ELSEVIER

Comput. Methods Appl. Mech. Engrg. 190 (2001) 4677–4690

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

The solution of a launch vehicle trajectory problem by an adaptive finite-element method

Donald J. Estep [a,*], Dewey H. Hodges [b], Michael Warner [c,1]

[a] *Department of Mathematics, Colorado State University, Fort Collins, CO 80523, USA*
[b] *School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA*
[c] *Graduate Research Assistant, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA*

Received 1 August 1999; received in revised form 30 June 2000

Abstract

We describe a second-order discontinuous Galerkin finite-element method for the solution of an optimal control problem for determining the trajectory of a launch vehicle. We derive an a posteriori error estimate that is subsequently implemented as the basis for adaptive error control. We demonstrate that the computational error estimate is reliable and accurate while the adaptive error control provides a significant gain in efficiency over uniform discretizations. © 2001 Elsevier Science B.V. All rights reserved.

MSC: 65L07; 65L10; 65L20; 65L50; 65L60; 65L70

Keywords: A posteriori error estimates; Accumulation of error; Adaptive error control; Efficient discretization; Error estimates; Finite-element methods; Launch vehicles; Optimal control problems; Residual error; Stability; Trajectory optimization

1. Introduction

A trajectory problem for a launch vehicle typically takes the form of an optimal control problem, for example, to maneuver a rocket into a specified orbit while using the least amount of fuel or maximizing the final velocity. In the modern setting, the goal is to solve for the optimal trajectory sufficiently quickly to allow "real-time" control of the launch vehicle. This raises serious hurdles for the numerical solution of such trajectory problems in terms of efficiency and reliability.

The dynamics of a launch vehicle can be represented as a first-order system of ordinary differential equations for a set of state variables, which includes a set of control variables that appear in the differential equation as parameters. An optimal control problem is formulated by constructing a cost function involving the differential equation along with the associated boundary conditions. The problem we consider in this paper is a launch vehicle model initially posed by the LTV Aerospace and Defense (currently Lockheed–Martin–Vought Systems, we use LTV to refer to both) and developed by Hodges and Johnson [17]. We consider the problem of maximizing the magnitude of the velocity at the final time while satisfying the boundary conditions set by LTV.

[*] Corresponding author. Present address: School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332-1060, USA.
E-mail addresses: estep@math.gatech.edu, estep@math.colostate.edu (D.J. Estep), dewey.hodges@aerospace.gatech.edu (D.H. Hodges), gt4543b@gypsy.cad.gatech.edu (M. Warner).

[1] Present address: National Air Intelligence Center, Wright-Patterson Air Force Base.

PII: S0045-7825(00)00339-X

In the "direct" approach, the optimal control problem is solved by representing the time history of the control variables by a finite set of values and then minimizing the cost function with respect to these values using a nonlinear programming algorithm, see [15]. This approach has been used to solve a variety of trajectory problems, see [5,19]. However, current implementations of this approach are computationally too intensive to allow real-time control.

In the "indirect" approach, the calculus of variations is used to determine the optimal solution as the solution of a two-point boundary-value problem for a set of differential-algebraic equations that includes the original model equations as well as differential equations for the co-states, or Lagrange multipliers, and an optimality condition involving the control and state variables. The solution of the two-point boundary-value problem is then computed numerically. Note that in the indirect approach, it is the actual optimal solution that is approximated. In contrast in the direct approach, the optimization is carried out only over a finite-dimensional representation.

There are a variety of possible numerical methods to solve the two-point boundary-value problem resulting from the indirect approach. For example, a popular method is multiple shooting, which allows very accurate numerical solutions to be computed. However, shooting has so far turned out to be too slow to be used in real-time applications. As an alternative, Hodges and Bless proposed a "discontinuous Galerkin" finite-element method in [16]. This method is implemented in the FORTRAN code GENCODE that solves real-life optimal control problems with constraints on the controls and states, see [2–4,21–23]. GENCODE computes numerical solutions sufficiently quickly that a version of this code was employed by engineers at LTV to compute optimal trajectories updates for a missile model every one second.

Since the regularity and stability properties of solutions of trajectory models vary greatly as time passes, it is inefficient to compute using a uniform discretization. Instead, the discretization needs to be adjusted according to the difficulty of solving the differential equation. In this paper, we derive an a posteriori error estimate for the discontinous Galerkin finite-element method proposed by Hodges and Bless and describe how to use the estimate as the acceptance criteria for an iterative adaptive error control procedure. We test the adaptive error control on the trajectory problem proposed by LTV and show that it yields significant improvements in computational speed over computations using uniform discretizations. We conclude that implementing the adaptive error control results in significant gains in both reliability, since the error of numerical solutions is estimated, and efficiency, since the error estimate provides a basis for optimizing the discretization.

The a posteriori error estimate is derived using a variational analysis to relate the error of a numerical solution to its residual, i.e., the remainder left over from substituting the numerical solution into the differential equation. This analysis takes into account both the regularity and the stability properties of a solution and thereby yields a robustly reliable and accurate estimate of the error. This approach to adaptive error control has garnered increasing interest in recent years, see [8] for an overview.

2. A launch vehicle trajectory problem

In this paper, we solve an optimal control problem for a model of a launch vehicle originally posed by LTV and later fully developed by Hodges and Johnson [17]. The vehicle is described in three frames of reference. The effect of gravity and the position states are given in an inertial frame, the values of lift and drag due to the motion are determined in the wind frame, and finally the aerodynamic forces and the thrust are described in the body frame. The orientation of the vehicle is defined by the direction cosine matrix between the wind frame and the inertial frame

$$C = \frac{1}{(1 + (1/4)\theta^\top \theta)} \left(\left(1 - \frac{1}{4}\theta^\top \theta \right) I + \frac{1}{2}\theta\theta^\top - \tilde{\theta} \right),$$

where the vector θ and matrix $\tilde{\theta}$ contain the Rodrigues parameters, viz.,

$$\theta = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} \quad \text{and} \quad \tilde{\theta} = \begin{pmatrix} 0 & -\theta_3 & \theta_2 \\ \theta_3 & 0 & -\theta_1 \\ -\theta_2 & \theta_2 & 0 \end{pmatrix}. \tag{1}$$

The use of the Rodrigues parameters implies that the orientation description between the wind frame and the inertial frame is free of singularities for $\theta < 180°$.

The kinematic equations are derived from Newton's second law in terms of three Cartesian coordinates in the inertial frame $x = (x_1, x_2, x_3)^\top$ plus the magnitude of the velocity vector

$$\dot{x} = VC^\top e_1 \quad \text{and} \quad \dot{V} = \frac{1}{m} e_1^\top F_\mathrm{W},$$

where x_1 is positive in the north direction, x_2 positive in the east direction, $-x_3$ the altitude, $e_1 = (1,0,0)^\top$, m the mass, and F_W is a vector containing the resultant force on the vehicle in the wind frame. F_W is determined by the lift L, drag D, thrust T, and weight mg (assuming a flat earth in the inertial frame). Thrust and drag are assumed to be aligned along the axis of the vehicle, while the weight is in the e_3 direction.

The control variables in this model are the angles ϕ and α between the body axes $b = (b_1, b_2, b_3)^\top$ of the vehicle and the components of the wind frame $w = (w_1, w_2, w_3)^\top$. If the two frames are aligned, then the body frame is rotated about b_1 through the *bank angle* ϕ and about b_3 through the *angle of attack* α. This brings b_1 into alignment with the axis of the vehicle so the lift vector is aligned along b_2 and the drag aligned along b_3. In this description, $\alpha > 0$ and $-\pi < \phi < \pi$. Due to symmetry, we can take $-\pi/2 \leqslant \phi \leqslant \pi/2$. It follows that the forces in the wind frame can be derived as

$$\begin{aligned} F_{\mathrm{W}_1} &= mgC_{13} + (T - D)\cos(\alpha) - L\sin(\alpha), \\ F_{\mathrm{W}_2} &= mgC_{23} + (T - D)\cos(\phi)\sin(\alpha) - L\cos(\phi)\cos(\alpha), \\ F_{\mathrm{W}_3} &= mgC_{33} + (T - D)\sin(\phi)\sin(\alpha) - L\sin(\phi)\cos(\alpha). \end{aligned}$$

For the angular velocities, Hodges and Johnson [17] give an expression for the derivative of the Rodrigues parameters,

$$\dot{\theta} = \frac{1}{mV}\left(I + \frac{1}{2}\tilde{\theta} + \frac{1}{4}\theta\theta^\top\right)\tilde{e}_1 F_\mathrm{W}.$$

Thus, this model has seven states consisting of three position states, three orientation states, and the magnitude of the velocity plus two control variables consisting of the bank angle and the angle of attack. The mass m is given in tabular form by LTV. The lift L and the drag D are assumed to have the standard forms

$$L = \frac{1}{2} C_\mathrm{L} \rho V^2 S, \quad D = \frac{1}{2} C_\mathrm{D} \rho V^2 S,$$

where ρ is the atmospheric density derived from an exponential atmospheric model, S the given reference area, and C_L and C_D are the lift and drag coefficients, which are calculated using bicubic splines of data provided for various Mach numbers and angles of attack. The thrust T is given as

$$T + T_0 - A_\mathrm{e} P_\mathrm{a},$$

where A_e is the nozzle exit area, p_a the ambient pressure, and T_0 is the thrust produced by the missile. In this problem, LTV supplies the thrust as three different values corresponding to stages in the missile. Therefore, the problem is a *multiphase problem*. Since the points at which the parameters change is known a priori, this causes no real difficulty.

The optimal control problem we consider is to compute the solution for which the magnitude of the velocity at the final time, $V(T)$, is maximized while satisfying the initial and final conditions set by LTV

$$
\begin{aligned}
&x_1(0) = 0.0\ \text{m}, \quad x_2(0) = 0.0\ \text{m}, \quad x_3(0) = -3.0\ \text{m},\\
&\theta_1(0) = 0.0, \quad \theta_2(0) = 0.68865, \quad \theta_3(0) = 0.0,\\
&x_1(T) = 15{,}000\ \text{m}, \quad x_2(T) = 15{,}000\ \text{m}, \quad x_3(T) = -12{,}000\ \text{m},\\
&\theta_1(T) = 0.0, \quad \theta_2(T) = 2(\sqrt{2}-1), \quad \theta_3(T) = 0,\\
&V(0) = 19.6673\ \text{m/s}^2
\end{aligned}
$$

and $T \approx 25$ s. In addition, continuity conditions are posed at the the internal times at which thrust changes, which are $t = 7$ and $t = 15$, respectively. Note that because the initial and final azimuth angle are zero, the vehicle begins and ends oriented to the north, while it is pitched up $\pi/4$ from the horizontal at the end. The vehicle does not remain in a single plane, however, since the final value of x_2 is nonzero.

3. Discretization

We use the calculus of variations to reduce the solution of the optimal control problem to the solution of a two-point boundary-value problem, which we then solve numerically.

We describe the reduction briefly, see [2,20] for more details. We write the the multiphase model for the launch vehicle as

$$
\begin{aligned}
&\dot{y} = f_i(y,u,t), \quad \tilde{t}_{i-1} < t < \tilde{t}_i, \qquad 1 \leqslant i \leqslant M,\\
&\Psi(y(0), y(\tilde{t}_1^-), y(\tilde{t}_1^+), \ldots, y(\tilde{t}_{M-1}^+), y(T), T) = 0,
\end{aligned}
\tag{3.1}
$$

where $y \in \mathbf{R}^{d_1}$ represents the states and $u \in \mathbf{R}^{d_2}$ represents the control variables, $0 = \tilde{t}_0 < \tilde{t}_1 < \cdots \tilde{t}_M = T$ is a partition of $[0, T]$, and we use the notation $\lim_{t\to\tilde{t}_i^\pm} y(t) = y(\tilde{t}_i^\pm)$. We assume that the functions $\{f_i\}$ are smooth inside the corresponding sub-intervals.

The cost associated to a solution is given by a cost functional of the form

$$
l(y(0), y(T), T) + \int_0^T L(y,u,t)\,\mathrm{d}t
$$

for some appropriate functions l and L. In this case, the differential equation and the boundary conditions act as constraints, so these conditions are adjoined to the cost functional using Lagrange multipliers λ and $\Lambda(t)$, respectively, to obtain

$$
\begin{aligned}
&l(y(0), y(T), T) + \lambda^\top \Psi(y(0), y(\tilde{t}_1^-), y(\tilde{t}_1^+), \ldots, y(\tilde{t}_{M-1}^+), y(T), T)\\
&\quad + \sum_{i=1}^{M} \int_{\tilde{t}_{i-1}}^{\tilde{t}_i} \Big(L(y,u,t) + \Lambda(t)^\top (f_i(y,u,t) - \dot{y}) \Big)\,\mathrm{d}t.
\end{aligned}
$$

$\Lambda(t)$ is the vector of co-states of the problem. Defining the Hamiltonian of the problem H so that $H_i = L + \Lambda^\top f_i$ on $[\tilde{t}_{i-1}, \tilde{t}_i)$ and setting $\Phi = l + \lambda^\top \Psi$, $y_0 = y(0)$, and $y_T = y(T)$, we can write this functional as

$$
\Phi\Big(y(0), y(\tilde{t}_1^-), y(\tilde{t}_1^+), \ldots, y(\tilde{t}_{M-1}^+), y(T), \lambda^\top, T\Big) + \sum_{i=1}^{M} \int_{\tilde{t}_{i-1}}^{\tilde{t}_i} \Big(H_i(y,u,\Lambda^\top,t) - \Lambda^\top \dot{y} \Big)\,\mathrm{d}t.
$$

Using the calculus of variations, we see that the optimal solution satisfies a set of differential-algebraic equations consisting of the state and co-state equations and the optimality condition

$$
\begin{aligned}
&\dot{y} - f_i(y,u,t) = 0, \quad \tilde{t}_{i-1} < t < \tilde{t}_i,\\
&\dot{\Lambda}^\top + \frac{\partial H_i}{\partial y}(y,\Lambda,u,t) = 0, \quad \tilde{t}_{i-1} < t < \tilde{t}_i,\\
&\frac{\partial H_i}{\partial u}(y,\Lambda,u,t) = 0,
\end{aligned}
\tag{3.2}
$$

for $1 \leqslant i \leqslant M$. The corresponding boundary conditions are

$$\Lambda^{\top}(0) = -\frac{\partial \Phi}{\partial y_0}, \quad \Lambda^{\top}(T) = \frac{\partial \Phi}{\partial y_T}, \quad \Psi = 0, \quad H(T) = -\frac{\partial \Phi}{\partial T}, \tag{3.3}$$

where the last condition arises if the final time is allowed to vary.

In the optimal control problem for the launch vehicle, the states corresponding to position and orientation are specified at the initial and final times so the co-states for these variables are unknown at those points, while the co-state corresponding to the velocity state is constrained to be -1 at the final time. The cost functional is $l(y(T)) = -V(T)$, so the Hamiltonian is simply $H = \Lambda^{\top} f$ and the co-state equations are $\dot{\Lambda}^{\top} = -\Lambda^{\top} f_y$.

For simplicity we assume that the control variables u can be eliminated from these equations by solving the optimality condition in these problems exactly. In the optimal control problem for the launch vehicle model, we are able solve the algebraic constraints to round-off accuracy.

The optimal solution may include discontinuities in the Hamiltonian and co-states of the form

$$\Lambda^{\top}(\tilde{t}_i^-) = \frac{\partial \phi}{\partial y(\tilde{t}_i^-)}, \quad \Lambda^{\top}(\tilde{t}_i^+) = -\frac{\partial \Phi}{\partial y(\tilde{t}_i^+)}, \quad H_{i+1}(t_i^+) = H_i(\tilde{t}_i^-) + \frac{\partial \Phi}{\partial \tilde{t}_i}, \qquad 1 \leqslant i \leqslant M.$$

If a given state is continuous across an intermediate node, then the corresponding co-state is also continuous across the same node. Likewise if the intermediate node is not specified as part of the problem, then the Hamiltonian is continuous across that node, despite the possibility of having different forms on either side. In the optimal control problem for the launch vehicle, the thrust takes on three different values during the flight and the state and co-state equations change across each internal node. But they are different only in the value of the thrust for each phase while the co-states are continuous across the internal nodes while the Hamiltonian is discontinuous across the nodes.

Combining the states and co-states in (3.2) into one variable x, we first discretize the single phase two-point boundary-value problem

$$\begin{aligned} &\dot{x} = f(x), \quad 0 < t < T, \\ &I_0 x(0) = x_0, \ I_T x(T) = x_T, \quad x \in \mathbf{R}^d, \end{aligned} \tag{3.4}$$

where I_0 and I_T are diagonal matrices with ones or zeroes on the diagonal and $\mathrm{rank}(I_0) + \mathrm{rank}(I_T) = d$. It is straightforward to handle more complicated linear boundary conditions and nonautonomous problems.

The finite-element method we used to discretize (3.4) was originally proposed by Hodges and Bless [16]. It has three immediate attractions: the boundary conditions are enforced weakly, the method preserves the Hamiltonian, and it yields second-order convergence while avoiding the need for quadrature in evaluating the integrals involving f.

To discretize (3.4), the time axis is partitioned as $t_0 = 0 < t_1 < t_2 < \cdots < t_N = T$, with $k_n = t_n - t_{n-1}$ and $I_n = (t_{n-1}, t_n)$, the trial space is chosen to be the space of piecewise constant polynomials and the test space is chosen to be the space of continuous piecewise linear polynomials, i.e.,

$$\mathscr{D}^0 = \{V | V|_{I_n} \in \mathscr{P}^0(I_n)\}, \quad \mathscr{C}^1 = \{V | V|_{I_n} \in \mathscr{P}^1(I_n), \ V \text{ is continuous}\},$$

where $\mathscr{P}^p(I_n)$ denotes the vector space of polynomials of degree less than or equal to p on I_n.

In their original formulation, Hodges and Bless ([16]) form a variational equation associated to (3.4) by multiplying the differential equation by a continuous test function, integrating over time, and then integrating the time-derivative term by parts to move the time-derivative on the test function while the initial and final conditions are posed in variational form. The equation they obtain is

$$\int_0^T (x, \dot{v})\,\mathrm{d}t + \int_0^T (f(x), v)\,\mathrm{d}t - (x, v)|_0^T - (I_T x(T) - x_T, \hat{v}_T) + (I_0 x(0) - x_0, \hat{v}_0) = 0, \tag{3.5}$$

where $\hat{v}_T$ and $\hat{v}_0$ denote the freely varying values of v at the endpoints. The finite-element approximation $X \in \mathscr{D}^0$ satisfies

$$\sum_{n=1}^{N}\int_{I_n}(X,\dot{V})\,\mathrm{d}t+\sum_{n=1}^{N}\int_{I_n}(f(X),V)\,\mathrm{d}t-(\hat{X}_N,V_N)+(\hat{X}_0,V_0)+(I_0\hat{X}_0-x_0,\hat{V}_0)\\ -(I_T\hat{X}_N-x_T,\hat{V}_N)=0. \tag{3.6}$$

To write this finite-element method in the manner usually used for discontinuous Galerkin methods, see [9,10], we avoid the initial integration by parts and instead introduce "jump" terms that arise when the discontinuous approximate solution is differentiated. In this framework, the finite-element approximation $X\in\mathscr{D}^0$ satisfies

$$\sum_{n=1}^{N}\left(\int_{I_n}(\dot{X}-f(X),V)\,\mathrm{d}t\right)+\sum_{n=0}^{N}([X_n],V_n)=0 \tag{3.7}$$

for all $V\in\mathscr{C}^1$, $I_0X_0^-=x_0$, and $I_TX_N^+=x_T$, where for $W\in\mathscr{D}^0$, $[W_n]=W_n^+-W_n^-$, $W_n^\pm=\lim_{t\to t_n^\pm}W(t)$, denotes the jump in the value of W at the indicated node and for $V\in\mathscr{C}^1$, $V_n=V(t_n)$.

Either approach yields a system of equations for the nodal values of X:

$$\begin{aligned}&I_0X_0^-=x_0,\\ &-X_1^-+X_0^-+k_1f(X_1^-)/2=0,\\ &-X_n^-+X_{n-1}^-+k_nf(X_n^-)/2+k_{n-1}f(X_{n-1^-})/2=0,\quad 2\leqslant n\leqslant N-1,\\ &X_N^-+X_N^++k_Nf(X_N^-)/2=0,\\ &{}_TX_N^+=x_T.\end{aligned} \tag{3.8}$$

Since the internal nodes are given a priori, discretization of a multiphase problem is performed by treating each two-point boundary-value problem posed on each sub-interval in the same way as (3.4). This is another advantage of using discontinuous functions in the finite-element method.

4. The a posteriori error analysis and adaptive error control

The adaptive error control is based on an a posteriori error estimate that is computed after the approximate solution is computed. The a posteriori analysis is based on the introduction of the *residual*, which is determined by substituting the numerical solution into the differential equation and computing the remainder. The residual, unlike the error $e=x-X$, is explicitly computable.

In order to determine a relationship between the error and the residual, we use a variational analysis after introducing the dual problem to the differential equation. This yields an exact representation of a projection of the error as a linear combination of weights times local residuals, where the weights are determined by norms of the solution of the dual problem. Then, we introduce norms into the error representation to obtain the error estimate. We call the normed weights in the estimate *stability factors* since they reflect the accumulation and propagation of error in the numerical solution arising from the local residuals at each point in time.

To obtain a linear dual problem, we linearize around an average of the true and approximate solutions. We define the average linearized coefficient matrix by

$$A(t)=\int_0^1\frac{\partial f}{\partial x}(\theta x(t)+(1-\theta)X(t))\,\mathrm{d}\theta,$$

where $\partial f/\partial x$ denotes the Jacobian of f. This choice yields the useful relation

$$(A(t)e(t),\phi)=(f(x)-f(X),\phi) \tag{4.1}$$

for any $\phi\in\mathbf{R}^d$. The corresponding forced dual problem to (3.4) is

$$\begin{aligned} &-\dot{\phi} = A(t)^\top \phi - e/\|e\|, \quad 0 < t < T, \\ &(I - I_0)\phi(0) = \phi_0, \;\; (I - I_T)\phi(T) = \phi_T. \end{aligned} \tag{4.2}$$

Multiplying the dual equation (4.2) by e and integrating, we start with

$$\int_0^T \|e(t)\| \, \mathrm{d}t = \int_0^T (e(t), \dot{\phi}(t)) \, \mathrm{d}t + \int_0^T (e(t), A^\top(t)\phi(t)) \, \mathrm{d}t.$$

Integrating by parts inside each interval in the first integral on the right, moving A onto e, using (4.1), and using the fact that x solves the differential equation exactly, we obtain

$$\int_0^T \|e\| \, \mathrm{d}t = \sum_{n=1}^N \int_{I_n} (\dot{X}, \phi) \, \mathrm{d}t + \sum_{n=1}^N \big((x_n - X_n^-, \phi_n) - (x_{n-1} - X_{n-1}^+, \phi_{n-1}) \big) - \sum_{n=1}^N \int_{I_n} (f(X), \phi) \, \mathrm{d}t.$$

Using the continuity of x, the definition of X_0^- and X_N^+, and choosing $\phi_0 = \phi_T = 0$, we simplify the sum in the middle on the right to get

$$\int_0^T \|e\| \, \mathrm{d}t = \sum_{n=1}^N \int_{I_n} (\dot{X} - f(X), \phi) \, \mathrm{d}t + \sum_{n=0}^N ([X_n], \phi_n). \tag{4.3}$$

This equation is similar to (3.7) defining X except that $\phi \notin \mathscr{C}^1$. To see that the difference is small, we use Galerkin orthogonality to insert a projection of ϕ into $\mathscr{C}^1$, i.e., we subtract (3.7) from (4.3) to obtain the *error representation formula* which holds for all $V \in \mathscr{C}^1$,

$$\int_0^T \|e\| \, \mathrm{d}t = \sum_{n=1}^N \int_{I_n} (\dot{X} - f(X), \phi - V) \, \mathrm{d}t + \sum_{n=0}^N ([X_n], \phi_n - V_n). \tag{4.4}$$

We can see the terms on the right-hand side are small by choosing $V = P_1\phi \in \mathscr{C}^1$ as a suitable approximation of ϕ in the test space. Taking norms and using the fact that $\dot{X} \equiv 0$, we get

$$\int_0^T \|e\| \, \mathrm{d}t \leqslant \sum_{n=1}^N \int_{I_n} \|f(X)\| \|(I - P_1)\phi\| \, \mathrm{d}t + \sum_{n=0}^N \|[X_n]\| \|(I - P_1)\phi_n\|. \tag{4.5}$$

Beginning with the first term on the right-hand side, we estimate

$$\sum_{n=1}^N \int_{I_n} \|f(X)\| \|(I - P_1)\phi\| \, \mathrm{d}t \leqslant \sum_{n=1}^N \|f(X_n^-)\| \|(I - P_1)\phi\|_{I_n} k_n,$$

where $\|\cdot\|_{I_n} = \max_{I_n} \|\cdot\|$. We write the second term on the right-hand side of (4.5) as

$$\sum_{n=0}^N \|[X_n]\| \|(I - P_1)\phi_n\| = \|[X_0]\| \|(I - P_1)\phi_0\| + \sum_{n=1}^{N-1} \|[X_n]\| \|(I - P_1)\phi_n\| + \|[X_N]\| \|(I - P_1)\phi_N\|.$$

We then expand the sum in the middle on the right-hand side as

$$\sum_{n=1}^{N-1} \|[X_n]\| \|(I - P_1)\phi_n\| = \sum_{n=1}^{N-1} \left(\frac{k_n}{k_n + k_{n+1}} + \frac{k_{n+1}}{k_n + k_{n+1}} \right) \|[X_n]\| \|(I - P_1)\phi_n\|,$$

which we write as

$$\begin{aligned} &\frac{k_1}{k_1 + k_2} \|[X_1]\| \|(I - P_1)\phi_1\| + \sum_{n=2}^{N-1} \left(\frac{k_n}{k_n + k_{n+1}} \|[X_n]\| \|(I - P_1)\phi_n\| + \frac{k_n}{k_{n-1} + k_n} \|[X_{n-1}]\| \|(I - P_1)\phi_{n-1}\| \right) \\ &\quad + \frac{k_N}{k_{N-1} + k_N} \|[X_N]\| \|(I - P_1)\phi_N\|. \end{aligned}$$

Using the continuity of ϕ, we have the estimates

$$\|(I-P_1)\phi_n\| \leqslant \|(I-P_1)\phi\|_{I_n} \quad \text{and} \quad \|(I-P_1)\phi_n\| \leqslant \|(I-P_1)\phi\|_{I_{n+1}}.$$

Using these estimates in (4.5), we conclude that

$$\begin{aligned}\int_0^T \|e\|\,\mathrm{d}t \leqslant & \left(\|[X_0]\| + \frac{k_1}{k_1+k_2}\|[X_1]\| + k_1\|f(X_1^-)\|\right)\|(I-P_1)\phi\|_{I_1} \\ & + \sum_{n=2}^{N-1}\left(\frac{k_n}{k_n+k_{n+1}}\|[X_n]\| + \frac{k_n}{k_{n-1}+k_n}\|[X_{n-1}]\| + k_n\|f(X_n^-)\|\right)\|(I-P_1)\phi\|_{I_n} \\ & + \left(\|[X_N]\| + \frac{k_N}{k_{N-1}+k_N}\|[X_{N-1}]\| + k_N\|f(X_N^-)\|\right)\|(I-P_1)\phi\|_{I_N},\end{aligned} \tag{4.6}$$

We assume that there is an *interpolation constant* C such that for all ϕ with two continuous derivatives

$$\|(I-P_1)\phi\|_{I_n} \leqslant Ck_n \int_{I_n}\left\|\frac{\mathrm{d}^2}{\mathrm{d}t^2}\phi\right\|\mathrm{d}t.$$

Such error bounds are satisfied by the L_2 projection and the standard interpolation operator, see [6]. We therefore define the discretization residual on the interval I_n, $1 \leqslant n \leqslant N$, by

$$\mathscr{R}(X,I_n)k_n = \begin{cases} \|[X_0]\| + \frac{k_1}{k_1+k_2}\|[X_1]\| + k_1\|f(X_1^-)\|_{I_1}, & n=1, \\ \frac{k_n}{k_n+k_{n-1}}\|[X_{n-1}]\| + \frac{k_n}{k_n+k_{n+1}}\|[X_n]\| + k_n\|f(X_n^-)\|_{I_n}, & 1<n<N, \\ \frac{k_N}{k_{N-1}+k_N}\|[X_{N-1}]\| + \|[X_N]\| + k_N\|f(X_N^-)\|_{I_N}, & n=N \end{cases}$$

and the corresponding stability factor

$$\mathscr{S}(\psi,I_n) = C\int_{I_n}\left\|\frac{\mathrm{d}^2}{\mathrm{d}t^2}\phi\right\|\mathrm{d}t.$$

From (4.6), we obtain the a posteriori error estimate

$$\int_0^T \|e\|\,\mathrm{d}t \leqslant \sum_{n=1}^{N}\mathscr{R}(X,I_n)\mathscr{S}(\psi,I_n). \tag{4.7}$$

Mesh selection is a complicated nonlinear constrained optimization problem to minimize the cost under the constraint of keeping a computable *mesh acceptance criteria* below a desired tolerance. The solution of the mesh selection problem is approximated by a iterative process:

1. Compute the numerical solution on the current mesh.
2. Compute the acceptance criteria for this solution.
3. Either decide to accept the solution or predict the mesh changes needed to compute a new solution that meets the constraint.

See for example [7,9,18]. We start with a coarse mesh as a way to keep the computational costs low. The mesh refinement decision is based on a *principle of equidistribution* in which the acceptance criteria is divided into components measuring the local contribution from each element and the mesh is adjusted to keep these local contributions roughly equal.

The main difference between the approach to adaptivity in this paper and to other approaches in the literature is the acceptance criteria, which in our case is

$$\sum_{n=1}^{N}\mathscr{R}(X,I_n)\mathscr{S}(\psi,I_n) \leqslant \delta, \tag{4.8}$$

where δ is a predetermined tolerance. The a posteriori result (4.7) actually estimates the true or global error by taking into account both the numerical solution's local regularity properties, in the form of residual, and stability properties, in the form of the stability factor. Classical approaches to constructing acceptance criteria consider only the local regularity of the solution and therefore cannot estimate the true error in general, see [13,14].

In this case, the residual is second-order in the time step. We define $\hat{\mathscr{R}} = \mathscr{R}/k_n^2$, so the principle of equidistribution means that the mesh should be chosen so that

$$k_n^2 \mathscr{S}(I_n)\hat{\mathscr{R}}(X, I_n) \leqslant \frac{\delta}{N}, \quad 1 \leqslant n \leqslant N. \tag{4.9}$$

If the acceptance criteria (4.8) is not satisfied on the current mesh, then steps in a new mesh $\{k_{n,\mathrm{new}}\}$ are determined from the current steps $\{k_{n,\mathrm{old}}\}$ by

$$\left(\frac{k_{n,\mathrm{new}}}{k_{n,\mathrm{old}}}\right)^2 = \frac{\delta}{k_{n,\mathrm{old}}^2 N\left(\mathscr{S}(I_n)\hat{\mathscr{R}}(X, I_n)\right)}, \tag{4.10}$$

for each n. (4.10) can also be used to determine if the mesh should be coarsened, namely when the quantity on the right in (4.10) is larger than one on adjacent time steps, we can combine neighboring elements to form larger elements. In practice, we do not coarsen in a region unless the step sizes are too small by a factor of 10.

5. Implementation

As we have seen, (3.7) yields a set of nonlinear algebraic equations of the form $F(X) = 0$, where we use X to denote the nodal values $\{X_n^-\}$. These equations are solved using a restricted step Newton–Raphson iteration, which requires the Jacobian of F. To compute this Jacobian, which involves partial derivatives of the system equations and boundary conditions with respect to states and controls, GENCODE interfaces to either of the symbolic manipulation packages MACSYMA or MAPLE.

As formulated, the Jacobian of the system becomes singular when the angle of attack α is zero, in which case the bank angle β becomes irrelevant. Moreover, α approaches zero in the problem we solve, which causes the Jacobian to become badly conditioned. To relieve this problem, we use the control formulation derived in [17] that is based on the Lagrangian equinoctial variables, see [1]. The control variables α and ϕ are replaced by

$$\beta_2 = \cos(\phi)\tan(\alpha) \quad \text{and} \quad \beta_3 = \sin(\phi)\tan(\alpha).$$

With the correct implementation, the problems associated to α approaching zero are avoided, see [17].

We found that the convergence of the Newton iteration for this problem is very sensitive to the initial guess. We generate a good initial guess by computing solutions of a sequence of simpler but related problems in an ad hoc homotopy method.

The most accurate numerical solution that can be reasonably computed by a HP-UX workstation has a Jacobian containing on the order of 120,000 nonzero elements. The sparsity of the Jacobian is exploited by using sparse solvers from the Harwell library. These routines store information about the structure of the linear systems between calls, so that subsequent Newton iterations take less time to solve.

We approximate the forcing function $e/\|e\|$ for the dual problem by using Richardson extrapolation on numerical solutions for two different meshes. At any given level in the refinement process, we store two consecutive meshes and corresponding numerical solutions, where time steps in the old mesh are always divided at least into two steps, and possibly more, when the new mesh is constructed. We estimate the error of the solution on the finer mesh by using the asymptotic estimate of the error of the solution on the coarser mesh obtained by Richardson extrapolation.

One difficulty that arises with the original variables is that the scales of the different components of the solution vary over a large range. For example, the Rodrigues parameter is order one while the position states range up to 10,000. Since we are estimating the vector norm of the error, it is desirable to keep the components of the solution roughly the same size. Consequently when computing, we rescale the launch vehicle model by multiplying the states and co-states by the scaling factors listed in Table 1. In the new

Table 1
Scaling factors for variables in the launch vehicle model

Variable	Scaling factor
x_1	0.0017
x_2	0.002
x_3	0.003
V	0.04
θ_1	23
θ_2	13
θ_3	20
t	3

variables, the maximum value of each state and its corresponding co-states are within 20% of each other while the maxima of all eight states and eight co-states are in the range $[7, 75]$.

To form the coefficient matrix $A^{\top}$ of the dual problem, we linearize around the approximate solution rather than the average of the true and approximate solutions in the definition. The effect of this substitution depends on the degree of nonlinearity of the problem as measured by how much the stability properties of nearby trajectories can differ. It is possible to show that the stability factors obtained from the dual problem linearized around the approximate solution converges to the stability factors obtained from the dual problem with the averaged coefficients as the discretization is refined, see [10]. Computationally, the linearization does not appear to be a problem.

To solve the dual problem, we use the same order finite-element method, quadrature, and mesh used to compute X. This means that the computational cost of the error estimation is roughly the same as the cost of a single Newton iteration in the computation of X. So the error estimation typically represents 20–25% of the total cost of the computation. In addition the numerical solution of the dual problem requires the same amount of memory to store as X, while the coefficients of the dual problem are already stored in order to implement the Newton method used to compute X. So the additional memory needed to compute the error estimate is not significant.

We compute the derivatives in the stability factor exactly after reducing the order through repeated substitution using the differential equation. The stability factor also involves the interpolation constant C. Standard interpolation theory gives a value for this constant that is "sharp" in the sense that there is some function in the appropriate function space whose interpolation error is given precisely by the theoretical estimate with that value for the constant. In practice, however, we find the analytic value leads to consistent over-estimation by 1–2 orders of magnitude for the solutions of the differential equations we compute. Instead of the analytic value, we *calibrate* (see [11]) the error estimate (4.7) by computing the seminorm on the dual solution defining the stability factor and the residual error of a numerical solution for a problem in which the true solution, and hence the error, is known, then we set the ratio of the error to the bounding quantity without the constant to be the constant C. In the computations in this paper, we calibrate using the exact solution of a control problem for a simple trajectory problem to get $C \approx 0.0048$. The reason for the consistent over-estimation when using the analytic value of the interpolation constant and the theoretical justification of the calibration process are interesting issues for future study.

The finite-element method used in this paper was originally implemented in the FORTRAN code GENCODE by Bless and Hodges. The original version of GENCODE can quickly solve optimal control problems involving single phase problems with constraints on the controls and states, see [2–4]. Warner and Hodges [21,23,22] extend GENCODE in several ways. First, the new version of GENCODE implements higher-order shape functions, allowing arbitrary order inside each element, see [20,12] for more details on higher-order versions of the finite-element method. Likewise, flexibility in the finite-element mesh was added, allowing each element to be sub-divided into an arbitrary number of sub-element at each refinement step. Lastly, GENCODE implements the *a posteriori* theory described in Section 4 as the means for computational error estimation for mesh and shape function order selection. We use GENCODE to produce all the numerical presented in this paper.

6. Numerical results

We begin by showing a numerical solution of the optimal control problem for the launch vehicle computed using GENCODE with 27 elements per phase for a total of 81 elements. In Fig. 1, we plot the position and velocity versus time, and in Fig. 2, we plot the Rodrigues parameters and the control variables. The effect of changing the thrust is seen clearly in the sharp changes in velocity while the positions states and Rodrigues parameters vary smoothly. Both the position states and velocity undergo large changes in scale. Both of the control variables vary rapidly during an initial transient while the bank angle also undergoes several rapid transient periods during the flight. In contrast, the Lagrange multipliers vary relatively smoothly with the co-states corresponding to position and velocity having values on the order of 0.1 and 1, respectively, while the co-states for time and the Rodrigues parameters are on the order of 100, see Figs. 3 and 4.

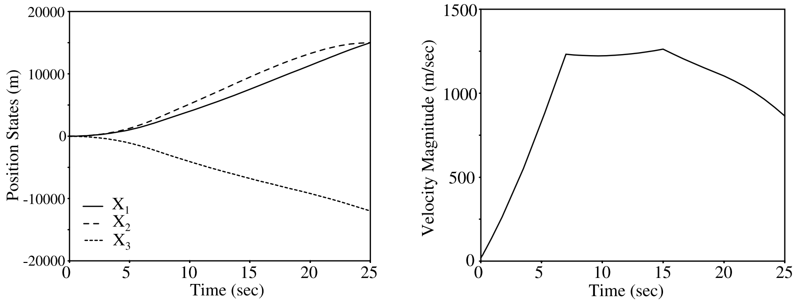

Fig. 1. Plots of position states and velocity.

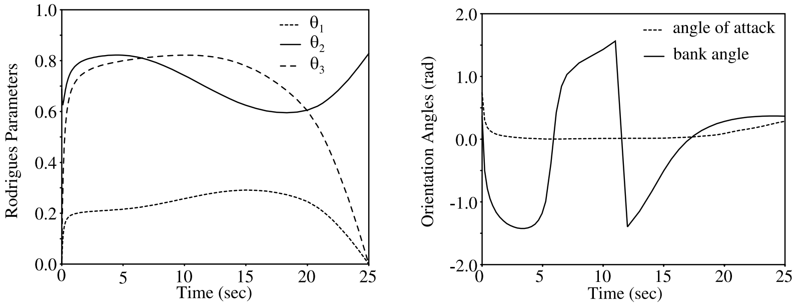

Fig. 2. Plots of Rodrigues parameters and orientation angles.

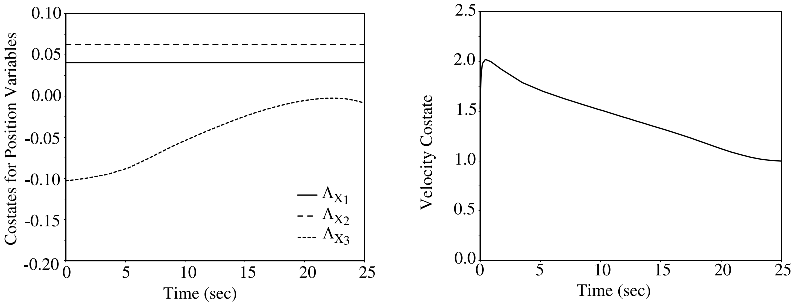

Fig. 3. Plots of the co-states for position and velocity.

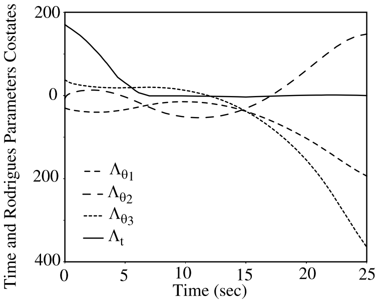

Fig. 4. Plots of the co-states for the Rodrigues parameters and time.

In Table 2, we show the initial uniform mesh input into GENCODE and the final mesh resulting from the adaptive error control with a tolerance of 20. The meshes are recorded as N_1:N_2:N_3 with N_i denoting the number of elements used in phase i of the flight path. In every case except the initial 9:9:9 mesh, the final mesh is generated after one iteration of the adaptive algorithm. We also show the error estimate computed by GENCODE corresponding to the final mesh. To test the accuracy of this estimate, we compute a very accurate "reference" solution using a piecewise quadratic approximation on a 18:8:12 mesh and then compute an approximate "error" using this reference solution. In Table 2, we record the corresponding error for each numerical solution and the error to bound ratio. Ideally, the error to bound ratio would be 1 for all computations. We find that the ratio does vary as the final mesh is altered, but the variations are acceptably small. The error estimate is predictably too conservative by a factor of 4–5, which is due to value of C determined by calibration. While this degree of over-estimation is acceptable for this application, it would be worthwhile to generate a more sophisticated calibration problem and thereby reduce the over-estimation.

Recall that both the choice of data for the dual problem and the effects of linearization mean that the error estimate indirectly depends on the accuracy of the discretization. In Fig. 5, we plot the error to bound ratios versus the refinement level for computations with 6:6:6 and 9:2:12 meshes. The estimate generally becomes more accurate as the mesh is refined and in all cases varies acceptably as the mesh is changed.

We illustrate the degree to which equidistribution is achieved by the adaptive error control in Fig. 6, where we plot the local quantities $\mathscr{S}(\psi, I_n)\mathscr{R}(X, I_n)$.

We conclude by demonstrating the gain in efficiency that results from using adaptive error control versus uniform refinement. Using the numerical reference solution computed with piecewise quadratic approximation on an 18:8:12 mesh to compute absolute errors, we plot these errors versus the number of elements for computations made with several tolerances beginning with a 9:2:12 mesh on the left in Fig. 7. To achieve a predicted error of 2, the adaptive error control uses 300 fewer elements than is required by a uniform discretization, saving about 16%. This translates to about 6.1 fewer CPU s, about a 37% reduction. On the right in Fig. 7, we plot the error versus the number of elements and the CPU time for computations made with several tolerances beginning with a 9:9:9 mesh. The gain in efficiency in this computation is even more evident.

Table 2
Table showing initial and final meshes and the error to bound ratios for various numerical solutions

Initial mesh	Final mesh	Estimate	Error	Estimate/Error
6:2:8	39:11:13	13.1	2.7	4.8
6:8:8	35:8:13	15.3	2.8	5.6
8:8:8	41:8:13	11.8	3.3	3.6
9:9:9	40:9:14	16.8	2.8	5.9
9:2:3	46:12:16	6.9	1.9	3.7
9:2:12	49:12:14	8.9	2.1	4.3
12:12:12	41:12:13	10.3	2.3	4.5

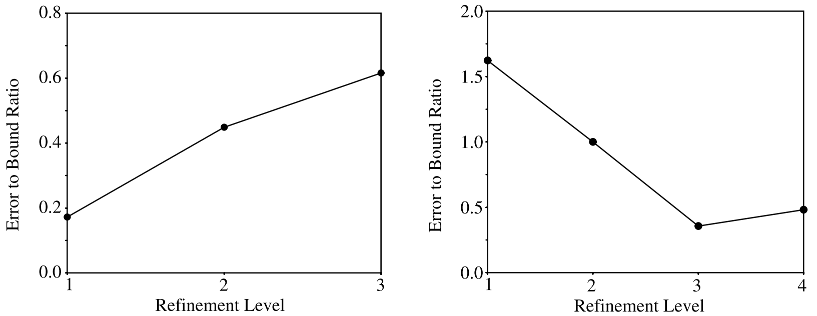

Fig. 5. Plots of the error to bound ratios versus number of elements for solutions beginning with 9:2:12 and 6:6:6 meshes.

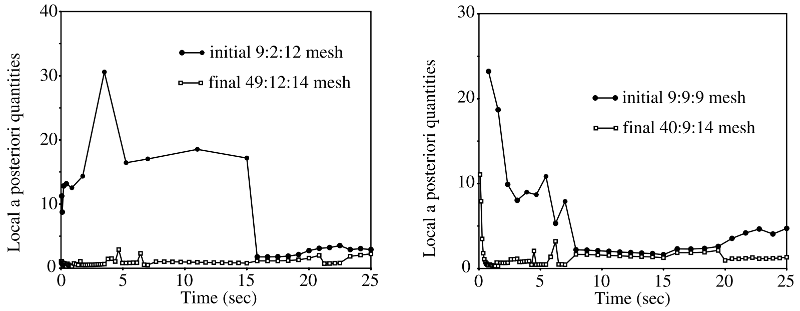

Fig. 6. Plots of the initial and final local quantities $\mathscr{S}(\psi, I_n)\mathscr{R}(X, I_n)$.

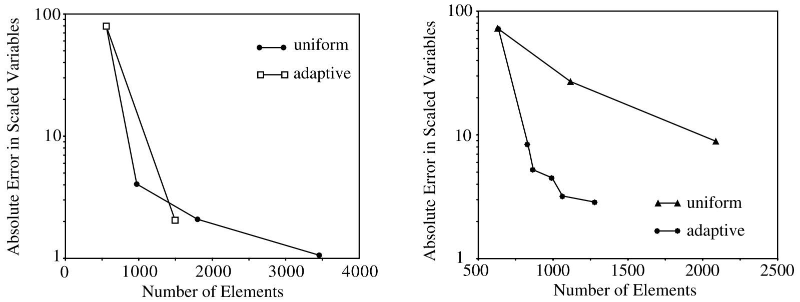

Fig. 7. Plots of the error versus number of elements in the scaled variables beginning with a 9:2:12 mesh on the left and a 9:9:9 mesh on the right.

Acknowledgements

The research of D. Estep was partially supported by the National Science Foundation, DMS 9805748. The research of D. Hodges was partially supported by the NASA-Langley Research Center, NAG-1-1435, Technical Monitor: Dr. Daniel Moerder.

References

[1] R.H. Battin, An Introduction to the Mathematics and Methods of Astrodynamics, AIAA, Washington, DC, 1987.
[2] R. Bless, Time-domain finite elements in optimal control with application to launch vehicle guidance, Ph.D. thesis, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, 1991.

[3] R. Bless, D. Hodges, Finite element solution of optimal control problems with stat-control inequality constraints, J. Guid. Control Dynam. 15 (1992) 1029–1032.

[4] R. Bless, D. Hodges, H. Seywald, Finite element method for the solution of state-constrained optimal control problems, J. Guid. Control Dynam. 18 (1995) 1036–1043.

[5] R.D. Braun, R.W. Powell, J.E. Lyne, Earth aerobraking strategies for manned return from mars, J. Spacecraft Rockets 29 (1992) 297–304.

[6] S. Brenner, L.R. Scott, The Mathematical Theory of Finite Element Methods, Springer, New York, 1994.

[7] L. Demkowicz, J. Oden, W. Rachowicz, O. Hardy, Towards a universal *hp* adaptive finite element strategy, part 1, constrained approximation and data structure, Comput. Methods Appl. Mech. Engrg. 77 (1989) 79–112.

[8] K. Eriksson, D. Estep, P. Hansbo, C. Johnson, Introduction to adaptive methods for differential equations, Acta Numer. (1995) 105–158.

[9] K. Eriksson, D. Estep, P. Hansbo, C. Johnson, Computational Differential Equations, Cambridge University Press, New York, 1996.

[10] D. Estep, A posteriori error bounds and global error control for approximations of ordinary differential equations, SIAM J. Numer. Anal. 32 (1995) 1–48.

[11] D. Estep, D. French, Global error control for the continuous Galerkin finite element method for ordinary differential equations, RAIRO Modél. Math. Anal. Numér. 28 (1994) 815–852.

[12] D. Estep, D. Hodges, M. Warner, Computational error estimation and adaptive mesh refinement for a finite element solution of launch vehicle trajectory problems, SIAM J. Sci. Comput. 21 (2000) 1609–1631.

[13] D. Estep, C. Johnson, The computability of the Lorenz system, Math. Models Methods Appl. Sci. 8 (1998) 1277–1305.

[14] D. Estep, M. Larson, R. Williams, Estimating the error of numerical solutions of systems of nonlinear reaction–diffusion equations, A.M.S. Memoirs 146 (2000) 1–109.

[15] C.R. Hargraves, S.W. Paris, Direct trajectory optimization using nonlinear programming and collocation, J. Guid. Control Dynam. 10 (1987) 338–342.

[16] D. Hodges, R. Bless, Weak Hamiltonian finite element method for optimal control problems, J. Guid. Control Dynam. 14 (1991) 148–156.

[17] D. Hodges Jr., M. Johnson, Control variables for finite element solution of missile trajectory optimization, J. Guid. Control Dynam. 18 (1995) 1208–1211.

[18] W. Rachowicz, J. Oden, L. Demkowicz, Towards a universal *hp* adaptive finite element strategy, part 3, design of $h - p$ meshes, Comput. Methods Appl. Mech. Engrg. 77 (1989) 79–112.

[19] S.A. Striepe, R.D. Braun, R.W. Powell, W.T. Fowler, Influence of interplanetary trajectory selection on mars atmospheric entry velocity, J. Spacecraft Rockets 30 (1993) 426–430.

[20] M. Warner, Numerical solutions to optimal-control problems by finite elements in time with adaptive error control, Ph.D. thesis, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, 1997.

[21] M. Warner, D.H. Hodges, Solving boundary-value problems using *hp*-version finite elements in time, Int. J. Numer. Methods Engrg. 43 (1998) 425–440.

[22] M.S. Warner, D.H. Hodges, Solving optimal control problems using *hp*-version finite elements in time, J. Guid. Control Dynam. 23 (1) (2000) 86–94.

[23] M. Warner, D.H. Hodges, Treatment of control constraints in finite element solution of optimal control problems, J. Guid. Control Dynam. 22 (1999) 358–360.