - x_i + \frac{h_i}{6}[f(x_{i-1}) + 4f(y_i) + f(x_i)] \tag{4}$$

be equal to zero for $i = 1, \ldots, N$. Note that this defect definition (4) is simply the collocation defect given by Hargraves and Paris[12] multiplied by the factor $2h_i/3$. (We have ignored the control process for now.) This pairing of an interpolant with a quadrature rule to generate an integration formula has been studied extensively. Butcher[26,27] and others[19,21] demonstrate that these methods are equivalent to implicit Runge-Kutta methods. The relationship between these methods and collocation methods has been studied by Wright,[18] Hulme,[28] Weiss,[19] and Ascher et al.[21] The Hermite-Simpson combination used before dates back at least to Kunz (1957).[29] The equivalence of this particular method with an implicit Runge-Kutta method was pointed out by Butcher,[26] and the collocation interpretation was provided by Weiss,[19] who also demonstrated the efficiency of this method for discretizing boundary-value problems.

We now consider the transcription (i.e., discretization) of an optimal trajectory problem using the Hermite-Simpson defects. For simplicity, let us consider a problem in the Mayer form with an autonomous dynamical system.

Determine the control $\boldsymbol{u}(t)$, the corresponding trajectory $\boldsymbol{x}(t)$ [$\boldsymbol{x}(0)$ given] and the final time t_f, to minimize the cost function $\varphi[\boldsymbol{x}(t_f), t_f]$, subject to the equation-of-motion constraint,

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \qquad \text{for } t \in [0, t_f] \tag{5}$$

with $\boldsymbol{x} \in \mathbb{R}^n$, and $\boldsymbol{u} \in \mathbb{R}^m$, and the terminal constraints,

$$\boldsymbol{\psi}[\boldsymbol{x}(t_f), t_f] = \boldsymbol{0} \tag{6}$$

with $\boldsymbol{\psi} \in \mathbb{R}^q$. The classical approach to this problem[8] is to adjoin the constraints to the cost in the following fashion to form the augmented-cost functional:

$$L = \varphi + \nu^T \boldsymbol{\psi} + \int_0^{t_f} \lambda^T \left[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) - \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} \right] \mathrm{d}t \tag{7}$$

with the multipliers $\nu \in \mathbb{R}^q$ and the adjoint variables $\lambda(t) \in \mathbb{R}^n$. Necessary conditions for optimality are derived by requiring that L be stationary with respect to variations in $\boldsymbol{x}(t)$, $\boldsymbol{u}(t)$, and t_f.

The well-known conditions are

$$\frac{\mathrm{d}\lambda}{\mathrm{d}t} = -F(\boldsymbol{x}, \boldsymbol{u})^T \lambda, \qquad \text{with } F(\boldsymbol{x}, \boldsymbol{u}) = \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} \tag{8}$$

with boundary condition,

$$\lambda(t_f)^T = \frac{\partial \Phi}{\partial \boldsymbol{x}(t_f)}, \qquad \text{with } \Phi[\boldsymbol{x}(t_f), t_f] = \varphi + \nu^T \boldsymbol{\psi} \tag{9}$$

the control optimality condition,

$$\lambda^T G(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{0}, \qquad \text{with } G(\boldsymbol{x}, \boldsymbol{u}) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} \tag{10}$$

(or the minimum principle if applicable) and the transversality condition,

$$\frac{\partial \Phi}{\partial t_f} + \lambda(t_f)^T \boldsymbol{f}[\boldsymbol{x}(t_f), \boldsymbol{u}(t_f)] = 0 \tag{11}$$

Introduce the uniform partition $[t_0, t_1, \ldots, t_N]$, with $t_0 = 0$ and $t_N = t_f$. Let $\boldsymbol{x}_i$ and $\boldsymbol{u}_i$ be the state and control vectors at the ith node, and assume the nodes are (evenly) spaced at $t_i - t_{i-1} = h = t_f/N$. The discrete-approximate NLP problem using Hermite-Simpson transcription can now be stated as follows.

Determine the control sequence $[\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_N]$, the corresponding trajectory $[\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]$ ($\boldsymbol{x}_0$ given), and the final time t_f, to minimize the cost function $\varphi[x_N,t_f]$ subject to the equation-of-motion constraints,

$$\Delta_i = 0, \qquad i = 1,\ldots,N \tag{12}$$

and the terminal constraints,

$$\psi[x_N,t_f] = 0 \tag{13}$$

where

$$\Delta_i = x_{i-1} - x_i + \frac{h}{6}[f(x_{i-1},u_{i-1}) + 4f(y_i,v_i) + f(x_i,u_i)] \tag{14}$$

$$y_i = \frac{1}{2}(x_{i-1} + x_i) + \frac{h}{8}[f(x_{i-1},u_{i-1}) - f(x_i,u_i)] \tag{15}$$

$$v_i = \frac{1}{2}(u_{i-1} + u_i) \tag{16}$$

The conversion performed is identical to that given by Hargraves and Paris,[12] except for the slightly different definition of the "defects," Eq. (14). Note the linear interpolation for the control Eq. (16).

As in the continuous case, we adjoin the constraints to the cost to obtain (the Lagrangian):

$$L = \varphi + \nu^T\psi + \sum_{i=1}^{N} \lambda_i^T\Delta_i \tag{17}$$

The Karush-Kuhn-Tucker (KKT) necessary conditions for optimality[14] are obtained by differentiating L with respect to the state and control variables at the nodes, and the final time, and setting the derivatives to zero. We examine the interior state variables $(x_1,\ldots,x_{N-1})$ first:

$$\frac{\partial L}{\partial x_k} = \sum_{i=1}^{N} \lambda_i^T \frac{\partial \Delta_i}{\partial x_k} = 0 \tag{18}$$

But x_k affects the adjacent defects Δ_k and Δ_{k+1} only, so

$$\lambda_k^T\frac{\partial \Delta_k}{\partial x_k} + \lambda_{k+1}^T\frac{\partial \Delta_{k+1}}{\partial x_k} = 0 \tag{19}$$

for $k = 1,\ldots,N-1$. From the defect definition, Eqs. (14) and (15), we can obtain

$$\frac{\partial \Delta_k}{\partial x_k} = -I + \frac{h}{3}\left\{\left[\frac{1}{2}I - \frac{h}{4}F(y_k,v_k)\right]F(x_k,u_k) + F(y_k,v_k)\right\} \tag{20}$$

and

$$\frac{\partial \Delta_{k+1}}{\partial x_k} = I + \frac{h}{3}\left\{\left[\frac{1}{2}I + \frac{h}{4}F(y_{k+1},v_{k+1})\right]F(x_k,u_k) + F(y_{k+1},v_{k+1})\right\} \tag{21}$$

where I is the $n \times n$ identity matrix. Substituting into Eq. (19) and rearranging gives

$$\lambda_k + \lambda_{k+1} + \frac{h}{3}[-F(y_k,v_k)^T\lambda_k - F(x_k,u_k)^T\eta_k - F(y_{k+1},v_{k+1})^T\lambda_{k+1}] = 0 \tag{22}$$

where we define

$$\eta_k = \frac{1}{2}(\lambda_k + \lambda_{k+1}) + \frac{h}{4}[-F(y_k,v_k)^T\lambda_k + F(y_{k+1},v_{k+1})^T\lambda_{k+1}] \tag{23}$$

for $k = 1,\ldots,N-1$.

This is clearly a discretization of the adjoint differential equation (8) that is imposed by the discrete optimality conditions, and we shall refer to Eqs. (22) and (23) collectively as the "discrete adjoint equation." The discrete multipliers approximate the adjoints of the continuous problem at the segment centers, $\lambda_k \cong \lambda_k(t_{k-1} + h/2)$. Abadie[22] and Canon et al.[5] derived similar conditions for other (explicit) integration methods used in the state-elimination approach. We have written the discrete adjoint equation of the Hermite-Simpson method as an interpolation/quadrature pair, but note that the interpolation (23) is not Hermite cubic, and the quadrature rule (22) is not Simpson. The Hermite-Simpson integration method has an order h^5 local truncation error.[21] But the adjoint integration has an order h^3 error.[30] The effects of this accuracy deficiency are discussed next.

Preceding as before, differentiation with respect to x_N provides the terminal boundary condition for Eq. (22):

$$\eta_N^T = \frac{\partial \Phi}{\partial x_N}, \qquad \text{with} \quad \Phi[x_N,t_f] = \varphi + \nu^T\psi \tag{24}$$

where

$$\eta_N = \lambda_N + \frac{h}{6}[-2F(y_N,v_N)^T\lambda_N - F(x_N,u_N)^T\eta_{N'}] \tag{25}$$

and

$$\eta_N' = \lambda_N + \frac{h}{2}[-F(y_N,v_N)^T\lambda_N] \tag{26}$$

which corresponds to the continuous condition, Eq. (9).

Next consider the control variables. Control at an interior node affects only the defects of the adjacent segments, so

$$\lambda_k^T\frac{\partial \Delta_k}{\partial u_k} + \lambda_{k+1}^T\frac{\partial \Delta_{k+1}}{\partial u_k} = 0 \tag{27}$$

for $k = 1,\ldots,N-1$. The derivatives are

$$\frac{\partial \Delta_k}{\partial u_k} = \frac{h}{3}\left\{\left[\frac{1}{2}I - \frac{h}{4}F(y_k,v_k)\right]G(x_k,u_k) + G(y_k,v_k)\right\} \tag{28}$$

and

$$\frac{\partial \Delta_{k+1}}{\partial u_k} = \frac{h}{3}\left\{\left[\frac{1}{2}I + \frac{h}{4}F(y_{k+1},v_{k+1})\right]G(x_k,u_k) + G(y_{k+1},v_{k+1})\right\} \tag{29}$$

Rearranging, and using Eq. (23), the condition becomes

$$\frac{h}{3}[G(y_k,v_k)^T\lambda_k + G(x_k,u_k)^T\eta_k + G(y_{k+1},v_{k+1})^T\lambda_{k+1}] = 0 \tag{30}$$

This is a discretized version of the continuous control condition (10). Note that the condition is averaged over the interval $[t_k - h/2,\ t_k + h/2]$ in some sense. The derivatives with respect to u_0 and u_N yield similar conditions, and the derivative with respect to final time yields a condition analogous to the continuous transversality condition (11).[30]

The deficient accuracy of the discrete adjoint equation has not prevented the accurate solution of optimal trajectory problems using the Hermite-Simpson procedure.[12,31] Hager[23] discusses similar deficiencies in the application of certain explicit integration procedures and concludes that they degrade the accuracy of the optimal control approximation. We have observed numerically that the adjoint approximations that result from the Hermite-Simpson method are indeed inferior to the state approximations, but a deleterious effect on the optimal control approximations has only been observed in conjunction with the "center controls" modification discussed later.

The linear treatment of the control presented before seems inefficient, especially in light of the cubic state representation. Indeed, Hargraves and Paris[32] used a cubic spline control representation for their trajectory optimization routine OTIS. In our version of cubic-spline controls, we introduce new variables w_i at each node, which correspond to normalized control derivatives. Hermite interpolation is used to determine the control at the center:

$$v_i = \frac{1}{2}(u_{i-1} + u_i) + \frac{1}{8}(w_{i-1} - w_i) \tag{31}$$

In addition, the following spline conditions (continuity of second derivative) are imposed at each interior node[25]:

$$3u_{i-1} - 3u_{i+1} + w_{i-1} + 4w_i + w_{i+1} = 0 \tag{32}$$

for $i = 1, \ldots, N-1$. Writing the KKT conditions as before, the control conditions are

$$\frac{h}{3}[G(y_k,v_k)^T\lambda_k + G(x_k,u_k)^T\eta_k + G(y_{k+1},v_{k+1})^T\lambda_{k+1}]$$
$$- 3\mu_{k-1} + 3\mu_{k+1} = 0 \tag{33}$$

and

$$\frac{h}{12}[-G(y_k,v_k)^T\lambda_k + G(y_{k+1},v_{k+1})^T\lambda_{k+1}]$$
$$+ \mu_{k-1} + 4\mu_k + \mu_{k+1} = 0 \tag{34}$$

where the μ_k are the multipliers corresponding to the spline constraints, Eq. (32). The method used in OTIS is similar, but the control interpolation involves the segment time, and this makes the spline conditions nonlinear in the discrete variables. In our normalized version, the spline constraints are linear, but the mesh must be uniform. This method is applied to some problems next.

A more straightforward approach is to simply add new variables v_i that correspond to the control at the segment centers, independent of the adjacent nodal values, $v_i \cong u(t_{i-1} + h/2)$ for $i = 1, \ldots, N$. This approach has been implemented by the authors and also by Betts and Huffman,[33] and we refer to it as the "center control" method. The control conditions become

$$\frac{h}{3}G(x_k,u_k)^T\eta_k = 0 \tag{35}$$

$$\frac{2h}{3}G(y_k,v_k)]^T\lambda_k = 0 \tag{36}$$

Note that this is a pointwise discretization of the continuous control condition. This feature couples with the inaccurate discrete adjoint interpolation equation and leads to an undesirable feature of the control approximation. *This is simply shown;* using the definition of η in Eq. (23) for $k-1$ and for k, and adding, we obtain

$$\eta_{k-1} + \eta_k = \lambda_k + \frac{1}{2}(\lambda_{k-1} + \lambda_{k+1})$$
$$+ \frac{h}{4}[-F(y_{k-1},v_{k-1})^T\lambda_{k-1} + F(y_{k+1},v_{k+1})^T\lambda_{k+1}] \tag{37}$$

Recognizing the Hermite interpolation over the $2h$ interval,

$$\lambda_k \cong \frac{1}{2}(\lambda_{k-1} + \lambda_{k+1}) + \frac{2h}{8}[-F(y_{k-1},v_{k-1})^T\lambda_{k-1}$$
$$+ F(y_{k+1},v_{k+1})^T\lambda_{k+1}] \tag{38}$$

Equation (37) becomes

$$\lambda_k \cong \frac{1}{2}(\eta_{k-1} + \eta_k) \tag{39}$$

which shows that the discrete adjoint is forced to be linear across the segment. For many optimal trajectory problems, the optimal thrust direction is the negative of the direction of the velocity adjoint (the "primer vector"). For transcriptions of these problems, the control conditions, Eqs. (35) and (36), force the control approximation to follow the linear behavior of the discrete adjoint across a segment. Numerical experimentation verifies that the center control approach leads to this linear control behavior and thus does not provide an improvement on the linear control method for this class of problems.

The cubic spline control approach avoids this problem by enforcing the spline constraints, Eq. (32), and by averaging the control condition over the segment, Eqs. (33) and (34). A better solution might be to change the form of the integration procedure to improve the discrete adjoint equation. The authors sought a simple modification to the collocation procedure that would render the discrete adjoint equation accurate to the same order as the state discretization. Many methods were formulated (and experimented with numerically), but none displayed the desired discrete adjoint form while maintaining the accuracy of the state discretization and remaining computationally competitive.[30] Eventually, we abandoned collocation and implicit integration procedures and considered implementation of an explicit integration procedure.

Transcription Using Runge-Kutta and Parallel-Shooting

Consider discretizing the equations of motion using an explicit Runge-Kutta formula. As before, we introduce the (uniform) partition $[t_0, t_1, \ldots, t_N]$ and define $h = t_i - t_{i-1}$. We have state variables at each mesh point $x_i \cong x(t_i)$. Control variables are provided at the mesh points t_i and also at the center points $t_i + h/2$: $u_i \cong u(t_i)$ for $i = 0, \ldots, N$ and $v_i \cong u(t_{i-1} + h/2)$ for $i = 1, \ldots, N$. From a given mesh point, t_{i-1}, the equations of motion are integrated forward from x_{i-1} to the next mesh point t_i using the control u_{i-1}, v_i, and u_i by a step of a four-stage Runge-Kutta formula:

$$y_i^1 = x_{i-1} + \frac{h}{2}f(x_{i-1},u_{i-1}) \tag{40}$$

$$y_i^2 = x_{i-1} + \frac{h}{2}f(y_i^1,v_i) \tag{41}$$

$$y_i^3 = x_{i-1} + hf(y_i^2,v_i) \tag{42}$$

$$y_i^4 = x_{i-1} + \frac{h}{6}[f(x_{i-1},u_{i-1}) + 2f(y_i^1,v_i) + 2f(y_i^2,v_i)$$
$$+ f(y_i^3,u_i)] \tag{43}$$

where y_i^4 is an estimate of the state at the next mesh point, so for continuity we require that the "Runge-Kutta defects"

$$\Delta_i = y_i^4 - x_i \tag{44}$$

all be zero for $i = 1, \ldots, N$. Abadie,[22] Hager,[23] and more recently Kelley and Sachs[24] used this Runge-Kutta formula to transcribe optimal control problems, but they used the state-elimination approach, i.e., they used Eqs. (40–43) recursively to propagate the state forward from t_0 to t_f using the control sequences u_k and v_k. This elimination of the interior state variables does not affect the convergence analysis (so the discrete adjoint equation is unchanged). Abadie showed that this Runge-Kutta formula results in a discrete adjoint equation of the same Runge-Kutta form. Hager investigated this symmetry property for a variety of single-step and multistep methods. Kelley and Sachs considered the behavior of a Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm[14] applied to this transcription.

In our current notation, the discrete adjoint equation becomes[30]

$$\eta_i^3 = \lambda_i + \frac{h}{2}F(y_i^3,u_i)^T\lambda_i \tag{45}$$

$$\eta_i^2 = \lambda_i + \frac{h}{2}F(y_i^2,v_i)^T\eta_i^3 \tag{46}$$

$$\eta_i^1 = \lambda_i + hF(y_i^1,v_i)^T\eta_i^2 \tag{47}$$

$$\lambda_i = \lambda_{i-1} + \frac{h}{6}[-F(y_i^3,u_i)^T\lambda_i - 2F(y_i^2,v_i)^T\eta_i^3$$

$$- 2F(y_i^1,v_i)^T\eta_i^2 - F(x_{i-1},u_{i-1})^T\eta_i^1] \tag{48}$$

with boundary condition

$$\lambda N^T = \frac{\partial\Phi}{\partial xN} \tag{49}$$

Note that Eq. (48) could be solved for λ_{i-1} and propagated backwards from Eq. (49). Note also that $\lambda_k \cong \lambda(t_k)$, i.e., the discrete multipliers approximate the adjoints at the mesh points. The control conditions are

$$\frac{h}{3}[G(y_i^1,v_i)^T\eta_i^2 + G(y_i^2,v_i)^T\eta_i^3] = 0 \tag{50}$$

and

$$\frac{h}{6}[G(y_i^3,u_i)^T\lambda_i + G(x_{i-1},u_{i-1})^T\eta_i^1] = 0 \tag{51}$$

For the final time condition (transversality), consult Ref. 30.

Both the Hermite-Simpson integration procedure and the Runge-Kutta procedure have order h^5 local truncation errors,[27] although numerical experience of the authors and others[32] suggests that the Hermite-Simpson procedure is usually more accurate for a given step size. However, the Runge-Kutta method has several advantages. Because the adjoint equation is of the same Runge-Kutta form, the adjoints are approximated with the same accuracy as the states (h^5). Also, because the Runge-Kutta method is explicit, it can be incorporated into a parallel-shooting approach that we now describe.

The idea is to replace the single step of the Runge-Kutta procedure described earlier with multiple steps, allowing the use of larger intervals, and resulting in smaller nonlinear programming problems to achieve the same accuracy. Additional control variables must be introduced in each interval to accommodate the multiple integration steps.

Let p be the number of integration steps per interval, $[t_{i-1},t_i]$. As usual we provide states and controls at the mesh points $x_i \cong x(t_i)$ and $u_i \cong u(t_i)$. In addition, we provide the center controls $v_{i,j} \cong u(t_{i-1} + jh/2p)$ for $j = 1, \ldots, 2p - 1$ and for $i = 1, \ldots, N$. First we integrate from t_{i-1} forward one step to $t_{i-1} + h/p$ using Eqs. (40–43) with the controls u_{i-1}, v_{i1}, and v_{i2}. Using the resulting estimate of the state at $t_{i-1} + h/p$, we continue to integrate forward, using the additional controls $v_{i2}, \ldots, v_{i,2p-1}$, and finally u_i, to obtain an estimate of the state at the next mesh point, x_i', which replaces y_i^4 in the defect formula, Eq. (44). Figure 1 depicts this situation for $p = 3$. This is an adaptation of the parallel-shooting method devised for the solution of TPBVPs.[20,21]

Note that the parallel-shooting approach is somewhere in between the direct transcription and state-elimination methods. If the number of integration steps per interval is one (with many intervals), the method becomes direct transcription, and if the number of intervals is one (with many integration steps), the method becomes state elimination. As the intervals get large, sensitivity problems can appear; however, the parallel-shooting approach allows one to control accuracy and robustness independently. The most efficient strategy is to use as many intervals as it takes to get robust behavior and to use as many integration steps per interval as it takes to achieve the required accuracy. Next we compare the performance of the Runge-Kutta parallel-shooting method with the Hermite-Simpson method for an optimal trajectory problem.

Exploitation of the Discrete Adjoints

We have shown that at the solution of the nonlinear programming problem, the Lagrange multipliers corresponding to the defects are a discrete approximation to the adjoint variables of the continuous optimal control problem. This information can be readily exploited, *in a postprocessing program,* to assess the adequacy of the discrete approximation. One simple test is to extract the adjoints at the final time, as well as the states, and to integrate the state and adjoint equations (5) and (8) backwards to t_0, using the control generated by the control optimality condition, Eq. (10). If the initial state $x(t_0)$ is accurately recovered, then the continuous TPBVP has been solved, and this verifies optimality as well as feasibility of the discrete-approximate solution. Note that for the Hermite-Simpson method, the Lagrange multipliers approximate the adjoint variables at the segment centers, and one must either extrapolate to obtain the final adjoints or use the multipliers corresponding to the terminal constraints (13) at the solution and the terminal boundary condition (9) to solve for the adjoints at the final time. (We prefer the latter.) For the parallel-shooting method, the Lagrange multipliers approximate the adjoints at the mesh points, so the values at the final time are available directly.

The discrete adjoint information can also be used a posteriori to evaluate the optimality of an assumed control switching

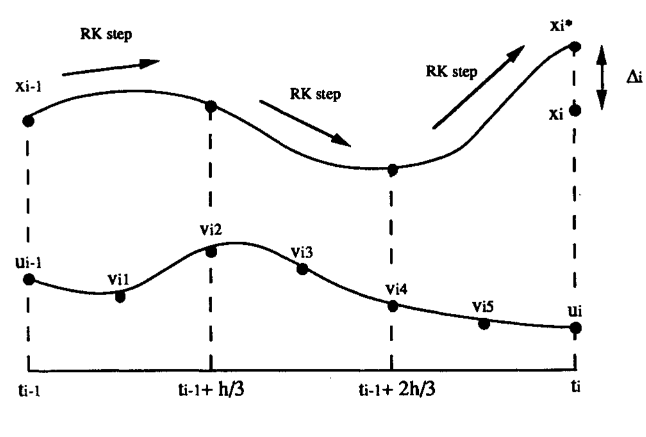

Fig. 1 Illustration of the parallel-shooting method.

structure for the application to finite-thrust spacecraft trajectories. The solutions to these problems usually consist of an alternating sequence of maximum-thrust (MT) arcs and coast arcs.[34] To solve such problems accurately and efficiently using the direct transcription method, it is necessary to guess the structure of this sequence a priori.[31] This is because the control discontinuities must be at "phase" boundaries.[12] *Also, numerical experience has shown that unless a solution structure is assumed, the additional degrees of freedom may prevent the optimizer from converging to a solution.* This assumed structure may not be optimal. However, the optimality may be determined by examining the general optimal control necessary conditions using the discrete approximation to the states and adjoints.

Consider the general thrust-limited spacecraft trajectory problem

$$\frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}t} = \boldsymbol{v} \tag{52}$$

$$\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t} = \boldsymbol{g}(\boldsymbol{r}) + sa\boldsymbol{u} \tag{53}$$

$$\frac{\mathrm{d}a}{\mathrm{d}t} = sa^2/c \tag{54}$$

where $\boldsymbol{g}(\boldsymbol{r})$ is the gravity model, a is the thrust acceleration corresponding to maximum thrust, s is the throttling parameter ($0<s<1$) and c is the effective exhaust velocity. Given $\boldsymbol{r}(t_0)$, $\boldsymbol{v}(t_0)$, and $a(t_0)$, we wish to satisfy some terminal constraints,

$$\boldsymbol{\psi}[\boldsymbol{r}(t_f),\boldsymbol{v}(t_f),t_f] = \boldsymbol{0} \tag{55}$$

while minimizing the integral of the throttling parameter (equivalent to minimizing the fuel expenditure or Δv)

$$J = \int_0^{t_0} s\,\mathrm{d}t \tag{56}$$

The necessary conditions are

$$\frac{\mathrm{d}\boldsymbol{\lambda}_r}{\mathrm{d}t} = -\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{r}}\boldsymbol{\lambda}_v \tag{57}$$

$$\frac{\mathrm{d}\boldsymbol{\lambda}_v}{\mathrm{d}t} = -\boldsymbol{\lambda}_r \tag{58}$$

$$\frac{\mathrm{d}\lambda_a}{\mathrm{d}t} = s[2\lambda_a a/c - \lambda_v] \tag{59}$$

The boundary conditions are

$$\boldsymbol{\lambda}_r(t_f) = \boldsymbol{\nu}_r^T\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{r}(t_f)} \tag{60}$$

$$\boldsymbol{\lambda}_v(t_f) = \boldsymbol{\nu}_v^T\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{v}(t_f)} \tag{61}$$

$$\lambda_a(t_f) = 0 \tag{62}$$

The control optimality conditions are

$$\boldsymbol{u} = -\boldsymbol{\lambda}_v \tag{63}$$

Table 1 Low-thrust escape transcription data

Method	Variables	CPU/s	Error
Hermite/Simpson (60)	427	190	3.4×10^{-5}
Parallel shooting (34 × 3)	385	95	1.5×10^{-5}
Parallel shooting (5 × 20)	270	72	3.1×10^{-5}

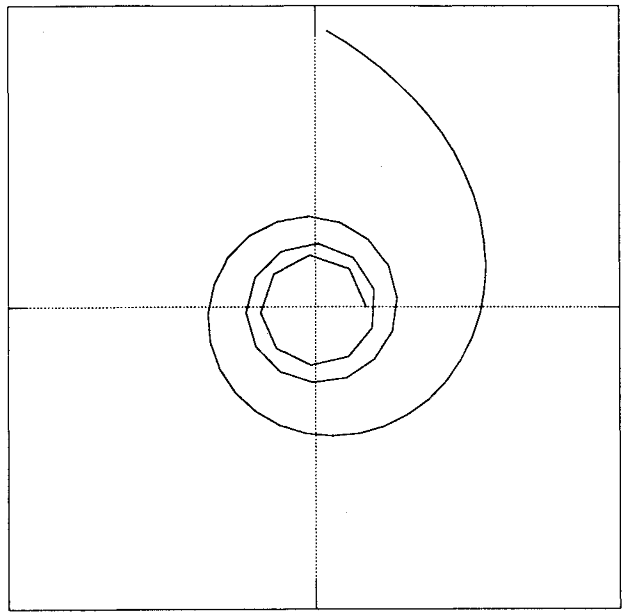

Fig. 2 Low-thrust escape trajectory.

$$s = 1 \quad \text{if} \quad \$>0, \qquad s = 0 \quad \text{if} \quad \$<0 \tag{64}$$

where the switching function \$ is

$$\$ = a\lambda_v - \lambda_a a^2/c - 1 \tag{65}$$

The negative of $\boldsymbol{\lambda}_v$ is referred to as the "primer vector." It determines the optimal thrust direction, and its magnitude is involved in the switching function (65).

For transcription, we first convert this problem into an assumed-structure form. For example, let us assume that the optimal solution consists of an MT arc followed by a coast arc and completed with another MT arc (valid for many transfer problems). We also use the integral-matching method (or "generalized defects")[31] for the coast arc, which replaces propagation of the state equations of motion with constraints that the integrals of the (unforced) motion be the same at the end of the first MT arc and the beginning of the second MT arc. The converted problem can be stated as follows.

Minimize the sum of the burn times,

$$J = (t_1 - t_0) + (t_f - t_2) \tag{66}$$

subject to equations of motion (52–54) on $[t_0,t_1]$ and $[t_2,t_f]$, the integral-matching conditions,

$$\boldsymbol{Q}[\boldsymbol{r}(t_1),\boldsymbol{v}(t_1),a(t_1)] = \boldsymbol{Q}[\boldsymbol{r}(t_2),\boldsymbol{v}(t_2),a(t_2)] \tag{67}$$

and the terminal constraints, Eq. (55), where t_1 is the time at the end of the first MT arc, and t_2 is the time at the beginning of the second MT arc. Over the MT arcs the adjoint equations (57–59) and the control condition (63) still obtain. But the switching condition (64) does not obtain since the thrust is assumed to be maximum (nor does this condition apply over the coast arc, which has essentially been removed from the problem[30]).

We solve the assumed-structure problem using direct transcription. Then we use the discrete state and adjoint information to evaluate the switching condition (64) over the MT arcs. (In this context, we refer to Eq. (65) as the "discrete switch function.") If this switching condition is violated, i.e., if the discrete switch function is negative on an MT arc, then the solution to the assumed-structure problem does not solve the general problem, and thus the assumed structure is not optimal. The switch function can also be tested over the coast arc (it should be negative), but to get state and adjoint information, Eqs. (52–54) and (57–59) must be integrated forward from the end of the previous MT arc or backward from the start of the following MT arc. If we observe a violation of the switching condition anywhere, we modify the assumed structure (by adding a coast arc or an MT arc) and solve the problem again. This process can be continued until the switching function agrees with the burn structure.

Solved Problems

Low-Thrust Escape

The first example we consider is a low-thrust escape. The vehicle starts in a circular orbit of radius 1 (canonical units are used in which the gravitational parameter $\mu = 1$) and burns at a constant thrust-acceleration level ($a = 0.0125$) for a fixed duration ($t_f = 16\pi$). The thrust-pointing history is to be optimized to maximize final energy. Coast arcs are not allowed, and so the trajectory is represented by a single phase. This problem was transcribed and solved using the Hermite-Simpson approach with cubic-spline controls, and also using the parallel-shooting approach developed before. The NLP routine used was a version of NPSOL[35] included in the Numerical Algorithms Group (NAG) library.[36] All cases were executed on the Cray X-MP/48 at the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign. Figure 2 shows the optimal trajectory. The final (specific) energy was -0.05. The accuracy of the discrete solutions was tested using the backward integration approach

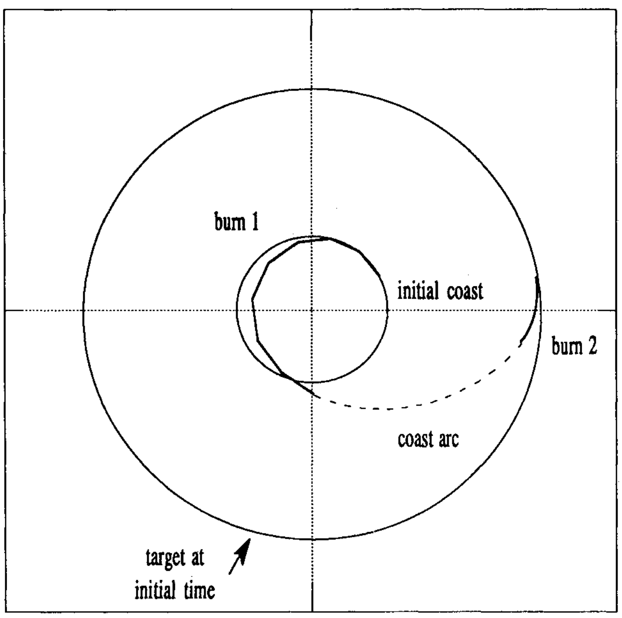

Fig. 3 Two-burn rendezvous trajectory.

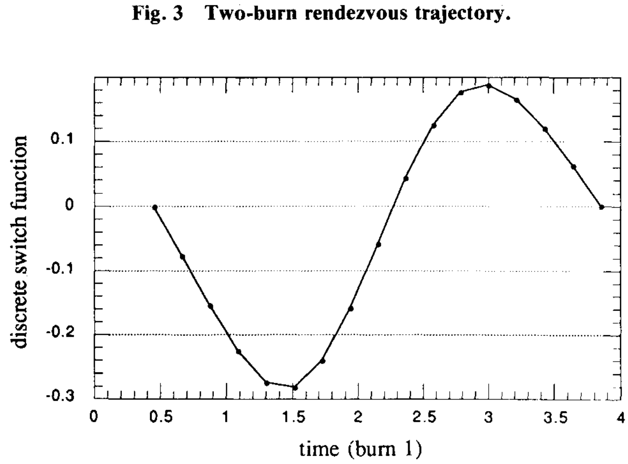

Fig. 4 Two-burn rendezvous discrete switch function for burn 1.

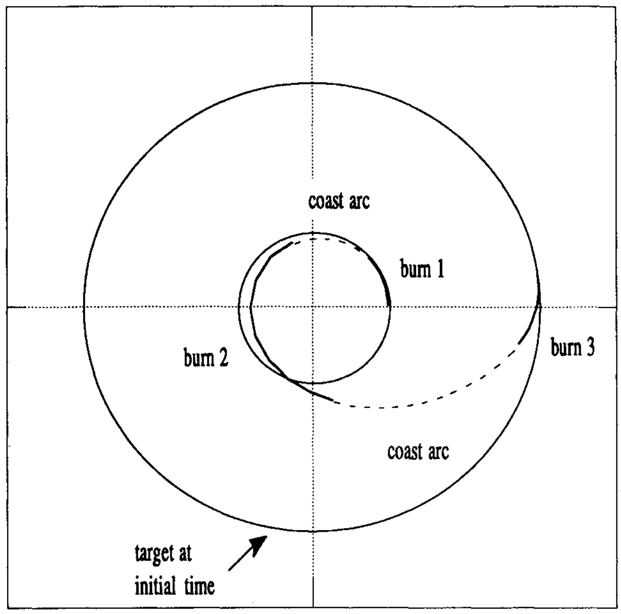

Fig. 5 Three-burn rendezvous trajectory.

mentioned earlier. Table 1 describes the various transcriptions. The accuracy given is the root-sum-square of the errors in the recovered initial states. The Hermite-Simpson version used 60 segments. Two versions of the parallel-shooting method are presented. The first used 34 intervals with 3 integration steps per interval. The second used only 5 intervals with 20 integration steps per interval. Note that both parallel-shooting approaches are better than the Hermite-Simpson approach, with regard to execution time and problem size, for roughly the same accuracy.

Two-Burn Rendezvous

The second example we consider is a thrust-limited rendezvous problem. The spacecraft starts in a circular reference orbit of radius 1 (again $\mu = 1$) and is required to rendezvous with a second spacecraft in a circular orbit of radius 3 within a specified final time. The final time was 10.0, the initial target lead angle was 258 deg, the initial acceleration was 0.1, and the exhaust velocity was 1.5 (all in canonical units). The impulsive version of this type of problem has been studied extensively,[37-39] and the optimal number of impulses has been shown to be dependent on the initial lead angle of the target and the allowed final time.

We first assumed that the solution consisted of two MT arcs separated by a coast art. Initial and final coasts were also allowed, and all coast arcs were handled via the matched-integral method. (For rendezvous problems, the method must be modified to calculate coast arc durations.[31]) This two-phase problem was transcribed using the Hermite-Simpson method with cubic spline controls. With 8 segments per phase and the 3 sets of generalized defects, there were a total of 18 nodes. The NLP problem had 128 variables and 105 constraints. Execution time was 5.1 s. The tentatively optimal trajectory is shown in Fig. 3. The solid lines represent MT arcs, and the dashed lines represent coast arcs. The total Δv in canonical units with respect to the initial reference orbit was 0.6425. We then examined the discrete switch function during the thrust arcs. Figure 4 shows the switch function $\$$ over the first burn and clearly indicates nonoptimality of the assumed structure.

Three-Burn Rendezvous

We then assumed a three-burn structure (three phases and four sets of matched-integral constraints) and resolved the problem. This structure required 27 nodes, generated an NLP problem with 192 variables and 159 constraints and required 24.6 s for solution. The optimal trajectory is given in Fig. 5;

the total Δv was 0.6045. Note that the first burn is essentially a retroburn and is driven by the final time constraint. Figure 6 shows the switch function over the whole trajectory obtained by a backward integration of the state and adjoint equations for this solution and verifies the correctness of the control switching structure.

Note that this procedure of adding burns until the general necessary conditions are satisfied is analogous to a similar procedure developed for impulsive problems.[37-39] Also note that if one assumes too many burns, the burn time for the extraneous burn is driven to zero, eliminating it from the problem structure. Nonoptimal initial and final coasts are similarly removed.

Earth-Moon Transfer

As a final example, we solved a low-thrust Earth-moon transfer problem. Circular-restricted three-body assumptions were used. The spacecraft starts in a geosynchronous orbit about the Earth and is required to transfer to a circular lunar orbit of radius 8448 km. The initial thrust acceleration was 2.2×10^{-3} m/s^2, and the effective exhaust velocity was 4.9×10^4 m/s. The solution was assumed to consist of a single Earth-escape MT arc and a single lunar-insertion MT arc, separated by a translunar coast arc. A dual coordinate system approach was used, where the equations of motion for the Earth MT arc and the first half of the translunar coast arc were written in Earth-centered polar coordinates. The equations for the second half of the translunar coast arc and the lunar insertion MT arc were written in polar coordinates centered at the moon. Both coordinate systems rotate with the Earth-moon system. A constraint is then introduced at the transition point (temporal center of the translunar coast arc) that enforces the appropriate coordinate transformation. This approach allowed us to take advantage of the smoothness of polar coordinates but avoided the ill-conditioning associated with parameterizing an essentially lunar orbit with Earth-centered polar coordinates (and vice versa).[30]

The transcription was done using the parallel-shooting approach. We used 12 intervals per phase and 5 integration steps per interval. The resulting NLP problem had 573 variables and 191 constraints. The execution time was 377 s. Figure 7 shows the optimal trajectory (in the rotating frame). The total Δv was 2.23 km/s. The total time of flight was 15.9 days.

This problem is the most ambitious we have solved to date. However, we suspect that much larger problems can be solved using this method. In particular, it is desirable to solve lunar-transfer problems for lower thrust levels, lower initial and final orbit radii, and for the noncoplanar case. The main obstacle is the size of the nonlinear programming problem that results. (Other enhancements, such as lunar eccentricity and a better Earth gravity potential, do not affect the problem size.) We have several suggestions for tackling this class of problems. First, the previous problem was solved using uniform mesh point distribution within a phase (and also the same number of mesh points for each phase). The data suggest that this is extremely inefficient, and an intelligent redistribution should be performed. Second, the NLP algorithm NPSOL is not designed to handle large sparse problems efficiently. The authors have had some success with the MINOS package[40] that is designed for large-scale systems, and we recommend it for problems with more than 400 variables. Finally, alternate coordinate systems could be employed to possibly increase the smoothness of the solution, reducing the number of mesh points required. A variation-of-parameters approach or other strategies[41] might be attempted.

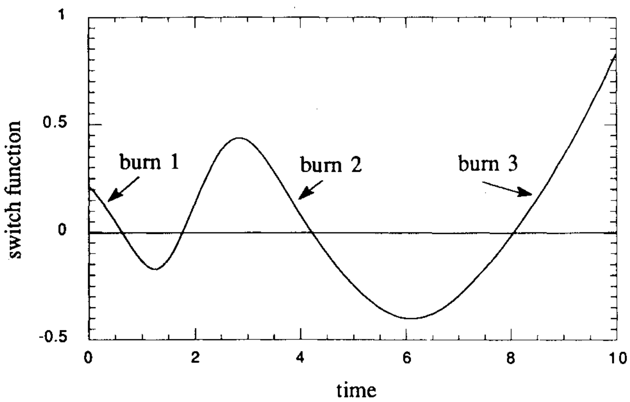

Fig. 6 Three-burn rendezvous switch function.

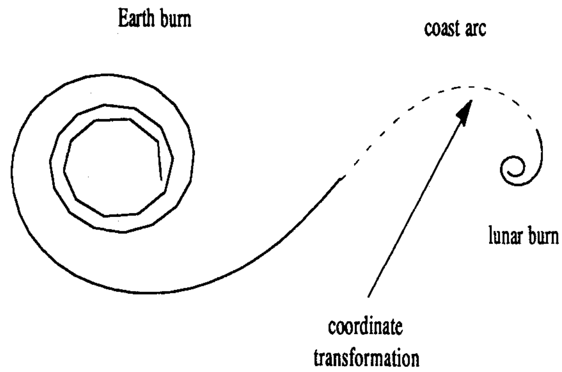

Fig. 7 Earth-moon transfer trajectory.

Conclusions

The collocation method of Hargraves and Paris has been shown to be a specific implementation of the general direct transcription method for approximating optimal control problems by nonlinear programming problems. Convergence analysis of the Hargraves-Paris method revealed a deficient accuracy of the discrete adjoint equation. An alternate scheme using a parallel-shooting implementation of a Runge-Kutta method was developed that has a discrete adjoint equation of the same form (and accuracy) as the state integration. Both of these methods have been applied successfully to several optimal spacecraft trajectory problems, including a low-thrust escape spiral, a three-burn thrust-limited rendezvous, and low-thrust lunar transfer. The discrete adjoint information was exploited to verify the accuracy of the discrete approximations and to evaluate the optimality of assumed control switching structures.

Acknowledgment

This work was supported by NASA Grants NAG 3-805 and NAG 3-1138 and administered by NASA Lewis Research Center.

References

[1]Kelley, H. J., "Gradient Theory of Optimal Flight Paths," *Journal of the American Rocket Society,* Vol. 30, No. 10, 1960, pp. 947–954.

[2]Bellman, R. E., and Kalaba, R. E., *Quasilinearization and Nonlinear Boundary Value Problems,* American Elsevier, New York, 1965.

[3]Keller, H. B., *Numerical Methods for Two-Point Boundary Value Problems,* Blaisdell, New York, 1968.

[4]Russell, R. D., and Shampine, L. F., "A Collocation Method for Boundary Value Problems," *Numerical Mathematics,* Vol. 19, No. 1, 1972, pp. 1–28.

[5]Canon, M. D., Cullum, C. D., and Polak, E., *Theory of Optimal Control and Mathematical Programming,* McGraw-Hill, New York, 1970.

[6]Polak, E., *Computational Methods in Optimization,* Academic Press, New York, 1971.

[7]Tabak, D., and Kuo, B. C., *Optimal Control by Mathematical Programming,* Prentice-Hall, Englewood-Cliffs, NJ, 1971.

[8]Bryson, A. E., and Ho, Y. C., *Applied Optimal Control,* Hemisphere, New York, 1975.

[9]Balakrishnan, A. V., "On a New Computing Method in Optimal Control," *SIAM Journal on Control,* Vol. 6, No. 2, May 1968, pp. 149–173.

[10]Johnson, F. T., "Approximate Finite-Thrust Trajectory Optimization," *AIAA Journal,* Vol. 7, No. 6, 1969, pp. 993–997.

[11]Hargraves, C. R., Johnson, F. T., Paris, S. W., and Rettie, I., "Numerical Computation of Optimal Atmospheric Trajectories," *Journal of Guidance and Control,* Vol. 4, No. 4, 1981, pp. 406–414.

[12]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.

[13]Dickmanns, E. D., and Well, K. H., "Approximate Solution of Optimal Control Problems Using Third-Order Hermite Polynomial Functions," *Proceedings of the 6th Technical Conference on Optimization Techniques,* Springer-Verlag, New York, IFIP-TC7, 1975.

[14]Fletcher, R., *Practical Methods of Optimization,* Wiley, New York, 1987.

[15]Neuman, C. P., and Sen, A., "A Suboptimal Control Algorithm for Constrained Problems Using Cubic Splines," *Automatica,* Vol. 9, No. 5, 1973, pp. 601–613.

[16]Renes, J. J., "On the Use of Splines and Collocation in a Trajectory Optimization Algorithm Based on Mathematical Programming," National Aerospace Lab., NLR-TR-78016 U., Amsterdam, The Netherlands, Sept. 1978.

[17]Kraft, D., "On Converting Optimal Control Problems Into Nonlinear Programming Problems," *Computational Mathematical Programming,* NATO ASI Series, Vol. F15, Springer-Verlag, New York, 1985.

[18]Wright, K., "Some Relationships Between Implicit Runge-Kutta, Collocation, and Lanczos τ Methods, and Their Stability Properties," *Informationbehandling* (BIT), Nordisk Tidskr, Vol. 10, 1970, pp. 217–227.

[19]Weiss, R., "The Application of Implicit Runge-Kutta and Collocation Methods to Boundary Value Problems," *Mathematics of Computation,* Vol. 28, No. 126, 1974, pp. 449–464.

[20]Morrison, D. D., Riley, J. D., and Zancanaro, J. F., "Multiple Shooting Method for Two-Point Boundary Value Problems," *Communications of the Association for Computing Machinery,* Vol. 5, No. 12, 1962, pp. 613–614.

[21]Ascher, U. M., Mattheij, R. M. M., and Russell, R. D., *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations,* Prentice-Hall, Englewood-Cliffs, NJ, 1988.

[22]Abadie, J., "Application of the GRG Algorithm to Optimal Control Problems," *Integer and Nonlinear Programming,* North Holland, Amsterdam, The Netherlands, 1970, pp. 191–212.

[23]Hager, W. W., "Rates of Convergence for Discrete Approximations to Unconstrained Control Problems," *SIAM Journal of Numerical Analysis,* Vol. 13, No. 4, 1976, pp. 450–472.

[24]Kelley, C. T., and Sachs, E. W., "Quasi-Newton Methods and Unconstrained Optimal Control Problems," *SIAM Journal of Control and Optimization,* Vol. 25, No. 6, 1987, pp. 1503–1516.

[25]Conte, S. D., and deBoor, C., *Elementary Numerical Analysis,* McGraw-Hill, New York, 1980.

[26]Butcher, J. C., "Implicit Runge-Kutta Processes," *Mathematics of Computation,* Vol. 18, No. 85, 1964, pp. 50–64.

[27]Butcher, J. C., *The Numerical Analysis of Ordinary Differential Equations,* Wiley, New York, 1987.

[28]Hulme, B. L., "Galerkin and Related One-Step Methods for Ordinary Differential Equations," *Mathematics of Computation,* Vol. 26, No. 4, 1972, pp. 881–891.

[29]Kunz, K. S., *Numerical Analysis,* McGraw-Hill, New York, 1957.

[30]Enright, P. J., *Optimal Finite-Thrust Spacecraft Trajectories Using Direct Transcription and Nonlinear Programming,* Ph.D. Thesis, Univ. of Illinois, Urbana, IL, 1991.

[31]Enright, P. J., and Conway, B. A., "Optimal Finite-Thrust Spacecraft Trajectories Using Collocation and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 5, 1991, pp. 981–985.

[32]Hargraves, C. R., and Paris, S. W., *Optimal Trajectories by Implicit Simulation (OTIS),* Air Force Wright Aeronautical Lab., AFWAL-TR-88-3057, Wright Patterson AFB, OH, Nov. 1988.

[33]Betts, J. T., and Huffman, W. P., "Application of Sparse Nonlinear Programming to Trajectory Optimization," *Journal of Guidance, Control, and Dynamics,* Vol. 15, No. 1, 1992, pp. 198–206.

[34]Lawden, D. F., *Optimal Trajectories for Space Navigation,* Butterworths, London, 1963.

[35]Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., *User's Guide for NPSOL (4.0),* TR SOL 86-2, Stanford Univ., Stanford, CA, Jan. 1986.

[36]Numerical Algorithms Group, "E04VCF-NAG FORTRAN Routine Library Document."

[37]Prussing, J. E., and Chiu, J. H., "Optimal Multiple-Impulse Time-Fixed Rendezvous Between Circular Orbits," *Journal of Guidance, Control, and Dynamics,* Vol. 9, No. 1, 1986, pp. 17–22.

[38]Lion, P. M., and Handelsman, M., "The Primer Vector on Fixed-Time Impulsive Trajectories," *AIAA Journal,* Vol. 6, No. 1, 1968, pp. 127–132.

[39]Jezewski, D. J., and Rosendaal, H. L., "An Efficient Method for Calculating Optimal Free-Space N-Impulsive Trajectories," *AIAA Journal,* Vol. 6, No. 11, 1968, pp. 2160–2165.

[40]Murtaugh, B. A., and Saunders, M. A., *MINOS 5.1 User's Guide,* TR SOL 83-20R, Stanford Univ., Stanford, CA, Jan. 1987.

[41]Jezewski, D. J., "An Ideal Reference Frame For Perturbed Orbital Motion," AIAA Paper 88-4218, Aug. 1988.