Experience Using Balakrishnan's Epsilon Technique to Compute Optimum Flight Profiles

LAWRENCE W. TAYLOR JR.,* HARRIET J. SMITH,† AND KENNETH W. ILIFF†
NASA Flight Research Center, Edwards, Calif.

A technique for computing optimum profiles is developed which differs from the classical gradient method in that a term representing the constraint of satisfying the equations of motion is included in the cost function to be minimized. Although the number of unknown independent functions is increased to include the state variables, the dimensionality of the gradient of the modified cost is greatly reduced, resulting in considerable savings in complexity and time. The unknown control and state variables are expressed in a functional expansion to facilitate solution by means of Newton's method. The effects of weighting terms and the number of functions on the convergence properties are discussed. Comparisons are made of solutions using the classical gradient method, dynamic programing, and Balakrishnan's epsilon technique.

Nomenclature‡

a_m = coefficient of functional expansion for velocity
b_m = coefficient of functional expansion for altitude
c_m = coefficient of functional expansion for flight-path angle
$\bar{c}$ = vector of unknown variables
D = drag, lb
e = gain used in the gradient technique
F = thrust, lb
f_m = sine functions used in functional expansion
g = gravitational constant, ft/sec^2
h = altitude, ft
J = cost
L = lift, lb
M = number of functions
m = mass, slugs
m = function, index
N = number of time points
n = time index
P = penalty function for missing end conditions in gradient method
T = total time, sec
t = time, sec
V = velocity, fps
γ = flight-path angle, deg (rad)
$\Delta\bar{c}$ = increment change in the vector of unknown variables
Δt = time increment
$\bar{\delta}$ = vector of $(\nabla t/\epsilon)^{1/2}\delta\dot{V},(\Delta t/\epsilon)^{1/2}\delta\dot{h},(N\Delta t)^{1/2}$
$\delta\dot{h}$ = error in $\dot{h}$ equation of motion
$\delta\dot{V}$ = error of $\dot{V}$ equation of motion
ϵ = weighting term in modified cost function
λ = percentage increase of diagonal elements of the $[\nabla\bar{c}\bar{\delta}^T\nabla\bar{c}\bar{\delta}]$ matrix
$\nabla(\cdot)$ = gradient of $(\cdot)$

Subscripts

k = iterative index
m = function index
N = final condition
n = time index
o = nominal

Superscript

T = transpose

Presented as Paper 69-75 at the AIAA 7th Aerospace Sciences Meeting, New York, January 20–22, 1969; submitted February 17, 1969; revision received August 4, 1969. The authors are indebted to A. V. Balakrishnan of the University of California at Los Angeles for the assistance he provided.

* Research Scientist. Member AIAA.
† Research Scientist.

‡ A dot over a symbol indicates differentiation with respect to time.

Introduction

THE problem of determining the control function and corresponding airplane flight profile which minimizes some cost such as time in flying between specified end conditions has received and continues to receive considerable attention. A variety of techniques[1–10] has been incorporated in digital computer programs developed to obtain these solutions.

One method of solving this problem is the energy approach.[1,2] This method consists of a local or instantaneous minimization of the cost, which is possible if only one state variable (energy) is used. The solutions are only approximate, however, with jumps in the control (altitude, for example) at the end conditions and typically in the transonic range between conditions of equivalent total energy. Although the resulting solutions are useful guides, they cannot be followed exactly by physical airplanes.

Another method that is used is dynamic programing,[3] in which the state space is discretized and a path through the network of nodes is generated which minimizes the cost. Although this process is very fast for small problems, it suffers from the curse of dimensionality for larger problems. Another disadvantage is the irregularity of the resulting optimal control time history.

Perhaps the most widely used method of obtaining optimum flight profiles is the gradient method.[4,5] Although the gradient solution is widely used,[6,7] it has the disadvantages of requiring considerable computation time and necessitating the derivation of complex formulations. This last requirement has been reduced considerably by some investigators,[8,9] by using repeated solutions of the equations of motion to approximate the gradient of the state with respect to the control, instead of using the more complex formulation involving the adjoint equations.

Although these methods are available for obtaining optimum flight profiles, there remains the need for faster and more economical solutions. It was with such a goal in mind that the NASA Flight Research Center applied Balakrishnan's epsilon technique[10] to the problem of optimum profiles for airplanes.

In Ref. 10, Balakrishnan suggests adding a term to the cost function to be minimized, which penalizes departures from the equations of motion, and then treating the state as well as the control as unknown independent variables. The increase of unknown variables is more than compensated for by a marked reduction in the amount of computation necessary to compute the gradient of the cost with respect to the state and control variables. In the same reference, Balakrishnan shows that if the more widely used form of the problem has a solution, the modified form does also, and the solutions become equivalent as the weighting of the additional term of the cost function becomes infinite.

The present paper shows the detailed formulation of the technique in applying it to the minimum-time-to-climb problem. The results for an example solution using the epsilon method are compared with solutions obtained by using other methods. Computation times are also presented.

Statement of the Problem

Given the dynamic equation of motion,

$$\dot{V} = \frac{F - D}{\mathbf{m}} - g\sin\gamma = r_1(h,V) + r_2(h,V)L^2 - g\sin\gamma$$

$$\dot{h} = V\sin\gamma$$

where

$$L = \mathbf{m}V\dot{\gamma} + \mathbf{m}g\cos\gamma$$

$$r_1 = (F - D)/\mathbf{m} \text{ for } L = 0$$

$$r_2 = \partial[(F - D)/\mathbf{m}]/\partial L^2$$

find the control input $\gamma(t)$ which minimizes the time T required to pass from the initial conditions V_1, h_1, γ_1 to the final conditions V_N, h_N, γ_N.

Gradient Technique

It is instructive to discuss first how the problem could be solved by using the gradient technique so that the differences in applying the epsilon technique can be more easily understood. In applying the gradient technique, the cost to be minimized might include not only time but a penalty for missing the final conditions. Theoretically, it is necessary for the penalty function for missing the end condition to approach infinity or a slight miss of the end condition will result. In practice, however, this is of little concern. The time is a function of the state and corresponds to the time at which the state is closest (minimum miss penalty) to meeting the final end conditions. So we have

$$J = T(V,h) + P(V,h)$$

The gradient of the cost with respect to the control $\nabla_\gamma J$ is computed by first taking the gradient with respect to the state and multiplying by the gradient of the change in state due to perturbations in the control $\gamma(t)$. The latter gradient can be computed as a solution to a system of differential equations or approximated by repeatedly solving the equations to determine changes in state due to perturbations in the control. The recursive formula

$$\gamma_{k+1}(t) = \gamma_k(t) + e\nabla_\gamma J$$

is then used to iterate until the control $\gamma(t)$ converges to that which minimizes the cost J, provided a suitable value of the scalar e is used.

Balakrishnan's Epsilon Technique

In Ref. 10, Balakrishnan suggests an alternative which eliminates the need to solve the system of differential equations. This is done by inserting a penalty in the cost function for deviations in the dynamic equations instead of a penalty for missing the final conditions.

With the addition of the term representing the constraint of satisfying the equations of motion, the cost function to be minimized becomes

$$J = \frac{1}{\epsilon}\left[\int_0^T \left(\dot{V} - \frac{F - D}{\mathbf{m}} + g\sin\gamma\right)^2 dt + \int_0^T (\dot{h} - V\sin\gamma)^2 dt\right] + \int_0^T dt$$

which can be written for the discrete case as

$$J = \frac{1}{\epsilon}\left\{\sum_{n=1}^{N}\left[\dot{V}_n - \left(\frac{F - D}{\mathbf{m}}\right)_n + g\sin\gamma_n\right]^2 \Delta t + \sum_{n=1}^{N}(\dot{h}_n - V_n\sin\gamma_n)^2\Delta t\right\} + \sum_{n=1}^{N}\Delta t$$

It is convenient to express the cost as

$$J = \bar{\delta}^T\bar{\delta}$$

where $\bar{\delta}$ is a $(2N + 1)$ column vector

$$\bar{\delta} = \begin{bmatrix} \left(\frac{\Delta t}{\epsilon}\right)^{1/2} \delta\dot{V}_n \\ \left(\frac{\Delta t}{\epsilon}\right)^{1/2} \delta\dot{h}_n \\ (N\Delta t)^{1/2} \end{bmatrix}$$

The velocity and altitude errors at each time point are given by the following expressions

$$\delta\dot{V}_n = \dot{V}_n - [(F - D)/\mathbf{m}]_n + g\sin\gamma_n$$

$$\delta\dot{h}_n = \dot{h}_n - V_n\sin\gamma_n$$

The independent variables to be determined are the velocity, altitude, flight-path angle, and time increment, V_n, h_n, γ_n, and Δt for $n = 1, N$.

It should be noted that, whereas in the gradient method the equations of motions are exactly satisfied and there is a penalty for missing the end conditions, in the epsilon method, the end conditions are exactly met and there is a penalty for not satisfying the equations of motion.

It is necessary for the value of epsilon to approach zero (resulting in an infinitely increasing penalty for not satisfying the dynamic equations) to theoretically attain the exact solution. This is analogous to the infinite penalty for missing the end condition that is required for an exact solution using the gradient method. In practice, the value of epsilon is of little importance, since the differences in the solutions that correspond to different values are negligible. Numerical examples are discussed later to support this statement.

Initially, a gradient solution for the epsilon technique was attempted; however, this approach was less than satisfactory. Although the cost was reduced by a few orders of magnitude, a false or apparent convergence was encountered which prevented solutions from being obtained.

A modified Newton-Raphson method was then used to circumvent the convergence problem, but it was necessary to first reduce the dimensionality of the problem. In order to reduce the dimensionality, the state and control variables were represented by a functional expansion as follows:

$$V(t) = V_1 + \frac{V_N - V_1}{T}t + \sum_{m=1}^{M} a_m \sin\frac{m\pi t}{T} = V_1 + \frac{V_N - V_1}{T}t + \sum_{m=1}^{M} a_m f_m(t)$$

$$h(t) = h_1 + \frac{h_N - h_1}{T}t + \sum_{m=1}^{M} b_m \sin\frac{m\pi t}{T} = h_1 + \frac{h_N - h_1}{T}t + \sum_{m=1}^{M} b_m f_m(t)$$

$$\gamma(t) = \gamma_1 + \frac{\gamma_N - \gamma_1}{T}t + \sum_{m=1}^{M} c_m \sin\frac{m\pi t}{T} = \gamma_1 + \frac{\gamma_N - \gamma_1}{T}t + \sum_{m=1}^{M} c_m f_m(t)$$

where $t = (n - 1)\Delta t$ and $T = (N - 1)\Delta t$ and, therefore, in the discrete case this becomes

$$V_n = V_1 + \frac{V_N - V_1}{N - 1}(n-1) + \sum_{m=1}^{M} a_m \sin\frac{m\pi(n-1)}{N-1} = V_1 + \frac{V_N - V_1}{N-1}(n-1) + \sum_{m=1}^{M} a_m f_{mn}$$

The derivatives of these functions become

$$\dot{V}(t) = \frac{V_N - V_1}{T} + \sum_{m=1}^{M} a_m \frac{m\pi}{T}\cos\frac{m\pi t}{T} = \frac{V_N - V_1}{T} + \sum_{m=1}^{M} a_m \dot{f}_m(t)$$

where

$$\dot{f}_m(t) = \frac{m\pi}{T}\cos\frac{m\pi t}{T}$$

and in the discrete case

$$\dot{V}_n = \frac{V_N - V_1}{T} + \sum_{m=1}^{M} a_m \frac{m\pi}{N-1}\cos\frac{m\pi(n-1)}{N-1} = \frac{V_N - V_1}{T} + \sum_{m=1}^{M} a_m \dot{f}_{mn}$$

where

$$\dot{f}_{mn} = [m\pi/(N-1)]\cos m\pi[(n-1)/(N-1)]$$

The problem, then, is to find the values for the coefficients a_m, b_m, and c_m and the time T which minimize the cost. It should be pointed out that the number of time points N remains constant throughout the solution, and therefore the time T is changed by changing Δt.

Consider linearizing the vector $\bar{\delta}$ in terms of changes in the vector $\bar{c}$. For small changes in the vector $\bar{c}$, $\bar{\delta}$ can be expressed as follows:

$$\bar{\delta} = \bar{\delta}_0 + \nabla_{\bar{c}}\bar{\delta}\Delta\bar{c}$$

where $\bar{c}$ is the $(3M + 1)$ column vector

$$\bar{c} = \begin{bmatrix} a_1 \\ \cdot \\ \cdot \\ a_M \\ b_1 \\ \cdot \\ \cdot \\ b_M \\ c_1 \\ \cdot \\ \cdot \\ c_M \\ \Delta t \end{bmatrix}$$

The gradient $\nabla_{\bar{c}}\bar{\delta}$ is a $2N + 1$ by $3M + 1$ matrix, as follows:

$$\nabla_{\bar{c}}\bar{\delta} = \begin{bmatrix} \dfrac{\partial\left(\frac{\Delta t}{\epsilon}\right)^{1/2}\delta\dot{V}_n}{\partial a_m} & \dfrac{\partial\left(\frac{\Delta t}{\epsilon}\right)^{1/2}\delta\dot{V}_n}{\partial b_m} & \dfrac{\partial\left(\frac{\Delta t}{\epsilon}\right)^{1/2}\delta\dot{V}_n}{\partial c_m} & \dfrac{\partial\left(\frac{\Delta t}{\epsilon}\right)^{1/2}\delta\dot{V}_n}{\partial\Delta t} \\ \dfrac{\partial\left(\frac{\Delta t}{\epsilon}\right)^{1/2}\delta\dot{h}_n}{\partial a_m} & \dfrac{\partial\left(\frac{\Delta t}{\epsilon}\right)^{1/2}\delta\dot{h}_n}{\partial b_m} & \dfrac{\partial\left(\frac{\Delta t}{\epsilon}\right)^{1/2}\delta\dot{h}_n}{\partial c_m} & \dfrac{\partial\left(\frac{\Delta t}{\epsilon}\right)^{1/2}\delta\dot{h}_n}{\partial\Delta t} \\ \dfrac{\partial(N\Delta t)^{1/2}}{\partial a_m} & \dfrac{\partial(N\Delta t)^{1/2}}{\partial b_m} & \dfrac{\partial(N\Delta t)^{1/2}}{\partial c_m} & \dfrac{\partial(N\Delta t)^{1/2}}{\partial\Delta t} \end{bmatrix}$$

where $n = 1, N$ and $m = 1, M$ and

$$\frac{\partial(\Delta t/\epsilon)^{1/2}\,\delta\dot{V}_n}{\partial a_m} = \left(\frac{\Delta t}{\epsilon}\right)^{1/2}\left[\dot{f}_{mn} - \left(\frac{\partial(F-D)/\mathbf{m}}{\partial V}\right)_n f_{mn}\right]$$

where

$$\left(\frac{\partial(F-D)/\mathbf{m}}{\partial V}\right)_n = \left(\frac{\partial r_1}{\partial V}\right)_n + \left(\frac{\partial r_2}{\partial V}\right)_n L_n^2 + 2r_2L_n\mathbf{m}\dot{\gamma}_n$$

$$\partial\,(\Delta t/\epsilon)^{1/2}\,\delta\dot{h}_n/\partial a_m = -\,(\Delta t/\epsilon)^{1/2}(\sin\gamma_n f_{mn})$$

$$\partial(N\Delta t)^{1/2}/\partial a_m = 0$$

$$\partial(\Delta t/\epsilon)^{1/2}\delta\dot{V}_n/\partial b_m = -(\Delta t/\epsilon)^{1/2}\{[(\partial(F-D)/\mathbf{m}]/\partial h\}_n f_{mn}]$$

where

$$(\partial F - D/\mathbf{m}/\partial h)_n = (\partial r_1/\partial h)_n + (\partial r_2/\partial h)_n L_n^2$$

$$\frac{\partial\,(\Delta t/\epsilon)^{1/2}\,\delta\dot{h}_n}{\partial b_m} = \left(\frac{\Delta t}{\epsilon}\right)^{1/2}(\dot{f}_{mn})$$

$$\frac{\partial(N\Delta t)^{1/2}}{\partial b_m} = 0$$

$$\frac{\partial(\Delta t/\epsilon)^{1/2}\,\delta\dot{V}_n}{\partial c_m} = \left(\frac{\Delta t}{\epsilon}\right)^{1/2}\times [(2r_{2_n}L_n\mathbf{m}g\sin\gamma_n + g\cos\gamma_n)f_{mn} - 2r_{2_n}\mathbf{m}V_nL_n\dot{f}_{mn}]$$

$$\frac{\partial(\Delta t/\epsilon)^{1/2}\,\delta\dot{h}_n}{\partial c_m} = -\left(\frac{\Delta t}{\epsilon}\right)^{1/2}(V_n\cos\gamma_n f_{mn})$$

$$\frac{\partial(N\Delta t)^{1/2}}{\partial c_m} = 0$$

$$\frac{\partial(\Delta t/\epsilon)^{1/2}\,\delta\dot{V}_n}{\partial\Delta t} = \frac{\delta V_n}{2(\epsilon\Delta t)^{1/2}} - \frac{\dot{V}_n}{(\epsilon\Delta t)^{1/2}} + \frac{r_{2_n}2L_n\mathbf{m}V_n\dot{\gamma}_n}{(\Delta t\epsilon)^{1/2}}$$

$$\partial(\Delta t/\epsilon)^{1/2}\,\delta\dot{h}_n/\partial\Delta t = \delta\dot{h}_n/2(\epsilon\Delta t)^{1/2} - \dot{h}_n/\epsilon\Delta t$$

$$\partial(N\Delta t)^{1/2}/\partial\Delta t = \tfrac{1}{2}(N/\Delta t)^{1/2}$$

Then J becomes

$$J = \bar{\delta}_0^T\bar{\delta}_0 + 2\bar{\delta}_0^T\nabla_{\bar{c}}\bar{\delta}\Delta\bar{c} + \Delta\bar{c}^T\nabla_{\bar{c}}\bar{\delta}^T\nabla_{\bar{c}}\bar{\delta}\Delta\bar{c}$$

Setting the gradients of the cost to zero

$$\nabla_{\bar{c}}J = 0 = 2\nabla_{\bar{c}}\bar{\delta}^T\bar{\delta}_0 + 2\nabla_{\bar{c}}\bar{\delta}^T\nabla_{\bar{c}}\bar{\delta}\Delta\bar{c}$$

and solving for the increments to $\bar{c}$ we get

$$\Delta\bar{c} = -[\nabla_{\bar{c}}\bar{\delta}^T\nabla_{\bar{c}}\bar{\delta}]^{-1}\nabla_{\bar{c}}\bar{\delta}^T\bar{\delta}_0$$

The previous expression is that for the Newton-Raphson method[11] except for the omission of the second gradient which diminishes to zero as the solution is approached. Since $\bar{\delta}$ was linearized, the second gradient of $\bar{\delta}$ was taken to be zero. Had this term not been omitted, the result would have been the complete formulation of the Newton-Raphson method. The authors believe that the inclusion of the additional term does not justify the added computation and complexity.

Application of Epsilon Technique to Sample Problem

Because it was desired to compare this technique with other methods, an example was chosen for which existing solutions were available. The problem chosen for comparison was the problem investigated by Bryson in the study reported in Ref. 5, that of computing the flight profile which a supersonic interceptor should follow to reach a given final altitude and velocity in minimum time. The performance data for this problem were provided by Bryson.

One of the earlier difficulties with the epsilon technique was the inconsistent behavior of the cost function with repeated iterations. Typically, the cost would increase after the first iteration, then decrease rapidly for a few iterations, and then oscillate so that there was little or no improvement in the cost function with additional iterations. To improve the convergence of the solution, it was suggested by Balakrishnan that constant terms be added to the diagonal elements of the $[\nabla_c\bar{\delta}^T\nabla_c\bar{\delta}]$ matrix before solving for the increments to the coefficients. This was done, and the results were dramatic. It was found that adding small numbers to the diagonal elements did indeed keep the cost function from increasing; however, the convergence was extremely slow. By experimenting with these numbers, values were found that resulted in a well-behaved function with relatively rapid convergence. The terms which were added to the diagonal elements were a percentage of the existing values of the diagonal elements. These percentages were denoted by λ. The effect of these terms is illustrated in Fig. 1, which shows the cost and time to reach the terminal conditions as a function of numbers of iterations for several values of λ. It is shown that by increasing the diagonal elements by as little as 0.01% the behavior of the cost function is dramatically improved. The solution for $\lambda = 0$ is not shown because the oscillations were too large for the plot; however, the solution for $\lambda = 0.0025$ % is shown to demonstrate the effect. The reason for the considerable effect on convergence of adding very small amounts to the diagonal of the system of equations is not yet fully understood by the authors. The phenomenon is partially caused by round-off errors in inverting nearly singular matrices. Currently, experimentation is required to select the most suitable value of λ.

Both the time to reach the terminal condition and the total cost which includes this time are shown in Fig. 1. The difference between the cost curves and the time curves represents the error in satisfying the equations of motion. For the $\lambda = 0.01\%$ solution, the cost is reduced to 490 sec and the time is approximately 398 sec.

It would be expected that increasing the number of functions allowed in the functional expansions for velocity, altitude, and flight-path angle would result in better fits of these quantities and therefore lower cost. However, there is a tradeoff here in that the more functions used the greater the computation time for obtaining a solution. Figure 2 presents the cost plotted against the number of functions used in the expansion. It is seen that, up to 15 functions, the reduction in cost is significant with increasing functions. Also shown in this figure is the time to reach the terminal conditions. The number of functions is shown to have little effect on this parameter. For the example of 15 functions, the amount that the total cost is in excess of the time represents an error in $\dot{V}$ of approximately 4% and an error in $\dot{h}$ of approximately 0.1%.

Figure 3 shows the effect of the number of functions on the computation time, using an SDS 9300 computer. Here it is noted that the computation time is significantly increased with an increase in the number of functions. Because computation time is an important factor in optimization

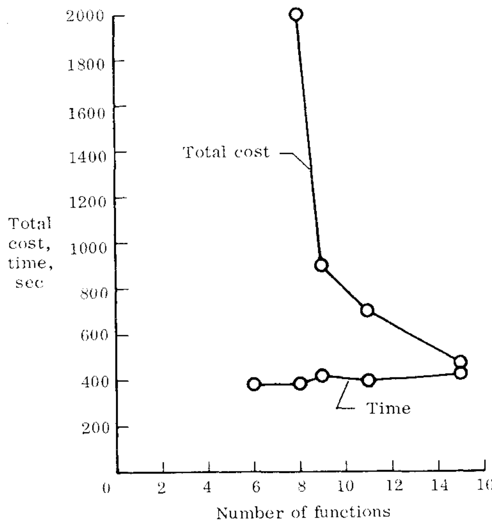

Fig. 2 Effect of number of functions on total cost and time.

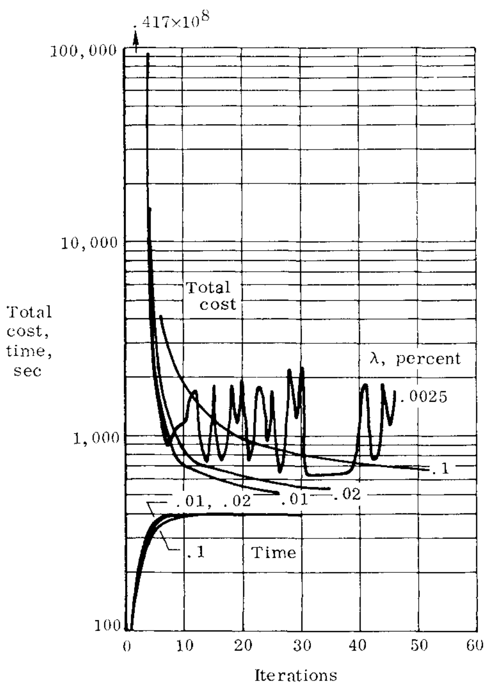

Fig. 1 Effect of the empirical factor λ on convergence of the total cost and time.

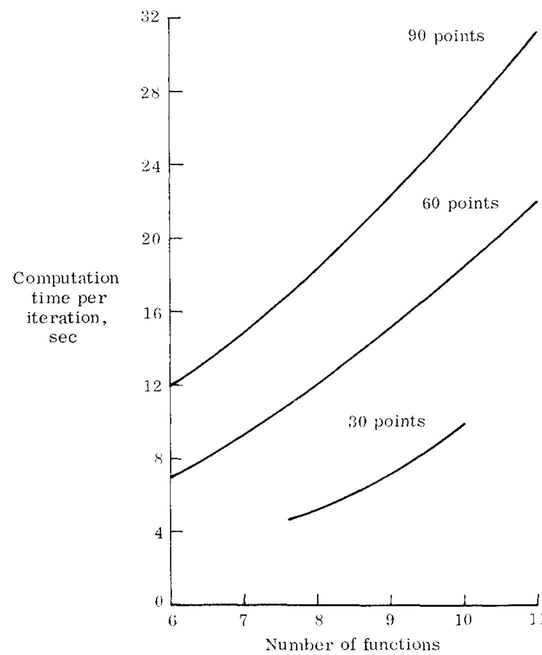

Fig. 3 Effect of number of time points and number of functions on computation time (SDS 9300 digital computer).

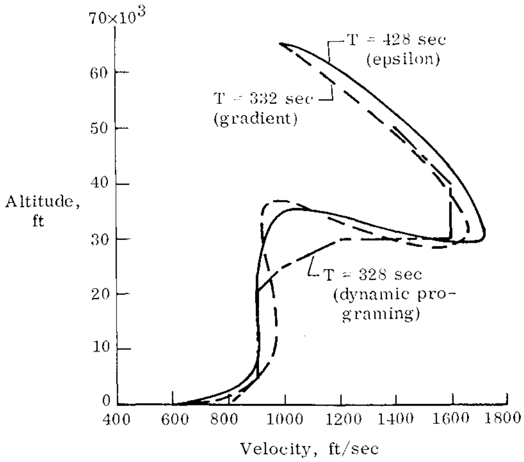

Fig. 4 Comparison of minimum time profiles obtained by using three different methods.

problems, this must be considered when choosing the number of functions to be used.

An increase in epsilon would decrease the relative weighting of the equations-of-motion term in the cost function and allow less than the actual value of the minimum time to be computed. It would be expected, therefore, that decreasing epsilon would result in a more accurate value of minimum time. The results for several values of epsilon are given in Table 1. Values of epsilon considerably beyond the values given in the table resulted in computational difficulties. Also shown is the effect of epsilon on the total cost. It is seen that the results in terms of both time and total cost are relatively insensitive to epsilon over a wide range.

The resulting profile in terms of altitude vs velocity for the example presented is shown by the solid curve in Fig. 4. Also shown, by the dashed curve, are the results of the gradient method of Ref. 5. The two methods are seen to give similar profiles, although the times to reach the end conditions differ considerably. Use of the gradient§ method resulted in a time of 332 sec, whereas in the present investigation the time was determined to be 428 sec. It should be pointed out, however, that the thrust data were somewhat lower for the latter investigation. It is not known if the difference in thrust is large enough to account for the entire difference in time.

With the dynamic programing method, which is shown in Fig. 4 by the long-short-long dashes, the desired terminal conditions were not reached because the solution for the missing portion of the flight was not possible under the assumptions made in the dynamic programing formulation. The time to reach 50,000-ft altitude and a velocity of 1400 fps was 328 sec for the dynamic programing solution compared with 356 sec for the epsilon technique. The performance data used in the dynamic programing solution were the same data as were used in the epsilon method; however, additional simplifying assumptions were made in the dynamic programing case. All maneuvering was done at the nodes, and steady-state flight was assumed between the nodes. These assumptions can result in limiting the trajectory if the steady-state equations do not have solutions for all conditions.

Table 1 Effect of epsilon on minimum time and cost

Epsilon	Time	Cost
0.01	397.94	490.52
0.1	397.89	490.43
1.0	397.59	490.14
10.0	394.46	487.43
100.0	365.28	504.92

Presented in Table 2 is a summary of several available optimization programs which shows the computation time required and the computer used in each instance. The time ratio shown in the last column is the real time (i.e., the time to reach the terminal conditions) divided by the computation time. It is evident from this table that the epsilon technique promises a considerable savings in computation time compared with the gradient solutions for airplane-type problems, whereas Balakrishnan's epsilon technique is about 20 times as fast. It should be pointed out, however, that the examples given in the table are intended to indicate only grossly the computation time required. None of the gradient programs[5–9, 12–14] has been written for the sole purpose of generating rapid solutions.

One of the problems with the gradient method has been the choice of a suitable initial trajectory. With the epsilon technique, almost any initial trajectory will suffice and therefore a straight line connecting the end points is used for convenience. This results in a considerable time savings which is not reflected in the times given in Table 2. The evolution of the altitude-velocity profile by means of the epsilon technique shown in Fig. 4 is presented in Fig. 5. Shown is the starting profile, which is the straight line, and the first, sixth, and final trajectories.

Admittedly, much more work than the example discussed is needed to show the full potential of the epsilon technique, but the results thus far have been most promising. Although the analytical procedures outlined in this paper are for the simple case illustrated, the technique is not limited to this problem. Optimizations involving a more accurate atmosphere, changing weight, fuel consumption, range, state constraints, and limited control pose no particular problem. However, it must be demonstrated that the advantages of the epsilon technique are realized for these problems.

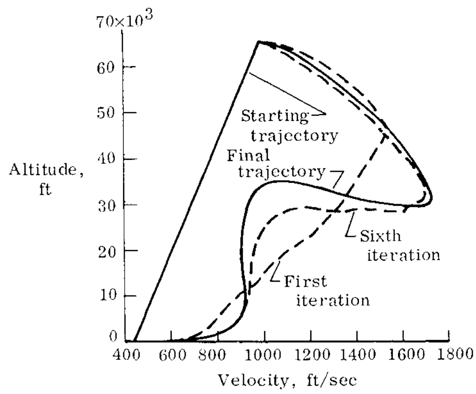

Fig. 5 Evolution of minimum time profile using the epsilon technique and starting from straight line.

§ Details concerning the sample gradient solutions were supplied by A. E. Bryson of Stanford University.

Concluding Remarks

Balakrishnan's epsilon technique has been successfully applied to the problem of determining optimum flight profiles for airplanes.

It was necessary to use the Newton-Raphson method to achieve satisfactory convergence properties. A functional expansion of the state and control variables reduced the dimensionality of the problem to a size amenable to a small digital computer.

It was found that by adding suitable constants to the diagonal elements of the $[\nabla_c\bar{\delta}^T\nabla_c\bar{\delta}]$ matrix, the solutions were well-behaved and converged rapidly. Fifteen functions were found to give good results.

The effect of varying epsilon was investigated and found to be very small over a wide range of values.

Table 2 Computation times for optimization programs

Technique	Ref.	Solution time, sec	Computer Add	Computer Multiply	Date	Problem	Time points	Iterations	Time ratio
Gradient	5 Bryson	?	IBM	704	1962	Airplane	?	7	?
Gradient	8 Schmidt	480	CDC 0.4 μsec	6600 1.0 μsec	1968	Hypersonic	250	10	0.5
Gradient	9 Cockayne	120	IBM 14 μsec	7090 24 μsec	1968	Airplane	10	10	1.3
Gradient	13 "Presto"	40	CDC 0.4 μsec	6600 1.0 μsec	1965	Booster	. . .	20	22.0
Gradient	7 "TOP" 6 "STOP"	250	CDC 0.4 μsec	6600 1.0 μsec	1967	Airplane	204	20	0.9
Gradient	12 "TOS"	900	IBM	7094	1964	Re-entry	. . .	20	1.0
Gradient—calculus of variation	14 Rempfer	120	CDC 0.4 μsec	6600 1.0 μsec	1966	Airplane	120	20	0.5
Epsilon	10 Balakrishnan	280	SDS 14 μsec	9300 12.3 μsec	1968	Airplane	30	6	0.5
Epsilon	Present report	14	CDC 0.4 μsec	6600 1.0 μsec	1968	Airplane	51	9	30.0
Epsilon	Present report	117	SDS 14 μsec	9300 12.3 μsec	1968	Airplane	34	9	3.4
Dynamic programing	Present report	5	SDS 14 μsec	9300 12.3 μsec	1968	Airplane	10	"3"	63.0

Determining a suitable starting trajectory is not a problem with the epsilon method as it is for the gradient method, inasmuch as a straight line between the initial and the final conditions can be used.

Because of the considerable savings in computation time, Balakrishnan's epsilon method is an attractive alternative to the classical gradient approach for determining optimum flight profiles.

References

[1] Lush, K. J., "A Review of the Problem of Choosing a Climb Technique, With Proposals for a New Climb Technique for High Performance Aircraft," R & M 2557, 1951, British Aeronautical Research Council.

[2] Rutowski, E. S., "Energy Approach to the General Aircraft Performance Problem," *Journal of the Aeronautical Sciences*, Vol. 21, March 1954, pp. 187–195.

[3] Bellman, R. and Kalaba, R., "*Dynamic Programming and Modern Control Theory*," Academic Press, New York, 1965.

[4] Kelly, H. J., "Gradient Theory of Optimal Flight Paths," *ARS Journal*, Vol. 30, No. 10, Oct. 1960, pp. 947–954.

[5] 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.

[6] Stein, L. H., Matthews, M. L., and Frenk, J. W., "Stop—A Computer Program for Supersonic Transport Trajectory Optimization," CR-793, 1967, NASA.

[7] Matthews, M. L. and Watson, D. A., "Trajectory Optimization Program (Top) AS 2081, Analytical Development and User's Manual," Vol. 1, No. D2-125323-1, Feb. 1968, The Boeing Co.

[8] Schmidt, H. E., Helgason, R. V., Witherspoon, J. T., and Geib, K. E., "Trajectory Optimization by Method of Steepest Descent," AFFDL-TR-67-108, Vol. II, April 1968, U.S. Air Force.

[9] Cockayne, W. G., "X-15 Optimal Control Study," Rept. 2500-950002, April 1968, Bell Aerosystems.

[10] Balakrishnan, A. V., "On A New Computing Technique in Optimal Control and its Application to Minimal Time Flight Profile Optimization," presented at 2nd International Colloquium on Methods of Optimization, Novosibirsk, USSR, June 1968.

[11] Balakrishnan, A. V., *Communication Theory*, McGraw-Hill, 1968.

[12] "TOS—Theoretical Optimization Study," Part I—Optimum Programming Formulation (Government Printing Office No. X65-10934), Sept. 1964, Raytheon Co.

[13] Willwerth, R. E., Jr., Rosenbaum, R. C., and Chuck, W., "PRESTO: Program for Rapid Earth-to-Space Trajectory Optimization," CR-158, 1965, NASA.

[14] Rempfer, P. S., "Combination of Calculus of Variation and Steepest Ascent to Construct Optimum Trajectory," presented at 2nd NASA Intercenter Symposium on Control Theory, Huntsville. Ala., Sept. 1966.