tions with either potassium chlorate or potassium nitrate as an oxidizer.

No concrete conclusions can be made concerning the effects of the change in percentage of polyvinylchloride on burn rates. The experimental evidence indicates that propellant B (the mixture with 15% PVC by weight) may be capable of producing higher burn rates than the 10% PVC propellant. However, the amount of data taken is not great enough to warrant drawing a sure conclusion.

There are many unexplored areas in the field of burn rates of propellants. The burn rates reported here are in the range between deflagration and detonation. The large variations in burn rates indicate instability in the burning process, and should be subject to further study.

Brown[1] has listed many uses for propellant formulations which would burn in the range intermediate between deflagration and detonation. Among these are explosively-actuated tools, chaff ejectors, gas generators, metal forming and welding, single-grain gun propellants, high-acceleration rockets, and bursters for materials which a detonation would destroy. These are sufficient reasons to institute a more complete search for, and investigation of, propellant formulations in the burn-rate region between deflagration and detonation.

References

[1] Rodenberger, C. A., Sawyer, M. L. and Tower, M. M., "Final Report on the Feasibility of Obtaining Hypervelocity Acceleration Using Propellant Lined Launch Tubes," Contract NAS 9-6812, Oct. 1970, Texas A&M Univ., College Station, Texas.

[2] Brown, J. A., "Explosion Phenomena Intermediate Between Deflagration and Detonation," Scientific and Technical Applications Forecast, Contract DA-49-092-ARD-140, Oct. 1967, Esso Research and Engineering Co., Linden, N. J.

[3] Steinz, J. A., Stang, P. L. and Summerfield, M., "The Burning Mechanism of Ammonium Perchlorate-Based Solid Propellants," Aerospace and Mechanical Sciences Rept. 830, Office of Naval Research Contract NR 1858 (32), Feb. 1969, Princeton Univ., Princeton, N.J.

[4] Bowden, F. P. and Yoffe, A. D., *Fast Reactions in Solids*, Butterworths, London, England, 1958, pp. 129–137.

[5] "Small Column Insulated Delays for Precision Pyrotechnic Delays and Ordinance Distribution Systems," June 1967, McCormick-Selph (a Teledyne company), Hollister, Calif.

[6] Moore, E. T., "Explosive Hypervelocity Launchers," NASA Contractor Rept. CR-982, Feb. 1968, Physics International Co., San Leandro, Calif.

SEPTEMBER 1972 AIAA JOURNAL VOL. 10, NO. 9

Comparison of Linear and Riccati Equations Used to Solve Optimal Control Problems

B. D. TAPLEY* AND W. E. WILLIAMSON†

The University of Texas at Austin, Austin, Texas

Optimal control problems can in general be formulated as nonlinear two point boundary value problems. One method of attempting to solve a nonlinear TPBVP is to iteratively solve for the solution of a corresponding linear TPBVP. By successively solving the linear TPBVP, an attempt is made to produce a trajectory which is a solution of the nonlinear problem. Such an approach is referred to as a linear perturbation method. It is reported in the literature that the linear methods suffer from numerical instabilities and that the unstable characteristics can be improved by uncoupling the linear differential equations with a Riccati-type transformation of variables. In this paper, five different methods of solving the linear TPBVP are discussed. Two of these methods involve integrating a coupled set of linear differential equations. The other three require the integration of a Riccati equation and associated linear equations and quadratures. These systems are used to calculate optimal solutions for the Brachistochrone problem and for a three dimensional Apollo-type re-entry problem. The results show that, for the examples considered, the linear system of equations is more effective in obtaining the solutions to nonlinear two point boundary value problems than algorithms based on the Riccati transformation.

Introduction

NECESSARY conditions for unconstrained optimal trajectories can in general be reduced to a nonlinear two point boundary value problem (TPBVP). Attempts to solve the nonlinear TPBVP usually involve successively solving a linear TPBVP. The linear TPBVP may be solved either by superimposing solutions of a system of linear differential equations or by the integration of a Riccati matrix and associated linear differential equations and quadratures. Much of the recent literature has favored the integration of Riccati equations instead of linear equations because of the improved stability properties.[1,2] For many optimization problems, the linear equations have eigenvalues which vary considerably in magnitude and are often positive. If the equations are integrated over a sufficiently long time interval, instability is a definite problem. For linear problems, however, the Riccati equation can be shown to be asymptotically stable in some regions where the linear equations are unstable.[2] Hence, it is reported that it is easier to integrate the system of Riccati equations than linear equations for many problems.

The purpose of this paper is to present a comparison between two linear systems of differential equations and three Riccati systems of equations which may be used to solve linear TPBVP. The standard perturbation method[3,4] and the adjoint method,[5] which integrate linear equations, are considered. Then two Riccati methods are derived, which in some cases are equivalent to the perturbation and adjoint methods, respectively. The standard Riccati method[2,6] is considered also. Finally, all of the methods are compared numerically for two example problems. A simple example, the Brachistochrone problem, is considered

Received December 17, 1971; revision received April 13, 1972.

Index categories: Entry Vehicle Dynamics and Control; Navigation, Control, and Guidance Theory.

* Chairman and Professor, Department of Aerospace Engineering and Engineering Mechanics. Member AIAA.

† Assistant Professor, Department of Aerospace Engineering and Engineering Mechanics.

to illustrate the application of the methods. Then a complex three-dimensional re-entry problem for an Apollo-type space vehicle is considered. The comparisons indicate that despite the instability of the linear equations they can often be used to calculate solutions to the TPBVP when the Riccati methods cannot.

Reduction of Optimization Problem to TPBVP

The optimal control problem considered here can be stated as follows: minimize

$$J = G(x_f, t_f) + \int_{t_0}^{t} Q(x, u, t)\, dt \tag{1}$$

subject to

$$\dot{x} = f(x, u, t) \tag{2}$$

$$x(t_0) = c, \quad M(x_f, t_f) = 0 \tag{3}$$

where x is an n vector of state variables, u is an m vector of control variables, M is a p vector of terminal constraints, f is an n vector of derivatives of x, c is an n vector of specified initial states, G is a scalar, and Q is a scalar. The initial time t_0 is assumed fixed and the final time t_f is free.

Under the assumptions that x and u are defined on completely open regions and that Q and f along with their first and second derivatives are continuous in the interval of interest, the necessary conditions for an extremal trajectory can be stated as follows[7]: everywhere in the interval $t_0 \leqq t \leqq t_f$,

$$\dot{x} = H_\lambda^T, \quad \dot{\lambda} = -H_x^T \tag{4}$$

and

$$H_u = 0, \quad H_{uu} \geqq 0 \tag{5}$$

where H is the variational Hamiltonian, $H = \lambda^T f + Q$ and λ is an n vector of multipliers. Subscripts of x, λ, u and t denote partial derivatives. The superscript T denotes transpose. At t_0 it is required that

$$x(t_0) = c \tag{6}$$

and at t_f,

$$M(x_f) = 0 \tag{7}$$

$$-\lambda^T(t_f) + \nu^T M_{x_f} + G_{x_f} = 0 \tag{8}$$

$$G_{t_f} + \nu^T M_{t_f} + H(t_f) = 0 \tag{9}$$

where ν is a p vector of multipliers. Using the appropriate p equations from Eq. (8), it is possible to solve for the ν vector. Then ν may be eliminated from Eq. (9). This allows the terminal conditions, Eqs. (7–9) to be written as

$$h(x_f, \lambda_f, t_f) = 0 \tag{10}$$

where h is an $n+1$ vector. If the final time is fixed, then h is an n vector. Thus, under the assumed conditions, an extremal trajectory must satisfy Eqs. (4–6 and 10).

These conditions indicate that the solution to the optimization problem is obtained by solving a nonlinear two-point boundary value problem. Thus if Eq. (5) can be used to determine u as an explicit function of x and λ, then this function may be used to eliminate u from Eqs. (4). Then Eqs. (4) may be written as

$$\dot{z} = F(z, t) \tag{11}$$

where z is a $2n$ vector defined so that $z^T = (x^T : \lambda^T)$. The $2n$ vector $F(z, t)$ describes the time-rate of change of the state variables and multipliers with the control eliminated by using the classical optimality condition. Specified boundary conditions for the TPBVP are Eqs. (6) and (10).

Since Eq. (11) represents a system of $2n$-nonlinear differential equations, numerical methods are used to obtain solutions to the nonlinear TPBVP. In using such methods, values for the unknown boundary conditions must be guessed at either t_0 or t_f. In the following discussion, all unknown conditions are guessed at t_0. This involves guessing values for the n-unknown multipliers λ_0 since the initial state is known. A nominal trajectory is then produced by using the specified values of x_0 and the guessed values of λ_0 to integrate Eq. (11) to a guessed final time, if the final time is not specified. In general this nominal will not satisfy the terminal boundary conditions. Thus Eq. (10) will not be equal to zero. Some procedure is then used to calculate corrections to λ_0 and t_f which will drive the nominal terminal conditions toward the specified terminal conditions, i.e., will cause Eq. (10) to approach zero.

All of the methods considered here to calculate the corrections to λ_0 and t_f are derived by linearizing the TPBVP. If the differential equations are linearized then

$$\delta\dot{z} = A\delta z \tag{12}$$

where $\delta z = z^{i+1} - z^i$ and $A = \partial F/\partial z$ evaluated along the ith trajectory. The matrix A is a $(2n \times 2n)$ matrix. Eq. (12) may be written in terms of δx and $\delta\lambda$ as

$$\delta\dot{x} = A_{11}\delta x + A_{12}\delta\lambda, \quad \delta\dot{\lambda} = A_{21}\delta x + A_{22}\delta\lambda \tag{13}$$

where A_{11}, A_{12}, A_{21}, and A_{22} are the appropriate $(n \times n)$ submatrices of the A matrix. The unsatisfied boundary conditions are linearized so that

$$\Delta h = (\partial h/\partial z_f)\delta z_f + \dot{h}\Delta t_f \tag{14}$$

where $\Delta h = h^{i+1} - h^i$. Here $\partial h/\partial z_f$ and $\dot{h}$ are evaluated along the ith trajectory. If h^{i+1} is set equal to its desired value, zero, then $\Delta h = -h^i$. The final value of δz, i.e., δz_f, will depend only on $\delta\lambda_0$ since $\delta x_0 = 0$ and $\delta t_0 = 0$. Hence, Eqs. (12) and (14) constitute a linear TPBVP in which the unknowns are $\delta\lambda_0$ and Δt_f. That is, the solution to the linear TPBVP requires calculating values for $\delta\lambda_0$ and Δt_f to satisfy Eq. (14). These values are used to iterate toward a solution of the nonlinear TPBVP. Five different methods of solving this linear TPBVP will now be discussed.

One standard approach to this problem is the perturbation method discussed in Refs. 3 and 4. This method solves Eq. (12) directly by superimposing solutions corresponding to unit perturbation in each of the guessed values of the initial multipliers. Thus the solution to Eq. (12) is written as

$$\delta z_f = \Phi(t_f, t_0)\delta\lambda_0 \tag{15}$$

where

$$\dot{\Phi}(t, t_0) = A\Phi(t, t_0), \quad \Phi(t_0, t_0) = \begin{bmatrix} 0 \\ \hline I \end{bmatrix} \tag{16}$$

and Φ is a $2n \times n$ matrix. I and O are then $n \times n$ identity and null matrices, respectively. Equation (14) can then be written as

$$\Delta h = (\partial h/\partial z_f)\Phi(t_f, t_0)\delta\lambda_0 + \dot{h}\Delta t_f \tag{17}$$

Equation (17) constitutes a linear set of $n+1$ algebraic equations for the changes in the $n+1$ guessed variables λ_0 and t_f. This system is solved for $\delta\lambda_0$ and Δt_f to drive h to zero in a linear sense. Note that, analytically, each column of the Φ matrix is linearly independent of the other columns. If instability is a problem, however, then the columns of Φ may not be numerically linearly independent due to the finite word length of the computer. In this case, it may be difficult to numerically solve Eq. (17) for values of $\delta\lambda_0$ and Δt_f, which are meaningful.

An alternate procedure which can be used to solve the linear TPBVP is the adjoint method.[5] Instead of solving Eq. (12) directly with the Φ matrix, its adjoint equation, $\dot{y} = -A^T y$, is solved. The variable y is a $2n$ vector of adjoint variables. If the solution to this equation is written as

$$\dot{\theta}(t, t_f) = -A^T\theta(t, t_f), \quad \theta(t_f, t_f) = \left(\frac{\partial h}{\partial z_f}\right)^T \tag{18}$$

where θ is a $(n+1) \times 2n$ matrix, then Eq. (14) may be written as

$$\Delta h = \theta_2^T(t_0, t_f)\delta\lambda_0 + \dot{h}\Delta t_f \tag{19}$$

where $\theta^T = (\theta_1^T : \theta_2^T)$. Both θ_1 and θ_2 are $(n+1) \times n$ matrices. Here $\theta_1 = \theta_1(t, t_f)$, etc. Then Eq. (19) is solved for $\delta\lambda_0$ and Δt_f. Note that again a linear system is integrated and the instability of the linear equations may be a problem.

Since instability may be a problem for both the linear perturbation and adjoint equations, a method of solving the linear TPBVP without integrating the linear equations would seem to be desirable. One approach which accomplishes this purpose is implemented by using a Riccati transformation to uncouple the δz variables.[1] Three methods based on the Riccati transformation are described.

The first method will be referred to as a forward Riccati method. To develop this method, define a Riccati transformation of the form

$$\delta x = W\delta\lambda \tag{20}$$

where W is an $n \times n$ matrix. Boundary conditions for W require $W_0 = 0$ since $\delta x_0 = 0$. In the usual manner,[1] a differential equation is derived for W by differentiating Eq. (20) and requiring that Eq. (20) and its derivative be satisfied everywhere in the time interval of interest. Equations (13) and (20) are used to eliminate $\delta\dot{x}$, $\delta\dot{\lambda}$, and δx from the derivative of Eq. (20). The condition which W must satisfy to achieve this end is the following:

$$\dot{W} = A_{11}W + A_{12} - W(A_{21}W + A_{22}) \tag{21}$$

If W is integrated from t_0 to t_f, then $\delta x_f = W_f\delta\lambda_f$ from Eq. (20) and Eq. (14) may be written as

$$\Delta h = (\partial h/\partial z_f)(W_f/I)\delta\lambda_f + \dot{h}\Delta t_f \tag{22}$$

Eq. (22) is then solved for $\delta\lambda_f$ and Δt_f. Thus corrections are calculated to $\delta\lambda_f$, but the correction for $\delta\lambda_0$ is needed. This correction may be calculated in one of two manners. Using Eqs. (20) and (13) the differential equation for $\delta\dot{\lambda}$ may be written as

$$\delta\dot{\lambda} = (A_{21}W + A_{22})\delta\lambda \tag{23}$$

Hence this equation may be integrated backward from t_f to t_0 using the specified value of $\delta\lambda_f$ obtained by solving Eq. (22). This gives the desired value of $\delta\lambda_0$. This procedure requires Eq. (23) to be integrated backwards from t_f to t_0 and hence the $n \times n$ matrix $(A_{21}W + A_{22})$ must be stored during the forward integrations. An alternate method which does not require the backward integrations will now be described. Since Eq. (23) is a linear equation, the transition matrix approach may be used to solve the equation. Thus define

$$\dot{S} = (A_{21}W + A_{22})S \tag{24}$$

where S is an $n \times n$ matrix and $S_0 = I$. Then $\delta\lambda_f = S_f\delta\lambda_0$ and Eq. (22) is written as

$$\Delta h = \frac{\partial h}{\partial z_f}\left(\frac{W_f S_f}{S_f}\right)\delta\lambda_0 + \dot{h}\Delta t_f \tag{25}$$

This method requires the integration of more equations than the backward method, but does not require any additional storage for the computer program.

Note that by comparing Eqs. (17) and (25) it is obvious that

$$WS = \Phi_1, \quad S = \Phi_2 \tag{26}$$

where $\Phi^T = (\Phi_1^T : \Phi_2^T)$.

These equations provide a means of checking the numerical integration accuracy obtained by using either the transition matrix or the forward Riccati method. For the optimization problem, A_{12} and A_{21} are symmetric matrices. Also $A_{22} = -A_{11}^T$. Hence W is symmetric. Thus, in general, the Riccati methods require the integration of fewer equations than the methods which integrate linear equations.

An alternate method based on a different Riccati transformation will now be derived. This transformation will be based on perturbations of the terminal conditions and will be referred to as a backward Riccati method. Before the method is derived, the terminal boundary conditions must be rewritten. Let the h vector consist of the n boundary conditions in Eqs. (7) and (8). Thus Eq. (9) is removed from the h vector and will be handled separately. Define k to be this last boundary condition which contains the Hamiltonian. Since the terminal boundary conditions contained in the h vector are assumed to be linearly independent, then the rank of the $2n \times n$ matrix $\partial h/\partial z_f$ is n. Define n of the z variables to be y and the other n to be μ. The z variables are then rearranged such that if h is linearized then

$$\Delta h = (\partial h/\partial y_f)\delta y_f + (\partial h/\partial \mu_f)\delta\mu_f + \dot{h}\Delta t_f \tag{27}$$

where the matrix $\partial h/\partial \mu_f$ is of rank n. Thus Eq. (27) may be solved for $\delta\mu_f$ as

$$\delta\mu_f = (\partial h/\partial \mu_f)^{-1}[\Delta h - (\partial h/\partial y_f)\delta y_f - \dot{h}\Delta t_f] \tag{28}$$

This suggests a Riccati transformation of the form

$$\delta\mu = W\delta y + s\Delta t_f + l \tag{29}$$

where W is an $(n \times n)$ matrix and s and l are n vectors. A similar transformation is considered for k such that

$$\Delta k = g^T\delta y + q\Delta t_f + r \tag{30}$$

where g is an n vector and q and r are scalars. The differential equations for δy and $\delta\mu$ are written as

$$\delta\dot{y} = B_{11}\delta y + B_{12}\delta\mu, \quad \delta\dot{\mu} = B_{21}\delta y + B_{22}\delta\mu \tag{31}$$

where B_{11}, B_{12}, B_{21}, and B_{22} are $n \times n$ matrices. The elements of the B matrices are the rearranged elements of the A matrix consistent with the definitions for y and μ. The following differential equations are obtained by differentiating Eqs. (29) and (30) and using Eq. (29) and (31) to eliminate $\delta\dot{y}$, $\delta\dot{\mu}$, and $\delta\mu$ from them:

$$\begin{aligned}
\dot{W} &= B_{12} - WB_{22} + (B_{11} - WB_{21})W, \quad \dot{s} = (B_{11} - WB_{21})s \\
\dot{l} &= (B_{11} - WB_{21})l, \quad \dot{g}^T = -g^T(B_{21}W + B_{22}) \\
\dot{q} &= -g^TB_{21}s, \quad \dot{r} = -g^TB_{21}l
\end{aligned} \tag{32}$$

Boundary conditions for W, s, and l are obtained by requiring that Eqs. (29) and (30) be satisfied at the final time. The terminal values of Eqs. (29) and (30) are compared with Eq. (28) and a linearized revision of Eq. (9) with v eliminated to obtain the following conditions:

$$\begin{aligned}
W(t_f) &= -h_{\mu_f}^{-1}h_{y_f}, \quad s(t_f) = -h_{\mu_f}^{-1}\dot{h} \\
l(t_f) &= h_{\mu_f}^{-1}\Delta h, \quad g^T(t_f) = \partial k/\partial y_f - (\partial k/\partial \mu_f)h_{\mu_f}^{-1}h_{y_f} \\
q(t_f) &= \dot{k} - (\partial k/\partial \mu_f)h_{\mu_f}^{-1}\dot{h}, \quad r(t_f) = (\partial k/\partial \mu_f)h_{\mu_f}^{-1}\Delta h
\end{aligned} \tag{33}$$

This method requires integration of Eq. (11) from t_0 with the guessed values for λ_0 and t_f. Then Eqs. (32) are integrated backward from t_f to t_0 using the boundary conditions specified by Eq. (33). Both Eqs. (29) and (30) are then evaluated at t_0. If δx_0 is set equal to zero in these equations, they may then be solved for $\delta\lambda_0$ and Δt_f.

A fifth procedure for solving optimal control problems is the standard Riccati method.[2,6,8] Note that for problems considered here, the control has been eliminated from the formulation. The Riccati method considered here is used to solve only a two-point boundary value problem and does not involve calculating changes to a control program as the sweep method[2] does.

The standard Riccati transformation proposed in Ref. (10) is of the form

$$\delta\lambda = W\delta x + S\Delta\lambda_f + s\Delta t_f, \quad \Delta h = R\delta x + r\Delta\lambda_f + T\Delta t_f \tag{34}$$

where W and S are $n \times n$ matrices, s is an n vector, R and r are $(n+1) \times n$ matrices, and T is an $(n+1)$ vector. Here h is again the full $(n+1)$ vector of terminal conditions. Boundary conditions require that

$$\begin{aligned}
&W(t_f) = 0, \quad S(t_f) = I, \quad s(t_f) = \dot{\lambda}_f \\
&R(t_f) = \partial h/\partial x_f, \quad r(t_f) = \partial h/\partial \lambda_f, \quad T(t_f) = (\partial h/\partial x_f)\dot{x}_f + \partial h/\partial t_f
\end{aligned} \tag{35}$$

By differentiating Eqs. (34) and using Eq. (13), the following differential equations may be derived for the Riccati variables:

$$\begin{aligned}
\dot{W} &= (A_{22} - WA_{12})W - WA_{11} + A_{21}, \quad \dot{S} = (A_{22} - WA_{12})S \\
\dot{s} &= (A_{22} - WA_{12})s, \quad \dot{R} = -R(A_{11} + A_{12}W) \\
\dot{r} &= -RA_{12}S, \quad \dot{T} = -RA_{12}s
\end{aligned} \tag{36}$$

This method requires that Eqs. (11) be integrated from t_0 to t_f and then Eqs. (36) be integrated from t_f to t_0. Then the equation for Δh is evaluated at t_0 and solved for $\Delta\lambda_f$ and Δt_f from

$$\Delta h = r(0)\Delta\lambda_f + T(0)\Delta t_f \tag{37}$$

and $\delta\lambda_0$ is calculated from Eq. (34) as

$$\delta\lambda_0 = S(0)\Delta\lambda_f + s(0)\Delta t_f \tag{38}$$

Boundary conditions for the perturbation and forward Riccati variables are known at t_0. Boundary conditions for all other methods are known at t_f. Thus the methods are programed in a slightly different manner. In all cases, values for the unknown initial conditions, λ_0, and the final time (if free) are guessed. For the perturbation and forward Riccati methods, the nonlinear equations and other appropriate equations are integrated from t_0 to t_f. Corrections are then calculated to λ_0 and t_f. The other methods, as they are formulated, require the integration of the linear equations or Riccati equations from t_f to t_0. The unknown variables, however, are guessed at t_0. Thus the nonlinear equations may be integrated from t_0 to t_f and the values of the variables stored at every integration step. Then the linear or Riccati equations may be integrated back from t_f to t_0 using the stored values of z to form the necessary coefficients for the integration. Alternately the nonlinear equations may be integrated from t_0 to t_f and then integrated backwards from t_f to t_0 simultaneously with the other equations. This increases the integration time but substantially reduces the storage requirements for long trajectories. This procedure will be followed for the examples considered later.

Note that for each of the methods described above, h^{i+1} will not be zero identically because of the errors involved in the linearization process. Hence, the procedure is repeated until $\|h^{i+1}\| \leqq \varepsilon$ where ε is a small positive number. Furthermore, the condition $h^{i+1} = 0$ used to specify Δh in Eqs. (22, 25, 27, and 34) may require a large correction for λ_0. This large value for $\delta\lambda_0$ may violate the assumption of small perturbations used in the linearization of Eqs. (10) and (11). In such a case, divergence of the iteration process will occur. To prevent this occurrence, the condition $h^{i+1} = (1-\alpha)h^i$ is used where α satisfies the condition $0 \leqq \alpha \leqq 1$. Ideally, $\alpha = 1$ would be used, but in the earlier stages of the iteration process, a value of $\alpha = 0.25$ or less will lead to a more stable iteration process. As $\|h^i\|$ approaches zero, the value of α can be increased to approach one.

Numerical Examples

The five methods described earlier will now be used to compute optimal trajectories for two example problems. The methods considered are the perturbation method, adjoint method, standard Riccati method, forward Riccati method, and backward Riccati method. A simple example problem, the Brachistochrone problem, is considered first. Then a three-dimensional re-entry problem for an Apollo-type space vehicle is considered as a second example. All numerical computation is performed on the CDC 6600 computer at The University of Texas at Austin. Numerical integration is performed using a fourth-order variable step-size Adams predictor-corrector using a Runge-Kutta starter.

Example 1: Brachistochrone

This example is considered in detail in many references including Ref. (8). The statement of the problem is as follows:

Minimize

$$I = t_f \tag{39}$$

subject to

$$\dot{x}_1 = (x_2)^{1/2}\cos u, \quad \dot{x}_2 = (x_2)^{1/2}\sin u \tag{40}$$

and

$$t_0 = 0.1, \quad x_1(t_0) = 1.2436363 \times 10^{-4}$$
$$x_2(t_0) = 2.4953532 \times 10^{-3} \tag{41}$$

and

$$x_1(t_f) = 1.0 \tag{42}$$

The control variable is u.

Necessary conditions for a minimizing trajectory require that

$$\dot{\lambda}_1 = 0$$
$$\dot{\lambda}_2 = -[1/2(x_2)^{1/2}](\lambda_1\cos u + \lambda_2 \sin u) \tag{43}$$

and that

$$\cos u = -\lambda_1/(\lambda_1^2 + \lambda_2^2)^{1/2}, \quad \sin u = -\lambda_2/(\lambda_1^2 + \lambda_2^2)^{1/2} \tag{44}$$

and

$$\lambda_2(t_f) = 0, \quad H_f + 1 = 0 \tag{45}$$

The TPBVP consist of the four differential equations, Eqs. (40) and (43) with u eliminated by using Eq. (44). Boundary conditions consist of Eqs. (41, 42 and 45).

In order to start the iteration procedure, guesses are made for $\lambda_1(t_0)$, $\lambda_2(t_0)$ and t_f. One set of guessed conditions and converged values are shown in Table 1. The number of iterations required by each method to obtain a converged solution is shown in Table 2. Several other guesses for λ_0 and t_f were considered for this example problem. The results obtained were always the same as those shown and are hence omitted. For every trajectory considered, equivalent results were obtained by all of the methods. The main conclusions obtained from this example would seem to be that the Brachistochrone is a sufficiently simple problem so that any second order method may be used to produce optimal trajectories for any reasonable guesses of λ_0 and t_f, and that the additional Riccati-type algorithms are at least as effective as the standard Riccati methods.

Table 1 Nominal and converged values for λ_0 and t_f for the Brachistochrone problem

Variable	Nominal	Converged
λ_x	− 1.5	− 1.25332667
λ_x	− 10.0	− 19.9793405
t_f	2.0	2.5066306923

Table 2 Number of iterations required for convergence of example problems

Problem	Perturbation	Adjoint	Standard Riccati	Backward Riccati	Forward Riccati
Brachistochrone	8	8	8	8	8
Re-entry case 1	8	8	...	...	...
Re-entry case 2	6	6	...	6	...
Re-entry case 3	15	15	...	15	...

Example 2: Re-Entry Problem

The second example considered is substantially more difficult than the Brachistochrone problem. The problem of optimizing a three-dimensional, high-energy, Apollo-type re-entry trajectory for the return from a lunar mission is considered.

Assuming that the vehicle is moving in an inverse square gravitational force field, the differential equations governing the re-entry trajectory are

$$\begin{aligned}
\dot{r} &= V\sin\gamma \\
\dot{\theta} &= (V\cos\gamma\cos\psi)/(r\cos\varphi) \\
\dot{\varphi} &= (V\cos\gamma\sin\psi)/r \\
\dot{V} &= (\mu/r^2)\sin\gamma - D \\
\dot{\gamma} &= [-(\mu/r^2V) + (V/r)]\cos\gamma + (L/V)\cos\beta \\
\dot{\psi} &= -(V/r)\cos\gamma\cos\psi\tan\varphi - (L\sin\beta/V\cos\gamma)
\end{aligned} \tag{46}$$

where r is the radial distance from the Earth's center of mass to the vehicle mass-center, θ is the longitude, φ is the latitude, V is the magnitude of the velocity, γ is the flight path angle, ψ is the heading angle, μ is the gravitational constant. The roll angle of the vehicle, β, is the control variable. Also, L and D are the lift and drag per unit mass.

Since deceleration and heating are the basic problems associated with the re-entry trajectory, the control variable, β, is to be chosen to minimize the integral I, where

$$I = \int_{t_0}^{t_f} [\tfrac{1}{2}(C_L^2 + C_D^2)^{1/2} S^* \rho V^2 + \lambda_C \rho^{1/2} V^3]\, dt \tag{47}$$

The first term is the aerodynamically induced deceleration and the second term is proportional to the convective heating rate. The variables are defined as follows: C_L is the lift coefficient, C_D is the drag coefficient, S^* is the reference area per unit mass,

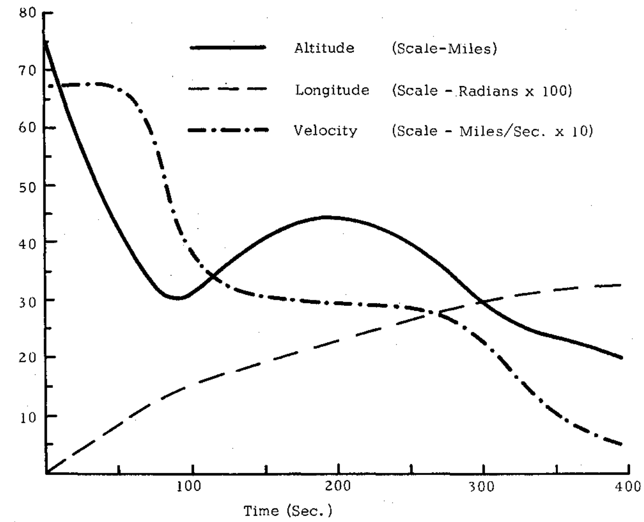

Fig. 1 State variables r, θ, and V for optimal re-entry trajectory.

ρ is the density of the atmosphere assumed to vary exponentially, and λ_C is a constant which can be used to give relative weight to the deceleration and the heating.

Boundary conditions consistent with the return from a lunar mission require that $r_0 = 4035.75758$ miles, $\theta_0 = 0.0$ rad, $\varphi_0 = 0.0$ rad, $V_0 = 6.81818182$ miles/sec, $\gamma_0 = -0.1134464$ rad, $\psi_0 = 0.0$ rad, $\theta_f = 0.33$ rad, $\varphi_f = -0.025$ rad, and $V_f = 0.5$ miles/sec. The initial time t_0 is assumed to coincide with the initiation of the re-entry maneuver while the final conditions are compatible with the deployment of a drogue chute.

The TPBVP for this example consists of 12 first order nonlinear equations. The twelve equations consist of the 6 state equations and six equations for the Lagrange multipliers. Also additional terminal conditions are obtained from transversality conditions. A detailed complete problem statement along with a reduction to the TPBVP is shown in Ref. 9. The state histories for the trajectory described above are shown in Figs. 1 and 2. The control, acceleration, and heat rate are shown in Fig. 3.

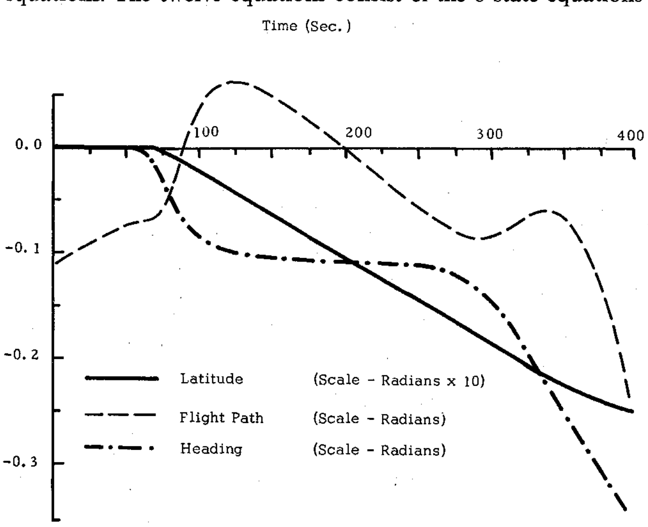

Fig. 2 State variables ϕ, γ, and ψ for optimal re-entry trajectory.

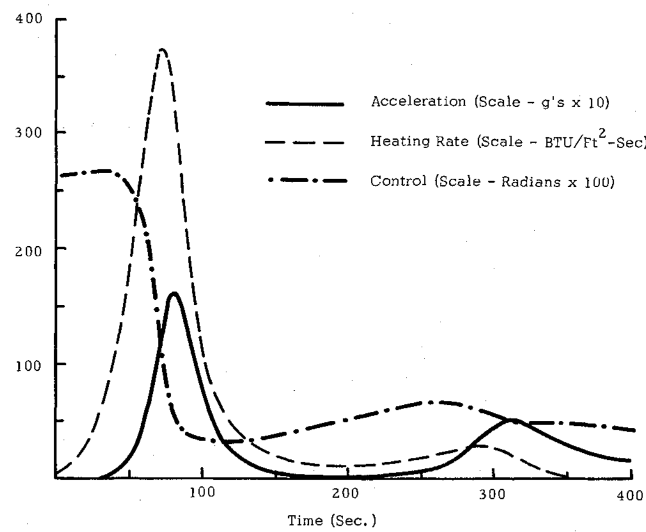

Fig. 3 Control, acceleration, and heating rate for optimal re-entry trajectory.

In order to start the iteration procedure, guesses are required for λ_0 and t_f. One set of nominal values is shown in Table 3. Converged values for λ_0 and t_f are also shown in Table 3 for various values of θ_f. Case 1 corresponds to the conditions described earlier. Case 2 changes only the specified longitude and requires that $\theta_f = 0.34$ rad. The initial nominal values of λ_0 and t_f used to converge case 2 are the converged values for case 1. Case 3 requires $\theta_f = 0.4$ rad, $\varphi_f = -0.025$ rad, and $V_f = 0.2$ miles/sec. Again, converged values for λ_0 and t_f from case 1 are used as starting values to converge the trajectory shown as case 3.

The number of iterations required by each method to converge to the trajectories described above is shown in Table 2. For case 1, it was impossible to integrate any of the Riccati equations along the nominal trajectory. In all cases, values for the exponents of the Riccati variables become larger than the computer can handle, i.e., the exponent was larger than approximately 320. This terminates the iteration procedure. The perturbation and adjoint method both converged, however.

For case 2, the perturbation, adjoint, and backward Riccati methods are able to converge to the new optimal. The standard Riccati variables could be integrated only along the nominal (in this case the optimal trajectory defined as case 1), but could not be integrated after the first iteration. Thus they could not be integrated along a nonoptimal trajectory. The forward Riccati

Table 3 Nominal and converged values for λ_0 and t_f for the re-entry problem

Variable	Initial nominal	Case 1	Case 2	Case 3
λ_{r_0}	3.98397469E-3	−1.24748687E-3	−4.012364492E-4	3.03171426E-3
λ_{θ_0}	−8.95206071	−9.17150992	−8.76383516	−7.46764616
λ_{ϕ_0}	8.31654552	2.65110657E-1	2.39511202E-1	1.62289265E-1
λ_{γ_0}	2.52107557	2.35519683	2.32059454	2.2486124
λ_{V_0}	1.69208133E-1	1.38346235E-1	1.38242815E-1	1.4457471E-1
λ_{ψ_0}	3.25899074	8.828288353	8.50955641	6.6587464
t_f	380.0	391.807	410.50914	489.132

method could not be integrated along the initial nominal.

Although case 2 corresponds to the calculation of a nearby optimal trajectory, case 3 requires relatively large changes in the trajectory. For case 3, both the perturbation and adjoint methods are able to converge to the desired optimal. Again, it is impossible to integrate the forward Riccati equations and, after the first correction, the Riccati matrix associated with the standard Riccati method could not be integrated. The backward Riccati method produced equivalent results to the perturbation and adjoint methods and converged in 15 iterations.

Several other nominal trajectories were considered for the re-entry problem. The results for all cases are similar to those cited above. Thus if the guesses for λ_0 and t_f are sufficiently close to the optimal values, the perturbation method and adjoint method converge and produce equivalent results. If the Riccati matrix for the backward Riccati method can be integrated, then the backward Riccati method converges and produces results equivalent to the perturbation and adjoint methods. For several nominals considered, however, the perturbation and adjoint methods converged but the backward Riccati variables could not be integrated.

It was impossible to integrate the Riccati variables associated with the forward Riccati method over any nominal trajectory considered. The standard Riccati variables could be integrated only along an optimal trajectory. If reasonable changes in terminal conditions were requested (i.e., given an optimal and then require the method to converge to a nearby optimal), the Riccati variables could not be integrated after the first iteration. Thus, after the nominal trajectory is not an optimal, the standard Riccati variables cannot be integrated numerically.

The conclusions reached in this study were obtained using necessary conditions in which the control variable is eliminated by using the classical optimality conditions, Eqs. (5). However, similar conclusions were reported in Ref. 10, which discusses a study which did not eliminate the control variable, but calculated a correction, δu, to a reference value of the control variable, u^i, at each point in the time interval of interest. The algorithms developed in this study agree with the procedure adopted in Refs. 2 and 8. In Ref. 10, as in the present study, it is concluded that the linear perturbation methods are better suited to obtaining solutions to complex nonlinear TPBVP than the Riccati-based algorithms.

Summary and Discussion

The results presented here indicate that the frequently stated conclusion that, for solving nonlinear TPBVP, Riccati equations should be integrated instead of linear equations, particularly for dissipative systems, is not correct. For the simple Brachistochrone example, considered in the previous discussion, any of the methods discussed may be used to calculate optimal trajectories. For the Apollo-type re-entry problem, however, the use of the linear equations, either perturbation or adjoint, produce substantially better results than any of the Riccati methods considered. For the re-entry problem, the backward Riccati method appears to have the best convergence characteristics of all the Riccati methods considered.

Eigenvalues were calculated for the A matrix associated with the re-entry problem and are shown in Ref. 9. Positive and negative eigenvalues exist over the entire trajectory. Thus the linear equations associated with the perturbation and adjoint methods are unstable. They can, however, be integrated and used to converge solutions to the TPBVP associated with optimal re-entry trajectories. It appears that the 14-digit word length of the CDC 6600 Computer and the fourth-order variable step-size integration routine used are sufficient to produce accurate integration of the unstable linear equations.

It was extremely difficult to integrate any of the Riccati equations considered here. They appear to be very "unstable" from the standpoint of numerical integration characteristics. Thus for the examples considered here, the linear equations appear to be superior to the Riccati equations. Similiar conclusions were reached in Ref. 10 in a study which did not use the optimality criteria to eliminate the control variable from the problems.

References

[1] Rybicki, G. G. and Usher, P. D., "The Generalized Riccati Transformation as a Simple Alternative to Invariant Imbedding," *The Astrophysical Journal*, Vol. 146, No. 3, Dec. 1966.

[2] McReynolds, S. R., "A Successive Sweep Method for Solving Optimal Programming Problems," Ph. D. dissertation, 1965, Harvard Univ., Cambridge, Mass.

[3] Goodman, T. R. and Lance, G. N., "The Numerical Integration of Two-Point Boundary Problems," *Mathematical Tables and Other Aids to Computation*, Vol. 10, No. 54, 1956.

[4] Tapley, B. D. and Lewallen, J. M., "Comparison of Several Optimization Methods," *Journal of Optimization Theory and Application*, Vol. 1, No. 1, July 1967.

[5] Jurovics, S. A. and McIntyre, J. E., "The Adjoint Method and Its Application to Trajectory Optimization, *ARS Journal*, Vol. 32, No. 9, 1962.

[6] Colunga, D., "A Modified Sweep Method for Control Optimization," AMRL Rept. 1016, 1970, Univ. of Texas, Austin, Texas.

[7] Hestenes, M. R., *Calculus of Variations and Optimal Control Theory*, Wiley, New York, 1966, Chap. 7.

[8] Bryson, A. E. and Ho, Y. C., *Applied Optimal Control*, Blaisdell, Waltham, Mass., 1969.

[9] Williamson, W. E., "Optimal Three Dimensional Re-entry Trajectories for Apollo Type Vehicles," AMRL Rept. 1013, 1970, Univ. of Texas, Austin, Texas.

[10] Williamson, W. E. and Tapley, B. D., "Riccati Transformations for Control Optimization Using the Second Variation," *Proceedings of the* 1970 *IEEE Symposium on Adaptive Processes, Decision and Control*, Dec., 1970, pp. XXI 3.1–3.7.