AIAA 79-1624R

Numerical Computation of Optimal Atmospheric Trajectories

Charles Hargraves,* Forrester Johnson,† Stephen Paris,‡ and Ian Rettie§
The Boeing Company, Seattle, Wash.

A second-order direct trajectory optimization method in which the state time history is described by Chebychev polynomials and the dynamical equations are satisfied by penalty functions is described. The convergence and fidelity of the method are demonstrated with solutions to the following problems: brachistochrone, subsonic transport minimum-time climb, supersonic interceptor minimum-time climb, Goddard rocket problem with a singular arc, subsonic transport minimum-fuel for a fixed-range mission including climb, cruise and descent, and optimal evasive maneuvers for an airplane pursued by a missile with proportional guidance. The results demonstrate that the method provides an efficient and reliable procedure for solving a wide variety of realistic trajectory optimization problems.

Nomenclature

D	= diagonal matrix used to make Hessian positive definite or drag
e	= function in first-order differential equations
f	= function in second-order differential equations
g	= gravitational constant
G	= residual vector
h	= constraint function or altitude
L	= lift
m	= mass
M	= Mach number
P	= generalized payoff function
Q	= penalty function
S	= aerodynamic reference area
S_{FC}	= specific fuel consumption
t	= time
t_i, t_f	= initial and final time
T	= thrust or elapsed time
u	= control vector
W, W_f, etc.	= diagonal weighting matrices
x	= state vector satisfying second-order differential equations
y	= state vector satisfying first-order differential equations
z	= vector of independent parameters
α	= angle of attack
γ	= flight path angle
ϵ	= penalty function weighting factor
$\epsilon_1, \epsilon_2, \epsilon_3$	= value of ϵ corresponding to various phases of optimization
η	= throttle control parameter
λ	= weighting parameter on diagonal matrix
ν	= weighting factor for throttle augmentation
ρ	= atmospheric density
ω_i, ω_f	= vector of independent variables determining initial and final states

I. Introduction

THIS paper describes a method by which a trajectory can be optimized reliably within a realistic set of performance limitations. The method has been implemented in a computer program called the Chebychev Trajectory Optimization Program (CTOP). This program has been found to be useful in a wide variety of practical applications. Our method is characterized by the use of penalty functions to enforce the equations of motion and path constraints. The state and control time histories are represented by patched polynomials that are formed by interpolation at Chebychev points. Optimization is accomplished through the use of a full second-order scheme.

Penalty functions were proposed by Courant[1] and then later brought to prominence in optimal control by the work of Kelley,[2] who used them to satisfy path and terminal state constraints. Balakrishnan[3] extended the use of penalty functions to solve differential equations in a manner similar to that presented in this paper. A complete discussion of penalty functions and their development is given by Fiacco and McCormick.[4] The discretization of both the control and state variable time histories by Chebychev polynomials was originally applied to the determination of optimum finite thrust interplanetary orbit transfers.[5,6] The optimization is accomplished by a modification of Newton's method which is related to the procedures described by Levenberg[7] and Marquardt.[8]

II. Mathematical Method

A. Statement of the Problem

We consider a wide class of trajectory optimization problems involving one or more vehicles for which equations of motion are defined over a range of flight conditions, thrust settings, and control positions. Performance limitations such as maximum speeds, incidences, and maneuverability may be imposed. We assume a dynamical system described by the differential equations

$$x'' = f(x, x', y, u, t) \qquad y' = e(x, x', y, u, t) \tag{1}$$

where x is a vector of states (e.g., position) governed by Newton's second law, y a vector of states governed by a first-order differential equation (e.g., mass flow) and u a vector of controls. The prime denotes differentiation with respect to time. The solutions to the differential equations are defined on the interval $t \epsilon [t_i, t_f]$ with boundary conditions

$$\begin{aligned} x(t_i) &= a_i(\omega_i, t_i) \quad & x'(t_i) &= b_i(\omega_i, t_i) \quad & y(t_i) &= c_i(\omega_i, t_i) \\ x(t_f) &= a_f(\omega_f, t_f) \quad & x'(t_f) &= b_f(\omega_f, t_f) \quad & y(t_f) &= c_f(\omega_f, t_f) \end{aligned} \tag{2}$$

where ω_i and ω_f are vectors of independent variables determining the initial and final values of the states, respectively.

Presented as Paper 79-1624 at the AIAA Atmospheric Flight Mechanics Conference, Boulder, Colo., Aug. 6-8, 1979; submitted Oct. 30, 1979; revision received Oct. 9, 1980.

*Specialist Engineer, Boeing Aerospace Company. Member AIAA.
†Principal Engineer, Boeing Military Airplane Company (BMAC). Member AIAA.
‡Senior Engineer, Boeing Aerospace Company. Member AIAA.
§Manager Aerodynamic Technology, Boeing Commercial Airplane Company (BCAC). Associate Fellow AIAA.

Inequality constraints in the form

$$|h(x,x',y,u,t)| \leq 1 \tag{3}$$

may also be imposed on the system.

The optimal solution to Eqs. (1-3) is defined to be the set of vector time histories $x(t)$, $y(t)$, $u(t)$, vector parameters ω_i, ω_f and possibly times t_i and t_f which satisfy these constraints and minimize a payoff function $\phi(t_i,t_f,\omega_i,\omega_f)$, which without loss of generality, is assumed to depend only on the boundary values.

The preceding formulation may be converted to a more unconstrained optimization problem (e.g., Ref. 3) by defining an augmented payoff P as the sum $\phi + Q/\epsilon$, where

$$Q = \int_{t_i}^{t_f} [(x'' - f, W_f, x'' - f) + (y' - e, W_e, y' - e) + (v(h), W_h, v(h))] \mathrm{d}t \tag{4}$$

where $v(h)$ is a twice continuously differentiable ramp function. (a, W, b) denotes the inner product of the vectors a and b relative to the diagonal weighting matrix W. We use the specific ramp function

$$\begin{aligned} v(h) &= 0 && (|h| \leq 1) \\ &= (|h| - 1)^3/A^2 - \tfrac{1}{2}(|h| - 1)^4/A^3 && (1 \leq |h| \leq 1 + A) \\ &= |h| - 1 - \tfrac{1}{2}A && (|h| \geq 1 + A) \end{aligned} \tag{5}$$

The parameter $A > 0$ allows v to be twice continuously differentiable. $A = 0.05$ has been found to be reasonable for the inequality constraints in Sec. IV.

The diagonal weighting matrices W_f, W_e, and W_h are chosen to make Q dimensionally invariant. W_f is selected to be $(32.2)^2 I/g^2 T_0$ where I is the identity matrix and T_0 the initial guess for elaspsed time. W_h is chosen as I/T_0 since h is already assumed nondimensional [Eq. (3)]. W_e is scaled similarly depending on the dimensions of y. The weighting matrices W_h and W_e are ordinarily considered fixed during the course of solution but on occasion may be updated as explained in the next section.

We assume the true payoff ϕ has been defined in a nondimensional way. The parameter ϵ then defines a nondimensional relative weighting between the true payoff and the constraints. The payoff P is to be minimized with respect to suitably smooth functions x, y, and u as well as the parameters ω_i, ω_f (and, possibly, t_i and t_f) with the only constraint being that test functions are required to satisfy the boundary conditions [(Eq. (2)]. We feel that such an approach is inherently less nonlinear and considerably more stable than "shooting" procedures wherein the test functions satisfy the en route constraints exactly but not the terminal conditions.

The integrated residuals of the en route constraints comprising Q of Eq. (4) are panalty functions. These functions attach a performance cost (or penalty) to a violation of each of the constraints. The use of penalty functions greatly simplifies the computational process of determining a minimum of ϕ since all the variables can be considered to be truly independent without explicit reference to constraint relations. Moreover, constraints may be added with little increase in cost. The location of the minimum depends on the value of the relative cost coefficient ϵ. In principle it should be possible to choose a sufficiently small value of ϵ such that at a minimum of P the constraint violations are negligible. For this value of ϵ one would expect that the functions $x(t)$, $y(t)$, $u(t)$, and parameters ω_i, ω_f, t_i, t_f, minimizing P will be close to those minimizing ϕ subject to zero constraint residuals. The equivalence of the two problems is usually, but not always, the case. One outstanding exception occurs when the constraints cannot possibly be satisfied. In this case the function P usually has a well-defined minimum and the nature of the penalty function violation will provide useful clues as to why the desired trajectory is impossible.

B. Sequence of Subproblems

As a practical matter, numerical optimization becomes more difficult with smaller values of ϵ because of the increasingly narrow canyons created by the penalty functions. Usually a moderately small value of ϵ will suffice for an acceptable approximation to the solution. Since it is almost impossible to guess such a value in advance, an adaptive procedure for determining ϵ is required. Thus, optimization problems for a sequence of values of ϵ must be solved. Our adaptive technique involves three phases.

For our first (or initialization) phase we choose $\epsilon = \epsilon_1$ where ϵ_1 is small. The justification for this phase is that optimization of the payoff ϕ is not meaningful unless the constraint residuals are "felt."

Once the constraint residuals are felt, the second (or optimization) phase is initiated. This phase is characterized by successively multiplying ϵ by a factor of 10 until the payoff ϕ begins decreasing substantially as the constraint residuals increase. The value of ϵ is then fixed at, say, $\epsilon = \epsilon_2$ and P is minimized until complete convergence is achieved. In this phase the constraints are purposely violated in order to achieve rapid convergence. Occasional problems arise whenever the states assume values for which the vehicle dynamics [Eqs. (1)] become unrealistic or nearly singular. Examples of these phenomena are: the mass of a vehicle approaching zero, the Mach number exceeding tabular ranges, or the angle of attack going beyond 90 deg. Presumably these problems do not occur at the true optimum ($\epsilon \approx 0$). Hence, they could be solved by limiting the size of ϵ in some manner. However, our approach is to attack each such problem individually through the weights W_e and W_h rather than adjusting ϵ_2. As an example, we impose a limit of 60 deg on the angle of attack. If this limit is exceeded on any iteration, the iteration is terminated and the weight on the C_L constraint $(h = C_L/(C_L)_{\max})$ is automatically doubled. This continues until the constraint is not longer violated.

Once convergence is achieved for $\epsilon = \epsilon_2$ the third (or constraint) phase is entered. In this phase optimization is performed for a decreasing sequence of values of ϵ. Specifically, ϵ is divided by a factor of 2 beginning with $\epsilon = \epsilon_2$. Full convergence is required for each such subproblem (usually from two to seven iterations suffice). This phase is terminated when the payoff, trajectory, and controls cease to change, say, when $\epsilon = \epsilon_3$. The constraint residuals are then examined for acceptability. Since it is difficult to determine what is an acceptably small residual, Eqs. (1) are always integrated numerically using the initial conditions of Eqs. (2). The vehicle control history is represented by polynomials produced from the optimization procedure at $\epsilon = \epsilon_3$. The measure of acceptability then becomes how well the terminal conditions of Eqs. (2) and the constraints of Eq. (3) are satisfied. Invariably, unacceptable violations are due to inadequate discretization. This will be discussed in the next section.

The adaptive procedure for determining ϵ_2 in phases 1 and 2 enhances the overall reliability and "hands-off" nature of the method. However, it can consume from 20 to 50% of the CPU costs of CTOP. Clearly if a good value of ϵ_2 is known, it can be used directly without executing the adaptive procedure. This leads to substantial savings since once a value of ϵ_2 is established for one trajectory in a class of problems it can often be used for all trajectories in the class. Also we note that there appear to be attactive methods for obtaining an a priori estimate of ϵ_2 which the authors have yet to explore, e.g., Betts.[9]

C. Discretization

In our method the time history of each state and control variable is represented by patched polynomials that satisfy the initial and terminal conditions at all times. For most atmospheric trajectories the state variables are relatively smooth functions of time. Therefore, it is feasible and economic to approximate the state variables by polynomials in time. Such a representation is also usually advantageous for controls, with the exception of bang-bang controls. Here, polynomials can still do a reasonable job from the standpoint of finding the optimum value of the payoff although the control discontinuities are inevitably smeared over a finite time interval.

The more complicated a trajectory, the higher the order of the polynomial required to describe it. In general, increasing the order beyond a certain point gives diminishing returns compared to patching lower-order polynomials together. We have found that eighth order is a rough break-even point, hence we have confined ourselves to the use of eighth-order polynomials. The time interval covered by a single polynomial is called a "leg." Polynomials are patched together at "patch points." The patching requires continuity of value and first derivative for state variables described by second-order differential equations. Continuity of value only is required for those described by first-order differential equations. No continuity is required for control variables. In many cases the controls will be continuous at patch points but only because such a result is optimal.

Details concerning the representation of a function of time by patched polynomials are presented in Refs. 5 and 6. To summarize, the polynomials are formally functions of a variable $s\epsilon$ [0,1] with leg time becoming a parameter. (For multileg trajectories with variable elapsed time the leg time proportions are usually held fixed.) The free parameters describing a function are the *values* of the function at "Chebychev" points of [0,1] with a slight modification for second-order states. The polynomials then interpolate these values. The polynomials representing x, y, and u may be differentiated and substituted into Eq. (4). The functions f, e, and h become functions of time that may in turn be approximated by polynomials. The indicated integration may then be performed analytically. The payoff then becomes a function of discrete parameters, i.e.,

$$P=P(z) \tag{6}$$

where z is a vector of independent parameters containing ω_i, ω_f (possibly, t_i and t_f) and the free parameters describing x, y, and u.

P may then be minimized with respect to z. Once convergence is achieved and a final value of ϵ is found, the adequacy of the discretization must be established. The numerically integrated trajectory furnishes an indication of the need for more legs or a better distribution of the patch points. It does not furnish any information about the adequacy of the representation of the controls. A good indication of the inadequacy of the polynomial representation of the controls is the magnitude of the discontinuity in the controls at patch points in cases where it is known that the controls should be continuous. To be on the safe side an extra leg should be added and the new value of the payoff compared with the old whenever adeuacy has not been established by experience. This procedure can be automated (e.g., Ref. 8), but currently the number of legs and distribution of patch points is chosen by the user. Generally patch points are equally distributed. However, considerable benefit can sometimes be gained by packing polynomials in regions where "action" is taking place, e.g., at the end of a missile-airplane encounter. When an extra leg is added the new trajectory should also be compared with the old in the case of a very shallow optimum, where every trajectory is near optimal. Here the method can take advantage of the discretization to produce minute improvements in the payoff. Often a large number of legs are required to pinpoint the precise optimum trajectory although the payoff ceases to change. Finally we note that in the early stages of determining ϵ_2 the representation of the states and controls may be rather crude, i.e., one or two polynomials.

D. Minimization Technique

In this section we describe the iterative technique to minimize P with respect to z [(Eq. (6)]. We use a second-order technique for several reasons. First, second-order iterations in our case are only moderately more expensive than first-order iterations, yet make possible considerably larger step sizes. Second, a precise optimum may be achieved, which is important when trade studies are being performed and valid trends must be established unhampered by convergence noise. Finally, and most importantly, the positive definiteness of the second derivative of the payoff provides a sufficient condition for a local optimum. Such a condition is not available with first-order methods. Optimization methods are notorious for producing suboptimal trajectories leading to inconsistencies and invalid conclusions. From considerable experience we feel that confidence in convergence is well worth the added expense of a true second-order technique and the extra iterations at the end required to prove convergence.

We rewrite Eq. (6) as

$$P(z)=\phi(z)+(1/2\epsilon)\left(G(z),W,G(z)\right) \tag{7}$$

The second term on the right represents the quadratic penalty functions with W as a positive definite integration matrix and $G(z)$ a residual vector. Assuming z_0 to be the starting value of z at the beginning of an iteration we desire to find a new value of z at the end of the iteration such that

$$P(z_0+\Delta z)<P(z_0) \qquad \Delta z=z-z_0 \tag{8}$$

To accomplish this P is expanded in a Taylor's series about z_0 with only terms up to and including second-order retained, i.e.,

$$P(z_0+\Delta z)=P(z_0)+P_{z_i}(z_0)\Delta z_i+\tfrac{1}{2}P_{z_iz_j}(z_0)\Delta z_i\Delta z_j \tag{9}$$

Here the subscript z implies differentiation with respect to z, z_i is the ith component of z, and repeated subscripts are summed. We have

$$P_{z_i}(z_0)=\phi_{z_i}(z_0)+(1/\epsilon)\left(G(z_0),W,G_{z_i}(z_0)\right) \tag{10}$$

and

$$P_{z_iz_j}(z_0)=\phi_{z_iz_j}(z_0)+(1/\epsilon)\left(G_{z_i}(z_0),W,G_{z_j}(z_0)\right)$$
$$+(1/\epsilon)\left(G(z_0),W,G_{z_iz_j}(z_0)\right) \tag{11}$$

Assuming $P_{z_iz_j}(z_0)$ is positive definite, the quadratic form on the right of Eq. (9) has a minimum at

$$\Delta z_i=-\left[P_{z_iz_j}(z_0)\right]^{-1}P_{z_j}(z_0) \tag{12}$$

An iteration of the form Eq. (12) is, of course, Newton's method. Two problems can arise if z_0 is not near a minimum. First, $P_{z_iz_j}(z_0)$ may not be positive definite, and second, the step implied in Eq. (12) may be so large that third- and higher-order terms of Eq. (9) become important with the result $P(z_0+\Delta z)>P(z_0)$. In the latter case Δz may be multiplied by a factor less than 1 so that the approximation of Eq. (9) is valid. However, it is better to include such scaling a priori via procedures of the type which have been described in the literature by Levenberg,[7] Marquardt,[8] and Fletcher.[10] In fact these procedures eliminate the first problem as well. In our case we add a term $D(1-\lambda)/\lambda$, $\lambda\epsilon[0,1]$ to the right side of Eq. (9). $D(z_0)$ is a diagonal matrix with positive elements.

Δz[and thus $P(z_0+\Delta z)$] become a function of λ, i.e.,

$$\Delta z_i(\lambda) = -[H_{ij}(\lambda,z_0)]^{-1}P_{z_j}(z_0) \tag{13}$$

where,

$$H_{ij}(\lambda,z_0) = P_{z_i z_j}(z_0) + [(1-\lambda)/\lambda]D_{ij}(z_0)$$

and a one-dimensional search may be employed to minimize P as a function of λ. If λ is near one, the matrix H may not be positive definite. In this case such a value of λ becomes an upper bound for our one-dimensional search. A minimum must exist between this value and $\lambda=0$ since

$$\frac{\mathrm{d}P}{\mathrm{d}\lambda}(z_0+\Delta z(\lambda))\,|_{\lambda=0} = -P_{z_i}D_{ii}P_{z_i} < 0 \tag{14}$$

An excellent choice for the diagonal elements of D corresponding to control angles has proven to be the absolute value of the corresponding diagonal element of the third term on the right of Eq. (11). This choice has the property that such an element automatically decreases in size as the constraints are better satisfied so that the optimal value of λ remains a reasonable number. For the states and remaining controls the same choice is made, although a contribution from the second term is included when the diagonal element of the third term on the right is near zero.

Near an optimum, the value of λ minimizing $P(\lambda)$ should approach 1. In fact such an occurrence is part of our convergence criterion. As is usually the case for Newton's method when $\lambda=1$ yields a decreasing payoff, the gradient P_z suddenly goes to zero (to within machine accuracy) along with Δz of Eq. (13), and convergence is established.

III. Programming Considerations

CTOP is by no means an efficient production code. Nevertheless, the program embodies several features that strongly enhance the overall feasibility of the method. These features are described in the following.

The calculation of $\Delta z(\lambda)$ from Eq. (13) is actually accomplished by solving the system of equations directly rather than inverting the matrix H. We note that H is symmetric and positive definite so that the highly efficient and stable square root method[11] may be employed. On a 14 digit computer this method has always been sufficiently accurate to allow for the value of ϵ_3 required to ensure adequate satisfaction of the constraints. The square root method destroys the elements below the diagonal in the process of decomposition. However the (symmetric) elements above the diagonal are untouched. Thus the matrix H can, for different values of λ, be reconstructed by reflecting the elements above the diagonal across the diagonal to the lower side after each solution.

A full one-dimensional search is rarely required. On any given iteration the value $\lambda=1$ is always attempted. If the square root of a negative number is encountered or P is not decreased, a value of λ somewhat greater than that found in the last iteration where $\lambda \neq 1$ is employed. If this value fails as well, the full one-dimensional search is then initiated. This procedure does employ derivative information, which is relatively cheap to calculate since $\mathrm{d}\Delta z(\lambda)/\mathrm{d}\lambda$ may be computed utilizing the same decomposition of H as that used for computing $\Delta z(\lambda)$ itself.

The cost of decomposing H grows as the square of the number of states and controls, but only grows linearly with the number of trajectory legs. This is because the state and control parameters are coupled only within each leg except at patch points, and advantage is taken of the resultant block diagonal structure of H. As a result of the efficiencies just described the actual cost of decomposing H averages only 7-10% of the cost per iteration.

The functions f, e, and h in Eqs. (1) and (3), respectively, may be in part defined by tables. Since second-order derivatives of these functions are required, the tables are all interpolated using a multidimensional quintic spline package. Quintic splines are preferred because of the local character of the interpolation as well as the resultant reduced storage requirements. Besides the data itself, only a few coefficients for each ordinate interval need be stored.

Chebychev polynomials have well-known symmetry properties which are exploited to reduce computation time (e.g., Ref. 5).

In order to provide accurate second partial derivatives and low computation cost, all derivatives are computed analytically. Ordinarily, the derivation, coding, and checkout of first and second partials for the complex equations describing realistic vehicle dynamics would be a formidable task, greatly inhibiting application of the code to problems involving new vehicle models. To alleviate this problem a library of analytic derivative subroutines has been developed. (These routines are execution time routines, not preprocessors.) A subroutine is available for each mathematical operation required. For example, the cross-product routine computes the cross product of two vectors along with the first and second derivatives of the product with respect to all variables by which the input vectors have been differentiated. These routines make it possible to code and check out the derivatives of a complicated set of differential equations in almost the same time as the equations themselves.

IV. Numerical Verification

This section describes six of the widely varying problems that have been solved using CTOP. Each was selected to illustrate one or more characteristics of the method, and all solutions shown have been iterated to convergence as described in Sec. II.

A. Brachistochrone

This classical problem of the calculus of variations shows that with one or two legs CTOP will give a very accurate solution to a simple problem. A brachistochrone problem can be described by the equations: $v'=\sin\theta$, $x'=v\cos\theta$, $y'=v\sin\theta$; with boundary conditions $v(0)=0$, $x(0)=0$, $y(0)=0$, $x(t_f)=1$, $y(t_f)=1$.

We seek the optimal angle $\theta(t)$ that minimizes the final time t_f. The solution is known to be a cycloid (e.g., see Ref. 12). The maximum root sum square deviation of the CTOP solution from the exact solution was $1.9\ 10^{-4}$ with one leg and $8.4\ 10^{-6}$ with two equally distributed legs. The difference between the CTOP solution and a numerically integrated solution with the CTOP controls was less than 10^{-5}. To test the CTOP method of treating inequality constraints [Eq. (5)], a constrained Brachistochrone problem (Ref. 13, pp. 119 and 120) was also run. The results are shown in Fig. 1. The

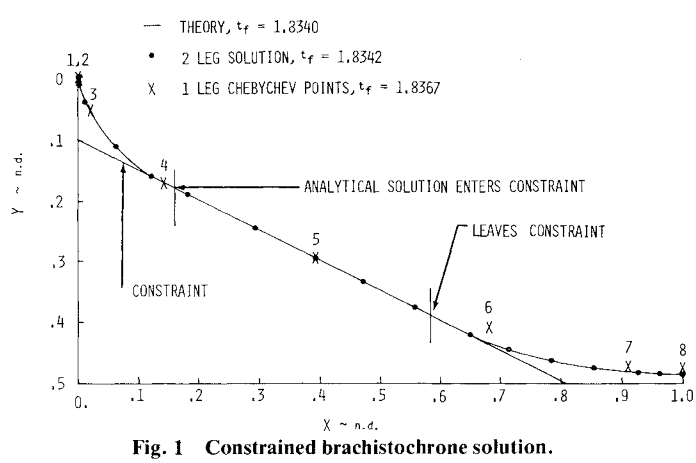

Fig. 1 Constrained brachistochrone solution.

two leg (equally distributed) solution is quite accurate and the one leg solution is quite good considering that only one Chebychev point lies within the constraint time interval.

B. Subsonic Transport Minimum-Time Climb

The objective of this problem is to fly a subsonic transport from 10,000 to 21,000 ft altitude in minimum time.

For this problem and all other airplane problems, the equations of motion used are for a point mass airplane in three-dimensional flight in still air above a flat Earth using a fixed nonrotating cartesian coordinate system.

$$x'' = (T+L+D)/m+g \qquad m' = \eta S_{FC} T_m \tag{15}$$

The vehicle is constrained to coordinated maneuvers (zero sideslip angle). Control is maintained by modulating the power setting η and the attitude of the vehicle's longitudinal axis b. b is defined by an elevation ϕ, and heading angle θ, such that

$$b^T = (\cos\phi\cos\theta, \cos\phi\sin\theta, \sin\phi)$$

The following definitions are also used:

$$T = \eta T_m b \qquad L = \tfrac{1}{2}\rho S C_{L_\alpha}(b \otimes V) \otimes V$$

$$D = \tfrac{1}{2}\rho V|V| S (C_{D0} + K C_L^2)$$

where C_{L_α}, C_{D0}, and K are given as tabular functions of Mach number. T_m and S_{FC} are tabular functions of Mach number and altitude. All atmospheric data are obtained from quintic

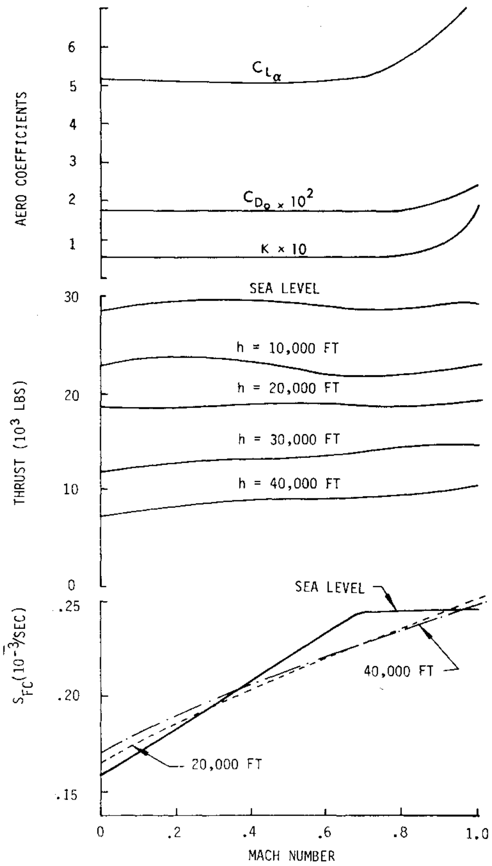

Fig. 2 Subsonic transport aerodynamic thrust and S_{FC} curves.

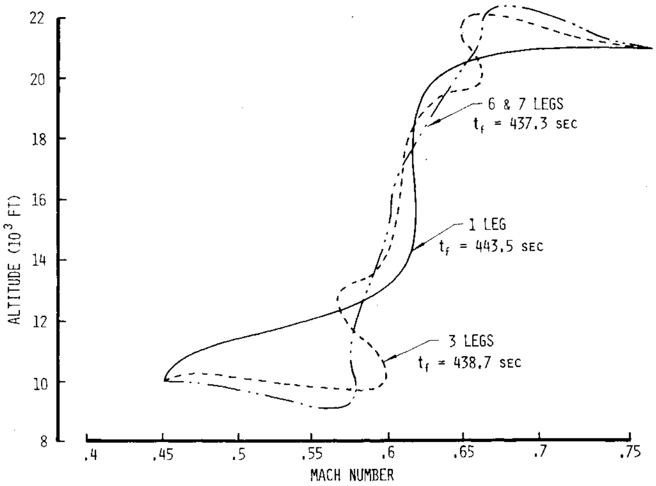

Fig. 3 Subsonic transport altitude vs Mach number.

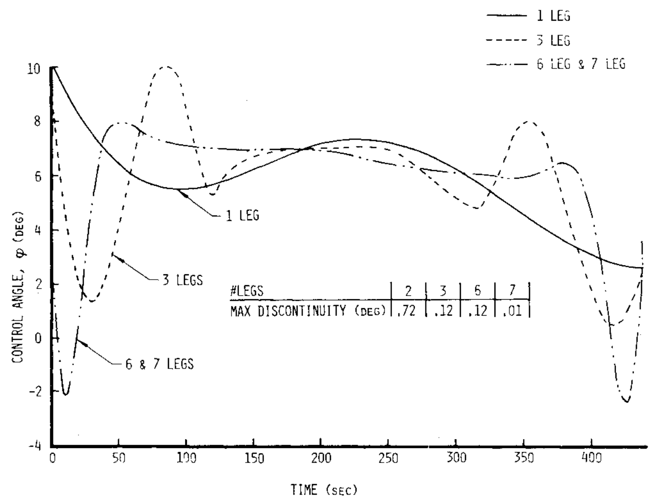

Fig. 4 Subsonic transport control angle vs time.

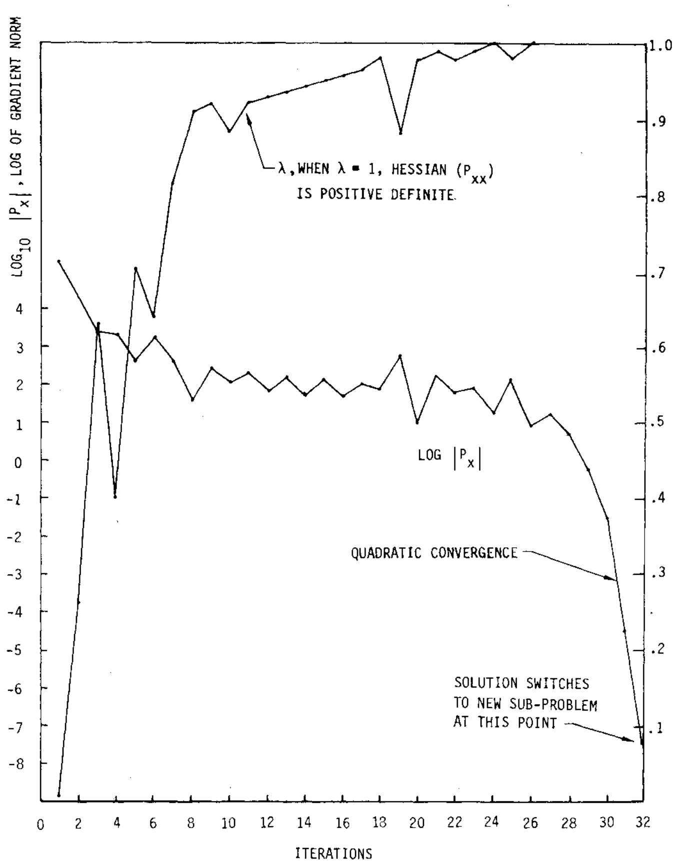

Fig. 5 Subsonic transport convergence for initial subproblem.

interpolation of values from the NACA 1962 Standard Atmosphere. Inequality constraints can be imposed on normal load factor, lift coefficient, dynamic pressure, altitude, and power setting.

The aerodynamic, thrust, and mass flow data used to describe the airplane are shown in Fig. 2. This problem is characterized by a very shallow optimum so that considerable variation in the trajectory (and controls) will give nearly the same value for the time to climb. With problems of this nature CTOP will give an accurate value for the time with one or two legs. In order to produce an accurate control and trajectory many legs are required. This is illustrated by the Mach number altitude plot shown in Fig. 3. As mentioned earlier, a good indication of the adequacy of the control variable description is the absence of discontinuities in the controls. Figure 4 shows that the control discontinuities get smaller with increasing legs.

Figure 5 shows the convergence history for a single subproblem. Note that once the Hessian becomes positive definite ($\lambda=1$) the gradient decreases rapidly to zero (limited by machine round-off). Figure 6 illustrates the convergence history at the end of each subproblem. Note that the residual Q decreases as the squre of ϵ. Also note that each subproblem requires about the same number of iterations after the first few subproblems.

C. Supersonic Interceptor Minimum-Time Climb

The third verification problem is a minimum-time problem proposed and solved by Bryson.[14] This problem consists of finding the angle-of-attack time history such that time is minimized for a supersonic interceptor to travel from a set of fixed initial conditions, $h(0)=0$, $M(0)=0.38$, $\gamma(0)=0$, to the fixed final conditions $h(t_f)=20$ km, $M(t_f)=1$, $\gamma(t_f)=0$.

For this particular problem the altitude constraint ($h>0$) was the only active inequality constraint. The power setting was fixed at full power. The defining aerodynamic and engine data were taken from Ref. 15.

The solution obtained with the boundary conditions just noted with an initial weight of 42,000 lbs is shown in Fig. 7. The initial iterate was obtained by matching a cubic polynomial to the boundary conditions. Perhaps the most interesting feature of the CTOP result is its greater similarity to the energy state solution than that of the reference method. This feature was also found in Ref. 16, which described a solution very similar to the CTOP solution. In contrast to the energy state solution the CTOP solution is a physically realizable solution. Integration of the equations of motion using the CTOP controls produced a trajectory that satisfied the terminal constraints and, moreover, closely matched the polynomial representation of the trajectory. An indication of the convergence for this problem with respect to discretization can be seen from examining Fig. 8. By comparing the trajectories, one can observe that for more than four legs the solution has stabilized, indicating that the discretization using six legs is more than adequate. The six-leg trajectory requires about 250 s of CPU time on a CDC CYBER 175 for a complete solution assuming full proof of convergence as well as no a priori knowledge of the appropriate value of ϵ_2.

D. Goddard Rocket Problem

This is a classical problem that often involves a singular arc (i.e., an arc over which the switching function is identically zero yielding no information about the optimal control). The problem is described in Ref. 13 and in its simplest form consists of programming the propellant burn of a single stage sounding rocket to achieve maximum altitude. The equations of motion are

$$h''=(\eta T_m - D)/m - g \qquad m'=-\eta T_m/c$$

where h is altitude, m mass, T_m maximum available thrust, D drag and $\eta\epsilon$ [0,1] a throttle factor. We use the drag formula considered by Tsien and Evans,

$$D=0.5\rho_0(h')^2C_DSe^{-\beta h}$$

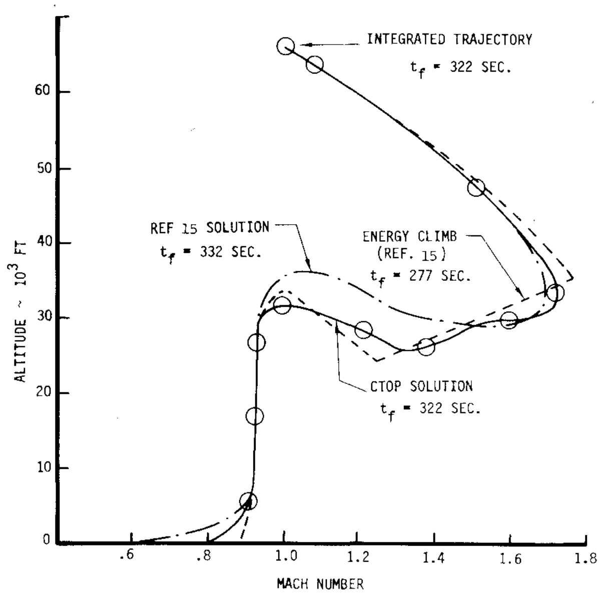

Fig. 7 Comparisons for supersonic interceptor minimum-time climbs.

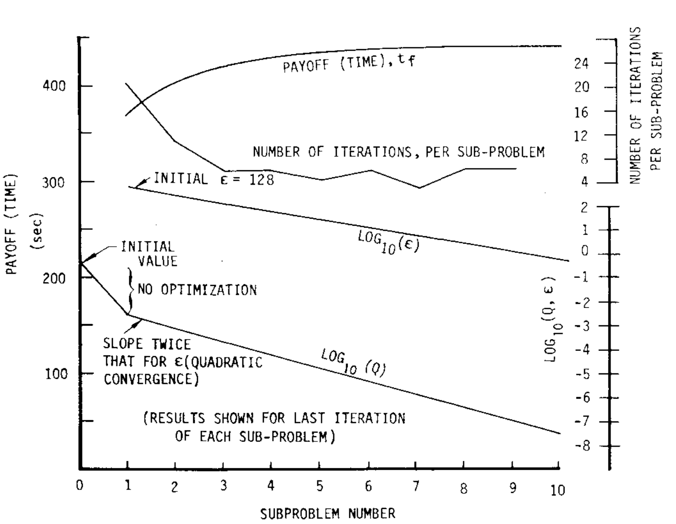

Fig. 6 Subsonic transport convergence at end of subproblems.

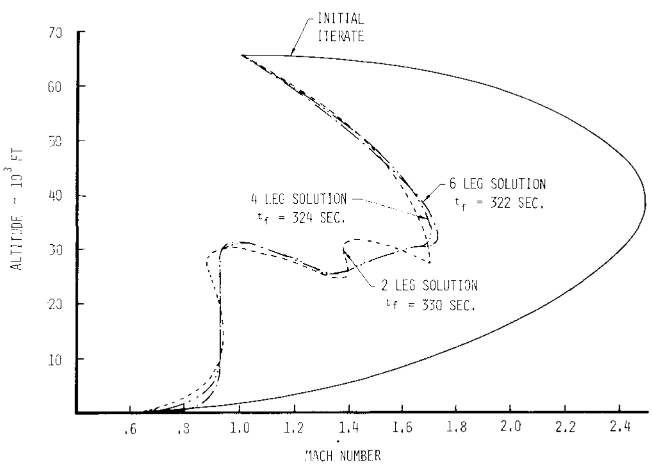

Fig. 8 Discretization study, supersonic interceptor minimum-time climb.

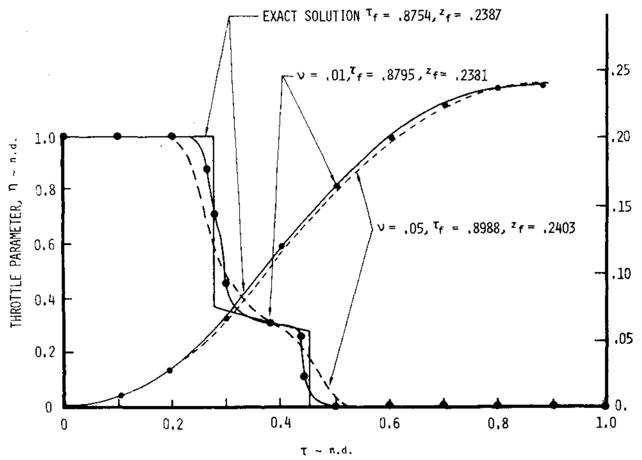

Fig. 9 Throttle parameter and trajectory for Goddard's problem.

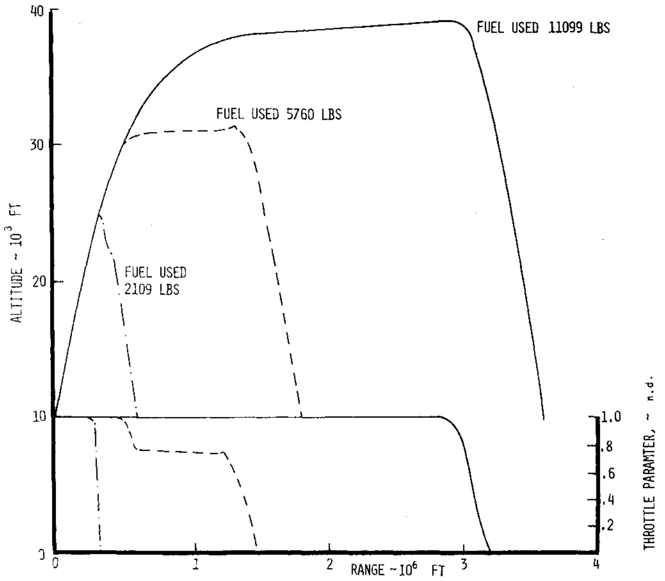

Fig. 10 Optimal fixed range trajectories with altitude free.

The boundary conditions are $h(0)=h'(0)=0$, $m(0)=m_0$, $h'(t_f)=0$, $m(t_f)=m_f$. Values which lead to a singular arc are $0.5\ \rho_0 C_D S c^2/T_M = 0.7110$, $\beta c^2/g = 3.264$, $m_0/m_f = 3$, $T_m/(m_0 g) = 2$. The (exact) integrated solution using the feedback law described in Ref. 13 is shown by solid lines in Fig. 9. The singular arc occurs between $\sigma = 0.279$ and $\sigma = 0.455$ where $\sigma = gt/c$. In Fig. 9, $z = gh\ c^2$.

The solution can be approximated numerically by the solution to a neighboring nonsingular problem via a device similar to that of Ref. 17. Here we replace η (in the h'' equation only) by $(\eta - \nu\eta^2)/(1-\nu)$ where ν is sufficiently small. The equidistantly distributed 6 leg solutions with $\nu = 0.05$ and 0.01 are shown by the dashed lines and lines with dots. The solution with $\nu = 0.01$ is off by less than 0.3% in altitude and 0.4% in time. Note that convergence with respect to ν is not necessarily monotonic due to the divisor $(1-\nu)$.

E. Subsonic Transport Fixed-Range/Minimum-Fuel Flight

The objective of this problem is to fly from a given position and velocity to the same position and velocity at varying fixed ranges away with minimum fuel use. This type of problem was solved in Ref. 18 where distinct climb, cruise, and descent phases were assumed. We make no such restriction and assume a single flight path between the initial and final conditions.

The airplane model used is described in Sec. IVB except that here the throttle control η is allowed to vary between 0 and 1. The initial and final conditions were $h = 10{,}000$ ft, $M = 0.45$, $\gamma = 0$ deg. Ranges of 0.6, 1.8 and 3.6×10^6 ft were selected. For each case the starting trajectory was level flight at 10,000 ft and Mach 0.45. The resulting optimal trajectories and power settings are shown in Fig. 10.

The general characteristics of these trajectories are in agreement with classical steady-state performance analysis. As expected, the cruise section of the longest flight has an angle of attack and Mach number such that the induced drag is equal to the zero lift drag. The cruise is at maximum power at the highest altitude possible with the drag as just described. The cruise altitude increases slightly as the plane loses weight. As also expected, the descent is essentially a maximum L/D glide with minimum power.

The optimal power setting is seen to consist of two arcs, one with full power and one with zero power. The minor deviations from this, seen in the figure, are the result of the polynomial representation and could be made as small as desired by adding more legs. We note here that the integrated trajectory using the CTOP control met the terminal conditions to within 17 ft in range, 20 ft in altitude, 0.06 deg in flight-path angle and 0.0004 in Mach number. The shortest trajectory contains no cruise section. The intermediate range trajectory contains three arcs. The center or cruise arc is a singular arc (the second derivative of the Hamiltonian is only semidefinite[16]). This situation arises because the throttle parameter appears linearly in the differential equations. This problem was treated in CTOP by adding a small quadratic term to the mass flow equation, i.e.,

$$m' = m'_m(h,m)\,(n + \nu\eta^2)/(1+\nu)$$

Successive solutions were run with ν decreasing until its effect on the payoff was small. In Ref. 19 it is noted that under certain circumstances a steady-state cruise is nonoptimal. The absence of an oscillatory cruise in our case may be due to the quadratic term in the mass flow equation. For some trajectories, in particular the intermediate range trajectory of Fig. 10, we do lose the steady-state cruise as ν becomes excessively small ($\nu \approx 0.01$). We have not yet performed convergence studies to determine whether the resultant oscillations are real or numerically induced. We also note that for cases with more exact mass flow models, i.e., $m' = f(T,h,M)$, we seem to obtain steady-state cruises at intermediate throttle settings without adding any quadratic term.

F. Missile Evasion

The basic missile evasion problem solved can be stated as follows. Obtain the control time history for an airplane being pursued by a guided missile with a proportional navigation control law, such that the separation distance at the point of closest approach is maximized. The point of closest approach is assumed to occur at the final time and is defined by requiring that the final relative position and relative velocity vectors be orthogonal. We then maximize the final separation distance. Because of the nature of the orthogonality terminal condition a more realistic starting solution is required for this problem. Such a solution is obtained as follows. For a given set of initial conditions, some simple evasive maneuver for the airplane is assumed and the trajectories are numerically integrated until the relative position and velocity are orthogonal. If the miss distance at this time is greater than the miss distance required for a "hit" (we assume 100 ft in the following results), this is accepted as the starting solution. If it is less than 100 ft, then an artificial lag is added to the missile control system and this is adjusted iteratively until the miss distance is 100 ft. These trajectories are then used for the starting solution (with the lag removed). For this case, the initial iterations are carried out with the miss distance held fixed (in this mode the optimization program becomes a two-point boundary-value differential-equation solver). If the program converges to a nonzero value for the penalty func-

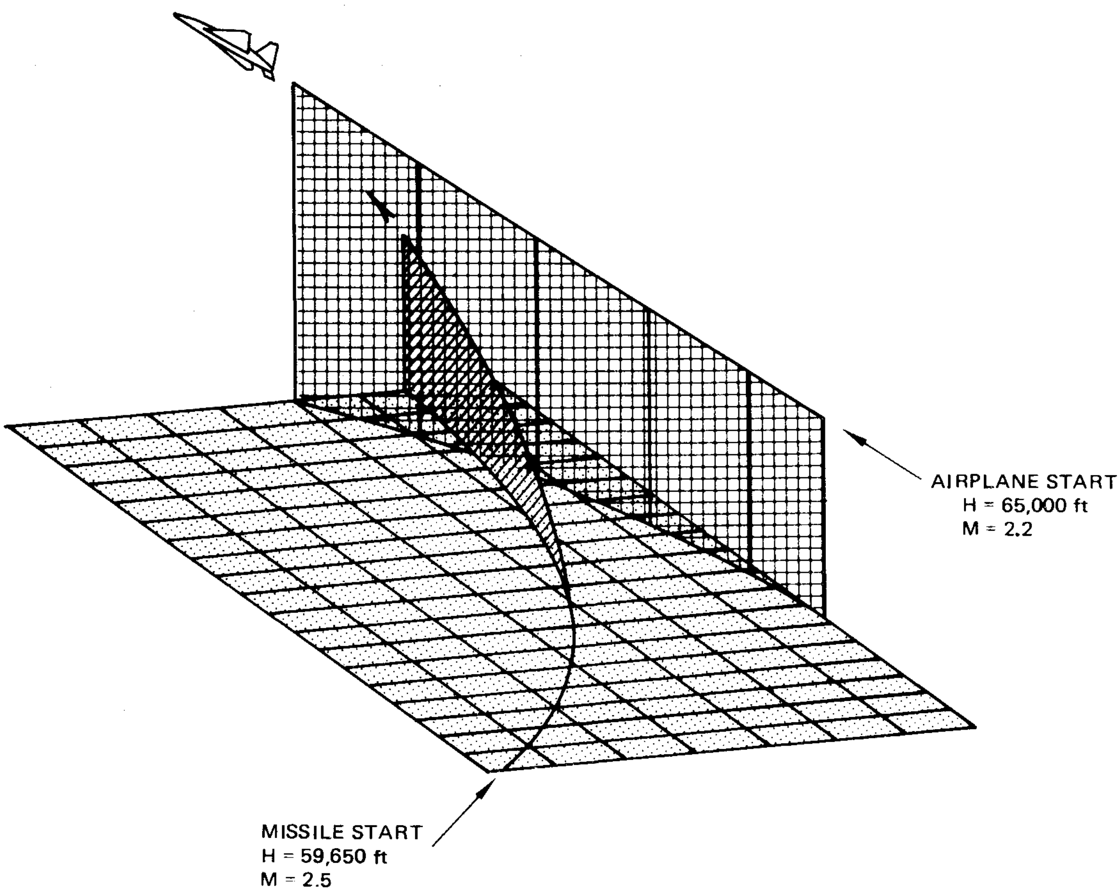

Fig. 11 Optimal missile evasion maneuver, example 1.

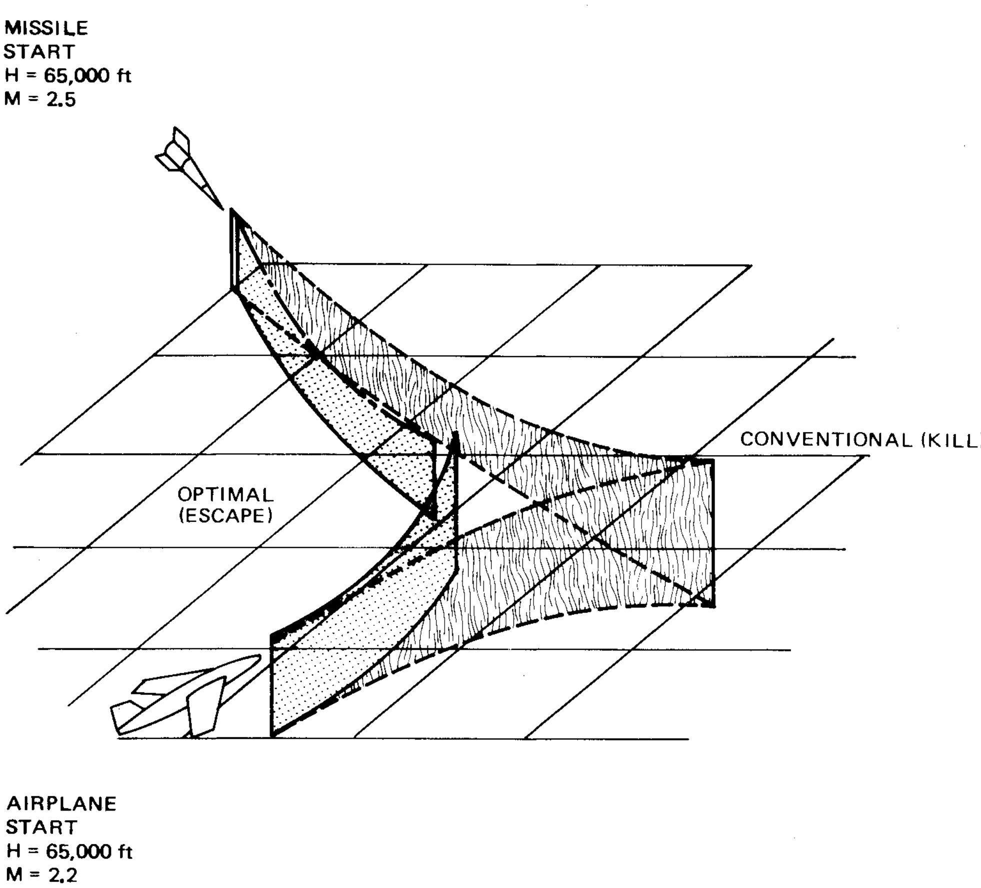

Fig. 12 Optimal missile evasion maneuver, example 2.

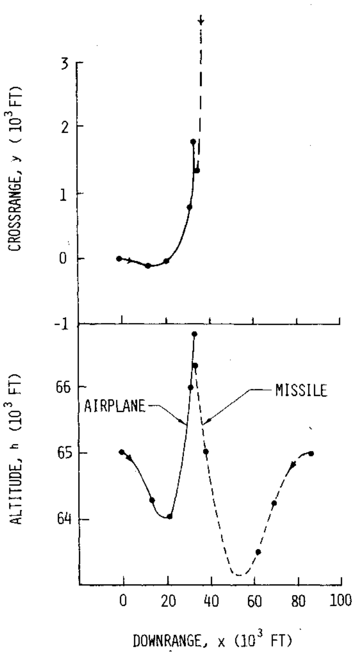

Fig. 13 Cross range and altitude vs downrange, example 2.

tion then all feasible trajectories (within some vicinity of our starting solution) have miss distance less than 100 ft. Thus, we conclude that the airplane is hit for any optimal maneuver and do not consider the case further. If the program converges to a sufficiently small penalty function, we then free the miss distance and run the optimization program in the usual manner. The procedure just described is completely automated and a converged optimal trajectory is obtained with a single computer run.

The airplane considered is a Mach 2.2 high level penetrator which is constrained to $(C_L)_{\max}$ of 1.0 and a load factor of 6 g's. The $(C_L)_{\max}$ constraint effectively keeps the load factor below about 4 g's so that the limit is never reached for this case. The missile for this case is a contemporary air-to-air missile that is launched from a supersonic interceptor at Mach 2.5. The angle of attack for the missile is limited to 6 deg. A rear quartering encounter is illustrated by a three-dimensional plot in Fig. 11. The final miss distance was 5583 ft.

From Fig. 11 one can see that the required optimal evasion is quite simple. It is basically a max-g turn. A more interesting maneuver results when the same airplane-missile combination is started in a near head-on encounter, Figs. 12 and 13.The optimal maneuver in this case is rather sophisticated. The airplane initially dives, then turns into the missile to reduce the missile's turn rate, then a few seconds before encounter makes a maximum-g pullup. The missile then undershoots and misses by a large distance. A number of tries were made to escape with simple maneuvers (diving or climbing turns at max-g). The miss distance was always less than 100 ft. It appears that in order to escape, in this case the airplane must truly take advantage of a weakness in the control law employed by the missile. This case illustrates the real value of optimal maneuver computation. Although in some cases (such as the previous one) the optimum turns out to be a simple maneuver that probably could have been discovered by trial and error, this would be unlikely in the present case.

V. Conclusions

The examples described in this paper indicate that the Chebychev Trajectory Optimization Program is capable of providing solutions to a wide variety of trajectory optimization problems. The nature of the method allows for realistic representation of vehicle modes and relatively little effort is required to change model characteristics. The second-order procedure employed provides a sufficient condition for the existence of a local optimum.

Acknowledgments

The authors wish to acknowledge the assistance of the late B. F. Itzen of The Boeing Aerospace Company and M. A. Epton of Boeing Computer Services in obtaining the results described in this paper.

References

[1] Courant, R., "Variational Methods for the Solution of Problems of Equilibrium and Vibrations," *Bulletin of the American Mathematical Society,* Vol. 49, 1943, pp. 1-23.

[2] Kelley, H.J., "Method of Gradients," *Optimization Techniques,* edited by G. Leitmann, Academic Press, New York, 1962, pp. 205-254.

[3] Balakrishnan, A.V., "On a New Computing Method In Optimal Control," *SIAM Journal on Control,* Vol. 6, May 1968, pp. 149-173.

[4] Fiacco, A.V. and McCormick, G.P., *Nonlinear Programming: Sequential Unconstrained Minimization Techniques,* John Wiley, New York, 1968.

[5] Johnson, F.T., "Approximate Finite-Thrust Trajectory Optimization," *AIAA Journal,* Vol. 7, June 1969, pp. 993-997.

[6] Hahn, D.W. and Johnson, F.T., "Final Report for Chebychev Trajectory Optimization Program (CHEBYTOP II)," The Boeing Company, Seattle, Wash., D180-12916-1, prepared for NASA under Contract NAS2-5994, June 1971.

[7] Levenberg, K., "A Method for the Solution of Certain Non-Linear Problems in Least-Squares," *Quarterly of Applied Mathematics,* Vol. 2, 1944, pp. 164-168.

[8] Marquardt, D.W., "An Algorithm for Least-Squares Estimation of Non-Linear Parameters," *SIAM Journal of Numerical Analysis,* Vol. 11, 1963, pp. 431-441.

[9] Betts, J.T., "An Improved Penalty Function Method for Solving Constrained Parameter Optimization Problems," *Journal of Optimization Theory and Applications,* Vol. 16, Nos. 1 and 2, 1975, pp. 1-24.

[10] Fletcher, R., "A Modified Marquardt Subroutine for Non-linear Least Squares," Theoretical Physics Div., Atomic Energy Research Establishment, Harwell, Rept. R. 6799 of Mathematics Branch, May 1971.

[11] Faddeeva, V.N., *Computational Methods of Linear Algebra,* Dover, New York, 1959, pp. 81-85.

[12] Bliss, G.A., *Calculus of Variations,* Open Court Publishing Company, La Salle, Ill., 1925, pp. 41-59.

[13] Bryson, A.E. and Ho, Y., *Applied Optimal Control,* Blaisdell Publishing Company, Waltham, Mass., 1969.

[14] Bryson, A.E. and Denham, W.F., "A Steepest-Ascent Method for Solving Optimum Programming Problems," *Journal of Applied Mechanics,* Vol. 29, June 1962, pp. 247-257.

[15] Bryson, A.E. and Desai, M.N., "Energy-State Approximation in Performance Optimization of Supersonic Aircraft," *Journal of Aircraft,* Vol. 6, Nov.-Dec. 1969, pp. 481-488.

[16] Rader, J.E. and Hull, D.G., "Computation of Optimal Aircraft Trajectories using Parameter Optimization Methods," *Journal of Aircraft,* Vol. 12, Nov. 1975, pp. 864-866.

[17] Jacobson, D.H., Gershwin, S.B., and Lele, M.M., "Computation of Optimal Singular Controls," *IEEE Transactions on Automatic Control,* Vol. AC-15, No. 1, 1970, pp. 67-73.

[18] Erzberger, H. and Lee, H., "Constrained Optimum Trajectories with Specified Range," *Journal of Guidance and Control,* Vol. 3, Jan.-Feb. 1980, pp. 78-85.

[19] Speyer, J.L., "Nonoptimality of the Steady-State Cruise for Aircraft," *AIAA Journal,* Vol. 14, Nov. 1976, pp. 1604 and 1610.